

TECHNISCHE UNIVERSITÄT MÜNCHEN

**Requirements Engineering für IT-gestützte
Product Service Systems**

Marina Berkovich

TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik

Lehrstuhl für Wirtschaftsinformatik (I 17)

Univ.-Prof. Dr. Helmut Krcmar

Requirements Engineering für IT-gestützte Product Service Systems

Marina Berkovich

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grads eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Dr. h.c. Manfred Broy

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Helmut Krcmar
2. Univ.-Prof. Dr. Jan Marco Leimeister,
Universität Kassel

Diese Dissertation wurde am 16.08.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 06.08.2012 angenommen.

Zusammenfassung

Heutzutage zeichnet sich ein erhöhter Wettbewerbsdruck ab, der dazu führt, dass eine Differenzierung von den Wettbewerbern alleine durch Sach- oder Dienstleistungen nicht mehr ausreicht. Daher bieten Unternehmen integrierte Bündel von Sach- und Dienstleistungen an, die das Ziel verfolgen, ein Kundenproblem zu lösen. Diese Bündel sind unter dem Begriff *Hybrides Produkt* oder *Product Service System (PSS)* bekannt. Im Falle von IT-gestützten PSS ist Software ein wichtiger Bestandteil der Sachleistung, die das Bindeglied zwischen Hardware und Dienstleistungen realisiert. Auf Grund des Ziels ein Kundenproblem zu lösen, spielt das Requirements Engineering in der Entwicklung von PSS eine wichtige Rolle. In der Literatur und Praxis fehlt bislang eine Methodik zum integrierten Requirements Engineering für Sach- und Dienstleistungen, das die Erfordernisse von PSS erfüllt. Daher stellt sich die Frage, wie das Requirements Engineering für PSS gestaltet werden soll, um die Anforderungen an PSS domänenübergreifend und integriert zu ermitteln und zu verwalten.

Zur Bearbeitung dieser Fragestellung wurde die gestaltungsorientierte Forschung als Grundlage verwendet. Im ersten Schritt wird das Problem anhand von Literaturanalysen, empirischen Studien und vertiefenden Fallstudien in der Praxis näher beleuchtet. Basierend darauf, wird im nächsten Schritt ein Ansatz zum Requirements Engineering für IT-gestützte PSS entwickelt, der das zuvor festgestellte Problem löst. Im dritten Schritt der Forschung wird der entwickelte Ansatz, anhand von analytischen Methoden und Fallstudien, evaluiert.

Der entwickelte Requirements Engineering Ansatz setzt sich aus einem Artefaktmodell und einem Vorgehensmodell zusammen. Das Artefaktmodell beschreibt die Inhalte einer Spezifikation von Anforderungen und gibt Strukturierungsprinzipien für Anforderungen vor. Dadurch unterstützt es die interdisziplinäre Erhebung und Konkretisierung von Anforderungen an PSS in Abstimmung mit dem Entwicklungsprozess. Das Vorgehensmodell definiert Aktivitäten, die im Rahmen des Requirements Engineering ausgeführt werden, um die Anforderungen zu erheben und zu verwalten. Für jede Aktivität schlägt das Vorgehensmodell zudem Techniken vor, die zur Durchführung der Aktivität verwendet werden. Der Requirements Engineering Ansatz liefert als Ergebnis konkrete Anforderungen, die von den an der Entwicklung des PSS beteiligten Domänen umgesetzt werden. Die Evaluation des Ansatzes durch analytische Methoden hat gezeigt, dass er die an ihn gestellten Anforderungen erfüllt. Anhand von Fallstudien in der Praxis konnte gezeigt werden, dass der Ansatz für die Praxis tauglich ist und zur Lösung von bestehenden Herausforderungen beiträgt.

Der theoretische Beitrag des entwickelten Ansatzes ist in Form eines Referenzmodells zu verstehen, da das Artefaktmodell ein allgemeingültiges Modell darstellt, das an die Erfordernisse von einzelnen Unternehmen angepasst werden kann. Aus der Sicht der Praxis unterstützt der entwickelte Ansatz die frühen Entwicklungsphasen von PSS. Eine Einschränkung des Ansatzes besteht darin, dass er sich hauptsächlich auf IT-gestützte PSS bezieht, die Hardware enthalten. Weiterer Forschungsbedarf besteht für PSS die nur aus Software und Dienstleistungen bestehen. Auch das Variantenmanagement für PSS, sowie die Nachvollziehbarkeit der Umsetzung der Anforderungen, stellen Themen für die zukünftige Forschung dar.

Die vorliegende Arbeit stellt einen ersten integrierten Ansatz zum Requirements Engineering für IT-gestützten PSS dar. Er realisiert eine Konkretisierung der Anforderungen unter Berücksichtigung der Besonderheiten, der an der Entwicklung von PSS beteiligten Domänen. Damit trägt der Ansatz dazu bei, Herausforderungen in der Praxis zu lösen und das Verständnis von PSS zu verbessern.

Abstract

Nowadays the competitive pressure increases, with the consequence that the differentiation from competitors by goods or services alone is no longer sufficient. For that reason, companies offer integrated bundles of tangible goods and services, which aim to solve a customer's problem holistically. These bundles are known as *Hybrid Products* or *Product Service Systems (PSS)*. In case of IT-enabled PSS, the software constitutes an important part of the tangible goods, by realizing the link between the hardware and services elements. Due to their goal of solving a customers' problem, requirements engineering plays an essential role in the development process of PSS. In literature as well as in practice, a methodology for an integrated requirements engineering for hardware and services, fulfilling the needs of PSS, is still missing. Therefore, the question arises, how requirements engineering for PSS should be designed in order to elicit and manage the requirements on PSS in an integrated way and across all domains.

For answering this question, design science is used as a basis. In the first step, the problem is examined in depth by means of literature analyses, empirical studies and case studies in practice. Subsequently, an approach for requirements engineering for IT-enabled PSS is developed, solving the problem determined previously. In the third step of the research, the developed approach is evaluated, using analytical methods and case studies.

The requirements engineering approach described in this thesis comprises an artifact model and a procedure model. The artifact model describes the contents of a requirements specification and defines structuring principles for the requirements to PSS. Thus, it supports the interdisciplinary elicitation and concretization of the requirements in accordance with the development process. The procedure model specifies all activities, being performed by requirements engineering in order to elicit and manage the requirements to PSS. Additionally, the procedure model suggests suitable techniques for the realization of each activity. This requirements engineering approach provides concretized requirements that can be implemented by the domains involved in the development of PSS. The analytical evaluation of the requirements engineering approach has shown that all requirements imposed on it are satisfied. By means of case studies, it could be demonstrated that the approach is also suitable in practice and contributes to solving existing challenges.

The theoretical contribution of the approach can be seen as a reference model, because the artifact model is a universal model that can be adapted to the needs of individual companies. From the practical point of view, this approach supports the early stages of the development process of PSS. A limitation of the approach is that it mainly refers to IT-enabled PSS including hardware components. Therefore, a topic for further research are PSS, consisting of solely software and service elements. Moreover, the variant management for PSS as well as the traceability of requirements' implementation are topics for future research.

This thesis presents a first integrated approach for requirements engineering for IT-enabled PSS. It realizes the concretization of requirements considering the characteristics of the domains participating in the development of PSS. Thereby, the approach contributes to solving the challenges in practice and improving the understanding of PSS.

Inhaltsverzeichnis

Zusammenfassung	i
Abstract	ii
Inhaltsverzeichnis	iii
Abbildungsverzeichnis	xi
Tabellenverzeichnis	xv
Abkürzungsverzeichnis	xvii
1 Zielsetzung und Aufbau der Arbeit	1
1.1 Motivation und Ziele der Arbeit	1
1.2 Forschungsleitende Fragestellungen	4
1.3 Ablauf und Aufbau der Arbeit	5
2 Wissenschaftstheoretische Grundlagen	9
2.1 Einordnung der Arbeit in die Wirtschaftsinformatik	9
2.2 Forschungsmethoden und Forschungsdesign	10
2.2.1 Methodisches Rahmenkonzept	10
2.2.2 Verwendetes Forschungsdesign	14
2.2.3 Fazit und Implikationen für den weiteren Gang der Arbeit	16
3 Merkmale von Product Service Systems	17
3.1 Begriffliche Grundlagen	17
3.1.1 Überblick zu Definitionen von Product Service Systems	17
3.1.1.1 Abgrenzung des Begriffs „Produkt“	17
3.1.1.2 Abgrenzung der Begriffe „Sachleistung“ und „Dienstleistung“	18
3.1.1.3 Abgrenzung des Begriffs „Product Service Systems“	23
3.1.1.4 Bedeutung von Product Service Systems für die Unternehmen	29
3.1.1.5 Zusammenfassung	32
3.1.2 Eigenschaften von Product Service Systems	32
3.2 Product Service Systems in der Praxis	34
3.2.1 Product Service Systems aus dem Blickwinkel von Anbieter und Kunden	34
3.2.2 Beispiele für Product Service Systems	35
3.3 Lebenszyklenperspektive von Product Service Systems	38
3.3.1 Entwicklungsprozess von Product Service Systems	40
3.3.2 Zyklische Wechselwirkungen im Lebenszyklus von Product Service Systems	42

3.4	Zusammenfassung.....	44
4	Requirements Engineering im Kontext von Product Service Systems	47
4.1	Bedeutung des Requirements Engineering	47
4.2	Begriffliche Grundlagen	50
4.2.1	Anforderung.....	50
4.2.2	Requirements Engineering und Management.....	52
4.3	Aktivitäten des Requirements Engineering im Lebenszyklus von Product Service Systems	55
4.4	Stand der Wissenschaft: Requirements Engineering für Product Service Systems in der Literatur	56
4.4.1	Aufbau der Literaturanalyse	57
4.4.1.1	Zielsetzung der Literaturanalyse	57
4.4.1.2	Auswahl der Literatur.....	58
4.4.1.3	Verwandte Arbeiten über die Literaturanalyse zum Requirements Engineering.....	62
4.4.1.4	Ordnungsrahmen zur Darstellung des Requirements Engineering - Gegenstandes für Product Service Systems.....	62
4.4.1.5	Kriterien für die Literaturanalyse und Bewertung der Literatur.....	65
4.4.2	Requirements Engineering in Produkt-, Software- und Dienstleistungsentwicklung.....	67
4.4.2.1	Requirements Engineering in der Produktentwicklung.....	67
4.4.2.1.1	Münchener Vorgehensmodell (Lindemann 2006).....	67
4.4.2.1.2	Vorgehenszyklus für die Lösungssuche (Ehrlenspiel 2002)	69
4.4.2.1.3	Methodisches Klären und Präzisieren der Aufgabe (Pahl et al. 2006).....	70
4.4.2.1.4	Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte (VDI 2221, 1993).....	72
4.4.2.1.5	Das Erfassen und Handhaben von Produkthanforderungen (Ahrens 2000).....	73
4.4.2.1.6	Zusammenfassung	74
4.4.2.2	Requirements Engineering in der Softwareentwicklung	77
4.4.2.3	Requirements Engineering in der Dienstleistungsentwicklung.....	83
4.4.2.3.1	Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen (Bullinger und Schreiner 2003)	83
4.4.2.3.2	Key Concepts for New Service Development (Edvardsson und Olsson 1996).....	84

4.4.2.3.3	Service Design (Ramaswamy 1996).....	84
4.4.2.3.4	Systematische Dienstleistungsentwicklung (Fritzsche 2007).....	86
4.4.2.3.5	Anforderungsanalyse für produktbegleitende Dienstleistungen (van Husen 2007).....	87
4.4.2.3.6	Zusammenfassung	88
4.4.2.4	Ansätze zur integrierten Entwicklung von Product Service Systems.....	89
4.4.2.5	Ergebnisse der Literaturanalyse.....	90
4.4.3	Zyklische Wechselwirkungen im Requirements Engineering für Product Service Systems in der Literatur	97
4.5	Requirements Engineering für Product Service Systems in der Praxis	97
4.5.1	Vorstudie I: Requirements Engineering für Product Service Systems in der Praxis - Eine Exploration von Problemen und Herausforderungen	98
4.5.1.1	Methodik und Aufbau der Vorstudie.....	98
4.5.1.1.1	Datenerhebung.....	99
4.5.1.1.2	Inhaltsanalyse	102
4.5.1.2	Auswertung der Untersuchungsergebnisse.....	103
4.5.1.2.1	Anforderungsermittlung	103
4.5.1.2.2	Anforderungsanalyse	104
4.5.1.2.3	Anforderungsvereinbarung.....	105
4.5.1.2.4	Anforderungsdokumentation	106
4.5.1.2.5	Anforderungvalidierung.....	106
4.5.1.2.6	Änderungsmanagement und Anforderungsverfolgung	107
4.5.1.2.7	Fazit: Herausforderungen im Requirements Engineering für Product Service Systems in der Praxis	107
4.5.2	Vorstudie II: Requirements Engineering für Product Service Systems in der Praxis – Eine vertiefende Fallstudie.....	108
4.5.2.1	Zielsetzung, Ablauf, Aufbau und Methodik der Vorstudie.....	108
4.5.2.2	Auswertung und Vergleich der Untersuchungsergebnisse.....	111
4.5.2.2.1	Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha.....	111
4.5.2.2.2	Herausforderungen im Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha resultierend aus den Experteninterviews.....	112
4.5.2.2.3	Herausforderungen im Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und	

	Trocknern bei Alpha resultierend aus der Analyse von Dokumenten.....	113
4.5.2.2.4	Herausforderungen im Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha resultierend aus den Fragebögen	116
4.5.2.2.5	Fazit: Zusammenfassung der Ergebnisse der vertiefenden Fallstudie im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha.....	116
4.5.3	Zyklische Wechselwirkungen im Requirements Engineering für Product Service Systems in Vorstudie I und Vorstudie II.....	117
4.6	Zyklusmanagement im Requirements Engineering für Product Service Systems	118
4.7	Zusammenfassung: Anforderungen an einen Requirements Engineering Ansatz für Product Service Systems	128
5	Requirements Engineering Ansatz für Product Service Systems.....	137
5.1	Lösungsidee zur Realisierung der Anforderungen an einen Requirements Engineering Ansatz für Product Service Systems.....	137
5.2	Abgrenzung von existierenden Requirements Engineering Ansätzen	140
5.2.1	Szenario-basiertes Requirements Engineering	140
5.2.2	Ziel-orientiertes Requirements Engineering.....	140
5.2.3	Aspekt-orientiertes Requirements Engineering.....	141
5.2.4	Viewpoint-orientiertes Requirements Engineering	141
5.2.5	Prozess-orientiertes Requirements Engineering.....	142
5.2.6	Fazit	142
5.3	Lösungskonzept für Requirements Engineering für Product Service Systems	143
5.3.1	Bezugsrahmen zur Beschreibung des Lösungskonzepts für das Requirements Engineering für Product Service Systems.....	143
5.3.2	Beschreibung des Bezugsrahmens für das Requirements Engineering für Product Service Systems.....	144
6	Artefaktmodell für die Anforderungen an Product Service Systems	147
6.1	Grundlagen des Artefaktmodells für Anforderungen an Product Service Systems	147
6.1.1	Begriffliche Grundlage	147
6.1.2	Abgrenzung zu den existierenden artefakt-orientierten Ansätzen des Requirements Engineering	148
6.1.2.1	Requirements Abstraction Model (Gorschek/Wohlin 2006).....	148

6.1.2.2	Requirements Engineering Reference Model (Geisberger 2005; Geisberger et al. 2006; Berenbach et al. 2009).....	149
6.1.2.3	COSMOD-RE (Pohl/Sikora 2007a; Pohl/Sikora 2007b)	149
6.1.2.4	Artefaktmodell zur integrierten Wiederverwendung von Anforderungen und technischer Architektur (Penzenstadler/Koss 2008).....	150
6.1.2.5	Meta-Modell für Artefaktmodelle und zwei Instanzen (Méndez Fernández et al. 2010)	151
6.1.2.6	Fazit	152
6.1.3	Prinzipien des Artefaktmodells für Anforderungen an Product Service Systems.....	152
6.1.3.1	Strukturierung von Artefakten.....	153
6.1.3.2	Aufbau des Artefaktmodells in Abstraktionsebenen	153
6.1.3.3	Detaillierungsgrad von Anforderungen	154
6.1.3.4	Flexibler Einsatz des Artefaktmodells.....	154
6.1.3.5	Iterativ-inkrementelle Konkretisierung von Anforderungen.....	154
6.1.3.6	Verfolgung von Anforderungen	156
6.1.3.7	Änderungsmanagement	157
6.1.3.8	Umgang mit neuen Anforderungen	157
6.2	Metamodell des Artefaktmodells für Anforderungen an Product Service Systems	157
6.2.1	Abstraktionsmodell.....	158
6.2.2	Artefakt-Strukturmodell	159
6.2.3	Artefakt-Spezifikationsmodell.....	161
6.3	Beschreibung des Artefaktmodells für die Anforderungen an Product Service Systems	161
6.3.1	Abstraktionsebene 1: Zielebene.....	164
6.3.1.1	Anforderungsartefakte der Zielebene	166
6.3.1.2	Externe Artefakte der Zielebene.....	166
6.3.1.3	Spezifikationsartefakt der Zielebene	168
6.3.2	Abstraktionsebene 2: Systemebene	168
6.3.2.1	Anforderungsartefakte Systemebene.....	170
6.3.2.1.1	Taxonomien.....	171
6.3.2.2	Externe Artefakte der Systemebene	173
6.3.2.3	Spezifikationsartefakte der Systemebene	174
6.3.2.3.1	Anforderungsdokumente	174

6.3.2.3.2	Szenarien	175
6.3.2.3.3	Konzeptionelle Modelle	176
6.3.2.3.4	Auswahl der geeigneten Spezifikationsform	177
6.3.3	Abstraktionsebene 3: Eigenschaftsebene.....	177
6.3.3.1	Entwicklungsartefakte der Eigenschaftsebene	179
6.3.3.1.1	Systemkontext	180
6.3.3.1.2	Funktionsstrukturen: Hauptfunktionen.....	180
6.3.3.1.3	Zusammenfassung	182
6.3.3.2	Anforderungsartefakte der Eigenschaftsebene	182
6.3.3.3	Externe Artefakte der Eigenschaftsebene.....	188
6.3.3.4	Spezifikationsartefakte der Eigenschaftsebene	188
6.3.4	Abstraktionsebene 4: Funktionsebene	190
6.3.4.1	Entwicklungsartefakte der Funktionsebene.....	191
6.3.4.2	Anforderungsartefakte der Funktionsebene.....	192
6.3.4.3	Externe Artefakte der Funktionsebene	194
6.3.4.4	Spezifikationsartefakte der Funktionsebene	194
6.3.5	Abstraktionsebene 5: Komponentenebene	194
6.3.5.1	Entwicklungsartefakte der Komponentenebene	196
6.3.5.2	Anforderungsartefakte der Komponentenebene.....	196
6.3.5.3	Externe Artefakte der Komponentenebene.....	197
6.3.5.4	Spezifikationsartefakte der Komponentenebene	197
6.4	Zusammenfassung.....	197
7	Vorgehensmodell für das Requirements Engineering für Product Service Systems.....	199
7.1	Begriffliche Grundlagen	199
7.2	Metamodells des Vorgehensmodells	199
7.3	Vorgehensmodell für das Requirements Engineering für Product Service Systems	200
7.3.1	Ermittlung und Analyse von Zielen an PSS	203
7.3.1.1	Aktivitäten zur Ermittlung und Analyse von Zielen an PSS	203
7.3.1.2	Techniken zur Ermittlung und Analyse von Zielen an PSS	205
7.3.2	Ermittlung und Analyse von PSS-Anforderungen	206
7.3.2.1	Aktivitäten zur Ermittlung und Analyse von PSS-Anforderungen	206
7.3.2.2	Techniken zur Ermittlung von Anforderungen an PSS	211

7.3.2.2.1	Traditionelle Techniken.....	211
7.3.2.2.2	„Group Elicitation“-Techniken.....	213
7.3.2.2.3	Prototyping	214
7.3.2.2.4	Modell-basierte Techniken	215
7.3.2.2.5	Kognitive Techniken	215
7.3.2.2.6	Kontextuelle Techniken.....	216
7.3.2.3	Techniken zur Anforderungsvereinbarung.....	217
7.3.2.4	Techniken zur Anforderungskonkretisierung.....	218
7.3.2.5	Techniken zur Anforderungsverfolgung	218
7.3.3	Ermittlung und Analyse von Design-Anforderungen.....	219
7.3.3.1	Aktivitäten zur Ermittlung und Analyse von Design-Anforderungen.....	219
7.3.3.1.1	Prinzip der wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung.....	221
7.3.3.1.2	Phase 1: Aufbau der Funktionsstruktur	224
7.3.3.1.3	Phase 2: Aufteilung von Anforderungen nach materiellen und immateriellen Bestandteilen des PSS	226
7.3.3.2	Techniken zur Ermittlung und Analyse von Design-Anforderungen.....	227
7.3.4	Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen.....	228
7.3.4.1	Aktivitäten zur Ermittlung und Analyse von Funktionsstruktur- Anforderungen und Domänen-Anforderungen	228
7.3.4.1.1	Phase 3: Iterative Konkretisierung von Anforderungen.....	230
7.3.4.1.2	Phase 4: Iterative Verfeinerung von Funktionen.....	231
7.3.4.2	Techniken zur Ermittlung und Analyse von Funktionsstruktur- Anforderungen und Domänen-Anforderungen	234
7.4	Zusammenfassung.....	235
8	Evaluation des Requirements Engineering Ansatzes für Product Service Systems.....	237
8.1	Ziele, Kriterien und Methoden der Evaluation	237
8.2	Analytische Evaluation des Requirements Engineering Ansatz für PSS.....	239
8.3	Evaluation des Requirements Engineering Ansatzes für PSS anhand eines Anwendungsfalls: Clean Laundry on Demand	242
8.4	Merkmalbasierte Evaluation des Requirements Engineering Ansatzes für PSS	250
8.5	Evaluation des Requirements Engineering Ansatzes anhand von Fallstudien.....	253

8.5.1	Fallstudie 1: Retrospektive Anwendung des Requirements Engineering Ansatzes auf das internetgestützte Bewegungsprogramm „Personal Health Manager“	253
8.5.1.1	Evaluationsvorgehen	253
8.5.1.2	Ergebnisse der Evaluation	254
8.5.1.2.1	Schritt 1: Kriterien festlegen.....	254
8.5.1.2.2	Schritt 2: Anwenden des Requirements Engineering Ansatzes.....	255
8.5.1.3	Diskussion der Ergebnisse der Evaluation	258
8.5.2	Fallstudie 2: Retrospektive Anwendung des Requirements Engineering Ansatzes im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha.....	261
8.5.2.1	Evaluationsvorgehen	261
8.5.2.2	Ergebnisse der Evaluation	262
8.5.2.3	Diskussion der Ergebnisse der Evaluation	265
8.5.2.3.1	Ergebnisse der Experteninterviews	265
8.5.2.3.2	Ergebnisse des Fragebogens zu den merkmalsbasierten Kriterien	268
8.5.2.3.3	Fazit der Expertenbewertung.....	268
8.6	Zusammenfassung.....	269
9	Fazit und Ausblick.....	271
9.1	Zusammenfassung der Problemstellung und Ergebnisse der Arbeit	271
9.2	Implikationen für die Praxis und die Forschung.....	273
9.3	Grenzen der Arbeit und weiterer Forschungsbedarf.....	274
	Literaturverzeichnis.....	277
10	Anhang	305

Abbildungsverzeichnis

Abbildung 1-1:	Ablauf der Arbeit (FF: Forschungsfrage).....	5
Abbildung 1-2:	Aufbau der Arbeit.....	7
Abbildung 2-1:	Taxonomie für Forschungsmethoden	11
Abbildung 2-2:	Vorgehensmodell der gestaltungsorientierten Forschung	13
Abbildung 3-1:	Leistungs- bzw. Produktarten	18
Abbildung 3-2:	Betriebswirtschaftlicher Dienstleistungsbegriff	20
Abbildung 3-3:	Dimensionen einer Dienstleistung.....	21
Abbildung 3-4:	Unterschiede beim Absatz von Sach-, Dienstleistungen und PSS	23
Abbildung 3-5:	Traditionelles Angebot an Produkten und Dienstleistungen versus PSS ..	25
Abbildung 3-6:	Unterschiedliche Bedarfsdeckung bei Sachleistungen, hybriden Produkten und Dienstleistungen	26
Abbildung 3-7:	Gütertypologische Abgrenzung von PSS	26
Abbildung 3-8:	Kategorien von Product Service Systems.....	28
Abbildung 3-9:	Finanzieller Erfolgsanteil von Dienstleistungen am Gesamterfolg der Unternehmen	30
Abbildung 3-10:	Bedeutung von Lösungen zur Kundenbindung	30
Abbildung 3-11:	Beitrag von Lösungen zur Differenzierung gegenüber Wettbewerbern....	31
Abbildung 3-12:	Bedeutung der integrierten Entwicklung von DL und Produkten	31
Abbildung 3-13:	Sterilisationskomplettlösungen für Kliniken.....	36
Abbildung 3-14:	Document management solution	36
Abbildung 3-15:	Personal Health Manager als IT-gestütztes, gerätebasiertes und personalisiertes Gesundheitsförderungsprogramm	38
Abbildung 3-16:	Problemlösungsbezug von PSS	39
Abbildung 3-17:	Lebenszyklus von PSS	40
Abbildung 3-18:	Entwicklungsprozess von PSS	41
Abbildung 3-19:	Zyklen bei der Gas-Versorgungslösung - Änderungen von Kundenanforderungen	43
Abbildung 3-20:	Zyklen bei der Gas-Versorgungslösung - Missverständnisse beim Klären von Anforderungen.....	44
Abbildung 4-1:	Projekterfolgsquoten aus den Standish Group (CHAOS-) Studien der Jahre 1994-2004	48
Abbildung 4-2:	Gründe für Ressourcenüberschreitungen und/oder Funktionseinschränkungen bei Projekten.....	49
Abbildung 4-3:	Verschiedene Sichtweisen auf Anforderungen.....	51
Abbildung 4-4:	Aktivitäten des Requirements Engineering	53
Abbildung 4-5:	Zusammenspiel zwischen dem Lebenszyklus von PSS und Requirements Engineering	56
Abbildung 4-6:	Ordnungsrahmen zur Darstellung des Requirements Engineering - Gegenstandes für Product Service Systems.....	64
Abbildung 4-7:	Das Münchner Vorgehensmodell	67
Abbildung 4-8:	Vorgehenszyklus für die Lösungssuche	69
Abbildung 4-9:	Planen und Klären der Aufgabe.....	71
Abbildung 4-10:	Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte.....	72
Abbildung 4-11:	Das Erfassen und Handhaben von Produkthanforderungen	74
Abbildung 4-12:	Konkretisierung von Anforderungen.....	79

Abbildung 4-13:	Analyse der Requirements Engineering Ansätze in Bezug auf ihre Eignung für PSS	95
Abbildung 4-14:	Ergebnisse der Analyse der Requirements Engineering Ansätze anhand der Kriterien	96
Abbildung 4-15:	Ablauf des empirischen Forschungsprozesses zur Erhebung und Interpretation qualitativen Daten	99
Abbildung 4-16:	Ablauf der qualitativen Inhaltsanalyse	102
Abbildung 4-17:	Ablauf der Datenerhebung	109
Abbildung 4-18:	Ablauf des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha	112
Abbildung 4-19:	Zyklus im Requirements Engineering für PSS ausgelöst durch die Verfügbarkeit und Reife der Technologie	120
Abbildung 4-20:	Zyklus im Requirements Engineering für PSS ausgelöst durch die wettbewerblichen Entwicklungen	122
Abbildung 4-21:	Zyklus im Requirements Engineering für PSS ausgelöst durch die Unterschiedliche Lebenszyklen von Hardware- und Software-Komponenten des PSS	123
Abbildung 4-22:	Zyklus im Requirements Engineering für PSS ausgelöst durch Änderungen an Kunden- und Stakeholderanforderungen	124
Abbildung 4-23:	Zyklus im Requirements Engineering für PSS ausgelöst durch finanzwirtschaftliche Änderungen	125
Abbildung 4-24:	Zyklus im Requirements Engineering für PSS ausgelöst durch Entwicklungsänderungen	126
Abbildung 4-25:	Zyklus im Requirements Engineering für PSS ausgelöst durch legislative Änderungen	127
Abbildung 4-26:	Zyklus im Requirements Engineering für PSS ausgelöst durch Missverständnisse über Anforderungen	127
Abbildung 5-1:	Kooperation am gemeinsamen Material	138
Abbildung 5-2:	Gestaltung der Kommunikation zwischen den Beteiligten anhand eines Artefakts	138
Abbildung 5-3:	Bezugsrahmen für das Requirements Engineering für PSS	145
Abbildung 6-1:	Prinzip der iterativ-inkrementellen Konkretisierung von Anforderungen an PSS	155
Abbildung 6-2:	Vorgehen bei der Anforderungskonkretisierung für PSS	156
Abbildung 6-3:	Metamodell des Artefaktmodells für die Anforderungen an PSS	158
Abbildung 6-4:	Artefaktmodell für die Anforderungen an PSS	162
Abbildung 6-5:	Artefakte der Zielebene	165
Abbildung 6-6:	Artefakte der Systemebene	169
Abbildung 6-7:	Taxonomie der Anforderungsartefakte der Systemebene	172
Abbildung 6-8:	Artefakte der Eigenschaftsebene	178
Abbildung 6-9:	Systemkontext und Systemgrenze eines PSS	180
Abbildung 6-10:	Taxonomie der Produktanforderungen der Eigenschaftsebene	183
Abbildung 6-11:	Taxonomie der prozessorientierten Anforderungen der Eigenschaftsebene	185
Abbildung 6-12:	Taxonomie der ressourcenorientierten Anforderungen der Eigenschaftsebene	187
Abbildung 6-13:	Artefakte der Funktionsebene	190

Abbildung 6-14:	Taxonomie der detaillierten produktorientierten Anforderungen der Funktionsebene	193
Abbildung 6-15:	Artefakte der Komponentenebene	195
Abbildung 7-1:	Metamodell des Vorgehensmodell für das Requirements Engineering für PSS	200
Abbildung 7-2:	Schematische Darstellung des Vorgehensmodells für das Requirements Engineering für PSS	201
Abbildung 7-3:	Iterative gegenseitige Analyse und Konkretisierung von Anforderungen.....	202
Abbildung 7-4:	Aktivitäten der Ermittlung und Analyse von Zielen an das PSS.....	204
Abbildung 7-5:	Aktivitäten der Ermittlung und Analyse von PSS-Anforderungen	207
Abbildung 7-6:	Aktivitäten der Ermittlung und Analyse von Design-Anforderungen....	220
Abbildung 7-7:	Prinzip der wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung.....	222
Abbildung 7-8:	Zuordnung von Anforderungen zu den Funktionen	223
Abbildung 7-9:	Phase 1: Aufbau der Funktionsstruktur	225
Abbildung 7-10:	Phase 2: Aufteilung von Anforderungen nach materiellen und immateriellen Leistungsbündeln.....	226
Abbildung 7-11:	Aktivitäten zur Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen	229
Abbildung 7-12:	Phase 3: Konkretisierung von Anforderungen	230
Abbildung 7-13:	Iterative Zuordnung von Anforderungen zu Funktionen.....	231
Abbildung 7-14:	Phase 4: Iterative Verfeinerung von Funktionen.....	232
Abbildung 7-15:	Schematische Darstellung der iterativen wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung	233
Abbildung 7-16:	Metamodell des Requirements Engineering Ansatzes für PSS	236
Abbildung 8-1:	Aufbau der Evaluation des Requirements Engineering Ansatzes für PSS	239
Abbildung 8-2:	Funktionsstruktur des PSS „Laundry on Demand“	244
Abbildung 8-3:	DSM zur Bestimmung der kausalen Abhängigkeiten für die Hauptfunktionen des PSS „Laundry on Demand“	245
Abbildung 8-4:	Clean Laundry on Demand.....	245
Abbildung 8-5:	Ausschnitt aus der DMM der Zuordnung von Anforderungen zu Funktionen	246
Abbildung 8-6:	Detaillierte Funktionsstruktur der Funktion „Wäsche waschen“	247
Abbildung 8-7:	DSM zur Bestimmung der kausalen Abhängigkeiten für die Sub-Funktionen der Funktion „Wäsche waschen“	247
Abbildung 8-8:	Detaillierte Funktionsstruktur der Funktion „Prozessschritte Planen“	248
Abbildung 8-9:	Detaillierte Darstellung der Funktion „Lauge erzeugen“	249
Abbildung 8-10:	Schematische Ableitung von Anforderungen an das PSS „Clean Laundry on Demand“	250
Abbildung 8-11:	Schematische Ableitung von Anforderungen an PHM	258
Abbildung 8-12:	Ergebnis der Fragebogenumfrage zur Bewertung des Requirements Engineering Ansatzes für PSS anhand von PHM.....	259
Abbildung 8-13:	Schematische Ableitung von Anforderungen an das PSS „Waschlösung“	265

Abbildung 8-14:	Ergebnis der Fragebogenumfrage zur Bewertung des Requirements Engineering Ansatzes für PSS im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha	268
Abbildung 10-1:	Kategorisierung nicht-funktionaler Anforderungen	311

Tabellenverzeichnis

Tabelle 2-1:	Richtlinien für die Erfüllung des Design-Science-Paradigma	12
Tabelle 2-2:	Verwendete Forschungsmethoden nach den Phasen des Vorgehensmodells nach Takeda et al. (1990) und Forschungsfragen	15
Tabelle 3-1:	Unterschiede zwischen Sach- und Dienstleistungen	19
Tabelle 3-2:	Definitionen von Product Service Systems	24
Tabelle 3-3:	Begriffsdefinitionen im Rahmen dieser Arbeit	32
Tabelle 4-1:	Analysierte Beiträge zum Requirements Engineering aus der Produktentwicklung	60
Tabelle 4-2:	Analysierte Beiträge zum Requirements Engineering aus der Softwareentwicklung.....	60
Tabelle 4-3:	Analysierte Beiträge zum Requirements Engineering aus der Dienstleistungsentwicklung	61
Tabelle 4-4:	Analysierte Beiträge zum Requirements Engineering in den integrierten Ansätzen zur integrierten Entwicklung von PSS	61
Tabelle 4-5:	Zuordnung von Unternehmen	100
Tabelle 4-6:	Übersicht zu den Experten-Interviews	101
Tabelle 4-7:	Designelemente der Fallstudien und deren Ausprägung.....	109
Tabelle 4-8:	Übersicht zu den analysierten Unterlagen.....	110
Tabelle 4-9:	Zyklusursachen im Requirements Engineering für PSS	119
Tabelle 4-10:	Anforderungen an einen Requirements Engineering Ansatz für Product Service Systems.....	135
Tabelle 6-1:	Externe Artefakte der Zielebene	167
Tabelle 6-2:	Externe Artefakte der Systemebene	174
Tabelle 8-1:	Methoden der Artefaktevaluation in der gestaltungsorientierten Forschung	238
Tabelle 8-2:	Geschäftsziel des Kunden an das PSS „Clean Laundry on Demand“.....	243
Tabelle 8-3:	PSS-Anforderungen an „Clean Laundry on Demand“.....	244
Tabelle 8-4:	Design-Anforderungen an „Clean Laundry on Demand“	246
Tabelle 8-5:	Funktionsstruktur-Anforderungen an „Clean Laundry on Demand“.....	248
Tabelle 8-6:	Domänen-Anforderungen an „Clean Laundry on Demand“	249
Tabelle 8-7:	Geschäftsziel des Kunden an PHM.....	255
Tabelle 8-8:	PSS-Anforderung an PHM.....	255
Tabelle 8-9:	Design-Anforderungen an PHM	256
Tabelle 8-10:	Funktionsstruktur-Anforderungen an PHM	257
Tabelle 8-11:	Domänen-Anforderungen an PHM	257
Tabelle 8-12:	Initiale Anforderungen aus dem Lastenheft des Unternehmensbereichs zur Herstellung von Waschmaschinen und Trocknern bei Alpha.....	262
Tabelle 8-13:	Anforderungen an das PSS „Waschlösung“	263
Tabelle 8-14:	Ziele an das PSS „Waschlösung“	263
Tabelle 8-15:	Funktionsstruktur-Anforderungen an das PSS „Waschlösung“	264
Tabelle 8-16:	Domänen-Anforderungen an das PSS „Waschlösung“	264
Tabelle 10-1:	Interviewleitfaden zum Ist-Zustand des Requirements Engineering in Vorstudie I.....	306
Tabelle 10-2:	Leitfaden zum Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern von Unternehmen Alpha in Vorstudie II	307

Tabelle 10-3:	Fragebogen zum Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern von Unternehmen Alpha in Vorstudie II	308
Tabelle 10-4:	Spezifikation von Zielen an PSS	309
Tabelle 10-5:	Systematisierung von Dienstleistungen	310
Tabelle 10-6:	Hauptmerkmalliste zur Charakterisierung von Anforderungen an Hardware-Komponenten	312
Tabelle 10-7:	Fragebogen zur Evaluation des Requirements Engineering Ansatzes für PSS	315
Tabelle 10-8:	Fragebogen zur Evaluation des Requirements Engineering Ansatzes für PSS	316
Tabelle 10-9:	Interviewleitfaden zur Evaluation des Requirements Engineering Ansatzes für PSS im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern in Unternehmen Alpha.....	318

Abkürzungsverzeichnis

B2C	Business-to-Consumer
COSMOD-RE	Scenario and Goal based System Development Method
DMM	Domain-Mapping-Matrix
DSM	Design Structure Matrix
EPK	Ereignisgesteuerte Prozessketten
ER-Diagramme	Entity-Relationship-Diagramme
HCI	Human-Computer-Interaction
HW	Hardware
ICSE	International Conference on Software Engineering
MVM	Münchner Vorgehensmodell
PHM	Personal Health Manager
PSS	Product Service Systems
QFD	Quality Function Deployment
RAM	Requirements Abstraction Model
REM	Requirements Engineering Reference Model
REMBIS	Requirements Engineering und Management für Business Informationssysteme
REMsES	Requirements Engineering und Management für software-intensive Eingebettete Systeme
SaaS	Software as a Service
SW	Software
UML	Unified Modeling Language

1 Zielsetzung und Aufbau der Arbeit

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung und Evaluation eines Requirements Engineering Ansatzes für IT-gestützte Product Service Systems¹. In diesem Kapitel werden zunächst die Motivation für das Vorhaben erläutert sowie die forschungsleitenden Fragestellungen vorgestellt. Zuletzt werden der Aufbau und Ablauf der Arbeit erklärt.

1.1 Motivation und Ziele der Arbeit

Der Erfolg eines Unternehmens hängt vom Erfolg seiner Produkte oder Dienstleistungen auf dem Markt ab. Als Folge davon stehen Unternehmen unter ständigem Wettbewerbsdruck, der sich in verkürzten Entwicklungs- und Produktionszeiten, steigenden Kosten sowie hohen Anforderungen an die Qualität der Angebote des Unternehmens ausdrückt (Lindemann 2006, 7-9; Pulm 2004, 13-15; Weber et al. 2004; Abramovici/Schulte 2007). Dazu kommen der Wunsch der Unternehmen sich von den Wettbewerbern zu differenzieren (Leimeister/Glauner 2008; Böhm/Krcmar 2006), technologisch neuartige Produkte hervorzubringen und dem Kunden dadurch einen neuartigen Nutzen zu stiften (Hauschildt 2004, 34).

Die Differenzierung von den Wettbewerbern alleine durch Sach- oder Dienstleistungen reicht nicht mehr für einen langfristigen Erfolg aus. Viele Bestandteile der Sachleistungen können leicht imitiert, standardisiert und damit austauschbar gemacht werden (Nippa et al. 2007). Darüber hinaus hat sich die Marktsituation verändert - von angebotsorientiert („supply-driven“) zu nachfrageorientiert („demand-driven“), wo die Nachfrage des Kunden das Angebot bestimmt (Halen et al. 2005, 17). Die Kunden verlangen nicht nach Produkten oder Dienstleistungen per se, sondern nach einer Lösung für ihre Probleme (Berkovich et al. 2009a; Leimeister/Glauner 2008). Die Unternehmen sind dadurch gezwungen, ihre Innovationsfähigkeit ständig auszubauen, die steigende Komplexität der Angebote zu bewältigen und gleichzeitig die Kundenanforderungen möglichst vollständig zu erfüllen (Beiter et al. 2006; Nuseibeh/Easterbrook 2000).

Um diesen Herausforderungen gerecht zu werden, ändern viele Unternehmen ihre Philosophie von „product-centric“, in der ein technisches Produkt (Sachleistung) oder eine Dienstleistung das Geschäft dominiert, in „customer-centric“ (Galbraith 2002) und entwickeln sich zu so genannten Lösungsanbietern (Berkovich et al. 2009b; Doster/Roegner 2000). Die Unternehmen bieten daher integrierte Bündel von Sach- und Dienstleistungen an, die für den Kunden nicht als einzelne Komponenten erkennbar sind (Krcmar 2010, 600; Knackstedt et al. 2008; Böhm et al. 2008; Mont 2002). Solche Lösungen sind unter dem Begriff *Hybride Produkte* oder *Product Service Systems (PSS)* bekannt (Halen et al. 2005, 16ff.; Mont 2002; Böhm/Krcmar 2007). Dabei kann die Sachleistung sowohl Software alleine als auch eine Kombination aus Software und physikalischen technologischen Elementen (Hardware) sein (Tan et al. 2007). Im Rahmen dieser Arbeit werden PSS betrachtet, dessen Sachleistungskomponente aus einer Kombination aus Hardware und Software besteht.

Die einzelnen Bestandteile des PSS, wie Hardware, Software und Dienstleistungen, werden von den für sie zuständigen Domänen, wie Produkt-, Software- und Dienstleistungsentwick-

¹ Die hier vorgestellten Ergebnisse wurden im Rahmen des von der DFG geförderten SFB 768 – Zyklusmanagement von Innovationsprozessen - erarbeitet. Für weitergehende Informationen siehe www.sfb768.de.

lung, konzipiert und entwickelt. Im Falle von IT-gestützten PSS ist Software ein wichtiger Bestandteil der Lösung, denn die Software ist das Bindeglied zwischen Hardware und Dienstleistung und wird zur Bereitstellung von Basisfunktionalitäten, wie Speicherung, Verarbeitung und Kommunikation zur Verfügung stehender Ressourcen verwendet (Krcmar 2010, 30). Im weiteren Verlauf der Arbeit werden *IT-gestützte PSS* als *PSS* bezeichnet.

Ein Beispiel soll veranschaulichen, was ein PSS ist. Der Kunde wünscht sich eine konstante Raumtemperatur von 21°C. Dabei geht es dem Kunden um die Erfüllung seiner Wünsche und nicht um die Gestaltung des Heizkörpers oder der Software für die Temperatursteuerung. Daher stellt er die Anforderungen an die ganze Lösung und nicht an die einzelnen Komponenten der Lösung. Basierend auf den Anforderungen stellt der Lösungsanbieter dem Kunden ein integriertes und abgestimmtes Bündel aus Komponenten (z. B. Heizungsanlage) und Dienstleistungen (z. B. Wartung) zur Verfügung (vgl. (Berkovich et al. 2009c)).

Im Vordergrund der Entwicklung und des Anbietens von PSS steht eine möglichst maximale Erfüllung von Kundenanforderungen (Lönngren et al. 2008; Baines et al. 2007). Laut der von Sturm et al. (2007, 24) durchgeführten Befragung von 90 mittelständischen Unternehmen bezeichnen ca. 40% von ihnen die Art und Weise, wie die Kundenanforderungen in ihren Unternehmen behandelt werden, als unzureichend. Obwohl die Bedeutung von PSS für Kunden und Unternehmen erkannt ist (Sturm/Bading 2008), fehlen sowohl in der Literatur als auch in der Praxis Vorgehensmodelle zur integrierten Entwicklung von Sach- und Dienstleistungen als ein Ganzes (Sturm/Bading 2008; Leimeister/Glauner 2008). Dazu kommt, dass ca. 98% der von Sturm et al. (2007, 18) befragten Unternehmen kundenspezifische Kombinationen aus technischen Produkten und Dienstleistungen als sehr wichtig bezeichnen. Allerdings werden in nur ca. 20% der befragten Unternehmen Sach- und Dienstleistungen aufeinander abgestimmt und integriert entwickelt.

In diesem Zusammenhang spielt das Requirements Engineering, das die Anforderungen ermittelt, modifiziert und verwaltet, eine wichtige Rolle im Entwicklungsprozess von PSS (Berkovich et al. 2009a; Berkovich et al. 2009b; Spath/Demuß 2003). Sowohl in der Literatur als auch in der Praxis fehlen aber entsprechende Vorgehensmodelle für das Requirements Engineering, die in der Lage sind, die Anforderungen an das PSS in geeigneter Form zu behandeln. Dazu gehört es, die Anforderungen an das PSS als Ganzes und an die einzelnen domänenspezifischen Komponenten, Hardware, Software und Dienstleistungen, zu betrachten und sie in Anlehnung an den Entwicklungsprozess so zu konkretisieren, dass die an der Entwicklung von PSS beteiligten Domänen diese Anforderungen anschließend integriert und abgestimmt umsetzen können (Berkovich 2010; Berkovich et al. 2009a; Berkovich et al. 2011a; Berkovich et al. 2009b; Berkovich et al. 2009c). Die Domänen sollen dafür ein gemeinsames Verständnis über das zu konzipierende PSS entwickeln und sich gegenseitig bezüglich der Anforderungen abstimmen. Die Abstimmung zwischen den Domänen wird unterstützt, wenn die Anforderungen in Form eines gemeinsamen Materials (vgl. (Zerbe 1998, 213)) vorliegen. In diesem Fall können die beteiligten Domänen auf eine gemeinsame Basis zurückgreifen und dadurch ihre Wahrnehmung von Anforderungen stärken. So kann sichergestellt werden, dass das vom PSS zu lösende Problem und die damit in Zusammenhang stehenden Anforderungen an das PSS von den Domänen richtig und in gleicher Weise interpretiert werden.

Als Folge von unterschiedlichen Interpretationen von Anforderungen, unterschiedlichem Hintergrundwissen und unterschiedlichen Vorstellungen der Domänen über das zu entwickelnde

PSS, treten Iterationen beim Ermitteln und Klären von Anforderungen zwischen den Entwicklern, Anforderungsanalysten und Kunden auf. Diese Iterationen sind auch als Zyklen bekannt und führen zur Wiederholung von Aktivitäten und Ergebnissen von Aktivitäten des Requirements Engineering (Berkovich et al. 2011e). Solche Zyklen haben Verzögerungen beim Bereitstellen des PSS, Kundenunzufriedenheit und steigende Kosten, Zeit- und Ressourcenaufwand zur Folge (Berkovich et al. 2011e). Die Notwendigkeit, Iterationen durchzuführen, wird weiter verstärkt, da die verschiedenen Stakeholder die Anforderungen auf unterschiedliche Weise ausdrücken, aber ein gemeinsames Verständnis über die Anforderungen erzielen müssen. Die Kunden drücken ihre Anforderungen an das PSS üblicherweise in Form von unkonkreten und lösungsneutralen Zielen aus, während die Entwickler die Anforderungen in einer sehr konkreten und oft technisch ausgeprägten Form äußern.

Weitere Iterationen ergeben sich aufgrund der fehlenden Verbindungen zwischen den Anforderungen, die es ermöglichen sicherzustellen, dass die konkreten, lösungsorientierten Anforderungen an die domänenspezifischen Komponenten des PSS den initialen Wünschen und Vorstellungen der Kunden entsprechen. Die existierenden Requirements Engineering Ansätze stellen keine Unterstützung dafür bereit (Berkovich 2010; Berkovich et al. 2009a; Berkovich et al. 2011a; Berkovich et al. 2009b; Berkovich et al. 2009c). Schließlich besteht eine konzeptionelle Lücke zwischen den Anforderungen und der Entwicklung, die im Rahmen des Requirements Engineering für PSS zu schließen ist. Dazu müssen die initialen unkonkreten Anforderungen an das PSS konkretisiert werden, d. h. „in die Sprache des Entwicklers“ übersetzt werden, und es muss überprüft werden, ob diese konkretisierten Anforderungen von den jeweiligen Domänen, die für die Entwicklung der Komponenten des PSS zuständig sind, korrekt verstanden werden. Nur so kann gewährleistet werden, dass die für den Kunden relevanten und wichtigen Anforderungen richtig umgesetzt werden, das PSS die initialen Kundenanforderungen erfüllt und somit das bestehende Kundenproblem löst (Berkovich et al. 2011a).

Angesichts der oben genannten Herausforderungen ergibt sich die Notwendigkeit einen integrierten Ansatz zum Requirements Engineering zu entwickeln, der speziell auf PSS ausgerichtet ist. Um dieses Ziel zu erreichen, ist es als erstes notwendig Probleme und Herausforderungen im Requirements Engineering für PSS detailliert zu beleuchten. Ebenso sind die Zyklen, die während des Requirements Engineering auftreten, zu identifizieren. Dazu sind empirische Untersuchungen nötig, die sich auf die Analyse des Standes der Wissenschaft und Praxis beziehen. Anhand von Literaturanalysen zum Requirements Engineering in den integrierten Ansätzen zur Entwicklung von PSS und in den einzelnen Domänen soll der Stand der Wissenschaft untersucht werden. Empirische Studien in Form von Interviews und Fallstudien in der Praxis sollen dabei den Stand der Praxis zum Requirements Engineering für PSS klären. Basierend auf den Ergebnissen der empirischen Untersuchungen werden die Anforderungen an einen Requirements Engineering Ansatz für PSS ermittelt.

Das Hauptziel dieser Arbeit besteht darin, einen Requirements Engineering Ansatz für PSS zu entwickeln, der die identifizierten Probleme und Herausforderungen löst. Dieser Ansatz soll in der Lage sein, die Anforderungen an das PSS als Ganzes zu betrachten, sie laut den Phasen des Entwicklungsprozesses zu strukturieren und zu konkretisieren, und in geeigneter Form an die an der Entwicklung von PSS beteiligten Domänen zur weiteren Umsetzung zu übergeben. Dabei sollen die Entwicklungsinformationen einbezogen werden, um den Entwurf des PSS und die Anforderungen schon in den frühen Phasen der Entwicklung aufeinander abzustimmen. Die Zyklen, die im Requirements Engineering auftreten und zu Iterationen führen, sind ebenso im Rahmen des zu entwickelnden Ansatzes zu adressieren.

1.2 Forschungsleitende Fragestellungen

Das Ziel des Requirements Engineering ist es, die Anforderungen an ein zu entwickelndes System zu ermitteln, zu verwalten und diese Anforderungen nahtlos an die Entwicklung zu übergeben (Nuseibeh/Easterbrook 2000; Jung 2006, 2ff.).

Im wissenschaftlichen und praktischen Schrifttum fehlen umfassende Ansätze zum Requirements Engineering für PSS weitgehend. Die meisten Ansätze zur Entwicklung von PSS behandeln das Requirements Engineering nur am Rande und auf einer abstrakten Ebene (vgl. Spath und Demuß (2003), Kortmann (2007), Steinbach (2005)). Obwohl die Notwendigkeit integrierter Ansätze zum Requirements Engineering für PSS sowohl in der Praxis als auch in der Wissenschaft erkannt wurde, werden in den meisten Ansätzen zur Entwicklung von PSS nur Gestaltungsvorschläge zur Anforderungsermittlung und -verwaltung gemacht, ohne konkret auf deren Ausgestaltung einzugehen (Berkovich et al. 2009a; Berkovich et al. 2009b; Berkovich et al. 2009c). Die exakte Definition der Requirements Engineering Prozessschritte, der im Rahmen des Requirements Engineering zu erarbeitenden Artefakte und die Identifikation der am Requirements Engineering beteiligten Akteure sind Aspekte, die für die Anwendung der Requirements Engineering Ansätze in der Praxis relevant sind.

Im Mittelpunkt dieser Arbeit steht ein Ansatz zum Requirements Engineering für PSS. Um so einen Ansatz zu entwickeln, sind zunächst die bestehenden Ansätze zum Requirements Engineering in den an der Entwicklung von PSS beteiligten Domänen, wie Produkt-, Software- und Dienstleistungsentwicklung, aufzuarbeiten. Die Erfahrungen im Requirements Engineering aus der Praxis der Unternehmen, die PSS entwickeln, sollen den Kenntnisstand aus der Literatur erweitern und vertiefen. Dabei ist es essentiell, die besonderen Bedürfnisse an das Requirements Engineering für PSS zu verstehen und daraus Anforderungen an einen Ansatz zum Requirements Engineering für PSS abzuleiten. Daher lautet die erste Forschungsfrage:

1) Was ist der Stand der Wissenschaft und Praxis des Requirements Engineering für PSS und was sind die Anforderungen an einen Requirements Engineering Ansatz für PSS?

Aufbauend auf Forschungsfrage 1 verfolgt Forschungsfrage 2 das Ziel, einen Requirements Engineering Ansatz für PSS zu entwickeln. Zu diesem Zweck soll ein Bezugsrahmen erstellt werden, der allgemeingültig gehalten wird und die einzelnen Gestaltungsobjekte des Requirements Engineering Ansatzes sowie die Beziehungen zwischen ihnen definiert, ohne auf ihre konkrete Ausgestaltung einzugehen (vgl. Böhm und Fuchs (2002), Brocke (2003)). Durch die konkrete Ausgestaltung aller Elemente des Bezugsrahmens, unter Einbezug der Anforderungen an einen Ansatz zum Requirements Engineering für PSS aus der ersten Forschungsfrage, wird der Ansatz zum Requirements Engineering für PSS vollständig definiert. Forschungsfrage 2 lautet daher:

2) Welche Elemente und Eigenschaften besitzt ein Requirements Engineering Ansatz für PSS, der die Anforderungen aus Forschungsfrage 1 erfüllt?

Der im Rahmen der vorangegangenen Forschungsfrage erstellte Ansatz wird zuletzt in Bezug auf seine Eignung für die Entwicklung von PSS evaluiert. Durch die Evaluation soll überprüft werden, inwieweit die in Forschungsfrage 1 definierten Anforderungen an den Requirements Engineering Ansatz für PSS erfüllt sind. Darüber hinaus steht im Fokus der Evaluation das Identifizieren erfolgskritischer Aspekte des Requirements Engineering Ansatzes für PSS, die den Einsatz des Ansatzes beeinflussen. Forschungsfrage 3 lautet:

3) Welchen Mehrwert und Nutzen bringt der Requirements Engineering Ansatz für die Entwicklung von PSS?

Im folgenden Abschnitt werden die methodischen Schritte zur Beantwortung der Forschungsfragen beschrieben.

1.3 Ablauf und Aufbau der Arbeit

Die Arbeit ist in 9 Kapiteln strukturiert. In den Kapiteln 1 bis 0 werden die Grundlagen für das Forschungsvorhaben erläutert sowie Forschungsfrage 1 beantwortet. In den Kapiteln 5, 6, 7 und 8 werden die Entwicklung und die Evaluation des Forschungsgegenstandes dargestellt und somit die Forschungsfragen 2 und 3 beantwortet. Kapitel 9 schließt mit einer kritischen Würdigung die Arbeit ab. Abbildung 1-1 stellt den Ablauf der Arbeit dar.

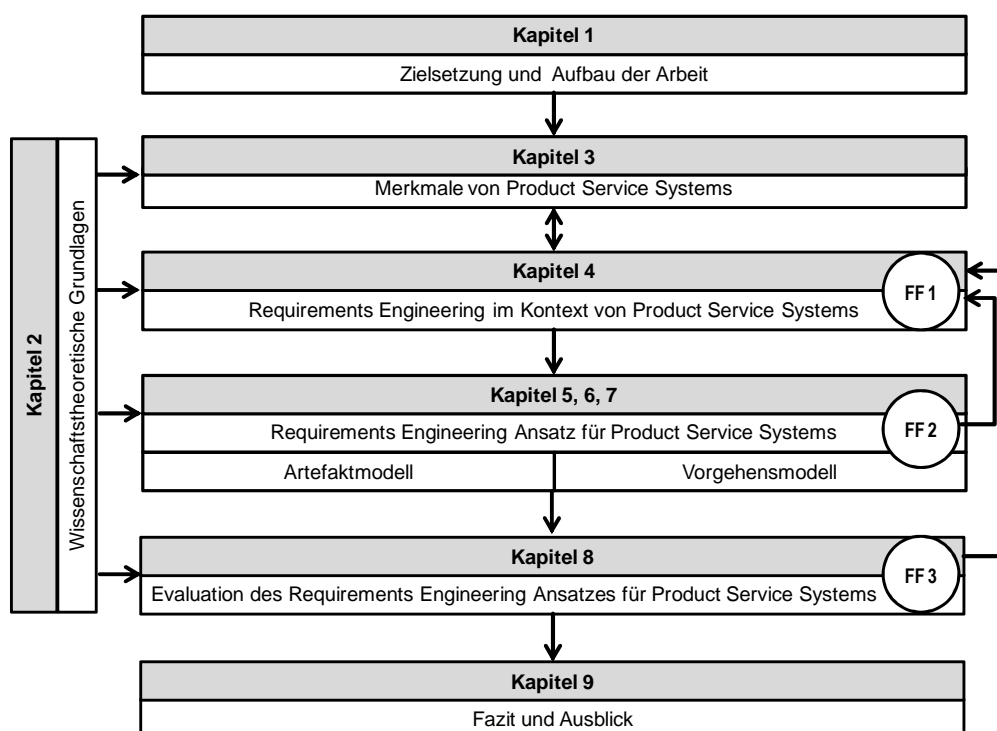


Abbildung 1-1: Ablauf der Arbeit (FF: Forschungsfrage)

Quelle: Eigene Darstellung

Der erste Schritt im Ablauf der vorliegenden Arbeit ist die Bereitstellung der wissenschaftstheoretischen Grundlagen. Dafür werden in Kapitel 0 die Einordnung der Arbeit in den Bereich der Wirtschaftsinformatik beschrieben, sowie die durch die Arbeit zu verfolgenden Forschungsziele vorgestellt. Das im Rahmen der Arbeit zu verwendende Forschungsdesign und die darin einzusetzenden Forschungsmethoden werden detailliert erläutert. Kapitel 0 beeinflusst die im Rahmen dieser Arbeit durchgeführte Forschung und ist deshalb grundlegend für alle nachfolgenden Kapitel.

In Kapitel 3 wird der Begriff „Product Service System“ (PSS) eingeführt und bezüglich seiner semantischen Bedeutung und seinen Eigenschaften analysiert. Ein besonderes Augenmerk liegt dabei auf der Analyse von zyklischen Wechselwirkungen im Lebenszyklus von PSS und deren Auswirkungen auf das Requirements Engineering. Ebenso wird eine Abgrenzung des Begriffs „Product Service System“ von den Begriffen „Sachleistung“ und „Dienstleistung“

vorgenommen. Anschließend werden die Begriffe „Anforderung“ und „Requirements Engineering“ erläutert.

PSS sind innovative kundenindividuelle Lösungen und werden von den Domänen Produkt-, Software- und Dienstleistungsentwicklung integriert entwickelt. In Kapitel 0 wird der Frage nachgegangen, wie die einzelnen Domänen mit dem Requirements Engineering sowie mit den zyklischen Wechselwirkungen im Requirements Engineering umgehen und in wie weit die Requirements Engineering Ansätze der beteiligten Domänen für PSS geeignet sind. Ausgehend von den dadurch gewonnen Erkenntnissen und den theoretischen Grundlagen zu PSS aus Kapitel 3 werden empirische Studien zum Stand des Requirements Engineering für PSS in der Praxis durchgeführt. Auch im Rahmen der empirischen Studien wird der Aspekt geklärt, welche Rolle zyklische Wechselwirkungen im Requirements Engineering spielen und wie mit diesen in der Praxis umgegangen wird. Aufbauend auf den gewonnenen Erkenntnissen werden Anforderungen an einen Requirements Engineering Ansatz für PSS abgeleitet.

Die in Kapitel 0 abgeleiteten Anforderungen an einen Requirements Engineering Ansatz werden in Kapiteln 5, 6 und 7 zur Konzeption der Bestandteile des Requirements Engineering Ansatzes herangezogen. Während in Kapitel 5 der Bezugsrahmen für den Requirements Engineering Ansatz für Product Service Systems erläutert wird, der die einzelnen Bestandteile des Ansatzes und deren Verbindungen aufzeigt, werden diese Bestandteile in den Kapiteln 6 und 7 ausführlich beschrieben.

Kapitel 8 hat das Ziel, die Evaluation des in Kapiteln 5, 6 und 7 erarbeiteten Requirements Engineering Ansatzes anhand der im Rahmen der methodischen Grundlagen beschriebenen Kriterien und Vorgehen zur Artefaktevaluation durchzuführen. Im ersten Teil der Evaluation wird die Erfüllung der an den Ansatz gestellten Anforderungen (Kapitel 0) analytisch gezeigt. Zudem wird dort eine beispielhafte Anwendung des Ansatzes vorgestellt, um dessen Machbarkeit zu zeigen. Außerdem wird eine merkmalsbasierte Evaluation auf Basis von Kriterien für dokumentierte Anforderungen durchgeführt. Im zweiten Teil der Evaluation wird eine retrospektive Anwendung des Ansatzes in zwei Fallstudien durchgeführt. Die beiden Fallstudien ergänzen sich, indem sich eine Fallstudie auf ein PSS aus dem IT-Bereich konzentriert, während die zweite Fallstudie ein PSS aus dem Bereich der technischen Produkte behandelt. Zur Bewertung werden dabei die Kriterien aus der merkmalsbasierten Evaluation, sowie die Herausforderungen in der Praxis (Kapitel 4.5) herangezogen.

Kapitel 9 fasst die Ergebnisse und Beiträge zusammen und stellt den Abschluss der Arbeit dar. Zunächst wird die Problemstellung zusammengefasst. Anschließend werden die Beiträge der Ergebnisse zur wissenschaftlichen Diskussion verdeutlicht. Schließlich werden die Implikationen für die Praxis sowie der weitere Forschungsbedarf diskutiert. Abbildung 1-2 verdeutlicht den Aufbau der Arbeit basierend auf den zentralen Aspekten der jeweiligen Kapitel.

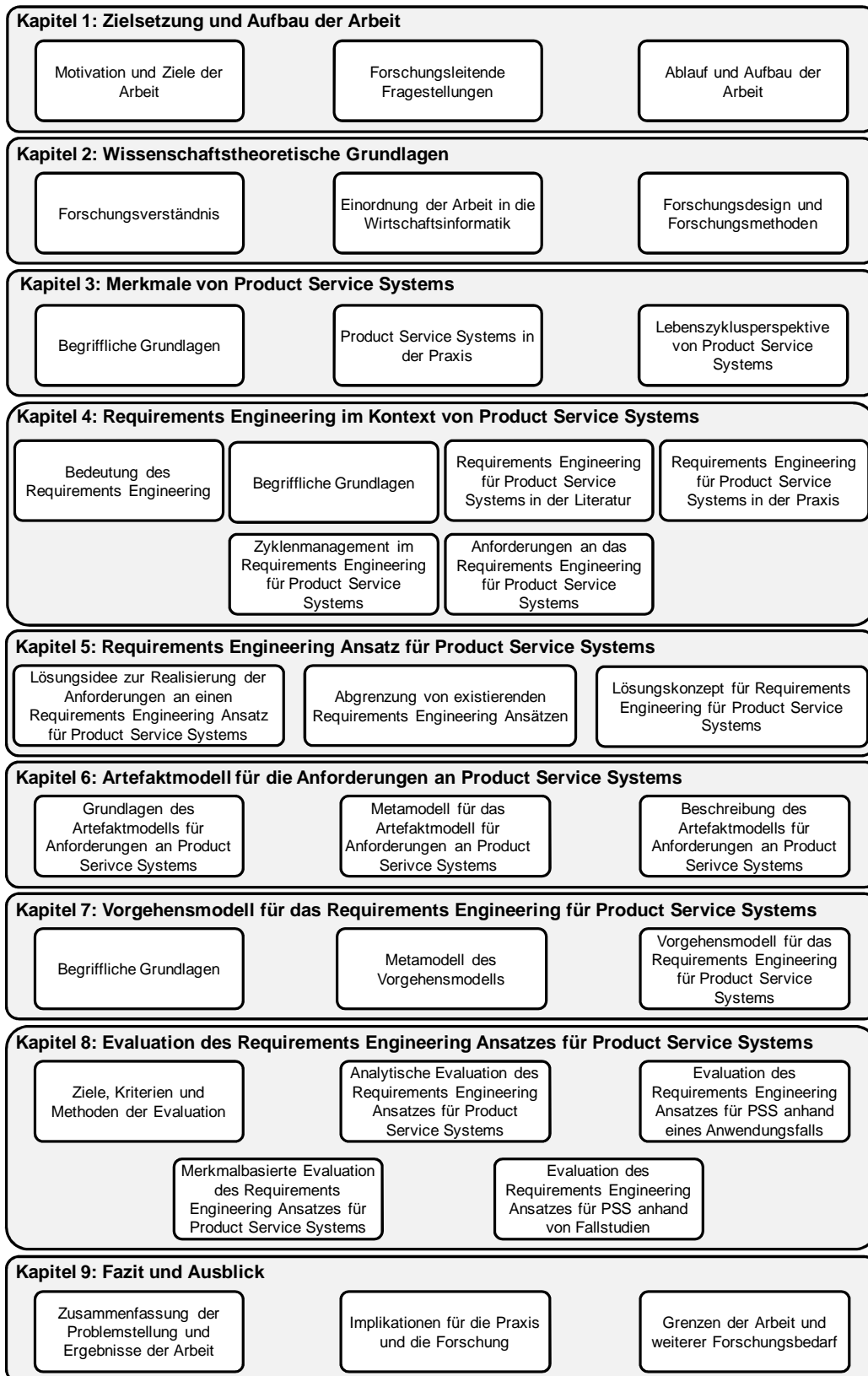


Abbildung 1-2: Aufbau der Arbeit

Quelle: Eigene Darstellung

2 Wissenschaftstheoretische Grundlagen

Die Beschreibung der wissenschaftstheoretischen Grundlagen wurde in zwei Unterkapitel aufgeteilt. In Kapitel 2.1 wird die Einordnung der Arbeit in die Wirtschaftsinformatik beschrieben. Im Kapitel 2.2 werden das Forschungsdesign und die verwendeten Forschungsmethoden vorgestellt.

Die Ergebnisse einer wissenschaftlichen Forschungsarbeit und die daraus abgeleiteten Erkenntnisse sollen intersubjektiv nachvollziehbar sein. Um das zu erreichen, werden die Forscher in ihrem Handeln durch Forschungsmethoden und Forschungsdesigns geleitet. In den jeweiligen Disziplinen haben sich Forschungsmethoden und -designs herausgebildet, die der jeweiligen wissenschaftstheoretischen Grundposition folgen. Während bspw. die Fragestellung der Geisteswissenschaften das Verstehen eines Gegenstandes ist, beziehen sich die Naturwissenschaften auf das Erklären des Gegenstandes (Tschamler 1996, 22). Erklären bedeutet im Unterschied zum Verstehen, dass auch die Ursache-Wirkungsbeziehungen erläutert werden.

Die Wissenschaftstheorie untersucht „die Grundsätze der Gewinnung, Überprüfung und Systematisierung wissenschaftlicher Erkenntnisse“ (Büttemeyer 1995, 15). Sie definiert den Rahmen, innerhalb dessen die Forschung erfolgt (Chmielewicz 1979; Bortz/Döring 2006). In ihrem Fokus stehen einerseits die Frage nach Merkmalen und Voraussetzungen der Wissenschaft und andererseits nach deren Zielen (Büttemeyer 1995, 15). Diese Aspekte beeinflussen die wissenschaftstheoretische Grundposition des Forschers und somit die Auswahl der Forschungsmethoden und die Ergebnisse der Arbeit.

2.1 Einordnung der Arbeit in die Wirtschaftsinformatik

Thematisch ist die Arbeit dem Bereich der Wirtschaftsinformatik zuzuordnen, die eine eigenständige Disziplin an der Schnittstelle zwischen ihren Mutterwissenschaften, Informatik und Betriebswissenschaft, darstellt (Heinzl et al. 2001; Winter 2009). Zu den Zielen der Wirtschaftsinformatik gehören die Gewinnung von Erkenntnissen und somit das Anbieten von Lösungen zu praxisrelevanten Problemen (Hill/Ulrich 1979; Petkoff 1999). Dabei geht die Wirtschaftsinformatik auf die Analyse, Implementierung, Entwurf und die anschließende Nutzung von betrieblichen Informationssystemen ein (Frank 2008; Becker et al. 2003). Ebenso wird in der Wirtschaftsinformatik ein großer Wert auf die Methoden zur Gestaltung von solchen Systemen gelegt (Frank 2008). Somit ist das Ziel der Forschung in der Wirtschaftsinformatik, die erarbeiteten Forschungsergebnisse in der Praxis anzuwenden und die dadurch entstehenden Erkenntnisse weiter in der Forschung und der Praxis zu nutzen.

Der Kern dieser Arbeit stellt einen Ansatz zum Requirements Engineering für PSS dar. Die Gestaltung von PSS, der die Entscheidungen über die Entwicklung von Anpassungs- und Integrationsmöglichkeiten bei materiellen und immateriellen Komponenten zu Grunde liegen, kann im Forschungsgebiet der Wirtschaftsinformatik angesiedelt werden (Zellner 2008, 187). PSS sind individuelle Kundenlösungen, bei denen die Erfüllung von Kundenanforderungen im Vordergrund steht (Lönngren et al. 2008; Baines et al. 2007). In diesem Zusammenhang spielt das Requirements Engineering, das sich mit der Ermittlung und Verwaltung von Anforderungen an ein System beschäftigt, eine grundlegende Rolle für den Entwicklungsprozess von PSS. Requirements Engineering ist überwiegend der Analysephase in der Entwicklung von Systemen zuzuordnen, aber auch aufgrund von Änderungen von Anforderungen sowie späteren Änderungen in der Nutzungsphase wird Requirements Engineering auch in den Pha-

sen der Implementierung, des Entwurfs und der Nutzung durchgeführt. Somit unterstützt Requirements Engineering die Planung und Realisierung von Anwendungssystemen, die zu den Inhalten der Wirtschaftsinformatik gehören (Petkoff 1999). Als Fazit lässt sich feststellen, dass sowohl die Thematik der PSS als auch das Thema des Requirements Engineering in die Wirtschaftsinformatik einzuordnen sind.

Während Informatik sich mit grundlegenden Themen, wie die Verarbeitung von Informationen bis hin zur der Konzeption und Entwicklung von komplexen Anwendungssystemen, beschäftigt, ist die Wirtschaftsinformatik der angewandten Informatik zuzurechnen, da sie sich mit der Entwicklung und Einführung von Informationssystemen in Unternehmen beschäftigt (Jarke 2009). Auf der anderen Seite teilt die Wirtschaftsinformatik ihren Gegenstandsbereich, der sich auf die Unternehmung sowie Unternehmensbereiche und Unternehmensverbände fokussiert, mit der Betriebswirtschaftslehre (Winter 2009). Da insbesondere der Dienstleistungsanteil von PSS es erfordert, dass ein PSS während seines gesamten Lebenszyklus stark an ein Unternehmen gekoppelt wird, ist für die Erbringung der Lösung, die ein integriertes Bündel aus Sach- und Dienstleistungen darstellt, eine Einbindung in die Informationssysteme des Unternehmens erforderlich, wodurch die Verbindung dieser Arbeit mit der Wirtschaftsinformatik zusätzlich begründet wird.

2.2 Forschungsmethoden und Forschungsdesign

Als methodisches Rahmenkonzept wird für diese Arbeit gestaltungsorientierte Forschung gewählt, die das Ziel hat vorhandene Probleme „durch das (proaktive) Schaffen und Evaluieren verschiedener Artefakte in Form von Modellen, Methoden oder Systemen“ (Wilde/Hess 2007) zu lösen. Die wichtigsten Merkmale und Implikationen der gestaltungsorientierten Forschung werden in Kapitel 2.2.1 beschreiben. Anschließend wird in Kapitel 2.2.2 das verwendete Forschungsdesign erläutert. Ein weiteres Konzept, das zum Erstellen des methodischen Konzepts herangezogen wurde, ist das „technical research“, das als „the investigation of the properties of artifacts“ definiert wird (Wieringa/Heerkens 2008). „Technical researchers may interleave designing (and building) artifacts with investigating them, but they may also investigate the properties of artifacts designed or built by others“ (Wieringa/Heerkens 2008). Die Forschung im Bereich des Requirements Engineering wird als ein Teil von „technical research“ betrachtet.

2.2.1 Methodisches Rahmenkonzept

Im Rahmen dieser Arbeit sollen Lösungen für Klassen von Problemen angeboten werden, die durch die forschungsleitenden Fragestellungen und die damit verbundenen Ziele (Kapitel 1.2) aufgestellt werden. Zur Gestaltung des methodischen Rahmenkonzepts werden Ansätze benötigt, die zur Untersuchung der Realität geeignet sind und somit die Realität durch bestimmte Artefakte beeinflussen (Järvinen 2000). Die Gestaltung des methodischen Rahmenkonzepts orientiert sich an den Taxonomien von Forschungsmethoden, vorgeschlagen von Järvinen (2000, 252). Der Pfad in Abbildung 2-1 zeigt die für die Arbeit gewählten Ansätze.

Demzufolge sollen „nützliche Artefakte“ entwickelt werden, die zur Lösung von Problemen mit IS-Bezug geeignet sind (Hevner et al. 2004, 87; Gericke/Winter 2009, 196; Frank 2000). Der zu entwickelnde Ansatz des Requirements Engineering für IT-gestützte PSS stellt ein solches Artefakt dar.

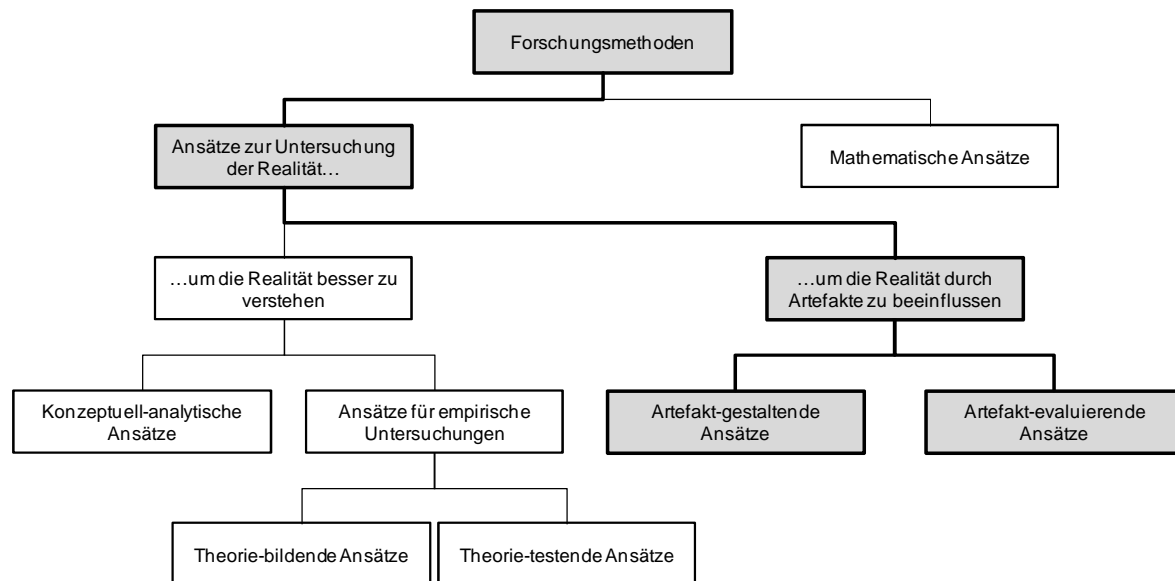


Abbildung 2-1: Taxonomie für Forschungsmethoden

Quelle: (Järvinen 2000) zitiert in (Hoffmann 2010)

Die Konzeption und Evaluation von IT-Artefakten zur Lösung von Problemen stehen im Fokus der gestaltungsorientierten Forschung, engl. Design Science (Baskerville 2008; Hevner et al. 2004; Kuechler/Vaishnavi 2008). Hevner et al. (2004, 77) definieren die gestaltungsorientierte Forschung als „the other side of the IS research cycle, [that] creates and evaluates IT artifacts intended to solve identified organizational problems“. Simon (1996) spricht über „sciences of the artificial“ als eine Forschung, in der Artefakte gestaltet werden, um Ziele durch das Bereitstellen bestimmter Funktionen zu erreichen. Dabei können Artefakte in Form von Modellen, Methoden, Konstrukten und Instanzen vorliegen und variieren von formalen Beschreibungen bis Software und informalen sprachlichen Beschreibungen (Hevner et al. 2004, 77; March/Smith 1995). Das Anwenden der gestaltungsorientierten Forschung soll nicht nur die Entwicklung eines Artefaktes enthalten, sondern auch seine Evaluation (March/Smith 1995). Nur dann kann der Grad der Zielerreichung durch die Artefakte definiert und somit die Nützlichkeit des Artefaktes bewertet werden (Simon 1996). Somit stellen Hevner et al. (2004, 81) fest, dass gestaltungsorientierte Forschung bei Problemen angewendet werden kann, wenn:

- sich die Anforderungen und Rahmenbedingungen aufgrund eines wenig erforschten Anwendungsfeldes häufig ändern,
- einzelne Problem- und Lösungskomponenten komplexe Zusammenhänge aufweisen,
- inhärente Flexibilität zum Ändern von Gestaltungsprozessen und Artefakten führt,
- kritische Abhängigkeiten von kognitiven Fähigkeiten der beteiligten Personen zur effektiven Lösungsfindung vorhanden sind,
- kritische Abhängigkeiten von sozialen Fähigkeiten der beteiligten Personen zur effektiven Lösungsfindung vorhanden sind.

Zur Gestaltung des Forschungsprozesses der gestaltungsorientierten Forschung schlagen Hevner et al (2004) die Richtlinien aus Tabelle 2-1 vor.

Richtlinie	Beschreibung
1. Konstruktion als Artefakt	Die gestaltungsorientierte Forschung verlangt die Entwicklung eines anwendbaren Artefakts in Form einer Methode, eines Modells oder einer Instanz.
2. Problemrelevanz	Das Ziel der gestaltungsorientierten Forschung ist die Entwicklung von technologiebasierten Lösungen für wichtige und relevante Probleme.
3. Evaluation des Artefakts	Die Anwendung, Qualität und Effektivität des konzipierten Artefakts müssen durch entsprechende Evaluationsmethoden nachvollziehbar dargestellt werden.
4. Forschungsbeitrag	Die gestaltungsorientierte Forschung muss klar nachvollziehbare und verifizierbare Ergebnisse liefern.
5. Stringenz	Die gestaltungsorientierte Forschung geht davon aus, dass bei der Konstruktion und Evaluation eines Artefakts forschungsmethodische Grundsätze bewahrt werden.
6. Suchprozess	Die gestaltungsorientierte Forschung ist ein Suchprozess, in dem nach einem effektiven Artefakt gesucht wird. Um dabei die gewünschten Ziele zu erfüllen, sollen die Gesetze, die durch die Umgebung vorgegeben werden, erfüllt werden.
7. Kommunikation der Ergebnisse	Die Ergebnisse der Anwendung der gestaltungsorientierten Forschung müssen sowohl einem technologieorientierten als auch einem management-orientierten Publikum effektiv vermittelt werden.

Tabelle 2-1: Richtlinien für die Erfüllung des Design-Science-Paradigma

Quelle: (Hevner et al. 2004, 83)

Diese Richtlinien lassen sich zur Durchführung der gestaltungsorientierten Forschung und zur Evaluation der Ergebnisse davon einsetzen (Hevner et al. 2004, 77). Takeda et al. (1990, 12ff.) (zitiert in Vaishnavi und Kuechler (2004)) schlagen ein Vorgehensmodell vor, das die einzelnen Schritte bei der Durchführung der gestaltungsorientierten Forschung angibt (Abbildung 2-2).

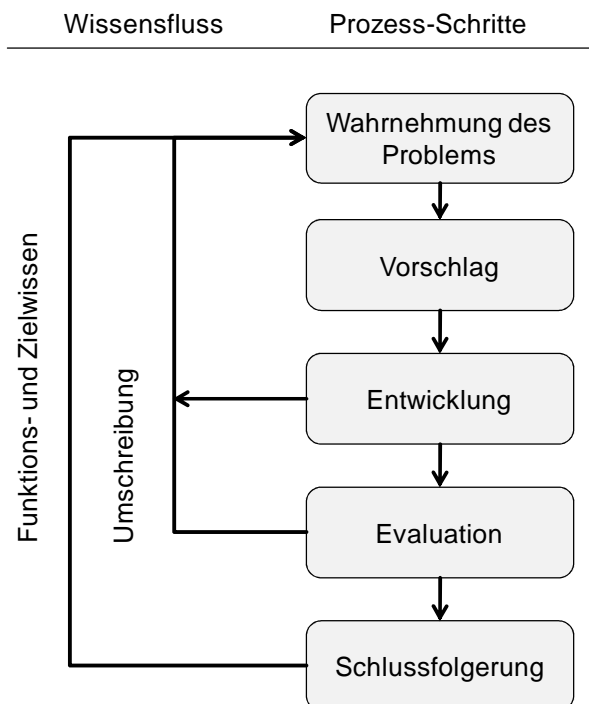


Abbildung 2-2: Vorgehensmodell der gestaltungsorientierten Forschung

Quelle: (Takeda et al. 1990, 12)

Das Vorgehensmodell besteht aus folgenden fünf Phasen (Takeda et al. 1990, 12ff.):

1. Wahrnehmung des Problems: Die Forschung beginnt mit dem Verstehen des Problems, das im Rahmen der Forschung zu lösen ist. Das Ergebnis dieser Phase ist eine formale oder informale Beschreibung des Problems und die Begründung eines Forschungsbedarfs.
2. Vorschlag: Anschließend wird ein vorläufiges Design zur Lösung des Problems erstellt, dem die davor erarbeitete Wissensbasis zu Grunde liegt.
3. Entwicklung: Im Rahmen der Phase findet die Realisierung des Lösungskonzeptes aus Phase 2 statt und ergibt ein fertiges Artefakt.
4. Evaluation: Während der Evaluation wird das entwickelte Artefakt aus Phase 3 in Bezug auf Güte- und Leistungsmaße bewertet. Das Ziel ist dabei zu klären, wie gut das Artefakt zur Lösung des vorhandenen Problems geeignet ist.
5. Schlussfolgerung: In der letzten Phase werden die gewonnenen Ergebnisse und Erkenntnisse gesammelt, kritisch bewertet und dokumentiert.

Somit lässt sich eine grundlegende Struktur zur Gliederung des gestaltungsorientierten Forschungsansatzes erstellen. Der erste Schritt besteht dabei darin, das Problem zu identifizieren und die damit im Zusammenhang stehenden Rahmenbedingungen zu erkennen, um darauf basierend einen Lösungsvorschlag zu konzipieren. Der Lösungsvorschlag stellt die Basis für die Entwicklung eines Artefakts, das den Lösungsvorschlag umsetzt. Dadurch wird erreicht, dass die Ziele, die mit dem Erstellen des Artefakts verfolgt werden, zur Lösung des im ersten Schritt identifizierten Problems passen. Anschließend wird das Artefakt evaluiert, indem es in Bezug auf die Erreichung der Ziele überprüft wird. Dieser Schritt hat einen Zusammenhang mit dem ersten Schritt des Vorgehens, um iterativ eine bessere Gestaltung und Umsetzung des Artefaktes zu ermöglichen. Der letzte Schritt stellt die Aufbereitung und Dokumentation der gewonnenen Ergebnisse und der damit im Zusammenhang stehenden Erkenntnisse dar.

Das Vorgehensmodell der gestaltungsorientierten Forschung sowie Richtlinien zur Durchführung der gestaltungsorientierten Forschung stellen die Basis für die vorliegende Arbeit dar.

2.2.2 Verwendetes Forschungsdesign

Das Ziel dieser Arbeit ist die Entwicklung eines Ansatzes zum Requirements Engineering für IT-gestützte Product Service Systems. Somit sollen die Kunden- und Stakeholderanforderungen, die an die gesamte Lösung und nicht an die einzelnen Komponenten gestellt werden, ermittelt und analysiert werden, indem sie strukturiert, konkretisiert und in geeigneter Form an die einzelnen an der Entwicklung von PSS beteiligten Domänen bereitgestellt werden. Diese analysierten Anforderungen werden als *Lösungsanforderungen* (Ebert 2005, 41) bezeichnet. Es soll ebenso sichergestellt werden, dass die initialen Anforderungen an das PSS den analysierten Lösungsanforderungen entsprechen und von den Entwicklern richtig verstanden werden. Der entwickelte Ansatz soll die Brücke zwischen den Anforderungen der Kunden und weiterer Stakeholder an die gesamte Lösung und den analysierten Anforderungen in „der Sprache des Entwicklers“ erstellen und somit eine möglichst effektive Unterstützung für die Validierung von Anforderungen ermöglichen. Damit wird die Voraussetzung für eine maximale Erfüllung von Kunden- und Stakeholderanforderungen durch die Lösung geschaffen. Zudem sollen die Zyklen, die in Form von Iterationen bei der Abstimmung von Anforderungen zwischen den Entwicklern und der Requirements Engineering Analysten entstehen, reduziert werden, was zu einer Reduktion der Entwicklungszeit für die Lösung führt.

Wie in Kapitel 2.2.1 erwähnt, lässt sich die gestaltungsorientierte Forschung gut auf Probleme anwenden, bei denen es um komplexe Verflechtungen zwischen den einzelnen Problem- und Lösungsbestandteilen geht und wenn das Anwendungsfeld nur wenig erforscht ist und zu ständigen Änderungen von Anforderungen und Rahmenbedingungen führt.

Die Beteiligung von mehreren Domänen mit unterschiedlichen Vorgehensweisen und unterschiedlichem Verständnis über Anforderungen und Requirements Engineering führen zu komplexen Verflechtungen im Entwicklungsprozess von PSS. Desweiteren führen die meist sehr ungenauen und oft nur implizit vorhandenen Anforderungen der Kunden an das PSS und die daraus durch die Anforderungsanalyse resultierenden konkreten Anforderungen an die Komponenten der Lösung zu komplexen Zusammenhängen in der Umsetzung der Anforderungen durch die einzelnen Domänen. Zudem muss die entwickelte Lösung in die Geschäftslandschaft des Kunden integriert werden, wodurch zusätzliche Komplexität im Requirements Engineering und Entwicklungsprozess ausgelöst wird. Aufgrund dieser Charakteristika des Entwicklungsprozesses von PSS und des Requirements Engineerings sind die Voraussetzungen zum Anwenden der gestaltungsorientierten Forschung gegeben.

Gemäß der gestaltungsorientierten Forschung stellt der Ansatz zum Requirements Engineering für IT-gestützte Product Service Systems das eigentliche Artefakt dar, das im Sinne des Vorgehens nach Hevner et al. (2004) zu evaluieren ist. Das Vorgehen im Rahmen dieser Arbeit ist angelehnt an die Richtlinien von Hevner et al. (2004) sowie das Vorgehensmodell von Takeda et al. (1990). Die in Kapitel 1.2 dargelegten Forschungsfragen sollen die Forschungsaktivitäten leiten. Tabelle 2-2 fasst die im Rahmen der Arbeit verwendeten Methoden zusammen und stellt sie den Phasen des Vorgehensmodells der gestaltungsorientierten Forschung nach Takeda et al. (1990) sowie den der Arbeit zu Grunde liegenden Forschungsfragen gegenüber.

Prozessschritt	Forschungsfrage	Methoden
Wahrnehmung des Problems	Forschungsfrage 1	<ul style="list-style-type: none"> • Literaturanalyse zum Stand der Wissenschaft • Qualitative Interviews mit Experten • Fallstudie
Vorschlag		
Entwicklung	Forschungsfrage 2	<ul style="list-style-type: none"> • Entwicklung eines Bezugsrahmens für das Requirements Engineering für PSS • Entwicklung eines Requirements Engineering Ansatzes für PSS
Evaluation	Forschungsfrage 3	<ul style="list-style-type: none"> • Qualitative Befragung • Qualitative Interviews mit Experten • Dokumentenanalyse (z.B. Richtlinien zum Requirements Engineering, Lasten- und Pflichtenhefte) • Anwendung des entwickelten Requirements Engineering Modells auf reale Fälle
Schlussfolgerung		

Tabelle 2-2: *Verwendete Forschungsmethoden nach den Phasen des Vorgehensmodells nach Takeda et al. (1990) und Forschungsfragen*
Quelle: Eigene Darstellung

Die erste Forschungsfrage soll den Stand der Wissenschaft und Praxis bezüglich des Requirements Engineering für PSS erläutern. Requirements Engineering ist ein kommunikationsintensiver Prozess (Hsia et al. 1993) und wird stark von den sozialen und organisationalen Kontexten des Unternehmens, das es einsetzt, beeinflusst. Obwohl es mehrere Ansätze zum Requirements Engineering gibt, steht im Vordergrund dieser Ansätze die Entwicklung von Software, Hardware oder eingebetteter Systeme. Durch die neuartige Zusammensetzung von diesen Komponenten und Dienstleistungen und der Ausrichtung des so entstandenen Produktes auf die Lösung des Kundenproblems stellt Requirements Engineering für PSS einen Untersuchungsbereich dar, zu dem wenige Erkenntnisse bestehen. Vielmehr muss auf die einzelnen an der Entwicklung von PSS beteiligten Domänen eingegangen werden und das Zusammenwirken der domänenspezifischen Methodiken untersucht werden, um daraus ein Verständnis für einen domänenübergreifenden und integrierten Ansatz für PSS zu entwickeln. Vor diesem Hintergrund wurde ein interpretativer Ansatz für die Beantwortung der Forschungsfrage 1 gewählt. Interpretative Ansätze werden eingesetzt, wenn es um Untersuchungen von unbekanntem und neuen Gegenständen sowie der Nachvollziehung des subjektiv gemeinten Sinns geht (Rosenthal 2005). Besonders in der Informationssystemforschung haben diese Ansätze Verbreitung gefunden, um die Informationssysteme und deren Kontext, sowie den Umgang der Menschen mit Informationssystemen zu verstehen (Doolin/McLeod 2005; Walsham 1993; Klein/Myers 1999). Wie bereits in Kapitel 2.2.1 und in diesem Kapitel erläutert wurde, wird für die Forschungsfragen 2 und 3 ein gestaltungsorientierter Ansatz gewählt. Dabei werden verschiedene Forschungsmethoden kombiniert. Aufgrund einer tiefgehenden Diskussion der Methoden in der Literatur wird an dieser Stelle auf eine detaillierte Beschreibung der Methoden verzichtet und auf einige ausgewählte Quellen verwiesen (z. B. Bortz und Döring (2006), Lamnek (2005), Atteslander (2008)).

2.2.3 Fazit und Implikationen für den weiteren Gang der Arbeit

In diesem Kapitel wurden das Forschungsverständnis und das Forschungsdesign der Arbeit erörtert, das den Rahmen für die Beantwortung der forschungsleitenden Fragestellungen bildet. Das Ziel dieser Arbeit ist das Verstehen der realweltlichen Problemsituation des Requirements Engineering für Product Service Systems, um darauf basierend das Gestalten einer Lösung vorzunehmen. Somit eignet sich die gestaltungorientierte Forschung besonders gut für diese Arbeit. Als Orientierung dafür dienen die Richtlinien von Hevner et al. (2004) und das Vorgehensmodell von Takeda et al. (1990).

Der erste Schritt zur Entwicklung eines Ansatzes für das Requirements Engineering für PSS ist das Verstehen des vorliegenden Problems, das Ermitteln des Standes der Forschung sowie des Standes der Praxis, um darauf basierend Anforderungen an einen Requirements Engineering Ansatz zu erheben. Diese Anforderungen liefern die Grundlage zur Konzeption eines Bezugsrahmens, der das Verständnis und die Bewertung von wesentlichen Begriffen und Konzepten eines Bereichs, in diesem Fall des Requirements Engineering für PSS, ermöglicht. Somit beschreibt der Bezugsrahmen die wesentlichen Teile des Requirements Engineering Ansatzes für PSS. Anschließend werden die im Bezugsrahmen definierten Teile des Requirements Engineering Ansatzes konzipiert. Die Evaluation des Requirements Engineering Ansatzes für PSS soll zum einen die Machbarkeit zeigen, indem er den im Rahmen der ersten Forschungsfrage definierten Anforderungen gegenübergestellt wird. Zum anderen sollen durch die Anwendung des Ansatzes auf einen realen Fall des PSS und die anschließende Bewertung durch die Experten die Vor- und Nachteile des entwickelten Ansatzes gezeigt werden. Die Diskussion der Ergebnisse, die Implikationen für die Praxis und die Forschung sowie das Aufzeigen der Grenzen der Arbeit und des weiteren Forschungsbedarfs schließen die Arbeit ab.

3 Merkmale von Product Service Systems

In Kapitel 3 wird der Begriff „Product Service Systems“ eingeführt und anhand mehrerer Fallbeispiele erklärt. Anschließend wird auf den Lebenszyklus und den Entwicklungsprozess von PSS eingegangen, die besonders für das Verstehen der Rolle des Requirements Engineering für PSS wichtig sind. Die Beschreibung der während des Lebenszyklus von PSS auftretenden Zyklen, die zur Wiederholung von Aktivitäten oder Ergebnissen dieser Aktivitäten im Requirements Engineering führen, schließt das Kapitel ab.

3.1 Begriffliche Grundlagen

Die Differenzierung in Sach- oder Dienstleistungen alleine reicht den Unternehmen nicht mehr aus, um den Markterfolg langfristig zu sichern (Böhmman/Krcmar 2007; Knackstedt et al. 2008). Darüber hinaus wollen immer mehr Kunden keine Produkte oder Dienstleistungen per se, sondern erwarten individuelle Lösungen für ihre Probleme (Leimeister/Glauner 2008; Davies 2004; Berkovich et al. 2009c). Ca. 90% der von Sturm et al. (2007) befragten mittelständischen Unternehmen sind sich einig, dass Kunden immer mehr nach kundenindividuellen Leistungen fragen. Daher werden viele Unternehmen zu sogenannten Lösungsanbietern (Galbraith 2002). Die von den Unternehmen angebotenen Lösungen sind unter dem Begriff *hybride Produkte* oder *Product Service Systems (PSS)* bekannt und stellen integrierte Bündel von Sach- und Dienstleistungen dar (An et al. 2008; Böhmman et al. 2008; Morelli 2002). Um eine genaue Definition des Begriffs „Product Service System“ zu geben, ist eine Abgrenzung des Begriffs von den Begriffen „Produkt“, „Sachleistung“ und „Dienstleistung“ erforderlich.

3.1.1 Überblick zu Definitionen von Product Service Systems

3.1.1.1 Abgrenzung des Begriffs „Produkt“

Im Zentrum der Produktentwicklung, im Sinne einer klassischen ingenieurwissenschaftlichen Disziplin, steht das Lösen von technischen Problemen (Pahl et al. 2006, 1) und somit die Konzeption und Entwicklung von technischen Systemen. Ein *System* ist eine Gesamtheit von Elementen, die durch ihre Wechselwirkungen als eine Einheit wirken (Stachowiak 1973). Dabei werden *technische Systeme* definiert als „künstlich erzeugte geometrisch-stoffliche Gebilde, die einen bestimmten Zweck (Funktion) erfüllen, also Operationen (physikalische, chemische, biologische Prozesse) bewirken“ (Ehrlenspiel 2002, 22). Von einem *technischen Produkt* spricht man, wenn es um das geometrisch-stoffliche Gebilde des technischen Systems geht, und nicht um den Prozess, den das Gebilde durchführt (Ehrlenspiel 2002, 27). Zur Charakterisierung von technischen Produkten unterscheidet Ehrlenspiel (2002, 24f.) drei Merkmale:

- Beschaffenheitsmerkmale können am Produkt selbst festgestellt werden und geben dadurch an, welche geometrischen oder stofflichen Eigenschaften ein Produkt aufweist. Dazu gehören bspw. Gestalt, Werkstoff oder Farbe.
- Funktionsmerkmale geben „den gewollten Zweck eines Produktes“ an. Dazu gehören zum Beispiel das zu übertragende Drehmoment oder die zu messende Temperatur.
- Relationsmerkmale sind erst im Zusammenhang mit anderen Systemen zu beachten. Beispiele für die Relationsmerkmale sind Herstellkosten, Bedienbarkeit oder Geräusche.

Aus den oben genannten Definitionen lässt sich insbesondere der materielle Charakter von technischen Produkten feststellen. Aber auch in der ingenieurwissenschaftlichen Literatur existieren unterschiedliche Auffassungen über den Begriff „Produkt“. So bezeichnen Ulrich und Eppinger (2003, 2) ein *Produkt* als ein Leistungsbündel, das dem Kunden von einem Unternehmen angeboten wird. Obwohl diese Definition auch Dienstleistungen als ein Produkt vorsieht, konzentrieren sich die Autoren ausschließlich auf die Betrachtung von technischen Produkten.

Weitere Definitionen können in der betriebswirtschaftlichen Literatur gefunden werden. So wird ein Produkt definiert als „eine Leistung oder eine Gruppe von Leistungen, die von Stellen außerhalb des jeweils betrachteten Fachbereichs (innerhalb oder außerhalb der Organisation) benötigt werden“ (Scheer et al. 2003, 21). Somit umfasst die Definition eines Produktes sowohl Sach- als auch Dienstleistungen (vgl. (Schneider et al. 2003; Hutzschenreuter 2009, 173; Fähnrich/Opitz 2003)). Dabei soll ein Produkt dem Kunden einen Grundnutzen bieten, der aus dem Gebrauch des Produktes entsteht (Thommen/Achleitner 2006, 162).

Als Fazit lässt sich feststellen, während in der Produktentwicklung der Begriff „*Produkt*“ im Sinne eines technischen Systems aufgefasst wird, wird in der Betriebswirtschaft unter dem Begriff eine Leistung verstanden, die einen Kundennutzen anbieten soll und dabei sowohl Sach- als auch Dienstleistungen umfasst.

Im Rahmen dieser Arbeit wird der Begriff „*Produkt*“ im Sinne einer Leistung verstanden. Um die Abgrenzung zu einem geometrisch-stofflichen Gebilde eines technischen Systems herzustellen, wird für so ein System der Begriff „*technisches Produkt*“ oder „*Sachleistung*“ verwendet.

3.1.1.2 Abgrenzung der Begriffe „*Sachleistung*“ und „*Dienstleistung*“

Eine Leistung kann entweder eine Sach- oder eine Dienstleistung sein (Scheer et al. 2003, 21-23; Corsten 2001, 20f.). Abbildung 3-1 zeigt die verschiedenen Leistungs- bzw. Produktarten.

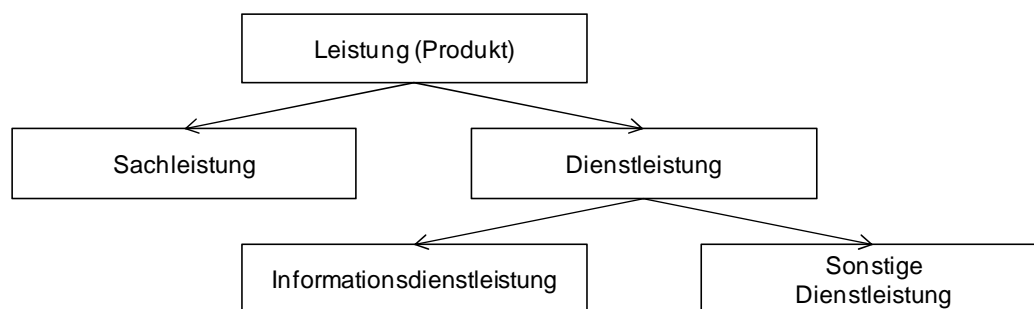


Abbildung 3-1: Leistungs- bzw. Produktarten

Quelle: In Anlehnung an (Scheer 2002, 23; Scheer et al. 2003, 21)

Sachleistungen werden definiert als „gegenständliche, künstlich hergestellte Objekte mit einer beschreibbaren technischen Funktion, die für bestimmte, jeweils spezifisch zu bestimmende Zwecke eingesetzt, d. h. durch spezifische Aktivitäten verwendet werden“ (Volz 1997, 32). Eine Sachleistung wird durch ihren materiellen Charakter gekennzeichnet und wird dem Kunden in Form eines materiell existierenden Gutes angeboten (Scheer et al. 2003, 22ff.; Haller 2005, 7-10), während eine Dienstleistung immaterieller Natur ist (Corsten 2001, 20; Klein 2007, 20). Die Dienstleistungen werden zusätzlich zur Immaterialität durch die Notwendigkeit der Integration externer Faktoren in die Leistungserstellung sowie die Produktion für den fremden Bedarf charakterisiert (Frietzsche/Maleri 2003, 197). Die Einbringung externer Fak-

toren bedeutet, dass der Kunde sich selbst in Form seiner Arbeitsleistung oder seine Verfügungsobjekte, die sowohl materiell als auch immateriell sein können, in die Dienstleistungsproduktion mit einbringt (Luczak et al. 2003, 448; Fähnrich/Opitz 2003, 94; Böhmman 2003, 10). Im Rahmen dieser Arbeit wird der Begriff „Sachleistung“ als Synonym zum Begriff „technisches Produkt“ verwendet.

Die Unterschiede zwischen Sach- und Dienstleistungen werden in Tabelle 3-1 in Anlehnung an Haller (2005, 10), Lehmann (1995, 21) und Boyta und Harvey (1997, 294) verdeutlicht.

Sachleistung	Dienstleistung
Sachleistung ist gegenständlich.	Dienstleistung ist immateriell.
Sachleistung kann vor dem Kauf vorgeführt werden.	Dienstleistung kann vor dem Kauf weder gezeigt noch geprüft werden.
Produktionsquantität und -qualität einer Sachleistung sind messbar.	Leistungsquantität und -qualität einer Dienstleistung sind schwer erfassbar.
Produktion der Sachleistung ist ohne Beteiligung des Käufers möglich.	Käufer ist an der Erstellung der Dienstleistung beteiligt.
Produktion und Übertragung der Leistung können auseinander fallen.	Produktion und Übertragung der Leistung fallen zeitlich und zum Teil räumlich zusammen.
Sachleistung ist lagerfähig und transportierbar.	Leistung ist nicht speicherbar und nicht transportierbar.
Sachleistung bekommt ihre Form im Rahmen des Produktentstehungsprozesses.	Leistung erhält ihre Form erst in der Servicesituation.
Eigentums- Besitzwechsel nach dem Kauf.	Kein Eigentumswechsel.

Tabelle 3-1: Unterschiede zwischen Sach- und Dienstleistungen

Quelle: In Anlehnung an (Haller 2005, 10; Lehmann 1995, 21; Boyta/Harvey 1997, 294)

Eine eindeutige Definition des Begriffs „*Dienstleistung*“ existiert nicht (vgl. (Haller 2005, 13; Bullinger/Schreiner 2003, 55)). Murdick et al. (1990, 4) zitiert in Cook et al. (1999, 320) definieren eine Dienstleistung als „economic activities that produce time, place, form, or psychological utilities“. Eine weitere Definition bringen Frietzsche und Maleri (2003, 197), die eine Dienstleistung als „unter Einsatz externer Produktionsfaktoren für den fremden Bedarf produzierte immaterielle Wirtschaftsgüter“ verstehen. Weiterhin können die Dienstleistungen anhand von vier Kategorien, wie enumerative, negative, institutionelle und konstitutive Kategorien (vgl. (Bullinger/Schreiner 2003, 55; Scheer et al. 2003, 23; Corsten 2001, 21; Burr/Stephan 2006, 18ff.)), definiert werden.

Bei der *enumerativen* Definition werden Beispiele von Dienstleistungen aufgezählt, um dadurch das Wesen von Dienstleistungen zu erfassen. Dadurch können sämtliche Dienstleistungsarten eines betrachteten Dienstleistungssektors erfasst werden. Allerdings kann die Vollständigkeit dieser Vorgehensweisen aufgrund der Heterogenität des Dienstleistungssektors und der kontinuierlichen Entwicklung innovativer Dienstleistungen nicht gewährleistet werden. Somit können dadurch einzelne Unternehmungen charakterisiert werden, aber eine präzise Definition von Dienstleistungen kann dadurch nicht erreicht werden (Bullinger/Schreiner 2003, 55; Scheer et al. 2003, 23; Corsten 2001, 21; Burr/Stephan 2006, 18ff.).

Bei der *Negativdefinition* wird alles als Dienstleistung bezeichnet, was nicht zu den Sachleistungen gehört (Bullinger/Schreiner 2003, 55; Scheer et al. 2003, 23; Corsten 2001, 21; Burr/Stephan 2006, 18ff.). Somit können Dienstleistungen als „Tätigkeiten, die sich nicht auf die unmittelbare Gewinnung, Verarbeitung oder Bearbeitung von Sachgütern richten“ (Scheer et al. 2003, 23) definiert werden. Mit diesem Definitionsansatz werden die Merkmale und Charakteristika von Dienstleistungen nicht erfasst; eine konstruktive Abgrenzung von Sach- und Dienstleistungen fehlt dabei weitgehend. Corsten (2001, 21) bezeichnet die Negativdefinition als eine „wissenschaftliche Verlegenheitslösung“.

Im Rahmen einer *institutionellen* Definition werden Dienstleistungen ausschließlich im tertiären Sektor einer Volkswirtschaft produziert. Dieser Wirtschaftssektor umfasst zum Beispiel Handel, Verkehr, Banken oder Versicherungen. Die institutionelle Definition schließt somit die Dienstleistungen der primären (Land- und Forstwirtschaft) und sekundären (Bergbau, verarbeitendes Gewerbe, Industrie) Sektoren aus und ist deswegen nicht vollständig (Bullinger/Schreiner 2003, 55; Scheer et al. 2003, 23; Corsten 2001, 21; Burr/Stephan 2006, 18ff.).

Die *konstitutive* Abgrenzung liegt vor, wenn der Dienstleistungsbegriff basierend auf konstitutiven Merkmalen explizit definiert wird. Ein konstitutives Merkmal ist „eine prägende Eigenschaft, die den Wesenskern einer Dienstleistung grundlegend beschreibt“ (Burr/Stephan 2006, 19). Mit der konstitutiven Abgrenzung des Dienstleistungsbegriffs wird versucht die Charakteristika von Dienstleistungen basierend auf den Gemeinsamkeiten von Services herauszuarbeiten und dadurch eine präzise Definition zu schaffen. Zu den konstitutiven Merkmalen zählen Immaterialität von Dienstleistungen und die Integration des externen Faktors in die Leistungserstellung (Bullinger/Schreiner 2003, 55; Scheer et al. 2003, 23; Corsten 2001, 21; Burr/Stephan 2006, 18ff.).

Abbildung 3-2 verdeutlicht die Kategorisierung von Dienstleistungen.

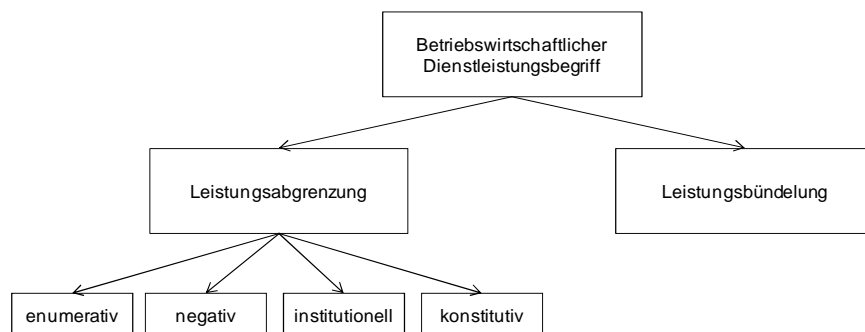


Abbildung 3-2: Betriebswirtschaftlicher Dienstleistungsbegriff

Quelle: In Anlehnung an (Bullinger/Schreiner 2003, 55; Scheer et al. 2003, 23; Corsten 2001, 21; Burr/Stephan 2006, 18ff.)

Die Charakterisierung von Dienstleistungen auf Grundlage konstitutiver Merkmale kann in weitere potenzial-, prozess- und ergebnisorientierte Definitionsansätze unterschieden werden (Corsten 2001, 26; Scheer et al. 2003, 23; Möller/Cassack 2008; Haller 2005, 13ff.; Fritzsche 2007, 6-9):

- Im Zentrum der *potenzialorientierten* Dienstleistungsdefinition steht die Fähigkeit eines Dienstleistungsanbieters zur Bereitstellung der Leistungserbringung. Somit wird die Dienstleistung als „menschliche oder maschinelle Leistungsfähigkeit interpretiert,

mit der dann am Nachfrager oder an dessen Verfügungsobjekt eine gewollte Änderung bewirkt oder ein Zustand erhalten werden soll“ (Corsten 2001, 21-22). Zur Erbringung einer Dienstleistung soll der Anbieter entsprechende Potenziale, wie Humanressourcen oder Maschinen, bereitstellen. Die potentialorientierte Dienstleistungsdefinition hebt den immateriellen Charakter der Dienstleistung als konstitutives Merkmal hervor. Dienstleistungen werden als Versprechen der Anbieter gegenüber dem Nachfrager in Bezug auf seine zu erbringenden Leistungen gesehen.

- Die *prozessorientierte* Definition von Dienstleistungen fasst eine Dienstleistung als Prozess zwischen dem Dienstleistungsanbieter und dem Dienstleistungsnachfrager auf. Durch die Definition einer Dienstleistung als Prozess wird der immaterielle Charakter von Dienstleistungen unterstrichen, da die Verrichtung eines Prozesses immateriell ist. Darüber hinaus wird durch diese Definition die Simultanität von Leistungserstellung und Leistungsabgabe hervorgehoben, d. h. die Erbringung und der Verbrauch von Dienstleistungen geschehen gleichzeitig. Somit wird bei der Dienstleistungserbringung das Leistungspotenzial des Anbieters aktiviert und mit externen Faktoren kombiniert. Die Dienstleistungserbringung kann erst dann stattfinden, wenn der externe Faktor, wie der Kunde selbst oder seine Verfügungsobjekte, in den Prozess integriert werden. Der Kunde ist dann das prozessauslösende und prozessbegleitende Element für die Umsetzung des Dienstleistungsproduktes (Scheer et al. 2003, 25).
- Die *ergebnisorientierte* Definition von Dienstleistungen versteht Dienstleistungen als ein Ergebnis eines Dienstleistungserstellungsprozesses und beschreibt somit „den Zustand, der nach vollzogener Faktorkombination, also nach Abschluss des Dienstleistungsprozesses, vorliegt“ (Scheer et al. 2003, 25). Im Fokus dieser Definition steht die Wirkung der Leistung auf Personen oder Objekte, die von der Erbringung der Dienstleistung betroffen sind. Somit kann sich das Dienstleistungsergebnis „in der Erhaltung, Wiederherstellung, Schaffung oder Verrichtung von Merkmalen und Merkmalausprägungen beim Kunden oder seinen Objekten“ (Fritzsche 2007, 7) ausdrücken. Auch bei dieser Definition wird der immaterielle Charakter von Dienstleistungen hervorgehoben.

Abbildung 3-3 verdeutlicht die Dimensionen einer Dienstleistung.

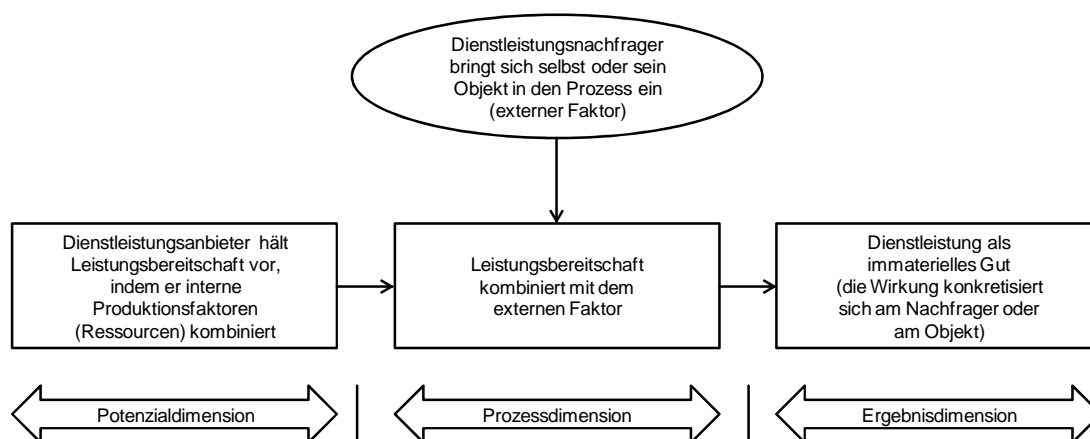


Abbildung 3-3: Dimensionen einer Dienstleistung

Quelle: In Anlehnung an (Corsten 2001, 26)

Es ist darauf hinzuweisen, dass allein das Merkmal der Immaterialität nicht als einziges Charakteristikum von Dienstleistungen angesehen werden darf. Zum einen beinhalten Dienstleis-

tungen auch materielle Komponenten, zum anderen gibt es auch immaterielle Güter, wie zum Beispiel Rechte, die aber nicht zu Dienstleistungen gehören.

Desweiteren soll eine Abgrenzung zu den produktbegleitenden sowie IT-basierten Dienstleistungen und IT-Dienstleistungen vorgenommen werden.

Produktbegleitende Dienstleistungen bezeichnen immaterielle Leistungen, die von einem Unternehmen ergänzend zu seinen Sachleistungen angeboten werden, um den Absatz seiner Sachleistungen zu fördern (Sturm/Bading 2008, 175; Schmid 2005, 70). Bereits 1997 haben 94% der befragten Unternehmen (von 1.329) angegeben, dass sie produktbegleitende Dienstleistungen anbieten ((Lay 1998, 3) zitiert in (Sturm/Bading 2008)). Zu den produktbegleitenden Dienstleistungen gehören bspw. Wartung, Betreibermodell, Finanzdienstleistungen oder technischer Kundendienst (Ernst 2008).

Durch das Anbieten von produktbegleitenden Dienstleistungen können Unternehmen sich noch stärker von der Konkurrenz differenzieren und dadurch neue Kunden gewinnen oder die Bindung der bestehenden Kunden verstärken (Rainfurth 2003, 1-3; Kalmbach et al. 2003). Produktbegleitende Dienstleistungen werden „rund um das Sachgut“ erbracht und sollen damit „das Problem des Kunden besser lösen als es die Lieferung des Sachguts allein könnte“ (Rainfurth 2003, 1). Diese Art der Dienstleistungen werden oft angeboten, wenn der Kunde nicht die entsprechenden Kompetenzen zur Verrichtung der Dienstleistung besitzt oder finanziell nicht in der Lage dazu ist (Koch 2010, 22f.). Allerdings sind nur wenige Kunden bereit für produktbegleitende Dienstleistungen zu zahlen, denn diese Dienstleistungen sind meist an das bestehende Produktangebot gekoppelt und „werden vom Kunden oft als Dreingabe verstanden“ (Sturm et al. 2007, 11). Da heutzutage die Dienstleistungen selbst zum Auslöser von Innovationen werden können, ist es Aufgabe von Unternehmen die Dienstleistungen profitabel zu gestalten, so dass sie das Produktgeschäft ihrerseits bestimmen (Sturm/Bading 2008).

Über *IT-Dienstleistungen* spricht man, „wenn IT-Systeme oder IT-Aktivitäten wesentlicher Bestandteil des Leistungsergebnisses sind und zusätzlich IT-Systeme oder andere IT-bezogene Faktoren sowohl als Potenzialfaktoren für die Sicherstellung der Leistungsbereitschaft bedeutsam als auch als externe Faktoren in den Leistungserstellungsprozess eingebunden sind“ (Böhmman 2003, 32). Typische Beispiele für IT-Dienstleistungen sind Beratungs-, Implementierungs- und Betriebsmanagementdienstleistungen. Somit werden IT-Dienstleistungen eingesetzt, um die Geschäftsaktivitäten von Unternehmen zu unterstützen. *IT-basierte* Dienstleistungen liegen vor, „wenn das Leistungsergebnis nicht auf IT-Systeme oder IT-Aktivitäten zielt, aber wenn IT-Systeme oder andere IT-bezogene Faktoren sowohl als Potenzialfaktoren für die Sicherstellung der Leistungsbereitschaft bedeutsam als auch als externe Faktoren in den Leistungserstellungsprozess eingebunden sind“ (Böhmman 2003, 32). Somit sind IT-basierte Dienstleistungen dadurch gekennzeichnet, dass das Leistungsergebnis aus informationslogistischen Leistungen besteht (Krcmar 2010, 690). Zur Erbringung von IT-basierten Dienstleistungen werden IT-Systeme und Kommunikationssysteme eingesetzt (Laqua 2007). Es werden drei Kategorien von IT-basierten Dienstleistungen unterschieden (Meyer/Husen 2007):

- IT-unterstützte Dienstleistungen, bei denen das Primärprodukt die Dienstleistung selbst ist (Terminals für den Fahrkartenverkauf).
- IT-begleitende Dienstleistungen, bei denen das Primärprodukt die Informationstechnologie darstellt. Diese Kategorie weist Ähnlichkeiten zu produktbegleitenden Dienst-

leistungen auf, wobei die Rolle der Sachleistung durch die Informationstechnologie übernommen wird.

- Integriertes Hybridprodukt, bei dem beide Komponenten (Dienstleistung und Informationstechnologie) gleiche Priorität besitzen und in Form eines Leistungsbündels dem Kunden zur Verfügung gestellt werden, so dass der Kunde die einzelnen Bestandteile nicht einzeln wahrnehmen kann (Meyer/Husen 2007), zu dessen Entwicklung eine enge Verzahnung zwischen Software- und Dienstleistungsentwicklung erforderlich ist (Böttcher/Meyer 2004).

Aus den oben angeführten Definitionen wird ersichtlich, dass der Übergang zwischen dem Absatz reiner Sachleistungen und reiner Dienstleistungen fließend ist (Abbildung 3-4). Der Wunsch nach einer abgestimmten und integrierten Entwicklung von Sach- und Dienstleistungen, wobei Software als Sachleistung zu verstehen ist, wird immer höher.

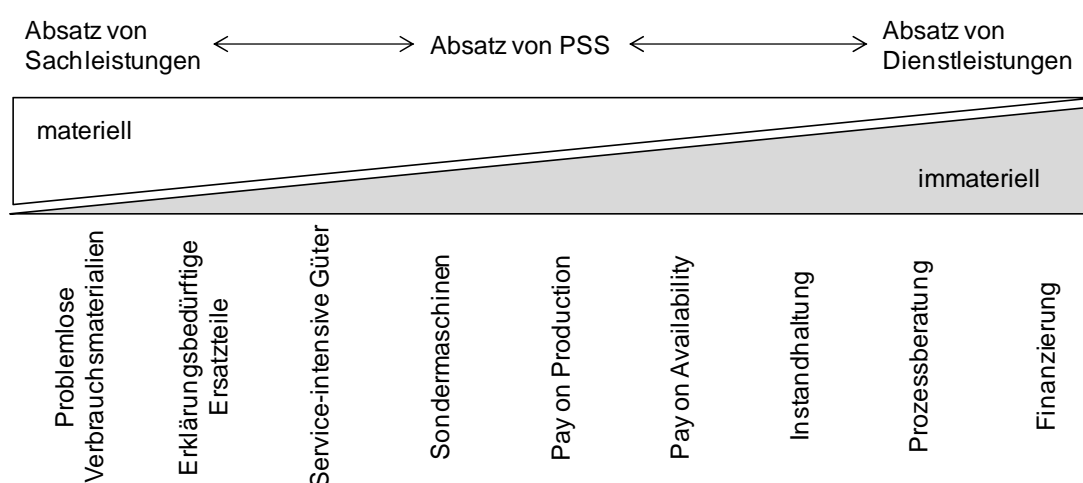


Abbildung 3-4: Unterschiede beim Absatz von Sach-, Dienstleistungen und PSS
Quelle: In Anlehnung an (Schmid 2005, 67; Wimmer et al. 2008)

3.1.1.3 Abgrenzung des Begriffs „Product Service Systems“

Um die Differenzierungspotenziale gegenüber der Konkurrenz auszubauen, auf die geänderten Bedingungen des nachfrageorientierten Markts zu reagieren und dabei umfassende Lösungen für kundenindividuelle Probleme anzubieten, ändern viele Unternehmen ihre Philosophie von „product-centric“, in der eine Sachleistung oder eine Dienstleistung das Kerngeschäft bildeten, in „customer-centric“ (Galbraith 2002) und entwickeln sich zu so genannten Lösungsanbietern (Doster/Roegner 2000; Böhmann/Krcmar 2006; Berkovich et al. 2009c). Der Kunde ist dabei ein Individuum oder eine Organisation, das oder die von der angebotenen Lösung direkt oder indirekt profitiert und dessen oder deren Anforderungen für die Bereitstellung des PSS entscheidend sind. Die von den Unternehmen angebotenen Lösungen stellen integrierte Leistungsbündel von Sach- und Dienstleistungen dar, die für den Kunden nur als ein Ganzes erkennbar sind (Leimeister/Glauner 2008; Krcmar 2010, 600; Knackstedt et al. 2008). Als Leistungsbündel werden „Aggregate“ bezeichnet, die „aus mehreren Sachleistungen bzw. mehreren Dienstleistungen [...] zusammengesetzt sind“ (Becker et al. 2009, 16). Solche Lösungen sind unter dem Namen *Hybride Produkte*, *Hybride Leistungsbündel* oder *Product Service Systems (PSS)* bekannt (Halen et al. 2005, 16ff.; Mont 2002; Böhmann/Krcmar 2007). Der Begriff „hybrides Leistungsbündel“ (Synonym für *Product Ser-*

vice System) drückt dabei die Zusammensetzung aus Sach- und Dienstleistungskomponenten aus. Miller et al. (2002) unterstreicht den Unterschied zwischen herkömmlichen Produkten oder Dienstleistungen und einer Lösung, da “[...] products are mostly about functionality, solutions are about outcomes that make life easier or better for the client“.

In Tabelle 3-2 wird ein Überblick über die Definitionen der Begriffe „Product Service Systems“, „Hybrides Produkt“ und „Lösung“ gegeben.

Begriffe	Definitionen
Lösung	„[...] a solution is a customized and integrated combination of goods and services for meeting a customer’s business needs“ (Tuli et al. 2007, 1).
	„A solution is a customized, integrated combination of products, services and information that solves a customer problem“ (Sawhney et al. 2006, 78).
	“[...] solutions are [...] integrated combinations of products and/or services that are unusually tailored to create outcomes desired by specific clients or types of clients“ (Miller et al. 2002, 3).
Product Service System / PSS / Hybrides Produkt	„Ein hybrides Produkt ist eine Leistung, die aus mehreren Teilen besteht, welche nicht mehr ohne weiteres einzeln erkennbar sind, deren unterschiedliche Eigenschaften aber das hybride Produkt prägen“ (Leimeister/Glauner 2008, 248).
	„A PSS is an integrated product and service offering that delivers value in use“ (Baines et al. 2007, 1545).
	„A product–service system (PSS) can be defined as consisting of ‘tangible products and intangible services designed and combined so that they jointly are capable of fulfilling specific customer needs’“ (Tukker 2004, 246).
	„A Product Service system (PS system, or product service combination) is a marketable set of products and services, jointly capable of fulfilling a client's need. Understanding PS systems is interesting for companies as well as governments. PS system knowledge enables governments to formulate a next step in policy concerning sustainable production and consumption patterns. PS system knowledge enables companies to find strategic options for business growth, renewal, innovation and diversification“ (Goedkoop et al. 1999, 3).
	„[...] a PSS should be defined a system of products, services, supporting networks and infrastructure that is designed to be: competitive, satisfy customer needs and have a lower environmental impact than traditional business models“ (Mont 2002, 239).
	„Nachhaltige Produkt-Service-Systeme (für Konsumenten) werden als kommerzielles Angebot einer Kombination von Produkten und Dienstleistungen (oder eines unentgeltlichen Angebots, das aber potentiell vermarktbar wäre) gesehen, die in der Lage sind, ein Konsumentenbedürfnis zu erfüllen, insbesondere als Alternative zum Produktkauf“ (Nippa et al. 2007, 21).
	„Hybride Produkte sind Kombinationen aus Sach- und Dienstleistungen, die am Markt als integrierte Leistungsbündel angeboten werden. Ziel der Kombination von Produkten und Dienstleistungen ist es, kundenspezifische Problemlösungen anbieten zu können“ (Böhm/Krcmar 2007).

Tabelle 3-2: Definitionen von Product Service Systems

Quelle: Eigene Darstellung

Aus den Definitionen in Tabelle 3-2 lässt sich feststellen, dass Product Service Systems kundenindividuelle Lösungen sind, die einen Mehrwert für die Kunden schaffen, indem sie mehr Funktionalität und Flexibilität im Vergleich zu herkömmlichen Produkten und Dienstleistungen anbieten (Halen et al. 2005, 22). Für die Kunden geht es deswegen beim Erwerb von PSS nicht mehr um den Besitz des Produktes, sondern um den Nutzen, den sie aus der Benutzung der Kombination aus Sach- und Dienstleistungen erhalten (Baines et al. 2007). Durch den Erwerb von Lösungen kann der Kunde bestimmte Geschäftsfelder nach außen verlegen und sich dadurch auf seine Kernkompetenzen fokussieren (Galbraith 2002).

Abbildung 3-5 zeigt den Unterschied zwischen dem traditionellen Angebot an reinen Sach- oder Dienstleistungen und PSS. Während bei reinen Sach- und Dienstleistungen das Geschäft mit dem Kunden mit dem Erwerb von Leistungen durch den Kunden endet, übernimmt der Anbieter die Integration der Lösung in die Prozesse des Kunden. Somit stellt der Anbieter dem Kunden die Funktionalität der Lösung bereit und erreicht eine langfristige Kundenbindung, Wettbewerbsvorteile und auch langfristig höhere Margen im Vergleich zum traditionellen Geschäft (vgl. (Halen et al. 2005, 19ff.; Tukker 2004; Tan et al. 2007)).

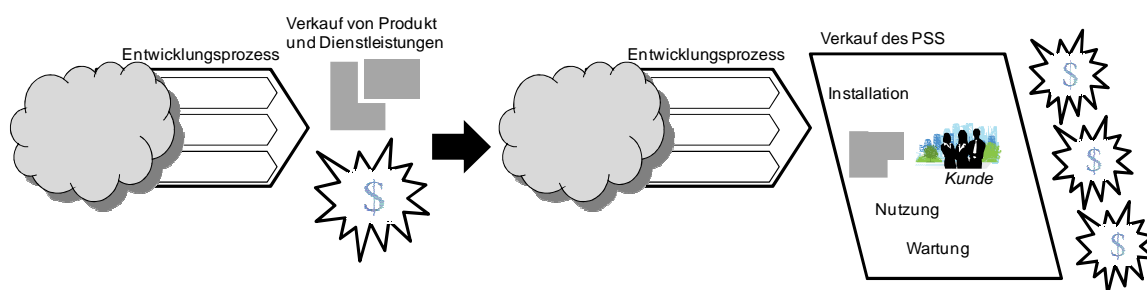


Abbildung 3-5: Traditionelles Angebot an Produkten und Dienstleistungen versus PSS
Quelle: In Anlehnung an (Tan et al. 2007)

Im Falle von PSS können daher nur die Komponenten auf Vorrat produziert werden, die nicht kundenindividuell sind und ohne Einwirkung des Kunden erstellt werden können (Schmid 2005, 68-70). Darüber hinaus ist ein wesentlicher Unterschied zwischen der Entwicklung und dem Angebot an Sachleistungen und PSS, dass in die Erstellung von PSS ein externer Faktor, der Nachfrager, integriert wird (Spath/Demuß 2003, 472-473). Es ist daher wichtig, den Kunden (Nachfrager) schon in die frühen Phasen des Entwicklungsprozesses von PSS einzubeziehen, um dadurch sicherzustellen, dass die erstellte Lösung die Kundenanforderungen vollständig erfüllt (Baines et al. 2007). Somit lässt sich der Leistungserbringungsprozess von PSS mit dem von Dienstleistungen vergleichen (Abbildung 3-6).

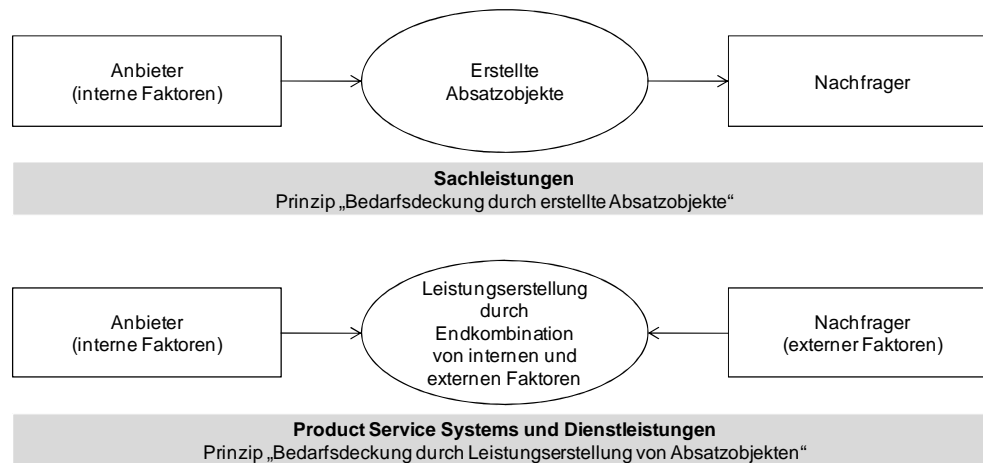


Abbildung 3-6: Unterschiedliche Bedarfsdeckung bei Sachleistungen, hybriden Produkten und Dienstleistungen

Quelle: In Anlehnung an (Spath/Demuß 2003, 472; Hull et al. 2011, 26; Meyer/Mattmüller 1987)

Product Service Systems können anhand von zwei Dimensionen charakterisiert werden: der Interaktion mit dem Kunden während des Erstellungsprozesses sowie des Grads der Immaterialität des Leistungsergebnisses (vgl. (Burianek et al. 2007, 2-5; Leimeister/Glauner 2008, 249f.; Engelhardt et al. 1993, 417)). Es werden vier Typen von Leistungsangeboten definiert: standardisierte Sach- bzw. Dienstleistungen („Massen“-Produkt bzw. -Dienstleistung) und kundenindividuelle Sach- bzw. Dienstleistungen (Abbildung 3-7).

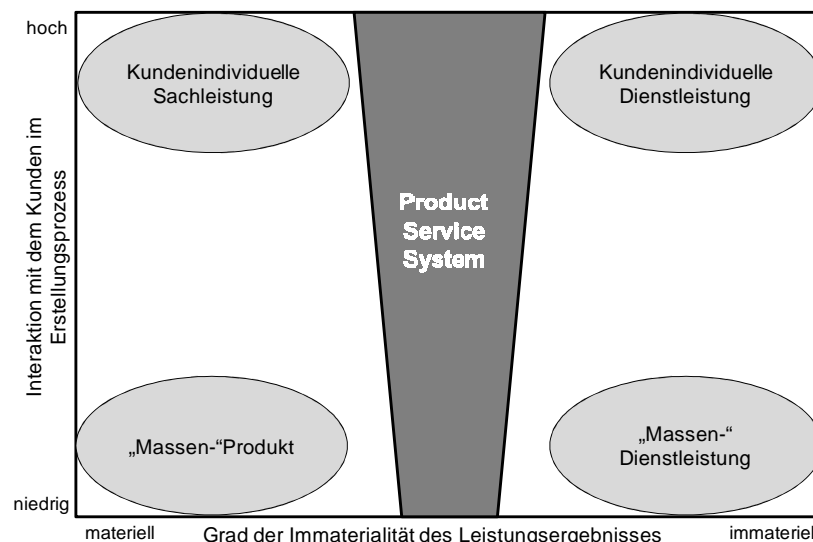


Abbildung 3-7: Gütertypologische Abgrenzung von PSS

Quelle: In Anlehnung an (Engelhardt et al. 1993, 417) zitiert in (Leimeister/Glauner 2008, 250)

Im Fall von PSS ist es schwierig, sie einem bestimmten Typ zuzuordnen. Das Leistungsangebot von PSS kann vom Verkauf einer Anlage und den dazugehörigen Installation und Wartung bis zum Verkauf von Ergebnissen einer Leistung (zum Beispiel ein IT-gestütztes Facility Management) und somit deren Integration in die Prozesse des Kunden stark variieren (vgl. (Burianek et al. 2007, 4ff.; Leimeister/Glauner 2008, 250)). Durch ihre Heterogenität können PSS auf der Achse der Immaterialität des Leistungsergebnisses zwischen materiell und imma-

teriell platziert werden, während die Interaktion mit dem Kunden eher hoch eingestuft wird. Die gütertypologische Abgrenzung von PSS, Sach- und Dienstleistungen trägt zum besseren Verständnis von PSS bei, für eine Typologisierung von PSS ist sie jedoch unzureichend (Leimeister/Glauner 2008, 249).

Burianek et al. (2007) haben einen Kategorisierungsansatz für PSS entwickelt, dem eine Charakterisierung von PSS basierend auf ihrer Komplexität zu Grunde liegt. Der Ansatz soll ermöglichen die unterschiedlichen Ausprägungen von PSS zu ordnen und die Entscheidungen bei der Gestaltung und Vermarktung von kundenindividuellen Leistungen zu unterstützen. Dem Ansatz liegt die Idee zu Grunde, dass die Komplexität durch ein Kontinuum ausgedrückt wird („gering“ bis „hoch“) und nicht nur zwei Extremausprägungen („gering“ und „hoch“) hat (Burianek et al. 2007). Um PSS zu kategorisieren, wurden sieben Kriterien vorgeschlagen, die auf Basis bestehender Typologisierungsansätze aus der Sach- und Dienstleistungsentwicklung ermittelt wurden. Folgende Kriterien wurden vorgeschlagen, die die Komplexität von PSS und die zusammenhängende Leistungsbringung bestimmen:

1. Art des Kundennutzens (produktorientiert \longleftrightarrow ergebnisorientiert),
2. Umfang des Leistungsangebotes (partiell \longleftrightarrow umfassend),
3. Anzahl/Heterogenität der Teilleistungen (gering \longleftrightarrow hoch),
4. Grad der technischen Integration (niedrig \longleftrightarrow hoch),
5. Grad der Integration in die Wertschöpfungsdomäne des Kunden (niedrig \longleftrightarrow hoch),
6. Grad der Individualisierung (niedrig \longleftrightarrow hoch),
7. Zeitliche Dynamik/Veränderlichkeit der Leistungserbringung (gering \longleftrightarrow hoch).

Abhängig vom Leistungsangebot kann die Ausprägung eines Kriteriums variieren. Je höher bspw. der Grad der Individualisierung einer Lösung ist, desto höher ist die Komplexität.

Eine weitere Kategorisierung von PSS schlägt Tukker (2004) vor (Abbildung 3-8).

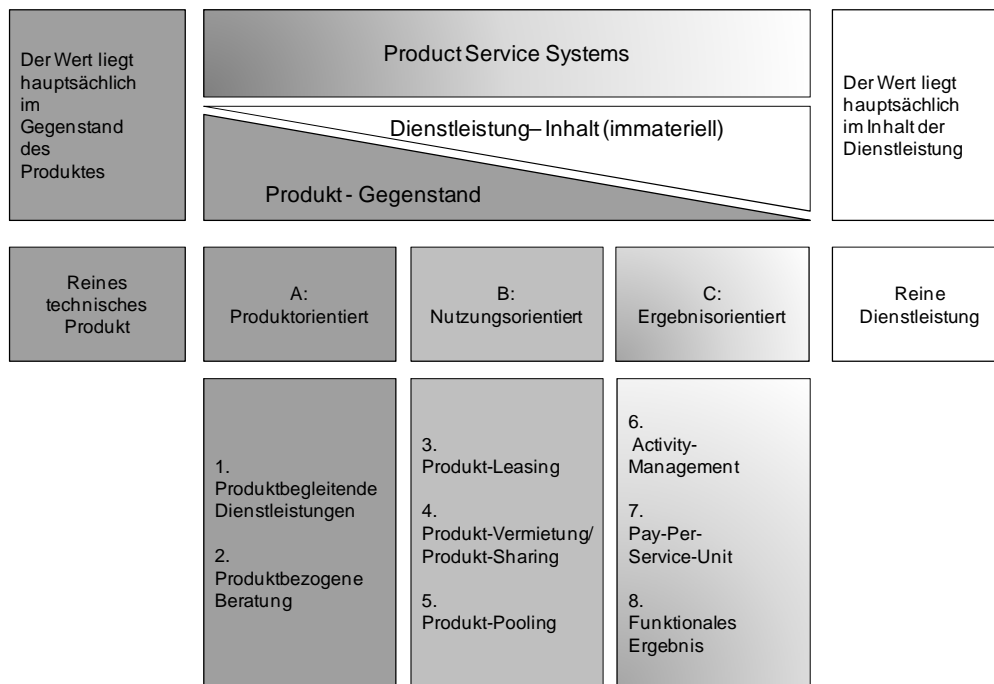


Abbildung 3-8: Kategorien von Product Service Systems

Quelle: In Anlehnung an (Tukker 2004) zitiert in (Wimmer et al. 2008)

Product Service Systems sind zwischen reinen Sach- und Dienstleistungen anzusiedeln und in drei Kategorien zu unterscheiden (Tukker 2004):

- A. *Produktorientierte PSS*: Der Fokus liegt auf dem Verkauf von Sachleistungen, zu denen zusätzliche Dienstleistungen angeboten werden. Zu solchen Dienstleistungen zählen Finanzierung, Wartung, Recycling oder Beratung.
- B. *Nutzungsorientierte PSS*: Der Verkauf von Sachleistungen spielt noch eine wichtige Rolle, steht aber nicht mehr im Vordergrund des Geschäftsmodells. Die Sachleistungen bleiben im Besitz des Anbieters, sind aber für die Abnehmer verfügbar. Ein Beispiel dafür sind Miet- oder Sharing-Systeme.
- C. *Ergebnisorientierte PSS*: Der Anbieter und der Kunde treffen entsprechende Vereinbarungen über das Ergebnis der Leistung. Es ist alleine dem Anbieter überlassen, wie er die Leistungserbringung gestaltet und das Ergebnis dem Kunden liefert.

Diese drei Kategorien lassen sich in weitere Unterkategorien einteilen (Tukker 2004):

A. Produktorientierte PSS:

- 1 Produktbegleitende Dienstleistungen: der Anbieter verkauft nicht nur das technische Produkt, sondern bietet auch Dienstleistungen an, die während der Nutzungsphase benötigt werden. Das Angebot an Dienstleistungen kann die Form eines Wartungsvertrags oder eines Finanzierungssystems annehmen.
- 2 Produktbezogene Beratung: der Anbieter bietet Beratung zum Beispiel zur effizienten Nutzung des verkauften Produktes.

B. Nutzungsorientierte PSS:

- 3 Produkt-Leasing: das Produkt wird vom Kunden geleast, wobei der Anbieter das Eigentumsrecht behält und für die Wartung, Kontrolle und Reparatur zuständig ist. Der Kunde zahlt für die Nutzung des Produktes und kann es unbegrenzt benutzen.

- 4 Produktvermietung oder -sharing: das Produkt bleibt im Besitz des Anbieters, der die Wartung, Kontrolle und Reparatur zu gewährleisten hat. Das technische Produkt wird dabei sequentiell genutzt. Das bedeutet, der Kunde zahlt für die Nutzung des Produktes, kann es aber nicht unbegrenzt nutzen, da auch andere Nutzer das Produkt nutzen können.
- 5 Produkt-Pooling: das technische Produkt bleibt im Besitz des Anbieters, der die Wartung, Kontrolle und Reparatur zu gewährleisten hat. Der Kunde zahlt für die Nutzung des Produktes, kann es aber zusammen mit anderen Kunden gleichzeitig benutzen.

C. Ergebnisorientierte PSS

- 6 Activity-Management / Outsourcing: Teile von Unternehmensaktivitäten werden an einen externen Anbieter ausgelagert. Beispiele dafür sind Catering oder Büoreinigung.
- 7 Pay-Per-Service-Unit: im Zentrum der vom Kunden gekauften Leistung liegt das technische Produkt. Allerdings bezahlt der Kunde nicht für das technische Produkt, sondern für die Funktion des Produktes basierend auf dem Nutzungsgrad oder der Quantität der Nutzung. Ein Beispiel dafür ist ein Angebot von Kopiergeräteherstellern, die Kopiergeräte anbieten und einen bestimmten Betrag für jede Einzelkopie abrechnen. Die Kopiergerätehersteller übernehmen dabei Aktivitäten, wie die Zustellung von Papier und Patronen, Wartung und Reparatur.
- 8 Funktionales Ergebnis: der Anbieter und der Kunde vereinbaren ein Ergebnis, das die Lösung des Problems des Kunden darstellt. Das Ergebnis der Lösung ist nicht an ein bestimmtes technologisches System oder technisches Produkt gebunden.

Somit lässt sich feststellen, dass unterschiedliche Arten von PSS existieren, dessen Ziel es ist individuelle Lösungen für Kunden bereitzustellen. Die unten angeführten empirischen Studien belegen die Bedeutung von PSS für die Unternehmen.

3.1.1.4 Bedeutung von Product Service Systems für die Unternehmen

Die empirische Studie, die von Sturm et al. (2007) unter 99 mittelständischen Unternehmen aus der Investitionsgüterindustrie aus Baden-Württemberg und seinen umliegenden Bundesländern durchgeführt wurde, belegt, dass die Bedeutung von Kundenlösungen für die Unternehmen immer wichtiger wird. Auf die Frage „wie hoch der Erfolgsanteil von Dienstleistungen am finanziellen Erfolg eines Unternehmens ist“, haben ca. 48% der Unternehmen mit „hoch“ und „sehr hoch“ beantwortet (vgl. Abbildung 3-9).

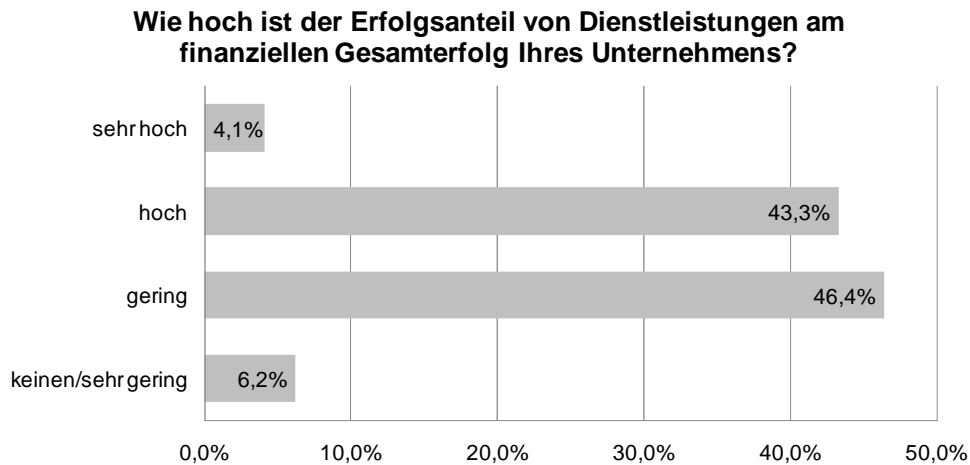


Abbildung 3-9: Finanzieller Erfolgsanteil von Dienstleistungen am Gesamterfolg der Unternehmen

Quelle: (Sturm et al. 2007, 20)

Weiterhin halten ca. 98% der Befragten das Angebot an PSS für wichtig, allerdings sind ca. 25% der Unternehmen mit ihrem Angebot an Kundenlösungen nicht zufrieden (Sturm et al. 2007, 18-19). Ebenso glauben ca. 89,8% der Unternehmen, dass die Kunden in Zukunft verstärkt nach kundenindividuellen Lösungen verlangen werden (Sturm et al. 2007, 23). Im Zentrum der Entwicklung von PSS stehen Kunden selbst mit ihren Bedürfnissen und Anforderungen. Ca. 72,4 % der Unternehmen halten eine regelmäßige Identifikation von Kundenbedürfnissen und -wünschen für sehr wichtig und ca. 27,6% halten es für wichtig (Sturm et al. 2007, 23). Die Identifikation von Kundenbedürfnissen und -wünschen ist die Grundlage für die Entwicklung von Lösungen, die einen echten Mehrwert für die Kunden schaffen und somit die Kundenbindung verstärken (vgl. Abbildung 3-10).

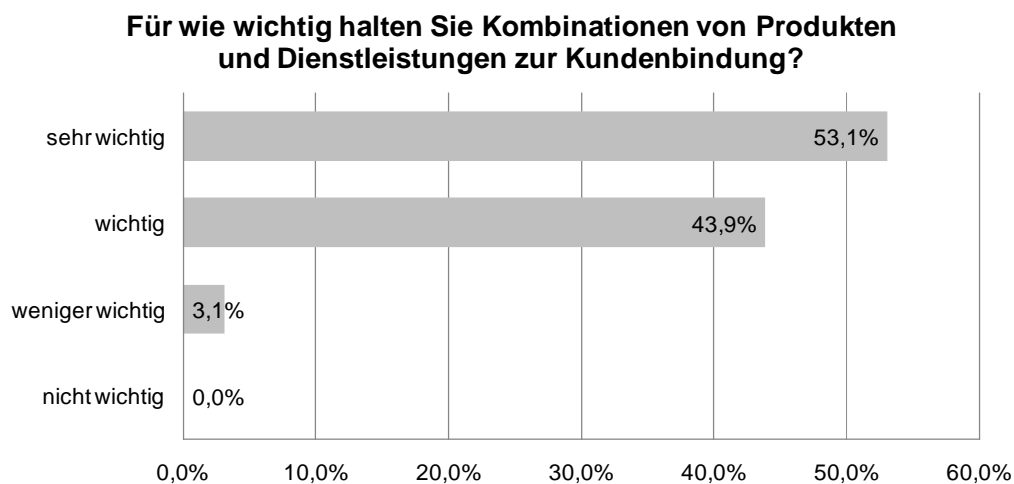


Abbildung 3-10: Bedeutung von Lösungen zur Kundenbindung

Quelle: (Sturm et al. 2007, 27)

Ebenso wird die Differenzierung von den Wettbewerbern als wichtiger Faktor für das Anbieten von PSS genannt. Auf die Frage „Wie viel können „Lösungen“ zur Differenzierung gegenüber Ihrem Wettbewerb beitragen?“ haben 57% der Unternehmen „viel“, während 34% der Unternehmen sogar „sehr viel“ angegeben (vgl. Abbildung 3-11).

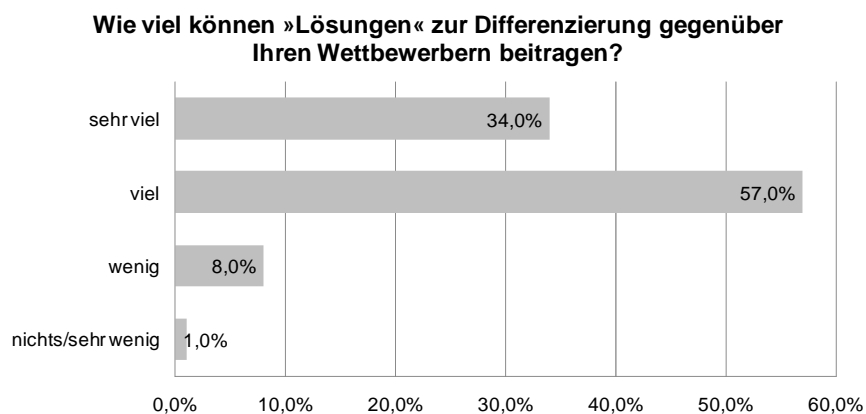


Abbildung 3-11: Beitrag von Lösungen zur Differenzierung gegenüber Wettbewerbern
Quelle: (Sturm et al. 2007, 27)

Die Studie hat weiterhin gezeigt, dass fast 95% der befragten Unternehmen positiv auf die Frage nach geplantem Wachstum durch Kombinationen aus Sach- und Dienstleistungen geantwortet haben (Sturm et al. 2007, 28). Als Fazit der Studie lässt sich feststellen, dass Unternehmen die Potenziale von PSS erkannt haben und deswegen den Fokus ihrer Geschäftsstrategie so ändern, dass nicht Sach- oder Dienstleistungen das Kerngeschäft bilden, sondern die Aktivitäten, die im Zusammenhang mit dem Nutzen des Produktes stehen und die dem Kunden einen Mehrwert bringen (Sturm/Bading 2008; Sturm et al. 2007).

PSS sind dadurch gekennzeichnet, dass “a supplier integrates products and/or services to deliver an especially useful outcome” (Miller et al. 2002, 5). Dafür sollen die Entwicklung von Sach- und Dienstleistungen aufeinander abstimmen werden. Obwohl die Relevanz der integrierten Entwicklung von den Unternehmen erkannt worden ist (Abbildung 3-12), sind ca. 49% der befragten Unternehmen mit der integrierten Entwicklung von technischen Produkten und Dienstleistungen in ihrem Unternehmen wenig oder nicht zufrieden (Sturm et al. 2007, 35).

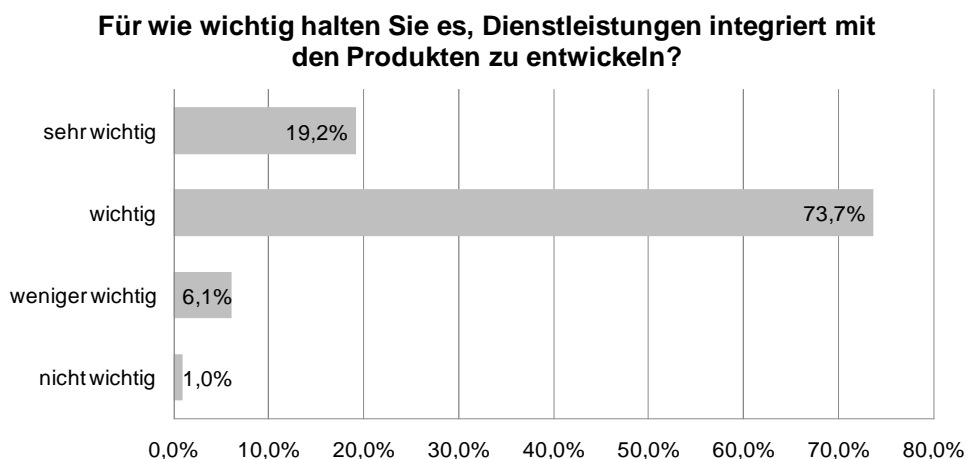


Abbildung 3-12: Bedeutung der integrierten Entwicklung von DL und Produkten
Quelle: (Sturm et al. 2007, 35)

Die unabhängige Entwicklung von Sach- und Dienstleistungen in der Praxis, führt oft dazu, dass Dienstleistungen unflexibel sind und eine Kombination aus Sach- und Dienstleistungen nicht effektiv gestaltet werden kann, wodurch die Kundenanforderungen nicht maximal er-

füllbar sind (Langer et al. 2009, 209). Dafür ist eine integrierte und abgestimmte Entwicklung von Sach- und Dienstleistungen erforderlich, um „abgestimmte, sich ergänzende Leistungsbündel“ anzubieten (Spath/Demuß 2003).

3.1.1.5 Zusammenfassung

Basierend auf den in Kapiteln 3.1.1.1, 3.1.1.2, 3.1.1.3 und 3.1.1.4 erläuterten Begriffen, stellt Tabelle 3-3 einen Überblick über die für diese Arbeit relevanten Begriffe dar.

Begriff	Definition
Produkt	Ein Produkt ist „eine Leistung oder eine Gruppe von Leistungen, die von Stellen außerhalb des jeweils betrachteten Fachbereichs (innerhalb oder außerhalb der Organisation) benötigt werden“ (Scheer et al. 2003, 21).
Technisches Produkt, Sachleistung	Ein technisches Produkt (wird im Rahmen der Arbeit als Synonym zu Sachleistung verwendet) ist ein gegenständliches, künstliches Objekt mit einer beschreibbaren technischen Funktion (Volz 1997, 32).
Dienstleistung	Eine Dienstleistung ist ein unter Einsatz externer Produktionsfaktoren für den fremden Bedarf produziertes immaterielles Wirtschaftsgut (Fritzsche/Maleri 2003, 197).
PSS	PSS sind „Kombinationen aus Sach- und Dienstleistungen, die am Markt als integrierte Leistungsbündel angeboten werden. Ziel der Kombination von Produkten und Dienstleistungen ist es, kundenspezifische Problemlösungen anbieten zu können“ (Böhmman/Krcmar 2007).

Tabelle 3-3: Begriffsdefinitionen im Rahmen dieser Arbeit

Quelle: Eigene Darstellung

3.1.2 Eigenschaften von Product Service Systems

Zur Charakterisierung von PSS schlägt Sawhney (2006) (zitiert in Sturm et al. (2007, 12-13)) drei Kriterien vor: Leistungsintegration, Prozessintegration und Individualisierung. Erst wenn ein Leistungsbündel alle diese Kriterien erfüllt, kann es als Lösung bezeichnet werden. Ähnliche Eigenschaften von PSS können bei Böhmman und Krcmar (2007), Sturm und Bading (2008), Halen et al. (2005) und Zellner (2008) gefunden werden. Zusammenfassend lassen sich folgende Eigenschaften von PSS feststellen:

1. *Bündelung materieller und immaterieller Komponenten.* PSS bestehen aus integrierten Bündeln aus Sach- und Dienstleistungskomponenten. Die Sachleistung kann entweder Hardware oder Software oder eine Kombination aus beiden sein. Im Falle einer Kombination aus Hardware- und Software-Komponenten spielt Software eine entscheidende Rolle bei der Bereitstellung von innovativen Funktionalitäten, da sie zur Bereitstellung von Basisfunktionalitäten, wie Speicherung, Verarbeitung und Kommunikation zur Verfügung stehender Ressourcen eingesetzt wird (Krcmar 2010). Wenn die Sachleistungskomponente eine Software ist, wird das PSS in Form von „Software as a Service“ angeboten (Berkovich et al. 2010a). In diesem Fall besitzen Dienstleistungen und Sachleistungen in Form von Informationstechnologien die gleiche Priorität (Meyer/Husen 2007, 14). Im Rahmen dieser Arbeit werden nur die PSS betrachtet, die sich aus einer Kombination von Hardware, Software und Dienstleistungen zusammensetzen.

Die Bündelung von materiellen und immateriellen Komponenten bedeutet, dass diese Komponenten von verschiedenen Domänen wie Produkt-, Software- und Dienstleistungsentwicklung erbracht werden (Zellner 2008; Leimeister/Glauner 2008). Die Domänen haben ein unterschiedliches Verständnis und unterschiedliche Vorgehensweisen bei der Entwicklung von den einzelnen Komponenten der Lösung. Während die Kunden bei immateriellen Leistungen mehr Unsicherheit zeigen, da Dienstleistungen oft auf Erfahrungs- oder Vertrauensaspekten beruhen, ist die Verfügbarkeit der materiellen Leistungen ein Erfolgsfaktor (Böhmman/Krcmar 2007). Um eine erfolgreiche Lösung für Kundenprobleme anzubieten, sollen die Domänen, die an der Entwicklung von PSS beteiligt sind, in der Lage sein ein gemeinsames Verständnis für das zu lösende Problem zu schaffen und die Komponenten des PSS integriert und abgestimmt zu entwickeln (Berkovich et al. 2010b; Kersten et al. 2006), so dass die Verbindung zwischen den Domänen über eine reine Kooperation hinausgeht (Meyer/Husen 2007, 15).

2. *Integration.* Die Integration bedeutet „die technisch-organisatorische Zusammenführung der Teilleistungen einer Lösung und deren Einbettung in den Wertschöpfungsprozeß des Kunden“ (Böhmman/Krcmar 2007, 244). Der technischen Zusammenführung der Teilleistungen liegt die Integration von Teilsystemen einer Lösung zu einem Gesamtsystem zu Grunde. Somit zielt eine technische Integration auf eine abgestimmte Verbindung von Sach- und Dienstleistungen ab (Bonnemeier et al. 2007; Burianek et al. 2007). Die organisatorische Zusammenführung der Teilleistungen erfordert eine übergreifende Festlegung von Geschäfts- und Unterstützungsprozessen, die für die Entwicklung, Nutzung und den Betrieb des PSS notwendig sind (Böhmman/Krcmar 2007). Ein weiterer Aspekt ist die Einbettung der Lösung sowohl technisch als auch organisatorisch in die Wertschöpfungsprozesse des Kunden. Dafür muss die Lösung in bestehende Systemlandschaften und Prozesse des Kunden aber auch in die Nutzungs-, Entwicklungs- Betriebsprozesse des Kunden integriert werden (Böhmman/Krcmar 2007). Deswegen muss sich der Anbieter von PSS mit den Abläufen des Kunden genau auskennen, um die Integration erfolgreich zu gestalten.
3. *Individualisierung.* Die Kombination von Sach- und Dienstleistungsbestandteilen soll auf die individuellen Bedürfnisse und Anforderungen des Kunden ausgerichtet sein. Dabei werden als Kunden sowohl Einzelkunden als auch Kundengruppen oder Branchen betrachtet (Galbraith 2002). Nur durch das Anbieten kundenspezifischer Problemlösungen kann der Mehrwert für den Kunden gesteigert werden. In diesem Kontext hängt der Erfolg eines Lösungsanbieters von seinen Fähigkeiten, seine Leistungen an die Kundenanforderungen anzupassen, ab (Sawhney et al. 2006). Ein wichtiger Aspekt dabei ist zu überprüfen, welche standardisierten Teilleistungen angeboten werden können und welche Teilleistungen auf die Kundenanforderungen hin angepasst werden sollen.
4. *Transformation.* Die Transformation bedeutet den Nutzen für den Kunden durch den Erwerb der Lösung zu steigern. Dafür müssen die Bedingungen beim Kunden zur Integration der Lösung sowie die Planung und Implementierung von möglichen Veränderungen in bestehenden Systemlandschaften und Prozessen des Kunden verstanden werden. Die Transformation kann sowohl die technische Ebene, wie die Migration oder Ablösung von bestehenden Altsystemen, als auch die organisatorische Ebene, wie Änderungen an der Organisations- und Mitarbeiterstruktur, betreffen (Böhmman/Krcmar 2007).

5. *Modularisierung*. Im Zusammenhang mit PSS spielt die Modularisierung eine wichtige Rolle. Dabei bedeutet die Modularisierung, dass zur Erstellung eines Systems mehrere Teilsysteme zusammengefügt werden. Durch die Modularisierung werden Systeme anpassungsfähiger gemacht, da Module eine spezifische Funktionalität kapseln und die Kommunikation mit anderen Teilsystemen über entsprechende Schnittstellen erfolgt (Beverungen et al. 2008, 221). PSS setzen sich aus standardisierten und kundeneindeividuellen Sach- und Dienstleistungskomponenten zusammen, die auch als Module bezeichnet werden und in Wechselwirkung zueinander stehen (Abramovici/Schulte 2005). Solche zusammengesetzte PSS können mit weiteren Sach- und Dienstleistungsmodulen zu einem neuen PSS kombiniert werden. Die Kombination der einzelnen Module kann entweder durch einen einzelnen Anbieter, einen Anbieterverbund oder einen dritten Anbieter zur Verfügung gestellt werden, der die unterschiedlichen Module (Sachleistungen, Dienstleistungen und PSS) bedarfsgerecht zusammenführt und kombiniert. Darüber hinaus wird durch die Modularisierung die Wiederverwendung sowie eine stärkere Standardisierung von Modulen unterstützt, da die Anpassung eines PSS auf die Bedürfnisse des Kunden durch eine Rekombination von Modulen erfolgen kann (Böhmman 2003, 2f.).

3.2 Product Service Systems in der Praxis

3.2.1 Product Service Systems aus dem Blickwinkel von Anbieter und Kunden

Nicht jede Kombination aus Sach- und Dienstleistungen stellt eine kundenindividuelle Lösung dar. „Some companies think they are selling solutions when they are merely bundling products that create little value when offered together“ (Johansson et al. 2003). Deswegen soll der Kunde durch den Erwerb von PSS einen Mehrwert im Vergleich zum Erwerb reiner Sach- oder Dienstleistungen erhalten (Doster/Roegner 2000, 52) und somit „nicht nur ein beliebig zusammengestelltes Produkt- und Dienstleistungsbündel, sondern ein auf seine Nutzungsanforderungen abgestimmtes und zuverlässig einsetzbares Gesamtsystem“ erhalten (Böhmman/Krcmar 2007, 244). Durch den Erwerb von PSS können die Kunden Aktivitäten, die sie vorher selbst ausgeführt haben, auf den Anbieter übertragen und dadurch auch die damit verbundenen Risiken abgeben. Zudem können sie Komplexitätsreduzierung im eigenen Geschäft erreichen, sowie sich auf das eigentliche Geschäft fokussieren (Mont 2002; Tukker 2004).

Die Anbieter von PSS profitieren durch die Möglichkeit des Erschließens neuer Märkte und einer noch stärkeren Differenzierung von den Wettbewerbern, da die Differenzierung alleine durch Sach- oder Dienstleistungen nicht mehr für einen langfristigen Erfolg ausreicht (Knackstedt et al. 2008; Leimeister/Glauner 2008; Tukker 2004). Der Kundenkontakt wird verbessert, die Kundenzufriedenheit sowie die Kundenbindung im Vergleich zum Angebot herkömmlicher Sach- oder Dienstleistungen steigen (Goedkoop et al. 1999, 11ff.; Mont 2002; Burianek/Reichwald 2009). Somit übernimmt der Anbieter von PSS eine zunehmende Verantwortung für den kompletten Produktlebenszyklus (Mont 2002). Die Herstellung von PSS verlangt vom Anbieter eine stärkere Konzentration auf die Nutzungsphase als im Falle von herkömmlichen Produkten oder Dienstleistungen.

Allerdings können PSS ebenso Nachteile sowohl für den Anbieter als auch für den Kunden mit sich bringen (Halen et al. 2005, 27ff.). Das Konzept von PSS ist noch ziemlich neu, komplex, nicht ausreichend erforscht und kann Risiken sowohl für den Anbieter als auch für den

Nachfrager bergen. Das Konzept kann zur Steigerung der Verwaltungskosten führen und verlangt neue Kompetenzen des Anbieters sowie die Bereitschaft des Kunden, den Anbieter mit dem eigenen Geschäft vertraut zu machen.

Sturm und Bading (2008) definieren die direkt beeinflussbaren und die nicht direkt beeinflussbaren Erfolgsfaktoren im Lösungsgeschäft.

Nicht direkt beeinflussbare Erfolgsfaktoren:

1. *Marktpotenzial*: Es muss einen entsprechenden Markt für PSS gefunden werden.
2. *Wettbewerbsvorteile*: Durch das Anbieten von PSS soll die Möglichkeit gegeben sein, sich von Wettbewerbern zu differenzieren.

Direkt beeinflussbare Erfolgsfaktoren:

3. *Kenntnis der Kunden*: Nur wenn der Anbieter den Kunden und damit das Kundengeschäft kennt, ist es möglich kundenindividuelle Lösungen zu entwickeln und sie in das Geschäft des Kunden zu integrieren. Die Lösungen sind in einer Zusammenarbeit mit dem Kunden zu entwickeln.
4. *Flexibles Leistungsangebot*: Das Lösungsangebot soll modular und flexibel sein, um immer neuen kundenindividuellen Leistungsbündeln zusammengefügt zu werden.
5. *Integrierte Lösungsentwicklung*: Die Entwicklung von Sach- und Dienstleistungskomponenten soll integriert und aufeinander abgestimmt werden, um ein Mehrwert für den Kunden dadurch erzielt wird.
6. *Lösungsadäquate Organisation und Mitarbeiter*: Die Mitarbeiter, die am Lösungsgeschäft mitwirken, sollen die notwendige Lösungskompetenz aufweisen.

Als Fazit lässt sich feststellen, dass PSS sowohl Vor- als auch Nachteile für den Anbieter und für den Kunden bergen und deswegen eine Bereitschaft aller Beteiligten voraussetzen, ein neues Geschäft zu gestalten.

3.2.2 Beispiele für Product Service Systems

Anhand von einigen Beispielen soll das Konzept von Product Service Systems verdeutlicht werden. Die Beispiele sollen die Vor- und Nachteile von PSS für den Lösungsanbieter und für den Kunden erläutern.

Beispiel 1: Sterilisationskomplettlösungen für Kliniken

Ein ehemals reines Medizintechnik-Unternehmen (Meditec) bietet neuerdings eine OP-Besteck-Sterilisationskomplettlösung für Kliniken auf einer nutzungsabhängigen Abrechnungsbasis an (Fähling et al. 2010). Die Gesamtlösung erlaubt dem Kunden, auf seine Bedarfe abgestimmt die OP's mit gereinigtem Besteck genau passend zu seiner OP-Belegungsplanung in seinem Kliniksystem vom Anbieter versorgt zu bekommen und zur Variabilisierung von Fixkosten rein nutzungsabhängig zu bezahlen (Fähling et al. 2010). Die Gesamtlösung hierfür umfasst eine entsprechende Software-Lösung, die in das Krankenhausinformationssystem integriert wird, eine auf die „on demand“ Bedarfe des Kunden ausgerichtete Service-Organisation sowie auf den Betrieb der Sterilisationslösung über mehrere Kunden hinweg ausgerichtete Sterilisationsanlagen beim Dienstleister. Alle diese Komponenten werden in einem integrierten Bündel angeboten und sind einzeln kaum nutzbar. Der Kunde profitiert vom Angebot einer solchen Lösung, indem er die Sterilisationsaktivitäten, die er davor selbst durchgeführt hat, an den Anbieter abgibt. Der Anbieter soll aber seine Infrastruktur entsprechend umstellen, damit seine Kosten durch den Verkauf der Lösung gedeckt wer-

den (Fähling et al. 2010). Ebenso muss er einen guten Zugang zum Kunden haben, um die Lösung „on demand“ bereitzustellen (Abbildung 3-13).

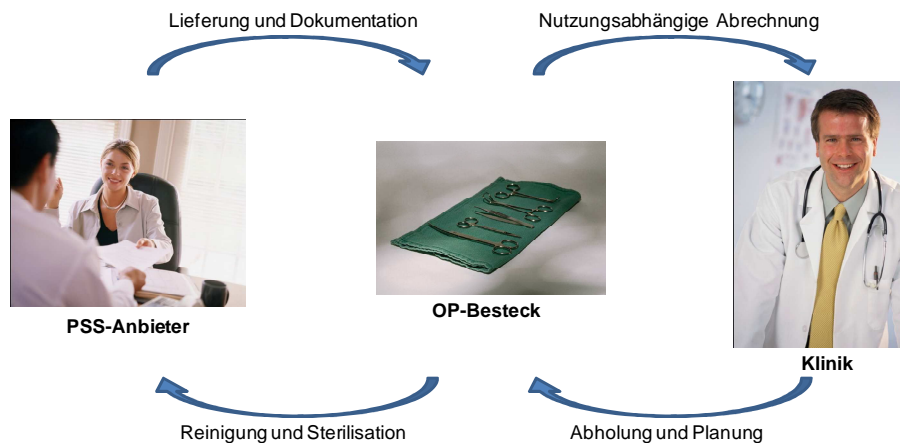


Abbildung 3-13: Sterilisationskomplettlösungen für Kliniken

Quelle: Eigene Darstellung (In Anlehnung an (Fähling et al. 2010))

Beispiel 2: Document management solution

Während einer traditionellen Nutzung werden Kopiergeräte durch den Kunden vom Anbieter gekauft. Der Anbieter kann im Sinne produktbegleitender Dienstleistungen Wartung, Installation oder Reparatur anbieten. Die Entsorgung, die Bereitstellung des Verbrauchsmaterials sowie die Überwachung sind vom Kunden zu erledigen. Das PSS „document management solution“ (Baines et al. 2007) bietet dem Kunden die Möglichkeit, den Kopierer zu nutzen, wobei alle anderen Aktivitäten, wie Entsorgung, Wartung, Reparatur, Installation, Aktualisierung der Technologien, Überwachung und Bereitstellung der Verbrauchsmaterialien ausschließlich beim Anbieter liegen (Abbildung 3-14).

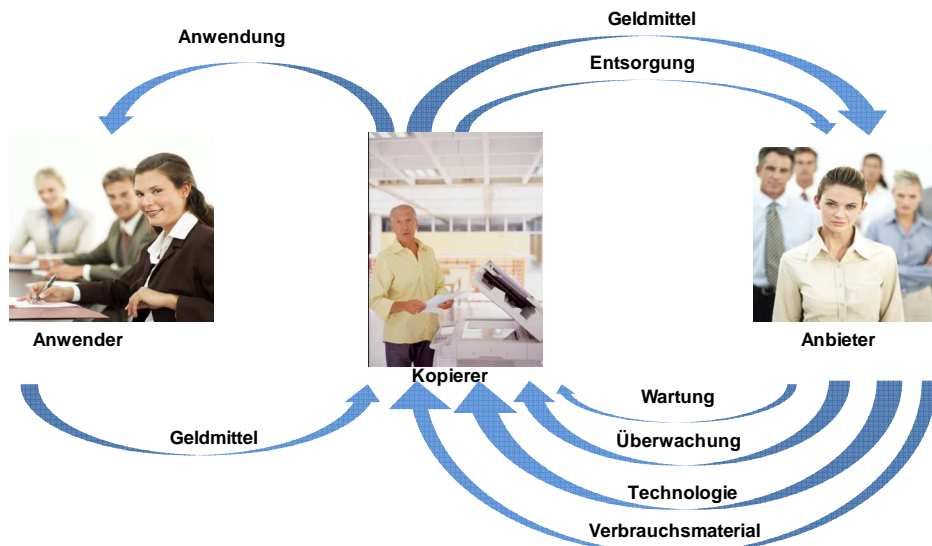


Abbildung 3-14: Document management solution

Quelle: In Anlehnung an (Baines et al. 2007)

Somit kauft der Kunde nicht das Gerät selbst, sondern die Funktionen dieses Gerätes, d. h. das Kopieren. Der Kunde kann sich dadurch besser auf sein eigentliches Geschäft konzentrieren und der Anbieter steigert dadurch die Kundenbindung. Wie im ersten Beispiel soll der Anbie-

ter seine Infrastruktur auf die Bedürfnisse des Kunden in Bezug auf die Lösung anpassen (Baines et al. 2007).

Beispiel 3: Personal Health Manager als IT-gestütztes, gerätebasiertes und personalisiertes Gesundheitsförderungsprogramm

Im Rahmen des vom BMBF (Bundesministerium für Bildung und Forschung) geförderten Projektes SPRINT² wurde ein IT-gestütztes Bewegungsprogramm - *Personal Health Manager (PHM)* - entwickelt, das ein zeit- und ortsunabhängiges Bewegungstraining und eine skalierbare Infrastruktur zur Bereitstellung der Betreuung des Sportlers durch den Fitnesstrainer ermöglicht (Esch et al. 2010b; Esch et al. 2010a; Knebel et al. 2009; Knebel et al. 2007). Eine direkte Interaktion zwischen dem Fitnesstrainer und dem Sportler ist meistens zeitaufwendig und kostenintensiv. Es werden Lösungen benötigt, die in der Lage wären, dem Sportler eine auf seine Bedürfnisse abgestimmte Betreuung zu geben. So eine Lösung stellt der PHM dar. Ein Bestandteil der Lösung ist ein Smartphone mit der PHM-Software, die eine kontinuierliche und auf die Bedürfnisse des Sportlers angepasste Rückkopplung mit dem Plan zu den Sportaktivitäten bereitstellt. Der Sportler soll im Gegenzug Informationen über seine Fortschritte, Ernährungsgewohnheiten oder Sensor-Daten von seinen Sportgeräten an das System übermitteln. Der Teilnehmer benutzt die PHM-Software als Bewegungsplan und -tagebuch. Die darauf folgende Rückkopplung bekommt der Sportler sowohl vom System, das die von ihm eingegebenen Daten analysiert, auswertet und Vorschläge zu seinen weiteren sportlichen Aktivitäten automatisch macht, als auch vom Fitnesstrainer selbst, der ebenso einen Zugang zu den Daten des Sportlers hat. Somit bleibt die persönliche Beziehung zwischen dem Trainer und dem Sportler (Teilnehmer) erhalten, denn sowohl der Trainer als auch der Teilnehmer benutzen die PHM-Software. Dem Trainer dient die Software als Entscheidungsgrundlage. Die PHM-Software ermöglicht dem Trainer die Eingaben der Teilnehmer zu überprüfen und dem Teilnehmer daraufhin Anweisungen in Bezug auf seine Sportaktivitäten zu geben (Abbildung 3-15). Demzufolge erhält der Kunde eine Kombination aus persönlichen, teilautomatisierten und vollautomatisierten Elementen zur Unterstützung für seine sportlichen Aktivitäten.

² <http://www.projekt-sprint.de>



Abbildung 3-15: Personal Health Manager als IT-gestütztes, gerätebasiertes und personalisiertes Gesundheitsförderungsprogramm

Quelle: In Anlehnung an (Esch et al. 2010b; Esch et al. 2010a; Knebel et al. 2009; Knebel et al. 2007)

3.3 Lebenszyklenperspektive von Product Service Systems

In der Praxis fehlen integrierte Vorgehen zur Entwicklung von PSS weitgehend (Leimeister/Glauner 2008; Berkovich et al. 2009a; Berkovich et al. 2009c; Sturm et al. 2007). Die meisten integrierten Vorgehensmodelle in der Literatur sind komplex und sehr abstrakt gehalten. Darüber hinaus fehlt diesen oft eine Evaluation und die nötige Standardisierung und Normierung, um in der Praxis eingesetzt zu werden (Langer et al. 2010; Gräßle/Dollmann 2010; Leimeister/Glauner 2008; Berkovich et al. 2009a). Ebenso fehlen integrierte und domänenübergreifende Vorgehen für das Requirements Engineering für PSS, die in der Lage wären, ein gemeinsames Verständnis über das zu lösende Problem bei allen an der Entwicklung von PSS beteiligten Domänen zu entwickeln und somit eine integrierte Entwicklung von PSS zu unterstützen (Spath/Demuß 2003; Berkovich 2010; Berkovich et al. 2010b; Berkovich et al. 2011a).

Der erste Schritt zur Konzeption eines integrierten und domänenübergreifenden Ansatzes zum Requirements Engineering für PSS ist die Betrachtung der Produktlebenszyklusperspektive von PSS, um die Zusammenhänge zwischen den einzelnen Phasen des Lebenszykluses und dem Requirements Engineering zu verstehen und darauf basierend die Rolle des Requirements Engineering und damit den Wertbeitrag, den das Requirements Engineering für die Entwicklung von PSS leistet, zu definieren. Im Rahmen der Produktlebenszyklusperspektive werden die Lebensdauer und der -verlauf eines Produktes von der Idee über Entwicklung und Konstruktion, Produktion sowie Vertrieb bis zur Außerbetriebnahme betrachtet (Arnold et al. 2005, 13).

In der Literatur wurden zwei Blickwinkel auf den Lebenszyklus eines PSS identifiziert. Zum einen kann der Lebenszyklus eines PSS aus der produkt- und prozesszentrischen Sicht, und zum anderen aus der Sicht des Kunden und des Lösungsanbieters betrachtet werden.

Tuli et al. (2007) betrachten den Lebenszyklus einer Lösung aus der produkt- und prozesszentrischen Perspektiven. Im Zentrum der produktzentrischen Sicht steht die Erbringung einer auf die Bedürfnisse des Kunden angepassten Kombination aus Sach- und Dienstleistungen, während im Zentrum der prozesszentrischen Sicht der Erbringungsprozess selbst und somit seine Phasen stehen. Abbildung 3-16 zeigt die verschiedenen Sichten auf den PSS-Lebenszyklus. Der Prozess zur Erbringung eines PSS besteht aus vier Schritten: Erhebung der Anforderungen, Individualisierung und Integration, Implementierung und After-Sales Services. In der ersten Phase der Anforderungserhebung finden die Identifikation von Anforderungsquellen sowie die Anforderungsermittlung statt. In der anschließenden Phase der Individualisierung und Integration sind die materiellen und immateriellen Komponenten des PSS basierend auf den Anforderungen der Kunden zu identifizieren, bestehende Komponenten sind auszuwählen und neue Komponenten sind zu entwickeln. In der Phase der Implementierung wird das PSS an den Kunden gebracht und in die Wertschöpfungsprozesse des Kunden integriert. Anschließend wird das PSS in Betrieb genommen, wobei der Anbieter entsprechende After-Sales-Dienstleistungen, wie zum Beispiel Wartung, Reparatur oder Schulung anbietet.

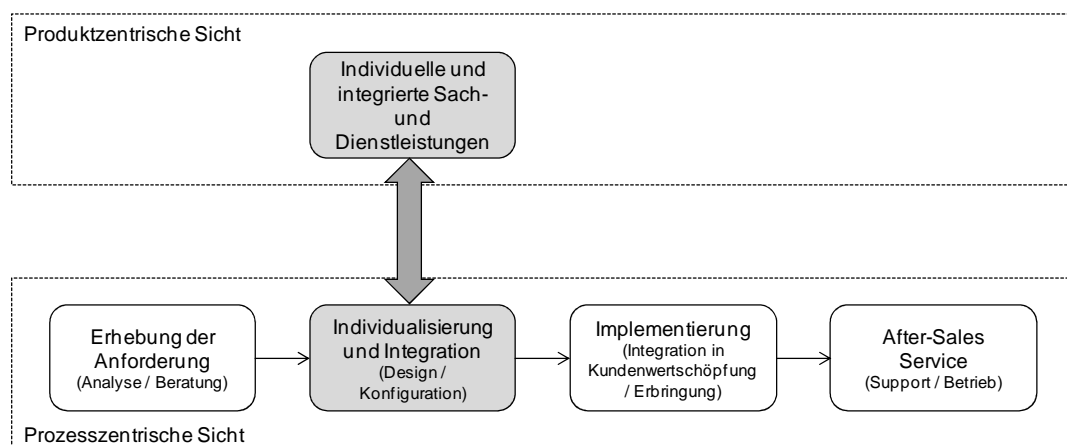


Abbildung 3-16: Problemlösungsbezug von PSS

Quelle: In Anlehnung an (Tuli et al. 2007, 5) zitiert in (Burianek/Reichwald 2009, 19)

Mannweiler et al. (2010) und Schweitzer (2010) betrachten den Lebenszyklus eines PSS aus der Perspektive des Kunden und des Lösungsanbieters. Aus der Anbieterperspektive beginnt der Lebenszyklus eines PSS mit der PSS-Planung und -Entwicklung im Rahmen der PSS-Gestaltung. Während der Planung sollen die Kunden- und Anbieterziele definiert werden sowie die Sachleistung, die dem Kunden den gewünschten Grundnutzen bereitstellt, identifiziert werden. Während der Entwicklung werden die einzelnen Komponenten von PSS entworfen, in der Phase der Produktion umgesetzt und dem Kunden zur Verfügung gestellt. Der Lebenszyklus endet in der Regel mit der Wieder- oder Weiterverwendung der Sachleistung (End-of-Life).

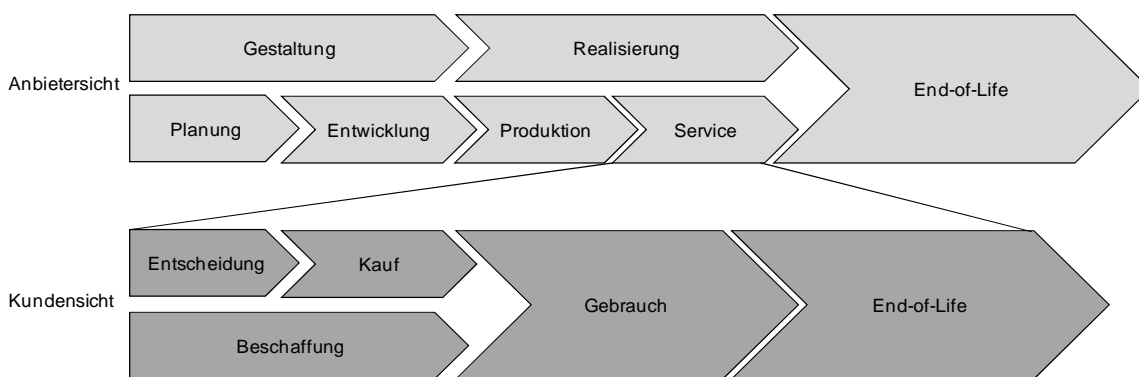


Abbildung 3-17: Lebenszyklus von PSS

Quelle: In Anlehnung an (Schweitzer 2010; Mannweiler et al. 2010)

Aus Sicht des Kunden besteht der PSS-Lebenszyklus aus den Phasen Beschaffung, Gebrauch und End-of-Life. Dabei tritt das PSS erst in die Sicht des Kunden, wenn die Entwicklung des Leistungsbündels abgeschlossen ist, also in der Phase der Realisierung aus der Sicht des Anbieters. In der Beschaffungsphase wird der Bedarf an die Lösung bestimmt und ein Vergleich von verschiedenen Lösungsangeboten durchgeführt. Anschließend finden die Vorführung und der Kauf statt. In der Gebrauchsphase finden die Inbetriebnahme, Einsatzplanung, Wartungs- und Instandhaltungsarbeiten statt. In den abschließenden Phase „End-of-Life“ werden neue Bedingungen für den Einsatz der Lösung gesucht. Es kann ein Neukauf oder Verschrottung erfolgen (Mannweiler et al. 2010).

Die betrachteten Lebenszykluskonzepte lassen sich somit in das erweiterte Lebenszyklusmodell für PSS von Blinn et al. (2008) einordnen und dadurch vereinheitlichen. Der Lebenszyklus von PSS besteht aus drei Hauptphasen. Während der ersten Phase (*Produktentstehung*) werden die Anforderungsquellen identifiziert, die Anforderungen an das PSS ermittelt sowie materielle und immaterielle Komponenten für das PSS identifiziert und, falls erforderlich, kombiniert oder neu entwickelt. Diese Phase umfasst die von Tuli et al. (2007) vorgeschlagenen Phasen Erhebung der Anforderungen und Individualisierung und Integration. Schweitzer (2010) und Mannweiler et al. (2010) bezeichnen diese Phase als Gestaltung. Anschließend werden die einzelnen Komponenten zusammengeführt und in die Wertschöpfungsprozesse des Kunden integriert (*Produktvermarktung*). Diese Aktivitäten werden im Lebenszyklusmodell von Tuli et al. (2007) in der Phase der Implementierung und im Modell von Schweitzer (2010) und Mannweiler et al. (2010) in der Phase der Realisierung durchgeführt. In der letzten Phase liegt der Fokus auf der Instandhaltung und der Entsorgung des Produktes am Ende des Produktlebenszyklus (*After-Sales-Phase*). Genau diese Phase stellt das größte Wertschöpfungspotenzial bereit. Typische Dienstleistungen in dieser Phase sind Wartung, Inspektion und Gewährleistung. Tuli et al. (2007) bezeichnen diese Phase als After-Sales-Service und Schweitzer (2010) und Mannweiler et al. (2010) als End-of-Life. Zuletzt soll das PSS vorschriftsgerecht entsorgt werden.

Für weitere integrierte Vorgehensmodelle zur Entwicklung von PSS sei auf Kapitel 4.4.2.4 verwiesen.

3.3.1 Entwicklungsprozess von Product Service Systems

Nicht immer sind alle Leistungsbündel zur Kombination der Lösung vorhanden. In diesem Fall spielt der Entwicklungsprozess eine essentielle Rolle im Lebenszyklus von PSS, der mit

der abstrakten Idee der Lösung für kundenspezifische Probleme startet und mit der Entwicklung domänenspezifischer Komponenten von PSS, wie Hardware, Software und Dienstleistungen, bis hin zu ihrer Zusammenführung endet.

Die Anforderungen an materielle und immaterielle Komponenten des PSS weisen starke Abhängigkeiten auf und sind nur schwer voneinander trennbar. Trotz dieser Eigenschaften der Anforderungen werden die Komponenten von PSS von den einzelnen Domänen getrennt entwickelt. Somit erfordert die Entwicklung der Komponenten eine ständige domänenübergreifende Abstimmung. Die initialen Anforderungen an das PSS sollen auf die an der Entwicklung von PSS beteiligten Domänen entsprechend aufgeteilt werden, damit die Domänen ihre Entwicklungsaktivitäten parallel und integriert durchführen können (Spath/Demuß 2003).

Spath und Demuß (2003, 494) unterscheiden drei Hauptphasen im Entwicklungsprozess von PSS: hybride Aufgabenklärung, hybride Produktkonzeption und entwicklungsspezifischer Komponentenentwurf (Abbildung 3-18).

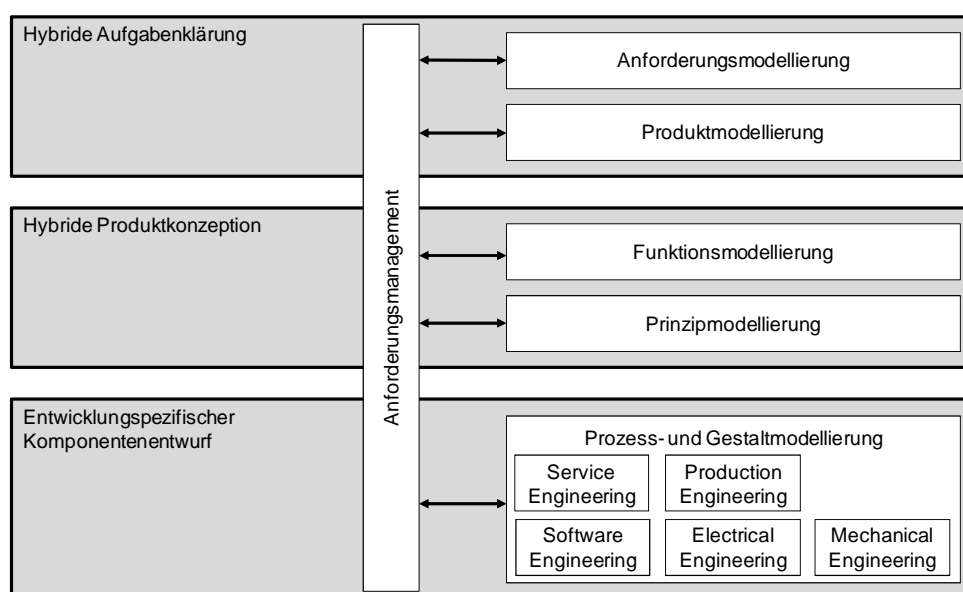


Abbildung 3-18: Entwicklungsprozess von PSS

Quelle: In Anlehnung an (Spath/Demuß 2003, 494)

Die erste Phase der *hybriden Aufgabenklärung* besteht aus zwei Schritten. Im ersten Schritt der Anforderungsmodellierung werden die Anforderungen an das zu entwickelnde PSS gesammelt und analysiert. Während der Phase der Produktmodellierung wird zuerst der Teil der Umgebung des PSS definiert, der für das Verständnis von Anforderungen an das PSS relevant ist. Anschließend werden basierend auf den ermittelten Anforderungen die materiellen und immateriellen Komponenten des PSS identifiziert und modular strukturiert, um eine effiziente Konfiguration der Lösung zu erreichen.

In der zweiten Phase der *hybriden Produktkonzeption* findet die Funktionsmodellierung in den jeweiligen Domänen statt. Somit werden Funktionsstrukturen sowohl für die technischen Produkte der Lösung (Sachleistungen) als auch für die Dienstleistungen aufgebaut. Zum Aufbau des Dienstleistungserbringungsprozesses werden Ressourcen benötigt, die im Rahmen der Funktionsmodellierung den Dienstleistungen zuzuordnen sind. Im Rahmen der anschließenden Prinzipmodellierung werden physikalische und chemische Effekte oder Algorithmen den technischen Produkten zugeordnet. Für die Dienstleistungen können erste konzeptionelle Lö-

sungen gesucht werden. Anschließend werden die einzelnen domänenspezifischen Komponenten des PSS identifiziert.

In der letzten Phase des *entwicklungsspezifischen Komponentenentwurfs* werden die einzelnen physikalischen domänenspezifischen Komponenten als auch die Dienstleistungen von PSS in einem Entwurf beschrieben und anschließend von den dafür zuständigen Domänen parallel und abgestimmt entwickelt (Spath/Demuß 2003).

Die beschriebenen Prozesse gehören laut Blinn et al. (2008) in die Phase der Produktentstehung, werden aber weder von Blinn et al. (2008), noch von den in Kapitel 3.3 betrachteten Lebenszyklusmodellen von Tuli et al. (2007) und Schweitzer (2010) und Mannweiler et al. (2010) detailliert behandelt.

3.3.2 Zyklische Wechselwirkungen im Lebenszyklus von Product Service Systems

Der Lebenszyklus von PSS ist von hoher Komplexität geprägt und wird von verschiedenen Faktoren, wie Verfügbarkeit neuer Technologien, unterschiedliche Lebenszyklen der einzelnen Bestandteile der Lösung oder Änderungen von Kundenanforderungen, beeinflusst. Die meisten dieser Faktoren sind temporär und oft repetitiv. Somit kann die Verfügbarkeit neuer Technologien zu einem Austausch von Komponenten von PSS führen, während die Änderungen von Kundenanforderungen zu Änderungen an der Gesamtlösung führen können (Langer/Lindemann 2009; Berkovich et al. 2011e).

Sowohl die Einflussfaktoren selbst als auch die durch sie induzierten Aktivitäten während der Entwicklung und Nutzung von PSS haben einen zyklischen Charakter (Langer/Lindemann 2009; Zaeh et al. 2009).

Ein *Zyklus* ist ein sich wiederholendes temporäres Verhalten eines Objektes (Langer/Lindemann 2009; Langer et al. 2008). Demzufolge können sich beliebige Ergebnisse, wie Teilprozesse, Artefakte, Vorgehen oder Entwicklungen, oder eine Aufeinanderfolge unterschiedlicher Ereignisse innerhalb eines Ablaufes wiederholen. Das Ergebnis solcher zyklischer Wechselwirkungen können höhere Komplexität, steigende Kosten durch mehrere Iterationen von Aktivitäten des Entwicklungsprozesses, verspätete Bereitstellung der Lösung an den Kunden und als Folge Beeinträchtigung der Kundenzufriedenheit und Umsatzverluste sein (Berkovich et al. 2011e). Deswegen ist es wichtig, die Zyklen und die Ursachen, die zum Entstehen von Zyklen führen, zu identifizieren und zu analysieren. Darauf basierend sollen Maßnahmen zur Kontrolle und eventuell Minimierung von Zyklen entworfen werden (Berkovich et al. 2011e).

Ein *Zyklus im Requirements Engineering* ist die Wiederholung von Requirements Engineering Aktivitäten, bspw. Anforderungsermittlung oder Anforderungsanalyse, oder von beliebigen Ergebnissen, zu denen Teilprozesse, Artefakte oder Vorgehen gehören, wie bspw. Aktualisierung von Lasten- und Pflichtenheften (Berkovich et al. 2011e).

Ein Beispiel soll verdeutlichen, welche Rollen Zyklen im Lebenszyklus von PSS spielen. Das Beispiel beschreibt eine Fallstudie aus dem Jahr 2007 zur Untersuchung von PSS in der Praxis (vgl. (Berkovich et al. 2011e)). Das Unternehmen „*SolutionAndCommodity*“³ bietet in seinem Kerngeschäft Lösungen rund um industrielle Gase an. Das Angebot reicht von Standardanlagen zur Erzeugung von Gasen über Flüssiggas in Flaschen bis hin zur Dienstleistung

³anonymisiert

der Gasversorgung. Das Unternehmen hat ca. 53.000 Mitarbeiter und einen Umsatz von ca. 12 Mrd. Euro (Angaben von 2007). Der Fallstudie liegen 4 Interviews mit dem Entwicklungsleiter der Abteilung chemische Prozessentwicklung und einem Mitarbeiter, der für das Entwickeln von Betreibermodellen im Gas-Geschäft tätig war, zu Grunde. Organisatorisch ist das Unternehmen in zwei Bereiche aufgeteilt, die soweit möglich unabhängig voneinander agieren:

- (1) Die Engineering-Sparte beschäftigt sich mit dem Bauen von Anlagen zur Gewinnung und Verarbeitung von Gas.
- (2) Die Gas-Sparte beschäftigt sich mit der Versorgung von Kunden mit Gas.

Die Engineering-Sparte entwickelt sowohl kundenindividuelle Anlagen, als auch fertige Anlagen aus Serienproduktion. Die Gas-Sparte bietet ihren Kunden individuelle Lösungen zur Lieferung von Gas zu bestimmten Bedingungen an. Die angebotene Lösung umfasst das Produkt Gas sowie dessen Lieferung unter Einhaltung der geforderten Rahmenbedingungen. Zu solchen Rahmenbedingungen gehören bspw. die Verfügbarkeit, der Druck oder die Reinheit des Gases. Somit kann die Produktionsanlage direkt beim Kunden aufgebaut werden oder eine Rohrleitung kann zu einer benachbarten Anlage gebaut werden. Eine Integration der Lösung in die Wertschöpfungsprozesse des Kunden findet bspw. durch die Einbindung von Abfallprodukten aus dem Prozess des Kunden in den Produktionsprozess des zu liefernden Gases statt. In der Nutzungsphase werden von *SolutionAndCommodity* entsprechende Dienstleistungen, wie Wartung, Installation oder Schulung angeboten. Bei der Entwicklung der Lösung für die Kunden sind zyklische Wechselwirkungen aufgetreten. Während die Engineering-Sparte sich mit dem Bau von Anlagen, Bereitstellung von Tanks und Pipelines (Hardware-Komponenten) beschäftigt, soll die Software Steuerungs- und Überwachungsfunktionen, wie Regelung des Gasdrucks und der Durchflussmenge pro Einheit übernehmen und die Menge des Gasverbrauchs für die spätere Abrechnung protokollieren. Änderungen an der Rohrleitung führen zu Änderungen an der Software zur Regelung der Menge pro Einheit. Das kann wiederum zu Änderungen an weiteren Hardware-Komponenten, wie zum Beispiel den Sensoren in der Gas-Anlage führen (Abbildung 3-19). Als Folge auf die Iterationen gab es Probleme bei der rechtzeitigen Lieferung der Lösung an den Kunden und als Folge wachsende Kundenunzufriedenheit und daher nötige Preisreduktionen seitens des Anbieters.

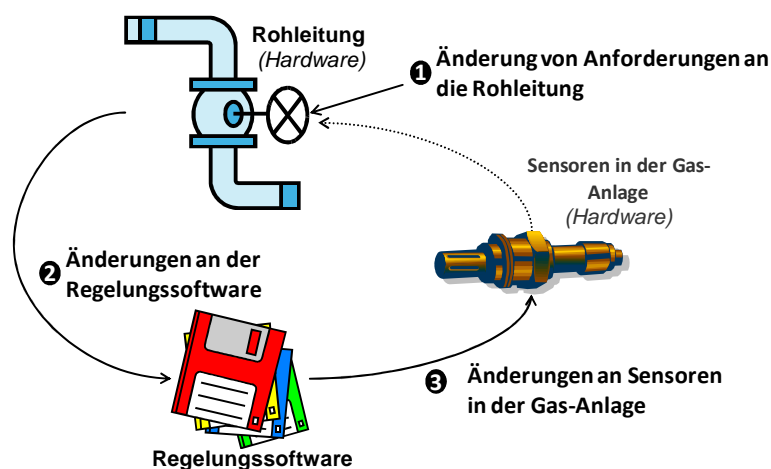


Abbildung 3-19: Zyklen bei der Gas-Versorgungslösung - Änderungen von Kundenanforderungen

Quelle: (Berkovich et al. 2011e)

Ein weiterer Zyklus ließ sich während der Entwicklung der Lösung beobachten. Die Entwicklung der Hardware- und Softwarekomponenten fand unabhängig voneinander und mit sehr wenigen Abstimmungen zwischen den beteiligten Domänen statt. Nachdem die Entwicklung der einzelnen Komponenten der Lösung abgeschlossen war, fanden mehrere Iterationen zwischen den Entwickler-Teams zur Klärung der Zusammenführung der Komponenten in eine Lösung statt. Aufgrund der fehlenden Zusammenarbeit zu Beginn der Entwicklung mussten Hardware- und Softwarekomponenten mehrmals geändert und auf die Bedürfnisse der anderen Komponenten angepasst werden. Diese Zyklen wären zu vermeiden gewesen, wenn die Entwickler-Teams von Anfang an miteinander kooperiert und dadurch die Lösung in abgestimmter und integrierter Weise entwickelt hätten.

Ein weiterer Zyklus konnte während der Anforderungsanalyse beobachtet werden. Die Kunden verlangten eine termingerechte Gaslieferung (Abbildung 3-20). Daraufhin wurde diese Anforderung von den Anforderungsanalysten durch die Anforderung „Die Lieferung soll mindestens 24 Stunden im Voraus erfolgen“ konkretisiert. Diese Anforderung wurde an die Entwickler weitergeleitet. Allerdings wurde diese Anforderung von den Entwicklern nicht verstanden, da die Angaben zu der Größe des Zwischenspeichers gefehlt haben.

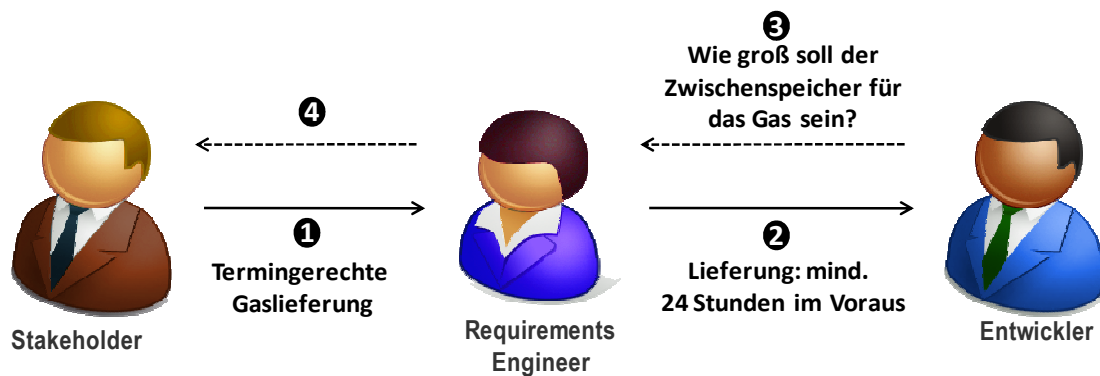


Abbildung 3-20: Zyklen bei der Gas-Versorgungslösung - Missverständnisse beim Klären von Anforderungen

Quelle: (Berkovich et al. 2011e)

Aus den Beispielen wird ersichtlich, dass die Bereitstellung von PSS durch zyklische Wechselwirkungen geprägt ist. Als Folge davon werden der Entwicklungsprozess von ständigen Iterationen und Rekursionen und das PSS selbst von ständigen Anpassungen beeinträchtigt (Langer et al. 2008). Die Ursachen für die Zyklen können sowohl im Entwicklungsprozess von PSS selbst als auch außerhalb des Entwicklungsprozesses entstehen, wie in der Unternehmensumgebung der Lösung, oder außerhalb auch auf dem Markt und in der Gesellschaft (Langer/Lindemann 2009). Daher ist es besonders wichtig, die Ursachen, die die Zyklen hervorrufen, zu erkennen und die Maßnahmen einzuleiten, um die Zyklen zu beherrschen und somit zu reduzieren.

3.4 Zusammenfassung

In diesem Kapitel wurden die begrifflichen Grundlagen zu PSS geklärt, indem die Begriffe „Produkt“, „Dienstleistung“, „Sachleistung“ und „Product Service System“ voneinander abgegrenzt wurden. Unter dem Namen *Hybride Produkte* oder *Product Service Systems* sind Lösungen bekannt, die integrierte Bündel von Sach- und Dienstleistungen darstellen, bei denen die einzelnen Komponenten für den Kunden nicht erkennbar sind. PSS sind individuelle

Lösungen für Kundenprobleme, die von Unternehmen angeboten werden, um die Differenzierung von den Wettbewerbern zu erhöhen und sich dadurch Wettbewerbsvorteile zu verschaffen, Kundenbindung zu stärken und eigene Innovationspotenziale auszunutzen. Dabei stehen bei der Bereitstellung von PSS die Kundenanforderungen und -wünsche im Vordergrund.

Empirische Studien belegen, dass die Unternehmen die entscheidende Rolle von PSS für den Gesamterfolg des Unternehmens erkannt haben. Dennoch sind viele Unternehmen mit der integrierten Entwicklung von Sach- und Dienstleistungskomponenten von PSS, die von unterschiedlichen Domänen erbracht werden, in ihren Unternehmen nicht zufrieden. Es fehlen entsprechende Vorgehensmodelle, die es ermöglichen die domänenspezifischen Komponenten von PSS integriert und abgestimmt zu entwickeln.

Um die Problemstellung eines integrierten Requirements Engineering Ansatzes anzugehen, ist es wichtig, den Lebenszyklus von PSS und insbesondere den Entwicklungsprozess, dessen Teil Requirements Engineering ist, zu verstehen, mit dem Ziel die Aktivitäten des Requirements Engineering für die Entwicklung von PSS zu identifizieren. Der Lebenszyklus eines PSS umfasst vier Phasen, zu denen die Anforderungserhebung, die Identifizierung und integrierte Entwicklung von materiellen und immateriellen Komponenten des PSS zur Lösung eines kundenindividuellen Problems, die Integration des PSS in den Wertschöpfungsprozess des Kunden und das Anbieten von After-Sales Dienstleistungen gehören.

Der Produktlebenszyklus von PSS ist sehr komplex und wird von verschiedenen Faktoren, wie Verfügbarkeit neuer Technologien oder Änderungen von Kundenanforderungen beeinflusst, welche zu zyklischen Wechselwirkungen führen. Somit können sich beliebige Ergebnisse oder eine Aufeinanderfolge unterschiedlicher Ereignisse innerhalb des Produktlebenszyklus wiederholen und mit sich höhere Komplexität, steigende Kosten und verspätete Bereitstellung der Lösung bringen. Es ist deswegen essentiell für die Entwicklung von PSS die Ursachen für die Zyklen zu erkennen und die Maßnahmen einzuleiten, um die Zyklen zu beherrschen.

4 Requirements Engineering im Kontext von Product Service Systems

Zu Beginn von Kapitel 4 wird die Bedeutung des Requirements Engineering beschrieben. Anschließend wird der Begriff „Anforderung“ eingeführt und es wird erklärt, welche Ziele Requirements Engineering und Management verfolgen und welche Aktivitäten zum Requirements Engineering gehören. Darauf aufbauend werden die Rolle und der Wertbeitrag des Requirements Engineering zur Entwicklung von PSS untersucht. In den darauffolgenden Unterkapiteln wird der Stand der Wissenschaft sowie Stand der Praxis zum Requirements Engineering für PSS analysiert, mit dem Ziel Probleme und Herausforderungen des Requirements Engineering für PSS zu identifizieren. Die aus der Analyse gewonnenen Erkenntnisse zur konzeptuellen und empirischen Forschungslücken sollen dadurch aufgegriffen werden, indem die Anforderungen an einen Requirements Engineering Ansatz für Product Service Systems, die in Kapitel 4.7 aufgelistet sind, aufgestellt und anschließend in einem Ansatz umgesetzt werden.

4.1 Bedeutung des Requirements Engineering

Dem Erfolg eines Produktes oder einer Dienstleistung liegt eine erfolgreiche Umsetzung der Wünsche und Vorstellungen von Kunden und Stakeholdern zu Grunde (Procaccino et al. 2005; Nuseibeh/Easterbrook 2000; Lindemann 2006, 7-9; Beiter et al. 2006; Ramaswamy 1996, 26f.). Dafür ist es notwendig, die Kunden- und Stakeholderanforderungen zu verstehen, was der Einsatz von Requirements Engineering unverzichtbar macht. *Requirements Engineering* ist ein wichtiger aber gleichzeitig kritischer Bestandteil des Entwicklungsprozesses (Jung 2006, 2ff.; Damian/Chisan 2006; Brooks 1987). Mangelhaftes Requirements Engineering ist eine der häufigen Ursachen für das Scheitern von Entwicklungsprojekten (Aurum/Wohlin 2005b, 1-3; Ahrens 2000, 12-23), denn Requirements Engineering beeinflusst den ganzen Entwicklungsprozess und kann aufgrund einer mangelhaften Ausführung zu steigenden Kosten und Aufwand führen (Abramovici/Schulte 2007; Finkelstein/Dowell 1996; Boehm/Basili 2001). So nennt Pohl (2007, 11) folgende Gründe für eine immer steigende Bedeutung des Requirements Engineering:

1. Immer mehr Projekte scheitern aufgrund von mangelhaften Anforderungsspezifikationen.
2. Das Beseitigen von Anforderungsfehlern in den späteren Phasen der Entwicklung führt zu signifikanten Kostensteigerungen.
3. Die Unternehmen stehen vor der Herausforderung innovative, individuellere und komplexere softwareintensive Systeme schneller, mit höherer Qualität und zu immer geringeren Preisen auf den Markt zu bringen.
4. Die Bedeutung von softwareintensiven Systemen mit wachsenden Funktionsumfängen, engerem Integrationsbedarf mit anderen Systemen sowie differenzierter Nutzung steigt.

Mehrere Studien und Analysen von Projekten belegen, dass Requirements Engineering ein wichtiger Bestandteil des Entwicklungsprozesses ist und kontinuierlich im Laufe des gesamten Entwicklungsprojektes durchgeführt werden soll (Pohl 2007, 7ff.). Kamata und Tamai (2007) berichten, dass der Erfolg eines Projektes in engem Zusammenhang mit der Qualität des Requirements Engineering steht.

Eine gute Veranschaulichung des Projekterfolgs in der Softwareentwicklung liefert „The Chaos Report“ der Standish Group (1995). Es wird berichtet, dass im Jahr 1995 ca. 52,7% der Softwareentwicklungs-Projekte abgeschlossen wurden, aber das ursprünglich geplante Budget um bis zu 189% überstiegen. Die Untersuchungen von Projekterfolgsquoten in den nächsten Jahren (Standish Group (1999, 2002, 2004) ergaben keine nennenswerten Unterschiede der erfolgreichen und gescheiterten Projekte zu den Ergebnissen aus dem Jahr 1995. Im Jahre 2004 sind ca. 18% der Projekte gescheitert und ca. 29% der Projekte wurden erfolgreich abgeschlossen. Abbildung 4-1 zeigt eine Gegenüberstellung der Projekterfolgsquoten aus den Jahren 1994, 1996, 1998, 2000, 2004. Ca. 53% der Projekte wurden 1994 mit Ressourcenüberschreitungen oder Funktionseinschränkungen abgeschlossen. Im Jahre 2004 wurden ebenso ca. 53% der Projekte Ressourcenüberschreitungen oder Funktionseinschränkungen abgeschlossen. Zu ähnlichen Ergebnissen kommen auch weitere Studien (vgl. (Hall et al. 2002; Sommerville/Ransom 2005; Buschermöhle et al. 2006; Nikula et al. 2000)).

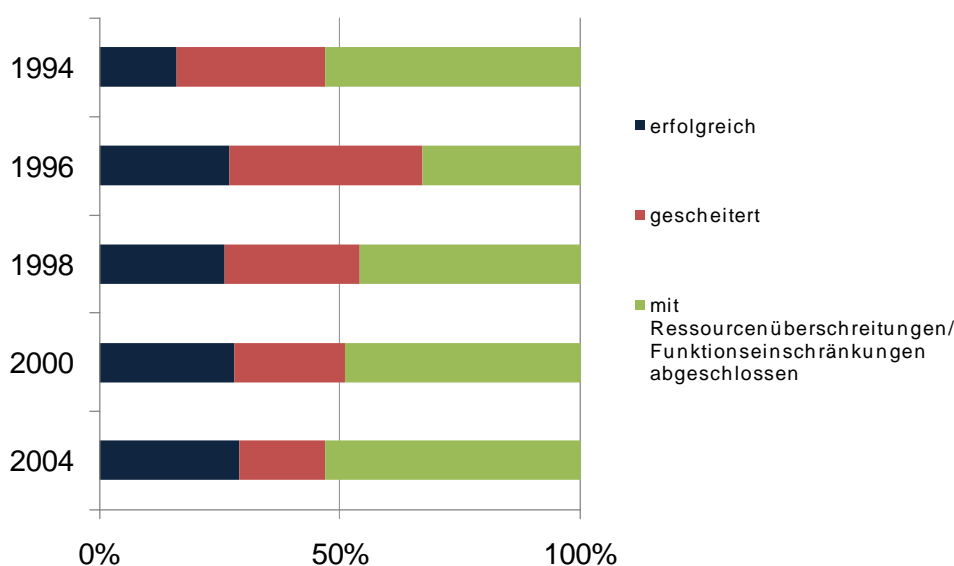


Abbildung 4-1: Projekterfolgsquoten aus den Standish Group (CHAOS-) Studien der Jahre 1994-2004

Quelle: (The Standish Group 1995, 1999, 2002, 2004) zitiert in (Pohl 2007, 8f.; Méndez Fernández et al. 2010)

Die häufigsten Ursachen für Ressourcenüberschreitungen oder Funktionseinschränkungen bei Projekten (Abbildung 4-2), die im Zusammenhang mit den Aktivitäten des Requirements Engineering stehen, waren mangelnde Einbeziehung der Benutzer (ca. 13%), unvollständige Anforderungen (ca. 12%), Änderungen von Anforderungen (ca. 12%), unrealistische Erwartungen (ca. 6%), unklare Ziele (ca. 5%) und unrealistischen Zielvorgaben (ca. 4%) (The Standish Group 1995) zitiert in (Pohl 2007, 8f.). Demzufolge lassen sich ca. 48% der Gründe auf ein mangelhaftes Requirements Engineering zurückzuführen.

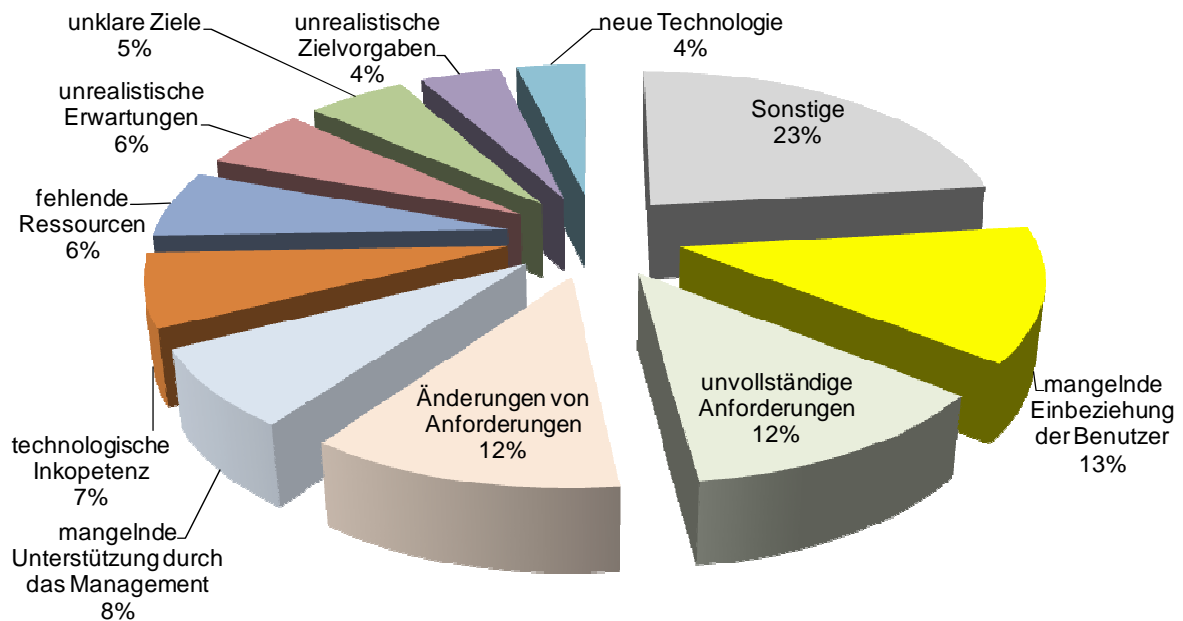


Abbildung 4-2: Gründe für Ressourcenüberschreitungen und/oder Funktionseinschränkungen bei Projekten

Quelle: (The Standish Group 1995) zitiert in (Pohl 2007, 8f.; Ebert 2005, 40; Méndez Fernández et al. 2010)

In einer Studie von Kujala et al. (2005) wurden 18 Experten, darunter Projektmanager, Produktmanager, Testmanager, Softwareentwickler, Zuständige für die Qualitätskontrolle sowie ein Geschäftsführer aus 13 Unternehmen zum Erfolg von Software- und Softwarerelevanten Projekten in Finnland befragt. Als sehr wichtige Faktoren für den Erfolg eines Projektes wurden mehr Zeit für die Durchführung des Requirements Engineering, eine systematische Sammlung von Anforderungen, Verbesserung der Anforderungsnachvollziehbarkeit, eine systematische Detaillierung von Anforderungen sowie Anforderungswerkzeuge genannt. Zu den Problemen im Requirements Engineering gehören ein Mangel an Ressourcen zur Durchführung der Anforderungsermittlung und -verwaltung, eine unzureichende Einbeziehung der Stakeholder in den Prozess des Requirements Engineering, eine mangelnde methodische Unterstützung der Anforderungsermittlung und -verwaltung sowie eine oft unklare Beschreibung von Anforderungen (Kujala et al. 2005).

Eine weitere Studie, die die Wichtigkeit des Requirements Engineering für den Erfolg von Entwicklungsprojekten betont, ist Damian und Chisan (2006). Ihr Ergebnis ist, dass ein guter Requirements Engineering Prozess in einem Unternehmen gute Auswirkungen auf die nachfolgenden Entwicklungsaktivitäten hat. So wurden positive Auswirkungen auf die Projektplanung, Change-Management, Testen, sowie der nötigen Nacharbeit nachgewiesen. Auch wurden die Fehler in Produkt reduziert und die Qualität erhöht. Die Studie von Procaccino und Vernerb (2006) nimmt einen anderen Blickwinkel ein. Diese Studie hat das Ziel herauszufinden, wann ein Projekt als Erfolg wahrgenommen wird. Das Ergebnis der Studie ist, dass Projektmanager ein Projekt als erfolgreich einstufen, falls es die Kundenanforderungen erfüllt. Durch diese Studie wird unterstrichen, dass die Kundenanforderungen von zentraler Wichtigkeit in Entwicklungsprojekten sind.

Zusammenfassend kann festgestellt werden, dass Requirements Engineering von herausragender Bedeutung für den Erfolg von Entwicklungsprojekten ist. Projektfehlschläge lassen

sich oft in Zusammenhang mit unzureichendem Requirements Engineering bringen. Zudem wurde festgestellt, dass ein gut durchgeführtes Requirements Engineering positive Auswirkungen auf die nachfolgenden Entwicklungsaktivitäten hat. Insgesamt wird ein Projekt dann als Erfolg wahrgenommen, falls es die Kundenanforderungen gut erfüllt.

Das Requirements Engineering für PSS ist aufgrund verschiedener Faktoren besonders herausfordernd. Es muss in der Lage sein die Anforderungen an die Lösung als Ganzes und an die einzelnen domänenspezifischen Komponenten von PSS zu ermitteln und zu verwalten. Verschiedene Faktoren erfordern einen integrierten, abgestimmten und domänenübergreifenden Ansatz zu Requirements Engineering, der ebenso in der Lage ist mit den zyklischen Wechselwirkungen umzugehen. Diese Faktoren sind: die Beteiligung von mehreren Domänen, die ein unterschiedliches Verständnis über das Requirements Engineering und den Begriff „Anforderung“ haben, die hohe Komplexität des Entwicklungsprozesses, der hohe Grad der technologischen Integration der Lösung, die unterschiedlichen Lebenszyklen der einzelnen Komponenten von PSS und der hohe Grad der Kundenintegration (Berkovich 2010; Berkovich et al. 2009a, 2010a; Berkovich et al. 2009c).

4.2 Begriffliche Grundlagen

4.2.1 Anforderung

Eine Anforderung (eng. *Requirement*) wird definiert als ((IEEE 1990) zitiert in (Pohl 2007, 13)):

- „(1) Eine Bedingung oder Eigenschaft, die ein System oder eine Person benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen.
- (2) Eine Bedingung oder Eigenschaft, die ein System oder eine Systemkomponente aufweisen muss, um einen Vertrag zu erfüllen oder einem Standard, einer Spezifikation oder einem anderen formell auferlegten Dokument zu genügen.
- (3) Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft wie in (1) oder (2) definiert“.

Kotonya und Sommerville (1998, 7) ergänzen diese Definition, indem sie eine spezifische Einschränkung an das zu entwickelnde Systems und eine Einschränkung an den Entwicklungsprozess des Systems als eine Anforderung bezeichnen. Eine ähnliche Definition geben auch Robertson und Robertson (2007, 9): “A requirement is something that the product must do or a quality that the product must have. A requirement exists either because the type of product demands certain functions or qualities, or the client wants that requirement to be part of the delivered product”. Diese Definitionen stammen aus der Softwareentwicklung und werden ebenso in der Dienstleistungsentwicklung verwendet (vgl. (Husen 2007, 25-28)). Sie drücken verallgemeinert aus, dass eine Anforderung Wünsche und Ziele von Benutzern als auch Bedingungen und Eigenschaften des zu entwickelnden Systems darstellen.

Im Bereich der Produktentwicklung definiert Lindemann (2006, 327) eine Anforderung als:

- „(1) Festlegung der qualitativen und quantitativen Eigenschaften eines Produktes aus der Sicht des Auftraggebers;
- (2) Eigenschaft, die das zu entwickelnde System (Produkt, Dienstleistung, Prozess, etc.) erfüllen muss;
- (3) Geforderte Ausprägung eines Merkmals“.

Ahrens (2000, 213) weist darauf hin, dass eine Anforderung „eine Vorgabe, deren Erfüllung den zielgerichteten Verlauf des jeweiligen Konstruktionsprozesses und/oder Eigenschaften des betreffenden Produktes bestimmt ist“.

Die Anforderungen, die die Soll-Eigenschaften, also die zukünftigen Eigenschaften des zu entwickelnden Produktes, darstellen, resultieren durch die Analyse und die damit verbundene Konkretisierung in den Ist-Eigenschaften des Produktes, die sich aus Merkmalen und Ausprägungen zusammensetzen (Spath/Demuß 2003, 485ff.). Durch die Angaben von qualitativen oder quantitativen Ausprägungen werden die initialen Stakeholderanforderungen konkretisiert und in eine technische Beschreibung übersetzt (Husen 2007, 37). Diese konkretisierten Anforderungen werden als Lösungsanforderungen (Ebert 2005, 41) bezeichnet.

Eine Anforderung beschreibt somit ein Bedürfnis oder einen Nutzen, der zu erreichen ist. Es gibt verschiedene Sichtweisen auf eine Anforderung, wie in Abbildung 4-3 dargestellt ist.

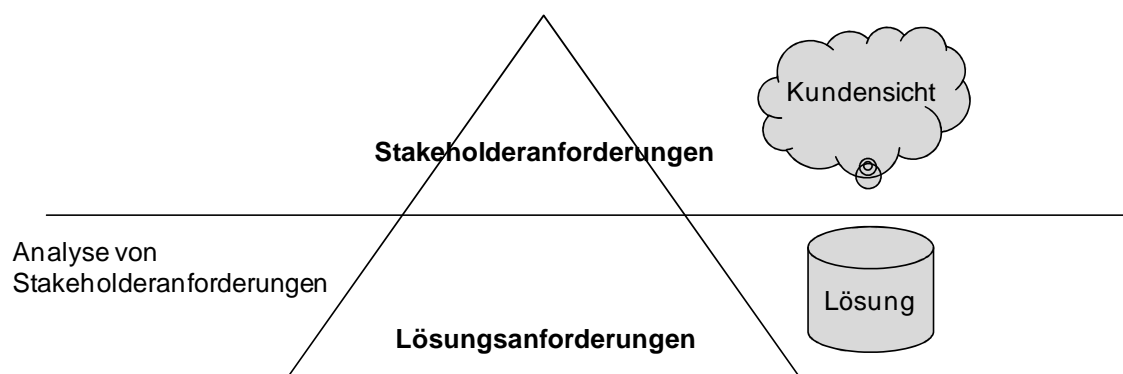


Abbildung 4-3: Verschiedene Sichtweisen auf Anforderungen

Quelle: In Anlehnung an (Hull et al. 2011, 27; Ebert 2005, 40)

Die initialen Stakeholderanforderungen werden „in der Sprache“ und im Kontext des Kunden formuliert und müssen somit, bevor sie an die Entwicklung übergeben werden, konkretisiert werden, so dass die Entwickler sie verstehen und umsetzen können (vgl. (Grady 2006, 7)). Die Kunden- und Stakeholderanforderungen sind meist unpräzise, lösungsneutral und können nicht unmittelbar realisiert werden. Die aus den Kunden- und Stakeholderanforderungen durch eine Analyse abgeleiteten Lösungsanforderungen sind detaillierter, mit qualitativen oder quantitativen Angaben versehen und setzen die Annahme einer bestimmten angestrebten Lösung voraus (vgl. (Ebert 2005, 40f.; Pohl 2007, 49; Hull et al. 2011, 26)). Anschließend werden die Lösungsanforderungen weiter konkretisiert und auf Komponenten des Produktes abgebildet (vgl. (Ebert 2005, 40f.; Hull et al. 2011, 26)). Somit bilden die Lösungsanforderungen den Lösungsraum.

„Good requirements are essential if we are to be sure that we are building the system the users want, and that we are not doing more than is needed“ (Hall 1997). Anforderungen sind feste Bestandteile von Verträgen, Entwicklungsaufträgen oder Projektplänen und dienen als Grundlage für Abschätzung, Planung, Durchführung und Verfolgung von Tätigkeiten im Rahmen eines Projektes (Ebert 2005, 9). Sie dienen als Grundlage zur Kommunikation und Diskussion mit allen am Entwicklungsprozess Beteiligten, indem sie das gemeinsame Verständnis und Wissen der Teammitglieder explizieren. Ebenso werden sie als Grundlage für die Integration, Wartung und Pflege des Systems nach der Abnahme des Systems durch den Kunden gesehen (Rupp 2004, 18f.).

4.2.2 Requirements Engineering und Management

Requirements Engineering (dt. *Anforderungsmanagement*) ist eine domänenneutrale Disziplin, d. h. es kann für die Entwicklung von Hardware, Software, mechatronischer Produkte oder Dienstleistungen eingesetzt werden (vgl. (Berenbach et al. 2009, 8)). Oft wird zwischen Requirements Engineering und Requirements Management unterschieden, wobei Requirements Management als ein Teil des Requirements Engineering betrachtet wird (bspw. (Wieggers 2005; Rupp 2004; Pohl 2007; Kotonya/Sommerville 1998)). Einige Quellen betrachten Requirements Engineering als Teil des Requirements Managements (bspw. (Leffingwell/Widrig 2003)) oder als Teil des Produkt- und Projektmanagements (bspw. (Schienmann 2001)). Im Rahmen dieser Arbeit wird Requirements Management als Teil des Requirements Engineering betrachtet, das sich mit der Verwaltung der Anforderungen beschäftigt.

Hull et al. (2011, 8) definieren Requirements Engineering als „the subset of systems engineering concerned with discovering, developing, tracing, analyzing, qualifying, communicating and managing requirements that define the system at successive levels of abstraction“. In der Softwareentwicklung ist Requirements Engineering eine eigenständige Disziplin, „concerned with the real world goals for, functions of, and constraints on software systems“ (Zave 1997), während in der Produkteentwicklung Requirements Engineering, auch bekannt unter Anforderungsklä rung oder Aufgabenklärung, als Teil des Entwicklungsprozesses angesehen wird (vgl. (Jung 2006; Pahl et al. 2006; Ehrlenspiel 2002; Cross 2008; Almfelt et al. 2003)). Die Dienstleistungsentwicklung ist eine noch ziemlich junge Disziplin, in der das Requirements Engineering nicht stark ausgeprägt ist. Allerdings existieren schon erste Ansätze zum Requirements Engineering (vgl. (Husen 2007)).

Requirements Engineering ist „ein kooperativer, iterativer, inkrementeller Prozess, dessen Ziel es ist zu gewährleisten, dass

- (1) alle relevanten Anforderungen bekannt und in dem erforderlichen Detaillierungsgrad verstanden sind,
- (2) die involvierten Stakeholder eine ausreichende Übereinstimmung über die bekannten Anforderungen erzielen,
- (3) alle Anforderungen konform zu den Dokumentationsvorschriften dokumentiert bzw. konform zu den Spezifikationsvorschriften spezifiziert sind“ (Pohl 2007, 43).

Somit zielt das Requirements Engineering „auf die systematische Erhebung der Anforderungen für ein zu entwickelndes Produkt oder System auf Basis genereller Geschäftsziele und Vorgaben ab. Angestrebt werden dokumentierte Anforderungen an Produkte oder Systeme, die optimal nutzbar und vermarktbar sind und in jeder Hinsicht möglichst gut den Erwartungen aller Beteiligten entsprechen“ (Broy et al. 2007, 128).

Zusammenfassend lässt sich feststellen, dass das Requirements Engineering eine Unterstützung für die Anforderungsermittlung, Anforderungsanalyse und -vereinbarung, Anforderungsdokumentation, Anforderungvalidierung, sowie das Änderungsmanagement und die Anforderungsverfolgung anbieten soll (vgl. (Lamsweerde 2009; Aurum/Wohlin 2005a; Kotonya/Sommerville 1998; Cheng/Atlee 2007; Pohl 2007)). Abbildung 4-4 zeigt das Zusammenspiel zwischen den Aktivitäten des Requirements Engineering.

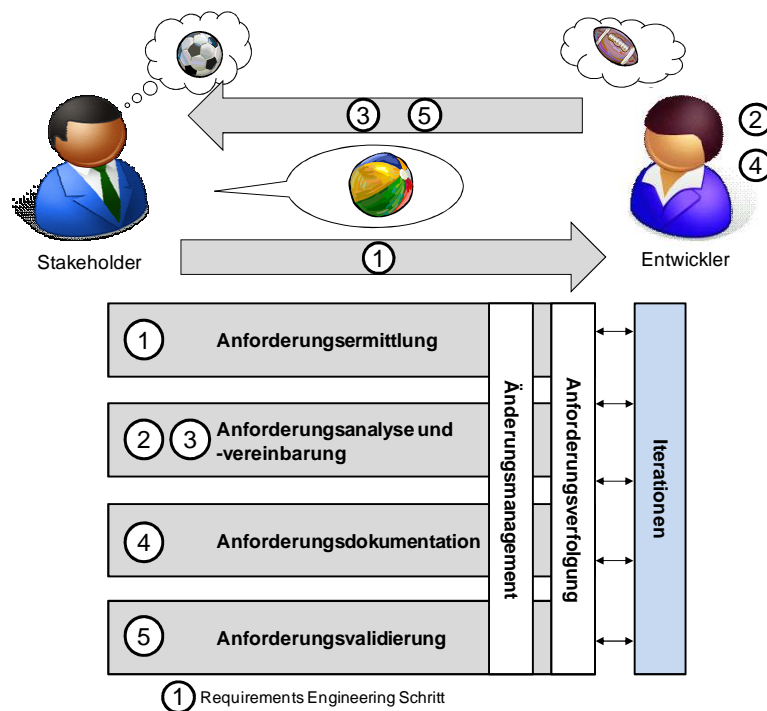


Abbildung 4-4: Aktivitäten des Requirements Engineering

Quelle: In Anlehnung an (Lamsweerde 2009; Aurum/Wohlin 2005a; Kotonya/Sommerville 1998; Cheng/Atlee 2007; Pohl 2007)

Schritt 1: Anforderungsermittlung. Der erste Schritt des Requirements Engineering ist die Anforderungsermittlung. Die Anforderungsermittlung soll die Ziele und Motivation für das zu entwickelnde System entdecken und die initialen Anforderungen und Wünsche der Kunden und Stakeholder identifizieren und erheben (Cheng/Atlee 2007; Berenbach 2006, 40; Lindemann 2006, 100ff.; Husen 2007, 70ff.). Dafür müssen das Einsatzgebiet für das zu entwickelnde System, das eigentliche Problem des Kunden, das durch das System zu lösen ist, verstanden werden, die Geschäftsprozesse des Kunden, die einen Einfluss auf das zu entwickelnde System haben, analysiert werden und die eigentlichen Anforderungen, Wünsche und Einschränkungen der Kunden bezüglich des Systems verstanden werden (Kotonya/Sommerville 1998, 54f.).

Schritt 2: Anforderungsanalyse und Schritt 3: Anforderungsvereinbarung. Die Anforderungsanalyse und die Anforderungsvereinbarung stehen in einem engen Zusammenhang. Die Anforderungsanalyse und -vereinbarung sind Aktivitäten, „which aim to discover problems with the system requirements and reach agreement on changes to satisfy all system stakeholders“ (Kotonya/Sommerville 1998, 77).

Während der Anforderungsanalyse geht es nicht nur darum, die Anforderungen und Bedürfnisse zu analysieren, sondern zu beschreiben, wie die zukünftige Lösung aussieht, die die Anforderungen erfüllt (Ebert 2005, 133; Ramaswamy 1996, 28-31). Die Anforderungsanalyse hat das Ziel, die Anforderungen auf Inkonsistenzen zu überprüfen, die initialen Anforderungen an das System, die im Rahmen der Anforderungsermittlung erhoben wurden, zu konkretisieren und sie in einer formalen Spezifikation festzuhalten (Berenbach et al. 2009, 40; Ehrlenspiel 2002, 118f.). Nach der Analyse werden die konkretisierten Anforderungen als Lösungsanforderungen bezeichnet (Hull et al. 2011, 27; Ebert 2005, 40). Konkretisierung bedeutet, dass die Anforderungen immer detaillierter und präziser in Bezug auf die zu entwi-

ckelnde Lösung werden, quantitative und qualitative Angaben erhalten und somit „in die Sprache des Entwicklers“ übersetzt werden (Hull et al. 2011, 26; Kotonya/Sommerville 1996, 139). Ein weiterer Aspekt der Anforderungsanalyse ist die Priorisierung von Anforderungen. Durch die Angabe der Priorität einer Anforderung wird die Bedeutung dieser Anforderung hinsichtlich eines oder mehrerer ausgewählter Priorisierungskriterien beschrieben. So können durch die Priorisierung bspw. die Stakeholder die für sie besonders wichtigen Anforderungen angeben.

Ein Konflikt zwischen den Anforderungen besteht dann, wenn die Bedürfnisse und Vorstellungen der Stakeholder an das zu entwickelnde System sich widersprechen oder nicht alle Bedürfnisse und Wünsche der Stakeholder im System berücksichtigt werden können. Das Ziel der Anforderungsvereinbarung (oder Übereinstimmung) ist die Identifikation von Konflikten, Analyse von Konfliktursachen, Auflösung von Konflikten durch geeignete Strategien und Dokumentation der Konfliktauflösung einschließlich der Begründung (Pohl 2007, 394; Husen 2007, 96-97).

Schritt 4: Anforderungsdokumentation. Das Ziel der Anforderungsdokumentation ist es die Anforderungen, die während des Entwicklungsprozesses ermittelt oder geändert werden, gemäß den Dokumentationsvorschriften zu dokumentieren und sie im finalen Anforderungsdokument zu spezifizieren (Hull et al. 2011, 77-79; Husen 2007, 109-113).

Schritt 5: Anforderungvalidierung. Je besser die Kundenanforderungen erfüllt sind, desto besser ist die Akzeptanz beim Kunden (Versteegen 2002, 15). Die Anforderungvalidierung soll sicherstellen, dass die analysierten und dokumentierten Lösungsanforderungen die Wünsche und Vorstellungen der Stakeholder ausdrücken (Cheng/Atlee 2007; Husen 2007, 112-116).

Anforderungsverfolgung. Die Verfolgung von Anforderungen (syn. *Anforderungsnachvollziehbarkeit*) ist eine Querschnittsaktivität, die während des gesamten Lebenszyklus eines Produktes zu gewährleisten ist. Unter der Verfolgung einer Anforderung versteht man laut Pohl (2007, 505ff.) die Fähigkeit den Lebenszyklus einer Anforderung von ihrem Ursprung bis hin zur Umsetzung in nachgelagerten Entwicklungsschritten zu verfolgen. Das Ziel der Verfolgung von Anforderungen besteht darin, die Auswirkungen von Änderungen an Anforderungen abzuschätzen, den Entwicklungszustand einer Anforderung sowie den Einfluss und die Interdependenzen zwischen den Anforderungen, die für weitere Entwicklungsergebnisse relevant sind, zu identifizieren. Durch die Identifikation der Interdependenzen zwischen den Anforderungen unterstützt die Anforderungsnachvollziehbarkeit das Änderungsmanagement, indem die Auswirkungen von Änderungen von Anforderungen auf weitere Anforderungen oder Komponenten des Systems festgelegt werden können (Dahlstedt/Persson 2005). Die Anforderungsverfolgung erlaubt es, den Zweck oder Nutzen jedes Entwicklungsergebnisses aus Kundensicht im Projektverlauf zu messen. So kann bspw. festgestellt werden, ob sich der bisher erreichte Stand eines Projektes aus den Kundenanforderungen ableiten lässt (Ebert 2005, 177-178).

Änderungsmanagement. Die Anforderungen ändern sich ständig, angefangen mit ihrer Entstehung und endend mit der Verwertung des Produktes (Robertson/Robertson 2007, 357). Das Änderungsmanagement ist ebenso wie die Anforderungsverfolgung eine Querschnittsaktivität, die während des gesamten Lebenszyklus eines Produktes durchzuführen ist. Das Änderungsmanagement soll die Änderungen der Anforderungen aufnehmen, die Auswirkungen dieser Änderungen auf andere Anforderungen, das System selbst, Stakeholder und die Umge-

bung des Systems prüfen, sowie die Kosten und Aufwand bei der Umsetzung von diesen Änderungen analysieren. Wenn eine Änderung umgesetzt werden soll, soll das Änderungsmanagement einen entsprechenden Auftrag zur Realisierung erstellen (vgl. (Pohl 2007, 534ff.; Kotonya/Sommerville 1998, 123f.; Hull et al. 2011, 170f.; Lamsweerde 2009, 247ff.))

4.3 Aktivitäten des Requirements Engineering im Lebenszyklus von Product Service Systems

Aus Kapitel 4.2.2 wird ersichtlich, welche Aktivitäten das Requirements Engineering bei der Entwicklung eines Systems abdeckt und aus welchen Aktivitäten es besteht. Weiterhin ist zu klären, welche Aktivitäten das Requirements Engineering im Lebenszyklus von PSS hat. Dafür müssen der Lebenszyklus von PSS (Kapitel 3.3) und die Ziele und Aktivitäten des Requirements Engineering (Kapitel 4.2.2) gegenübergestellt und intersubjektiv nachvollziehbar analysiert werden. Somit können die kausalen Abhängigkeiten zwischen den Lebenszyklusphasen und den Zielen und Aktivitäten des Requirements Engineering identifiziert werden.

Zu Beginn der Entwicklung von PSS sollen im Rahmen der *Anforderungsmodellierung* die relevanten Stakeholder identifiziert und die Anforderungen an das PSS ermittelt werden (**Anforderungsermittlung**). Diese Anforderungen sind zu analysieren und im Rahmen der *Produktmodellierung* (vgl. Kapitel 3.3.1) nach materiellen und immateriellen Bündeln aufzuteilen.

Die ermittelten Anforderungen liefern die Grundlage zum Aufstellen der Funktionsstrukturen im Rahmen der *Funktionsmodellierung* für materielle und immaterielle Bestandteile des PSS. Dafür werden die Anforderungen erneut analysiert, konkretisiert und den Funktionen der Funktionsstrukturen zugeordnet. Im Rahmen der anschließenden *Prinzipmodellierung* werden die Anforderungen konkretisiert und auf die einzelnen Komponenten, wie Hardware, Software und Dienstleistungen, aufgeteilt. Damit stellen die so konkretisierten Anforderungen (**Lösungsanforderungen**), die sich auf einzelnen domänenspezifischen Komponenten beziehen, die Grundlage für eine integrierte Entwicklung der domänenspezifischen Komponenten im Schritt „*Prozess- und Gestaltmodellierung*“ dar (**Anforderungsanalyse**). Im Rahmen dieser Arbeit werden die analysierten und konkretisierten domänenspezifischen Anforderungen als Lösungsanforderungen (vgl. (Ebert 2005, 40)) bezeichnet. So beschreiben die Lösungsanforderungen die gewünschte Funktionalität der domänenspezifischen Komponenten des zu entwickelnden PSS, d. h. Aufgaben, die jede Komponente erfüllen muss, damit die Gesamtheit der Komponenten eine auf die Anforderungen der Kunden abgestimmte Lösung ergibt.

Während der Phasen der *Produktmodellierung*, *Funktionsmodellierung* und *Prinzipmodellierung* können Konflikte zwischen den Anforderungen vorkommen. Diese Konflikte sind zu identifizieren und aufzulösen (**Anforderungsvereinbarung**). Sowohl die initialen Anforderungen als auch die analysierten Anforderungen sind entsprechend den Dokumentationsvorschriften zu dokumentieren (**Anforderungsdokumentation**).

Ein weiterer Aspekt ist die Sicherstellung, dass die analysierten Anforderungen nach jeder Phase der *Produktmodellierung*, *Funktionsmodellierung*, *Prinzipmodellierung* und *Prozess- und Gestaltmodellierung* die initialen Anforderungen der Kunden und Stakeholder abdecken (**Anforderungvalidierung**).

Während der oben genannten Phasen und der integrierten Entwicklung der domänenspezifischen Komponenten des PSS, der anschließenden Integration des PSS in die Wertschöpfungsprozesse des Kunden sowie der Instandhaltung, Wartung und Inspektion können sich die

Anforderungen ändern. Diese Änderungen sind entsprechend aufzunehmen, zu analysieren und Maßnahmen zum Umgang mit den Änderungen an Anforderungen sind einzuleiten (**Änderungsmanagement**). In diesem Zusammenhang soll die Anforderungsverfolgung ermöglichen den Lebenszyklus einer Anforderung bis zu ihrer Übergabe an die Entwicklung zu verfolgen und die Interdependenzen zwischen den Anforderungen abzubilden (**Anforderungsverfolgung**). Abbildung 4-5 zeigt das Zusammenspiel zwischen dem Lebenszyklus von PSS und dem Requirements Engineering.

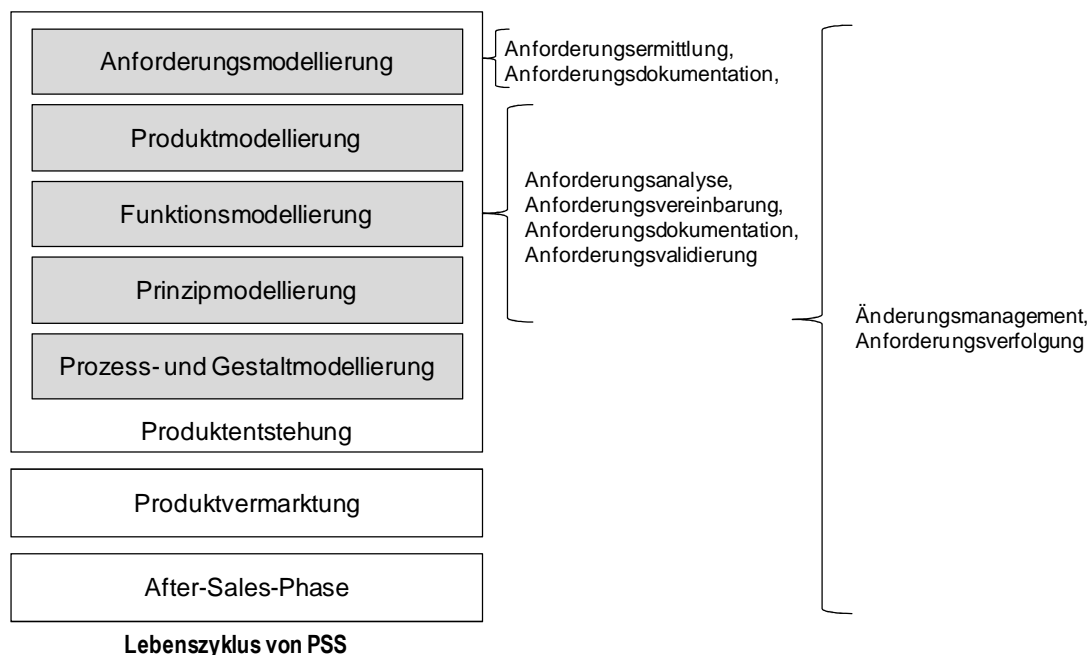


Abbildung 4-5: Zusammenspiel zwischen dem Lebenszyklus von PSS und Requirements Engineering

Quelle: In Anlehnung an (Berkovich et al. 2010b)

Das Requirements Engineering stellt somit eine phasenübergreifende Aktivität dar, die über mehrere Phasen der Entwicklung und nicht nur am Anfang, bevor die eigentliche Entwicklung beginnt, erfolgt.

4.4 Stand der Wissenschaft: Requirements Engineering für Product Service Systems in der Literatur

Die an der Entwicklung von PSS beteiligten Domänen, Produkt-, Software- und Dienstleistungsentwicklung, haben unterschiedliche Vorgehensweisen zur Ermittlung und Verwaltung von Anforderungen (Berkovich 2010; Berkovich et al. 2009a, 2010a). Um zu überprüfen, wie die einzelnen domänenspezifischen Vorgehen zum Requirements Engineering für IT-gestützte PSS geeignet sind, soll eine strukturierte Literaturanalyse durchgeführt werden. Die Beschreibung und Ergebnisse der Literaturanalyse sind auch unter Berkovich et al. (2011d)⁴ zu finden.

⁴ Accepted for publication

4.4.1 Aufbau der Literaturanalyse

4.4.1.1 Zielsetzung der Literaturanalyse

Durch die immer größere Anzahl an wissenschaftlichen Quellen steigt die Bedeutung systematischer Literaturanalysen stark (Webster/Watson 2002). Eine Literaturanalyse ermöglicht vorhandene Erfahrungen in der Wissenschaft und Praxis über einen bestimmten Aspekt zu ergreifen und die Forschungslücken zu identifizieren (Budgen/Brereton 2006). Um eine Forschungsfrage zu beantworten oder ein Phänomen zu untersuchen, können mit Hilfe einer systematischen Literaturanalyse die aktuellen verfügbaren Forschungsergebnisse identifiziert, evaluiert und erklärt werden (Budgen/Brereton 2006). Zur Sicherstellung eines wissenschaftlichen Wertes einer Literaturanalyse ist es notwendig bei der Durchführung der Analyse einer wissenschaftlichen Methode zu folgen.

Die Idee, bestehende Forschungsergebnisse mit Hilfe einer systematischen Literaturanalyse zu erfassen und in der Praxis umzusetzen, stellt die Grundlage der evidenzbasierten Forschung dar (Kitchenham et al. 2009). In der evidenzbasierten Softwareentwicklung hat bspw. eine Literaturanalyse die Aufgabe, Best Practices aus der Praxis und Erkenntnisse aus der Forschung zu entdecken und den Prozess der Entwicklung und Wartung von Software zu integrieren (Kitchenham et al. 2004).

Der Auswahl der Forschungsmethode der Literaturanalyse liegt das Klassifizierungsschema der Forschungsmethoden in der Wirtschaftsinformatik von Wilde und Hess (2007) zu Grunde. Die Literaturanalyse wird intersubjektiv-nachvollziehbar durchgeführt, somit werden die Schlussfolgerungen aus der Literaturanalyse argumentativ-deduktiv begründet (vgl. (Wilde/Hess 2007)). Zur Durchführung der in diesem Kapitel beschriebenen Literaturanalyse wurde der autorenbasierte Ansatz nach Webster und Watson (2002) gewählt, in dem die betrachteten Konzepte den Autoren zugeordnet werden. Zur Planung der einzelnen Schritte bei der Literaturanalyse wurden die Ansätze von Brereton et al. (2007) sowie Kitchenham et al. (2009) herangezogen. Somit werden folgende Schritte durchgeführt (Berkovich et al. 2011d):

- 1) **Definition der Forschungsfragen.** Im ersten Schritt wurden die Forschungsfragen definiert, die im Rahmen der Literaturanalyse beantwortet werden sollen. Die forschungsleitenden Fragestellungen der Literaturanalyse lauten „*Sind die ausgewählten domänenspezifischen Ansätze zum Requirements Engineering für PSS geeignet?*“ und darauf folgend „*In wieweit können diese Ansätze für PSS eingesetzt werden?*“.
- 2) **Definition und Validierung des Analyseprotokolls.** Das Analyseprotokoll beschreibt die Kriterien, nach welchen die gefundene Literatur bewertet wird. Die Kriterien der Literaturanalyse sollen zur Beantwortung der Forschungsfragen verwendet werden, indem sie relevante Merkmale für die Eignung der Requirements Engineering Ansätze für PSS detailliert beschreiben und dabei die Entscheidung, wie gut die betrachteten Ansätze für den Einsatz im Requirements Engineering für PSS geeignet sind, unterstützen. Die Kriterien basieren auf dem Zusammenspiel zwischen dem Lebenszyklus von PSS und den Aktivitäten des Requirements Engineering (Kapitel 4.3), welche in einem Ordnungsrahmen zusammengefasst werden. Der Ordnungsrahmen wird generisch gestaltet, um älterer und künftiger Forschung dadurch Struktur verleihen zu können (Goeken/Patas 2010). Der Ordnungsrahmen beschreibt den Lebenszyklus und die Eigenschaften von PSS und deren Zusammenhang mit den Aktivitäten des Requirements Engineering (Kapitel 4.4.1.4). Zur Validierung des Analyseprotokolls wurde ein

Testanlauf mit dem Vorgehensmodell von Lindemann (2006) durchgeführt, das aus früheren Recherchen bekannt war (s. (Berkovich et al. 2009a)). Daraufhin wurden die Kriterien entsprechend angepasst, indem der Fokus auf die PSS-spezifischen Aspekte geschärft wurde (z. B. Berücksichtigung der Zugehörigkeit der Anforderungen zu verschiedenen Domänen bei der Suche nach Konflikten).

- 3) **Identifikation und Auswahl der relevanten Literatur.** In diesem Schritt wurden die für die Literaturanalyse relevanten wissenschaftlichen Quellen identifiziert. Das genaue Vorgehen ist in Kapitel 4.4.1.2 beschrieben.
- 4) **Erhebung und Analyse der Daten.** Die zur Beantwortung der Forschungsfragen benötigten Daten sind zu extrahieren und entsprechend den im Protokoll definierten Kriterien zu analysieren. Die Ergebnisse der Literaturanalyse sind in Kapitel 4.4.2.5 beschrieben.

Die Entwicklung von PSS ist von mehreren zyklischen Wechselwirkungen geprägt, die ebenso einen Einfluss auf das Requirements Engineering haben (Kapitel 3.3.2). Im Rahmen der Literaturanalyse wurden zyklische Wechselwirkungen im Requirements Engineering identifiziert. Sie wurden in Kapitel 4.4.3 beschrieben.

4.4.1.2 Auswahl der Literatur

Aufgrund des verschiedenen Verständnisses von Requirements Engineering in den jeweiligen Domänen wurde die Auswahl der zu analysierenden Quellen in jedem Bereich einzeln vorgenommen. Zur Auswahl der untersuchten Literatur wurden die am häufigsten zitierten Bücher laut „google scholar“ und die am häufigsten verkauften Bücher laut „amazon.de“ und „amazon.com“ herangezogen. Im Bereich der Ingenieurwissenschaften und der Informatik deckt google scholar einen Großteil der Journale und elektronischen Publikationen ab. Meier und Conkling (2008) haben gezeigt, dass für die Ingenieurwissenschaften ca. 90% der Publikationen die nach 1990 veröffentlicht wurden in google scholar indiziert sind. Im Bereich der Informatik sind wichtige Verlage wie beispielsweise Elsevier indiziert (Jacsó 2008). Ebenso sind die bekanntesten non-profit Organisationen, wie beispielsweise ACM und IEEE indiziert (Meier/Conkling 2008).

Insgesamt wurden die nach diesen Kriterien 15 führenden generische Vorgehensmodelle und Ansätze zum Requirements Engineering aus Produkt-, Software- und Dienstleistungsentwicklung (Top 5 je Domäne) sowie fünf integrierte Ansätze zur Entwicklung von PSS untersucht. Als Auswahlkriterium galt auf Grund der großen Anzahl an verfügbaren Quellen die Generik der ausgewählten Quellen, die für das Requirements Engineering eine Allgemeingültigkeit und gewisse Verbreitung aufweisen sollen (Goeken/Patas 2010). Laut Goeken und Patas (2010) wird dies ermöglicht, wenn sie als „Referenzmodelle“ bezeichnet werden, durch Lehrbücher eine Verbreitung fanden, aus Standardisierungsvorhaben hergeleitet wurden oder als (Quasi-) Standards etabliert sind. Durch die Auswahl generischer Ansätze soll eine übergeordnete Strukturierung des für das Requirements Engineering spezifischen Wissens geschaffen werden (Goeken/Patas 2010).

Das Ziel der Literaturanalyse ist es, generische Komponenten des Requirements Engineering für PSS auf einer Metaebene und keine konkreten Komponenten, wie z. B. „Anforderungsspezifikation am Ende der Phase der Validierung“, zu definieren (Berkovich et al. 2011d). Es wurden deswegen je fünf generische Vorgehensmodelle und Ansätze, die in den Lehrbüchern veröffentlicht sind, zur detaillierten Literaturanalyse gewählt, weil die generischen Vorgehen

gewisse Ähnlichkeiten aufweisen, was die Analyse weiterer Vorgehen überflüssig macht. Die Lehrbücher eignen sich in diesem Fall zur Analyse auch deswegen gut, weil sie die generischen Modelle detailliert beschreiben, während die Konferenz- und Journal-Beiträge oft keine nötige Breite des betrachteten Gegenstandes aufweisen und sich auf bestimmte Aspekte, wie z. B. methodische Unterstützung zur Durchführung einer Aktivitäten im Requirements Engineering, konzentrieren.

In der Softwareentwicklung wird Requirements Engineering als eigene Disziplin verstanden. Dementsprechend gibt es umfassende Literatur, die sich speziell auf das Thema des Requirements Engineering konzentriert. Daher wurden für die Literaturanalyse die am meisten verkauften Bücher für die Suchbegriffe „Requirements Engineering“ und „Anforderungsmanagement“ auf *amazon.de* sowie *amazon.com* (zugegriffen im Juni 2008) herangezogen. Um die neuesten relevanten Forschungsergebnisse nicht zu übersehen, wurden ebenso einschlägige Quellen aus der Wissenschaft und Praxis zur Literaturanalyse herangezogen. Es wurden Konferenz- und Journal-Beiträge, die sich mit dem Requirements Engineering beschäftigen, in den Suchportalen IEEEExplore⁵, ACM Digital library⁶ und Inspec⁷ (vgl. (Brereton et al. 2007)) gesucht. Die WI-Orientierungslisten (2008) geben einen Überblick über die für den Wirtschaftsinformatikbereich relevanten Journals und Konferenzen. Für die Suche nach relevanten Quellen wurden alle A-Journals und A-Konferenzen herangezogen, indem nach Beiträgen, die das Requirements Engineering behandeln, aus den Jahren 2003-2010 gesucht wurde. Als Schlüsselwörter wurden „Requirements Engineering“, „Requirements Management“, „Anforderungsmanagement“, „Requirement“ und „Anforderung“ verwendet. Die Auswahl der Beiträge fand basierend auf den Inhalten der Analysekriterien statt (Kapitel 4.4.1.5). Die erste Auswahl fand anhand der Titel der Beiträge statt. Anschließend wurden die Abstracts der Beiträge analysiert. Insgesamt wurden 154 Beiträge zur näheren Analyse ausgewählt.

In der Produktentwicklung und in der Dienstleistungsentwicklung wird Requirements Engineering im Rahmen allgemeiner Entwicklungsmethodiken behandelt. Zur Auswahl der allgemeinen Entwicklungsmethodiken wurden die Begriffe „Produktentwicklung“ / „Product Engineering“ und „Service Engineering“ / „Dienstleistungsentwicklung“ in *google-scholar* eingegeben und die 100 am meisten zitierten Arbeiten ausgewählt. Diese Arbeiten wurden bezüglich ihrer Behandlung des Themas des Requirements Engineering bewertet. In die letztendliche Analyse wurden nur jene Ansätze aufgenommen, die einen substantiellen Beitrag zum Requirements Engineering leisten, d.h. die mindestens einen eigenen Themenbereich dem Requirements Engineering widmen. Da durch diese Suche vor allem Ansätze, die den gesamten Entwicklungsprozess abdecken, gefunden wurden, wurde anschließend nach den oben genannten Begriffen in Kombination mit „Requirements Management“ / „Anforderungsmanagement“ / „Requirements Engineering“ gesucht, mit dem Ziel speziellere Ansätze für das Requirements Engineering zu finden. Auch hier wurden aus den Suchergebnissen jene Ansätze ausgewählt, welche sich zu einem bedeutenden Teil mit dem Requirements Engineering beschäftigen. In diese Suche wurden jeweils die ersten 100 Ergebnisse von „*google-scholar*“ einbezogen. Ebenso wurden im Bereich der Produktentwicklung auf die interne Richtlinie mit der Liste von relevanten Journals und Konferenzen vom Lehrstuhl für Produktentwicklung⁸ der Technischen Universität München zurückgegriffen und manuell Beiträge,

⁵ <http://ieeexplore.ieee.org/>

⁶ <http://portal.acm.org/>

⁷ <http://www.iee.org/publish/inspec/>

⁸ <http://www.pe.mw.tum.de>

die das Thema des Requirements Engineering im Fokus haben, aus den Jahren 2003-2010 ausgewählt. Im Bereich der Produktentwicklung wurden 90 Beiträge und in der Dienstleistungsentwicklung 46 Beiträge analysiert.

Darüber hinaus umfasste die Recherche fünf Ansätze zur Entwicklung von PSS, die im Laufe der Recherche zu den Eigenschaften von PSS bekannt geworden sind. Bei der Auswahl der Ansätze stand ihre Beschäftigung mit der Anforderungsverwaltung im Vordergrund.

Tabelle 4-1 listet die analysierten Beiträge zum Requirements Engineering aus der Produktentwicklung auf (Berkovich et al. 2011d).

Ausgewählte Beiträge aus der Produktentwicklung
Lindemann, U. (2006): Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden (2 Aufl.), Springer, Berlin 2006.
Ehrlenspiel, K. (2002): Integrierte Produktentwicklung (2 Aufl.), Hanser Fachbuchverlag 2002.
Pahl, G.; Beitz, W.; Feldhusen, J. (2006): Konstruktionslehre: Grundlagen Erfolgreicher Produktentwicklung. Methoden Und Anwendung. (7 Aufl.), Springer, Berlin 2006.
Verein Deutschr Ingenieure (1993): VDI 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, 1993.
Ahrens, G. (2000): Das Erfassen und Handhaben von Produkthanforderungen - methodische Voraussetzungen und Anwendung in der Praxis -. Dissertation, Technische Universität Berlin 2000.

Tabelle 4-1: *Analysierte Beiträge zum Requirements Engineering aus der Produktentwicklung*

Quelle: (Berkovich et al. 2011d)

Tabelle 4-2 listet die analysierten Beiträge zum Requirements Engineering aus der Softwareentwicklung auf.

Ausgewählte Beiträge aus der Softwareentwicklung
Aurum, A.; Wohlin, C. (2005): Engineering and Managing Software Requirements (1 Aufl.), Springer, Berlin 2005.
Ebert, C. (2005): Systematisches Requirements Management. (1 Aufl.), Dpunkt Verlag 2005.
Lamsweerde, A. van (2009): Requirements Engineering: From System Goals to UML Models to Software Specifications (1 Aufl.), Wiley & Sons 2009.
Pohl, K. (2007): Requirements Engineering. Grundlagen, Prinzipien, Techniken (1 Aufl.), Dpunkt Verlag 2007.
Kotonya, G., Sommerville, I. (1998): Requirements Engineering: Processes and Techniques (1 Aufl.), Wiley & Sons 1998.

Tabelle 4-2: *Analysierte Beiträge zum Requirements Engineering aus der Softwareentwicklung*

Quelle: (Berkovich et al. 2011d)

Tabelle 4-3 listet die analysierten Beiträge zum Requirements Engineering aus der Dienstleistungsentwicklung auf.

Ausgewählte Beiträge aus der Dienstleistungsentwicklung
Bullinger, H.-J.; Schreiner, P. (2003): Service Engineering: Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Hrsg.: Bullinger, H.-J.; Scheer, A.-W., Springer 2003.
Edvardsson, B.; Olsson, J. (1996): Key Concepts for New Service Development. In: Service Industries Journal, Vol. 16 (1996) Nr. 2, S. 140-164.
Frietzsche, U.; Maleri, R. (2003): Dienstleistungsproduktion. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Hrsg.: Bullinger, H.-J.; Scheer, A.-W., Springer 2003.
Husen, C. van (2007): Anforderungsanalyse für produktbegleitende Dienstleistungen, Universität Stuttgart 2007.
Ramaswamy, R. (1996): Design and Management Service Processes: Keeping Customers for Life, Addison Wesley 1996.

Tabelle 4-3: *Analysierte Beiträge zum Requirements Engineering aus der Dienstleistungsentwicklung*
Quelle: (Berkovich et al. 2011d)

Tabelle 4-4 listet die analysierten Beiträge zum Requirements Engineering aus der Dienstleistungsentwicklung auf.

Ausgewählte Beiträge zur integrierten Entwicklung von PSS
Spath, D.; Demuß, L. (2003): Entwicklung hybrider Produkte – Gestaltung materieller und immaterieller Leistungsbündel. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Hrsg.: Bullinger, H.-J.; Scheer, A.-W., Springer York 2003.
Botta, C. (2007): Rahmenkonzept zur Entwicklung von Product-Service Systems, Eul-Verlag 2007.
Lindahl, M.; Sundin, E.; Sakao, T.; Shimomura, Y. (2007): Integrated Product and Service Engineering versus Design for Environment – A Comparison and Evaluation of Advantages and Disadvantages. In: Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses. Hrsg.: Takata, S.; Umeda, Y. Springer London, 2007, S. 137-142.
Böhm, T.; Langer, P.; Schermann, M. (2008): Systematische Überführung von kundenspezifischen IT-Lösungen in integrierte Produkt-Dienstleistungsbausteine mit der SCORE-Methode. In: Wirtschaftsinformatik, Vol. 50 (2008) Nr. 3.
Aurich, J.C.; Schweitzer, E.; Fuchs, C. (2007): Life Cycle Management of Industrial Product-Service Systems. In Takata, S.; Umeda, Y. (Eds.), <i>Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses: Proceedings of the 14th CIRP Conference on Life Cycle Engineering, Waseda University, Tokyo, Japan</i> , (pp. 171-176): Springer.

Tabelle 4-4: *Analysierte Beiträge zum Requirements Engineering in den integrierten Ansätzen zur integrierten Entwicklung von PSS*
Quelle: (Berkovich et al. 2011d)

4.4.1.3 Verwandte Arbeiten über die Literaturanalyse zum Requirements Engineering

Verwandte Arbeiten zum Thema des Requirements Engineering für PSS lassen sich in zwei Bereiche aufteilen (Berkovich et al. 2011d):

- Analysen und Literaturreviews zum Requirements Engineering in den einzelnen Domänen, ohne auf die PSS Thematik einzugehen.
- Recherchen und Analysen zu Vorgehensmodellen für die Entwicklung von PSS.

Zum ersten Bereich gehören Beiträge, die sich der Analyse bestehender Literatur zum Requirements Engineering widmen. Byrd et al. (1992) haben Techniken aus den Bereichen der Anforderungsermittlung und der Wissensakquisition analysiert und auf mögliche Gemeinsamkeiten und Unterschiede untersucht. Wieringa et al. (2006) haben ein Klassifikationsschema für Forschungsbeiträge im Bereich des Requirements Engineering entwickelt. Diese Klassifikation ist aber nicht darauf ausgerichtet, verschiedene Themenbereiche im Requirements Engineering abzudecken, sondern überwiegend darauf, verschiedene Arten von wissenschaftlichen Beiträgen zu entdecken. Hickey und Davis (2003) haben anhand von Interviews Kriterien gesammelt, die von Requirements Engineering Analysten für die Auswahl von Anforderungsermittlungstechniken verwendet wurden. Der Fokus liegt daher bei den Auswahlkriterien und weniger bei der Darstellung verschiedener Techniken.

Dem zweiten Bereich sind Beiträge zuzuordnen, die sich mit der Literaturanalyse der Vorgehensmodelle für PSS beschäftigen. So beschreiben Gräßle et al. (2010) Vorgehensmodelle zur Entwicklung von PSS, wobei der Aufbau und die Struktur der Vorgehensmodelle selbst analysiert werden. Kriterien sind also z.B. die Hierarchisierung des Vorgehensmodells oder der Komplexitätsgrad des Vorgehensmodells. Langer et al. (2010) analysieren ebenso Vorgehensmodelle zur Entwicklung von PSS und konzentrieren sich dabei auf den Grad der Abdeckung einzelner Lebenszyklusphasen durch die Vorgehensmodelle, sowie auf die Abdeckung der verschiedenen an der Entwicklung von PSS beteiligten Domänen. Das Requirements Engineering wird in den beiden Beiträgen nicht dediziert behandelt und nur am Rande in Form einer Lebenszyklusphase betrachtet. Aus den vorliegenden Beiträgen können keine Vorgaben zur Gestaltung der Aktivitäten von Requirements Engineering für PSS abgeleitet werden.

4.4.1.4 Ordnungsrahmen zur Darstellung des Requirements Engineering - Gegenstandes für Product Service Systems

Um die Kriterien zur Analyse von Requirements Engineering Ansätzen in der Produkt-, Software und Dienstleistungsentwicklung sowie in den integrierten Ansätzen zur Entwicklung von PSS abzuleiten, wird ein Ordnungsrahmen konstruiert, der die Aufgabe hat den Gegenstandsbereich des Requirements Engineering darzustellen und das Hintergrundwissen und die Annahmen der Forscher explizit zu machen (vgl. Goeken und Patas (2010)). Dies ist im Bereich des Requirements Engineering insbesondere erforderlich, weil dort auf Theorien zur Stützung der Forschungsarbeit weitgehend nicht zurückgegriffen werden kann. Ein Ordnungsrahmen ist ein spezielles Modell, „dessen Konstruktion Verzeichnisbereiche liefert, die der Zuordnung anderer Modelle dienen“ (Brocke 2003, 128). Somit wird der Ordnungsrahmen im Sinne eines Abstraktionskonzepts verwendet und ermöglicht eine abstrahierte Sicht auf ein Problem und schafft somit ein gemeinsames Verständnis des vorliegenden Problems (Dangelmaier et al. 2008).

Zur Konstruktion des Ordnungsrahmens wurde an die Konzepte eines architektonischen Ordnungsrahmens im IT-Bereich orientiert, der von Vogel et al. (2009) vorgeschlagen wurde. Die

Konzepte des architektonischen Ordnungsrahmens sind abstrakt gehalten und eignen sich deswegen gut, um auch im Bereich des Requirements Engineering eingesetzt zu werden.

Ein Ordnungsrahmen muss die folgenden Anforderungen erfüllen (Goeken/Patas 2010):

- (1) Der Ordnungsrahmen muss die wesentlichen Komponenten des Requirements Engineering für PSS wiedergeben.
- (2) Der Ordnungsrahmen muss alle relevanten Beziehungen zwischen diesen Komponenten beschreiben.
- (3) Die Beziehungen zwischen den Komponenten müssen im Ordnungsrahmen wissenschaftlich begründbar sein.

Durch die Erfüllung dieser Anforderungen wird gewährleistet, dass der Ordnungsrahmen dazu geeignet ist bestehendes Wissen zu strukturieren. Um dies zu erreichen, muss in der Definition des Ordnungsrahmens ein geeignetes – hinreichend hohes – Abstraktionsniveau gewählt werden. Auf diese Weise lassen sich verschiedene Forschungsergebnisse, die sowohl unterschiedliche Terminologie als auch Methoden verwenden, strukturieren.

Der Ordnungsrahmen wird aus bestehenden Inhalten und Forschungsergebnissen abgeleitet, wie von Goeken und Patas (2010) empfohlen wird. Dabei wird auf existierende Ansätze und Konzepte zurückgegriffen. Auf diese Weise wird eine bestehende Übereinstimmung bezüglich des Requirements Engineering angenommen und in expliziter Form im Ordnungsrahmen festgehalten. Der Ordnungsrahmen stellt jedoch nicht den Anspruch an Vollständigkeit aufgrund möglicher Erweiterungen und Veränderungen durch zukünftige Forschungsergebnisse. Der Ordnungsrahmen integriert die Eigenschaften von PSS (beschrieben in Kapitel 3.1.2) und bringt sie in ein Zusammenspiel mit den Aktivitäten des Requirements Engineering im Lebenszyklus von PSS (beschrieben in Kapitel 4.3). Dadurch wird aus dem Ordnungsrahmen ersichtlich, wie die Eigenschaften von PSS die Aktivitäten des Requirements Engineering beeinflussen. Somit liefert der Ordnungsrahmen die Grundlage zum Aufstellen von Kriterien zur Analyse von Requirements Engineering Ansätzen (Berkovich et al. 2011d).

In Abbildung 4-6 wird der Ordnungsrahmen zur Darstellung des Requirements Engineering - Gegenstandes für PSS mittels UML dargestellt.

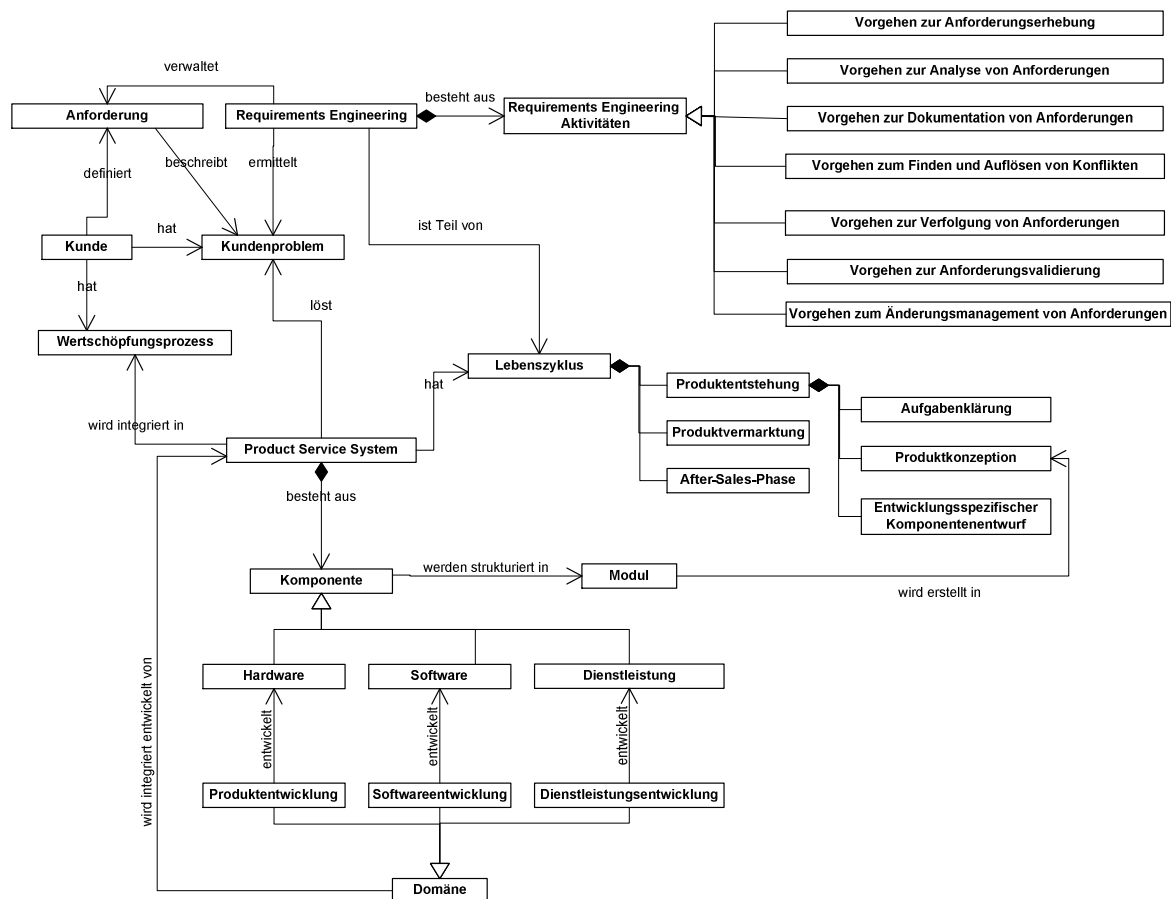


Abbildung 4-6: Ordnungsrahmen zur Darstellung des Requirements Engineering - Gegenstandes für Product Service Systems

Quelle: (Berkovich et al. 2011d)

Im Folgenden werden die wesentlichen Inhalte der Abbildung 4-6 in Prosa beschrieben. Das *Requirements Engineering* hat die Aufgabe das *Kundenproblem* zu verstehen und die sich dadurch ergebenden *Anforderungen* an das *PSS* zu erheben und zu verwalten. In diesem Zusammenhang sollen nicht nur die Anforderungen des *Kunden*, sondern auch die des *Wertschöpfungsprozesses*, in den das zukünftige *PSS* integriert wird, erhoben werden. Um die Kriterien abzuleiten, muss die Natur des *PSS* berücksichtigt werden. So besteht ein *PSS* (was durch eine Route dargestellt wird) aus verschiedenen *Komponenten*, die durch die einzelnen *Domänen* entwickelt werden. Der *Lebenszyklus* von *PSS* besteht aus drei *Phasen*, wobei die erste Phase wiederum aus drei weiteren Sub-Phasen besteht, die substantiell sind, um die *Aktivitäten* des *Requirements Engineering* bei *PSS* zu verstehen. Die *Requirements Engineering* Aktivitäten, die in Kapitel 4.3 beschrieben wurden, werden ebenso im Diagramm dargestellt.

Aus dem Ordnungsrahmen ergibt sich sachlogisch das Zusammenspiel zwischen den *Requirements Engineering* Aktivitäten, den grundlegenden Eigenschaften von *PSS* und dem *Lebenszyklus* von *PSS*. Darauf basierend werden die folgend dargestellten Kriterien für die Analyse der führenden Quellen zum *Requirements Engineering* in der Produkt-, Dienstleistungs- und Softwareentwicklung dargelegt.

4.4.1.5 Kriterien für die Literaturanalyse und Bewertung der Literatur

Die Kriterien für die Analyse und Bewertung der Vorgehensmodelle, die Requirements Engineering behandeln, sowie der führenden Requirements Engineering-Ansätze aus Produkt-, Software- und Dienstleistungsentwicklung und aus der integrierten Entwicklung von PSS bzgl. ihrer Eignung für PSS basieren auf den Eigenschaften von PSS und dem Beitrag von Requirements Engineering im Lebenszyklus von PSS (Kapitel 4.3), deren Inhalte in die Erstellung des Ordnungsrahmens eingegangen sind. Daher wird in der Beschreibung und Begründung für die Kriterien Bezug auf die Begriffe des Ordnungsrahmens gegeben.

Die Literaturanalyse beschreibt nicht einzelne Methoden oder Techniken, sondern untersucht komplette Vorgehen zum Requirements Engineering.

Folgende Kriterien wurden erarbeitet (Berkovich et al. 2011d):

- K1. Systematische Durchführung des Requirements Engineering.** Im Fokus dieses Kriteriums steht die Frage, ob die Verwaltung von Anforderungen systematisch und während des gesamten Lebenszyklus des PSS durchgeführt wird.
- K2. Bereitstellung von Vorgehen zur Anforderungsermittlung.** Im Falle von PSS äußert der Kunde Wünsche an eine Lösung als Ganzes. Die Vorgehen der Anforderungserhebung sollen in der Lage sein, die relevanten Anforderungsquellen zu identifizieren und die Anforderungen möglichst vollständig zu erheben. Im Rahmen der Anforderungsermittlung soll berücksichtigt werden, dass auch die Anforderungen der Wertschöpfungsprozesse des Kunden an das zu entwickelnde PSS zu erheben sind. Darüber hinaus äußern die Kunden und Stakeholder die Anforderungen, die sich auf die Dienstleistungen beziehen, oft implizit. Die Vorgehen der Anforderungsermittlung sollen in der Lage sein, auch implizit ausgedrückte Anforderungen zu erheben.
- K3. Bereitstellung von Vorgehen zur Analyse von Anforderungen.** Die in der Anforderungsermittlung erhobenen Anforderungen müssen analysiert werden, indem sie konkretisiert, detaillierte qualitative und quantitative Eigenschaften erhalten und somit in die „Sprache der Entwickler“ übersetzt werden. Sie sollen in den Lösungsanforderungen an die einzelnen domänenspezifischen Komponenten des PSS resultieren, die anschließend von den Domänen in integrierter und abgestimmter Weise umgesetzt werden. Demzufolge sind die Anforderungen so zu analysieren, dass die daraus abgeleiteten Lösungsanforderungen den einzelnen Domänen zugeordnet werden können. So müssen aus den initialen Anforderungen durch die Analyse die Anforderungen an die Dienstleistungen, die durch drei Dimensionen Potenzial, Prozess und Ergebnis (vgl. (Husen 2007)) charakterisiert sind, sowie an das technische Produkt und an die Software abgeleitet werden. Ebenso sollen die Vorgehen der Anforderungsanalyse in der Lage sein, die Anforderungen auf Konsistenz und Vollständigkeit zu überprüfen sowie die Anforderungen zu priorisieren, um ihre Wichtigkeit für die Stakeholder und die Entwickler dadurch hervorzuheben.
- K4. Bereitstellung von Vorgehen zum Finden und Auflösen von Konflikten zwischen Anforderungen (Anforderungsvereinbarung).** Die Anforderungen können einander widersprechen. Es müssen Vorgehen zur Identifikation und Auflösung von Konflikten zwischen den Anforderungen bereitgestellt werden. Beim Finden

und Auflösen von Konflikten ist zu beachten, dass Konflikte auch zwischen den Anforderungen, die zu verschiedenen Domänen gehören, auftreten können.

- K5. Bereitstellung von Vorgehen zur Dokumentation von Anforderungen.** Es müssen Vorgehen bereitgestellt werden, die eine vollständige, durchgängige und möglichst eindeutige Dokumentation von Anforderungen und ihren Änderungen ermöglichen. Die Anforderungsdokumentation dient als Basis für die Kommunikation zwischen allen Domänen, daher sind die Eigenheiten der einzelnen Domänen zu berücksichtigen. Die domänenspezifischen Informationen, die zum Verständnis von Anforderungen oder zur Durchführung weiterer Aktivitäten des Requirements Engineering benötigt werden, sind ebenso zu dokumentieren. So müssen die Quellen einer Anforderung, die von ihr durch die Analyse abgeleiteten Anforderungen sowie die Beziehungen zu den Anforderungen innerhalb der Domäne als auch zu den Anforderungen anderer Domänen mit dokumentiert werden.
- K6. Bereitstellung von Vorgehen zur Verfolgung von Anforderungen.** Die fortschreitende Projektarbeit, neue Erkenntnisse oder Änderungen der Kundenwünsche sind Gründe für Änderungen an Anforderungen. Deswegen sind die Anforderungen zu kontrollieren und zu verfolgen, um den aktuellen Stand der Anforderungen zu jedem Zeitpunkt der Entwicklung und Nutzung festzustellen. Die Vorgehen sollen in der Lage sein, die Anforderungen von ihrem Ursprung und bis zu ihrer Übermittlung in die domänenspezifischen Lösungsanforderungen zu verfolgen sowie die Interdependenzen zwischen den Anforderungen innerhalb einer Domäne und zwischen den Anforderungen unterschiedlicher Domänen zu identifizieren und festzuhalten.
- K7. Bereitstellung von Vorgehen zur Behandlung von Änderungen an Anforderungen.** Es müssen Vorgehen bereitgestellt werden, um die Auswirkungen von Anforderungsänderungen auf andere Anforderungen sowie auf Entwicklungsergebnisse von Komponenten zu analysieren. Bei PSS müssen insbesondere Änderungen während der Nutzungsphase berücksichtigt werden. Darüber hinaus soll es möglich sein, die Auswirkungen von Änderungen von Anforderungen sowohl auf initiale Anforderungen als auch auf die analysierten Lösungsanforderungen unterschiedlicher Domänen zu überprüfen.
- K8. Bereitstellung von Vorgehen zur Anforderungvalidierung.** Es sind Vorgehen bereitzustellen, die ermöglichen analysierte Lösungsanforderungen, die sich auf die Komponenten der einzelnen Domänen beziehen, auf Übereinstimmung mit initialen Anforderungen zu prüfen. Dies ist insbesondere wichtig, weil die Lösungsanforderungen „in den Sprachen“ der jeweiligen Domänen ausgedrückt sind, wodurch es schwierig ist zu überprüfen inwieweit diese Anforderungen den initialen Anforderungen an das PSS entsprechen und somit die Ziele des Kunden und des Auftragnehmers erfüllen.
- K9. Einbeziehung von Kunden und Stakeholder ins Requirements Engineering.** Kunden und Stakeholder sind die wichtigsten Anforderungsquellen und von ihrer Entscheidung hängt der Erfolg eines Produktes ab (Berkovich et al. 2009b; Meier et al. 2006). Daher sind Kunden und Stakeholder in den Prozess des Requirements Engineering einzubeziehen. Dadurch kann erreicht werden, dass die Realisierung von Anforderungen, die schrittweise konkretisiert werden, von den Kunden und

Stakeholdern eine Zustimmung erhalten. Die Einbeziehung von Kunden und Stakeholdern in das Requirements Engineering stellt eine möglichst vollständige Erfüllung von Kundenanforderungen erreicht werden.

K10. Unterstützung der Modularisierung durch das Requirements Engineering. Das Bilden von Modulen und deren Wiederverwendung stellt eine wichtige Eigenschaft von hybriden Produkten dar. Das Requirements Engineering muss bereits frühzeitig die Anforderungen in geeigneter Weise aufbereiten, um frühzeitig im Entwicklungsprozess Module zu identifizieren, die wiederverwendet werden können.

4.4.2 Requirements Engineering in Produkt-, Software- und Dienstleistungsentwicklung

4.4.2.1 Requirements Engineering in der Produktentwicklung

Die in Tabelle 4-1 (Kapitel 4.4.1.2) angegebenen Beiträge beschreiben Vorgehensmodelle zur Entwicklung technischer Produkte und ein Vorgehensmodell zum Requirements Engineering technischer Produkte. Die Ermittlung und Verwaltung von Anforderungen werden in der Produktentwicklung im Rahmen der Vorgehensmodelle zur Entwicklung technischer Produkte betrachtet. Um die Rolle des Requirements Engineering in den betrachteten Vorgehensmodellen zu verstehen, werden in diesem Kapitel die einzelnen Vorgehensmodelle kurz erläutert. Anschließend werden die wichtigsten Aspekte bezüglich der Ermittlung und Verwaltung von Anforderungen in den Vorgehensmodellen zusammengefasst (Berkovich et al. 2011d).

4.4.2.1.1 Münchener Vorgehensmodell (Lindemann 2006)

Das Münchner Vorgehensmodell (MVM) von Lindemann (2006) enthält sieben Schritte, genannt Elemente des Vorgehensmodells, die in Form eines Netzwerkes miteinander verbunden sind. Das Modell unterstützt somit das sprunghafte Verhalten der Abläufe im Entwicklungsprozess. Das MVM wird für folgende Zwecke eingesetzt (Lindemann 2006, 45):

- Unterstützung der Planung von Entwicklungsprozessen,
- Orientierungshilfe innerhalb von Prozessen zur Problemlösung,
- Zur Analyse des Vorgehens.

Abbildung 4-7 gibt einen Überblick über das Münchner Vorgehensmodell.

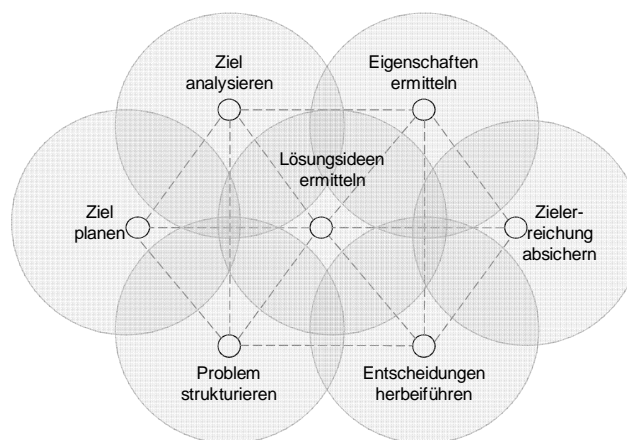


Abbildung 4-7: Das Münchner Vorgehensmodell
Quelle: (Lindemann 2006, 46)

Die sieben Schritte des MVM sind: Ziel planen, Ziel analysieren, Problem strukturieren, Lösungsideen ermitteln, Eigenschaften ermitteln, Entscheidungen herbeiführen, Zielerreichung absichern. Es wird ein Standardweg durch dieses Modell angeboten, allerdings wird darauf hingewiesen, dass im Unterschied zu iterativen Vorgehensmodellen, bei denen zu einer früheren Phase zurückgekehrt werden darf, oder zu linearen Vorgehensmodellen, bei denen kein Zurückkehren erlaubt ist, ein schon einmal bearbeitetes Element nochmals aufgesucht werden kann, um es zu wiederholen und dabei zu konkretisieren (Lindemann 2006, 45ff.).

Desweiteren werden die Elemente des Vorgehensmodells beschrieben, die sich mit dem Requirements Engineering beschäftigen. Das sind Elemente „Ziel planen“, „Ziel analysieren“, „Problem strukturieren“ und „Eigenschaften ermitteln“.

Das Element **„Ziel planen“** ist grundlegend für die darauf folgende Ermittlung von Anforderungen an das technische Produkt. Während dieser Phase werden das Umfeld der zukünftigen Entwicklung analysiert sowie mögliche Einflussfaktoren und Rahmenbedingungen des Entwicklungsprozesses identifiziert und analysiert. Die Einflussfaktoren und Rahmenbedingungen können aus den Bereichen Markt, Kunde, Wettbewerb oder Unternehmen stammen. Darüber hinaus sollen in dieser Phase alternative Zukunfts- und Ergebnismodelle erarbeitet werden. Aus diesen Modellen sind konkrete Maßnahmen und Zielsetzungen in Bezug auf weitere strategische Produkt- und Prozessplanung des Unternehmens abzuleiten.

Das Element **„Ziel analysieren“** beschäftigt sich mit der Klärung der Anforderungen an das zu entwickelnde technische Produkt und der Beschreibung des gewünschten Zielzustandes. Während dieser Phase werden Anforderungen an das technische Produkt ermittelt und in einer geeigneten Form dokumentiert. Es können widersprüchliche Anforderungen vorhanden sein, die zu Konflikten zwischen den Anforderungen führen können. Diese Konflikte sind zu identifizieren und Maßnahmen zur Beseitigung von Konflikten sind einzuleiten. Darüber hinaus sollen die Anforderungen gewichtet werden. Die Gewichtung von Anforderungen soll helfen, den Schwerpunkt der Entwicklung zu bestimmen und dabei die Präferenzen der Kunden in Bezug auf das zu entwickelnde Produkt zu klären. Statt der Gewichtung nach „kann“, „muss“ und „soll“ Anforderungen wird von Lindemann (2006, 103-105) eine numerische Gewichtung vorgeschlagen, um den Grad der Übererfüllung einer Anforderung differenzierter zu betrachten. Anschließend sind die Anforderungen zu dokumentieren.

Nachdem die Anforderungen geklärt sind, werden während der Phase **„Problem strukturieren“** die Schwerpunkte, die für die Entwicklung relevant sind, identifiziert. Dafür soll das Problem auf einem abstrahierten Niveau betrachtet werden, um damit die Informationen über das System, die für die aktuelle Zielsetzung nicht relevant sind, auszugliedern. Anschließend sollen die Anforderungen und die Produktmerkmale miteinander verbunden werden. Die Anforderungen, die der Kunde äußert, werden über mehrere Phasen der Entwicklung immer konkreter und resultieren in den Produktmerkmalen, die die Sicht der Entwickler darstellen. Bei der Übermittlung von Anforderungen an die Entwicklung können Informationen verloren gehen bzw. neu interpretiert werden. Deswegen sollen die Anforderungen und die Produktmerkmale miteinander durch eine Verknüpfungsmatrix verknüpft werden, die die Kundenanforderungen und die technischen Merkmale des Produktes gegenüberstellt.

Eine Eigenschaft eines Produktes setzt sich aus einem Merkmal und einer Ausprägung dieses Merkmals zusammen (Lindemann 2006, 158). Im Rahmen der Anforderungsklä rung wurden die Soll-Werte der Produkteigenschaften ermittelt, die in einer Anforderungsliste zusammengetragen sind. Die Anforderungsliste stellt die Grundlage für die Ermittlung von Ist-Werten

der Produkteigenschaften, die aus Merkmalen und ihren Ausprägungen bestehen. Das Element „**Eigenschaften ermitteln**“ beschäftigt sich mit der Ermittlung von Produkteigenschaften, also von Merkmalen und Ausprägungen. Somit werden die Anforderungen, die die Soll-Eigenschaften des Produktes darstellen, in die „Sprache des Entwicklers“ übersetzt und an die Entwicklung übergeben.

4.4.2.1.2 Vorgehenszyklus für die Lösungssuche (Ehrlenspiel 2002)

Der Vorgehenszyklus von Ehrlenspiel (2002, 81ff.) stellt den Entwicklungsprozess als eine Folge von drei Arbeitsschritten dar, deren Ausführung zu einer Lösung des vorliegenden Problems oder der vorliegenden Aufgabe führt. Die in den ersten Arbeitsschritten steigende Lösungsvielfalt wird in den letzten Schritten stark eingeschränkt und resultiert in der Auswahl einer konkreten Lösung. Die Arbeitsschritte sind miteinander durch Iterationen und Rekursionen verbunden. Folgende drei Arbeitsabschnitte werden im Vorgehenszyklus unterschieden (Ehrlenspiel 2002, 81ff.):

- 1) Aufgabe (Problem) klären,
- 2) Lösung suchen (und darstellen),
- 3) Lösung auswählen.

Abbildung 4-8 zeigt den Vorgehenszyklus für die Lösungssuche nach Ehrlenspiel (2002, 81).

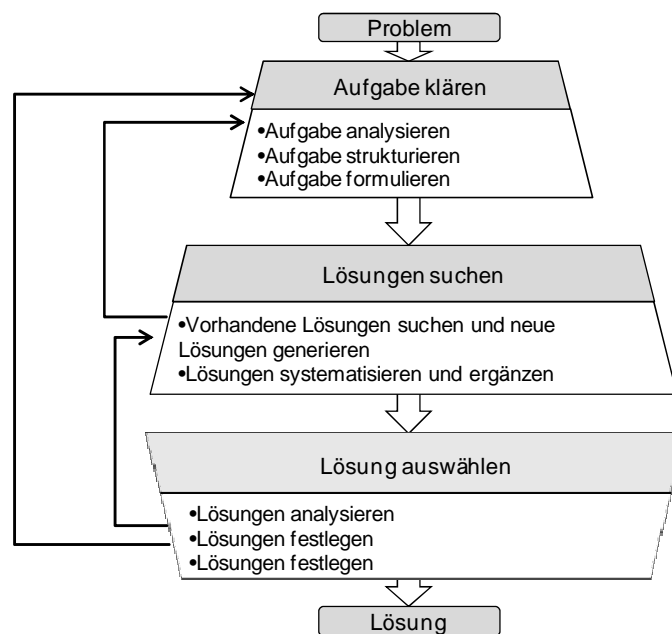


Abbildung 4-8: Vorgehenszyklus für die Lösungssuche
Quelle: (Ehrlenspiel 2002, 83)

Das Ziel des Arbeitsschritts „**Aufgabe klären**“ besteht in der Ermittlung und Strukturierung von Anforderungen an das zu entwickelnde Produkt. Falls es sich nicht um eine Neuentwicklung handelt und ein Produkt schon existiert, sollen Eigenschaften dieses Produktes hinsichtlich der Übereinstimmung mit den neuen Anforderungen analysiert werden, um dadurch den Anpassungsbedarf des Produktes abzuleiten (Ehrlenspiel 2002, 361f.). Die Anforderungen stellen die Soll-Eigenschaften dar, die mit den Ist-Eigenschaften eines Produktes übereinstimmen sollen. Die Summe aller Anforderungen, die an das Produkt gestellt werden, ergibt somit eine abstrakte Beschreibung der Lösung (Ehrlenspiel 2002, 345ff.). Die ermittelten An-

forderungen werden in Anforderungslisten festgehalten, die während des gesamten Entwicklungsprozesses aktuell zu halten sind. Eine wichtige Quelle für die Anforderungen ist der Kunde. Deswegen ist die Kommunikation mit dem Kunden grundlegend für die Entwicklung eines Produktes und soll schon in den frühen Phasen der Entwicklung stattfinden. Bei der Anforderungsermittlung sollen nicht nur die Endkunden, d.h. die Nutzer des Produktes, sondern auch die Zwischenkunden, wie Lieferanten des Produktes an den Endkunden, und die Anforderungen des Produktherstellers berücksichtigt werden.

Das Klären der Aufgabe umfasst mehrere Teilschritte:

- *Aufgabe analysieren*: Die vorliegende Aufgabe muss analysiert werden, um die durch die Aufgabenerfüllung zu erwartenden Ergebnisse festzustellen. Durch die Analyse der Aufgabe werden Anforderungen gewonnen, die die Lösung erfüllen soll. Ebenso sollen widersprüchliche Anforderungen identifiziert und Maßnahmen zur Auflösung von dadurch entstehenden Konflikten zwischen den Anforderungen eingeleitet werden.
- *Aufgabe formulieren*: Die identifizierten Anforderungen und Ziele an die Lösung sind in einer Anforderungsliste zu dokumentieren. Die richtige Formulierung von Zielen ist grundlegend für das Erarbeiten von Lösungen.
- *Aufgabe strukturieren*: Das Ziel der Strukturierung ist die Identifikation der Schwerpunkte einer Aufgabe. Die Anforderungen werden dabei nach Wichtigkeit gewichtet, die Module, bspw. Baugruppen, werden identifiziert sowie die Aufgabe wird in weitere Teilaufgaben zerlegt, die nach organisatorischen Merkmalen, bspw. Bearbeiter, Entwicklungsphasen, organisiert werden.

Die so analysierten und in Anforderungslisten dokumentierten Anforderungen werden zur Lösungssuche verwendet. Bei der Lösungssuche werden schon bestehende Lösungen dahingehend überprüft, wie gut sie die ermittelten Anforderungen erfüllen, oder neue Lösungen durch die Zuordnung von physikalischen Effekten entwickelt, die die Anforderungen erfüllen.

4.4.2.1.3 Methodisches Klären und Präzisieren der Aufgabe (Pahl et al. 2006)

In ihrem Vorgehensmodell orientieren sich Pahl et al. (2006) an der in VDI 2221 (1993) vorgeschlagene Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Die einzelnen Schritte des Vorgehensmodells geben einen sequentiellen Ablauf vor, in dem die schon abgeschlossenen Arbeitsschritte miteinander durch Iterationen verbunden sind. Im Vordergrund der Phase „Planen und Klären der Aufgabe“ stehen die Ermittlung und die anschließende Verwaltung von Anforderungen, die im Rahmen der Phase „Konzipieren“ in geeigneter Form an die Entwicklung übergeben werden. Diese Phasen werden desweiteren näher erläutert. Abbildung 4-9 stellt die einzelnen Schritte der Phase „Planen und Klären der Aufgabe“ dar.

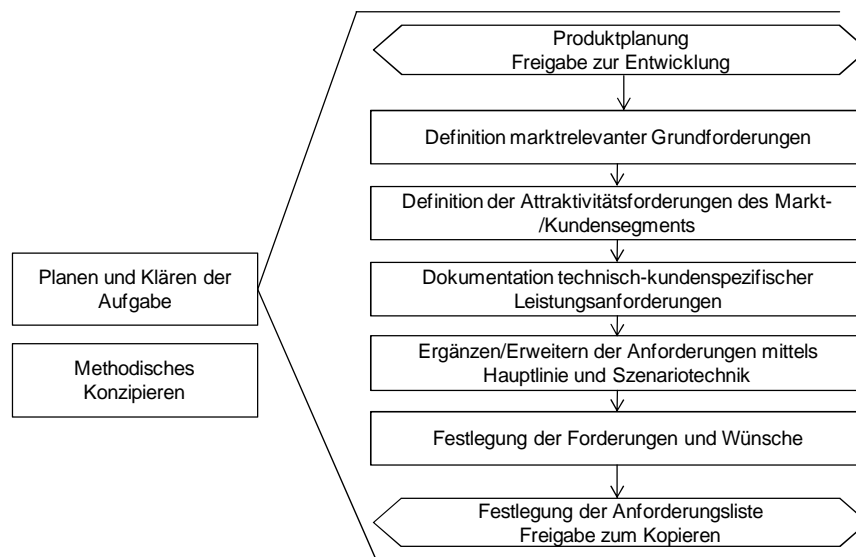


Abbildung 4-9: Planen und Klären der Aufgabe

Quelle: (Pahl et al. 2006, 188)

Das Ergebnis der Phase „**Planen und Klären der Aufgabe**“ ist eine Anforderungsliste, die die Anforderungen an das zu entwickelnde System definiert und die Grundlage für die Spezifikation des Produktes darstellt.

Die Aufgabenstellung wird meist in Form eines Entwicklungsauftrags, einer konkreten Bestellung eines Kunden oder als Verbesserungsvorschläge bzw. Kritik durch Verkauf oder aus einem Konstruktionsbereich bereitgestellt. Die Anforderungen an das Produkt sind aus den Kundenwünschen sowie aus existierenden Dokumenten, Richtlinien, Gesetzen und Vorschriften abzuleiten. Die Anforderungen werden nach Möglichkeit mit quantitativen und qualitativen Angaben versehen und in eine Anforderungsliste einzutragen (Pahl et al. 2006, 187ff.). Eine Anforderungsliste kann nach Hauptmerkmalen, wie Geometrie, Energie bzw. Kräfte, oder nach Teilsystemen, wie Funktions- bzw. Baugruppenstrukturen, gegliedert werden. Änderungen an Anforderungen sind in die Anforderungsliste zu übernehmen, damit die Entwicklung stets aktuelle Informationen zur Verfügung hat.

Die bisher ermittelten Anforderungen werden anschließend erweitert und ergänzt, um impliziten Anforderungen, die von den Stakeholdern nicht direkt geäußert werden, zu erheben. Dafür können Leitlinien mit Hauptmerkmalslisten und die Szenariotechnik eingesetzt werden, die die Gewinnung von Anforderungen mittels Assoziation und Beschreibung der Anwendungsfälle in jeder Phase des Produktlebenszyklus unterstützen. Falls sich Änderungen oder Ergänzungen an Anforderungen ergeben, sind diese in die Anforderungsliste einzutragen. Anschließend sind die Anforderungen in Forderungen, die vom Produkt obligatorisch zu erfüllen sind, und Wünsche, die vom Produkt nach Möglichkeit zu erfüllen sind, zu unterteilen und entsprechende Hinweise sind in die Anforderungsliste einzutragen (Pahl et al. 2006).

Die fertige Anforderungsliste wird in der Phase „**Methodisches Konzipieren**“ als Grundlage für die Entwicklung eines Produktes verwendet. Anhand der Anforderungsliste soll entschieden werden, ob die Aufgabenstellung bereits so geklärt ist, dass die Entwicklung der Lösung begonnen werden kann. Um den Wesenskern des vorliegenden Problems zu identifizieren, soll die Aufgabe abstrahiert und lösungsneutral betrachtet werden. Dafür werden Funktionsstrukturen aufgestellt, die aus Funktionen und den Beziehungen zwischen denen bestehen.

Eine Funktion wird als „lösungsneutrale Formulierung des gewollten (geplanten, bestimmungsgemäßen) Zwecks eines technischen Gebildes“ (Ehrlenspiel 2002, 691) definiert. Zu den Funktionen werden anschließend Wirkprinzipien gesucht, die „den für die Erfüllung einer Funktion erforderlichen physikalischen Effekt sowie die geometrischen und stofflichen Merkmale“ enthalten (Pahl et al. 2006, 227). Darauf basierend werden prinzipielle Lösungsvarianten identifiziert. Wenn eine Lösung ausgewählt wird, wird sie an die Entwicklung weitergegeben.

4.4.2.1.4 Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte (VDI 2221, 1993)

Die VDI 2221 (1993) stellt eine Methodik für ein allgemeingültiges und branchenunabhängiges methodisches Entwickeln und Konstruieren dar. Das Vorgehen besteht aus sieben Arbeitsschritten, die durch iterative Vor- und Rücksprünge miteinander verbunden sind (Abbildung 4-10). Der Methodik liegt die Idee zu Grunde, das Gesamtproblem in Teilprobleme und anschließend in Einzelprobleme zu zerlegen. Nachdem die Einzellösungen erarbeitet sind, werden sie erst zu Teillösungen und anschließend zur Gesamtlösung integriert. Die Methodik versteht sich im Sinne einer Richtlinie ohne konkrete Vorgaben zur Durchführung der Arbeitsschritte oder dazu benötigten Methoden und Techniken.

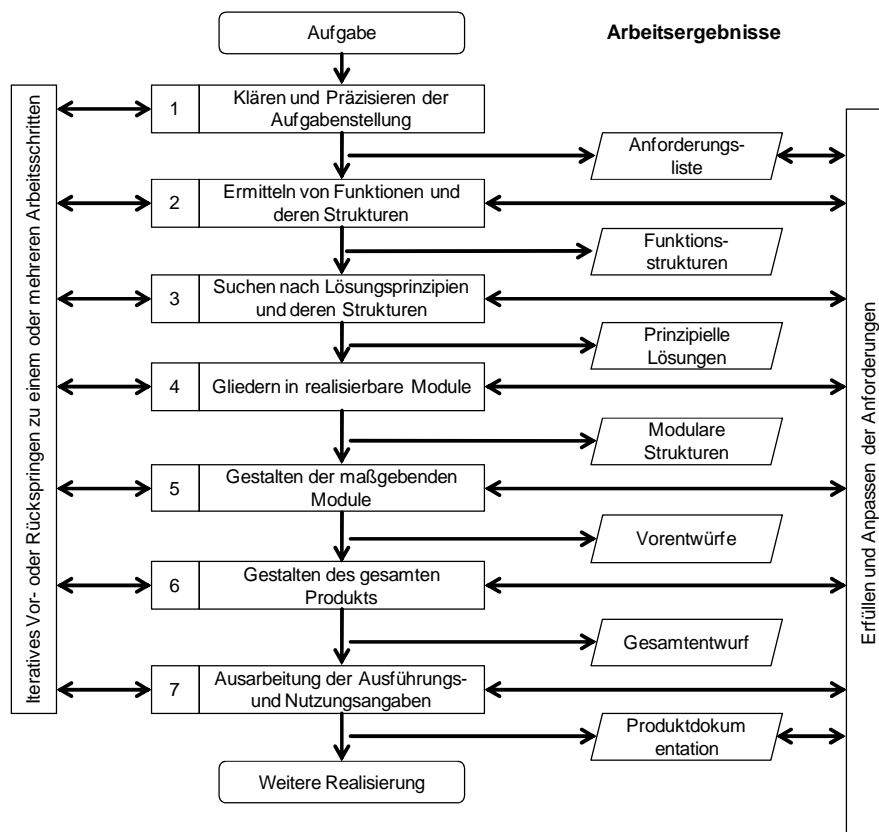


Abbildung 4-10: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte

Quelle: (VDI 2221, 1993)

Zur Analyse der Aktivitäten des Requirements Engineering sind die Schritte „Klären und Präzisieren der Aufgabenstellung“ und „Ermitteln von Funktionen und deren Strukturen“ von Interesse.

Das Ziel der Phase „**Klären und Präzisieren der Aufgabenstellung**“ besteht darin, die Anforderungen, die durch die Aufgabenstellung vorgegeben sind, zu ermitteln und zu klären. Dafür werden alle verfügbaren Informationen, die die Aufgabenstellung beschreiben, gesammelt, die Informationslücken identifiziert, die externen Anforderungen ergänzt und überprüft, sowie unternehmensinterne Anforderungen identifiziert. Anschließend wird die Aufgabenstellung aus der Sicht des Entwicklers formuliert. Das Ergebnis dieser Phase stellt die Anforderungsliste dar, die alle Anforderungen an das zu entwickelnde Produkt enthält. Die Anforderungsliste ist stets aktuell zu halten. Die sich während der Entwicklung ergebenden Änderungen und Ergänzungen der Anforderungen sollen in die Anforderungsliste eingetragen werden. Die Anforderungen sind vollständig, genau und lösungsneutral zu formulieren. Darüber hinaus sollen die Anforderungen gewichtet werden.

Im Arbeitsschritt „**Ermitteln von Funktionen und deren Strukturen**“ werden die Funktionen des zu entwickelnden Produktes, die ein Bindeglied zwischen den Anforderungen und ihrer Umsetzung durch die Entwicklung darstellen, ermittelt. Die Funktionen beschreiben das Verhalten des Systems in seiner Einsatzumgebung. Die Bedingungen, die sich durch die Einsatzumgebung ergeben, bilden die Eingangsgrößen, während die Ausgangsgrößen das geforderte Verhalten des Systems darstellen. Die Funktionen werden basierend auf der Anforderungsliste formuliert. Das Ergebnis dieses Arbeitsabschnittes ist eine oder mehrere Funktionsstrukturen.

In den anschließenden Arbeitsschritten werden den Funktionen der Funktionsstruktur Lösungsprinzipien zugeordnet, die physikalische, chemische oder andere Effekte sowie wirkstrukturelle Festlegungen beschreiben. Für die so entstandenen Wirkstrukturen wird eine Lösung aus mehreren prinzipiellen Lösungen ausgewählt, die anschließend realisiert wird.

4.4.2.1.5 Das Erfassen und Handhaben von Produktanforderungen (Ahrens 2000)

Das Vorgehensmodell von Ahrens (2000) zum Erfassen und Handhaben von Produktanforderungen ist ein iteratives Modell, das auf bestehende Methoden aus der Produktentwicklung zurückgreift.

Ahrens (2000, 11ff.) definiert sechs Dimensionen, die die Anforderungsklärun aktiv beeinflussen und die Inhalte der einzelnen Arbeitsschritte des Vorgehensmodells zum Erfassen und Handhaben von Anforderungen definieren. Das sind Kunden-, Transformations-, Konkretisierungs-, Handhabungs-, Wettbewerbs- und Flexibilitätsdimension:

1. Die Kundendimension beschäftigt sich mit dem Erfassen von Kundenanforderungen bzw. der Marktbedürfnisse, welche bereits in den frühen Phasen der Entwicklung und der Konstruktion zu ermitteln sind. Ein direkter Kontakt zum Kunden erhöht die Qualität der ermittelten Anforderungen.
2. Die Transformationsdimension soll die Kundenanforderungen in technische Anforderungen übersetzen, die an die Entwicklung übergeben werden. Der Kunde formuliert die Anforderungen meistens umgangssprachlich und ausschließlich aus der Anwendersicht. Für die Produktentwicklung sind diese Anforderungen „in die Sprache des Entwicklers“ zu übersetzen, indem sie möglichst detailliert, lösungsneutral und in quantifizierter Form formuliert werden.
3. Im Rahmen der anschließenden Konkretisierungsdimension werden die technischen Anforderungen auf die Vollständigkeit überprüft, sowie präzisiert. Die Präzisierung

von technischen Anforderungen soll in interdisziplinären Teams, die an der Entwicklung von Produkten beteiligt sind, erfolgen.

4. Im Rahmen der Handhabungsdimension geht es um die weitere Nutzung und Verwaltung von Anforderungen während der Entwicklung. Zur Aktivitäten der Verwaltung von Anforderungen gehören die Anforderungsdokumentation, die Gliederung von Anforderungen anhand einer Leitlinie, Funktionsstruktur oder Baustuktur sowie das Änderungsmanagement.
5. Unter Wettbewerbsdimension versteht Ahrens (2000) die Analyse von Wettbewerbsprodukten und ihre Berücksichtigung bei der Anforderungsermittlung.
6. Die Flexibilitätsdimension beschäftigt sich mit der Flexibilität und Anpassbarkeit von Vorgehensmodellen an die Bedingungen der Entwicklung eines Produktes, wie bspw. eine Neuentwicklung oder eine Anpassung eines bestehenden Produktes.

Abbildung 4-11 zeigt das Vorgehensmodell zum Erfassen und Handhaben von Anforderungen (Ahrens 2000, 118).

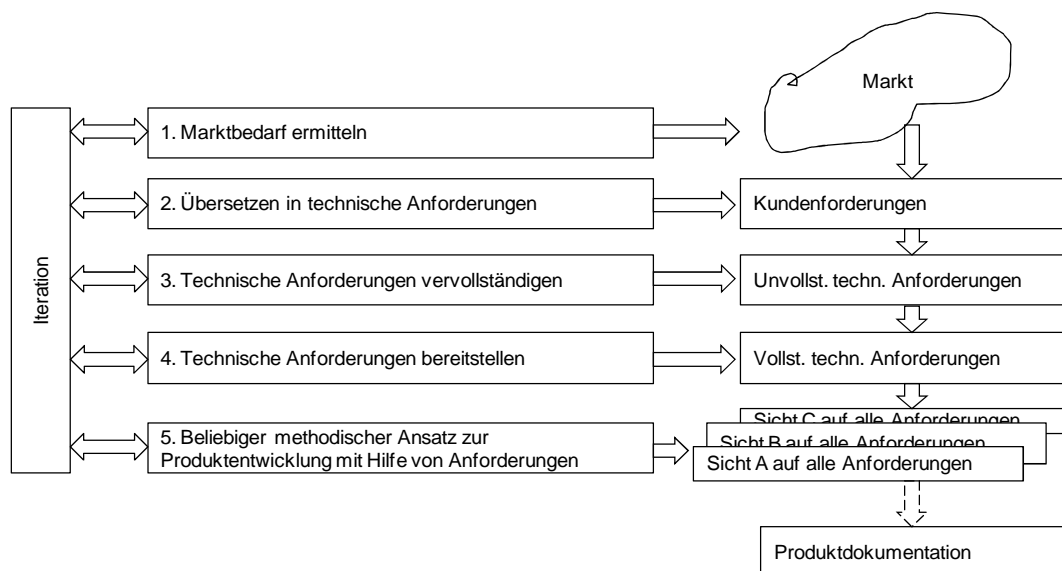


Abbildung 4-11: Das Erfassen und Handhaben von Produktanforderungen

Quelle: In Anlehnung an (Ahrens 2000, 118)

Das Vorgehensmodell bietet vier Arbeitsschritte an, wobei der fünfte Arbeitsschritt einen beliebigen methodischen Ansatz zur Produktentwicklung auf Basis der vorhandenen Anforderungen vorsieht. Die Arbeitsschritte sind iterativ durchzuführen und sind in ihrem Inhalt an die oben beschriebenen Dimensionen angelehnt. So werden im ersten Arbeitsschritt der Marktbedarf und die Kundenanforderungen erhoben, die anschließend in die technischen Anforderungen, die für die Entwicklung relevant sind, übersetzt werden. Im dritten Arbeitsschritt werden die technischen Anforderungen auf Konsistenz und Widersprüchlichkeit geprüft und ergänzt. Im vierten Arbeitsschritt geht es um die Handhabung von Anforderungen, wie ihre Dokumentation oder Behandlung von Anforderungsänderungen.

4.4.2.1.6 Zusammenfassung

Die hier beschriebene Zusammenfassung verschafft einen Gesamtüberblick über die in Kapitel 4.4.2.1 vorgestellten Ansätze zum Requirements Engineering aus der Produktentwicklung

und soll somit allgemeingültige Aspekte bei der Ermittlung und Verwaltung von Anforderungen hervorheben.

Als erster Schritt im Entwicklungsprozess nennen Lindemann (2006), Ehrlenspiel (2002), VDI 2221 (1993) und Pahl et al. (2006) die Analyse des Umfelds der zukünftigen Entwicklung, um mögliche Einflussfaktoren auf den Entwicklungsprozess zu identifizieren und dadurch das Gesamtziel der Entwicklung zu definieren. Darauf basierend werden die Anforderungen an das zu entwickelnde Produkt ermittelt und geklärt. Der Ansatz von Ahrens (2000) sieht ein iteratives Vorgehen zum Erfassen und Handhaben von Produktanforderungen mit der anschließenden Übergabe der ermittelten und konkretisierten Anforderungen an die Entwicklung vor.

Die ermittelten Anforderungen sind zu gewichten, z. B. nach der Wichtigkeit für den Kunden, und zu konkretisieren (Ulrich/Eppinger 2003). Die Anforderungsanalyse, bei der die Anforderungen konkretisiert werden, stellt eins der zentralen Elemente der Anforderungsklä rung und -verwaltung dar, wird aber von den analysierten Konzepten nicht explizit und einheitlich beschrieben. Im Rahmen der Konkretisierung werden die ermittelten Anforderungen in die Sprache des Entwicklers übersetzt und erhalten somit qualitative und quantitative Angaben (Pahl et al. 2006; Ahrens 2000, 144ff.). Als weiterer Schritt zur Übergabe von Anforderungen an die Entwicklung schlagen die meisten Vorgehensmodelle vor, Funktionsstrukturen basierend auf den Anforderungen aufzustellen, die die gesamte Funktionalität des Produktes beschreiben (Lindemann 2006, 115-120; Ehrlenspiel 2002, 369ff.; Pahl et al. 2006; VDI 2221 1993). Für eine erfolgreiche Analyse und Konkretisierung von Anforderungen ist es wichtig, die Experten aus den an der Entwicklung beteiligten Bereichen möglichst früh in den Entwicklungsprozess einzubeziehen (Stechert et al. 2007). In einer empirischen Studie untersuchen Almfelt et al. (2006) die schrittweise Konkretisierung von Anforderungen bei der Entwicklung von Fahrzeugen, beginnend mit den initialen Stakeholderanforderungen über Systemanforderungen, die die Funktionalitäten des Systems beschreiben, Architekturanforderungen, die sich auf die Architektur des Systems beziehen, bis zu den Anforderungen an die Komponenten des Systems. Konkrete Hinweise zur Konkretisierung von Anforderungen fehlen allerdings.

Falls die Anforderungen redundant oder inkonsistent sind bzw. sich widersprechen, spricht man von Konflikten zwischen den Anforderungen. Diese Konflikte sind aufzudecken und Maßnahmen zu ihrer Auflösung sind einzuleiten (Lindemann 2006, 100-102; Ehrlenspiel 2002, 215ff.). Ebenso sollen die Interdependenzen zwischen den Anforderungen identifiziert werden. Somit können die Auswirkungen von Anforderungsänderungen oder Konflikten zwischen den Anforderungen auf andere Anforderungen überprüft und gegebenenfalls entsprechende Maßnahmen zur Behandlung von Änderungen eingeleitet werden (Kruse 1996). Ein weiterer wichtiger Aspekt ist laut allen analysierten Ansätzen die Anforderungsdokumentation, die eine Basis für den gesamten Entwicklungsprozess darstellt und ständig aktuell zu halten ist. Es ist dabei besonders wichtig, dass alle an der Entwicklung Beteiligten die Anforderungslisten aktuell halten (Ehrlenspiel 2002, 359f.; Ahrens 2000, 130f.).

Die betrachteten Ansätze schlagen keine expliziten Maßnahmen zur Bewältigung von Änderungen an Anforderungen vor. Ändernde Kundenwünsche, fortschreitende Technologien, Berücksichtigung der Marktanforderungen sowie zusätzliche Wünsche der Kunden sind die Gründe für die Änderung von Anforderungen (Peterson et al. 2007). Die Änderungen sind im Rahmen der Iterationen zu bewältigen, indem die einzelnen Arbeitsschritte mit der Einbezie-

hung von neuen oder angepassten Informationen wiederholt werden. Darüber hinaus sollen die Änderungen an Anforderungen in der Anforderungsliste dokumentiert werden, um die Verfolgung von Anforderungen zu gewährleisten. Ein weiterer Aspekt des Änderungsmanagements ist die Abschätzung möglicher Änderungen in der Zukunft (Peterson et al. 2007).

Die Erfüllung von Kundenanforderungen ist entscheidend für den Erfolg eines Produktes (Stechert et al. 2007). Um die Kundenzufriedenheit mit dem Produkt zu steigern, ist es wichtig die Rückkopplung vom Kunden in den Produktentwicklungsprozess zu integrieren und somit die Kommunikation mit dem Kunden schon zu Beginn des Entwicklungsprozesses intensiv zu gestalten (Lindemann 2006, 95ff.; Ehrlenspiel 2002, 221-222). Oft beschränkt sich aber der Kontakt zu den Kunden nur auf die anfänglichen Phasen der Entwicklung (Ericson et al. 2007).

Ein weit verbreiteter Ansatz im Zusammenhang mit dem Requirements Engineering ist „Quality Function Deployment“ (QFD) (Hauser/Clausing 1988; Danner 1996), der Kundenanforderungen Produkteigenschaften verknüpft. Das Ziel der Methode ist die Bewertung verschiedener Lösungsmöglichkeiten bezüglich ihrer Erfüllung der Kundenanforderungen und weniger die systematische Ableitung von konkretisierten Anforderungen aus Kundenanforderungen (Ahrens 2000, 122-123; Jung 2006, 100-101).

Die Modellbildung und Simulation spielt eine immer größere Rolle in der Entwicklung (Schäppi et al. 2005, 42ff.). Die Simulation bietet oftmals die Möglichkeit, frühzeitig in der Entwicklung Aussagen über erreichbare Eigenschaften eines Produktes zu treffen. Durch Simulation verschiedener Alternativen können Erkenntnisse gewonnen werden, die auch im Requirements Engineering von Bedeutung sein können (Lindemann 2006, 165). Einerseits werden durch die Modellierung und Simulation Lücken in den Anforderungen entdeckt, andererseits können den Stakeholdern die konkreten Auswirkungen ihrer Anforderungen verdeutlicht werden. Das Thema der Modularisierung tritt in der Produktentwicklung in verschiedenen Formen auf. Das grundsätzliche Prinzip dahinter ist Module und deren Schnittstellen so zu definieren, dass sie in unterschiedlichen Produkten wiederverwendet werden können (Ehrlenspiel 2002, 235; Schäppi et al. 2005, 46ff.). Neu zu entwickelnde Produkte bauen dabei auf bestehenden Vorgaben auf. Diese Vorgaben können in Form von Standard-Designs, Referenzarchitekturen oder Produktfamilien vorliegen. Schäppi et al. (2005, 328) stellen eine Methode zur Beschreibung von Produkten einer Produktfamilie vor. Dabei gibt die Produktfamilie einzelne variable Merkmale vor, die individuell für jedes Produkt festgelegt werden. An dieser Stelle ist auch die Verbindung zum Requirements Engineering zu sehen, die von Schäppi et al. (2005, 328) jedoch nur skizziert wird. Ehrlenspiel (2002, 636ff.) behandelt das Thema der Varianten ebenso ausführlich, jedoch nur als Teil der Entwicklung, ohne die Rolle des Requirements Engineering dabei zu erläutern.

Weitere Vorgehensmodelle zum Requirements Engineering orientieren sich stark an den Vorgehen aus der Softwareentwicklung und unterscheiden dabei die Phasen der Anforderungsermittlung, Anforderungsanalyse und Anforderungsspezifikation (vgl. (Jiao/Chen 2006; Jiao/Tseng 1999; Bulent/Atila 2004; Stechert et al. 2008)). Während der Anforderungsermittlung werden die Anforderungen an das Produkt erhoben, die in der Phase der Anforderungsanalyse analysiert, konkretisiert und somit qualitative und quantitative Angaben erhalten. Anschließend sind die Anforderungen auf mögliche Konflikte und auf ihre Übereinstimmung mit den initialen Anforderungen zu überprüfen und in einer Spezifikation zu dokumentieren. Das

Änderungsmanagement und die Verfolgung von Anforderungen sind Querschnittsaktivitäten, die über alle Phasen des Requirements Engineering durchzuführen sind.

Viele der betrachteten Ansätze beschreiben ein klassisches Vorgehen nach dem Wasserfallmodell, bspw. Pahl et al. (2006). In der Realität werden jedoch häufig iterative Vorgehensmodelle angewendet. Nichtsdestotrotz beschreiben diese klassischen Modelle typische Phasen im Entwicklungsprozess, die unabhängig von der konkreten Ausgestaltung des Prozesses im Wesentlichen dieselben Aufgaben erfüllen. Die Aktivitäten der Anforderungsklärun und -verwaltung werden iterativ durchgeführt und sind in den Entwicklungsprozess zu integrieren.

4.4.2.2 *Requirements Engineering in der Softwareentwicklung*

Im Unterschied zur Produkt- und Dienstleistungsentwicklung stellt Requirements Engineering in der Softwareentwicklung eine eigenständige Disziplin dar. Dieser Aspekt macht das Betrachten der Vorgehensmodelle zur Entwicklung von Software zweitrangig. Daher werden in diesem Kapitel stattdessen Ansätze zum Requirements Engineering aus der Softwareentwicklung analysiert (Berkovich et al. 2011d). Tabelle 4-2 (Kapitel 4.4.1.2) fasst die analysierten Ansätze zusammen.

Der Fokus dieses Kapitels liegt auf den generischen Vorgehensmodellen zum Requirements Engineering und den damit in Zusammenhang stehenden Aktivitäten. In Kapitel 5.2 wird eine Abgrenzung des entwickelten Requirements Engineering Ansatzes für PSS von den existierenden Ansätzen gezeigt, die aber laut der Zielsetzung der vorliegenden Literaturanalyse nicht generisch sind, sondern sich auf bestimmte Entwicklungssituationen beziehen.

Die im Rahmen der Literaturanalyse betrachteten Requirements Engineering Ansätze in der Softwareentwicklung definieren Aktivitäten zur Ermittlung und Verwaltung von Anforderungen. Diese Aktivitäten werden im Folgenden vorgestellt und detailliert beschrieben.

Requirements Engineering ist ein Prozess der Definition relevanter Anforderungen durch die Identifikation aller Stakeholder und ihrer Bedürfnisse und die Dokumentation von Anforderungen in einer Form, die für eine weitere Analyse, Kommunikation und darauf basierende Umsetzung geeignet ist (Nuseibeh/Easterbrook 2000). Das Requirements Management befasst sich mit dem Verwalten der Anforderungen und wird oft als Teilbereich des Requirements Engineering betrachtet (Pohl 2007, 493).

Das Requirements Engineering Rahmenwerk von Pohl (2007, 39ff.) definiert die strukturellen Bestandteile des Requirements Engineering Prozesses, dessen Aktivitäten iterativ-inkrementell durchgeführt werden. Zu den Bestandteilen des Requirements Engineering Prozesses gehören Systemkontext, der die relevante Umgebung für die Definition und das Verständnis von Anforderungen definiert, Kernaktivitäten, die die Ermittlung, Analyse und Dokumentation sowie Vereinbarung von Anforderungen umfassen, und Querschnittsaktivitäten, wie die Validierung, sowie die Management-Aktivitäten, wie die Anforderungsverfolgung, das Änderungsmanagement und die Priorisierung von Anforderungen. Ebenso sind Anforderungsartefakte, die dokumentierte Anforderungen darstellen und zu denen Ziele, Szenarien und lösungsorientierte Anforderungen gehören, Bestandteile des Requirements Engineering Rahmenwerks. Als Ausgangspunkt für den Requirements Engineering Prozess dienen die Vision und der Systemkontext.

Laut der im Rahmen der Literaturrecherche analysierten weiterer Quellen gehören Anforderungsermittlung, Anforderungsanalyse, Anforderungsdokumentation, Anforderungsvalidie-

zung, Anforderungsvereinbarung, Änderungsmanagement und Anforderungsverfolgung zu den wichtigsten Aktivitäten des Requirements Engineering, die im Folgenden erläutert werden.

Während der **Anforderungsermittlung** werden die Anforderungsquellen identifiziert und die Anforderungen erhoben. Die Anforderungsermittlung wird als “learning, uncovering, extracting, surfacing, or discovering needs of customers, users, and other potential stakeholders” definiert (Hickey/Davis 2004). Der erste Schritt bei der Anforderungsermittlung stellt die Kontext- und Systemabgrenzung dar (Pohl 2007, 55-62; Rupp 2004, 101-105). Unter einem Kontext versteht man einen Teil der Umgebung eines Systems, der für die Definition und das Verständnis der Anforderungen an das System relevant ist. Der Kontext unterteilt die Umgebung des zu entwickelnden Systems in relevante und für die Entwicklung des Systems irrelevante Teile. Die Systemgrenze trennt das zu entwickelnde System von seiner Umgebung, die durch den Entwicklungsprozess nicht verändert werden darf (Pohl 2007, 101-105). Die Kontext- und Systemabgrenzung unterstützt die Korrektheit der Definition und Interpretation von Anforderungen durch die Angabe von Kontextinformationen (Beus-Dukic/Alexander 2008).

Die Anforderungsermittlung kann in drei Kernaktivitäten unterteilt werden: (1) Identifikation der Anforderungsquellen im Kontext des geplanten Systems, (2) Entwicklung innovativer Anforderungen und (3) Gewinnung von existierenden Anforderungen (Pohl 2007, 313).

- (1) Zu den Anforderungsquellen gehören Kunden und Stakeholder, existierende Dokumente, wie Lasten- und Pflichtenhefte, Altsysteme, Gesetze, Richtlinien oder Wettbewerb. Als Stakeholder gelten bspw. die Kunden selbst, weitere Personen oder Organisationen, die ein finanzielles Interesse am zu entwickelnden System haben, oder die Entwickler und Tester des Systems (Glinz/Wieringa 2007).
- (2) Innovative Anforderungen sind Anforderungen, die nur in einem kreativen und meist iterativen Prozess mit verschiedenen Stakeholdern entwickelt werden (Pohl 2007, 312). Oft werden zwei Kategorien von Anforderungen unterschieden: funktionale und nicht-funktionale Anforderungen (Sommerville 2004, 119). So beschreiben funktionale Anforderungen den Funktionsumfang eines Systems, also beantworten sie die Frage „Was soll das System tun?“ und beschreiben die Aktionen, die von einem System selbständig ausgeführt werden. Die nicht-funktionalen Anforderungen beschreiben die Eigenschaften des zu entwickelnden Systems. Die beantworten die Frage „Wie gut soll das System etwas tun?“ und stellen somit die Einschränkungen an das zu entwickelnde System dar (Lamsweerde 2009, 23-25; Robertson/Robertson 2007, 9-10).
- (3) Existierende Anforderungen sind Anforderungen, die aus den Anforderungsquellen gewonnen werden können, wie z. B. Anforderungen von Stakeholdern oder Anforderungen aus existierenden Dokumenten. Eine intensive und zielgerichtete Kommunikation mit den Kunden und Stakeholdern ist grundlegend zur Ermittlung von relevanten Anforderungen (Coughlan/Macredie 2002). Es ist dabei erforderlich die Kunden und die Stakeholder zu verstehen und ihre Wünsche über das zu entwickelnde System zu identifizieren (Davis et al. 2006b). So soll der Kunde schon in die ersten Phasen der Entwicklung involviert werden und die Kommunikation zwischen dem Kunden und den Entwicklern soll über die Phasen der Entwicklung bestehen (Ebert 2005, 27). Ebenso soll das Umfeld, in das das zu entwickelnde System eingesetzt wird, analysiert und verstanden werden, um daraus Anforderungen abzuleiten aber auch die Kunden- und Stakeholderanforderungen besser zu interpretieren. Somit ist das Ziel der Anforderungsermittlung die Anforderungen besser zu verstehen und zu interpretieren.

derungsermittlung möglichst viel Wissen über das zu entwickelnde System und sein Umfeld zu gewinnen (Lamsweerde 2009, 61).

Im nächsten Schritt der **Anforderungsanalyse** soll die Brücke zwischen den Kunden- und Stakeholderanforderungen und den Lösungsanforderungen geschaffen werden (Hull et al. 2011, 26; Kotonya/Sommerville 1996, 139). Dabei werden die Anforderungen auf Inkonsistenzen überprüft und anschließend konkretisiert, indem sie detaillierter beschrieben werden, qualitative und quantitative Angaben bekommen und somit von der Entwicklung umgesetzt werden können. Die Kunden- und Stakeholderanforderungen drücken die Sicht der Stakeholder aus und beantworten die Frage „*Was wollen die Kunden und Stakeholder durch das Verwenden des zu entwickelnden Systems erreichen?*“. Diese Anforderungen sind lösungsneutral. Die daraus abgeleiteten Systemanforderungen drücken die Sicht des Requirements Analysten und beantworten die Frage „*Was soll das zu entwickelnde System leisten, um die Anforderungen der Kunden und Stakeholder zu erfüllen?*“ (Hull et al. 2011, 20). Sie repräsentieren Leistungen, die ein System erbringen soll (Gupta/Prakash 2001). Darauf basierend werden die Systemanforderungen weiter konkretisiert und münden zuletzt in den Anforderungen an die Komponenten des Systems (Abbildung 4-12). Ebenso ist es möglich die Anforderungen an die Komponenten weiter auf Sub-Komponenten aufzuteilen und dadurch zu konkretisieren. Die Systemanforderungen und die Komponentenanforderungen gehören zu den Lösungsanforderungen, die den Lösungsraum bilden (s. Kapitel 4.2.2).

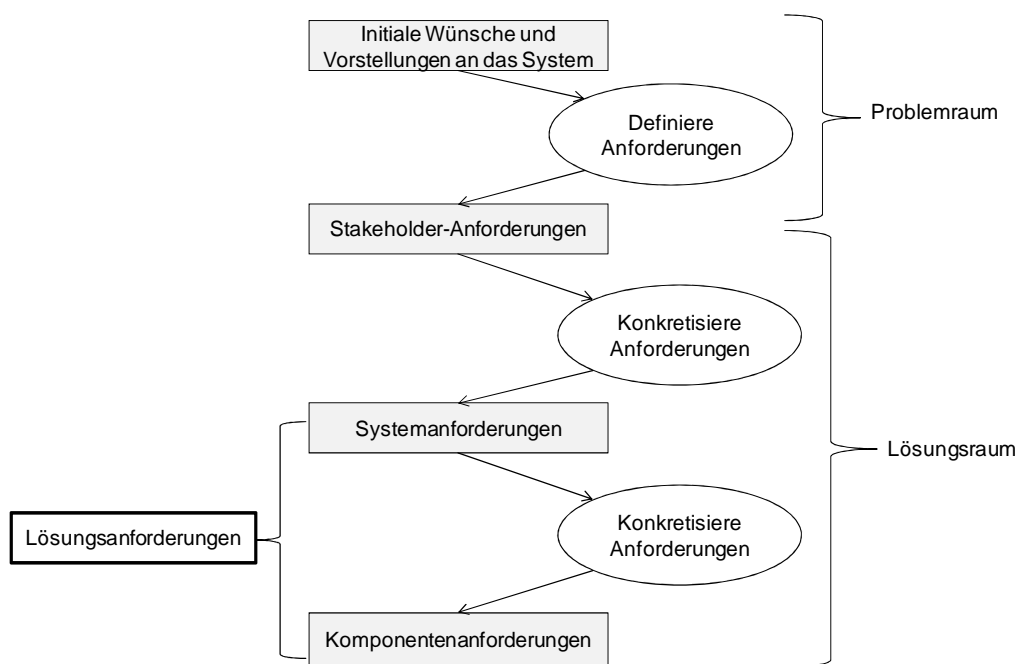


Abbildung 4-12: Konkretisierung von Anforderungen

Quelle: In Anlehnung an (Hull et al. 2011, 28)

Die Modellierung von Anforderungen wird im Rahmen der Anforderungen eingesetzt, um dadurch die verschiedenen Facetten der Anforderungen zu betrachten (Jarke 1998), die Anforderung durch das Hinzufügen von umsetzungsrelevanten Details zu konkretisieren und somit die Transformation von initialen Stakeholder-Anforderungen in die Lösungsanforderungen zu schaffen. Ebenso werden die konzeptionellen Modelle zur Anforderungsdokumentation verwendet (Pohl/Rupp 2009, 67). Dadurch können Lösungsaspekte besser identifiziert und dokumentiert sowie die spezifischen Aspekte des zugrunde liegenden Problems mo-

delliert werden (Pohl/Rupp 2009, 69; Gross/Yu 2001). Darüber hinaus ermöglicht die Modellierung von Anforderungen sich nur auf bestimmte Aspekte des Gegenstandsbereichs zu fokussieren und dadurch die Anforderungen zu konkretisieren, indem nur die Informationen abgebildet werden, die für den jeweiligen Verwendungszweck relevant sind (Pohl 2007, 281; Larsen et al. 2007). Die Modellierung ermöglicht eine detaillierte Sicht auf die Anforderungen sowie eine vereinfachte Sicht auf die Beziehungen zwischen den Anforderungen und trägt somit zur Konkretisierung von Anforderungen bei (Berenbach et al. 2009, 77; Gulla 1996). Im Rahmen der Modellbildung findet die Abbildung von initialen Stakeholder-Anforderungen in einem gedanklichen Bild (Konzeption) des betrachteten Realitätsausschnitts statt. Diese Konzeption wird durch die Verwendung einer geeigneten Modellierungssprache in einem Modell realisiert. Bei einer anschließenden Modellinterpretation werden die Modelle interpretiert, indem ein gedankliches Bild vom zugehörigen Gegenstandsbereich konstruiert wird (Pohl 2007, 294-295).

Desweiteren sollen die Anforderungen im Rahmen der Anforderungsanalyse auf Konsistenz und Vollständigkeit überprüft werden. Die Anforderungen dürfen keine widersprüchliche Angaben sowie keine Angaben, die Missverständnisse hervorrufen, enthalten (Kotonya/Sommerville 1998, 59; Darke/Shanks 1996).

Die Anforderungsanalyse dient auch dazu, die Priorität einer Anforderung festzulegen. Die Priorität einer Anforderung gibt ihre Bedeutung in Bezug auf die Umsetzung an, wobei die Priorität bspw. durch die Wichtigkeit für den Kunden, Kosten oder Risikograd der Umsetzung bestimmt werden kann (Pohl 2007, 525). Darüber hinaus hilft die Priorisierung in Bezug auf die Abschätzung der Kundenzufriedenheit (Berander/Andrews 2005, 70-72). Eine gängige Form der Priorisierung ist die Unterteilung von Anforderungen nach Mandatory-Anforderungen, die auf jeden Fall erfüllt werden müssen, Optional-Anforderungen, die zwar wichtig sind, aber bei denen es wünschenswert wäre sie zu erfüllen, Nice-to-have-Anforderungen, deren Fehlen im System keinen Einfluss auf den Erfolg des Systems hat (vgl. (Pohl 2007, 531)).

Im Rahmen der **Anforderungsdokumentation** sind alle wichtigen Informationen über die Anforderungen zu dokumentieren, wie z. B. die Beschreibungen, Änderungen an Anforderungen, Prioritäten oder Verantwortlichkeiten (Pohl 2007, 215ff.). Ebenso sollen wichtige Informationen, die zur Durchführung anderer Aktivitäten des Requirements Engineering benötigt werden, wie die Liste aller Stakeholder oder die relevanten Geschäftsprozesse dokumentiert werden. Je besser eine Anforderung formuliert ist, desto größer ist die Wahrscheinlichkeit, dass die Anforderung richtig umgesetzt wird (Versteegen 2002, 15). Die am weitesten verbreitete Form der Dokumentation ist die natürlichsprachliche, welche jedoch den Nachteil der Mehrdeutigkeit aufweist (Gervasi/Zowghi 2005; Pohl 2007, 229-233). Zu den weiteren Formen der Dokumentation gehören die formale und semi-formale Dokumentationsformen (Nuseibeh/Easterbrook 2000). So werden die Anforderungen mit Hilfe von UML-Diagrammen, EPK und Szenarien aufgeschrieben (Rupp 2004, 196-207).

Die Anforderungsdokumente stellen die Grundlage für eine konkrete Planung der Realisierung des Systems, den Entwurf der Systemarchitektur, die Implementierung des Systems, die Entwicklung der Testfälle zur Überprüfung des Systems basierend auf den Anforderungen, das Änderungsmanagement zur Abschätzung des Änderungsaufwandes bei den Änderungen an Anforderungen sowie zur Systemnutzung und Systemwartung zur Analyse von Mängeln in der Nutzung (Pohl/Rupp 2009, 49; Hull et al. 2011, 77-79).

Übliche Formen der Dokumentation im deutschsprachigen Raum stellen Lasten- und Pflichtenhefte dar (Pohl/Rupp 2009, 46-47). Das Lastenheft wird von Auftraggeber erstellt und enthält die Anforderungen der Auftraggeber an das zu entwickelnde System sowie alle Randbedingungen. Ebenso werden die Forderungen bezüglich der Lieferungen und Leistungen an das vom Auftraggeber zu entwickelnde System dokumentiert (Pohl 2007, 232-233). Das Pflichtenheft basiert auf den Informationen des Lastenheftes und beschreibt die vom Auftragnehmer definierten Realisierungsvorgaben bezüglich der Anforderungen des Lastenheftes. Somit enthält das Pflichtenheft die konkretisierten lösungsorientierten Anforderungen und Randbedingungen an das System (Rupp 2004, 505).

Die **Anforderungsvereinbarung** hat die Aufgabe Konflikte zwischen Anforderungen zu identifizieren und aufzulösen (Sommerville 2005, 147; Grünbacher/Seyff 2005). Konflikte erscheinen durch widersprüchliche Anforderungen, die oft auf Grund von verschiedenen Sichten der Stakeholder auf das zu entwickelnde System entstehen (Lamsweerde 2000; Lamsweerde 2009, 14; Cheng/Atlee 2007). Die Suche nach Konflikten soll deswegen auf einer abstrahierten Ebene schon während der Anforderungsermittlung stattfinden, um die grundlegenden Widersprüche aufzudecken (Ahmad 2008). Besonders ist die Kommunikation mit den Stakeholdern entscheidend für die Auflösung von Konflikten, die durch die Priorisierung von Anforderungen entscheiden, welche Anforderungen modifiziert werden dürfen oder auf welche Anforderungen verzichtet werden kann (Boehm et al. 1994). Falls eine Verhandlung zwischen den Konfliktparteien nicht zur Auflösung der Konflikte führt, kann auch eine autoritäre Entscheidung den Konflikt auflösen.

Der Prozess zu Identifikation und Auflösung von Konflikten sieht vor, als erstes die widersprüchlichen Anforderungen zu identifizieren, die Ursachen für Konflikte zu analysieren, um darauf basierend anschließend mehrere Vorschläge zur Auflösung von Konflikten zu erarbeiten und nach der entsprechenden Evaluation dieser Vorschläge eine Möglichkeit zur Auflösung von Konflikten auszuwählen (Lamsweerde 2009, 90; Pohl 2007, 394).

Die **Anforderungsverfolgung** wird als „the ability to describe and follow the life of a requirement in both forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)“ definiert (Gotel/Finkelstein 1994). Sie soll ermöglichen die Anforderungen zu jedem Stand ihres Lebenszyklus auf initiale Anforderungen sowie auf Komponenten des Systems, die sie beeinflussen, abzubilden (Kotonya/Sommerville 1998, 160; Ebert 2005). Darüber hinaus ermöglicht die Anforderungsverfolgung die Interdependenzen zwischen den Anforderungen zu identifizieren und somit die Suche nach Konflikten und das Änderungsmanagement zu unterstützen (Kotonya/Sommerville 1998, 128ff.).

Es werden drei Arten der Anforderungsverfolgung unterschieden (Pohl/Rupp 2009, 136; Kotonya/Sommerville 1998, 128-132):

- Pre-Traceability definiert die Verfolgbarkeitsbeziehungen der Anforderung zu den Artefakten, wie die Quelle der Anforderung oder die Entwicklungsinformationen, die der Anforderung im Projektverlauf vorgelagert sind.
- Post-Traceability definiert die Verfolgbarkeitsbeziehungen der Anforderung zu den Artefakten, wie Komponenten des Systems, Implementierung oder Testfälle, die der Anforderung im Projektverlauf nachgelagert sind.

- Traceability zwischen den Anforderungen definiert die Interdependenzen zwischen den Anforderungen. Beispiele für diese Art der Verfolgung sind Konkretisierung einer Anforderung durch eine andere Anforderung oder die Abhängigkeiten zwischen zwei Anforderungen.

Die Anforderungen ändern sich seit ihrer Identifikation über die Entwicklung hinweg bis zur Nutzungsphase (Berenbach et al. 2009, 195-198). Das **Änderungsmanagement** soll die Änderungen aufnehmen, auf Umsetzbarkeit, dadurch entstehende Kosten sowie Auswirkungen auf weitere Anforderungen überprüfen, für weitere Schritte aufbereiten und entsprechend dokumentieren (Kotonya/Sommerville 1998, 123-127; Sommerville 2004, 143-146). Grundsätzlich werden fünf Arten von Anforderungsänderungen unterschieden: Integration einer neuen Anforderung, Löschen, Ergänzen, Reduktion und Veränderung einer bestehenden Anforderung (Pohl 2007, 543ff.). Die Ursachen für die Änderungen können Fehler im Betrieb, die durch die Anforderungen ausgelöst werden, Veränderungen im Kontext des Systems, die zu den Änderungen an Anforderungen führen, oder Veränderungen in der Nutzung des Systems sein (Pohl 2007, 547-549; Cox et al. 2009). Wenn sich eine Anforderung geändert hat, soll als erstes ein Plan zur Behandlung der aufgetretenen Änderung aufgestellt werden. Anschließend sind die Anforderungen, die von der Änderung betroffen sind, zu identifizieren, entsprechende Änderungen an Anforderungen zu ermitteln, Kosten und Aufwand, die durch die Änderungen entstehen, zu bewerten und Änderungen durchzuführen (Jönsson/Lindvall 2005; Kotonya/Sommerville 1998, 123-127).

Während der **Anforderungvalidierung** sollen die Anforderungen auf inhaltliche Fehler, Mehrdeutigkeiten und Inkonsistenzen überprüft werden (Pohl 2007, 417-420; Jönsson/Lindvall 2005). Das Ziel der Validierung ist ebenfalls die Überprüfung der Lösungsanforderungen auf ihre Übereinstimmung mit den initialen Anforderungen. Boehm (1984) weist darauf hin, dass die Validierung von Anforderungen die Frage „*Am I building the right product?*“ beantworten soll. Laut (Kotonya/Sommerville 1998, 87-90) beinhaltet die Anforderungvalidierung die Prüfung der Anforderungsdokumente auf Erfüllung von Vorgaben und Standards. Es wird also vorrangig überprüft, ob die Anforderungen richtig dokumentiert worden sind und ob sie eindeutig formuliert sind. Zudem werden die Anforderungen nochmals auf Konsistenz und Konflikten untersucht, die während der Analyse übersehen worden waren.

In der Softwareentwicklung ist die Problematik der **Modularisierung und Wiederverwendung** unter dem Begriff Software-Produktlinien bekannt. Eine gezielte Wiederverwendung soll ermöglicht werden, indem eine domänenspezifische Basis an Anwendungen erstellt wird (Käkölä/López 2006). In Käkölä und López (2006, 130ff.) wird ein Überblick über Methoden zum Requirements Engineering für Produktlinien gegeben und ein eigener Ansatz vorgestellt. Dieser Ansatz basiert auf einem definierten Artefaktmodell der Requirements Engineering Spezifikationsergebnisse und definierten Prozessschritten des Requirements Engineering.

Neben den beschriebenen Aktivitäten des Requirements Engineering existieren zahlreiche Vorgehensmodelle, die sich auf bestimmte Problemstellungen beziehen und angeben, in welcher Reihenfolge die bestehenden Vorgehen des Requirements Engineering angewandt werden und welche Besonderheiten dabei zu berücksichtigen sind. So schlägt Macaulay (1996) ein iteratives Vorgehensmodell zum Requirements Engineering vor, während Kotonya und Sommerville (Kotonya/Sommerville 1998) ein iteratives Modell vorschlagen, das die Wiederholung der Requirements Engineering Aktivitäten und somit die Einbeziehung der relevanten Informationen in die Konkretisierung von Anforderungen ermöglicht.

4.4.2.3 *Requirements Engineering in der Dienstleistungsentwicklung*

Im Zentrum der Dienstleistungsentwicklung steht eine systematische Entwicklung und Gestaltung von Dienstleistungen durch die Angabe von Methoden, Vorgehensmodellen und Werkzeugen (Bullinger 1999). In der Literatur finden sich mehrere Ansätze zur Dienstleistungsentwicklung, die auch als Service Design, Service Development und Service Engineering bekannt sind.

4.4.2.3.1 Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen (Bullinger und Schreiner 2003)

Dem Service Engineering Rahmenkonzept zur Entwicklung und Gestaltung innovativer Dienstleistungen nach Bullinger und Schreiner (2003) liegt die Verbindung zwischen den durch die konstruktiven Merkmale von Dienstleistungen aufgebauten Dienstleistungsdimensionen (Potenzial, Prozess, Ergebnis und Markt) und den Service Engineering Dimensionen der Dienstleistungsentwicklung zu Grunde.

Die „Service Engineering Dimensionen“ setzen sich aus einem Vorgehensmodell, das die einzelnen Schritte der Entwicklung von Dienstleistungen definiert, sowie Methoden und Werkzeuge zur Unterstützung der Entwicklungsaktivitäten zusammen. Das Vorgehensmodell umfasst sechs Phasen. In der Startphase werden Ideen für die Dienstleistungen erzeugt. Anschließend werden die Anforderungen der Nachfrager und der Anbieter ermittelt und bewertet. Falls keine geeigneten Ideen gefunden werden, wird die Startphase wiederholt. Im anderen Fall wird mit den Spezifikationen auf der Potenzial-, Prozess-, Ergebnis- und Marktdimension begonnen. Diese Spezifikationen werden in der Konzeptphase zu einer Gesamtspezifikation integriert. Anschließend werden die Ressourcen vorbereitet, die für die Dienstleistungserbringung benötigt werden. Bevor das gesamte Dienstleistungskonzept implementiert wird, wird die Gesamtspezifikation getestet, um mögliche Schwachstellen und Probleme zu identifizieren. Falls Fehler während der Testphase auftreten, sind die Einzelspezifikationen zu überarbeiten. Die einzelnen Phasen des Vorgehensmodells haben flexiblen Charakter, indem die Reihenfolge ihrer Durchführung frei gewählt werden kann. Konkrete Vorschläge zur Gestaltung der einzelnen Phasen werden nicht gegeben.

Die methodische Unterstützung für die Gestaltung der Phasen des Vorgehensmodells ist in Form von Handlungsanweisungen definiert, die vorgeben, welche Aktivitäten zu einer bestimmten Zielerreichung notwendig sind. So können in der Analysephase Kundenbefragungen, Feedbackauswertungen des Außendienstes oder Lead User- Konzepte eingesetzt werden. Das Ziel besteht dabei darin, die Kundenanforderungen zu ermitteln. Die Kapazitätsplanung wird eingesetzt, um den Bedarf an Ressourcen für die Dienstleistungserbringung festzustellen. Die Modellierungstechniken werden verwendet, um die Komponenten des Dienstleistungsergebnisses und die Interaktion zwischen Anbieter und Nachfrager zu entwerfen.

Das Service Engineering Rahmenkonzept basiert auf einer Gegenüberstellung von Dienstleistungsdimensionen (Potenzial-, Prozess-, Markt- und Ergebnisdimension) und Service Engineering-Dimensionen (Vorgehensmodell, Methoden und Werkzeuge), die in Form einer Matrix dargestellt werden. So unterstützt das Rahmenkonzept die Dienstleistungsentwicklung durch eine systematisierte Betrachtung der gesamten Aktivitäten. Schrittweise soll auf den Vorgehensmodell-, Methoden- und Werkzeug-Ebenen entschieden werden, welche Aktivitäten zur Gestaltung der Dienstleistungsdimensionen benötigt werden. Obwohl die Bedeutung der systematischen Anforderungsermittlung und -verwaltung erkannt ist und die entsprechen-

den Aktivitäten im Vorgehensmodell dafür vorgesehen sind, werden keine konkreten Hinweise zur Durchführung der Aktivitäten oder zur methodischen Unterstützung gegeben.

4.4.2.3.2 Key Concepts for New Service Development (Edvardsson und Olsson 1996)

Im Zentrum des Konzepts zur Dienstleistungsentwicklung von Edvardsson und Olsson (1996) steht die Dienstleistungsqualität, die für die Akzeptanz der Dienstleistung durch den Kunden grundlegend ist. Das Angebot an Dienstleistungen soll dem Kunden einen Mehrwert verschaffen, den er als solchen wahrnimmt. Die Kundenorientierung ist in diesem Zusammenhang ein wichtiges Element des Konzeptes, die sich dadurch ausdrückt, dass „it is important to understand and respect the customer's needs, wishes and requirements but not to follow them slavishly“ (Edvardsson/Olsson 1996). Das setzt voraus, dass der Dienstleistungsanbieter die Wünsche und Anforderungen der Kunden gut kennt und versteht. Dafür soll der Kunde in den Dienstleistungsentwicklungsprozess eingebunden werden, um durch eine ständige Kommunikation die Anforderungen mit dem Kunden abzustimmen und somit die Korrektheit bei der Entwicklung von Dienstleistungen sicherzustellen.

Im Konzepts zur Dienstleistungsentwicklung von Edvardsson und Olsson (1996) werden drei grundlegende Elemente unterschieden:

- Service Concept. Dieses Element beschreibt die Kundenanforderungen und –wünsche in detaillierter Weise. Darüber hinaus wird im Rahmen dieses Elements angegeben, in welcher Art und Weise diese Anforderungen und Wünsche zu erfüllen sind.
- Service System. Unter dem Service System versteht man Ressourcen, die benötigt werden, um die Dienstleistungen zu realisieren, wie bspw. die Mitarbeiter des Dienstleistungsanbieters, die Kunden, die physisch-technische Umgebung, die Organisation und Kontrolle.
- Service Process. Service Process umfasst Aktivitäten zur Dienstleistungserbringung.

Zusammenfassend lässt sich feststellen, dass das Konzept von Edvardsson und Olsson (1996) sehr abstrakt ist und, obwohl die Bedeutung der systematischen Anforderungsermittlung und -verwaltung erkannt ist, konkrete Vorgehensweisen dazu sowie eine methodische Unterstützung fehlen weitgehend.

4.4.2.3.3 Service Design (Ramaswamy 1996)

Ramaswamy (1996, 3) definiert eine Dienstleistung als „[...] the business transactions that take place between a donor (service provider) and a receiver (customer) in order to produce an outcome that satisfies the customer“. Ramaswamy (1996) weist darauf hin, dass es nicht möglich ist, Produkte und Dienstleistungen getrennt zu entwickeln.

Einer erfolgreichen Dienstleistungsentwicklung und -erbringung liegen folgende Aspekte zu Grunde (Ramaswamy 1996, 7-11):

- Einbeziehung des Kunden in den Dienstleistungserbringungsprozess,
- Entwicklung des Dienstleistungsangebots in Kooperation mit dem Kunden,
- Entwicklung eines Prozesses zur Dienstleistungsentwicklung und -erbringung, der die Leistungserwartung der Kunden erfüllt,
- Bereitstellung individualisierter Dienstleistungen, die die Kundenanforderungen erfüllen,

- Bereitstellung von Dienstleistungen, die dem Unternehmen einen Gewinn ermöglichen.

Somit ist die Bereitstellung hochqualitativer Dienstleistungen grundlegend für den Erfolg eines Unternehmens. Im Zentrum des Modells von Ramaswamy (1996) stehen Design und Erbringung von Dienstleistungen. Die Anforderungen an die zu erbringenden Dienstleistungen und Performance-Standards sind Vorgaben zur Gestaltung des „Service Design“, das aus vier Elementen besteht:

- *Service Product Design*, das die Gestaltung der materiellen Komponenten der Leistung angibt,
- *Service Facility Design*, das sich auf den Entwurf der physisch wahrnehmbaren Umgebung bezieht, in der die Dienstleistung erbracht wird,
- *Service Operations Process Design* umfasst Aktivitäten im Rahmen der Erbringung bzw. Aufrechterhaltung von Dienstleistungen, die sich auf die Infrastruktur der Dienstleistung beziehen,
- *Customer Service Process Design* bezieht sich auf die Gestaltung der Interaktion zwischen der dienstleistungsanbietenden und der –nachfragenden Einheit.

Der Prozess der Entwicklung und Erbringung von Dienstleistungen besteht aus mehreren Phasen, von denen drei für die Analyse und Bewertung des Requirements Engineering von Interesse sind.

Im Rahmen der Phase „*Defining design attributes*“ werden zuerst die zur Identifikation von Anforderungen an die zu erbringenden Dienstleistungen relevanten Stakeholder identifiziert. Anschließend werden die Kunden- und Stakeholderanforderungen ermittelt und von den Stakeholdern priorisiert. Zur Anforderungsermittlung werden Interviews, Beobachtungen, Marktanalysen, Recherchen und Focus Groups eingesetzt. Die Anforderungen werden konkretisiert und in Lösungsanforderungen umgewandelt, indem den Anforderungen qualitative und quantitative Attribute zugeordnet werden. Dafür wird oft die Methode QFD eingesetzt. Im letzten Schritt werden die Anforderungen dokumentiert.

Im Rahmen der Phase „*Specifying performance standard*“ wird der Performance Level für jedes Attribut der zu erbringenden Dienstleistungen ermittelt. Anschließend soll der Performance Level der Wettbewerber analysiert werden, um darauf basierend die Interdependenzen zwischen der Leistungserfüllung und der Kundenzufriedenheit zu identifizieren. Anschließend wird der Performance Level für jedes Attribut der zu entwickelnden Dienstleistung spezifiziert.

In der Phase „*Generating and evaluating design concepts*“ werden die Kernfunktionen für die Erbringung der Dienstleistung definiert, die die Funktionalitäten der Dienstleistungen angeben. Darauf basierend werden die Aktivitäten zur Gestaltung der Dienstleistungen identifiziert und dokumentiert. In der anschließenden Phase werden diese Aktivitäten umgesetzt.

Zusammenfassend kann festgestellt werden, dass das Konzept von Ramaswamy (1996) der Einbindung des Kunden und der Zufriedenstellung des Kunden großen Wert zumisst. Detaillierte Konzepte zum Requirements Engineering werden jedoch nicht eingeführt.

4.4.2.3.4 Systematische Dienstleistungsentwicklung (Fritzsche 2007)

Fritzsche (2007) definiert folgende Anforderungen an ein Konzept zur systematischen Dienstleistungsentwicklung:

- Das Konzept soll einfach und verständlich sein.
- Das Konzept soll passende und dienstleistungsspezifische Methoden und Werkzeuge zur Unterstützung von Dienstleistungsentwicklungsaktivitäten enthalten.
- Bei der Konzeption von Dienstleistungen sollen die vier Dienstleistungsdimensionen - Potenzial-, Prozess-, Markt- und Ergebnisdimension - berücksichtigt werden.
- Die Aktivitäten im Rahmen des Managements von Dienstleistungsentwicklungsprozessen sollen detailliert beschrieben werden.
- Der Dienstleistungsentwicklungsprozess soll kundenintegrativ und kundenorientiert sein.
- Der Dienstleistungsentwicklungsprozess muss ein integriertes Bewertungsverfahren enthalten.
- Der Entwicklungsprozess ist zu dokumentieren.
- Der Entwicklungsprozess von Dienstleistungen soll Rücksprünge zu den vorhergehenden Phasen erlauben, um Erfahrungen in der Zukunft nutzbar zu machen.
- Der Entwicklungsprozess soll eine Testphase enthalten.
- Den einzelnen Phasen werden konkrete Arbeitsschritte zugeordnet.

Der Prozess der systematischen Dienstleistungsentwicklung besteht aus mehreren Phasen, von denen die Start-, Analyse- und Konzeptionsphase einen direkten Bezug zum Requirements Engineering haben.

Im Rahmen der *Startphase* werden die Ideen für die zu entwickelnden Dienstleistungen identifiziert und bewertet. Als Quellen für die Dienstleistungsideen gelten die Mitarbeiter, das aktuelle Dienstleistungsangebot, Kunden, Wettbewerber sowie technologische Entwicklungen. Zur Ermittlung von Dienstleistungsideen können Brainstorming, Methode 635 oder Mind-Mapping eingesetzt werden. Der Bewertung der Dienstleistungsideen liegen Kriterien, wie bspw. die Umsetzbarkeit der Idee, zu Grunde.

Im Rahmen der *Analysephase* wird als erstes das Umfeld, das die Erbringung der Dienstleistungen beeinflusst, analysiert. Dazu gehören die Markts- und Wettbewerbsanalyse, Technologieanalyse sowie die Analyse der Kundenwünsche. Anschließend werden die Erfolgchancen der verbleibenden Dienstleistungsideen nach folgenden Kriterien bewertet:

- Erfüllung der Kundenanforderungen,
- Ausnutzung von Unternehmenspotentialen,
- Chancen um zum Markterfolg zu führen,
- Wirtschaftlichkeitspotentiale.

Basierend auf den Ergebnissen der Bewertung werden die passenden Dienstleistungskonzepte ausgewählt, konkretisiert und ins Pflichtenheft eingetragen.

In der *Konzeptionsphase* werden die ausgewählten Dienstleistungskonzepte umgesetzt. Basierend auf dem Pflichtenheft werden die Dienstleistungsspezifikationen erstellt, die die zu er-

stellenden Dienstleistungen in vier Dimensionen - Produkt-, Prozess-, Ressourcen- und Marketingdimensionen - beschreiben.

Auch dieser Ansatz legt einen großen Wert auf das Erheben von Kundenwünschen. Jedoch wird davon ausgegangen, dass zu Beginn der Entwicklung der Dienstleistung noch kein Kunde vorhanden ist, sondern dass die Dienstleistung auf dem Markt angeboten wird, um dann Kunden zu akquirieren. Daher werden keine Details zur Requirements Engineering angeboten.

4.4.2.3.5 Anforderungsanalyse für produktbegleitende Dienstleistungen (van Husen 2007)

Das Konzept von van Husen (2007) sieht eine systematische Ermittlung und Aufbereitung von Anforderungen für produktbegleitende Dienstleistungen vor.

Van Husen (2007) sieht die Wichtigkeit des Requirements Engineerings für Dienstleistungen folgendermaßen begründet: Durch die steigende Komplexität des Primärproduktes steigen die Anforderungen an die begleitenden Dienstleistungen. Die daraus folgende Konsequenz ist die ständige Anpassung der angebotenen Dienstleistungen an die Eigenschaften des Primärproduktes, die auch im Zentrum der Entwicklung stehen. Der Kunde äußert oft die Anforderungen an die Dienstleistungen implizit, was auch auf die Immaterialität der Dienstleistungen und dadurch erschwerte Vorstellung des Gegenstandes der Dienstleistungen durch den Kunden zurückzuführen ist. Die Anforderungsanalyse soll die Anforderungen an die Dienstleistungen erfassen, priorisieren und klären, welche Eigenschaften die Dienstleistungen tatsächlich haben muss. Das Konzept der Anforderungsanalyse für produktbegleitende Dienstleistungen wird in den Service Engineering-Prozess integriert und schafft eine vollständige Verbindung zwischen den Dienstleistungen und dem Primärprodukt. Zusätzlich zum Vorgehensmodell zur Anforderungsanalyse wird eine methodische Unterstützung dazu angeboten.

Der erste Schritt im Vorgehensmodell der Anforderungsanalyse für produktbegleitende Dienstleistungen ist die Auswahl der relevanten Dienstleistungsidee, die bzgl. der verfolgten Strategie, der Zielgruppe und des Primärproduktes zu bewerten ist. Darauf basierend werden interne Anforderungen des Auftragnehmers und externe Anforderungen der Kunden erhoben.

Die erhobenen Anforderungen sind in Wünsche und Forderungen zu strukturieren und zu Lösungsanforderungen zu konkretisieren, die für die Entwicklung relevant sind. So werden die Anforderungen durch die Konkretisierung entsprechend den Entwicklungsdimensionen - Potenzial, Prozess und Ergebnis - zugeordnet. Die Ergebnisdimension beschreibt die tangiblen Bestandteile der Leistung, wie Material, Form, Farbe, haptische, gustatorische und olfaktorische Gestaltung, die aber nicht zu den Ressourcen zur Erbringung der Dienstleistung gehören. Diese Anforderungen ergeben sich an das Primärprodukt. Ebenso werden in der Dimension die Ergebnisse der Erbringung der Dienstleistung beschrieben. Die Anforderungen der Prozessdimension beschreiben die Prozessgestaltung, wie die Abfolge von Aktivitäten zur Dienstleistungserbringung. Die Potenzialdimension beschäftigt sich mit den Anforderungen an Ressourcen, die benötigt werden, um eine Dienstleistung zu erbringen. Dazu gehören Human Resources, Facilities, wie Standort und Service-Netzgestaltung, Betriebsmittel, wie Informations- und Kommunikationstechnologien, und das Material zur Dienstleistungserbringung, wie Verbrauchsmaterial und Hilfsstoffe. Zur Übersetzung von initialen Kunden- und Stakeholderanforderungen (Rohanforderungen) in die Lösungsanforderungen empfiehlt Husen (2007) den Einsatz der QFD Methode. Wenn die entwicklungsspezifischen Lösungsan-

forderungen vorliegen, werden den quantifizierbaren Anforderungen Zielwerte zugeordnet. Anschließend werden die Anforderungen in Anforderungskatalogen dokumentiert. Das Ergebnis der Anforderungsanalyse stellt eine Anforderungsliste dar, die die spezifizierten Anforderungen an die Dienstleistungen enthält.

Bevor die Anforderungen an die Konzeption und Entwicklung weitergeleitet werden, sind sie in Bezug auf Konflikte und Unklarheiten zu prüfen. Falls die Prüfung durch ein Team erfolgen soll, dann wird ein Review-Meeting verwendet. Im Rahmen der Anforderungsprüfung sollen unklare Anforderungen neu formuliert werden und fehlende Angaben zu den Anforderungen werden ergänzt.

4.4.2.3.6 Zusammenfassung

Die Zusammenfassung der Ansätze der Dienstleistungsentwicklung in diesem Abschnitt verschafft einen allgemeinen Überblick über das Vorgehen in der Dienstleistungsentwicklung und das Requirements Engineering dabei (Berkovich et al. 2011d).

Zu Beginn der Dienstleistungsentwicklung werden Rahmeninformationen, wie Dienstleistungsideen, Schlüssel-Kunden und Anforderungsquellen identifiziert. Als Anforderungsquellen dienen sowohl der Kunde als auch der Dienstleistungsanbieter (Bullinger/Schreiner 2003). Für produktbegleitende Dienstleistungen soll dabei das Primärprodukt ermittelt werden, auf das sich die Dienstleistungen beziehen (Husen 2007). Anschließend sind Ziele, Chancen und Risiken der neu zu entwickelnden Dienstleistungen zu erkennen (Fritzsche 2007; Husen 2007) sowie die entsprechenden Anforderungen an die Dienstleistungen zu ermitteln. Die Kundenanforderungen sind nach Wünschen und Bedürfnissen zu unterscheiden (Edvardsson/Olsson 1996). Interviews, Beobachtungen, Marktanalysen und Recherchen können zur Anforderungsermittlung eingesetzt werden (Husen 2007; Ramaswamy 1996). Die ermittelten Anforderungen sind zu priorisieren (Ramaswamy 1996), auf mögliche Abhängigkeiten zu überprüfen und entsprechend zu strukturieren (Husen 2007). Darauf basierend sind die Anforderungen zu konkretisieren, indem sie möglichst quantifizierbare Attribute erhalten, die sich auf die weitere Implementierung der Dienstleistungen beziehen (Husen 2007; Ramaswamy 1996). Dadurch werden die Anforderungen „in die Sprache des Entwicklers“ übersetzt. Dieser Prozess kann bspw. durch QFD unterstützt werden (Husen 2007). Die Aufteilung der konkretisierten Anforderungen nach Potenzial-, Prozess-, und Ergebnisdimension, die als Entwicklungsdimensionen bezeichnet werden, ermöglicht die Identifikation des Ergebnisses der Dienstleistungserbringung sowie der dafür benötigten Aktivitäten und Ressourcen (Husen 2007; Bullinger/Schreiner 2003). Dieser Aufteilung liegen die konstruktiven Merkmale von Dienstleistungen zugrunde, die als das Leistungspotenzial des Anbieters, die Immaterialität der Dienstleistung und die Kundenintegration in die Leistungserstellung definiert werden. Anschließend sind die Anforderungen in geeigneter Form zu dokumentieren (Husen 2007).

In der Dienstleistungsentwicklung wurden die Vorteile der Modularisierung und Wiederverwendung auch erkannt. Die Wiederverwendung von nicht differenzierenden Dienstleistungsbestandteilen führt zur Steigerung der Profitabilität (Böhmman et al. 2008). Böhmman und Krcmar (2003) schlagen einen Ansatz zur Entwicklung modularisierter Dienstleistungen vor, der auch die Phasen des Requirements Engineering beinhaltet, jedoch nicht im Detail darauf eingeht.

Die meisten Ansätze bieten Vorgehensmodelle zur Entwicklung von Dienstleistungen, die die Bedeutung einer systematischen Anforderungsermittlung und -verwaltung zwar erkennen, keine genauen Vorgehen sowie keine methodische Unterstützung dafür anbieten. Das Konzept von Husen (2007) stellt eine Anforderungsanalyse für produktbegleitende Dienstleistungen mit einer entsprechenden methodischen Unterstützung dar. Die Eigenschaften des Primärproduktes bleiben aber weitgehend außer Acht. Desweiteren beschreibt er nur die Aktivitäten des Requirements Engineering, geht aber nicht auf die Struktur und Eigenschaften von unterschiedlichen Anforderungen ein.

4.4.2.4 Ansätze zur integrierten Entwicklung von Product Service Systems

Es existieren einige Ansätze zur integrierten Entwicklung von PSS. So unterscheiden Aurich et al. (2007) in ihrem Vorgehensmodell für das PSS-Lebenszyklusmanagement (Life Cycle Management – LCM) vier Phasen. Die erste Phase ist die Organisationsgestaltung, die das Ziel hat entsprechende organisatorische Voraussetzungen zur Einführung des PSS zu schaffen. Dazu sollen sowohl die Ablauf- als auch die Aufbauorganisationsstrukturen betrachtet werden. Die Voraussetzungen zur Standardisierung und Integration der Sach- und Dienstleistungsentwicklungsprozesse sowie die Verteilung von Aufgaben im Unternehmen sollen analysiert werden. In der anschließenden Phase der PSS-Planung werden Ideen für PSS identifiziert, bewertet und ausgewählt, die Kunden- und Anbieterziele maximal erfüllen. Die ausgewählten Ideen werden in der Phase der PSS-Entwicklung umgesetzt, indem die Sach- und Dienstleistungskomponenten integriert und abgestimmt entwickelt werden. Anschließend erfolgt eine Abstimmung des kompletten Leistungsbündels auf die kundenindividuellen Anforderungen sowie die Integration der Lösung in die Wertschöpfungsprozesse des Kunden. Der Anbieter stellt dabei sicher, dass er die Wartung, Reparatur und weitere Dienstleistungen, die zur Gewährleistung der Nutzung erforderlich sind, bereitstellen kann.

Die „hybride Produktentwicklung“ (Spath/Demuß 2003) konzentriert sich auf die Entwicklung von PSS. Dem Konzept liegt die Idee zu Grunde, dass Sach- und Dienstleistungen parallel und abgestimmt von den beteiligten Domänen entwickelt werden (Spath/Demuß 2003). Zu Beginn der Entwicklung entsteht ein integriertes Anforderungsmodell, das alle Anforderungen an das PSS enthält. Dieses Modell soll über alle Phasen der Konzeption und Entwicklung gepflegt werden. Es ist dabei wichtig, die Wechselwirkungen zwischen den Anforderungen an das technische Produkt und an die Dienstleistungen zu berücksichtigen. Allerdings werden von den Autoren keine Unterstützung zur Erstellung, Modifikation und Pflege des Anforderungsmodells gegeben. Basierend auf den gesammelten Anforderungen für die komplette Lösung werden Produktstrukturen für materielle und immaterielle Komponenten von PSS getrennt abgeleitet und strukturiert.

Der von Böhmann et al. (2008) vorgeschlagene Ansatz zur Modularisierung standardisierbarer Lösungen ist speziell auf IT-Dienstleistungen zugeschnitten, die sowohl aus Software/Hardware, als auch aus klassischen Dienstleistungen bestehen. Die Anforderungen an diese Lösungen werden anhand von vier verschiedenen Sichten klassifiziert und anschließend werden die standardisierbaren Teilleistungen identifiziert.

Das Rahmenkonzept zur Entwicklung von PSS von Botta (2007) betrachtet die Anforderungen in Form von geforderten Eigenschaften des PSS als Ausgangsbasis für die Entwicklung. Die Anforderungen stellen dem Kunden die Grundlage zur Qualitätssicherung des fertig entwickelten PSS bereit. Aus den Anforderungen wird mittels Synthese eine Merkmalstruktur

abgeleitet, die das zu entwickelnde PSS darstellt. Darauf basierend werden die Merkmale bezüglich der Ist-Eigenschaften analysiert und bewertet. Die Differenz zwischen den Ist-Eigenschaften und den Merkmalen, die sich aus dem Abgleich ergibt, stellt der Entwicklungsbedarf dar. Nach jeder Iteration wird ein Produktmodell erstellt, das das Ergebnis der Synthese und des Vergleichs bildet. Jede Iteration führt zu einer Annäherung der Ist-Eigenschaften an die Anforderungen. So wird der Entwicklungsprozess so oft wiederholt, bis die Ist-Eigenschaften den Anforderungen hinreichend entsprechen.

Die „Integrated Product and Service Engineering Methodik“ (IPSE) von Lindahl et al. (2007) konzentriert sich auf den gesamten Lebenszyklus eines PSS. Dem Modell liegt die Idee der umweltschonenden Entwicklung zu Grunde. Das Konzept baut auf den Forschungsergebnissen der Produkt-, Software- und Dienstleistungsentwicklung auf. Das IPSE stellt ein Vorgehen bereit, das sowohl die Perspektive des Auftragnehmers als auch des Kunden und der Lieferanten einnimmt. In der ersten Phase „Need- & requirement analysis“ findet die Analyse der Kundenbedürfnisse und -anforderungen statt. Dabei sollen die Funktionalitäten, die der Kunde vom PSS erwartet, erfasst werden. Ebenso sollen weitere Unternehmen, die an der Wertschöpfung beteiligt sind, wie Lieferanten, in Bezug auf die Anforderungen an das PSS als Ganzes aber auch an die einzelnen Komponenten gefragt werden. Basierend auf den Ergebnissen der ersten Phase wird in der zweiten Phase „Concept generation“ ein Angebotskonzept erstellt. Dieses Konzept wird in der nächsten Phase „Check & contract“ dem Kunden zur Bewertung vorgelegt und stellt somit die Grundlage zum Vertragsabschluss dar. In der darauffolgenden Phase „Concept realization“ werden die Sach- und Dienstleistungskomponenten der Lösung erstellt und an den Kunden ausgeliefert. In der Phase „Support & Maintenance“ werde die Wartung und weitere Dienstleistungen angeboten.

Ein weiterer Ansatz ist der Entwicklungsprozess von Thomas et al. (2008), dem die Methoden der Produktentwicklung zu Grunde liegen. Der erste Schritt im Entwicklungsprozess ist die Ermittlung von Anforderungen an das PSS, aus denen die Soll-Eigenschaften und basierend auf den gewonnenen Erkenntnissen, Merkmale des PSS abgeleitet werden. Dennoch kommt Thomas et al. (2008) zum Schluss, dass sich die Methoden der Produktentwicklung nur beschränkt für PSS eignen. Zur Übersetzung von initialen Kunden- und Stakeholderanforderungen in die Lösungsanforderungen an die einzelnen Komponenten des PSS wird empfohlen das QFD einzusetzen (Möhrle/Spilgies 2005). Allerdings erweist sich der Einsatz der Methode bei Neuentwicklungen als schwierig.

Die analysierten Ansätze zur integrierten Entwicklung von PSS beschreiben den kompletten Lebenszyklus von PSS oder fokussieren sich auf den Entwicklungsprozess. Die Notwendigkeit eines systematischen und integrierten Requirements Engineering, das das Verständnis aller beteiligten Domänen fördert, wurde in allen Ansätzen erkannt. Während die Anforderungsermittlung weitgehend detailliert erläutert wird, fehlen bei den weiteren Aktivitäten des Requirements Engineering und insbesondere für die Anforderungsanalyse, Suche und Auflösung von Konflikten sowie Behandlung von Änderungen sowohl inhaltliche als auch methodische Unterstützung (Berkovich et al. 2011d).

4.4.2.5 Ergebnisse der Literaturanalyse

Basierend auf den gewonnenen Erkenntnissen aus der Literaturrecherche und den davor aufgestellten Kriterien (Kapitel 4.4.1.5) sind die betrachteten Ansätze zu analysieren und auf ihre Eignung für PSS zu überprüfen (Berkovich et al. 2011d).

K1. Systematische Durchführung des Requirements Engineering. Sowohl in der Produktentwicklung als auch in der Softwareentwicklung spielt das Requirements Engineering eine zentrale Rolle und wird systematisch durchgeführt. In beiden Domänen existieren zahlreiche Vorgehensmodelle, Techniken und Werkzeuge zur Anforderungsverwaltung, jedoch fehlen in der Produktentwicklung eine allgemeingültige Definition des Requirements Engineering sowie eine klare thematische Abgrenzung zum Entwicklungsprozess. In der Dienstleistungsentwicklung wurde die Bedeutung des Requirements Engineering erkannt, aber die Anforderungsverwaltung und deren Vorgehen sind nicht festgelegt. Oft werden Ansätze aus der Produktentwicklung oder Softwareentwicklung als Grundlage für die Verwaltung der Anforderungen an die Dienstleistungen genommen, ohne aber auf die Spezifika der Dienstleistungen, wie bspw. ihre Immaterialität oder konstruktive Merkmale, im Detail einzugehen. Bei den integrierten Ansätzen zur Entwicklung von PSS bleibt kritisch festzustellen, dass diese Ansätze noch sehr abstrakt und weitgehend ohne empirische Evidenz sind. Die Wichtigkeit des Requirements Engineering wurde von ihnen erkannt, jedoch werden keine konkreten Vorgehen zur Durchführung der Requirements Engineering Aktivitäten angeboten. Ebenso fehlt bei denen eine umfassende Integration des Requirements Engineering in den Entwicklungsprozess und somit eine nahtlose Übergabe von Anforderungen an die Entwicklung.

K2. Bereitstellung von Vorgehen zur Anforderungsermittlung. Die Anforderungsermittlung wird in allen Domänen breit behandelt. Viele Vorgehen werden vorgeschlagen; zu den wichtigsten gehören Interviews, Workshops oder Analyse existierender Dokumente. In den einzelnen Domänen wird aber bei der Anforderungsermittlung stark auf den eigenen Bereich fokussiert. In der Produktentwicklung werden keine Vorgehen zur Erhebung von Anforderungen an die im Zusammenhang mit dem Produkt stehenden Dienstleistungen, die oft nur implizit geäußert werden, vorgeschlagen.

Die integrierten Ansätze erkennen die Notwendigkeit Anforderungen zu erheben, aber konkrete Vorgehen hierzu fehlen weitgehend. Schwächen bestehen in allen Ansätzen in der Ableitung von Anforderungen aus den Wertschöpfungsprozessen des Kunden. Obwohl eine Umfeldanalyse, die dies leisten soll, vorgeschlagen wird, fehlen genaue Angaben zur Identifikation relevanter Wertschöpfungsprozesse und Erhebung von Anforderungen. Ebenso wird in den untersuchten Ansätzen nicht berücksichtigt, dass bei einer domänenübergreifenden Entwicklung die Anforderungsanalysten entsprechendes Wissen aus anderen Domänen erbringen müssen, um richtige Fragen an den Kunden zu stellen.

K3. Bereitstellung von Vorgehen zur Analyse von Anforderungen. In den analysierten Ansätzen der Produktentwicklung und Softwareentwicklung sind die Analyse und die damit in Verbindung stehenden Konkretisierung von Kunden- und Stakeholderanforderungen in Lösungsanforderungen sowie die Priorisierung von Anforderungen ein zentrales Element des Requirements Engineering. Es werden Vorgehen dazu angeboten, die jedoch stark auf die jeweilige Domäne spezialisiert sind. Die integrierten Ansätze erkennen die Notwendigkeit der Konkretisierung von Anforderungen, stellen jedoch keine konkreten Vorgehen zu Verfügung. Weit verbreitet ist die QFD-Methode, die in allen Domänen eingesetzt wird. Der Einsatz dieser Methode erweist sich jedoch als schwierig, wenn es um die Entwicklung neuer Produkte geht, da sie auf einer Gegenüberstellung mit bereits existierenden Produkten beruht. Das QFD wurde auch für PSS adaptiert, dessen Einsatz sich sowohl bei der Neuentwicklung, als auch bei der Ableitung von Lösungsanforderungen aus Kundenanforderungen als schwierig erweist und sich fast ausschließlich zur Bewertung von verschiedenen Lösungsmöglichkeiten bezüglich ihrer Erfüllung der Kundenanforderungen eignet (Möhrle/Spilgies 2005). Allerdings

bleibt die Übersetzung von initialen Anforderungen an das PSS in die Lösungsanforderungen, die den einzelnen Domänen zugeordnet sind, weitgehend ungelöst.

Keines der Vorgehen aus der Produktentwicklung und Softwareentwicklung ist geeignet, um die Gliederung von Dienstleistungen nach Entwicklungsdimensionen zu gewährleisten und dabei das Zusammenspiel zwischen den Anforderungen an das technische Produkt und an die Dienstleistungen abzubilden. Darüber hinaus sind die Ansätze aus Produktentwicklung und Softwareentwicklung bestrebt die Anforderungen möglichst zu quantifizieren, was bei Dienstleistungen aber mit Schwierigkeiten verbunden ist (Husen 2007). Die Ansätze der Dienstleistungsentwicklung selbst beschreiben zwar das Konzept der Anforderungskonkretisierung, konkrete Vorgehen fehlen aber weitgehend. In Bezug auf produktbegleitende Dienstleistungen sind in den Ansätzen der Dienstleistungsentwicklung die Vorgehen zur Identifikation des Primärproduktes, auf das sich die Dienstleistungen beziehen, und der Verbindung zwischen dem Primärprodukt und der Dienstleistung nicht genau beschrieben.

Die Priorisierung von Anforderungen wird von allen Domänen sowie in den Ansätzen zur integrierten Entwicklung von PSS behandelt. Die vorgeschlagenen Vorgehen lassen sich weitgehend auf PSS übertragen.

K4. Bereitstellung von Vorgehen zum Finden und Auflösen von Konflikten zwischen Anforderungen. Das Finden und Auflösen von Konflikten in der Dienstleistungsentwicklung und in den Ansätzen zur integrierten Entwicklung wird kaum behandelt. Lediglich Husen (2007) schlägt ein Vorgehen zur Konfliktidentifikation und -auflösung vor, das aber an die Vorgehen aus der Produktentwicklung und Softwareentwicklung angelehnt ist. Die in der Produktentwicklung, Softwareentwicklung und Dienstleistungsentwicklung angebotenen Vorgehen zur Konfliktidentifikation, wie Einflussmatrizen zum Prüfen der Abhängigkeitsbeziehungen zwischen den Anforderungen, orientieren sich auf die eigenen Domänen, was dazu führt, dass mögliche Konflikte zwischen den Anforderungen unterschiedlicher Domänen, wie bspw. zwischen den Anforderungen an Software und Dienstleistungen, dabei unentdeckt bleiben. Zur Konfliktauflösung werden von den meisten Ansätzen Verhandlungen mit dem Stakeholder unter Einbeziehung der Entwickler vorgeschlagen. Auch in diesem Fall können die Anforderungen an die Dienstleistungen aufgrund ihrer Immaterialität sowie der Entwicklungsdimensionen an die Stakeholder schwer vermittelbar sein.

K5. Bereitstellung von Vorgehen zur Dokumentation von Anforderungen. Die natürlichsprachliche Dokumentation von Anforderungen wird in allen Ansätzen beschrieben. In der Dienstleistungsentwicklung und in den integrierten Ansätzen werden keine genauen Angaben zur Gestaltung der Anforderungsdokumente gemacht. Die Vorteile der natürlichsprachlichen Dokumentation liegen an der Einfachheit und Verständlichkeit. Zu den Nachteilen gehört die Mehrdeutigkeit, auf die besonders in einem domänenübergreifenden Umfeld geachtet werden soll. Das Anforderungsdokument verlangt eine durchgehende Pflege über den gesamten Entwicklungsprozess hinaus, lässt sich aber gut für PSS anwenden.

Die Anwendung der modellbasierten Dokumentation von Anforderungen, die insbesondere in der Softwareentwicklung verbreitet ist, ist auf PSS schwierig anzuwenden. Der Grund dafür ist, dass es weitgehend keine Vorgehen und Modelle zur Darstellung von Anforderungen an die Dienstleistungen sowie für das Zusammenspiel zwischen den Anforderungen, die zu verschiedenen Domänen gehören, existieren. In diesem Fall muss für jede Art der Modellierungstechnik einzeln geprüft werden, in wieweit sie für PSS geeignet ist.

K6. Bereitstellung von Vorgehen zur Verfolgung von Anforderungen. Die Verfolgung von Anforderungen muss zwei Aspekte berücksichtigen. Zum Einen sollen die Abhängigkeiten zwischen den Anforderungen untereinander und den Komponenten des Produktes im Rahmen der Verfolgung identifiziert werden. Zum Anderen soll die Verfolgbarkeit einer Anforderung zu ihrer Quelle gewährleistet werden.

In der Produktentwicklung und Softwareentwicklung wird die Verfolgung von Anforderungen über Einfluss- oder Verknüpfungsmatrizen festgehalten. In der Dienstleistungsentwicklung sowie in den integrierten Ansätzen werden keine konkreten Vorgehen zur Gewährleistung der Verfolgbarkeit vorgegeben. Im Falle von PSS sollen bei der Identifikation der Abhängigkeiten zwischen Anforderungen berücksichtigt werden, dass verschiedene Domänen beteiligt sind. Somit sind die vorgeschlagenen Einfluss- und Verknüpfungsmatrizen an PSS anzupassen, indem die Beteiligung mehrerer Domänen und die drei Entwicklungsdimensionen der Dienstleistungen einbezogen werden und die Matrizen entsprechend mehrdimensional gestaltet werden. Alle Ansätze weisen darauf hin, dass Anforderungsdokumente entsprechende Verweise auf abhängige Anforderungen und Komponenten, Anforderungsquellen sowie Änderungen enthalten sollen.

K7. Bereitstellung von Vorgehen zur Behandlung von Änderungen an Anforderungen. Weder in der Produktentwicklung noch in der Dienstleistungsentwicklung werden die Änderungen der Anforderungen ausführlich behandelt. Die Abschätzung der Auswirkungen von Änderungen auf weitere Anforderungen sowie auf Komponenten des Produktes basiert auf den Informationen, die durch die Anforderungsverfolgung gewonnen werden. In der Dienstleistungsentwicklung und in den integrierten Ansätzen wird darauf hingewiesen, dass sich die Anforderungen ändern können, ein Prozess zur Behandlung von Änderungen wird aber nicht angegeben. Es wird dabei auf die entsprechenden Vorgehen der Softwareentwicklung hingewiesen. Die Änderungen in der Nutzungsphase werden in keiner Domäne behandelt. Ebenso bleibt die Herausforderung, die Auswirkungen der Änderungen von Anforderungen unterschiedlicher Domänen abzuschätzen. So sollen bspw. die Auswirkungen von Änderungen der Softwareanforderungen auf davon abhängigen Dienstleistungsanforderungen unter Berücksichtigung der drei Entwicklungsdimensionen abgeschätzt werden. Alle analysierten Ansätze weisen darauf hin, dass die Änderungen erst dann realisiert werden können, wenn sie von allen Beteiligten akzeptiert sind und keine Konflikte hervorrufen.

K8. Bereitstellung von Vorgehen zur Anforderungvalidierung. In den meisten Ansätzen der Dienstleistungsentwicklung ist die Validierung von Anforderungen ein Teil eines Abgleichs zwischen dem auf Basis der Anforderungen erstellten Dienstleistungskonzept und den initialen Anforderungen. Dabei werden keine konkreten Schritte zur Validierung von Anforderungen vorgegeben. Husen (2007) schlägt vor, die Vorgehen der Softwareentwicklung zu benutzen. Auch in der Produktentwicklung und in den integrierten Ansätzen wird die Validierung unzureichend diskutiert. In der Softwareentwicklung werden dafür Techniken, wie z. B. Inspektionen oder Reviews eingesetzt, die sich darauf ausrichten, Fehler und Inkonsistenzen in Anforderungsdokumenten zu finden und auch für PSS eingesetzt werden können. Gegenüber den analysierten Ansätzen ist zu beachten, dass die beteiligten Domänen in die Validierung einzubeziehen sind. So sollen die vorgeschlagenen Vorgehen, wie Interviews oder Inspektionen, die drei Entwicklungsdimensionen der Dienstleistungen, die Abhängigkeiten zwischen den domänenspezifischen Anforderungen sowie die implizite Äußerung von initialen Anforderungen an die Dienstleistungen berücksichtigen.

K9. Einbeziehung von Kunden und Stakeholder ins Requirements Engineering. In allen Domänen sowie in den Ansätzen zur integrierten Entwicklung wird die Rolle des Kunden hervorgehoben, der über den Erfolg eines Produktes entscheidet. In der Dienstleistungsentwicklung ist der Kunde der Leistungsempfänger und daher in den Entwicklungsprozess einzubeziehen. In den meisten Ansätzen beschränkt sich die Kundenintegration aber auf die Phasen der Ermittlung und Vereinbarung. Auch die Anforderungspriorisierung wird größtenteils durch den Kunden bestimmt. Die Kommunikation mit dem Kunden gestaltet sich häufig als schwierig, denn der Kunde hat oft ein anderes Verständnis über das zu entwickelnde Produkt. Darüber hinaus ist der Entwickler nicht immer in der Lage, die Wünsche und Vorstellungen der Kunden richtig zu interpretieren. Genau diese Aspekte sind bei der Kundenintegration zu berücksichtigen.

K10. Unterstützung der Modularisierung durch das Requirements Engineering. Die Vorteile der Modularisierung und Wiederverwendung von Modulen wurde in allen drei Domänen erkannt. Auch im Bereich der integrierten Entwicklung von PSS liegen erste Arbeiten hierzu vor. Das Thema der Modularisierung wird jedoch größtenteils als eines der dem Requirements Engineering nachgelagerten Entwicklungsschrittes gesehen. Die Integration des Requirements Engineering in die Modularisierung wird daher nur skizziert, jedoch nicht detailliert dargelegt.

Abbildung 4-13 gibt einen Überblick über die analysierten Ansätze, in wieweit die Ansätze die Kriterien erfüllen und entsprechend für das Requirements Engineering für PSS geeignet sind. Dabei kann ein Ansatz ein Kriterium vollständig, teilweise oder nicht erfüllen.

Analysierte Ansätze Kriterien der Literaturanalyse	Ansätze aus der Produktentwicklung	Ansätze aus der Softwareentwicklung	Ansätze aus der Dienstleistungsentwicklung	Ansätze zur integrierten Entwicklung hybrider Produkte
Systematische Durchführung des Requirements Engineering				
Bereitstellung von Vorgehen zur Anforderungsermittlung				
Bereitstellung von Vorgehen zur Analyse von Anforderungen				
Bereitstellung von Vorgehen zum Finden und Auflösen von Konflikten zwischen Anforderungen				
Bereitstellung von Vorgehen zur Dokumentation von Anforderungen				
Bereitstellung von Vorgehen zur Verfolgung von Anforderungen				
Bereitstellung von Vorgehen zur Behandlung von Änderungen an Anforderungen				
Bereitstellung von Vorgehen zur Anforderungvalidierung				
Einbeziehung von Kunden und Stakeholder ins Requirements Engineering				
Unterstützung der Modularisierung durch das Requirements Engineering				

Voll erfüllt
 Teilweise erfüllt
 Nicht erfüllt

Abbildung 4-13: Analyse der Requirements Engineering Ansätze in Bezug auf ihre Eignung für PSS

Quelle: (Berkovich et al. 2011d)

Abbildung 4-14 fasst diese Bewertung zusammen und gibt einen Überblick darüber, welche Lücken in Bezug auf das Requirements Engineering für PSS basierend auf der Literatur bestehen. Für jedes Kriterium wurde der Mittelwert der vier Einzelbewertungen berechnet. Falls sich der Mittelwert nicht eindeutig auf die Skala „voll erfüllt“, „teilweise erfüllt“, „nicht erfüllt“ abbilden lässt, so wurde einer dieser Werte gewählt und anhand der Beschreibung der Analyse begründet.

Kriterium	Erfüllung	Beschreibung
K1: Systematische Durchführung des Requirements Engineering		In allen Domänen und in den integrierten Ansätzen wurde die Wichtigkeit des Requirements Engineering erkannt und es bestehen zu großen Teilen systematische Ansätze. Jedoch ist keiner dieser Ansätze direkt auf PSS übertragbar. Einzelne Ideen und Konzepte können jedoch in einen neuen Ansatz für PSS integriert werden.
K2: Bereitstellung von Vorgehen zur Anforderungsermittlung		Die Anforderungserhebung ist in den einzelnen Domänen und in den integrierten Ansätzen gut ausgearbeitet, jedoch individuell auf diese zugeschnitten. Vor allem die domänenübergreifende Erhebung von Anforderungen, in der auch die impliziten Anforderungen an die Dienstleistungen erhoben werden, wird unzureichend behandelt. In diesem Bereich müssen für PSS spezielle Vorgehen entwickelt werden.
K3: Bereitstellung von Vorgehen zur Analyse von Anforderungen		Es lässt sich feststellen, dass trotz der Bereitstellung von zahlreichen Methoden und Vorgehensweisen für die Anforderungsanalyse, integrierte Vorgehen fehlen, die für PSS geeignet sind. So besteht keine Unterstützung zur Übersetzung von initialen Anforderungen an das PSS in die domänenspezifischen Lösungsanforderungen und zur Berücksichtigung des Zusammenspiels zwischen den Anforderungen verschiedener Domänen untereinander.
K4: Bereitstellung von Vorgehen zum Finden und Auflösen von Konflikten zwischen Anforderungen		In den bestehenden Ansätzen wird nur die Behebung von Konflikten durch Verhandlungen zwischen Kunden und Entwickler beschrieben. Konflikte zwischen verschiedenen Domänen werden nicht berücksichtigt.
K5: Bereitstellung von Vorgehen zur Dokumentation von Anforderungen		Die betrachteten Ansätze schlagen größtenteils natürlichsprachliche Dokumentation der Anforderungen vor. Diese Art der Dokumentation kann auch für PSS eingesetzt werden.
K6: Bereitstellung von Vorgehen zur Verfolgung von Anforderungen		In Produktentwicklung und Softwareentwicklung existieren Vorgehen zur Anforderungsverfolgung, während in der Dienstleistungsentwicklung und in den integrierten Ansätzen weitgehend keine vorhanden sind. Jedoch fehlt eine domänenübergreifende Anforderungsverfolgung, wie sie für PSS nötig ist.
K7: Bereitstellung von Vorgehen zur Behandlung von Änderungen an Anforderungen		In allen Domänen und in den integrierten Ansätzen werden Änderungen nicht zufriedenstellend behandelt. Für PSS ist daher keiner dieser Ansätze verwendbar.
K8: Bereitstellung von Vorgehen zur Anforderungvalidierung		Für die Anforderungvalidierung sind ausgereifte Vorgehen in der Softwareentwicklung vorhanden. Diese können entsprechend angepasst auch für PSS eingesetzt werden
K9: Einbeziehung von Kunden und Stakeholder ins Requirements Engineering		Die Wichtigkeit der Einbeziehung des Kunden ist in allen Domänen und in den integrierten Ansätzen erkannt worden. Meisten wird der Kunde jedoch nur in den ersten Phasen eingebunden. Zudem findet diese Einbindung unstrukturiert statt. Für PSS müssen daher neue Vorgehen hierzu entwickelt werden.
K10: Unterstützung der Modularisierung durch das Requirements Engineering		Die Modularisierung wird in allen Domänen und in den integrierten Ansätzen thematisiert. Die Rolle des Requirements Engineering hierbei ist jedoch nicht klar erkannt und muss für PSS klarer ausgearbeitet werden.



Voll erfüllt



Teilweise erfüllt



Nicht erfüllt

Abbildung 4-14: Ergebnisse der Analyse der Requirements Engineering Ansätze anhand der Kriterien

Quelle: (Berkovich et al. 2011d)

In allen Domänen wurde die Notwendigkeit eines systematischen Requirements Engineering erkannt und Rahmenwerke für die Durchführung des Requirements Engineering aufgebaut. In der Dienstleistungsentwicklung und in den integrierten Ansätzen sind noch am wenigsten Vorgehen zur Anforderungsverwaltung vorgeschlagen. Explizite Vorgehen zur domänenübergreifenden Zusammenarbeit und Kommunikation werden in den Ansätzen nicht erwähnt. Zudem sind in allen Domänen die einzelnen Vorgehen sehr stark auf den jeweiligen Bereich zugeschnitten. Dadurch wird die Kooperation zwischen den Domänen – d.h. die Verwendung von Ergebnissen einer Domäne in einer anderen – stark behindert. Durch die unterschiedlichen Terminologien wird sogar der Blick auf etwaige Gemeinsamkeiten verstellt.

Eine der Hauptkenntnisse der Literaturanalyse ist, dass die einzelnen Domänen bereits ausgereifte Entwicklungstechniken anbieten, die sich bewährt haben. Diese sollten an PSS angepasst werden, was einer domänenübergreifenden Integration gleichkommt. Großer Handlungsbedarf besteht in der Vernetzung der Vorgehen zum Requirements Engineering der verschiedenen Domänen. Trotz konzeptueller und methodischer Analogien im Requirements Engineering, verschärfen verschiedene Terminologien die Hürden für die Zusammenarbeit.

Die größten Herausforderungen im Requirements Engineering für PSS stellen die Anforderungsanalyse mit der Konkretisierung von Anforderungen an PSS und das Änderungsmanagement dar. Ebenso besteht Verbesserungspotenzial bei der Anforderungsermittlung, Anforderungsvereinbarung, Anforderungsverfolgung und Unterstützung der Modularisierung. Bis jetzt hat die Forschung diesen Handlungsbedarf nur punktuell adressiert, wie z.B. in der Mechatronik. Besonders das Requirements Engineering hat die Chance, integrativ zu wirken. Durch seine Aufgabe eine Dokumentation aller Anforderungen an das gesamte PSS zu erstellen ist es prädestiniert für die Integration der Domänen. Das übergeordnete Ziel, den Kunden zufriedenzustellen, könnte in einer gemeinsamen Anforderungsdokumentation münden, die für alle Beteiligten verständlich ist. Darauf aufbauend kann eine abgestimmte Gestaltung des weiteren Entwicklungsprozesses angegangen werden.

Die Ergebnisse der Literaturanalyse, die durch die Erkenntnisse zum Requirements Engineering für PSS in der Praxis ergänzt werden (Kapitel 4.5) dienen als Grundlage zum Aufstellen von Anforderungen an einen Requirements Engineering Ansatz für PSS in Kapitel 4.7.

4.4.3 Zyklische Wechselwirkungen im Requirements Engineering für Product Service Systems in der Literatur

Wie in Kapitel 3.3.2 beschrieben, treten zyklische Wechselwirkungen in der Entwicklung von PSS auf. Im Rahmen der Literaturrecherche in diesem Kapitel wurden verschiedene Zyklen im Requirements Engineering für PSS identifiziert. Im Nachfolgenden werden die häufigsten Zyklen kurz zusammengefasst. Für eine detaillierte und umfassende Darstellung aller Zyklen ist auf Kapitel 4.6 zu verweisen.

Ulrich und Eppinger (2003, 55) geben an, dass die Entwicklung von neuen Technologien einen Einfluss auf den Entwicklungsprozess eines Produktes nehmen kann. Eine neu auf dem Markt erschienene Technologie kann in bereits entwickelte domänenspezifische Komponenten eines PSS integriert werden. Das führt dazu, dass die Anforderungen an die Komponenten des PSS erneut ermittelt und analysiert werden müssen. Ein weiterer Zyklus mit einer externen Ursache ist das Erscheinen eines Konkurrenzproduktes am Markt. Ulrich und Eppinger (2003) und Spath et al. (2001, 121-124) schreiben, dass das Erscheinen eines Konkurrenzproduktes es erforderlich macht, das eigene Produkt zu erweitern, um dessen Attraktivität auf dem Markt sicherzustellen. Dafür werden die Anforderungen an die gesamte Lösung ermittelt, analysiert und bezüglich ihrer Erfüllung durch das existierende PSS bewertet. Kotonya und Sommerville (1998, 123-126) und Pohl (2007, 548) erwähnen, dass ein häufiger Grund für Iterationen im Requirements Engineering durch sich ändernde Kundenwünsche und Stakeholderanforderungen entsteht. Solche Änderungen treten häufig auch erst spät im Lebenszyklus eines Produktes auf, bspw. bei der Nutzung des Produktes durch den Kunden. Aufgrund von Anforderungsänderungen werden die geänderten Anforderungen erneut analysiert. Weitere Iterationen können aufgrund von Anforderungsänderungen durch interne Stakeholder, wie bspw. Entwickler, auftreten (Lamsweerde 2009, 222). Auch Probleme bei der Realisierung der einzelnen Komponenten des Produktes führen zu Zyklen (Pohl 2007, 549). Wenn eine domänenspezifische Komponente nicht realisierbar ist, kann der Grund dafür bei den Anforderungen liegen, die in der jetzigen Form nicht umsetzbar sind.

4.5 Requirements Engineering für Product Service Systems in der Praxis

Unter Einbeziehung der Ergebnisse der Literaturanalyse (Kapitel 4.4), der darin identifizierten Stärken und Schwächen der vorhandenen Ansätze zum Requirements Engineering für PSS

soll das Requirements Engineering in der Praxis analysiert werden. Im Rahmen der Untersuchung des Requirements Engineering in der Praxis wurden zwei Vorstudien durchgeführt. Das Ziel der Vorstudien war den Stand der Praxis zur Ermittlung und Verwaltung von Anforderungen an PSS zu analysieren und darauf aufbauend Probleme und Herausforderungen zu identifizieren. Die erste Vorstudie soll dabei einen breiten Überblick über das Requirements Engineering liefern, während die zweite Vorstudie die identifizierten Spannungsfelder im Requirements Engineering für PSS anhand eines ausgewählten Beispiels und in einem Unternehmen, das an der Herstellung von PSS beteiligt ist, vertiefen soll.

Somit ist die übergreifende Zielsetzung der beiden Studien die aus der Literatur identifizierten Herausforderungen im Requirements Engineering für PSS zu vertiefen, um aufbauend auf den Erkenntnissen aus der Empirie Anforderungen an einen Requirements Engineering Ansatz für PSS aufzustellen.

4.5.1 Vorstudie I: Requirements Engineering für Product Service Systems in der Praxis - Eine Exploration von Problemen und Herausforderungen

Obwohl es mehrere Handlungsleitfäden gibt, die sich mit der Entwicklung und Bereitstellung von PSS beschäftigen, fehlen entsprechende detaillierte Vorgehensweisen zum Requirements Engineering für PSS (vgl. (Berkovich et al. 2009a, 2010a; Berkovich et al. 2009c; Langer et al. 2010)) sowie entsprechende empirische Evidenz dazu.

Das Ziel der Vorstudie ist die Ermittlung von Erkenntnissen aus der unternehmerischen Praxis zu Problemen und Herausforderungen im Requirements Engineering für PSS, die in der Literatur noch keine Begründung gefunden haben (Berkovich et al. 2009b). Diese Erkenntnisse ergänzen die Ergebnisse der Literaturanalyse zum Stand der Wissenschaft zum Requirements Engineering für PSS (Kapitel 4.4) und ermöglichen somit eine bessere Ausrichtung von Anforderungen an einen integrierten und domänenübergreifenden Ansatz zum Requirements Engineering für PSS.

Wie in Kapitel 3 beschrieben, existieren PSS in verschiedenen Ausrichtungen. Bei einem PSS kann beispielsweise die Hardware-Komponente im Vordergrund stehen, während ein anderes PSS Software als zentrales Element beinhaltet. Um in dieser Vorstudie die verschiedenen Arten von PSS abzudecken, wurden daher Unternehmen aus verschiedenen Branchen ausgewählt. Auf diese Weise ist es möglich den Stand des Requirements Engineering für PSS in der Breite zu erfassen.

4.5.1.1 Methodik und Aufbau der Vorstudie

Zur Erhebung und Auswertung des empirischen Datenmaterials wurden die Methoden der qualitativen Sozialforschung herangezogen, die einen interpretativen Charakter haben und es ermöglichen die erhobenen Daten sowohl zum Aufstellen als auch zum Beantworten der Forschungsfragen zu verwenden (Kaplan/Duchon 1988). Im Zentrum von qualitativen Methoden steht das Verstehen und Interpretieren von sozialen Sachverhalten, indem Aussagen und Eindrücke über den zu untersuchenden Gegenstand verbal beschrieben werden. Weder wird die Komplexität sozialer Sachverhalten durch das Anwenden qualitativer Methoden reduziert noch standardisiert (Gläser/Laudel 2004, 23).

Abbildung 4-15 zeigt den Ablauf des verwendeten Forschungsprozesses für die Vorstudie.

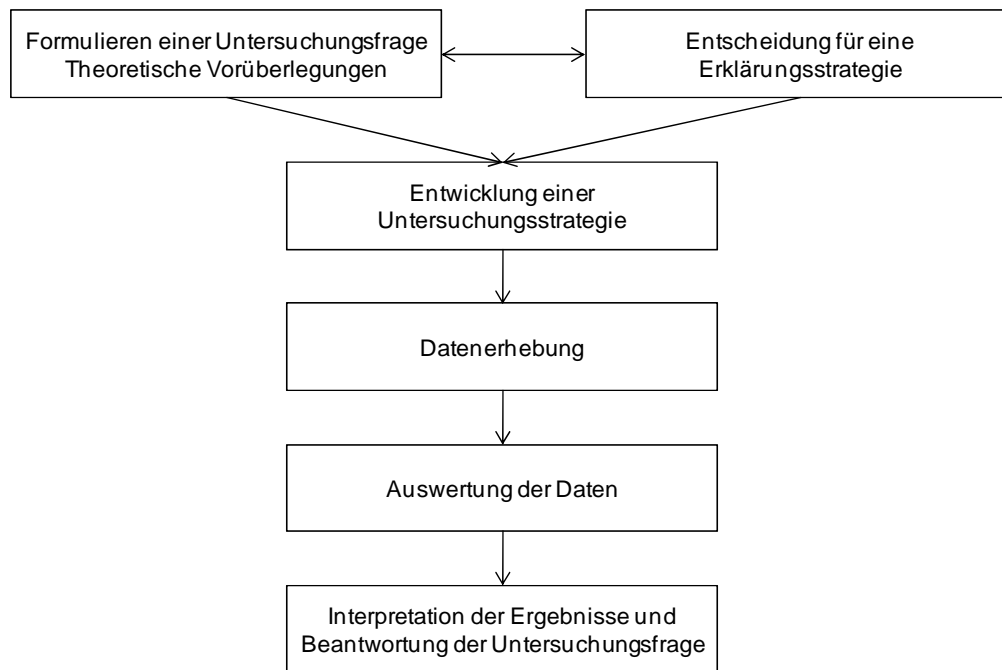


Abbildung 4-15: Ablauf des empirischen Forschungsprozesses zur Erhebung und Interpretation qualitativen Daten

Quelle: In Anlehnung an (Gläser/Laudel 2004, 30-33)

Die Untersuchungsfrage, die den Forschungsprozess leitete, lautet „*Welche Probleme, Herausforderungen und Best-Practices zum Requirements Engineering für PSS bestehen in der Praxis?*“. Als theoretische Vorüberlegung dient die analysierte Literatur zum Requirements Engineering für PSS (Kapitel 4.4.1), die den Kontext zur Formulierung der Untersuchungsfrage liefert. Zur Erhebung der Daten wird eine qualitative Befragung eingesetzt (Kapitel 4.5.1.1.1), die die subjektive Sichtweise der Akteure über Ereignisse, die in Vergangenheit liegen, Zukunftsereignisse oder Erfahrungen ermittelt (Bortz/Döring 2006, 308). Die Daten wurden anschließend ausgewertet und interpretiert (Kapitel 4.5.1.1.2).

4.5.1.1.1 Datenerhebung

Im Rahmen der empirischen Untersuchung wurden 16 semi-standardisierte Experteninterviews in den Branchen Medizintechnik, Automotive und Beratungs- und Systemintegrations-Unternehmen durchgeführt. Ein Experteninterview ist eine Art der Befragung, an der Experten zu einem vorgegebenen Bereich oder Thema befragt werden (Bortz/Döring 2006, 315). Als Experten wurden Personen bezeichnet, die sich im Rahmen ihrer Tätigkeit mit dem Requirements Engineering beschäftigen.

Die betrachteten Unternehmen stellen PSS selbst her oder sind als Zulieferer an der Herstellung von PSS beteiligt. Die Auswahl von verschiedenen Branchen ermöglicht einen tieferen Blick auf den untersuchten Gegenstand aufgrund der verschiedenen Rahmenbedingungen, Kontextinformationen, verwendeten Vorgehensmodelle sowie den Adressaten der hergestellten Leistungen. So werden die Produkte der Medizinbranche unter strenger Berücksichtigung von Gesetzen und Normen zur Entwicklung und Verbreitung medizinischer Geräte hergestellt. Im Bereich der Automotive handelt es sich um Produkte und Dienstleistungen rund um das Gut „Fahrzeug“, während die Beratungs- und Systemintegrations-Unternehmen meist

Lösungen in Form von Software as a Service ohne Selbstentwicklung der Hardware-Komponente herstellen.

Im Automotive-Bereich wurden drei Premium Automobilhersteller ausgewählt. Es wurden sechs Unternehmen anhand der Lünendonk-Liste der „Top 25 IT-Beratungs- und Systemintegrations-Unternehmen in Deutschland“ aus dem Jahr 2007 (http://www.luenendonk.de/it_beratung.php) identifiziert. Die fünf Unternehmen aus dem Medizintechnikbereich wurden anhand von existierenden Kontakten bestimmt. Die Auswahl der Interviewpartner basiert auf dem Konzept des theoretischen Sampling (Glaser/Strauss 1967).

Tabelle 4-5 zeigt die Zuordnung der im Rahmen der empirischen Untersuchung interviewten Unternehmen nach Branchen abhängig von ihrer Größe, dem Umsatz und der Mitarbeiterzahl. Die Zuordnung erfolgt in Anlehnung an die Unternehmensgrößenstatistik des IfM (2002).

Branche	Größe	Umsatz	Mitarbeiter 2007	Anzahl Unternehmen	Anzahl Interviews
Automotive	Groß	ab 50 Mio. €	500 u. mehr	3	5
Beratungs- und Systemintegratoren	Groß	ab 50 Mio. €	500 u. mehr	5	5
	Mittel	Von 1 bis 50 Mio. €	Von 10 bis 499	1	1
Medizintechnik	Groß	ab 50 Mio. €	500 u. mehr	5	5

Tabelle 4-5: Zuordnung von Unternehmen

Quelle: In Anlehnung an (Berkovich et al. 2009b)

Die Interviews fanden im Zeitraum von Juni 2008 bis Oktober 2008 statt. Es wurden vier telefonische und zwölf persönliche Interviews durchgeführt. Jedes Interview hat zwischen 60 und 80 Minuten gedauert. Tabelle 4-6 gibt einen Überblick über die durchgeführten Interviews.

Interviewte	Position des Interviewten	Art des Interviews	Größe des Unternehmens	Branche
Beta	Verantwortliche im Bereich Beratung	telefonisch	groß	Medizintechnik
Gamma	Verantwortliche für das Anforderungsmanagement aus der zentralen IT	persönlich	groß	Automotive
Delta	Verantwortlicher für das fachliche Anforderungsmanagement	persönlich	groß	Automotive
Epsilon	Verantwortlicher für das Requirements & Usability Engineering	persönlich	groß	Automotive
Zeta	Mitglied des Fachgebiets Business Engineering, Schwerpunkt Requirements Engineering	persönlich	groß	Automotive

Eta	Manager, an der Konzeption des Vorgehens im Bereich des Requirements Engineering	persönlich	groß	Beratungs- und Systemintegratoren
Theta	Geschäftsführer, Verantwortlicher für Projekte	persönlich	groß	Beratungs- und Systemintegratoren
Iota	Leiter des Bereiches Beschaffung mit der Schnittstelle zum Requirements Engineering	telefonisch	mittel	Beratungs- und Systemintegratoren
Kappa	Unternehmensberatung im Bereich IT mit der Schnittstelle zu Requirements Engineering	persönlich	groß	Beratungs- und Systemintegratoren
Lambda	Manager, an der Konzeption des Vorgehens für die Durchführung der Projekte, Executive Partner	persönlich	groß	Beratungs- und Systemintegratoren
My	Principal Health Care, Schnittstelle zum Requirements Engineering	persönlich	groß	Beratungs- und Systemintegratoren
Ny	Projektleiter	persönlich	groß	Medizintechnik
Omikron	Bereichsleiter Vertrieb und Marketing, Schnittstelle zum Requirements Engineering	telefonisch	groß	Medizintechnik
Xi	Projektleiter	persönlich	groß	Medizintechnik
Chi	Strategic Account Manager	telefonisch	groß	Medizintechnik
Pi	Mitglied der Projektmanagementgruppe	persönlich	groß	Automotive

Tabelle 4-6: Übersicht zu den Experten-Interviews

Quelle: In Anlehnung an (Berkovich et al. 2009b)

Die Durchführung der Interviews wurde in Anlehnung an das Vorgehen von Bortz und Döring (2006, 310ff.) vorbereitet. Der Interviewleitfaden wurde basierend auf den aus der Literatur abgeleiteten Schwerpunkten zum Requirements Engineering für PSS (vgl. Kapitel 4.4.2) strukturiert. Um generische Antworten zu vermeiden und dabei die Spezifika des Requirements Engineering in einem bestimmten Umfeld zu verstehen, wurde den Interviewten vorgeschlagen, sich auf ein bestimmtes Projekt zu konzentrieren, im Rahmen dessen Requirements Engineering durchgeführt wurde oder wird, und die Fragen in Bezug auf das Projekt zu beantworten. So wurden Fragen zu den folgenden Themenbereichen gestellt:

- Verwendete Vorgehensmodelle in der Entwicklung und im Requirements Engineering,
- Erläuterung des betrachteten Projektes und der Rahmenbedingungen, die einen Einfluss auf das Requirements Engineering haben,

- Spezifika von Dienstleistungen, die im Rahmen des Leistungsbündels mit dem technischen Produkt oder Software angeboten werden,
- Anforderungsermittlung und -analyse,
- Anforderungsdokumentation,
- Anforderungsvereinbarung,
- Anforderungvalidierung,
- Änderungsmanagement und Verfolgung von Anforderungen,
- Auftretende Zyklen und Zyklenursachen im Requirements Engineering.

Zu Beginn des Interviews wurden Angaben zum Unternehmen (Anzahl der Mitarbeiter, Umsatz, Branche, etc.) sowie Angaben zum Interviewten (Position im Unternehmen, Erfahrung im Requirements Engineering, etc.) abgefragt. Der vollständige Interviewleitfaden ist im Anhang (Tabelle 10-1) zu finden. 14 Interviews wurden auf Band aufgezeichnet und anschließend transkribiert. In zwei Fällen, in denen der Interviewte einer Aufzeichnung auf Band nicht zugestimmt hat, wurden stichpunktartige Notizen angefertigt und später zur Erstellung eines schriftlichen Protokolls verwendet. Da es bei den Interviews um die inhaltlichen Aspekte geht, wurden sinngemäße Protokolle als ausreichend betrachtet.

4.5.1.1.2 Inhaltsanalyse

Die aufgezeichneten Interviews wurden transkribiert und ausgewertet. Das Ziel der qualitativen Inhaltsanalyse besteht in der Interpretation der Inhalte des Materials unter dem vorliegenden Kontext (Bortz/Döring 2006, 329). Der Ausgangspunkt der Inhaltsanalyse bildet das erhobene und aufbereitete Datenmaterial. Zur Auswertung der Daten wurde die qualitative Inhaltsanalyse nach Mayring (2007) gewählt. Abbildung 4-16 veranschaulicht die einzelnen Schritte der qualitativen Inhaltsanalyse.

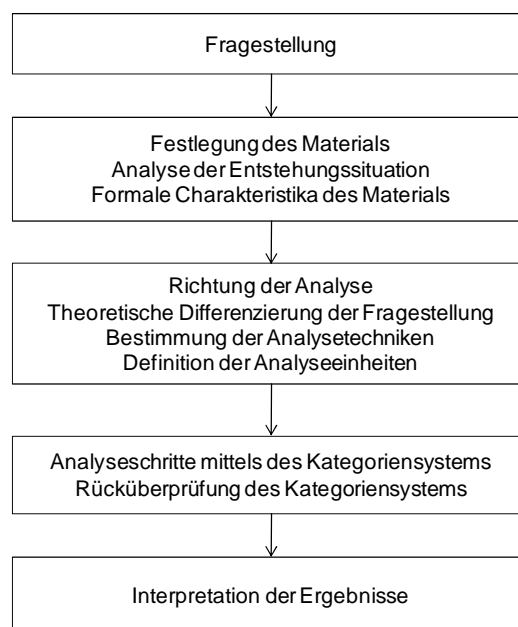


Abbildung 4-16: Ablauf der qualitativen Inhaltsanalyse

Quelle: In Anlehnung an (Mayring 2007, 50ff.)

Im ersten Schritt der Inhaltsanalyse fand die Festlegung des Materials statt, indem bestimmt wurde, welches Datenmaterial und in welchem Umfang in die Auswertung einbezogen werden soll. Anschließend erfolgte die formale Charakterisierung des Materials, indem die Form

der Interviewaufzeichnung definiert wurde. Im dritten Schritt wurde die Richtung der Analyse festgelegt, die die Frage nach dem Inhalt der Interpretation aus dem Material beantwortet. Im vierten Schritt wurde ein Kategoriensystem erarbeitet, das die herauszufilternden Themenbereiche enthält (Atteslander 2008, 189ff.). Die Kategorien wurden in Anlehnung an die während des Interviews abgefragten Themenbereiche definiert (Kapitel 4.5.1.1.1). Im anschließenden Schritt fand die Auswahl einer geeigneten Analysetechnik aus den drei Formen der Zusammenfassung, Explikation und Strukturierung statt. Für die vorliegende empirische Untersuchung wurde die strukturierende Inhaltsanalyse ausgewählt. Durch die strukturierende Inhaltsanalyse sollen nach vorher identifizierten Ordnungskriterien wichtige Aspekte aus dem erhobenen Material herausgefiltert werden. Im abschließenden Schritt der Inhaltsanalyse wurden die Daten interpretiert.

4.5.1.2 Auswertung der Untersuchungsergebnisse

Die meisten untersuchten Unternehmen führen ein systematisches Requirements Engineering durch. In 14 Fällen handelt es sich um ein iteratives Modell, das sich stark an die Modelle aus der Literatur, wie bspw. von Kotonya und Sommerville (1998), anlehnt. Dabei werden folgende Aktivitäten im Rahmen des Requirements Engineering unterschieden: Anforderungsermittlung, Anforderungsanalyse und -vereinbarung, Anforderungsdokumentation und Anforderungvalidierung. Darüber hinaus gehören das Änderungsmanagement und die Verfolgung von Anforderungen auch zum Bereich des Requirements Engineering. In zwei Fällen werden lineare Modelle zum Requirements Engineering eingesetzt, die auf dem Konzept des Change-Request-Verfahrens basieren (*Pi und Eta*).

Die interviewten Unternehmen stellen nicht nur Sach- oder Dienstleistungen her, sondern komplette Lösungen für die Kundenprobleme oder sind an der Herstellung von PSS als Zulieferer beteiligt. *Chi* betrachtet dabei die Dienstleistungen als eigenständige Objekte: „Dienstleistungen werden eigentlich speziell im IT Bereich nicht als Bestandteil des Produktes gesehen, sondern es gibt sozusagen eine eigene Dienstleistungsorganisation [...], deren Aufgabe die Erbringung, Weiterentwicklung, Gestaltung von diesen Dienstleistungen und den dazugehörigen Prozessen ist.“ Zu den angebotenen Dienstleistungen gehören zum Beispiel Unterstützung bei der Erstellung des Lastenheftes, IT-Beratung, Strategieberatung, Prozessanalyse, Unterstützung des Kunden beim Requirements Engineering, Wartung, etc. Die Anforderungen der Kunden sind entscheidend für die Erbringung von Dienstleistungen (*Eta*). Um ein auf die Bedürfnisse des Kunden zugeschnittenes PSS bereitzustellen, ist es wichtig die Anforderungen an Sach- und Dienstleistungen integriert zu ermitteln und anschließend zu verwalten.

Zur Präsentation der Ergebnisse wird im Folgenden jede Kategorie in einem eigenen Unterkapitel detailliert beschrieben.

4.5.1.2.1 Anforderungsermittlung

Im Rahmen der Anforderungsermittlung sollen die Anforderungsquellen identifiziert und die Anforderungen erhoben werden. Zu den wichtigsten Anforderungsquellen gehören der Kunde, der auch einen Fachbereich verkörpern kann, der Markt, gesetzliche Anforderungen sowie interne Anforderungen aus dem Unternehmen. Im Automotive-Bereich wird als Kunde die Fachabteilung betrachtet, die die Anforderungen liefert. So sagte *My*: „Grundsätzlich kommen die Anforderungen vom Kunden“. Ebenso wurden bestehende Dokumente und Systeme als Anforderungsquellen genannt (*Eta und Beta*). Gesetzliche Anforderungen sind besonders für die Medizintechnikbranche bei der Entwicklung medizinischer Geräte wichtig, weil für die

Zulassung medizinischer Geräte die Einhaltung entsprechender Richtlinien erforderlich ist (*Chi*). Sowohl im Automotive-Bereich als auch in der Medizintechnikbranche sind die Marktanforderungen die wichtigsten Quellen zur Entwicklung des technischen Produktes. Beim Anbieten des PSS, das aus technischen Produkten und Dienstleistungen besteht, ist der Kontakt zum Kunden, der die Anforderungen an die gesamte Lösung ohne Unterscheidung in einzelne Sach- und Dienstleistungskomponenten liefert, sehr wichtig und soll über alle Aktivitäten des Requirements Engineering ausgebaut werden. In allen Interviews wurden Workshops als Methode zur Anforderungsermittlung angegeben. Sehr häufig wurden auch Interviews genannt. Die Interviewten erklärten, dass diese Techniken mit verhältnismäßig wenig Vor- und Nachbereitungsaufwand durchgeführt werden können und dass damit üblicherweise gute Ergebnisse erzielt werden. Diese Techniken setzen auch keine besonderen Kenntnisse seitens des Kunden voraus und werden vom Kunden akzeptiert. Vier Unternehmen (*Ny*, *Omikron*, *Xi*, *Delta*) setzen Prototyping zur Anforderungsermittlung ein. Dabei ist zu bemerken, dass diese vier Unternehmen drei verschiedenen Branchen zuzuordnen sind. Weitere Unternehmen haben auf explizite Nachfrage erklärt, dass für sie der Einsatz von Prototypen nicht geeignet ist, da sie nicht für Dienstleistungen eingesetzt werden können. Beobachtungen (*Iota und Ny*) und Brainstorming (*Ny*) spielen nur eine untergeordnete Rolle in der Anforderungsermittlung. *Ny* erklärte, dass Brainstorming vor allem in innovativen Produkten bei schlechter Verfügbarkeit der Stakeholder zu Anwendung kommt. Allerdings ist der Einsatz von Prototypen nicht immer geeignet, besonders wenn es um die Anforderungen an die Dienstleistungen geht.

Zu den größten Herausforderungen bei der Anforderungsermittlung zählen der Aufwand, der sich durch ständige Abstimmungen mit dem Kunden ergibt (*Zeta*) und komplexe Kommunikation mit dem Kunden, der seine Anforderungen nicht genau spezifizieren kann und viele Anforderungen nur implizit äußert (*Eta*, *Ny*). In diesem Fall können Prototypen eingesetzt werden, die jedoch nur für die Darstellung von zentralen Funktionalitäten des Systems verwendet werden, da ansonsten der Aufwand zu groß ist (*Zeta*). Zu den weiteren Herausforderungen bei der Anforderungsermittlung zählt die richtige Auswahl der Stakeholder. So können oft die Stakeholder, die direkt vom Kunden vorgegeben werden, keine relevanten Informationen zur Entwicklung des Systems geben (*Beta und Theta*). Die verschiedenen Ausdrucksweisen der Stakeholder und der Entwickler führen oft zur falschen Interpretation der Anforderungen und somit zu Verständigungsproblemen. Die Stakeholder stellen oft sehr unpräzise Anforderungen an die Lösung, während die Entwickler ihre Anforderungen sehr genau, detailliert und in den meisten Fällen quantitativ ausdrücken (*Eta*, *Lambda*, *My*).

4.5.1.2.2 Anforderungsanalyse

Im Rahmen der Anforderungsanalyse werden die Anforderungen konkretisiert, priorisiert sowie auf Konsistenz überprüft (*Chi*). Die Anforderungen werden nach ihrer Wichtigkeit für den Kunden im Sinne von „muss/soll/kann“ (*Gamma*, *Eta*, *Theta*, *Beta*, *Omikron*) oder nach Nutzen und Aufwand, die entweder von den Entwicklern oder von Kunden selbst festgelegt werden (*Delta*, *Zeta*, *Lambda*, *My*) priorisiert. Dadurch wird ermöglicht die Anforderungen, die für den Kunden und die Stakeholder unwesentlich sind, zu identifizieren und gegebenenfalls auf ihre Umsetzung zu verzichten.

Bei der Konkretisierung von initialen Anforderungen liegt der Fokus auf ihrer „Übersetzung in die Sprache des Entwicklers“ (*Kappa*), indem den unkonkreten initialen Anforderungen quantitative oder definierte qualitative Eigenschaften zugeordnet werden. Dafür müssen der Projektkontext, die Stakeholder und die beteiligten Entwickler in den Prozess der Konkretisie-

ung einbezogen werden (*Kappa*). Desweiteren sollen die Zusammenhänge zwischen den Anforderungen erkannt werden, um somit die Auswirkungen von Anforderungsänderungen zu kontrollieren (*Lambda*). Die Konkretisierung findet mittels UML-Diagramme statt, indem die Anforderungen in atomare Einheiten zerlegt werden, die sich nicht weiter konkretisieren lassen (*Chi*). *Iota* beteiligt den Kunden am Prozess der Konkretisierung, um dadurch die Anforderungen zu verifizieren. Allerdings setzt das voraus, dass sich der Kunde mit den technischen Details der Anforderungen auseinandersetzt. *Eta* setzt für die Anforderungsanalyse die folgende Vorgehensweise ein: Die Anforderungen werden in natürlicher Sprache beschrieben. Anschließend werden sie konkretisiert. Die Konkretisierung findet in mehreren Schritten statt. Als Erstes werden die Ziele und die Anforderungen grob definiert. Anschließend werden die Anforderungen präzisiert, indem ihre Eigenschaften definiert werden. Dadurch wird das Grobkonzept erstellt. Im nächsten Schritt werden die Anforderungen weiter detailliert ausgearbeitet und die ersten Lösungen werden vorgeschlagen.

Eine große Herausforderung stellen die Konkretisierung und Priorisierung von Anforderungen an die Dienstleistungen dar. So äußert der Kunde oft die Anforderungen an eine Lösung und nicht an die einzelnen Sach- und Dienstleistungskomponenten. Deswegen ist es wichtig, die initialen Anforderungen so zu konkretisieren, um daraus die Lösungsanforderungen an die einzelnen Komponenten abzuleiten und dabei das Zusammenspiel zwischen diesen Anforderungen zu berücksichtigen und entsprechend abzubilden (*Chi*). Die Priorisierung von Anforderungen an die Dienstleistungen ist aufgrund ihrer nur impliziten Äußerung durch die Stakeholder sowie ihrer Immaterialität, die sich in einer schweren Vorstellung der Leistung durch den Kunden äußert, schwierig (*Xi*).

Als Fazit lässt sich feststellen, dass die Konkretisierung und Priorisierung von Anforderungen an die Lösungen und ihre Einteilung nach Sach- und Dienstleistungen insbesondere für den Medizintechnikbereich relevant ist. Genau in der Branche stellen Vorschriften und Gesetze Anforderungen bereit, deren Realisierung zur Erhaltung der entsprechenden Zulassung erforderlich ist.

4.5.1.2.3 Anforderungsvereinbarung

Alle Interviewten bestätigten, dass konfliktäre Anforderungen oft während des Projektablaufs vorkommen und entsprechende Maßnahmen zur Auflösung von Konflikten eingeleitet werden müssen. Während der Diskussionen mit den Stakeholdern sollen konfliktäre Anforderungen identifiziert und Konflikte aufgelöst werden. *Eta* meinte: „Anforderungskonflikte bestehen typischerweise zwischen Menschen und müssen zwischen Menschen gelöst werden“. Bei der Auflösung von Konflikten sollen die Interdependenzen zwischen den Anforderungen genau analysiert werden, um bei möglichen Änderungen an Anforderungen, die sich aus den Maßnahmen zur Konfliktauflösung ergeben, die Auswirkungen dieser Änderungen auf weitere Anforderungen zu identifizieren. Eine Herausforderung stellen Konflikte zwischen den Anforderungen unterschiedlicher Domänen dar, die aufgrund ihrer unterschiedlichen Natur schwer zu identifizieren und aufzulösen sind. So wird im Automotive-Bereich verstärkt auf Diskussionen zwischen den Fachbereichen gesetzt, die Anforderungen an die einzelnen Komponenten liefern, um Konflikte zu identifizieren und anschließend zu lösen (*Epsilon*, *Zeta*). Im Medizintechnik-Bereich sollen insbesondere Konflikte zwischen den Kundenanforderungen und Anforderungen, die vom Gesetzgeber vorgegeben sind, identifiziert werden (*Chi*).

4.5.1.2.4 Anforderungsdokumentation

Alle Interviewten haben bestätigt, dass die Anforderungen gemäß den bestehenden Spezifikationsvorschriften zu dokumentieren sind. Durch die Anforderungsdokumentation wird eine Informationsbasis bereitgestellt, die für alle Beteiligten am Entwicklungsprozess, dem Requirements Engineering-Prozess, sowie den Stakeholdern verfügbar ist. So stellt die Anforderungsdokumentation die Grundlage für die Verfolgung von Anforderungen dar, indem die initialen und konkretisierten Anforderungen sowie die Beziehungen zwischen denen festgehalten werden (*Delta*). Laut *Gamma* werden folgende Informationen dokumentiert: eindeutige Identifikationsnummer der Anforderung, Anforderungsbeschreibung in natürlicher Sprache, Verantwortlicher für die Umsetzung der Anforderung, Release, dem die Anforderung zugeordnet ist, Initiator der Anforderung, Reifegrad, Standardfähigkeit, zeitliche Angaben in Bezug auf die Umsetzung und Priorität. Diese Aspekte werden in den meisten Interviews genannt. Eine mögliche Vorgehensweise bei der Anforderungsdokumentation laut *Eta* ist die Beschreibung der Anforderungen erst in natürlicher Sprache und dann die Erarbeitung eines Fachkonzepts, das die genaue Umsetzung der Anforderungen, ohne auf die Implementierungsaspekte einzugehen, darstellt.

In allen Branchen ist die Dokumentation der Anforderungen in Lasten- und Pflichtenheften üblich. Diese Dokumentationsform ist an die klassischen Konzepte der Software- und Produktentwicklung angelehnt. Laut *Epsilon* wird das Lastenheft vom Auftraggeber geschrieben, während das Pflichtenheft vom Auftragnehmer geschrieben wird. Somit kommt es oft vor, dass die Informationen, die im Lastenheft stehen, im Pflichtenheft nicht mehr finden sind. Dadurch wird es schwierig Änderungen an Anforderungen zu integrieren, wenn die zur Durchführung der Änderungen benötigten Informationen fehlen.

4.5.1.2.5 Anforderungvalidierung

Bei der Validierung von Anforderungen geht es darum festzustellen, ob die konkretisierten Anforderungen an die Lösung mit den initialen Anforderungen übereinstimmen, um darauf basierend die Entscheidung zu treffen, welche Anforderungen im welchem Maße umgesetzt werden (*Eta*). Die Validierung wird insbesondere in der Beratungs- und Systemintegrationsbranche auf den Kunden übertragen, der die Entscheidung treffen soll, ob die Anforderungen korrekt sind und umgesetzt werden (*Lambda, Iota, Theta, Kappa*).

Im Automotive-Bereich und in der Medizintechnik-Branche wird diese Entscheidung meist von den Entwicklern getroffen (*Pi, Beta, Gamma, Delta, Zeta, Epsilon, Xi und Chi*). Allerdings wird beim Treffen der Entscheidung über die Anforderungsrealisierung in der Medizintechnik-Branche auf ihre Übereinstimmung mit den Gesetzen und Richtlinien geachtet (*Omikron*).

Eine Herausforderung bei der Validierung stellt die Überprüfung der analysierten Dienstleistungsanforderungen mit den initialen Anforderungen dar. So sollen die analysierten Dienstleistungsanforderungen auch Angaben zu den benötigten Ressourcen zur Erbringung der Dienstleistungen enthalten. Desweiteren soll die Validierung nicht nur am Ende des Requirements Engineering, wenn das Pflichtenheft erstellt ist, stattfinden, sondern fortlaufend während der gesamten Entwicklung.

4.5.1.2.6 Änderungsmanagement und Anforderungsverfolgung

Der Prozess des Änderungsmanagements ist komplex und muss so gestaltet werden, dass die Änderungen an Anforderungen möglichst früh erkannt werden (*Gamma*). Deswegen ist es wichtig, dass bei Änderungen an Anforderungen erst der Einfluss der Änderungen auf das System und weitere Anforderungen überprüft wird. Die Überprüfung von Änderungen an Anforderungen an Dienstleistungen ist mit Schwierigkeiten verbunden, da dabei der Prozess der Dienstleistungserbringung und die Ressourcen zur Erbringung der Dienstleistung analysiert werden sollen.

Die Entscheidung ob eine Änderung umgesetzt wird oder nicht, muss den für die Umsetzung erforderlichen Aufwand berücksichtigen (*Delta, Lambda*). Zudem ist es wichtig auch die benötigte Zeit für die Umsetzung einzubeziehen, um die Auswirkungen auf Liefertermine abzuschätzen (*Delta, Lambda*). Die Entscheidung der Umsetzung wird üblicherweise von einem entsprechenden Gremium getroffen (*Zeta, Pi, Beta*). Diesem Gremium gehören oft auch Vertreter von Kunden und weiteren Stakeholdern an. *Lambda, My, Ny, Kappa, Theta* beziehen den Kunden auch darüber hinausgehend in den Änderungsmanagementprozess ein. Auch im Änderungsmanagement wurde von *Omikron* die Wichtigkeit der gesetzlichen Restriktionen in der Medizintechnikbranche erwähnt.

4.5.1.2.7 Fazit: Herausforderungen im Requirements Engineering für Product Service Systems in der Praxis

Alle Interviewten haben auf die Wichtigkeit des Requirements Engineering für den Entwicklungsprozess hingewiesen. „Das richtige Erfassen, Bewerten und Abstimmen und Übergeben von Anforderungen trägt wesentlich zum Erfolg von Projekten bei. Umgekehrt können Projekte an Fehlern des Anforderungsmanagements scheitern“ (*Eta*). Das Requirements Engineering wird kontinuierlich während des gesamten Entwicklungsprozesses durchgeführt.

Die interviewten Unternehmen stellen nicht nur reine Sach- oder Dienstleistungen her, sondern Lösungen für Kundenprobleme, im Rahmen derer das Zusammenspiel zwischen den Komponenten der Lösung abgestimmt sein soll. Obwohl alle Interviewten darauf hinwiesen, dass eine abgestimmte und integrierte Entwicklung von Sach- und Dienstleistungen erwünscht ist, kann sie nicht immer erfolgen, so dass die Dienstleistungen entweder separat oder im Nachgang entwickelt werden. Daraus resultiert das Problem, dass die Anforderungen an die Sach- und Dienstleistungen nicht abgestimmt ermittelt, konkretisiert und validiert werden und somit oft Konflikte verursachen, die in mehreren Iterationen zwischen den Entwicklern und Kunden geklärt werden. Das führt zu erhöhten Zeit- und Kostenaufwand zur Durchführung der Requirements Engineering Aktivitäten sowie zur Verzögerungen bei der Umsetzung der Anforderungen und zu einer möglichen Unzufriedenheit der Kunden.

Die Analyse von Anforderungen an das PSS, die die initialen Anforderungen in die Lösungsanforderungen an die disziplinspezifischen Komponenten des PSS übersetzen soll, ist mit der Herausforderung verbunden, die Anforderungen an die Dienstleistungen korrekt in Abstimmung mit den weiteren Anforderungen an die Sachleistungen abzuleiten. Dabei sind unter Sachleistungen sowohl eine Zusammensetzung aus Hardware- und Softwarekomponenten, im Sinne von eingebetteten Systemen, als auch die Software alleine, als Teil von Software as a Service, zu verstehen. Zu den weiteren Herausforderungen gehört die Identifikation der Interdependenzen zwischen den Anforderungen an Sachleistungen und an Dienstleistungen. Das liegt daran, dass die Anforderungen an die Dienstleistungen immaterieller Natur sind und die

Angaben zu den Entwicklungsdimensionen (vgl. Kapitel 3.1.1.2) des Prozesses der Dienstleistungserbringung und der dafür benötigten Ressourcen enthalten. Weitere Herausforderungen stellen die richtige Auswahl der Stakeholder und die vollständige Erfassung von Anforderungen dar. Das sind die häufigsten Gründe des Projektabbruchs.

4.5.2 Vorstudie II: Requirements Engineering für Product Service Systems in der Praxis – Eine vertiefende Fallstudie

Die aus den empirischen Untersuchungen (Kapitel 4.5.1) gewonnenen Erkenntnisse wurden vertieft, indem das Requirements Engineering in einem Unternehmen anhand eines ausgewählten Beispiels mit Hilfe von Methoden der qualitativen Forschung analysiert wurde.

4.5.2.1 Zielsetzung, Ablauf, Aufbau und Methodik der Vorstudie

Die Zielsetzung der Vorstudie besteht in der Ermittlung von Erkenntnissen, Problemen und Herausforderungen im Requirements Engineering für PSS in der Praxis. Dafür soll eine vertiefende Fallstudie in einem Unternehmen, das PSS herstellt oder an der Herstellung von PSS beteiligt ist, durchgeführt werden. Die Wahl der Fallstudie als Erhebungsinstrument ist dadurch begründet, dass die Ermittlung von Herausforderungen und Anwendungsbeispielen verschiedene Aspekte, wie Aktivitäten des Requirements Engineering und Ergebnisse dieser Aktivitäten, auftretende Zyklen oder die am Requirements Engineering beteiligten Unternehmensbereiche mit einschließt. Dafür werden unterschiedliche Erhebungsquellen, wie Anforderungsdokumente aber auch Meinungen der Experten benötigt. Der Umgang mit verschiedenen Quellen stellt eine wesentliche Stärke von Fallstudien dar (Yin 2009, 11). Benbasat et al. (1987) definieren die Ziele einer Fallstudie als Untersuchung eines Phänomens in seiner natürlichen Umgebung unter Verwendung unterschiedlicher Methoden zum Sammeln von Informationen. Der Einsatz von Fallstudien ist berechtigt, wenn es darum geht, die Frage nach „Wie?“ und „Warum?“ zu beantworten (Yin 2009, 8). Die vertiefende Fallstudie ergänzt die erste Vorstudie der empirischen Untersuchung des Requirements Engineering in der Praxis (Kapitel 4.5.1), indem auf die im Rahmen der ersten Vorstudie identifizierten Problemen und Herausforderungen im Requirements Engineering in der zweiten Vorstudie verstärkt eingegangen wird.

Die vertiefende Fallstudie fand im Unternehmen *Alpha*⁹ statt, das Haushaltsgeräte herstellt und in Deutschland angesiedelt ist. Die Produkte von *Alpha* werden für den B2C-Bereich entwickelt. Allerdings stellt *Alpha* nicht nur reine technische Produkte her, sondern bietet ebenso Lösungen für Kunden an, die in Form einer Kombination aus technischen Produkten und Dienstleistungen angeboten werden. Die Unternehmensstruktur von *Alpha* enthält mehrere Sparten, die verschiedene Elektro-Groß- und Kleingeräte herstellen. Im Rahmen der Fallstudie wurde das Requirements Engineering im *Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern*¹⁰ untersucht. Dazu wurde ein Entwicklungsprojekt ausgewählt, in dem es um die Weiterentwicklung einer vorhandenen Waschmaschine geht. Tabelle 4-7 zeigt die wesentlichen Elemente der durchgeführten Fallstudie.

⁹ Alle Angaben sind anonymisiert

¹⁰ Anonymisiert

Designelement	Ausprägung
Ziel	Exploration von Herausforderungen im Requirements Engineering für PSS zur Formulierung von Anforderungen an einen Ansatz zum Requirements Engineering sowie Identifizierung von Anwendungsbeispielen.
Forschungsleitende Fragestellung:	Wie wird das Requirements Engineering für PSS im untersuchten Unternehmensbereich bei Alpha durchgeführt?
Forschungsfragen (FF):	FF1: Welche Aktivitäten finden im Rahmen des Requirements Engineering für PSS statt? FF2: Wie werden die Anforderungen an PSS ermittelt, analysiert und vereinbart? FF3: Wie werden die Anforderungen an PSS dokumentiert und validiert? FF4: Wie werden das Änderungsmanagement und die Verfolgung von Anforderungen gewährleistet? FF5: Wann und wie treten welche Zyklen im Requirements Engineering für PSS auf? Was sind ihre Ursachen?
Untersuchungsgegenstand:	Lastenhefte, Pflichtenhefte, Richtlinien und Spezifikationen zur Durchführung des Requirements Engineering.
Erhebungsmethoden:	Workshops, Experteninterviews, Analyse vorhandener Dokumente.

Tabelle 4-7: Designelemente der Fallstudien und deren Ausprägung
Quelle: Eigene Darstellung

Der Ablauf der Fallstudie orientiert sich an die sieben Schritte zur Durchführung von Fallstudien nach Yin (2009), zu denen die Planung, das Design, die Vorbereitung der Fallstudie, die Erhebung der relevanten Daten und ihre Analyse sowie die Aufbereitung der Ergebnisse gehören.

Zur Erhebung der relevanten Daten wurde ein dreistufiger Ablauf angewendet, der in Abbildung 4-17 dargestellt ist.

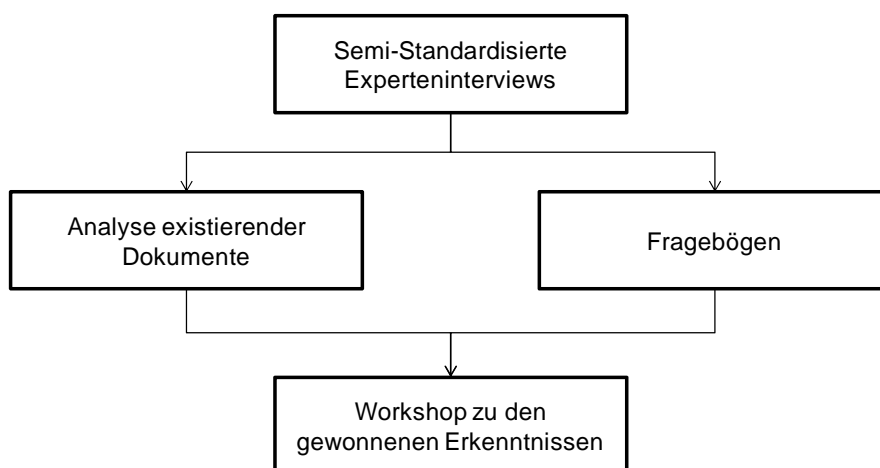


Abbildung 4-17: Ablauf der Datenerhebung
Quelle: Eigene Darstellung

So wurden im ersten Schritt der Datenerhebung semi-standardisierte Experteninterviews im gewählten Unternehmensbereich von Alpha anhand eines Gesprächsleitfadens (Tabelle 10-2

im Anhang) durchgeführt, mit dem Ziel, den Ist-Stand im Requirements Engineering bei *Alpha* im betrachteten Unternehmensbereich zu klären und die Verbesserungspotenziale aus der Sicht der Experten zu ermitteln. Solche Experteninterviews sind teilstrukturierte Interviews, denen ein Gesprächsleitfaden mit den zu besprechenden Themenbereichen zu Grunde liegt (Atteslander 2008, 124). So wurden den Interviewten keine Fragen gestellt, sondern Themenbereiche vorgegeben und basierend auf den Antworten der Interviewten die thematische Ausrichtung angepasst. Während der Gespräche wurden Stichwortprotokolle geführt, die auch die Grundlage der anschließenden Auswertung waren. Der Fokus lag dabei auf inhaltlichen Informationen, weswegen sinnigere Protokolle ausreichend waren. Es wurden insgesamt vier persönliche Interviews mit zwei Experten im Zeitraum zwischen Juni und Juli 2010 durchgeführt. Jedes Interview hat zwischen 1 und 1,5 Stunden gedauert. Beide Interviewte (im Verlauf der Arbeit als **Experte I** und **Experte II** bezeichnet) sind für das Requirements Engineering im *Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern* bei *Alpha* zuständig.

Basierend auf den gewonnenen Erkenntnissen zu den Herausforderungen im Requirements Engineering für PSS im gewählten Unternehmensbereich bei *Alpha* wurden existierende Unterlagen ausgewertet. Der Zweck der Dokumentanalyse ist das Verstehen des Sinnes der vorliegenden Dokumente (Kromrey 2000, 169). Die Auswertung fand zwischen Juli und September 2010 statt. Zu den Vorteilen der Analyse vorhandener Dokumente gehören bereits fertige Daten, die nicht mehr abgefragt werden müssen. Die Auswahl von Dokumenten basiert auf den Kriterien, die nach Mayring (2002) definiert wurden. Dazu gehören die Art des Dokuments, wie bspw. innere Merkmale (Inhalt und Aussagekraft), Intendiertheit (ursprüngliche Funktion des Dokuments), Gegenstandsnahe, (zeitliche, räumliche oder soziale), und Herkunft des Dokuments. Zur Identifikation der erforderlichen Daten wurden ebenso die der Fallstudie zu Grunde liegenden Forschungsfragen herangezogen, die die Richtung der Interpretation der Daten vorgaben. Tabelle 4-8 gibt eine Übersicht über die analysierten Unterlagen.

Art der Unterlagen	Anzahl
Lastenheft (<i>im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern</i>)	2
Pflichtenheft (<i>im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern</i>)	2
Übersicht über das vorhandene Sortiment (<i>im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern</i>)	1
Produktplan (<i>im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern</i>)	1
Richtlinie zur Durchführung des Requirements Engineering (<i>im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern</i>)	1
Richtlinie zur Durchführung des Entwicklungsprozesses (<i>bereichsübergreifend</i>)	1

Tabelle 4-8: Übersicht zu den analysierten Unterlagen

Quelle: Eigene Darstellung

Gleichzeitig wurden Fragebögen (Tabelle 10-3 im Anhang) an drei Mitarbeiter, die am Requirements Engineering im gewählten *Unternehmensbereich* bei *Alpha* operativ beteiligt sind, zur Einschätzung des Ist-Standes des Requirements Engineering und zur Ergänzung der Er-

gebnisse der Interviews verteilt. Die Fragen wurden basierend auf den Erkenntnissen aus den Experteninterviews erstellt.

Die aus den Experteninterviews, der Analyse der vorhandenen Dokumenten und Fragebögen gewonnenen Erkenntnisse wurden in einem Workshop den Experten I und Experten II präsentiert und mit ihnen diskutiert.

4.5.2.2 Auswertung und Vergleich der Untersuchungsergebnisse

Die Untersuchungsergebnisse werden in zwei Teilen präsentiert: In Kapitel 4.5.2.2.1 wird der Ist-Stand des Requirements Engineerings im gewählten Unternehmensbereich bei *Alpha*, in dem Waschmaschinen und Trockner hergestellt werden, beschrieben. In Kapiteln 4.5.2.2.2, 4.5.2.2.3, 4.5.2.2.4 werden die im Rahmen der Analyse identifizierten Herausforderungen und Probleme im Requirements Engineering diskutiert.

4.5.2.2.1 Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha

Der komplette Entwicklungsprozess im Unternehmensbereich bei *Alpha*, in dem Waschmaschinen und Trockner hergestellt werden, besteht aus mehreren Phasen, von denen in den ersten drei Phasen das Requirements Engineering hauptsächlich stattfindet. So werden in der ersten Phase der Entwicklung die Stakeholder, zu denen die Entwicklung, der Kunde, der Markt, der Kundenservice, der Gesetzgeber gehören, identifiziert und die Ideen für das zu entwickelnde Produkt gesammelt. Die organisatorische Struktur des Unternehmensbereichs zur Herstellung von Waschmaschinen und Trocknern kann grob in Marketing-Abteilung, Produktion-Abteilung und Entwicklung aufgeteilt werden.

Ausgehend von den identifizierten Ideen an das Produkt werden die Anforderungen von der Marketing-Abteilung mit Hilfe von Checklisten, existierenden Anforderungskatalogen sowie Quality Function Deployment (QFD) formuliert. Anschließend werden die Anforderungen ins Lastenheft übernommen. In der zweiten Phase werden die Anforderungen sowie die Realisierungsvorschläge aus der Entwicklung (Konzepte) von der Produktion-Abteilung abgestimmt und bewertet. Die Anforderungen werden dabei den Funktionen, die zu den Funktionsstrukturen zusammengefasst sind, zugeordnet und konkretisiert. Die so gewonnenen lösungsorientierten Anforderungen werden ins Pflichtenheft übernommen. Um das Pflichtenheft zu vervollständigen, wird das vorhandene Wissen der Entwickler aufgegriffen und in mehreren Iterationen zur Konkretisierung von Lösungsanforderungen, bis sie den benötigte Detaillierungsgrad erreichen, eingesetzt. Das fertige Pflichtenheft wird an die Entwicklung zur weiteren Umsetzung gegeben. Dafür werden die Anforderungen bezüglich der Erzeugnisstruktur iterativ konkretisiert, bis sie direkt realisiert werden können. Abbildung 4-18 zeigt den Ablauf des Requirements Engineering im gewählten Unternehmensbereich bei *Alpha*.

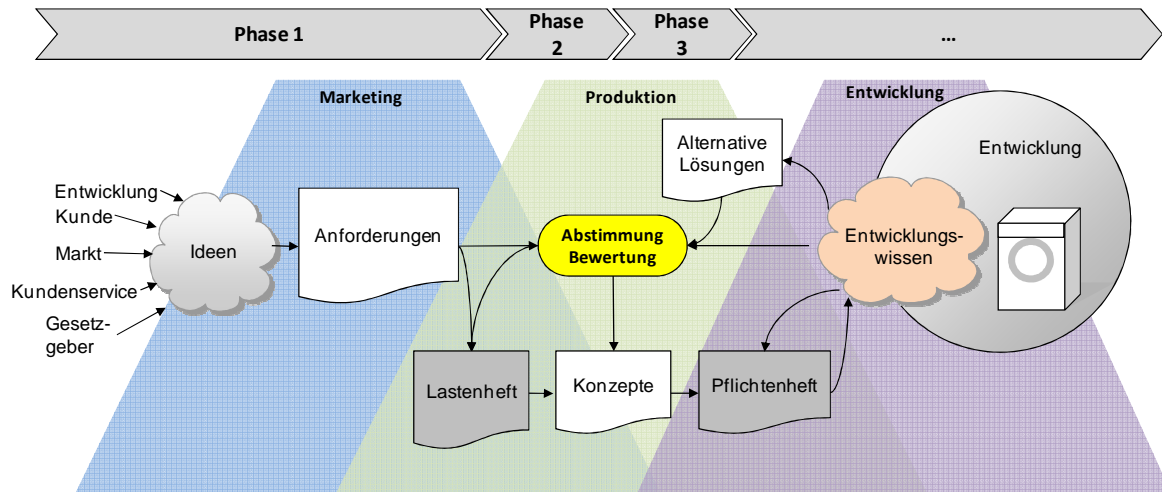


Abbildung 4-18: Ablauf des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha
Quelle: Eigene Darstellung

4.5.2.2.2 Herausforderungen im Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha resultierend aus den Experteninterviews

Als eine der größten Herausforderungen im Requirements Engineering wurde von den Experten in den Interviews die weitgehend fehlende Verknüpfung zwischen den Anforderungen der Lasten- und Pflichtenhefte genannt. So werden die Anforderungen des Lastenhefts ausschließlich von der Produktions-Abteilung konkretisiert, indem sie detaillierte quantitative und qualitative Ausprägungen erhalten und lösungsorientiert beschrieben werden. Dadurch werden oft die Anforderungen aus dem Lastenheft, und dabei insbesondere die Anforderungen an die Dienstleistungen, falsch interpretiert. Desweiteren werden solche Anforderungen ins Pflichtenheft aufgenommen, die von der Entwicklung für erforderlich gehalten werden, von denen jedoch die entsprechenden Begründungen im Lastenheft fehlen. Die Konkretisierung von Anforderungen und ihre Übernahme ins Pflichtenheft berücksichtigen die Entwicklungsinformationen nicht in vollem Maße. Das führt dazu, dass die Übertragung von Anforderungen an die Entwicklung mehrere Iterationen zwischen der Entwicklung und der Produktion-Abteilung benötigt. Eine Verbindung zwischen den Anforderungen aus den Lasten- und Pflichtenheft und ihrer anschließenden Umsetzung fehlt weitgehend. Dadurch können Abweichungen in den Anforderungen oder in der Umsetzung nicht verfolgt werden.

Eine weitere Herausforderung ist die Übernahme von neuen Anforderungen in bestehende Lasten- und Pflichtenhefte. Neue Anforderungen, die während der Entwicklung auftreten, sollen auf eine Art und Weise in die Lasten- und Pflichtenhefte übernommen werden, so dass es möglich ist, den Ursprung der Anforderungen aber auch die von ihr abgeleiteten Anforderungen nachzuvollziehen. Somit dürfen keine Anforderungen ins Pflichtenheft aufgenommen werden, die nicht von den Anforderungen im Lastenheft abgeleitet wurden.

Die Strukturierung der Anforderungen in den Lasten- und Pflichtenheften ist eine weitere Herausforderung. Da an der Erstellung von Lasten- und Pflichtenheften mehrere Abteilungen des betrachteten Unternehmensbereichs von *Alpha* beteiligt sind, ist es wichtig, dass die Anforderungen eine einheitliche Struktur in Bezug auf die Dokumentation und die Attribute be-

sitzen. Weiterhin sollen die Anforderungen, die von verschiedenen Abteilungen stammen, nach Kategorien gegliedert sein, um dadurch ihre Handhabung zu vereinfachen. Durch die Kategorisierung können die Anforderungen auf einfachere Weise adressiert werden.

Die Zuordnung von konkretisierten Anforderungen zu den einzelnen Komponenten der Erzeugnisstruktur soll vom Requirements Engineering abgebildet werden. Zurzeit wird diese Zuordnung noch nicht systematisch und einheitlich gemacht. Falls sich die Anforderungen an die Komponenten ändern, soll der Grund der Änderung klar sein und die Auswirkungen der Änderung auf weitere Anforderungen aber auch auf die Komponenten des PSS ersichtlich sein.

4.5.2.2.3 Herausforderungen im Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha resultierend aus der Analyse von Dokumenten

Weitere Erkenntnisse wurden anhand der Analyse von Dokumenten gewonnen. So wurden zwei wesentliche Erkenntnisse identifiziert:

- **Unterschiedliche Detaillierungsgrade von Anforderungen.** Sowohl in Lasten- als auch Pflichtenheften wurden Anforderungen identifiziert, deren Detaillierungsgrad nicht zu dem Konzept des Dokuments oder zu den anderen Anforderungen passte. Der Detaillierungsgrad einer Anforderung gibt an, wie konkret eine Anforderung in Bezug auf quantitative und qualitative Angaben beschrieben ist. Die untersuchten Lastenhefte enthielten lösungsorientierte technische Anforderungen, während in den Pflichtenheften unpräzise und lösungsneutrale Anforderungen vorhanden waren.

So wurden bspw. im Lastenheft folgende Anforderungen (von insgesamt über 50) identifiziert¹¹

- (1) Die Waschmaschine soll eine emotionale Bindung zum Kunden herstellen und gleichzeitig über die bestmögliche Technologie verfügen.
- (2) Der Kunde soll in der Lage sein, den Geräuschpegel der Waschmaschine selbst einzustellen.
- (3) Geräuschpegel während des Waschvorgangs: Meisterklasse.
- (4) Die Wassersensoren sollen von der Variante X¹² sein.

Die Anforderung (1) stellt ein übergeordnetes Ziel dar, jedoch keine Anforderung, die direkt an die Entwicklung gegeben werden kann. Die Anforderung (2) stellt eine konkrete produktorientierte Anforderung dar. Die Anforderung (3) ist eine allgemeine Rahmenbedingung, die sich auf das gesamte Produkt bezieht. Und die Anforderung (4) ist eine konkrete produktorientierte und technische Anforderung.

Aus den unterschiedlichen Detaillierungsgraden von Anforderungen resultiert eine komplexe Koordination der Anforderungsaufnahme zwischen den Unternehmensbereichen. Während die Marketing-Abteilung, die direkt mit dem Kunden und dem Markt arbeitet, sehr unspezifische und lösungsneutrale Anforderungen an das Produkt liefert (wie z.B. Anforderung (1)), stellen die Entwicklung und das Projektmanagement initiale konkrete und lösungsorientierte Anforderungen bereit, die ins Lastenheft

¹¹ Alle Anforderungen spiegeln nur sinngemäß den Inhalt wider, um die Anonymisierung zu gewährleisten

¹² Anonymisiert

aufgenommen werden. Da sich die Anforderungen von der Struktur und vom Inhalt unterscheiden, ist es schwierig sie einheitlich ins Lastenheft zu übernehmen und anschließend zu adressieren.

Desweiteren resultiert aus den unterschiedlichen Detaillierungsgraden von Anforderungen eine komplexe Suche nach Konflikten zwischen lösungsneutralen und lösungsorientierten Anforderungen. So ist es schwierig festzustellen, ob sich die Anforderungen, wenn sie unterschiedliche Detaillierungsgrade haben, widersprechen. Ebenso ist es schwierig die Interdependenzen zwischen solchen Anforderungen festzustellen, um bei möglichen Änderungen die Auswirkungen dieser Änderungen auf weitere Anforderungen vorausschauend zu identifizieren.

Ein weiteres Problem ist eine unstrukturierte und unübersichtliche Konkretisierung von Anforderungen zu ihrer Übernahme ins Pflichtenheft. Die präzisen lösungsorientierten Anforderungen wurden oft aus dem Lastenheft durch eine geringe Umformulierung ins Pflichtenheft übernommen. Die Konkretisierung solcher Anforderungen bleibt unklar.

- **Weitgehend keine Unterstützung der Anforderungsverfolgung.** Die Anforderungen aus dem Lastenheft haben oft keine Verbindung zu den Anforderungen im Pflichtenheft. Die Konkretisierung von Anforderungen aus dem Lastenheft ist dadurch nicht nachvollziehbar. Desweiteren enthält das Pflichtenheft oft Anforderungen, die über keine Verbindung zum Lastenheft verfügen. Durch die fehlende Verbindung von den Anforderungen aus dem Pflichtenheft zu den Anforderungen im Lastenheft besteht keine Begründung für die Existenz von Anforderungen im Pflichtenheft und somit keine Begründung für ihre Umsetzung.

Durch die Betrachtung des Prozesses der Lasten- und Pflichtenhefterstellung konnte der Grund für dieses Problem gefunden werden: Die Anforderungen aus dem Lastenheft werden von der Produktion-Abteilung analysiert und konkretisiert und ins Pflichtenheft übernommen. Dabei fehlt oft eine Rücksprache mit der Marketing-Abteilung, die für die Erstellung der Lastenhefte zuständig ist. Somit werden die Anforderungen oft nicht auf Grund der Angaben im Lastenheft konkretisiert, sondern stammen direkt aus dem Wissen der Entwickler. Das resultiert in der Aufnahme von Anforderungen ins Pflichtenheft, deren Vorgänger nicht im Lastenheft definiert wurden. Deswegen ist die Begründung für die Existenz und eine Abbildung auf die initialen Anforderungen an das PSS nicht klar.

Im Lasten- und Pflichtenheft wurden bspw. folgende Anforderungen (von insgesamt über 70) identifiziert¹³:

- (1) Die zu entwickelnde Waschmaschine soll eine emotionale Bindung zum Kunden entwickeln und dabei über die beste Technologie verfügen (Anforderung aus dem Lastenheft).
- (2) Stiftung „Warentest“ soll der Waschmaschine die Note „sehr gut“ vergeben (Anforderung aus dem Lastenheft).
- (3) Minimal ausreichende Sensoren sollen verwendet werden (Anforderung aus dem Pflichtenheft).

¹³Alle Anforderungen spiegeln nur sinngemäß den Inhalt wider, um die Anonymisierung zu gewährleisten

Anforderung (1) gibt keinen Aufschluss über ihre Umsetzung. Im Pflichtenheft wurden keine Anforderungen identifiziert, die aus Anforderung (1) abgeleitet wurden. Somit ist unklar, wie die vorliegende Anforderung umgesetzt wurde. Bei Anforderung (2) ist ebenso unklar, wie sie realisiert wird. In dieser Form handelt sich hier mehr um ein Ziel, nicht um eine konkrete Anforderung. Die genauen Zielwerte der Stiftung Warentest müssen in abgeleiteten Anforderungen angegeben werden. Anforderung (3) hat keine Verbindung zum Lastenheft. Es ist unklar, warum sie benötigt wird und welche Kunden- und Stakeholderanforderungen durch sie ausgedrückt werden.

Die weitgehend fehlende Unterstützung der Verfolgung zwischen den Anforderungen resultiert in verschiedenen Problemen:

1. Die Überprüfbarkeit der Umsetzung von Anforderungen ist nicht gegeben. So ist bspw. nicht klar, wie Anforderungen (1) und (2) genau realisiert werden. Somit ist es nicht möglich zu überprüfen, ob das Produkt die identifizierten Anforderungen des Lastenheftes erfüllt.
2. Die unvollständige Verfolgbarkeit von Anforderungen führt weiterhin zu einem komplexen Änderungsmanagement. Es ist nicht möglich, den Einfluss einer Änderung einer Anforderung aus dem Pflichtenheft auf die initialen Kunden- und Stakeholderanforderungen zu bestimmen. Somit können Änderungen der Lösungsanforderungen im Pflichtenheft nicht auf mögliche Auswirkungen auf Anforderungen im Lastenheft überprüft werden.

Ebenso können aufgrund der weitgehend fehlenden Anforderungsverfolgung die Konkretisierung von den Anforderungen aus dem Lastenheft und ihre anschließende Übernahme ins Pflichtenheft unstrukturiert und unübersichtlich erscheinen. Die Entwickler können deswegen nicht nachvollziehen, wie Anforderung (3) konkretisiert wurde, um sie ins Pflichtenheft zu übernehmen. Dadurch ist es nicht möglich, die Konkretisierungsschritte zu wiederholen oder sie anders zu gestalten.

3. Daraus resultieren auch Schwierigkeiten bei der Abstimmung von Anforderungen zwischen den verschiedenen Stakeholdern. Oft verstehen die Stakeholder nur die Anforderungen, die im Lastenheft stehen, aber nicht die lösungsorientierten Anforderungen aus dem Pflichtenheft. Wenn es Anforderungen gibt, die nicht aus dem Lastenheft abgeleitet wurden, ist es für die Stakeholder schwierig die Gründe für diese Anforderungen zu verstehen und somit über sie abzustimmen. Mögliche Konflikte, die aufgrund von verschiedenen Auffassungen der Stakeholder auftreten, können nur schwer identifiziert werden. Die Entscheidungen über die Priorisierung von Anforderungen können schwer nachvollzogen werden. Für die Anforderungen im Lastenheft ist weitgehend nicht möglich zu entscheiden, welche von denen eine höhere Priorität besitzen sollen, wenn unklar ist, wie und ob sie im Pflichtenheft erscheinen. Die Auswirkungen der Nichterfüllung der Anforderungen bleiben ebenso unklar.

4.5.2.2.4 Herausforderungen im Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha resultierend aus den Fragebögen

Zeitgleich mit der Analyse von Dokumenten wurden in einem Workshop die am Requirements Engineering beteiligten Mitarbeiter im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei *Alpha* anhand von Fragebögen befragt. Das Ziel der Befragung war das Ermitteln des Ist-Standes im betrachteten Unternehmensbereich. Dazu wurden die zuvor identifizierten Probleme und Herausforderungen als positiv formulierte Aussagen angegeben, denen die Befragten auf einer Skala von „stimme voll und ganz zu“ (=1) bis „stimme überhaupt nicht zu“ (=5) antworten konnten (Tabelle 10-3 im Anhang). Das Ziel dieser Befragung ist die Validierung der Ergebnisse der Analyse der existierenden Dokumenten und der Erkenntnisse aus den Experteninterviews. Insgesamt konnten drei Mitarbeiter für die Beantwortung des Fragebogens gewonnen werden. Aufgrund dieser geringen Anzahl an Antworten können keine ausgereiften statistischen Verfahren angewendet werden. Daher erfolgte ein Ranking der Aussagen anhand der Antworten der Befragten.

Die geringste Zustimmung erhielt die Aussage Nr. 9, die wie folgt lautet: „Im Lastenheft wird klar festgehalten, warum eine Anforderung vorhanden ist“. An nächster Stelle folgen die Aussagen Nr. 4 und Nr. 5: „Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) ermöglicht es auf einfache Weise, bei der Änderung einer Anforderung, davon betroffene Anforderungen zu finden.“ Und „Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) gibt beim Einfügen von neuen Anforderungen eine eindeutige Stelle vor, an der die Anforderung abgelegt wird.“ Aus diesen Antworten lässt sich schließen, dass die größten Probleme in der Anforderungskonkretisierung und -verfolgung liegen. Sowohl Begründungen für Anforderungen als auch abhängige Anforderungen bei Änderungen können nur schwer identifiziert werden. Zudem ist unklar, an welchen Stellen neue Anforderungen eingefügt werden müssen.

4.5.2.2.5 Fazit: Zusammenfassung der Ergebnisse der vertiefenden Fallstudie im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha

Die aus der Analyse von Dokumenten und den Fragebögen gewonnenen Erkenntnisse wurden anschließend in einem Workshop dem Experten I und Experten II (s. Kapitel 4.5.2.1) präsentiert.

Zuletzt wurden von den Experten folgende Wünsche an das Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei *Alpha* geäußert:

- Das Requirements Engineering soll eine Art „Wissensdatenbank“ für Anforderungen sein, indem alle Anforderungen und ihre Konkretisierungen über die Phasen der Entwicklung festgehalten werden.
- Das Requirements Engineering soll die Wiederverwendbarkeit der Anforderungen ermöglichen.
- Das Requirements Engineering soll die Verfolgung der Anforderungen auch über die Phasen der Entwicklung ermöglichen. Der Übergang von Lastenheft ins Pflichtenheft soll klar und nachvollziehbar sein, so dass ersichtlich ist, aus welchen Anforderungen des Lastenhefts welche Anforderungen des Pflichtenhefts abgeleitet wurden.

- Das Requirements Engineering soll die Durchgängigkeit der Anforderungsverfolgung von der Definition der Anforderungen bis hin zu dem Entwurf des Produktes gewährleisten. Somit kann auch eine fortlaufende Validierung von Anforderungen zur Sicherstellung ihrer Qualität gewährleistet werden. Dabei soll sichergestellt werden, dass die Anforderungen basierend auf dem Entwicklungsfortschritt über identische Detaillierungsgrade verfügen.

Desweiteren haben die Experten darauf hingewiesen, dass die initialen Anforderungen in mehreren Schritten unter Einbeziehung der Entwicklungsinformationen konkretisiert werden. Allerdings werden die Anforderungen ausschließlich in Lasten- und Pflichtenheften festgehalten, was die Verfolgung der Konkretisierung erschwert. Dadurch sind oft Entscheidungen, die bezüglich der Konkretisierung von Anforderungen gefällt wurden, nicht nachvollziehbar. Folglich ist die Integration der Änderungen an Anforderungen erschwert, weil unklar ist, wie sich die Änderungen im Pflichtenheft auf die Anforderungen im Lastenheft auswirken. Und ebenso spielt die Suche und Auflösung von Konflikten zwischen den Anforderungen eine wichtige Rolle. Aufgrund von unterschiedlichen Detaillierungsgraden der Anforderungen sowohl im Lasten- als auch im Pflichtenheft ist es problematisch, Konflikte zwischen solchen Anforderungen zu identifizieren und anschließend aufzulösen.

4.5.3 Zyklische Wechselwirkungen im Requirements Engineering für Product Service Systems in Vorstudie I und Vorstudie II

Zusätzlich zu den in der Literatur gefundenen Zyklen (Kapitel 4.4.3), wurden zyklische Wechselwirkungen ebenso im Rahmen von Vorstudie I und Vorstudie II identifiziert. Größtenteils konnten in der Literatur gefundene Zyklen auf diese Weise bestätigt werden. Im Nachfolgenden werden die wichtigsten Zyklen kurz zusammengefasst. Die umfassende Darstellung aller Zyklen ist in Kapitel 4.6 zu finden.

Dem Interviewten *Eta* sind Zyklen auf Grund sich verändernder Stakeholderanforderungen bekannt. Das Unternehmen, in dem der Interviewte tätig ist, koordiniert IT-Projekte, bei denen eine große Anzahl von Stakeholdern aus verschiedenen Unternehmen beteiligt ist. Daher treten Änderungen von Stakeholderanforderungen auf, die Auswirkungen auf die Realisierung des Gesamtproduktes haben.

Eine ähnliche Form von Zyklen hat der Interviewte *Xi* beobachtet. In seinem Unternehmen werden neue Produkte und Dienstleistungen mit einer sehr kurzen Time-to-Market entwickelt. Daher beginnen die ersten Umsetzungsschritte bereits während der Anforderungserhebung. Als Folge davon treten umfangreiche Anforderungsänderungen auch in späten Phasen der Entwicklung auf, wodurch Wiederholungen von Anforderungs- und Umsetzungsaktivitäten hervorgerufen werden.

Das Unternehmen des Interviewten *Iota* bietet Dienstleistungen im Zusammenhang mit SAP-Integrationsprojekten an. In diesem Umfeld treten vor allem Zyklen ausgelöst durch externe Faktoren, wie der wirtschaftlichen Situation des Kunden, auf. Während der Durchführung von Großprojekten mit einer Dauer von mehreren Jahren, ändert sich die Geschäftssituation des Kunden beispielsweise durch Zukauf oder Abstoß von neuen Sparten oder Sub-Unternehmen. Dadurch ergeben sich umfangreiche Veränderungen an den Produkten und Dienstleistungen, die Wiederholungen in der Anforderungserhebung und Entwicklung erforderlich machen.

In Vorstudie II im Unternehmen *Alpha* wurden Zyklen in der Abstimmung zwischen Lasten- und Pflichtenheften identifiziert. Aufgrund von Missverständnissen zwischen den Require-

ments Analysten und der Entwicklung sind mehrere Iterationen zum Klären der Anforderungen erforderlich. Falls Anforderungen im Pflichtenheft nicht lösungsorientiert formuliert sind, müssen diese Anforderungen von der Entwicklung an die Requirements Analysten zurückgegeben werden, damit diese die betreffenden Anforderungen weiter konkretisieren.

4.6 Zyklenmanagement im Requirements Engineering für Product Service Systems

Im Rahmen der Literaturanalyse (Kapitel 4.4) und der empirischen Untersuchung (Kapitel 4.5.1 und Kapitel 4.5.2) wurden umfassende Erkenntnisse zum Zyklenmanagement gewonnen. Eine wichtige Erkenntnis ist, dass das Bewusstsein für Zyklen und das Management von Zyklen im Requirements Engineering eine große Herausforderung darstellt. Zyklen kommen in allen Domänen vor und wurden in den Unternehmen auch erkannt. Jedoch fehlen noch Vorgehensweisen, um bewusst mit Zyklen umzugehen und sie adäquat im Requirements Engineering zu verwalten.

Im Folgenden werden die in der Literaturanalyse und in den empirischen Untersuchungen gefundenen Beispiele für Zyklenursachen und die daraus resultierenden Zyklen im Requirements Engineering dargestellt (Berkovich et al. 2011e). Die Zyklenursachen wurden anhand inhaltlicher Ähnlichkeiten charakterisiert. Tabelle 4-9 gibt einen Überblick über die identifizierten Zyklenursachen.

Nr.	Kategorie der Zyklenursachen	Beschreibung der Zyklenursachen
1.	Verfügbarkeit und Reife von Technologien	<ul style="list-style-type: none"> • Neue Softwaretechnologie ist vorhanden. • Neue Produkttechnologie ist vorhanden. • Neue Dienstleistungen sind vorhanden.
2.	Wettbewerbliche Entwicklungen	<ul style="list-style-type: none"> • Das angebotene PSS wird nicht mehr vom Kunden benötigt. • Konkurrenzprodukt ist für den Kunden attraktiver als das angebotene PSS. • Angebotene Dienstleistungen werden nicht mehr benötigt. • Neuer Trend auf dem Markt ist entstanden.
3.	Unterschiedliche Lebenszyklen von Hardware- und Software-Komponenten des PSS	<ul style="list-style-type: none"> • Hardware veraltet. • Software veraltet.
4.	Änderung von Kunden- und Stakeholderanforderungen	<ul style="list-style-type: none"> • Kunde / Stakeholder ändern ihre Anforderungen.
5.	Finanzwirtschaftliche Änderungen	<ul style="list-style-type: none"> • Neues PSS muss sofort dem Kunden bereitgestellt werden. • Preis des PSS muss reduziert werden.

6.	Entwicklungsänderungen	<ul style="list-style-type: none"> • Realisierung der Hardware-Komponente ist nicht möglich. • Realisierung der Software-Komponente ist nicht möglich. • Realisierung der Dienstleistung ist nicht möglich. • Differenzen im Ergebnis der einzelnen Domänen werden festgestellt. • Interne Stakeholder ändern die Anforderungen.
7.	Legislative Änderungen	<ul style="list-style-type: none"> • Änderung eines Gesetzes / einer Norm etc.
8.	Abgleich von Anforderungen	<ul style="list-style-type: none"> • Missverständnisse bezüglich der Anforderungen zwischen den Requirements Engineering Analysten und Entwicklern.

Tabelle 4-9: *Zyklusursachen im Requirements Engineering für PSS*

Quelle: In Anlehnung an (Berkovich et al. 2011e)

Im Nachfolgenden werden die Zyklusursachen und die daraus resultierenden Zyklen detailliert beschrieben (Berkovich et al. 2011e).

1. Verfügbarkeit und Reife der Technologie. Diese Kategorie beinhaltet die Zyklenauslöser, die mit dem Erscheinen einer neuen Technologie verbunden sind. Das bedeutet, wenn eine neue Technologie auf den Markt kommt, kann diese Technologie zur Entwicklung domänenspezifischer Komponenten von PSS eingesetzt werden (vgl. (Dodgson et al. 2008, 55-90; Lamsweerde 2009, 222; Pohl 2007, 548-549; Ulrich/Eppinger 2003, 56)). Der Wechsel der Technologie kann zu Änderungen an Lösungsanforderungen an die disziplinspezifischen Komponenten führen, deren Realisierung auf der alten Technologie basiert. Daraufhin können sich weitere Änderungen sowohl an initialen Anforderungen als auch an den Komponenten des PSS ergeben. Es ist daher zu überprüfen, ob die initialen Kunden- und Stakeholderanforderungen, die durch die alte Technologie realisiert werden, durch die neue Technologie abgedeckt werden. Die Auswirkungen von möglichen Änderungen der initialen Anforderungen auf die domänenspezifischen Lösungsanforderungen sollen kontrolliert werden. Mögliche Konflikte sowohl zwischen den initialen Anforderungen als auch zwischen den Lösungsanforderungen sollen identifiziert und aufgelöst werden.

Das folgende Beispiel verdeutlicht, was passiert, wenn eine neue Hardware-Technologie verwendet wird¹⁴: Die Technologie Voice-over-IP (VoIP) ist erstmals auf dem Massenmarkt eingesetzt worden. Dabei bieten Provider breite Palette an Services und Angeboten, so dass Unternehmen und Kunden eine Umstellung auf neue Technologie erwägen. Das führte zu möglichen Wettbewerbsvorteilen für den Provider und möglichen Kosteneinsparungen für Kunden.

Als Ergebnis von solchen Zyklenauslösern entsteht ein Zyklus im Requirements Engineering. Um dem Zyklus entgegenzuwirken, sollen folgende Aktivitäten durchgeführt werden. So müssen die Lösungsanforderungen erneut analysiert werden. Falls sich aufgrund des Technologiewechsels bestimmte Lösungsanforderungen ändern, sollen die initialen Anforderungen, aus denen die Lösungsanforderungen abgeleitet wurden, identifiziert und auf mögliche Änderungen überprüft werden. Die mit den Lösungsanforderungen in Abhängigkeit stehenden An-

¹⁴<http://www.msnbc.msn.com/id/6354872/> zugegriffen am 27.04.09

forderungen an weitere Komponenten des PSS sind ebenso zu identifizieren und auf mögliche Änderungen zu überprüfen. Anschließend sind die geänderten Anforderungen zu dokumentieren und zu validieren. Zu beachten sind dabei mögliche Konflikte zwischen den Lösungsanforderungen an die Komponenten des PSS (vgl. (Tanabe et al. 2008)). Wenn sich bspw. die Anforderungen an die Hardware-Komponente ändern, können daraus Änderungen an die Software- und Dienstleistungskomponenten resultieren. Die oben beschriebenen Aktivitäten sind solange zu wiederholen, bis alle Konflikte sowohl zwischen den initialen Anforderungen an das PSS als auch zwischen den Lösungsanforderungen aufgelöst werden und somit die sich aufgrund des Technologiewechsels ergebenden Änderungen an Anforderungen aufgenommen und umgesetzt werden.

Abbildung 4-19 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch die Verfügbarkeit und Reife der Technologie.

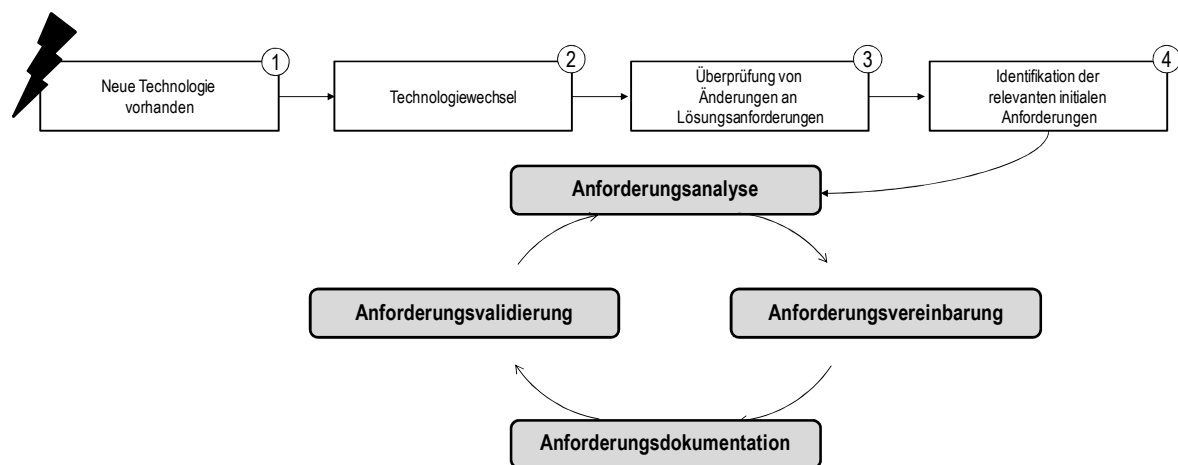


Abbildung 4-19: Zyklus im Requirements Engineering für PSS ausgelöst durch die Verfügbarkeit und Reife der Technologie

Quelle: In Anlehnung an Berkovich et al. (2011e)

2. Wettbewerbliche Entwicklungen. Diese Kategorie enthält mehrere Zyklenauslöser. Wenn das angebotene PSS nicht mehr vom Kunden benötigt wird, ist es auf veränderte Erwartungen oder Bedürfnisse des Kunden in Bezug auf das PSS zurückzuführen (wurde von Omikron und Theta im Rahmen der empirischen Studie (s. Kapitel 4.5.1) genannt). Oft kann der Kunde den Nutzen des PSS nicht mehr erkennen (Spath et al. 2001, 97-121). In diesem Fall ist zu klären, ob der Kunde noch grundsätzlich Interesse an der Lösung hat und welche Eigenschaften er von der Lösung erwartet, damit er das PSS wieder erwerben will. Beispielsweise hat der Computerhersteller Dell¹⁵ 2003 auf Diskettenlaufwerke bei der Ausstattung der Rechner verzichtet, indem er motiviert hat, dass das Verhalten der Kunden sich geändert hat und die Kunden nicht mehr bereit sind einen Aufpreis für Diskettenlaufwerke zu bezahlen, da sie sich stattdessen USB-Anschlüsse für USB-Speichersticks mit mehr Speicherplatz und angenehmerer Handhabung wünschten.

Eng verbunden damit ist ein weiterer Zyklenauslöser, wenn die angebotenen Dienstleistungen nicht mehr vom Kunden benötigt werden. Dieser Zyklenauslöser wurde von *Iota* und *Kappa* im Rahmen der empirischen Studie (s. Kapitel 4.5.1) genannt. Wenn der Kunde die Software,

¹⁵http://www.computerbase.de/news/hardware/komplettssysteme/2003/februar/dell_floppy-laufwerk (zugegriffen am 22.02.11)

die vom Auftragnehmer entwickelt und für den Kunden betrieben wird, selbst betreiben will, führt das zum Verlust des Umsatzes aus dem Betreiben der Software für den Auftragnehmer. In diesem Fall sollen die Gründe des Kunden für die gewünschte Änderung der vereinbarten Geschäftsbedingungen, d. h. für den Verzicht auf die Dienstleistung, geklärt werden.

Zu einem weiteren Zyklenauslöser gehört die Attraktivität des Konkurrenzproduktes. So werden oft von Unternehmen Produkte angeboten, die ein herkömmliches Produkt imitieren, aber einige charakteristische Merkmale des Produktes erweitern und dadurch ein qualitativ hochwertigeres Produkt anbieten (Fuchs/Apfelthaler 2008, 139-140; Spath et al. 2001, 121-124; Ulrich/Eppinger 2003, 55). Daraufhin wechselt der Kunde zum Konkurrenten. Auch *Iota* (s. Kapitel 4.5.1) nannte im Rahmen der empirischen Studie die Attraktivität des Konkurrenzproduktes als ein möglicher Zyklenauslöser. In diesem Fall soll geklärt werden, welche Merkmale des Konkurrenzproduktes für den Kunden mehr Attraktivität bieten, als die herkömmliche Lösung. Benchmarking ermöglicht in diesem Fall einen Vergleich mit dem Konkurrenzprodukt. Eine erneute Anforderungsermittlung hilft dabei die Anforderungen zu identifizieren, die zur Verbesserung des PSS beim Kunden führen.

Ebenso kann ein Zyklus im Requirements Engineering aufgrund eines neuen Trends auf dem Markt ausgelöst werden. Falls neue Trends beobachtet werden, die einen Einfluss auf das PSS haben, ist es wichtig den Kunden zu fragen, ob er das PSS an den neuen Trend anpassen will. In diesem Fall sind die Auswirkungen vom Trend auf die initialen Kundenanforderungen festzustellen. Wenn sich die Kundenanforderungen ändern, sind die Auswirkungen der Änderungen auf die Lösungsanforderungen zu identifizieren.

In allen beschriebenen Fällen ereignen sich die Zyklen aufgrund der veränderten Kundenanforderungen. Beim Auftreten solcher Zyklenauslöser sollen als erstes die Kundenanforderungen neu ermittelt werden. Anschließend sind sie zu analysieren, auf mögliche Konflikte zu überprüfen, zu dokumentieren und zu validieren. Die Aktivitäten sind so lange zu wiederholen, bis der Kunde sich sicher ist, dass seine Anforderungen durch das PSS richtig erfüllt sind. Abbildung 4-20 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch die wettbewerblichen Entwicklungen.

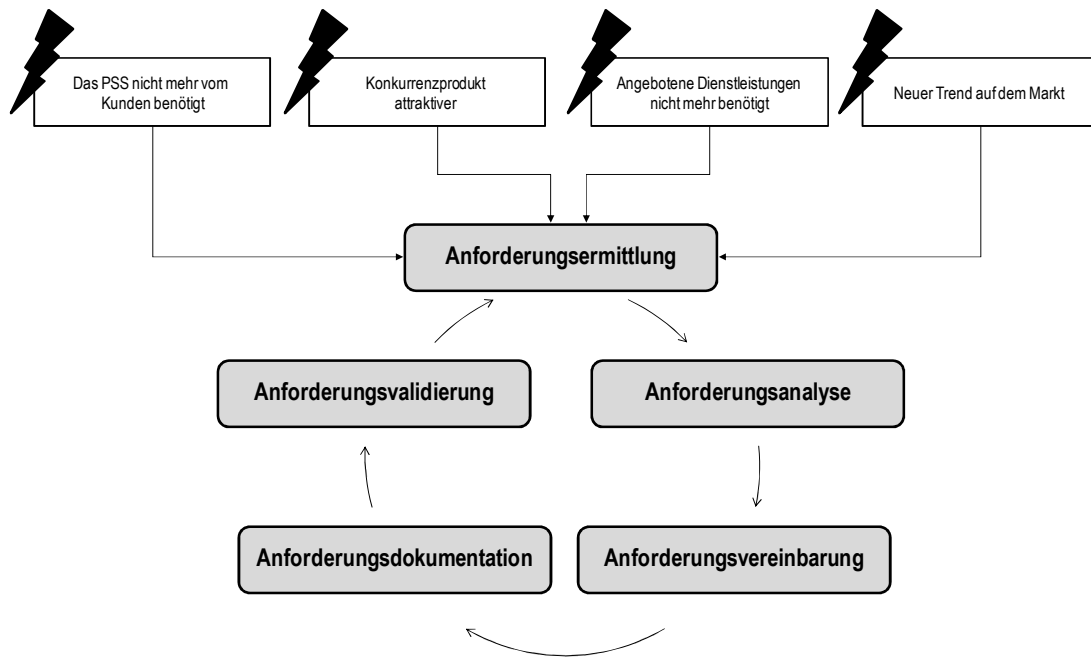


Abbildung 4-20: Zyklus im Requirements Engineering für PSS ausgelöst durch die wettbewerblichen Entwicklungen

Quelle: In Anlehnung an Berkovich et al. (2011e)

3. Unterschiedliche Lebenszyklen von Hardware- und Software-Komponenten des PSS.

Die einzelnen Komponenten des PSS, Hardware und Software, haben unterschiedliche Lebenszyklen (Lavagno/Passerone 2005; Berkovich et al. 2009a). Diese unterschiedlichen Lebenszyklen lassen sich bspw. bei Geldautomaten beobachten¹⁶. Wenn die Software vom Hersteller der Automaten nicht mehr unterstützt wird, muss daraufhin oft auch die Hardware ausgetauscht werden.

Die unterschiedliche zeitliche Alterung der einzelnen Komponenten führen zu Zyklen im Requirements Engineering. So führt die Alterung einer domänenspezifischen Komponente zu einem Austausch dieser Komponente. In diesem Fall sollen die Lösungsanforderungen an die alte Komponente in Bezug auf die neuen Bedingungen aufgrund des Komponentenaustausches analysiert werden. Es kann vorkommen, dass einige Lösungsanforderungen nicht mehr erfüllbar sind. In so einem Falle müssen die initialen Anforderungen, aus denen die domänenspezifischen Lösungsanforderungen abgeleitet werden, identifiziert werden. Die Auswirkung der Nicht-Erfüllung der Lösungsanforderungen auf initiale Anforderungen werden analysiert. Anschließend sind die Änderungen an den initialen Anforderungen zu dokumentieren und zu validieren. Ebenso sind nach Konflikten sowohl zwischen den initialen Anforderungen als auch zwischen den Lösungsanforderungen zu suchen. Die gefundenen Konflikte sind aufzulösen.

Abbildung 4-21 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch die Unterschiedliche Lebenszyklen von Hardware- und Software-Komponenten des PSS.

¹⁶<http://www.stern.de/wirtschaft/geld/87195.html> (zugegriffen am 05.09.2010)

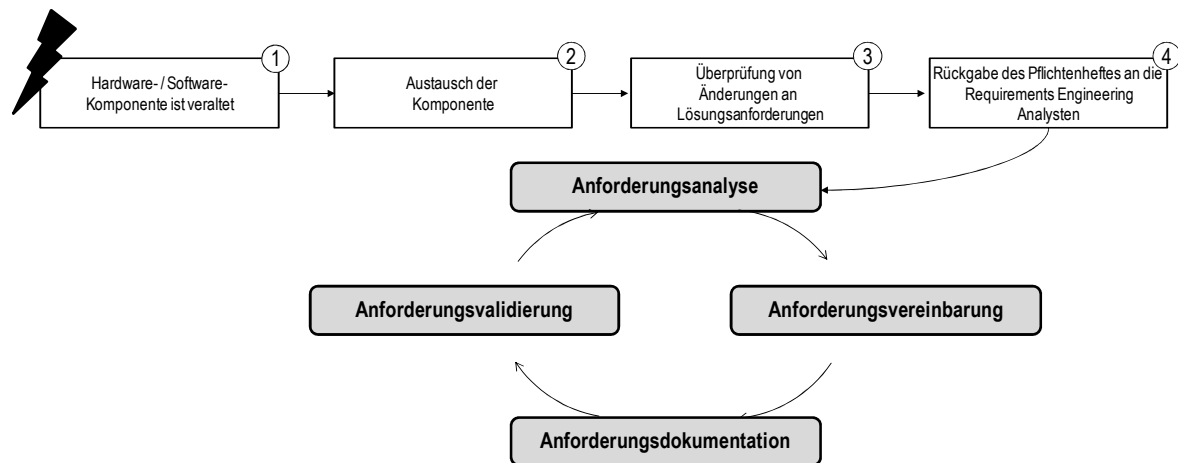


Abbildung 4-21: Zyklus im Requirements Engineering für PSS ausgelöst durch die Unterschiedliche Lebenszyklen von Hardware- und Software-Komponenten des PSS

Quelle: In Anlehnung an Berkovich et al. (2011e)

4. Änderungen von Kunden- und Stakeholderanforderungen. Die initialen Kunden- und Stakeholderanforderungen können sich ändern (Kotonya/Sommerville 1998, 123-126; Pohl 2007, 548; Nuseibeh/Easterbrook 2000; Ulrich/Eppinger 2003, 56), wodurch zyklische Wechselwirkungen im Requirements Engineering auftreten. Die Änderungen von Kundenanforderungen wurden auch von *Eta* und *Xi* im Rahmen der empirischen Studie als Zyklusursachen genannt (s. Kapitel 4.5.1). Solche Änderungen treten oft auf, wenn sich bspw. die Nutzung des Systems durch den Kunden oder die Stakeholder ändert (Pohl 2007, 548). So will der Kunde, dass im Winter die Raumtemperatur von der Heizungsanlage automatisch auf 23°C gestellt wird, während im Sommer 21°C gewünscht sind. In diesem Fall müssen die Anforderungen des Kunden neu erfasst werden, analysiert und auf mögliche Konflikte zu den bestehenden initialen Anforderungen sowie den Lösungsanforderungen an den Heizkörper, die Software, die die Temperatur reguliert, und die entsprechenden Dienstleistungen, wie Wartung, überprüft werden. Möglicherweise muss die Lösung komplett geändert werden, indem eine Klimaanlage eingeführt wird. Abbildung 4-22 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch die Änderungen an Kunden- und Stakeholderanforderungen.

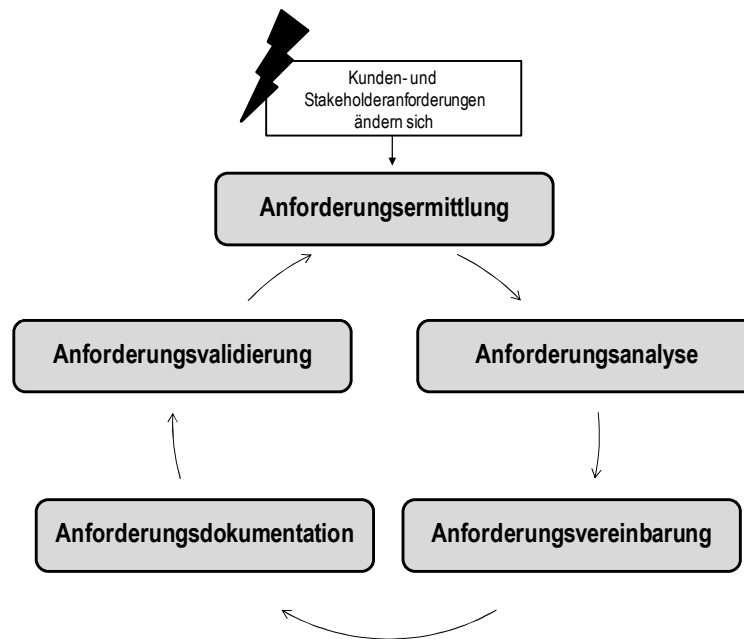


Abbildung 4-22: Zyklus im Requirements Engineering für PSS ausgelöst durch Änderungen an Kunden- und Stakeholderanforderungen
 Quelle: In Anlehnung an Berkovich et al. (2011e)

5. Finanzwirtschaftliche Änderungen. Diese Kategorie besteht aus zwei Zyklenauslösern. Der erste Zyklenauslöser ist, wenn das PSS sofort dem Kunden bereitgestellt werden soll, bspw. aufgrund der veränderten Projektanforderungen (Rupp 2004, 427). Dieser Zyklenauslöser wurde ebenso von Iota und Lambda im Rahmen der empirischen Studie genannt (s. Kapitel 4.5.1). Falls die Bereitstellung sich verzögert, kann der Kunde bspw. zum Konkurrenzprodukt wechseln (wurde von *Alpha* im Rahmen der empirischen Studie (s. Kapitel 4.5.1) genannt). Ein weiterer Zyklenauslöser ist, wenn der Preis des PSS für den Kunden zu hoch ist und deswegen reduziert werden soll, damit der Kunde die Lösung abnimmt (Fuchs/Apfelthaler 2008, 139; Rupp 2004, 427). Bei einer erforderlichen Preisreduzierung ist zu klären, an welchen Stellen Einsparungen und somit Einschränkungen möglich sind.

Sowohl bei der sofortigen Bereitstellung des PSS an den Kunden als auch der Preisreduktion ist es erforderlich die Anforderungen an das PSS neu zu analysieren. Abbildung 4-23 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch finanzwirtschaftliche Zyklen.

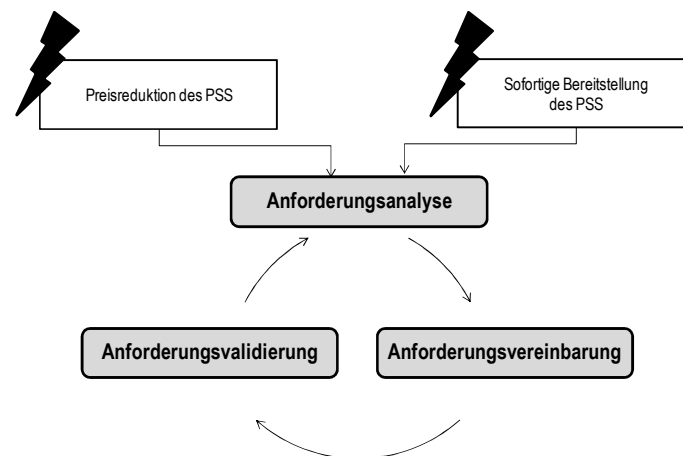


Abbildung 4-23: Zyklus im Requirements Engineering für PSS ausgelöst durch finanzwirtschaftliche Änderungen

Quelle: In Anlehnung an Berkovich et al. (2011e)

Im ersten Fall soll geklärt werden, ob alle Anforderungen durch die Vorverlegung des Termins der Bereitstellung erfüllt sind. Ebenso ist es wichtig, nach möglichen Konflikten zwischen den Anforderungen zu suchen. Im zweiten Fall ist ebenso wichtig durch die Anforderungsanalyse die Anforderungen zu identifizieren, die durch die Preisreduktion nicht erfüllt werden können. Diese Anforderungen sollen den Kunden und Stakeholdern präsentiert und von ihnen priorisiert werden. Anschließend soll nach möglichen Konflikten gesucht und die Anforderungen anschließend validiert werden.

6. Entwicklungsänderungen. Die Zyklusursachen dieser Kategorie entstehen aufgrund von Problemen bei der Realisierung der einzelnen Komponenten (Mellerowicz 1958, 188f.; Hauschildt 2004, 540; Rupp 2004, 427; Pohl 2007, 549). Diese Ursache wurde ebenso von Chi und Ny im Rahmen der empirischen Studie (s. Kapitel 4.5.1) genannt. Weitere Ursachen stellen die Differenzen im Ergebnis der Entwicklung der einzelnen Domänen, die nicht vorhersehbar waren (genannt von *Delta* (s. Kapitel 4.5.1)) und Änderungen von Anforderungen durch die internen Stakeholder dar (Lamsweerde 2009, 222). Falls eine domänenspezifische Komponente nicht realisierbar ist (bspw. die Software konnte nicht realisiert werden¹⁷), bedeutet das, dass die Möglichkeiten ausgewertet werden sollen, ob die davon betroffenen Anforderungen weglassen oder durch andere Komponenten realisiert werden können.

Die Differenzen im Ergebnis der Entwicklung entstehen, wenn die Anforderungen an eine domänenspezifische Komponente (bspw. Software) nicht erfüllt sind. Daraus resultiert, dass das PSS in seiner Gesamtheit den Kundenanforderungen nicht entspricht. In diesem Fall ist es notwendig, die Anforderungen zu identifizieren, die nicht erfüllt sind, um darauf basierend entweder mit dem Kunden zu verhandeln oder die Anforderungen durch andere Komponenten zu realisieren. Falls interne Stakeholder, bspw. Entwickler auf der Seite des Auftragnehmers ihre Anforderungen ändern, ist es wichtig zu überprüfen, in wieweit diese geänderten Anforderungen für die Erfüllung von Kundenanforderungen hinderlich sind.

Somit startet der Zyklus im Requirements Engineering mit der Analyse von Anforderungen, um zu bestimmen, welche Anforderungen nicht erfüllt werden können. Anschließend sind die

¹⁷ <http://www.heise.de/newsticker/Bericht-Software-Projekt-fuer-Finanzaeamter-gescheitert--/meldung/48843/>

Anforderungen den Kunden und Stakeholdern vorzustellen, möglicherweise neu zu priorisieren und auf Konflikte zu überprüfen. Abbildung 4-24 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch die Entwicklungszyklen.

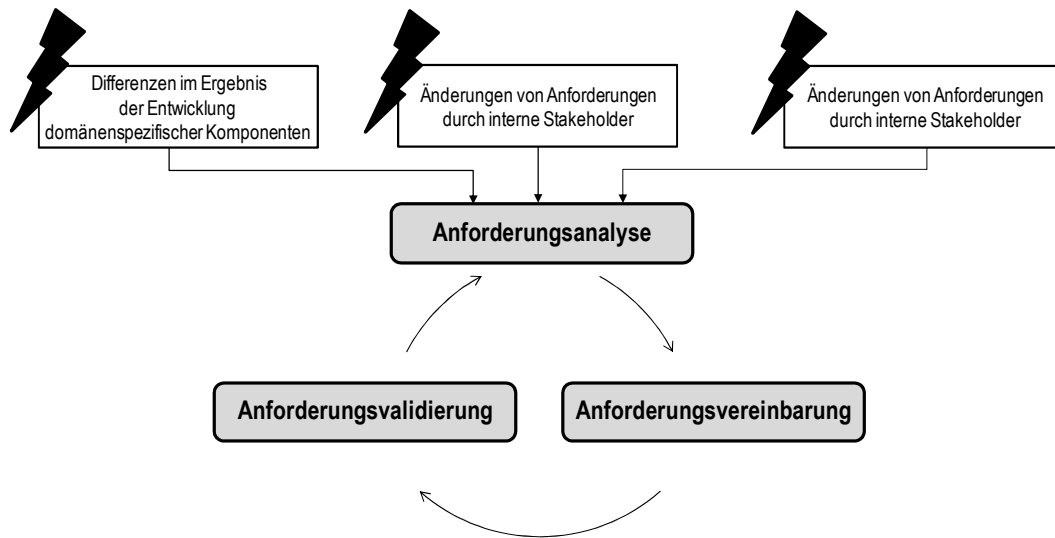


Abbildung 4-24: Zyklus im Requirements Engineering für PSS ausgelöst durch Entwicklungsänderungen

Quelle: In Anlehnung an Berkovich et al. (2011e)

7. Legislativen Änderungen. Die Änderungen an Gesetzen oder Richtlinien können zu Änderungen an Anforderungen führen (Lamsweerde 2009, 222; Rupp 2004, 427). Dadurch können sich sowohl initiale Anforderungen der Kunden und Stakeholder als auch Lösungsanforderungen an die Komponenten des PSS ändern. Falls sich ein Gesetz oder eine Norm ändert, müssen die Anforderungen identifiziert werden, die von den Änderungen betroffen sind. Anschließend sind diese Anforderungen zu analysieren, auf mögliche Konflikte zu überprüfen, falls erforderlich, zu dokumentieren und anschließend zu validieren. Abbildung 4-25 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch die legislativen Änderungen.

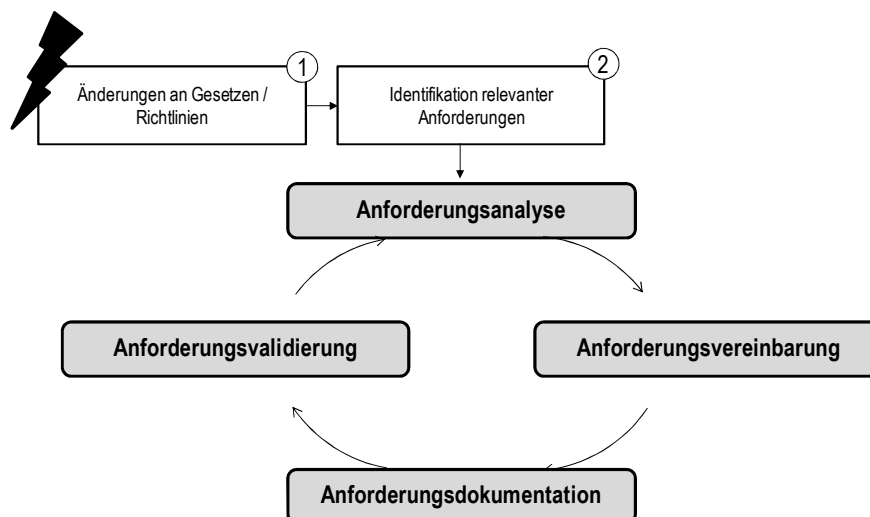


Abbildung 4-25: Zyklus im Requirements Engineering für PSS ausgelöst durch legislative Änderungen

Quelle: In Anlehnung an Berkovich et al. (2011e)

8. Abgleich von Anforderungen. Die vertiefende Fallstudie bei *Alpha* im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern (s. Kapitel 4.5.2) hat gezeigt, dass Iterationen zur Klärung von Anforderungen sowie der Aktualisierung von Lasten- und Pflichtenheften aufgrund von Missverständnissen zwischen der Entwicklung und den Requirements Engineering Analysten bezüglich der Lösungsanforderungen entstehen. Abbildung 4-26 zeigt schematisch den Zyklus im Requirements Engineering ausgelöst durch die Missverständnisse bei den Anforderungen.

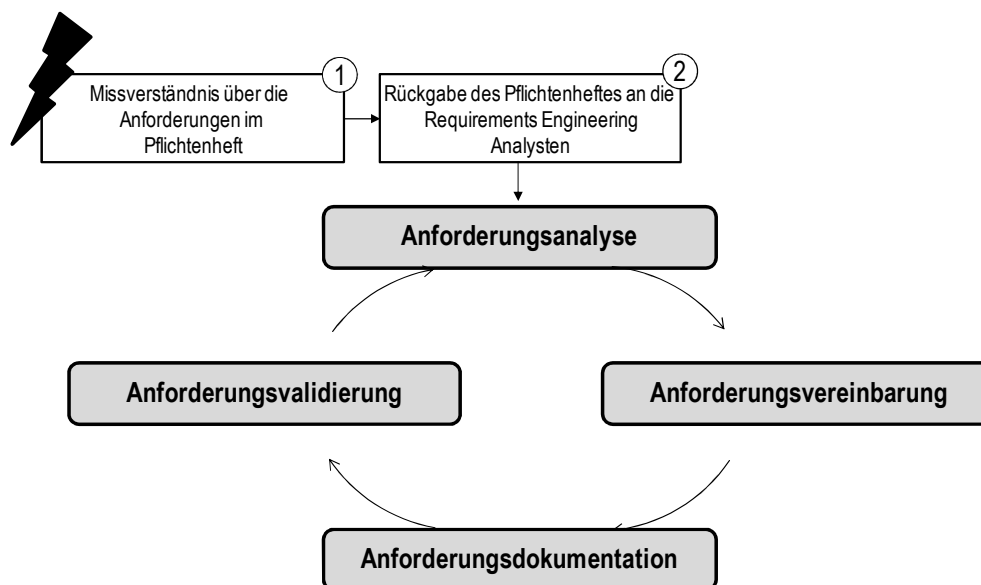


Abbildung 4-26: Zyklus im Requirements Engineering für PSS ausgelöst durch Missverständnisse über Anforderungen

Quelle: In Anlehnung an Berkovich et al. (2011e)

Die Missverständnisse beim Klären von Anforderungen entstehen, wenn die Anforderungen im Pflichtenheft nicht lösungsorientiert sind und deswegen von der Entwicklung nicht direkt umgesetzt werden können. In diesem Fall wird das Pflichtenheft zurück an die Requirements Engineering Analysten gegeben, damit sie die Anforderungen weiter konkretisieren können.

Die identifizierten Zyklen und ihre Ursachen erheben keinen Anspruch auf Vollständigkeit, sondern sind in Form eines morphologischen Kastens dargestellt, der fortlaufend erweitert und ergänzt werden darf. Die Zyklen im Requirements Engineering und ihre Ursachen legen die Grundlage zur Formulierung von Anforderungen an einen Requirements Engineering Ansatz für PSS, der den Zyklen gerecht werden soll.

4.7 Zusammenfassung: Anforderungen an einen Requirements Engineering Ansatz für Product Service Systems

Die systematische Literaturanalyse über die Eignung der domänenspezifischen sowie der integrierten Ansätze zum Requirements Engineering für PSS (Kapitel 4.4) zeigte, dass entsprechende Ansätze, die es ermöglichen, die Anforderungen an PSS integriert und domänenübergreifend zu behandeln, weitgehend fehlen. Die Vorgehen der einzelnen Domänen sind stark auf den jeweiligen Bereich zugeschnitten, während die integrierten Ansätze sich mit dem Requirements Engineering nur am Rande, im Rahmen der Entwicklungsprozesse, beschäftigen. Die Kooperation zwischen den Domänen – d.h. die Abstimmung der jeweiligen Ergebnisse und ihre anschließende Verwendung in einer anderen Domäne – wird beeinträchtigt.

Die Erkenntnisse der Literaturanalyse wurden durch zwei weitere Vorstudien ergänzt - eine Serie von qualitativen Experteninterviews (Kapitel 4.5.1) und eine vertiefende Fallstudie in einem Unternehmen, das Haushaltsgeräte herstellt (Kapitel 4.5.2). Die beiden Vorstudien zeigten, dass die Notwendigkeit eines systematischen Requirements Engineering, das die Sichten der beteiligten Domänen integriert, erkannt wurde. Jedoch fehlen entsprechende Vorgehen weitgehend. Die größten Herausforderungen bereiten die Analyse von Anforderungen, um die Anforderungen zu konkretisieren und an die domänenspezifischen Komponenten des PSS aufzuteilen, die Anforderungsvereinbarung, die die Konflikte zwischen den Anforderungen an die domänenspezifischen Komponenten des PSS identifizieren und auflösen soll, sowie Zyklen, die aufgrund der Missverständnisse über die Anforderungen zwischen der Entwicklung und den Requirements Engineering Analysten entstehen und somit zu erhöhten Kosten und Aufwand sowie einer verspäteten Bereitstellung des PSS an den Kunden führen.

Das Requirements Engineering für PSS soll es ermöglichen, dass die an der Entwicklung beteiligten Domänen ein übergreifendes Verständnis über die zu erbringende Lösung aufbauen. Die initialen unspezifischen Anforderungen an die gesamte Lösung sollen so konkretisiert werden, dass sie die Sichten der Domänen widerspiegeln und anschließend nahtlos an die Entwicklung übergeben werden können.

Basierend auf den Erkenntnissen aus der systematischen Literaturanalyse (Kapitel 4.4), der ersten (Kapitel 4.5.1) und zweiten Vorstudien (Kapitel 4.5.2), sowie den dabei identifizierten Zyklen (Kapitel 4.6), wurden die Anforderungen an einen Requirements Engineering Ansatz für PSS abgeleitet, die im Folgenden beschrieben werden.

Zusätzlich zu den Erkenntnissen aus der Literaturanalyse und den zwei Vorstudien flossen in die Aufstellung von Anforderungen an einen Requirements Engineering Ansatz für PSS weitere im Zusammenhang mit den spezifischen Eigenschaften von PSS und den Aktivitäten des Requirements Engineering für PSS stehenden Herausforderungen ein:

1. Die erste Herausforderung stellt die *Kommunikation zwischen den an der Entwicklung beteiligten Personen und Unternehmensbereichen* dar (Davis 1982). Solche Unternehmensbereiche können Marketing, Entwickler oder Requirements Engineering Analysten sein, die unterschiedliche Vorstellungen und unterschiedliches Hintergrundwissen haben (Brugger 2005, 55). Eine bekannte Weise, um die Kommunikation zu verbessern ist es, über ein Artefakt zu kommunizieren (Dix 1994).
2. Eine weitere Herausforderung im Requirements Engineering ist die *Unterschiedlichkeit und Komplexität der Anforderungen*, die zu Schwierigkeiten in der Strukturierung der Anforderungen führt (Byrd et al. 1992; Davis 1992). Stakeholder drücken ihre Anforderungen üblicherweise auf unterschiedlichen Abstraktionsebenen aus. Kunden stellen ihre Anforderungen oft in Form von Zielen, die ein Produkt erfüllen soll, dar. Diese Anforderungen sind unpräzise und beschreiben die zu entwickelnde Lösung abstrakt. Manager drücken ihre Anforderungen in Form von Geschäftszielen und allgemeinen Erfordernissen aus. In Gegensatz dazu haben Entwickler eine sehr konkrete, technische Sicht und drücken ihre Anforderungen dementsprechend detailliert aus. Die Aufgabe des Requirements Engineering ist es, die Verbindung zwischen den sehr konkreten Anforderungen und den Anforderungen auf einer allgemeineren Ebene zu finden. Auf diese Weise wird sichergestellt, dass die allgemeinen Anforderungen durch die konkretisierten domänenspezifischen Lösungsanforderungen erfüllt werden (Nikora 2005). Die Verbindung zwischen den allgemeinen Anforderungen und den Lösungsanforderungen zu kennen, ist auch deswegen erforderlich, um den Einfluss von Änderungen am PSS auf die initialen Anforderungen ermitteln zu können und um dadurch das Treffen von Entscheidungen bzgl. der realisierenden Änderungen unterstützen zu können (Berenbach et al. 2009, 14f.).
3. Eine dritte Herausforderung im Requirements Engineering ist die *konzeptionelle Lücke zwischen den Anforderungen und der Entwicklung*. Das Requirements Engineering muss den Übergang von den Anforderungen in das Design des Produktes unterstützen (Nikora 2005). Dazu müssen die initialen Anforderungen „in die Sprache des Entwicklers übersetzt“ werden und es muss überprüft werden dass alle Anforderungen korrekt verstanden werden (Jacobson et al. 1999; Adler 1994). PSS stellen komplexe Systeme dar, die nicht unabhängig von Entwicklungsentscheidungen spezifiziert werden können. Einer detaillierten Anforderungsspezifikation liegen explizite und implizite Entwurfsentscheidungen zu Grunde (Pohl 2007, 565). Ein nahtloser Übergang von den Anforderungen in die Entwicklung stellt sicher, dass die Anforderungen von der Entwicklung richtig und korrekt aufgenommen werden und somit das entwickelte Produkt die initialen Anforderungen erfüllt. Ebenso können Iterationen zwischen der Entwicklung und dem Requirements Engineering zur Klärung von Anforderungen verhindert werden, die zu einem steigenden Kosten- und Ressourcenaufwand und zu einer verspäteten Bereitstellung des Produktes an den Kunden führen können. Somit ist die Einbindung des Requirements Engineering in den Entwicklungsprozess unabdingbar (Pohl/Sikora 2007a; Pohl/Sikora 2007b).

Diese Herausforderungen verdeutlichen die Problematik, die mit Hilfe eines Requirements Engineering Ansatzes adressiert werden soll. Somit ergeben sich folgende Anforderungen an einen Requirements Engineering Ansatz für PSS. Das Requirements Engineering soll dabei nicht nur am Anfang des Entwicklungsprozesses, bevor die eigentliche Entwicklung angefangen hat, sondern über die Phasen der Entwicklung erfolgen. Die Entwicklungsinformationen

sollen zur Ermittlung und Verwaltung von Anforderungen einbezogen werden, um dadurch die Architektur des PSS und die Anforderungen schon in den frühen Phasen der Entwicklung abzustimmen.

Anforderung 1: Domänenübergreifende Kooperation. Die Zusammenarbeit zwischen den unterschiedlichen an der Entwicklung beteiligten Domänen muss schon während der ersten Phasen der Entwicklung im Requirements Engineering unterstützt werden (s. Kapitel 4.4.2.5). Dazu muss eine gemeinsame Begriffswelt aufgebaut werden, die in den unterschiedlichen Domänen sehr verschieden sein kann (Berkovich et al. 2009b). Zusätzlich müssen verschiedene domänenspezifische Sichten auf die Anforderungen in das Requirements Engineering integriert werden, damit die beteiligten Domänen die Ergebnisse der Requirements Engineering Aktivitäten gegenseitig verstehen und für weitere Aktivitäten der Entwicklung verwenden können (s. Kapitel 4.4.2.5). Somit soll es ermöglicht werden, dass die beteiligten Domänen ein gemeinsames Verständnis über das zu lösende Problem aufbauen und die einzelnen Schritte im Requirements Engineering abgestimmt durchführen. Dadurch lassen sich auch die Zyklen, die aufgrund von unterschiedlichen Erwartungen bei den Ergebnissen der Komponententwicklung der einzelnen Domänen entstehen (s. Kapitel 4.6), minimieren.

Daraus lässt sich die folgende Anforderungen an den Requirements Engineering Ansatz für PSS ableiten: *Der Requirements Engineering Ansatz für PSS muss in der Lage sein, die Kooperation zwischen den an der Entwicklung von PSS beteiligten Domänen zu ermöglichen, damit die Domänen ein übergreifendes und integriertes Verständnis über das zu lösende Problem aufbauen können und die Ergebnisse der Requirements Engineering Aktivitäten gegenseitig verstehen und nutzen können.*

Anforderung 2: Durchgängige Kommunikation der Anforderungen an den Kunden und die Stakeholder. Laut der Ergebnisse der Literaturanalyse spielen der Kunde sowie weitere Stakeholder eine wichtige Rolle nicht nur bei der Abnahme eines Produktes, sondern schon im Requirements Engineering. Der Kunde soll über die Abnahme des PSS entscheiden. Dafür muss er überprüfen, wie gut seine Anforderungen an das PSS erfüllt sind. Ein frühzeitiges Feedback vom Kunden muss ermöglicht werden, um seine Vorstellungen und Wünsche besser zu verstehen. Die im Rahmen der Literaturanalyse betrachteten Ansätze erkennen die Rolle des Kunden für das Requirements Engineering, beschränken jedoch seine Teilnahme weitgehend auf die Ermittlung seiner Anforderungen und Wünsche. Zudem findet die Kundeneinbindung in das Requirements Engineering unstrukturiert statt (s. Kapitel 4.4.2.5). Eine durchgängige Kommunikation der Anforderungen an den Kunden im Requirements Engineering soll laut der Expertenmeinungen in der ersten Vorstudie (s. Kapitel 4.5.1.2.7) die Iterationen zwischen den Stakeholdern und den Requirements Engineering Analysten bei der Suche nach Konflikten zwischen den Anforderungen minimieren. Das Verständnis der Kunden und Stakeholder bzgl. der Anforderungen bei der Validierung soll dadurch gesteigert werden. Dadurch wird gesichert, dass die entwickelte Lösung den initialen Kunden- und Stakeholderanforderungen entspricht.

Daraus lässt sich die folgende Anforderung an den Requirements Engineering Ansatz für PSS ableiten: *Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen während der Phasen des Requirements Engineering an den Kunden und die Stakeholder zu kommunizieren und damit ihr Verständnis über die zu entwickelnde Lösung zu fördern.*

Anforderung 3: Bereitstellung eines Vorgehens zur Anforderungsermittlung. Der Kunde stellt die Anforderungen an das gesamte PSS, nicht an die einzelnen Sach- und Dienstleistungskomponenten des PSS. Dabei ist insbesondere die Ermittlung von impliziten Anforderungen an die Dienstleistungen herausfordernd (s. Kapitel 4.4.2.5). Es wird ein integriertes Vorgehen zur Ermittlung von Anforderungen benötigt, das sicherstellt, dass die Anforderungen an die Sach- und Dienstleistungen vollständig erhoben werden. Ebenso soll das Vorgehen die Abstimmung von Anforderungen bereits in der Anforderungsermittlung unterstützen, um dadurch möglichst präzise und konfliktfreie Anforderungen an das PSS zu liefern. Dies ist insbesondere im Kontext der Interdisziplinarität von PSS erforderlich (s. Kapitel 4.5.1.2.1), damit die Domänen ein übergreifendes Verständnis über die Anforderungen bekommen. Die ermittelten Anforderungen sollen eine Art „Wissensdatenbank“ darstellen, aus der die Anforderungen während der Entwicklung entnommen werden (s. Kapitel 4.5.2.2.5). Desweiteren sollen bei der Anforderungsermittlung verschiedene Quellen der Anforderungen, wie bspw. Kunden, interne Stakeholder oder relevante Wertschöpfungsprozesse des Kunden, in die das PSS integriert wird, berücksichtigt werden.

Zusammenfassend lässt sich folgende Anforderung formulieren: *Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen an PSS integriert zu ermitteln, so dass auch implizite Anforderungen erhoben werden können und alle Quellen von Anforderungen berücksichtigt werden.*

Anforderung 4: Bereitstellung eines Vorgehens zur Analyse von Anforderungen an PSS. Die initialen Anforderungen an das PSS sind lösungsneutral und sehr unspezifisch. Die einzelnen Komponenten der Lösung, wie Hardware, Software und Dienstleistungen, werden aber von den Domänen, Produkt-, Software- und Dienstleistungsentwicklung, konzipiert und integriert entwickelt. Daraus folgt, dass die initialen Anforderungen konkretisiert werden müssen, um den einzelnen Domänen konkrete Angaben zu den zu entwickelnden Komponenten zu geben. Sowohl die Wichtigkeit hiervon, als auch das Fehlen von entsprechenden Ansätzen wurde in der Literaturanalyse, den Interviews und in den Fallstudien gezeigt: (1) In der Literaturanalyse in Kapitel 4.4.2.5 wurde gezeigt, dass die domänenspezifischen und die integrierten Ansätze aus der Literatur keine Vorgehen zur Analyse und Übersetzung von initialen Anforderungen in die Lösungsanforderungen an die Komponenten des PSS bieten. (2) Im Rahmen der ersten Vorstudie in Kapitel 4.5.1.2.2 wurden die Konkretisierung und die damit im Zusammenhang stehende Priorisierung von Anforderungen an PSS als wichtige Herausforderungen für das Requirements Engineering genannt. So äußert der Kunde die Anforderungen an die ganze Lösung und nicht an die einzelnen Sach- und Dienstleistungskomponenten. (3) Die zweite Vorstudie in Kapitel 4.5.2.2.5 zeigte, dass in der Praxis Vorgehen zur Konkretisierung der Anforderungen „in die Sprache der Entwickler“ weitgehend fehlen. Wünschenswert wäre es, dass das Requirements Engineering eine Art „Wissensdatenbank“ für Anforderungen zur Verfügung stellt, die initiale Anforderungen und aus ihnen abgeleiteten Lösungsanforderungen enthält. Dabei sollen die einzelnen Schritte der Analyse nachvollziehbar sein.

Aus diesen Erkenntnissen folgt die Anforderung an einen Requirements Engineering Ansatz für PSS: *Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die initialen Anforderungen an das PSS schrittweise zu konkretisieren, so dass sie den einzelnen domänenspezifischen Komponenten des PSS zugeordnet werden können.*

Anforderung 5: Bereitstellung eines Vorgehens zur Anforderungsvereinbarung. Oft kommt es zu Konflikten zwischen Anforderungen, die dadurch charakterisiert sind, dass die

Anforderungen sich widersprechen. Solche Konflikte können sowohl zwischen den Anforderungen an eine domänenspezifische Komponente vorkommen, also innerhalb einer Domäne, als auch zwischen den Anforderungen an verschiedene domänenspezifische Komponenten (wie zwischen den Anforderungen an die Hardware- und Dienstleistungskomponente), also zwischen verschiedenen Domänen. Die Identifikation und Auflösung von Konflikten ist im Rahmen des Zyklusmanagements im Requirements Engineering wichtig (s. Kapitel 4.6). Durch die in Kapitel 4.6 erläuterten Zyklusursachen können sich die Anforderungen ändern, was zu möglichen Konflikten zwischen den Anforderungen führt. Damit die Identifikation und die Auflösung von Konflikten erfolgen können, sollen die Anforderungen über einen vergleichbaren Detaillierungsgrad verfügen (s. Kapitel 4.5.2.2.5, Kapitel 4.5.1.2.3). Die domänenspezifischen Ansätze sind zur Suche und Auflösung von Konflikten zwischen den Anforderungen an verschiedene domänenspezifische Komponenten weitgehend nicht geeignet (s. Kapitel 4.4.2.5).

Daraus lässt sich die folgende Anforderung an einen Requirements Engineering Ansatz für PSS formulieren: ***Der Requirements Engineering Ansatz für PSS muss es ermöglichen, Konflikte zwischen den Anforderungen sowohl domäneninternen als auch domänenübergreifend zu identifizieren und aufzulösen.***

Anforderung 6: Bereitstellung eines Vorgehens zur Anforderungsdokumentation. Die systematische Literaturanalyse zeigte, dass die Vorgehen zu Dokumentation, die in den einzelnen Domänen sowie durch die integrierten Ansätze zur Entwicklung von PSS vorgeschlagen sind, zur Dokumentation von Anforderungen an PSS ausreichend sind (s. Kapitel 4.4.2.5). Jedoch muss die Dokumentation von Anforderungen im Falle von PSS auch einen nahtlosen Übergang vom Lastenheft ins Pflichtenheft erlauben, so dass die dokumentierten Informationen im Lastenheft im Pflichtenheft wiederzufinden sind (s. Kapitel 4.5.1.2.4). Desweiteren sollen die domänenspezifischen Informationen, wie bspw. die Angaben zu den benötigten Ressourcen zur Erbringung von Dienstleistungen, mit dokumentiert werden.

Daraus folgt: ***Der Requirements Engineering Ansatz für PSS muss Vorgehen anbieten, um die Anforderungen sowie die domänenspezifischen Informationen, die zur Umsetzung der Anforderungen benötigt werden, durchgängig zu dokumentieren.***

Anforderung 7: Bereitstellung eines Vorgehens zur Anforderungsverfolgung. Die domänenspezifischen Vorgehen bieten weitgehend keine Möglichkeit, um die Interdependenzen zwischen den Anforderungen domänenübergreifend, d. h. zwischen den verschiedenen Komponenten des PSS, festzustellen (s. Kapitel 4.4.2.5). Dies ist jedoch insbesondere für einen adäquaten Umgang mit den Änderungen an Anforderungen erforderlich. Ändert sich eine Anforderung, so ist es erforderlich anhängige Anforderungen zu identifizieren, da diese auch von den Änderungen betroffen sein könnten. Um die abhängigen Anforderungen zu identifizieren, sind die Interdependenzen zwischen den Anforderungen festzuhalten. Dadurch kann ebenso der Umgang mit den Zyklen im Requirements Engineering verbessert werden, da diese sehr häufig Änderungen an Anforderungen verursachen (s. Kapitel 4.6). Desweiteren soll das Requirements Engineering die Verfolgung von Anforderungen über die Phasen der Entwicklung bis zu den daraus resultierenden Lösungsanforderungen im Pflichtenheft ermöglichen und nicht nur die Verbindungen zwischen den initialen Anforderungen und den domänenspezifischen Lösungsanforderungen aufzeigen (s. Kapitel 4.5.2.2.5).

Somit lautet die Anforderungen an einen Requirements Engineering Ansatz für PSS: ***Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen an PSS***

von ihrem Ursprung über die Phasen der Entwicklung bis zu den domänenspezifischen Lösungsanforderungen zu verfolgen sowie die Interdependenzen zwischen den Anforderungen innerhalb einer Domäne und auch zwischen den Anforderungen, die zu verschiedenen Domänen gehören, aufzuzeigen.

Anforderung 8: Bereitstellung eines Vorgehens zum Änderungsmanagement. In engem Zusammenhang mit der Anforderungsverfolgung steht das Änderungsmanagement an Anforderungen. Die Ansätze, die im Rahmen der systematischen Literaturanalyse identifiziert wurden, sind nicht ausreichend, um die Änderungen an Anforderungen adäquat zu behandeln (s. Kapitel 4.4.2.5). So müssen die Auswirkungen von Änderungen an domänenspezifischen Lösungsanforderungen auf weitere Lösungsanforderungen sowohl innerhalb der betrachteten Domäne als auch domänenübergreifend und auf initiale Anforderungen an das gesamte PSS identifiziert werden (s. Kapitel 4.5.1.2.6, Kapitel 4.5.2.2.5). Desweiteren sind Auswirkungen von Änderungen an initialen Anforderungen auf Lösungsanforderungen zu identifizieren.

Daraus lässt sich folgern: *Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Änderungen an Anforderungen aufzunehmen und die Auswirkungen von Änderungen auf domänenspezifische Lösungsanforderungen und auf initiale Anforderungen an die gesamte Lösung zu identifizieren.*

Anforderung 9: Bereitstellung eines Vorgehens zur Anforderungvalidierung. Bevor die Anforderungen an die Entwicklung weitergeleitet werden, sollen sie an den Kunden und die Stakeholder kommuniziert werden, um dadurch die Qualität von Anforderungen zu sichern. Die in der Literatur identifizierten Ansätze zur Anforderungvalidierung sind weitgehend geeignet, um für PSS angewendet zu werden. Allerdings muss dabei die Validierung der Lösungsanforderungen, zu denen die Anforderungen an Sach- und Dienstleistungskomponenten des PSS gehören, berücksichtigt werden (s. Kapitel 4.4.2.5). In der ersten und in der zweiten Vorstudie (s. Kapitel 4.5.1.2.5, Kapitel 4.5.2.2.5) wurde auch erwähnt, dass die Validierung nicht nur am Ende des Requirements Engineering Prozesses erfolgen muss, wenn die Anforderungen bereits im Pflichtenheft festgehalten sind, sondern fortlaufend während den Entwicklungsphasen bis zu ihrer Übernahme ins Pflichtenheft, um dadurch die Qualität der Anforderungen zu steigern.

Daraus folgt: *Der Requirements Engineering Ansatz für PSS muss Vorgehen anbieten, um die Anforderungen über die Phasen der Entwicklung bis zu ihrer Übernahme ins Pflichtenheft fortlaufend zu validieren.*

Anforderung 10: Bereitstellung eines Vorgehens zur Wiederverwendung. In der ersten und in der zweiten Vorstudie (s. Kapitel 4.4.2.5 und Kapitel 4.5.2.2.5) wurde erwähnt, dass die Anforderungen wiederverwendet werden sollen. Die Wiederverwendung ermöglicht bereits dokumentierte Anforderungen aus anderen Projekten zu verwenden und den Aufwand zur Ermittlung und Analyse von Anforderungen zu minimieren (Robertson/Robertson 2007, 304). Zudem können durch die Wiederverwendung die aus den Änderungen an Kundenanforderungen oder an Lösungsanforderungen resultierenden Zyklen reduziert werden, da die nicht von den Änderungen betroffenen Anforderungen nicht erneut ermittelt und analysiert werden müssen, sondern wiederverwendet werden können (s. Kapitel 4.6). Als wesentliche Voraussetzung für die Wiederverwendung von Anforderungen wurde die Unterstützung der Modularisierung erkannt. Aufgrund des modularen Aufbaus von PSS (s. Kapitel 3.1.2) ist es wichtig, dass die Modularisierung schon im Rahmen des Requirements Engineering unterstützt wird (s. Kapitel 4.4.2.5).

Daraus lässt sich die folgende Anforderung an einen Requirements Engineering Ansatz für PSS formulieren: *Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen wiederzuverwenden.*

Anforderung 11: Festlegung des Geltungsbereichs für legislative Anforderungen. Die Änderungen von Gesetzen und Richtlinien führen oft zu Änderungen an Anforderungen, die sich an Gesetzen und Richtlinien orientieren (s. Kapitel 4.6). Somit soll jede legislative Anforderung, die von solchen Änderungen betroffen werden kann, einen Geltungsbereich bekommen, der angibt, wann die Anforderung auf Aktualität überprüft werden soll. Da diese legislativen Änderungen zu Zyklen im Requirements Engineering führen können, ist es besonders wichtig, solche Anforderungen früh zu identifizieren und zu kennzeichnen.

Daraus lässt sich die folgende Anforderung an einen Requirements Engineering Ansatz für PSS formulieren: *Der Requirements Engineering Ansatz für PSS soll es ermöglichen, den legislativen Anforderungen einen Geltungsbereich zuzuordnen, um mögliche Änderungen zu antizipieren.*

Tabelle 4-10 gibt einen Überblick über die Anforderungen an einen Requirements Engineering Ansatz für PSS.

Anforderung	Beschreibung
Anforderung 1: Domänenübergreifende Kooperation	Der Requirements Engineering Ansatz für PSS muss in der Lage sein, die Kooperation zwischen den an der Entwicklung von PSS beteiligten Domänen zu ermöglichen, damit die Domänen ein übergreifendes und integriertes Verständnis über das zu lösende Problem aufbauen können und die Ergebnisse der Requirements Engineering Aktivitäten gegenseitig verstehen und nutzen können.
Anforderung 2: Durchgängige Kommunikation der Anforderungen an den Kunden und die Stakeholder	Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen während der Phasen des Requirements Engineering an den Kunden und die Stakeholder zu kommunizieren und damit ihr Verständnis über die zu entwickelnde Lösung zu fördern.
Anforderung 3: Bereitstellung eines Vorgehens zur Anforderungsermittlung	Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen an PSS integriert zu ermitteln, so dass auch implizite Anforderungen erhoben werden können und alle Quellen von Anforderungen berücksichtigt werden.
Anforderung 4: Bereitstellung eines Vorgehens zur Analyse von Anforderungen an PSS	Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die initialen Anforderungen an das PSS schrittweise zu konkretisieren, so dass sie den einzelnen domänenspezifischen Komponenten des PSS zugeordnet werden können.
Anforderung 5: Bereitstellung eines Vorgehens zur Anforderungsvereinbarung	Der Requirements Engineering Ansatz für PSS muss es ermöglichen, Konflikte zwischen den Anforderungen sowohl domäneninternen als auch domänenübergreifend zu identifizieren und aufzulösen.
Anforderung 6: Bereitstellung eines Vorgehens zur Anforderungsdokumentation	Der Requirements Engineering Ansatz für PSS muss Vorgehen anbieten, um die Anforderungen sowie die domänenspezifischen Informationen, die zur Umsetzung der Anforderungen benötigt werden, durchgängig zu dokumentieren.

<p>Anforderung 7: Bereitstellung eines Vorgehens zur Anforderungsverfolgung</p>	<p>Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen an PSS von ihrem Ursprung über die Phasen der Entwicklung bis zu den domänenspezifischen Lösungsanforderungen zu verfolgen sowie die Interdependenzen zwischen den Anforderungen innerhalb einer Domäne und auch zwischen den Anforderungen, die zu verschiedenen Domänen gehören, aufzuzeigen.</p>
<p>Anforderung 8: Bereitstellung eines Vorgehens zum Änderungsmanagement</p>	<p>Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Änderungen an Anforderungen aufzunehmen und die Auswirkungen von Änderungen auf domänenspezifische Lösungsanforderungen und auf initiale Anforderungen an die gesamte Lösung zu identifizieren.</p>
<p>Anforderung 9: Bereitstellung eines Vorgehens zur Anforderungvalidierung</p>	<p>Der Requirements Engineering Ansatz für PSS muss Vorgehen anbieten, um die Anforderungen über die Phasen der Entwicklung bis zu ihrer Übernahme ins Pflichtenheft fortlaufend zu validieren.</p>
<p>Anforderung 10: Bereitstellung eines Vorgehens zur Wiederverwendung</p>	<p>Der Requirements Engineering Ansatz für PSS muss es ermöglichen, die Anforderungen wiederzuverwenden.</p>
<p>Anforderung 11: Festlegung des Geltungsbereichs für legislative Anforderungen</p>	<p>Der Requirements Engineering Ansatz für PSS soll es ermöglichen, den legislativen Anforderungen einen Geltungsbereich zuzuordnen, um mögliche Änderungen zu antizipieren.</p>

Table 4-10: *Anforderungen an einen Requirements Engineering Ansatz für Product Service Systems*

Quelle: Eigene Darstellung

5 Requirements Engineering Ansatz für Product Service Systems

Das vorliegende Kapitel beschäftigt sich mit den Bestandteilen des Requirements Engineering Ansatzes für PSS, die in Form eines Bezugsrahmens vorgestellt werden. Der Requirements Engineering Ansatz ist basierend auf den Anforderungen aus Kapitel 4.7 konzipiert worden, denen eine systematische Literaturanalyse und zwei Vorstudien in Form von qualitativen Interviews und einer vertiefenden Fallstudie in einem Unternehmen zu Grunde liegen. In Kapitel 5.1 wird die Lösungsidee zur Realisierung der Anforderungen an einen Requirements Engineering Ansatz für PSS präsentiert. In Kapitel 5.2 werden existierende Requirements Engineering Ansätze erläutert und die Wahl für den artefakt-basierten Requirements Engineering Ansatz begründet. In Kapitel 5.3 wird der Bezugsrahmen beschrieben, der die wesentlichen Teile des Requirements Engineering Ansatzes für PSS definiert und beschreibt.

5.1 Lösungsidee zur Realisierung der Anforderungen an einen Requirements Engineering Ansatz für Product Service Systems

Aus den Anforderungen (Kapitel 4.7) wird ersichtlich, dass der Requirements Engineering Ansatz entsprechende Aktivitäten zur Ermittlung und Verwaltung von Anforderungen an PSS anbieten soll. Desweiteren sollen Techniken angeboten werden, die die Aktivitäten des Requirements Engineering unterstützen, auf PSS zugeschnitten sind und die oben beschriebenen Anforderungen erfüllen.

Allerdings können durch das Anbieten von Requirements Engineering Aktivitäten sowie entsprechenden Techniken bestimmte Forderungen nicht gerecht bedient werden. Zu diesen Forderungen gehören vor allem die Herausforderungen, die in Kapitel 4.7 beschreiben wurden. Die Verfolgung der schrittweisen Konkretisierung von Anforderungen in Anlehnung an den Entwicklungsprozess und die Einbindung des Requirements Engineering in den Entwicklungsprozess sind dadurch nicht möglich. Weiterhin ergibt sich aus der Existenz von Anforderungen mit unterschiedlichem Detaillierungsgrad und unterschiedlicher Komplexität die Notwendigkeit der Strukturierung von Anforderungen, die nicht alleine durch die Definition von Aktivitäten angegangen werden kann.

Eine weitere Herausforderung stellt die Kommunikation zwischen den an der Entwicklung von PSS beteiligten Domänen dar. Das Ziel einer Kommunikation ist ein Austausch von Informationen zur aufgabenbezogenen Verständigung über den Kooperationsgegenstand und über bestehende Abhängigkeiten (Zerbe 1998, 212) in Anlehnung an (Reichwald 1993). Allerdings ist die domänenübergreifende Kommunikation im Requirements Engineering nicht alleine durch Prozessdefinitionen lösbar. Für die Kooperation und die Kommunikation spielt ein gemeinsames Material, „das Kooperationspartnern gemeinsam zur Erreichung eines Kooperationszieles zur Verfügung steht“ (Zerbe 1998, 213), eine wichtige Rolle. Das gemeinsame Material ist allen am Informationsaustausch beteiligten Parteien zugänglich, so dass sie die Informationen teilen statt sie verteilen (Zerbe 1998, 214). Abbildung 5-1 zeigt die Kooperation von zwei an der Kooperation beteiligten Parteien, die über ein gemeinsames Material (Modell) kommunizieren.

Zu den Vorteilen eines gemeinsamen Materials zählt seine Aufbewahrungsmöglichkeit. So können der Sender und der Empfänger einer Information auf eine gemeinsame Basis zurück-

greifen und dadurch ihre Wahrnehmung in Form eines mentalen Modells von der erhaltenen oder gesandten Information stärken. Somit wird das gemeinsame Material zum Träger der auszutauschenden Information (Zerbe 1998, 214). Und das mentale Modell stellt dabei eine Abbildung der erhaltenen Information dar. Ein gemeinsames Material ermöglicht „ein gemeinschaftliches Verständnis über die zu erfüllende Aufgabe zu schaffen, indem es verschiedene Sichten auf den Problemgegenstand erlaubt und die Zusammenarbeit der Teammitglieder auf den gerade aktuellen Informationsstand des Materials fokussiert“ (Zerbe 1998, 218).

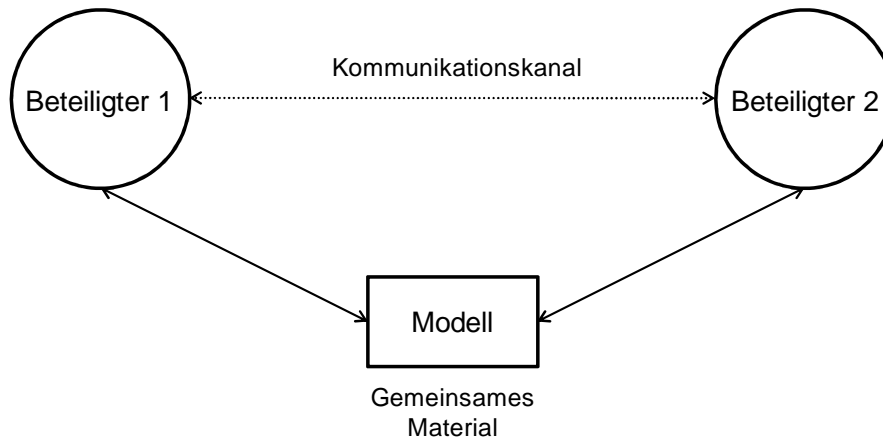


Abbildung 5-1: Kooperation am gemeinsamen Material

Quelle: (Zerbe 1998, 214) in Anlehnung an (Schrage 1990, 98)

Einen ähnlichen Vorschlag zur Gestaltung der Kommunikation macht auch Dix (1994). Er schlägt vor, die Kommunikation zwischen den Beteiligten anhand eines (Arbeits-) Artefakts zu gestalten. Dabei versteht er unter einem Artefakt eine Information, die „greifbar“ präsentiert ist. Abbildung 5-2 zeigt die Gestaltung der Kommunikation anhand eines Artefakts.

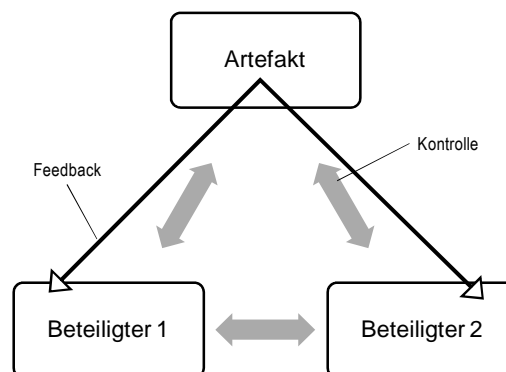


Abbildung 5-2: Gestaltung der Kommunikation zwischen den Beteiligten anhand eines Artefakts

Quelle: In Anlehnung an Dix (1994)

Durch die Einführung eines Artefaktes in die Kommunikation wird den Beteiligten die Möglichkeit gegeben, den Informationsaustausch über das Artefakt, das in diesem Fall ein gemeinsames Material darstellt, zu gestalten, und somit die auszutauschenden Informationen besser zu kontrollieren, gegenseitig zu validieren und ihre Anmerkungen bzgl. der Information direkt im Artefakt abzulegen. Beispiele für Artefakte sind Anforderungsspezifikationen oder Abbildungen.

Da das Ergebnis des Requirements Engineering dokumentierte Anforderungen sind, ist es daher wichtig, dass allen am Requirements Engineering beteiligten Personen die Anforderungen in strukturierter Weise vorliegen. Zur Lösung der in Kapitel 4.7 beschriebenen Herausforderungen muss der Requirements Engineering Ansatz deswegen zusätzlich zu den Aktivitäten und Techniken, Verfahren zur Strukturierung von Anforderungen und somit zum Informationsaustausch und zur gegenseitigen Validierung von Entscheidungen anbieten.

Zerbe (1998) geht in seinem Ansatz über den reinen Austausch des gemeinsamen Material hinaus, und schlägt auch vor, Planungs- und Projektmanagementinformationen als gemeinsamen zugängliches Material zur Verfügung zu stellen (Zerbe 1998, 225). Dies resultiert in seiner Aufteilung in ein „Modell zur Problemlösung“, das das eigentliche Arbeitsergebnis der Kooperation darstellt, und zum anderen in einem „Modell zur Koordination“, das Planungs- und Abhängigkeitsinformationen enthält.

Diese beiden Vorschlägen von Zerbe (1998) und Dix (1994) werden im Ansatz zum Requirements Engineering für PSS in dieser Arbeit aufgegriffen. Zum einen lässt sich schließen, dass es ein gemeinsames „Modell zur Problemlösung“ benötigt wird, das alle relevanten Informationen beinhaltet. Dieses Modell wird durch das Bereitstellen eines Artefaktmodells für Anforderungen gegeben. Das Artefaktmodell bietet eine einheitliche Strukturierungsvorgabe für Anforderungen an, auf dessen Basis die einzelnen Domänen gemeinsam an den Anforderungen arbeiten können. Das „Modell zur Koordination“ wird in Form eines Vorgehensmodells und Technikbaukastens zur Verfügung gestellt. Es beschreibt in welcher Reihenfolge die Artefakte des Artefaktmodells von den beteiligten Kooperationspartnern erstellt werden. Damit legt es auch die Abhängigkeiten und die zeitliche Reihenfolge der Arbeitsschritte fest.

Das Prinzip die zu erarbeitenden Anforderungen im Requirements Engineering in Form eines Artefaktmodells zu beschreiben, wird vom artefakt-orientierten Requirements Engineering vorgeschlagen. Es konzentriert sich auf die Ergebnisse der Requirements Engineering Aktivitäten und definiert, welche Arten von Anforderungen dokumentiert werden müssen (Geisberger 2005; Berenbach 2006; Pohl/Sikora 2007a; Gorschek/Wohlin 2006). Als Artefakte werden dabei dokumentierte Anforderungen oder Entwicklungsinformationen bezeichnet, die zur Ermittlung von Anforderungen benötigt werden. Die Artefakte werden Abstraktionsebenen zugeordnet, die sich an bestimmte Ereignisse im Entwicklungsprozess, bspw. die Phasen der Entwicklung, anlehnen. Durch diese Zuordnung besteht die Möglichkeit den Lebenszyklus einer Anforderung schrittweise zu verfolgen, mögliche Änderungen von Anforderungen und die Auswirkungen dieser Änderungen auf weitere Anforderungen schnell festzustellen. Im Zentrum der artefakt-orientierten Ansätze steht ein Artefaktmodell (Méndez Fernández et al. 2010). Ein Artefaktmodell stellt ein Schema zur Strukturierung von Anforderungen dar, allerdings gibt es keine Vorgehen und Methoden zur Erstellung der Artefakte vor. Das artefakt-orientierte Requirements Engineering eignet sich gut, um die Anforderungen zu ermitteln, zu analysieren, zu konkretisieren und zu validieren, indem die Artefakte die Quellen und die Strukturierungsrichtlinien für die Anforderungen definieren. Desweiteren verfügen diese Ansätze über einen hohen Grad an Flexibilität, da die Artefakte, abhängig von den Bedürfnissen des Projektes gewählt werden können.

Basierend auf der Lösungsidee sollen die existierenden Ansätze zum Requirements Engineering analysiert und bezüglich der Erfüllung der an den Requirements Engineering Ansatz gestellten Anforderungen (Kapitel 4.7) geprüft werden.

5.2 Abgrenzung von existierenden Requirements Engineering Ansätzen

In diesem Kapitel werden bekannte Requirements Engineering Ansätze vorgestellt (Unterkapitel 5.2.1 bis 5.2.5). Anschließend werden die Vor- und Nachteile dieser Ansätze in Bezug auf die Anforderungen an einen Requirements Engineering Ansatz für PSS diskutiert. Dabei werden die Gründe für die Wahl eines artefakt-orientierten Requirements Engineering Ansatzes aufgezeigt.

5.2.1 Szenario-basiertes Requirements Engineering

Das Ziel des szenario-basierten Requirements Engineering ist es, die Anforderungen an die konkrete Systemanwendung zu beschreiben. Ein Szenario ist eine exemplarische Sequenz von Ereignissen und Aktionen, die die Interaktion zwischen dem Benutzer und dem System beschreiben (Anderson/Durney 1993). Szenarien dienen als „Intermediärabstraktion zwischen konzeptuellen Modellen einerseits und der Realität andererseits“ (Pohl 2007, 120). Szenarien werden eingesetzt, um ökonomische, soziale und technische Aspekte des zu entwickelnden Systems zu behandeln (Jarke et al. 1999). Seit den 1980er Jahren wurden Szenarien im Bereich der Human-Computer-Interaction (HCI) verwendet, um Schlüsselsituationen der Benutzung zu beschreiben. Später wurden sie auch im Bereich des Requirements Engineerings eingesetzt (Jacobson 1995), um effizient mit den Stakeholdern zu kommunizieren. Laut Jarke et al. (1999) können Szenarien dabei helfen, die Interaktion zwischen dem Benutzer und dem Entwickler gezielt auf das zu entwickelnde System zu lenken, indem beschrieben wird, wie der Benutzer mit dem System umgehen soll. Ebenso können die Szenarien die Aufgaben des Benutzers beschreiben. Dadurch wird geklärt, auf welche Weise das System den Benutzer unterstützt und welche Ziele das System erfüllt. Im Requirements Engineering werden Szenarien bei der Erhebung von Anforderungen eingesetzt (Dano et al. 1996; Lamsweerde 2009, 65-70; Robertson/Robertson 2007, 135f.). Eine Form von Szenarien stellen Use Cases dar. Unified Modeling Language (UML) definiert sog. Use-Case Diagramme, die dazu dienen einen Überblick über die Benutzer und deren Umgang mit dem System zu geben.

5.2.2 Ziel-orientiertes Requirements Engineering

Das ziel-orientierte Requirements Engineering stellt die Anforderungen in Form von Zielen dar (Mylopoulos 2006; Lamsweerde 2001). Das ziel-orientierte Requirements Engineering geht auf die Arbeit von Dardenne et al. (1993) zurück und wird von Lamsweerde (2009) umfassend erweitert. Lamsweerde (2003) definiert ein Ziel als „a prescriptive statement of intent about some system (existing or to-be) whose satisfaction in general requires the cooperation of some of the agents forming that system“. Somit drückt ein Ziel einen Zustand aus, der durch das zu entwickelnde System und dessen Umgebung erreicht werden soll. Der bekannteste ziel-orientierte Ansatz KAOS (vgl. (Lamsweerde 2009; Darimont et al. 1997)) ist vollständig formalisiert. Zum Ausdrücken von Anforderungen auf einer hohen Abstraktionsebene werden abstrakte „high-level“ Ziele eingesetzt. Diese beschreiben Geschäftsziele, die nicht exakt gefasst werden können und daher informell als Text beschrieben werden. Im Gegensatz dazu gibt es „low-level“ Ziele, die mathematisch formalisiert werden können und durch Temporallogik ausgedrückt werden. In der Erhebung von Anforderungen anhand von Zielen werden Zielbäume eingesetzt. Ein Zielbaum entsteht dadurch, dass Ziele durch UND und ODER Beziehungen verfeinert werden. Ein Ziel wird durch eine AND Beziehung verfeinert, wenn alle Unterziele erfüllt sein müssen, um das Oberziel zu erfüllen. Bei der ODER Beziehung muss mindestens ein Unterziel erfüllt sein. Der Ansatz eignet sich gut zur Anforderungser-

mittlung und Anforderungsanalyse. Ein Problem bei der Anforderungsverfeinerung durch Ziele stellt die Frage dar, wieweit Ziele verfeinert werden müssen. Es existieren keine allgemein gültigen Regeln, wann ein Ziel ausreichend detailliert ist. Eine weitere Schwäche ist, dass die Vollständigkeit der Anforderungen nur sehr schwer überprüft werden kann. Es gibt keine Kriterien, die festlegen, ob alle wichtigen Ziele gefunden worden sind. Die mathematische Formalisierung der Anforderungen bringt jedoch den Vorteil mit sich, dass Konflikte zwischen Anforderungen durch mathematische Methoden gefunden werden können. Auch die Verifikation von Anforderungen kann mit Hilfe von mathematischen Methoden erfolgen.

5.2.3 Aspekt-orientiertes Requirements Engineering

Ein weiterer Ansatz stellt das aspekt-orientierte Requirements Engineering dar. Dabei werden die ermittelten Anforderungen in spezielle Kategorien eingeteilt. Dafür werden die relevanten Problembereiche der Stakeholder, die sog. Aspects, identifiziert, gemäß denen die Anforderungen in Kategorien aufgeteilt werden (Rashid/Chitchyan 2008). Zusätzlich zur Aufteilung der Anforderungen in Kategorien, wird festgehalten, wie sich diese Kategorien gegenseitig beeinflussen. Durch die Zusammenfassung von Anforderungen, die sich auf dasselbe Thema/Problem beziehen, in Kategorien sollen die Konflikte zwischen Anforderungen leichter gefunden werden (Rashid/Chitchyan 2008; Sampaio et al. 2007). Wenn sich eine Anforderung ändert, dann sind größtenteils nur die Anforderungen innerhalb derselben Kategorie betroffen. Wenn dennoch andere Kategorien betroffen sind, so kann man diese über die Abhängigkeiten zwischen den Kategorien herausfinden. Das aspekt-orientierte Requirements Engineering ist vor allem für die Behandlung von Anforderungen geeignet, die sich auf viele Teile des Systems auswirken. Daher wird es in der Softwareentwicklung vor allem für die nicht-funktionalen Anforderungen eingesetzt. Ein typischer *Aspect* ist bspw. Sicherheit. Anforderungen an die Sicherheit eines Systems haben Auswirkungen auf alle Funktionen des Systems, sowie auf alle Daten, die das System verarbeitet. Das aspekt-orientierte Requirements Engineering ist vor allem auf die Analyse und das Finden von Konflikten beschränkt. Eine der Herausforderungen des aspekt-orientierten Requirements Engineering ist die Einteilung von Anforderungen in klare Kategorien (Rashid/Chitchyan 2008). Wenn eine Anforderung nicht eindeutig in eine Kategorie eingeordnet werden kann, dann ergeben sich Schwierigkeiten bei der Identifikation von Konflikten zwischen dieser und anderen Anforderungen und schwierige Verhandlungen mit den Stakeholdern bzgl. dieser Anforderung.

5.2.4 Viewpoint-orientiertes Requirements Engineering

Bei der Entwicklung eines Systems spielen verschiedenartige Einflüsse eine Rolle, bspw. sind unterschiedliche Stakeholder an der Anforderungsermittlung beteiligt, die Umgebung des zu entwickelnden Systems muss verstanden werden, die Anwendungsdomäne muss bei der Entwicklung berücksichtigt werden und der Entwicklungsprozess selbst stellt Einschränkungen an die Anforderungen dar. Das viewpoint-orientierte Requirements Engineering versucht diese Einflüsse zu erfassen und zu analysieren, indem es gezielt verschiedene Blickwinkel (engl. Viewpoint) auf das System einsetzt (Sommerville/Sawyer 1997b; Sommerville et al. 1998, 1999; Kotonya/Sommerville 1996). Ein Viewpoint ist eine Sammlung von Informationen über das System, ein dazu in Verbindung stehendes Problem oder die Umgebung des Systems. Das viewpoint-orientierte Requirements Engineering kann als ein Konzept der Analyse von Anforderungen betrachtet werden. Die Betrachtung eines Gegenstandes aus verschiedenen Blickwinkeln, um verschiedene Aspekte zu analysieren, beschränkt sich nicht nur auf das Requirements Engineering. Diese Technik wird bspw. im Systems Engineering allgemein

eingesetzt (Viewpoint-oriented Systems Engineering (VOSE)-Methode) (Kotonya/Sommerville 1998).

5.2.5 Prozess-orientiertes Requirements Engineering

Klassische Ansätze des Requirements Engineering geben Aktivitäten zur Durchführung des Requirements Engineering vor (vgl. Sommerville und Kotonya (1998), Robertson und Robertson (2007), Pohl (2007)). Allerdings fehlen oft die Verbindungen zu den Anforderungsdokumenten. Dadurch werden die Validierung und die schrittweise Konkretisierung erschwert. Die wesentlichen Aktivitäten des Requirements Engineering sind im Rahmen der vergleichenden Literaturanalyse in Kapitel 4.4.2.2 zu finden.

5.2.6 Fazit

Die meisten Ansätze zum Requirements Engineering verfolgen einen prozess-orientierten Ansatz. Das bedeutet, sie beschreiben den Prozess der Anforderungsermittlung und -verwaltung indem sie konkrete Aktivitäten zur Durchführung vorschreiben. Diese Sicht auf das Requirements Engineering stellt einen wichtigen Teil eines Requirements Engineering Ansatzes dar, da konkrete Handlungsanweisungen in Form von Aktivitäten und Techniken wichtig für Anwender in der Praxis sind. Die Anforderungen 3 bis 10 aus Kapitel 4.7 lassen sich daher über einen prozess-orientierten Ansatz behandeln. Allerdings hat sich herausgestellt, dass die reine Definition von Aktivitäten in der Praxis nicht ausreichend ist, sondern eine Beschreibung der Ergebnisse der Aktivitäten wesentlich zum Erfolg eines Ansatzes in der Praxis beiträgt (Gorschek/Davis 2008). Zusammenfassend lässt sich feststellen, dass ein Vorgehensmodell ein wesentlicher Teil eines Requirements Engineering Ansatzes ist und daher auch Teil des in dieser Arbeit entwickelten Ansatzes ist. Jedoch kann ein reines Vorgehensmodell nicht alle Herausforderungen des Requirements Engineerings lösen.

Die szenario-basierten Requirements Engineering Ansätze beschäftigen sich vor allem mit der Kommunikation zwischen den Stakeholdern und den Entwicklern. Daher sind sie besonders gut für die Anforderungserhebung geeignet. Für alle anderen Aufgaben des Requirements Engineering stellen sie jedoch keine Hilfestellung bereit. Insbesondere die Fragestellung der Dokumentation (Anforderung 6 aus Kapitel 4.7) und die Herausforderungen der Strukturierung der Anforderungen und Einbindung des Requirements Engineerings in den Entwicklungsprozess werden durch die szenario-basierten Ansätze nicht adressiert. In dieser Arbeit werden die Ansätze des szenario-basierten Requirements Engineerings als eine von mehreren Möglichkeiten zur Erhebung von Anforderungen aufgefasst.

Die ziel-orientierten Ansätze des Requirements Engineering stellen die Anforderungen in Form von Zielen dar, die vom System erreicht werden sollen. Diese Ziele werden durch definierte Beziehungen verfeinert. Somit eignen sich diese Ansätze gut für die Anforderungsermittlung. Die Überprüfung der Vollständigkeit von Anforderungen sowie die Anforderungvalidierung erweisen sich als schwierig. Zusammenfassend kann gesagt werden, dass das ziel-orientierte Requirements Engineering vor allem in den ersten Phasen der Anforderungserhebung erfolgreich eingesetzt wird. Dort lässt es den Stakeholdern die nötige Freiheit, um innovative Anforderungen zu finden. Allerdings gibt es keine Strukturierungsprinzipien für die Anforderungen vor, wodurch die Analyse der Anforderungen und deren Konkretisierung zu Lösungsanforderungen nur unzureichend unterstützt werden. Ebenso ist die Anwendung die-

ser Ansätze bei einer großen Anzahl von Anforderungen mit Schwierigkeiten verbunden, da die Ziele nicht mehr übersichtlich dargestellt werden können. In dieser Arbeit werden die Prinzipien des ziel-orientierten Requirements Engineering in den ersten Phasen des Requirements Engineering zur Darstellung von abstrakten Kundenwünschen und Geschäftszielen eingesetzt.

Die Ansätze des aspekt-orientierten Requirements Engineering beschäftigen sich mit der Frage, wie Anforderungen in Kategorien eingeteilt werden können und wie die Abhängigkeiten zwischen diesen Kategorien gefunden werden können. Ein großes Problem des aspekt-orientierten Requirements Engineering ist, dass es keine festen Kategorien definiert, sondern die Kategorien (genannt Aspekte) individuell festlegt. Daher bietet es nur unzureichende Unterstützung bei den verschiedenen Aufgaben des Requirements Engineering. Die Prinzipien des aspekt-orientierten Ansatzes werden in dieser Arbeit nicht übernommen. Im Gegensatz dazu werden in dieser Arbeit fertige Kategorien und Taxonomien vorgegeben.

Das viewpoint-orientierte Requirements Engineering verfolgt die Idee, Anforderungen an ein System als verschiedene Einflüsse auf das zu entwickelnde System darzustellen. Das bedeutet, um Anforderungen zu ermitteln und zu konkretisieren wird das System aus verschiedenen Blickwinkeln betrachtet. Das viewpoint-orientierte Requirements Engineering bleibt im Allgemeinen sehr abstrakt und liefert keine konkrete Hilfestellung zur schrittweisen Konkretisierung von Anforderungen und zu dessen Strukturierung. Die Idee des viewpoint-orientierten Requirements Engineering findet in dieser Arbeit nicht direkt Verwendung. Jedoch werden in dieser Arbeit feste Kategorien von Anforderungen vorgeschrieben, die verschiedene Blickwinkel auf das Produkt beinhalten.

Zusammenfassend kann festgestellt werden, dass die einzelnen Ansätze zum Requirements Engineering jeweils Stärken in verschiedenen Bereichen des Requirements Engineering haben. Keiner dieser Ansätze bietet jedoch eine umfassende Lösung der Probleme und Herausforderungen im Requirements Engineering für PSS an. Einzelne Prinzipien dieser Ansätze können jedoch in einen artefakt-orientierten Ansatz integriert werden. Beispielsweise werden die Prinzipien des szenario-orientierten Requirements Engineering in Form einer Technik in den Ansatz in dieser Arbeit aufgenommen.

5.3 Lösungskonzept für Requirements Engineering für Product Service Systems

Aus den Überlegungen, angeführt in den Kapiteln 5.1 und 5.2, sowie den Anforderungen an einen Requirements Engineering Ansatz für PSS aus Kapitel 4.7 lässt sich schließen, dass der zu konzipierende Requirements Engineering Ansatz mehrere Bestandteile enthält, die in diesem Kapitel in Form eines Bezugsrahmens definiert und beschrieben werden.

5.3.1 Bezugsrahmen zur Beschreibung des Lösungskonzepts für das Requirements Engineering für Product Service Systems

Ein Bezugsrahmen soll das Verständnis und die Bewertung von wesentlichen Begriffen und Konzepten eines Bereichs ermöglichen (vgl. (Rössl 1990; Zaugg 2002)). Somit lässt sich ein System durch einen Bezugsrahmen eindeutig beschreiben, indem alle zum Verständnis des Systems notwendigen Betrachtungsperspektiven definiert werden. Laut Rössl (1990, 100f.) wird ein prä-strukturierender heuristischer Bezugsrahmen eingesetzt, wenn es um noch nicht untersuchte Bereiche geht und das für die Untersuchung benötigte Verständnis gefördert wer-

den soll. Der Bezugsrahmen ist abstrakt gehalten und soll dadurch allgemeine Gültigkeit erreichen. Der Bezugsrahmen für das Requirements Engineering für PSS beschreibt die wesentlichen Teile des Requirements Engineering Ansatzes für PSS und definiert Begriffe und Konzepte, die vorgeben, wie das Requirements Engineering zu gestalten ist (Berkovich et al. 2010b). Somit definiert der Bezugsrahmen die Struktur des Requirements Engineering Ansatzes für PSS, dessen konkrete Ausgestaltung anhand der in Kapitel 4.7 vorgestellten Anforderungen entwickelt wurde.

5.3.2 Beschreibung des Bezugsrahmens für das Requirements Engineering für Product Service Systems

Laut den Vorüberlegungen, die in Kapiteln 5.1 und 5.2 bezüglich eines möglichen Lösungsansatzes zur Gestaltung des Requirements Engineering für PSS angeführt sind, eignet sich besonders das artefakt-orientierte Requirements Engineering, in dessen Zentrum ein Artefaktmodell steht, um die in Kapitel 4.7 erläuterten Herausforderungen im Requirements Engineering für PSS zu adressieren. Dabei zielt die Konstruktion eines Artefakts (in diesem Fall der Requirements Engineering Ansatz für PSS, der sich aus einem Artefaktmodell und einem Vorgehensmodell zusammensetzt) auf die „Lösung von Problemen mit IS-Bezug“ ((Hevner et al. 2004, 87) zitiert in (Gericke/Winter 2009, 196)). In diesem Fall soll der Requirements Engineering Ansatz die Entwicklung von PSS unterstützen.

Ein im Rahmen des artefakt-orientierten Requirements Engineering zu konzipierende Artefaktmodell stellt dabei das eigentliche Artefakt im Sinne von Dix (1994) dar (s. Kapitel 5.1), das die Kommunikation zwischen den Unternehmensbereichen und den Domänen, die an der Entwicklung von PSS beteiligt sind, unterstützt (vgl. (Berenbach et al. 2009, 21)). Denn ein Artefaktmodell enthält Artefakte, die dokumentierte Anforderungen oder Entwicklungsinformationen sind, und strukturiert sie anhand der Schritte des Entwicklungsprozesses. Somit können die Unternehmensbereiche oder Domänen in jedem Schritt der Entwicklung auf die einzelnen Informationen zurückgreifen und sie gegenseitig iterativ validieren und verifizieren.

Ebenso leistet das artefakt-orientierte Requirements Engineering einen Beitrag zum Schließen der konzeptionellen Lücke bei der Übergabe von Anforderungen an die Entwicklung. Durch die Angabe von Artefakten und die Herleitung von Anforderungen unter Einbeziehung der Entwicklungsinformationen lassen sich die Anforderungen schrittweise konkretisieren, indem sie abhängig vom Fortschritt der Entwicklung immer detaillierter werden und technische Ausprägungen erhalten. Desweiteren lassen sich die Anforderungen validieren, indem in jedem Schritt der Entwicklung anhand der Artefakte überprüft wird, ob alle Anforderungen korrekt verstanden werden.

Dementsprechend enthält der Requirements Engineering Ansatz für PSS ein Artefaktmodell, das es ermöglicht die Kommunikation zwischen den an der Entwicklung von PSS beteiligten Domänen sowie zwischen den Entwicklern und den Kunden zu unterstützen, die Anforderungen zu analysieren, schrittweise zu konkretisieren, die Konfliktsuche und Konfliktauflösung zu ermöglichen und damit eine Grundlage für die Validierung von Anforderungen für die Stakeholder und für die Entwickler zu schaffen. Ebenso sollen die Anforderungen über die Phasen der Entwicklung schrittweise verfolgt werden, so dass es möglich ist, die Auswirkungen von Änderungen an Anforderungen schnell zu erkennen.

Zusätzlich zu einem Artefaktmodell sollen ein Vorgehensmodell, das Aktivitäten zu Erstellung und Verwaltung von Artefakten und zur Durchführung weiterer Requirements Enginee-

ring Aktivitäten anbietet, konzipiert werden. Zusätzlich zu den Aktivitäten werden Techniken benötigt, die konkret beschreiben, was in einer Aktivität gemacht wird. Die Beziehungen zwischen den einzelnen Bestandteilen werden im Bezugsrahmen (Abbildung 5-3) dargestellt.

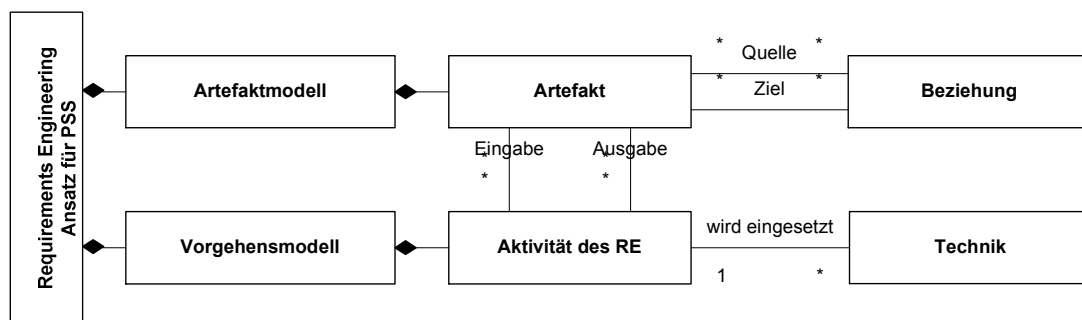


Abbildung 5-3: Bezugsrahmen für das Requirements Engineering für PSS

Quelle: In Anlehnung an Berkovich et al. (2010b)

Alle Bestandteile des Requirements Engineering Ansatzes sollen zur Erfüllung von in Kapitel 4.7 beschriebenen Anforderungen beitragen. Auf die Erfüllung von Anforderungen durch den Requirements Engineering Ansatz wird in der Evaluation (Kapitel 8.3) eingegangen.

Das Artefaktmodell für die Anforderungen an PSS besteht aus Artefakten, die die Spezifikationsergebnisse der Entwicklungsaktivitäten oder der Requirements Engineering Aktivitäten darstellen (Geisberger 2005, 67). Diese Artefakte sind miteinander durch Beziehungen verbunden. Das Vorgehensmodell bietet Aktivitäten an, die die Artefakte des Artefaktmodells erstellen und verwalten; das bedeutet eine Aktivität erhält Artefakte als Eingaben und erstellt Artefakte als Ausgaben. Die Requirements Engineering Techniken werden zur Unterstützung der Requirements Engineering Aktivitäten eingesetzt, indem sie genau angeben, wie eine Aktivität auszuführen ist.

Im Rahmen der Arbeit werden keine strategischen Ziele, organisatorischen Anforderungen oder Anforderungen an den Entwicklungsprozess betrachtet, weil sie nicht zum Betrachtungsgegenstand des Requirements Engineering gehören, sondern im Rahmen des Projektmanagement behandelt werden (Wiegiers 2005, 44).

6 Artefaktmodell für die Anforderungen an Product Service Systems

Kapitel 6 widmet sich der Erläuterung des Artefaktmodells und seiner Bestandteile. In Kapitel 6.1 werden die Grundlagen des Artefaktmodells für die Anforderungen an PSS gelegt. In Kapitel 6.1.1 werden die begrifflichen Grundlagen aufgebaut, indem die zum Verständnis des Artefaktmodells nötigen Begriffe und Konzepte erläutert werden. Anschließend werden in Kapitel 6.1.2 die existierenden Ansätze zum artefakt-orientierten Requirements Engineering vorgestellt und von dem entwickelten Ansatz abgegrenzt. Darauf aufbauend werden in Kapitel 6.1.3 die Prinzipien des Artefaktmodells erklärt. In Kapitel 6.2 werden das Metamodell des Artefaktmodells und die einzelnen Elemente des Artefaktmodells beschrieben. Kapitel 6.3 beschäftigt sich mit der detaillierten Vorstellung des Artefaktmodells für das Requirements Engineering für PSS. Kapitel 6.4 bildet die Zusammenfassung dieses Kapitels.

6.1 Grundlagen des Artefaktmodells für Anforderungen an Product Service Systems

6.1.1 Begriffliche Grundlage

In der IS-Forschung wird oft der Begriff „IT Artefakt“ verwendet. Ein IT-Artefakt ist „an object produced or shaped by human craft, something viewed as a product of human conception or agency, or an inaccurate observation, effect, or result or an error“ (Alter 2008, 454-455). Hevner et al. (2004, 77) versuchen eine formale Definition für IT-Artefakte zu finden: “IT artifacts are broadly defined as constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems)”.

Orlikowski und Iacono (2001, 130-132) definieren fünf Prämissen eines IT-Artefakts:

- (1) IT-Artefakte sind per Definition nicht natürlich, neutral, universal, oder gegeben.
- (2) IT-Artefakte sind immer eingebettet in einen Zeit- und Ortskontext, einen Gegenstandsbereich und einer Gesellschaft.
- (3) IT-Artefakte bestehen gewöhnlicher Weise aus mehreren, oft lose kombinierten Komponenten, dessen Verbindungen oft durch Überbrückungen, Integration und Artikulation zusammengehalten werden müssen.
- (4) IT-Artefakte sind weder fix noch unabhängig. Aber sie entstehen aus der laufenden sozialen und ökonomischen Praxis.
- (5) IT-Artefakte sind nicht statisch und nicht unveränderbar, sondern dynamisch.

Zusammenfassend stellt ein IT-Artefakt ein Konstrukt dar, das im Rahmen von bestimmten Aktivitäten erstellt wird. Die oben angeführten Definitionen sowie die Prämissen lassen sich weiterhin auf das Requirements Engineering übertragen, indem sie auf die Betrachtungsperspektive der Anforderungen angepasst werden.

Das artefakt-orientierte Requirements Engineering ist darauf ausgerichtet, die Erstellung und Verwaltung von Artefakten, die in Artefaktmodellen zusammengefasst sind und Anforderungs- und Entwicklungsinformationen enthalten, zu beschreiben. Balzert (2000, 54) definiert ein Artefakt als Ergebnis einer Aktivität des Entwicklungsprozesses, die durch das Vorge-

hensmodell vorgegeben wird. Ein Artefakt ist ein Ergebnis einer Aktivität des Requirements Engineering (Geisberger et al. 2006, 5; Berenbach et al. 2009, 19-21; Pohl et al. 2005, 20-24).

Somit wird ein *Artefakt* im Rahmen dieser Arbeit als ein disjunkter Teil der Information definiert, der von einer Entwicklungsaktivität erzeugt wird und der dieselben vorher definierten Charakteristika aufweist. Ein Artefakt gibt somit eine Anforderungs- oder Entwicklungskategorie vor (Berkovich et al. 2011a). Die Granularität der Zusammenfassung der Informationen in die Artefakte soll plausibel sein und unter Zuhilfenahme von erforderlichen Aspekten, wie bspw. des Entwicklungsprozesses, zu dem auch die Requirements Engineering Aktivitäten gehören, und die darin erzeugten und benötigten Informationen, erfolgen. Ein Artefakt gibt ebenso die Darstellungsform für die Anforderungen und Entwicklungsinformationen vor.

Die Artefakte werden in einem Artefaktmodell strukturiert und im Rahmen der Requirements Engineering Aktivitäten modifiziert (Berenbach et al. 2009, 27). Ein Artefaktmodell soll die Artefakte festlegen und zueinander in Beziehung setzen (Broy et al. 2007, 134). Ein Artefaktmodell unterstützt ein kontinuierliches Requirements Engineering, indem die Anforderungen nach Abstraktionsebenen strukturiert werden und abstrakte initiale Anforderungen schrittweise in detailliertere Anforderungen konkretisiert werden (Gorschek/Wohlin 2006, 80). Eine *Abstraktionsebene* beinhaltet Anforderungen und Entwicklungsinformationen, die im selben Entwicklungsschritt erstellt werden (Berkovich et al. 2011a).

Ein *Artefaktmodell* ermöglicht eine Gliederung von Anforderungen in problemlösungsorientierte Kategorien (Artefakte) und stellt ein Konkretisierungsschema für Anforderungen dar, das vorgibt, wie die Anforderungen schrittweise über die Phasen der Entwicklung in Abstraktionsebenen immer detaillierter werden und technische Merkmale und Ausprägungen erhalten (vgl. (Berkovich et al. 2011a; Pohl/Sikora 2007a; Gorschek/Wohlin 2006; Méndez Fernández et al. 2010; Geisberger et al. 2006, 5)).

6.1.2 Abgrenzung zu den existierenden artefakt-orientierten Ansätzen des Requirements Engineering

In Kapitel 5.2 wurden existierende Requirements Engineering Ansätze analysiert, ihre Vor- und Nachteile diskutiert, um dadurch die Wahl des artefakt-orientierten Requirements Engineering zu begründen. Es existieren aber zahlreiche Artefaktmodelle, die vom Artefaktmodell, das in dieser Arbeit entwickelt wird, abzugrenzen sind bzw. es sollen die im entwickelten Artefaktmodell verwendeten Aspekte aus den existierenden Ansätzen aufgezeigt werden. Zur Bewertung der Ansätze wurden die Herausforderungen im Requirements Engineering (Kapitel 4.7) herangezogen.

6.1.2.1 Requirements Abstraction Model (Gorschek/Wohlin 2006)

Das Requirements Abstraction Model (RAM) von Gorschek and Wohlin (2006) unterstützt das Requirements Engineering über den gesamten Entwicklungsprozess mit dem Ziel, die zu Beginn der Entwicklung abstrakten und lösungsneutralen Anforderungen schrittweise auf detailliertere Abstraktionsebenen herunterzubrechen und dabei eine durchgängige Verbindung von den konkreten Anforderungen bis zu den initialen Anforderungen anzubieten. Dabei stellt das RAM ein Framework dar, das aus einem Artefaktmodell besteht und Empfehlungen über die Aktivitäten und Methoden zur Erzeugung und Verwaltung von Artefakten vorgibt. Das RAM ist mit seinen Abstraktionsebenen vier Abstraktionsebenen- Product Level, Feature Level, Function Level und Component Level - an den Softwareentwicklungsprozess angelehnt.

Die Anforderungen auf dem „Product Level“ sind abstrakt, lösungsneutral und geben grob die Wünsche der Stakeholder bezüglich des zu entwickelnden Produktes an. Die nächste Abstraktionsebene „Feature Level“ enthält Anforderungen, die die Eigenschaften des zu entwickelnden Produkts beschreiben, ohne auf die eigentlichen Funktionalitäten des Produktes einzugehen. Darauf basierend werden die Anforderungen in funktionale und nicht funktionale aufgeteilt und bilden dadurch eine weitere Ebene „Function Level“. Diese Anforderungen beschreiben die Funktionalität des zu entwickelnden Produktes. Anschließend werden die Anforderungen konkretisiert, den einzelnen Komponenten des zu entwickelnden Produktes zugeordnet und bilden somit die Ebene „Component Level“. Die Anforderungen dieser Ebene sind so präzisiert, dass sie an die Entwicklung weitergeleitet werden können. Das RAM unterstützt keine Strukturierung von Anforderungen in Kategorien (Artefakte), wodurch die Nachvollziehung der schrittweisen Konkretisierung von Anforderungen weitgehend nicht möglich ist. Ebenso fehlt die Beschreibung der Spezifikationsform für Anforderungen. Eine methodische Unterstützung zur Analyse, Konkretisierung und Verwaltung von Anforderungen wird nicht angegeben. Auch die Einbindung des Modells in den Entwurf fehlt weitgehend. Der Zusammenhang zwischen den Anforderungen und dem Entwurf wird nicht erläutert.

6.1.2.2 Requirements Engineering Reference Model (Geisberger 2005; Geisberger et al. 2006; Berenbach et al. 2009)

Das Requirements Engineering Reference Model (REM) stellt eine methodische Grundlage für die interdisziplinäre Erarbeitung der Anforderungs- und Systemspezifikation eingebetteter Systeme dar (Geisberger 2005; Geisberger et al. 2006; Berenbach et al. 2009). Im Mittelpunkt der Arbeit steht ein domänenübergreifender Modellierungsansatz für das Herausarbeiten, Analysieren, Abstimmen und Vervollständigen von Anforderungen und Systementwürfen, die gemeinsam mit den an der Entwicklung eingebetteter Systeme beteiligten Domänen erfolgt. Dem Artefaktmodell liegt die Unterscheidung nach Anforderungsklassen (Artefakten) zu Grunde, die ein Klassifikationsschema für Anforderungen darstellen. Die Anforderungen werden basierend auf drei Abstraktionsebenen (Ziele, funktionale Anforderungen und Rahmenbedingungen, detaillierte Systemanforderungen und Constraints) konkretisiert, indem sie detaillierter werden und schließlich an die Entwicklung übergeben werden. Es wird nicht genau definiert, was ein Artefakt ist. Das führt zur Vermischung der Spezifikationsformen für Anforderungen und ihrer Inhalte im Artefaktmodell. So stellt das Artefakt „SW-/HW-Design Constraints“, das die Einschränkungen an die Software- und Hardwarekomponenten beschreiben sollen, die Inhalte dar, ohne auf die eigentliche Spezifikationsform einzugehen. Desweiteren wird die Einbindung des Artefaktmodells in den Entwicklungsprozess nicht genau beschrieben. Die Abstraktionsebenen werden rudimentär eingeführt. Es bleibt ebenso unklar, wie die Abstraktionsebenen aufgebaut wurden. Der Zusammenhang zwischen den Anforderungsartefakten und dem Entwurf wird nicht detailliert erläutert. Auch die entsprechenden Aktivitäten zur Erstellung und Verwaltung von Artefakten werden nicht angegeben.

6.1.2.3 COSMOD-RE (Pohl/Sikora 2007a; Pohl/Sikora 2007b)

COSMOD-RE (Scenario and Goal based System Development Method) ist eine ziel- und szenariobasierte Methode zur Unterstützung des Hardware/Software Codesign bei der Entwicklung eingebetteter Systeme (Pohl 2007, 553-600; Pohl/Sikora 2007a; Pohl/Sikora 2007b). Die Methode basiert auf einer Definition von Anforderungs- und Entwurfsartefakten. Ein Anforderungsartefakt ist eine dokumentierte Anforderung (Pohl 2007, 85). Es werden drei Arten von Anforderungsartefakten unterschieden: Ziele, die eine Intention der Stakehol-

der bzgl. des zu entwickelnden Systems beschreiben, Szenarien, die ein Beispiel der Erfüllung oder Nichterfüllung von Zielen angeben, und lösungsorientierte Anforderungen, die Eigenschaften des Systems in Bezug auf seine Realisierung darstellen (Pohl 2007, 47-50). Die Entwurfsinformationen werden zu Entwurfsartefakten zusammengefasst. Zur Konkretisierung von Anforderungen wird iterativ zwischen der Anforderungsperspektive und der Entwurfsperspektive gewechselt. COSMOD-RE definiert sechs Abstraktionsebenen, die die Anforderungen zunehmend konkretisieren, den Funktionsgruppen und anschließend den genauen Softwarekomponenten zuordnen. Folgende Abstraktionsebenen werden unterschieden (Pohl 2007, 553-600; Pohl/Sikora 2007a; Pohl/Sikora 2007b):

1. *Systemanforderungen* stellen grobe Anforderungen an das System dar.
2. *Systemgrobarchitektur* strukturiert das System in Funktionsgruppen. Die Funktionsgruppen stellen dabei abstrahierte Hardware- und Softwarekomponenten dar, indem sie ihre Funktionalität beschreiben. Eine Funktionsgruppe bündelt die Systemanforderungen, die logisch zu einer vorher definierten Funktionalität gehören.
3. *Funktionsgruppenanforderungen* beschreiben eine Konkretisierung der Systemanforderungen bzgl. der Funktionsgruppen.
4. *HW/SW-Grobarchitektur* stellt die Aufteilung des Systems in Hardware- und Softwarekomponenten dar, indem die Systemgrobarchitektur unter Berücksichtigung der Funktionsgruppenanforderungen detailliert wird.
5. *HW/SW-Anforderungen* sind Anforderungen an Hardware- und Softwarekomponenten.
6. *SW-Deployment* beschreibt die Zuordnung von Softwarekomponenten zu Hardwarekomponenten.

Zusätzlich zur Aufteilung von Anforderungen und Entwurfsinformationen in Abstraktionsebenen wird ein Prozessmodell zur Unterstützung des Co-Designs angeboten (Pohl/Sikora 2007b). Dem Prozessmodell liegt die Idee zu Grunde, die Anforderungs- und Entwurfsperspektive inhaltlich iterativ zu vergleichen, zu konsolidieren und anschließend daraus detaillierte Anforderungen abzuleiten (Pohl/Sikora 2007a).

COSMOD-RE bietet keine Strukturierung von Anforderungen in Kategorien an. Die Verbindungen zwischen den Anforderungen bleiben daher unklar, wodurch die Verfolgung der Anforderungen und das Änderungsmanagement weitgehend nicht unterstützt werden. Die Abstraktionsebenen sind an den Entwicklungsprozess eingebetteter Systeme angelehnt, wobei im Mittelpunkt die Anforderungen an die Software betrachtet werden, während die Hardwarekomponenten als gegeben angenommen werden.

6.1.2.4 *Artefaktmodell zur integrierten Wiederverwendung von Anforderungen und technischer Architektur (Penzenstadler/Koss 2008)*

Das Artefaktmodell von Penzenstadler und Koss (2008) sollen die Wiederverwendung von Softwarekomponenten in großen Systemen unterstützen und dabei eine Verfeinerung von Anforderungs- und Entwurfsinhalten gewährleisten. Dieses Artefaktmodell basiert auf den existierenden Modellen COSMOD-RE (Pohl (2007)) und RAM (Geisberger et al. (2006), Broy et al. (2008)). Für jedes Subsystem, das wiederverwendet werden kann, wird ein Anforderungsartefaktmodell erstellt. Dabei teilt das Artefaktmodell die Artefakte in drei Kategorien ein: Kontext, Anforderungen und Entwurf. Weiterhin teilt das Artefaktmodell die Artefakte in drei Schichten ein: Anwendungsschicht, logische Architektur und technische

Architektur. Die drei Schichten des Artefaktmodells stellen Abstraktionsebenen dar, die die Verfeinerung von Anforderungs- und Entwurfsartefakten vornehmen, während die drei Kategorien angeben, welche Inhalte zu verfeinern und zu dokumentieren sind. Das vorgeschlagene Artefaktmodell bezieht sich auf eingebettete Systeme. Entsprechend sind die Anforderungsartefakte sowie die Entwurfsartefakte auf die Softwareentwicklung unter Berücksichtigung der Hardwarekomponenten angepasst.

6.1.2.5 *Meta-Modell für Artefaktmodelle und zwei Instanzen (Méndez Fernández et al. 2010)*

Ein Metamodell für ein Artefaktmodell im Requirements Engineering wird von Méndez Fernández et al. (2010) vorgestellt. Im Artefaktmodell werden die „Artefact Structure“, die die Artefakte und Beziehungen zwischen ihnen beschreibt, und der „Artefact Content“, der die Inhalte des Artefaktmodells beschreibt, unterschieden. Somit werden die eigentlichen Inhalte und die Spezifikationsform der Artefakte voneinander getrennt. Zusätzlich werden ein „Generic Process Model“ mit den Aktivitäten zur Erstellung und Verwaltung von Artefakten und ein „Artefact Abstraction Model“ mit der Beschreibung von Abstraktionsebenen angeboten. Ebenso werden die Akteure, die an der Erstellung und Verwaltung von Artefakten beteiligt sind, in einem „Generic Role Model“ abgebildet. Basierend auf dem Metamodell wurden zwei weitere Artefaktmodelle entwickelt. Das REMsES (Requirements Engineering und Management für softwareintensive Eingebettete Systeme) stellt ein Artefaktmodell zur Entwicklung eingebetteter Systeme dar und basiert auf dem COSMOD-RE Ansatz von Pohl und Sikora (2007a; 2007b). Das REMbIS (Requirements Engineering and Management for Business Information System) wurde für betriebliche Informationssysteme entwickelt (Méndez Fernández/Kuhrmann 2009). Es unterscheidet fünf Abstraktionsebenen:

- *Organisation's Context* beschreibt strategische Ziele des Unternehmens,
- *Business Process Hierarchy* beschreibt die Geschäftsprozesse sowie die Organisationshierarchie,
- *Business Process Logic* beschreibt die Informationsflüsse die vom Informationssystem bereitgestellt werden,
- *Information System Service Hierarchy* beschreibt die Funktionalitäten des Informationssystems,
- *Information System's Constraints* beschreibt die Anforderungen, logische und technische Einschränkungen bzgl. des Informationssystems.

Jede Abstraktionsebene verfügt über vier Sichten: Verhaltenssicht, die Nutzungsprozesse der Anwendung betrachtet, Struktursicht, die die logische Struktur der Anwendung und die beteiligten Einheiten der Anwendung betrachtet, Umgebungssicht, die die Umgebung und die Interaktion des Systems mit der Umgebung betrachtet, und Informationssicht, die Daten und Datenstrukturen der Anwendung betrachtet. Die Abstraktionsebenen und die Sichten bilden ein Raster, in das die Artefakte platziert werden (Méndez Fernández/Kuhrmann 2009, 33). Wie im Metamodell definiert (s. (Méndez Fernández/Kuhrmann 2009, 20f.)) werden zusätzlich zum eigentlichen Artefaktmodell ein Vorgehensmodell mit den Aktivitäten zur Erstellung und Verwaltung von Artefakten und ein Rollenmodell zur Beschreibung der Akteure, die an der Erstellung und Verwaltung von Artefakten beteiligt sind, angeboten.

6.1.2.6 Fazit

Die in der Literatur angebotenen Artefaktmodelle sind auf unterschiedliche Anwendungsgebiete zugeschnitten und konzentrieren sich auf einzelne Aspekte und Herausforderungen im Requirements Engineering. Zusammenfassend lässt sich feststellen, dass keiner der untersuchten Ansätze in der Lage ist, die Herausforderungen des Requirements Engineering für PSS zu lösen und die an den Requirements Engineering Ansatz gestellten Anforderungen (s. Kapitel 4.7) zu erfüllen. Im Detail werden vor allem die folgenden Punkte nicht gelöst:

1. Keines der vorhandenen Artefaktmodelle berücksichtigt Anforderungen an Dienstleistungen. RAM und REMbIS beschränken sich auf Anforderungen für betriebliche Informationssysteme. Sie ziehen also ausschließlich Anforderungen an Software in Betracht. COSMOD-RE wurde auf die Entwicklung von eingebetteten Systemen zugeschnitten und berücksichtigt also nur den Fall der Entwicklung von Software, wobei die Hardware bereits vorgegeben ist. Auch REM ist auf eingebetteten Systemen zugeschnitten. Diese Artefaktmodelle können nicht auf triviale Weise um Anforderungen an Dienstleistungen erweitert werden, weil dabei auch die Abhängigkeiten und Interaktionen zwischen den Anforderungen an das technische Produkt und den Dienstleistungen berücksichtigt werden sollen.
2. Die bestehenden Ansätze realisieren keine durchgängige Integration des Artefaktmodells in den Entwicklungsprozess. Der Ansatz COSMOD-RE betont die Wichtigkeit der Berücksichtigung von Entwicklungsinformationen zur Konkretisierung von Anforderungen, beschreibt jedoch nur eine iterative Anforderungskonkretisierung. Er gibt keine detaillierten Entwicklungsartefakte und Anforderungsartefakte vor und beschreibt folglich auch keine Zusammenhänge zwischen diesen im Detail. REM erwähnt zwar auch die Wichtigkeit der Entwicklungsinformationen, macht aber keine Vorgaben, auf welche Weise diese berücksichtigt werden sollen. Alle anderen Ansätze berücksichtigen keine die Integration der Entwicklungsinformationen nicht.
3. Eine Integration der Artefaktmodelle in ein Vorgehensmodell, das Aktivitäten und Techniken anbietet, wird von den bestehenden Ansätzen nicht realisiert. REM und REMbIS definieren einen allgemeinen Prozess, der eine iterative Anforderungserhebung und -verwaltung vorsieht. Sie beschreiben jedoch nicht im Detail die Aktivitäten, die dazu nötig sind. Zudem geben diese Ansätze keine Techniken an, die zur Durchführung der Aktivitäten verwendet werden können. RAM und COSMOD-RE bleiben in ihrer Definition eines Prozesses auf einer ähnlich abstrakten Ebene und liefern auch keine konkreten Vorgaben zur Durchführung des Requirements Engineering.
4. Die bestehenden Ansätze geben keine detaillierten Taxonomien für die einzelnen Artefakte vor. COSMOD-RE und RAM definieren im Wesentlichen lediglich Abstraktionsebenen und eine grobe Einteilung der Artefakte in drei Kategorien. REM und REMbIS geben eine größere Anzahl von Artefakten vor, erreichen jedoch nicht den Detaillierungsgrad von konkreten Taxonomien. Zudem wird in bestehenden Artefaktmodellen oft der eigentliche Inhalt von Artefakten mit ihren Spezifikationsformen vermischt, wodurch der Umgang mit den Artefakten erschwert wird.

6.1.3 Prinzipien des Artefaktmodells für Anforderungen an Product Service Systems

Aufbauend auf den existierenden artefakt-orientierten Ansätzen und den Anforderungen an einen Requirements Engineering Ansatz für PSS (s. Kapitel 4.7) wurde ein Artefaktmodell entwickelt, dessen Prinzipien in diesem Kapitel beschrieben werden. Diese Prinzipien sollen

die Funktionsweise des Artefaktmodells für die Anforderungen an PSS verdeutlichen, ohne auf die genaue Ausgestaltung der Elemente des Artefaktmodells einzugehen.

6.1.3.1 Strukturierung von Artefakten

Durch die Strukturierung von Anforderungen in Artefakte und ihre schrittweise Konkretisierung in Anlehnung an den Entwicklungsprozess von PSS ermöglicht das Artefaktmodell nachzuvollziehen, wie die initialen Anforderungen an das PSS über die Phasen der Entwicklung in immer konkretere Anforderungen zerlegt werden und schließlich in den lösungsorientierten Anforderungen an die domänenspezifischen Komponenten des PSS münden.

Die Strukturierung von Anforderungen und Entwicklungsinformationen in Artefakte ermöglicht es, die Anforderungen und Entwicklungsinformationen nach ihren Inhalten zusammenzufassen und dadurch eindeutig zu klassifizieren. Somit stellen Artefakte Klassifikationskategorien dar. Diese Klassifikationskategorien lassen sich hierarchisch zerlegen, wofür Taxonomien verwendet werden (Berenbach et al. 2009, 21). Eine Taxonomie ist eine strukturell orientierte Form der Wissensdarstellung und wird definiert als eine Klassifikation, in der Begriffe bzw. Objekte geordnet und zusammengefasst werden (Kienreich/Strohmaier 2006, 362-363). Als Objekte einer Taxonomie dienen dabei die Kategorien von Anforderungen oder Entwicklungsinformationen. Diese Kategorien, die in den Taxonomien zusammengefasst sind, stellen ihrerseits ebenso Artefakte dar.

Die Strukturierung von Artefakten in Taxonomien ermöglicht, die Zusammenhänge zwischen den einzelnen Kategorien von Anforderungen oder Entwicklungsinformationen festzustellen. Ebenso ist es durch detailliertere Kategorien möglich, vollständigere Anforderungen zu erhalten und die Anforderungen besser zu verstehen.

6.1.3.2 Aufbau des Artefaktmodells in Abstraktionsebenen

Das Artefaktmodell strukturiert die Artefakte, die Anforderungen oder Entwicklungsinformationen beinhalten, in Abstraktionsebenen (Berkovich et al. 2011a). Während auf den ersten Abstraktionsebenen die Anforderungen an das komplette PSS betrachtet werden, werden sie in den nächsten Abstraktionsebenen immer weiter konkretisiert, indem sie mit mehr lösungsorientierten Details angereichert werden und technische Merkmale und Ausprägungen erhalten. Die letzte Abstraktionsebene enthält die lösungsorientierten Anforderungen an die domänenspezifischen Komponenten des PSS. Die Abstraktionsebenen des Artefaktmodells sind an den Phasen des Entwicklungsprozesses von PSS (s. Kapitel 3.3.1) angelehnt. Somit werden fünf Abstraktionsebenen benötigt, die folgende Artefakte enthalten:

- 1) die allgemeinen Ziele des Kunden und des Lösungsanbieters,
- 2) die initialen Anforderungen an das PSS,
- 3) die Aufteilung des PSS in Sach- und Dienstleistungskomponenten und die Anforderungen an diese Komponenten,
- 4) die Funktionsstrukturen für Sach- und Dienstleistungskomponenten des PSS und die die Funktionalität dieser Komponenten beschreibenden Funktionsstruktur-Anforderungen,
- 5) sowie die domänenspezifischen Komponenten des PSS und die domänenspezifischen Anforderungen, die die Komponenten des PSS beschreiben.

Jede Anforderung, die in Abstraktionsebene i vorkommt, hat Vorgängeranforderungen in den Abstraktionsebenen $(i-1)$ bis 1 und Nachfolgeranforderungen in den Abstraktionsebenen $(i+1)$ bis zur Abstraktionsebene mit den Lösungsanforderungen.

6.1.3.3 Detaillierungsgrad von Anforderungen

Hull et al. (2004, 117) weisen darauf hin, dass „The level of detail that should be provided in derived requirements depends upon the level of development at which requirements engineering is being done“. Während die Artefakte die Kategorien für die Anforderungen, nach denen sie zusammengefasst werden, vorgeben, geben die Abstraktionsebenen den Detaillierungsgrad von Anforderungen in Anlehnung an die Phasen des Entwicklungsprozesses an. Der Detaillierungsgrad einer Anforderung gibt das Ausmaß an technischen Details zur Beschreibung einer Anforderung an.

6.1.3.4 Flexibler Einsatz des Artefaktmodells

Durch die Strukturierung von Anforderungen in Artefakte ermöglicht das Artefaktmodell einen flexiblen Einsatz abhängig von den Projektbedingungen, indem die benötigten Artefakte frei gewählt werden können (Berenbach et al. 2009, 21).

6.1.3.5 Iterativ-inkrementelle Konkretisierung von Anforderungen

Bei der Konkretisierung von Anforderungen müssen explizite und implizite Entwicklungsentscheidungen berücksichtigt werden (Pohl 2007, 565; Hull et al. 2004, 115-119). Deswegen müssen die Anforderungen und die Entwicklungsinformationen iterativ-inkrementell konkretisiert werden. In diesem Zusammenhang wird auf das Prinzip der iterativ-inkrementellen Entwicklung von Pohl (2007, 565) verwiesen: die Artefakte, die Anforderungen und Entwicklungsinformationen enthalten und in einer Iteration i entwickelt wurden, haben einen Einfluss auf die Artefakte der nächsten Iterationen $(i+1)$ bis n . In jeder Iteration werden die Artefakte mit den Anforderungen und die Artefakte mit den Entwicklungsinformationen, die im Rahmen früherer Iterationen erstellt wurden, ergänzt, detailliert oder revidiert.

Der iterativ-inkrementellen Konkretisierung von Anforderungen an PSS liegt somit die Idee zu Grunde, dass die Anforderungen in der Abstraktionsebene i iterativ durch die Anforderungen in der Abstraktionsebene $(i+1)$ konkretisiert werden. Die Iteration ist abgeschlossen, wenn die Anforderungen der Abstraktionsebene $(i+1)$ den erforderlichen Detaillierungsgrad erreicht haben. Zur Konkretisierung von Anforderungen werden dabei die Entwicklungsinformationen der Ebene i herangezogen. Abbildung 6-1 zeigt das Prinzip der Anforderungskonkretisierung für PSS.

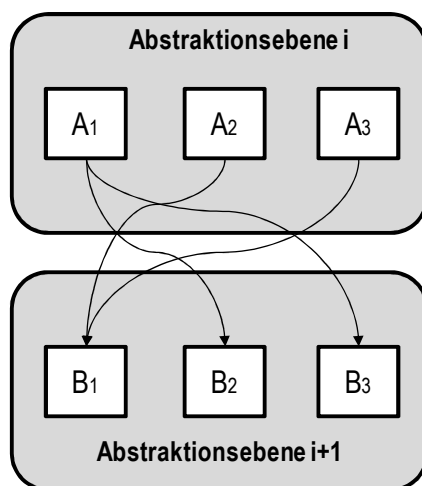


Abbildung 6-1: *Prinzip der iterativ-inkrementellen Konkretisierung von Anforderungen an PSS*

Quelle: Eigene Darstellung

Die Iterationen sind abgeschlossen, wenn die Lösungsanforderungen an die disziplinspezifischen Komponenten des PSS geklärt und dokumentiert sind. Die Konkretisierung erfolgt im Verhältnis $n:m$. Eine Anforderung von der Abstraktionsebene i kann durch m Anforderungen der Abstraktionsebene $(i+1)$ konkretisiert werden und eine Anforderung der Abstraktionsebene $(i+1)$ kann von n Anforderungen der Abstraktionsebene i stammen. So ist aus der Anforderung A_1 die Anforderungen B_2 und B_3 , aus den Anforderungen A_2 und A_3 die Anforderung B_1 abgeleitet.

In Detail erfolgt die Konkretisierung folgendermaßen. Im ersten Schritt wird jede Anforderung der Abstraktionsebene i durch m Anforderungen der Abstraktionsebene $(i+1)$ konkretisiert. Anschließend werden die Anforderungen der Abstraktionsebene $(i+1)$ verglichen, um inhaltlich ähnliche (redundante) Anforderungen zu identifizieren. Redundante Anforderungen beschreiben denselben Sachverhalt, werden aber unterschiedlich benannt und können somit zusammengefasst werden, indem eine Anforderung inhaltlich ähnliche Anforderungen ersetzt (vgl. (Lindemann 2006, 100)). Abbildung 6-2 zeigt die beiden Schritte des Vorgehens bei der Konkretisierung von Anforderungen. Wenn die Anforderungen B_4 und B_1 redundant sind, wird die Anforderung B_4 eliminiert. Da B_4 aus der Anforderung A_2 abgeleitet wurde, gilt nach der Eliminierung die Anforderung B_1 als Nachfolger von der Anforderung A_2 .

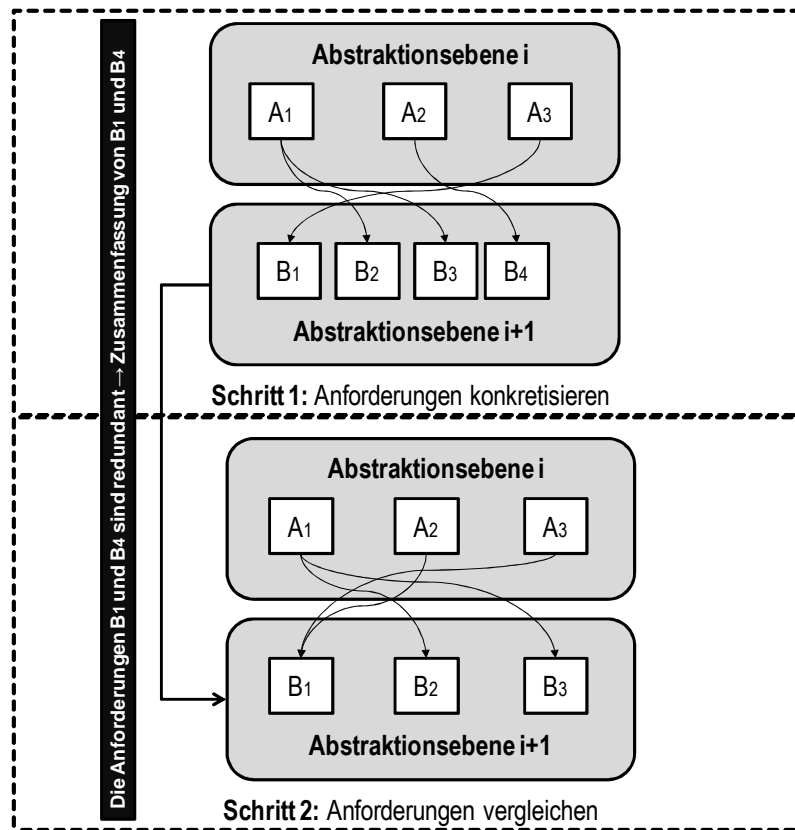


Abbildung 6-2: Vorgehen bei der Anforderungskonkretisierung für PSS

Quelle: Eigene Darstellung

Durch das Vorgehen der Konkretisierung wird für eine Anforderung auf der Abstraktionsebene i möglich ihre Vorgängeranforderungen auf der Abstraktionsebene $(i-1)$ bis zu den initialen Anforderungen an die komplette Lösung und ihre Nachfolgeranforderungen auf der Abstraktionsebene $(i+1)$ bis zu den Lösungsanforderungen an die domänenspezifischen Komponenten des PSS zu bestimmen. Somit wird die Verfolgung von Anforderungen über die Phasen der Entwicklung sowohl von den initialen Anforderungen bis zu den Lösungsanforderungen als auch von den Lösungsanforderungen bis zu den initialen Anforderungen gewährleistet.

Die Konkretisierung von Anforderungen findet nicht nur zwischen den Abstraktionsebenen, sondern ebenso innerhalb einer Abstraktionsebene statt. Die Anforderungen einer Abstraktionsebene beeinflussen einander. So werden die Dienstleistungen unter Berücksichtigung der Vorgaben des technischen Produktes entwickelt und vice versa. Diese Interdependenzen zwischen den Komponenten des PSS spiegeln sich in den Anforderungen wider. D. h. die Konkretisierung von den Anforderungen an die Dienstleistungen soll die Anforderungen an das technische Produkt einbeziehen und vice versa.

6.1.3.6 Verfolgung von Anforderungen

Eine vollständige Anforderungsverfolgung ist erreicht, wenn die Post-Traceability, die die Verfolgung der Anforderung zu ihren Nachfolgeranforderungen bereitstellt, und die Pre-Traceability, die die Verfolgung von Anforderungen zu ihren Vorgängeranforderungen bereitstellt, gewährleistet werden (Gotel/Finkelstein 1994). Dafür sollen für jede Anforderung auf der Abstraktionsebene i Vorgängeranforderungen auf den Abstraktionsebenen $(i-1)$ bis zu der ersten Abstraktionsebene existieren und ermittelbar sein. Zusätzlich müssen auch alle

Nachfolgeranforderungen auf den Abstraktionsebenen ($i+1$) bis zur letzten Abstraktionsebene mit den Lösungsanforderungen existieren und ermittelbar sein. Somit können die Anforderungen über alle Abstraktionsebenen, und d. h. über alle Phasen der Entwicklung, verfolgt werden. Die Verfolgung unterstützt das kontinuierliche Verständnis der Anforderungen und führt dadurch zu einer besseren Entscheidungsunterstützung für das Management und die Entwickler (Gorschek/Wohlin 2006, 80). Nur so kann gewährleistet werden, dass jede Lösungsanforderung auf eine initiale Anforderung an das PSS abgebildet und somit die Existenz der Lösungsanforderung begründet wird.

6.1.3.7 Änderungsmanagement

Wenn sich eine Anforderung auf der Abstraktionsebene i ändert, sollen alle Vorgängeranforderungen, aus denen die geänderte Anforderung abgeleitet wurde, und alle Nachfolgeranforderungen, die aus der geänderten Anforderung abgeleitet wurden, analysiert und auf mögliche Änderungen überprüft werden.

6.1.3.8 Umgang mit neuen Anforderungen

Wenn während der Entwicklung eine neue Anforderung auftritt, soll ihr Detaillierungsgrad mit dem von den Abstraktionsebenen vorgegebenen Detaillierungsgrad verglichen und in die richtige Abstraktionsebene aufgenommen werden. Da aber eine Anforderung nicht ohne Vorgängeranforderungen und Nachfolgeranforderungen existieren darf, sollen entsprechende Anforderungen auf allen Abstraktionsebenen, die einen niedrigeren Detaillierungsgrad und einen höheren Detaillierungsgrad der Anforderungen verlangen, ergänzt werden. Schematisch heißt das, wenn eine Anforderung durch ihren Detaillierungsgrad zur Abstraktionsebene i gehört, soll diese Anforderungen in Abstraktionsebene i aufgenommen werden. Zudem müssen Vorgängeranforderungen auf den Abstraktionsebenen ($i-1$), ($i-2$), bis ($i-n$), wobei die Abstraktionsebene ($i-n$) die initialen Anforderungen an das zu entwickelnde PSS enthält, eingefügt werden. Die Nachfolgeranforderungen auf den Abstraktionsebenen ($i+1$), ($i+2$), bis ($i+m$), wobei die Abstraktionsebene ($i+m$) die Lösungsanforderungen an die domänenspezifischen Komponenten des PSS enthält, müssen auch ergänzt werden. Somit sind die Existenz und die Legitimität jeder Anforderung durch ihre Verwurzelung in den initialen Anforderungen begründet. Die Begründung für die Umsetzung jeder Anforderung ist durch ihre Konkretisierung durch die Nachfolgeranforderungen angegeben. Die Anforderungsverfolgung wird demzufolge gewährleistet und das Änderungsmanagement wird durch eine schnelle Identifikation der von den Änderungen betroffenen Vor- und Nachfolgeranforderungen unterstützt.

6.2 Metamodell des Artefaktmodells für Anforderungen an Product Service Systems

Der erste Schritt in der Konzeption eines Artefaktmodells ist das Erarbeiten eines Metamodells für das Artefaktmodell. Ein Metamodell ist definiert als „a model that defines and interrelates conceptual abstractions in terms of which other models are defined“ (Lamsweerde 2009, 85). In einem Metamodell werden keine konkreten Ereignisse oder Objekte eines Anwendungsbereichs beschrieben, sondern die Elemente und Konstrukte eines darauf basierend zu erarbeiteten Modells festgelegt (Born et al. 2004, 60-63; Brugger 2005, 493).

Das Metamodell des Artefaktmodells für die Anforderungen an PSS beschreibt die Bestandteile des eigentlichen Artefaktmodells und die Beziehungen zwischen ihnen. Abbildung 6-3 zeigt das Metamodell für das Artefaktmodell für die Anforderungen an PSS.

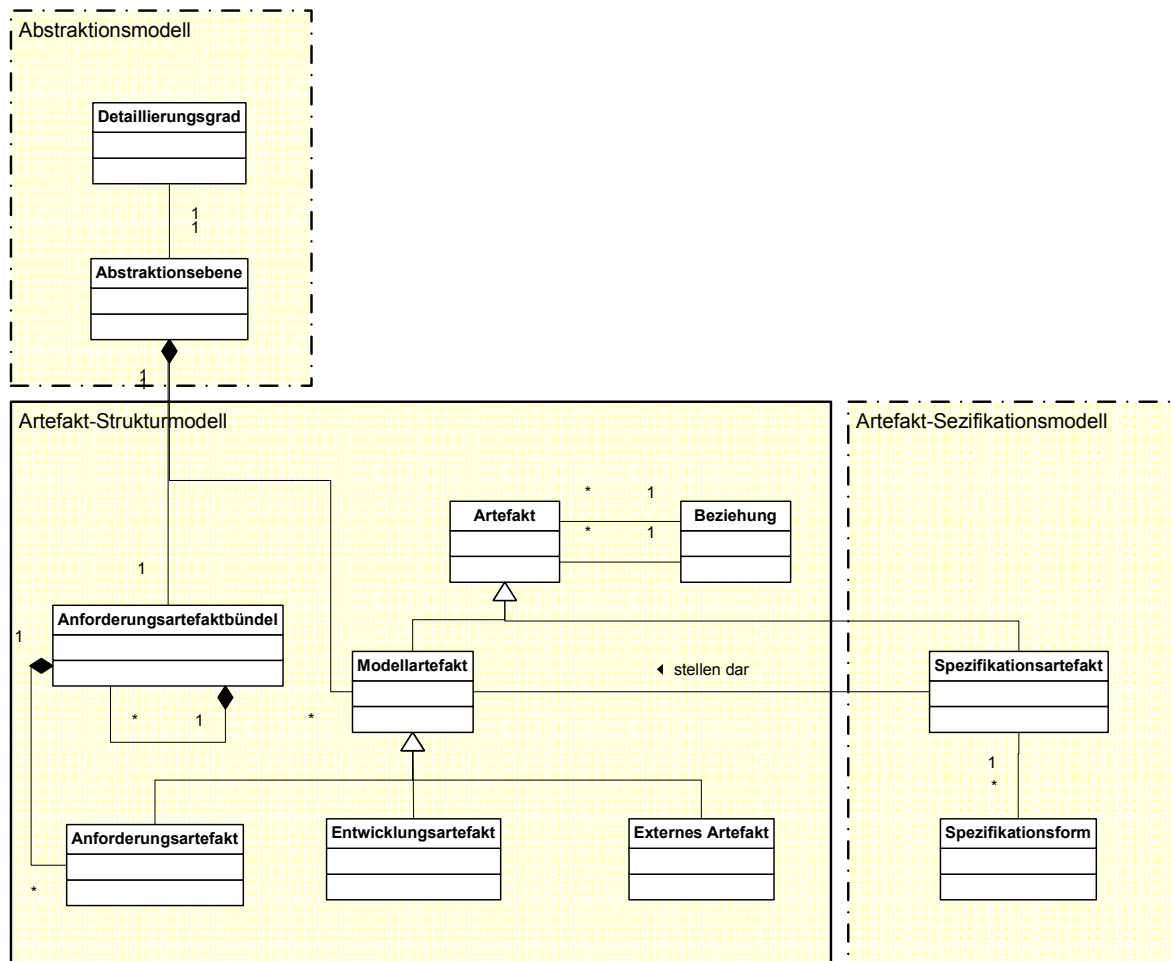


Abbildung 6-3: Metamodell des Artefaktmodells für die Anforderungen an PSS

Quelle: Eigene Darstellung (In Anlehnung an (Berkovich 2010; Berkovich et al. 2011a; Méndez Fernández et al. 2010))

Als Grundlage zur Konzeption des Metamodells dient das Metamodell von Méndez Fernández et al. (2010) (s. Kapitel 6.1.2), das Metamodell für ein Artefaktmodell von Berkovich et al. (2011a) und die Prinzipien des Artefaktmodells für die Anforderungen an PSS aus Kapitel 6.1.3.

Das Artefaktmodell setzt sich aus drei Modellen zusammen: einem Abstraktionsmodell, einem Artefakt-Strukturmodell und einem Artefakt-Spezifikationsmodell. In Anlehnung an das Artefaktmodell von Méndez Fernández et al. (2010) und Berkovich et al. (2011a) wird zwischen den eigentlichen Inhalten und den Spezifikationsformen der Anforderungen und Entwicklungsinformationen unterschieden.

6.2.1 Abstraktionsmodell

Das Abstraktionsmodell beschreibt die Abstraktionsebenen des Artefaktmodells. Abstraktion bedeutet das Weglassen von Details, um etwas Allgemeineres oder Einfacheres herauszuarbeiten. Jede *Abstraktionsebene* hat das Ziel einen bestimmten Sachverhalt der Entwicklung (bspw. Anforderungsermittlung) darzustellen, was durch das Weglassen von unterschiedlichen Details erreicht wird. Mit jedem Schritt der Entwicklung geben die Abstraktionsebenen den nötigen Grad der Detaillierung (*Detaillierungsgrad*) von Anforderungen an und ändern sich vom abstrakten zum konkreten (Méndez Fernández et al. 2010; Berkovich et al. 2011a).

Jede Abstraktionsebene hat einen Detaillierungsgrad, der an den Entwicklungsprozess angelehnt ist und somit den Abstraktionsgrad von Anforderungen bestimmt.

6.2.2 Artefakt-Strukturmodell

Das Artefakt-Strukturmodell beschreibt die *Artefakte* und die *Beziehungen* zwischen ihnen. Das Artefakt-Strukturmodell ist mit dem Abstraktionsmodell verbunden, so dass jede Abstraktionsebene mehrere Artefakte, die denselben vorgegebenen Detaillierungsgrad haben, enthalten kann. Artefakte sind untereinander durch verschiedene Beziehungen verbunden.

Da jede Abstraktionsebene mehrere Anforderungsartefakte enthält, können sie zu einer einfacheren Handhabung gebündelt werden. Im Diagramm (Abbildung 6-3) wird die Bündelung der Anforderungsartefakte, die zur selben Abstraktionsebene gehören, durch die Klasse „*Artefaktbündel*“ präsentiert. So können bspw. die Geschäftsziele des Kunden und die Geschäftsziele des Auftragnehmers als Geschäftskontext adressiert werden.

Es gibt zwei Arten von Artefakten - *Modellartefakte* und *Spezifikationsartefakte*. Die Modellartefakte beschreiben den Inhalt, die Struktur und den logischen Aufbau der Anforderungen oder der Entwicklungsinformationen unabhängig von der Art und Technik ihrer Darstellung. Die Modellartefakte definieren die verschiedenen Kategorien von Anforderungen, Entwicklungsinformationen oder die zur Erstellung oder Änderung von Artefakten benötigten Informationen. Somit geben die Modellartefakte bspw. den Inhalt von Kundenanforderungen wieder, ohne darauf einzugehen, ob diese Anforderungen natürlichsprachlich oder mit Hilfe von UML Diagrammen dokumentiert sind. Die Unterteilung in Modell- und Spezifikationsartefakte ermöglicht, zwischen dem Inhalt des Artefakts und seiner Darstellung zu unterscheiden.

Es werden drei Arten von Modellartefakten unterschieden: *Anforderungsartefakte*, *Entwicklungsartefakte* und *externe Artefakte*. Die Anforderungsartefakte fassen die Anforderungen an PSS basierend auf definierten Kategorien in Anlehnung an den Vorgaben des Entwicklungsprozesses zusammen. So stellen die Anforderungen des Lösungsanbieters (Auftragnehmers) ein Anforderungsartefakt dar. Das Anforderungsartefakt gibt an, nach welchen Merkmalen die Anforderungen des Lösungsanbieters in ein Artefakt zusammengefasst werden können, und gibt die Attribute für diese Anforderungen vor. Die Anforderungsartefakte legen keine Form der Anforderungsdokumentation fest.

Jede Anforderung des Anforderungsartefakts verfügt über Attribute, die die Anforderung eindeutig identifizieren, ihren Inhalt und ihre Eigenschaften beschreiben (Hull et al. 2004, 81; Cross 2008, 76). Die Attribute können abhängig vom Projektkontext gewählt werden. Folgende Attribute wurden aus der Literatur identifiziert:

- *Anforderungstitel (obligatorisch)*: eindeutiger und charakterisierender Name der Anforderung (vgl. (Pohl 2007, 259; Ebert 2005, 116; IEEE Std 1233 1998; Robertson/Robertson 2007, 245; Rupp 2004, 385; Hull et al. 2004, 81)).
- *ID (obligatorisch)*: eindeutige Bezeichnung für eine Anforderung (vgl. (Pohl 2007, 259; Ebert 2005, 116; IEEE Std 1233 1998; Robertson/Robertson 2007, 245; Rupp 2004, 385; Hull et al. 2004, 81)).
- *Verantwortlicher*: Verantwortlicher für die Verwaltung der Anforderung (vgl. (Pohl 2007, 260; Rupp 2004, 385; Hull et al. 2004, 81)).
- *Begründung*: Erklärung, warum eine Anforderung aufgenommen wurde und welchen Nutzen sie durch die Erfüllung leisten soll (vgl. (Robertson/Robertson 2007, 245; Pohl 2007, 260; Hull et al. 2004, 81)).

- *Version*: aktuelle Versionsnummer der Anforderung (vgl. (Rupp 2004, 385)).
- *Nutzende Stakeholder*: Stakeholder, die aus der Anforderung einen Nutzen ziehen (vgl. (Pohl 2007, 260)).
- *Quelle (obligatorisch)*: Stakeholder, die die Anforderung geäußert haben, oder andere Anforderungen, aus denen die Anforderung abgeleitet wurde (vgl. (Pohl 2007, 260; Ebert 2005, 116; Robertson/Robertson 2007, 245; Rupp 2004, 385; Hull et al. 2004, 81)).
- *Erläuterung (obligatorisch)*: präzise, verständliche Beschreibung der Anforderung (vgl. (Ebert 2005, 116; Robertson/Robertson 2007, 245; Hull et al. 2004, 81)).
- *Anforderungstyp (obligatorisch)*: das Anforderungsartefakt, zu dem die Anforderung gehört. Es folgt aus den Prinzipien des Artefaktmodells, dass alle Anforderungen durch die Artefakte strukturiert werden.
- *Abstraktionsebene (obligatorisch)*: Abstraktionsebene, zu der die Anforderung gehört (vgl. (Hull et al. 2004, 81)).
- *Nachfolgeranforderungen (obligatorisch)*: Anforderungen, die aus der Anforderung abgeleitet wurden (vgl. (Ebert 2005, 116)).
- *Status*: Status der Realisierung der Anforderung (vgl. (Rupp 2004, 385)).
- *Autoren*: Ersteller der Anforderung (vgl. (Pohl 2007, 266; Robertson/Robertson 2007, 245; Rupp 2004, 385; Hull et al. 2004, 81)).
- *Änderungshistorie*: alle Änderungen der Anforderung im Laufe ihres Lebenszykluses (vgl. (Pohl 2007, 266; Rupp 2004, 385)).
- *Gültigkeitsdauer*: Zeitintervall, in dem die Anforderung gültig ist.
- *Spezifikationsform*: Form, in der die Anforderung dokumentiert wird (vgl. (Pohl 2007, 261; Hull et al. 2004, 81)).
- *Validierungsstatus*: Status der Validierung (ungeprüft, in Prüfung, partiell geprüft, überprüft, in Korrektur, freigegeben) (vgl. (Pohl 2007, 261)).
- *Übereinstimmungsstatus*: Übereinstimmung der Anforderungen zu anderen Anforderungen. Hier werden konfliktäre, und sich gegenseitig ausschließende Anforderungen festgehalten (vgl. (Pohl 2007, 263; Robertson/Robertson 2007, 245)).
- *Risiko*: Risiko, das mit der Umsetzung der Anforderung verbunden ist (vgl. (Pohl 2007, 265; Hull et al. 2004, 81)).
- *Priorität*: Wichtigkeit der Anforderungen in Bezug auf das zu lösende Problem für den Kunden (vgl. (Robertson/Robertson 2007, 245; Hull et al. 2004, 81)).
- *Aufwand*: Aufwand, der für die Realisierung der Anforderung anfallen (vgl. (Pohl 2007, 266; Ebert 2005, 116; Hull et al. 2004, 81)).
- *Kosten*: Kosten, die für die Realisierung der Anforderung benötigt werden (vgl. (Pohl 2007, 266; Ebert 2005, 116; Hull et al. 2004, 81)).

Die obligatorischen Attribute sind in allen Fällen zu verwenden, während alle anderen Attribute abhängig von den Projektbedingungen frei gewählt werden können.

Um die Einbindung des Requirements Engineering in den Entwicklungsprozess zu realisieren (wie in Kapitel 4.7 beschrieben, muss die Analyse, Konkretisierung und Validierung von Anforderungen unter Einbeziehung der relevanten Entwicklungsinformationen erfolgen), werden zur Konkretisierung und Strukturierung von Anforderungen die Entwicklungsinformationen benötigt. Die *Entwicklungsartefakte* orientieren sich an den Phasen des Entwicklungsprozesses und fassen die Entwicklungsinformationen zusammen, die zur Analyse, Konkretisierung, Validierung oder Verfolgung von Anforderungen benötigt werden. Ebenso wie die Anforderung

rungsartefakte werden sie den Abstraktionsebenen, auf denen sie durch die Entwicklung erzeugt werden, zugeordnet. Ein Beispiel für ein Entwicklungsartefakt stellen Funktionsstrukturen dar, die vom Entwicklungsprozess erzeugt werden und die Funktionalität des PSS beschreiben (s. Kapitel 3.3.1). Auch bei den Entwicklungsartefakten werden keine Spezifikationsformen angegeben. So definiert das Artefakt „Funktionsstrukturentwurf“ die charakteristischen Merkmale von Funktionen des PSS.

Desweiteren werden *externe Artefakte* unterschieden. Diese Artefakte fassen Informationen zusammen, die weder Anforderungen sind noch vom Entwicklungsprozess erzeugt werden. Vielmehr sind das Informationen, die zur Erstellung oder Änderung von Anforderungs- oder Entwicklungsartefakten benötigt werden. Dazu gehört bspw. die Liste aller relevanten Stakeholder, die die Anforderungen an das PSS liefern können. Für jede Abstraktionsebene werden diese Artefakte gesondert identifiziert. Da externe Artefakte keine Anforderungs- oder Entwicklungsinformationen enthalten, sondern nur die Erstellung von Anforderungs- und Entwicklungsartefakten in Form von Zusatzinformationen beeinflussen, wird im Rahmen dieser Arbeit darauf verzichtet, die Spezifikationsformen für die externen Artefakte anzugeben.

6.2.3 Artefakt-Spezifikationsmodell

Eine Spezifikation ist ein Dokument, das die Anforderungen an das zu entwickelnde System enthält (Grady 2006, 298). Eine Spezifikation wird daher auch als Anforderungsdokument bezeichnet. Eine Spezifikation fasst die Attribute der Anforderungen in einer strukturierten Form zusammen. Die in einem Modellartefakt enthaltene Information kann auf verschiedene Weise in einem Dokument festgehalten werden. Daher gibt es zu jedem Modellartefakt ein oder mehrere Spezifikationsartefakte, die die Art der Darstellung der Information festlegen. Beispielsweise kann das Anforderungsartefakt „*Prozessorientierte Anforderungen*“ durch die Spezifikationsartefakte „*EPKs*“ und „*UML Aktivitätsdiagramme*“ dokumentiert werden. Das Artefakt-Spezifikationsmodell umfasst die Spezifikationsartefakte, die die Modellartefakte darstellen, indem sie die Art und die Struktur vorgeben, wie die Anforderungen, Entwicklungsinformationen oder externe Artefakte dokumentiert werden.

6.3 Beschreibung des Artefaktmodells für die Anforderungen an Product Service Systems

Aufbauend auf dem Metamodell des Artefaktmodells wurde das Artefaktmodell für die Anforderungen an Product Service Systems erstellt. Abbildung 6-4 stellt das Artefaktmodell schematisch dar. Es wird bewusst darauf verzichtet, in der Abbildung auch externe Artefakte dazustellen, um die Komplexität der Darstellung zu minimieren.

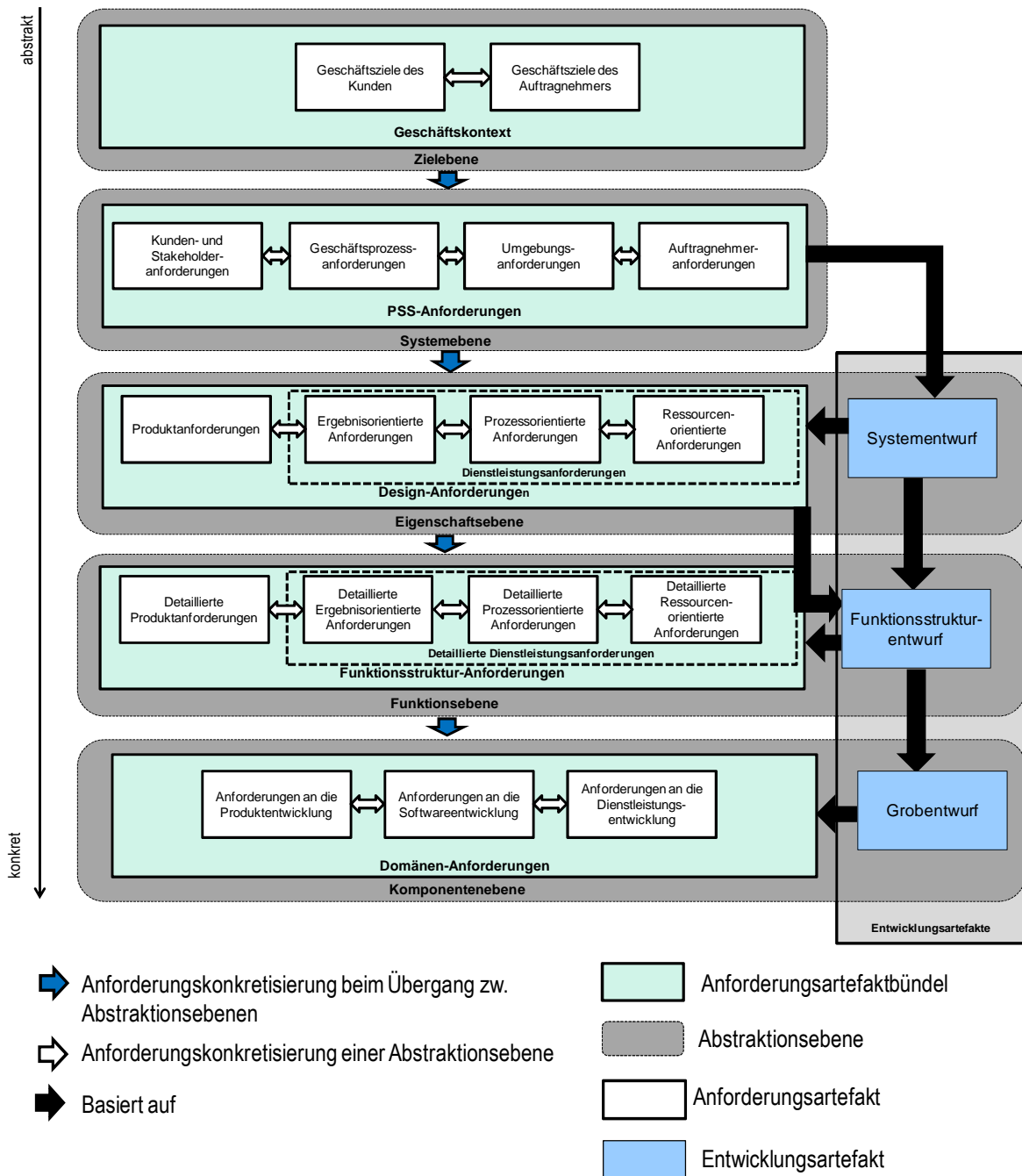


Abbildung 6-4: Artefaktmodell für die Anforderungen an PSS

Quelle: In Anlehnung an Berkovich et al. (2011a) und Berkovich (2010)

Das Artefaktmodell ist ein Strukturierungs- und Konkretisierungsschema für die Anforderungen an PSS. Obwohl die Analyse, Konkretisierung, Validierung und Vereinbarung von Anforderungen iterativ stattfindet, werden diese Iterationen nicht im Artefaktmodell, sondern im entsprechenden Vorgehensmodell (s. Kapitel 7) Abgebildet. Denn das Artefaktmodell ist ein Datenmodell zur Beschreibung von Anforderungen an PSS und Entwicklungsinformationen und zeigt die Artefakte und die Interdependenzen zwischen ihnen auf Daten- und nicht auf Prozessebene. Das Artefaktmodell für die Anforderungen an PSS besteht aus fünf Abstraktionsebenen, die an die Phasen des Entwicklungsprozesses von PSS (s. Kapitel 3.3.1) angelehnt sind. Die Namen der Abstraktionsebenen wurden in Anlehnung an das Requirements Abstraction Model von Gorschek und Wohlin (2006) gewählt.

Im Folgenden werden die Abstraktionsebenen kurz beschrieben. In den anschließenden Kapiteln werden sie dann einzeln mit den dazugehörigen Artefakten detailliert erläutert.

1. *Zielebene.* Diese Ebene umfasst zwei Anforderungsartefakte „Geschäftsziele des Kunden“ und „Geschäftsziele des Auftragnehmers“, die allgemeine Ziele bzgl. des zu entwickelnden PSS beschreiben. Diese Abstraktionsebene liegt noch außerhalb des eigentlichen Entwicklungsprozesses von PSS.
2. *Systemebene.* Diese Abstraktionsebene fasst die initialen, lösungsneutralen Anforderungen an das PSS zusammen (PSS-Anforderungen). Dies sind folgende Anforderungsartefakte: Kunden- und Stakeholderanforderungen, Geschäftsprozessanforderungen, Umgebungsanforderungen und Auftragnehmeranforderungen.
3. *Eigenschaftsebene.* Im Rahmen der Produktmodellierung des Entwicklungsprozesses von PSS (s. Kapitel 3.3.1) werden basierend auf den initialen Anforderungen an das PSS materielle und immaterielle Komponenten des PSS identifiziert und beschrieben, ohne aber auf ihre Funktionalität oder Gestaltung einzugehen. Die Aufteilung des PSS in materielle und immaterielle Komponenten wird im Entwicklungsartefakt „Systementwurf“ festgehalten. Die initialen PSS-Anforderungen der Systemebene werden so konkretisiert, dass sie den identifizierten Sach- und Dienstleistungskomponenten zugeordnet werden können. Die sich durch die Konkretisierung von PSS-Anforderungen ergebenden Design-Anforderungen werden als Anforderungsartefakte in dieser Abstraktionsebene festgehalten. So entstehen vier Anforderungsartefakte: Produkthanforderungen an die Sachleistungskomponenten und ergebnisorientierte, prozessorientierte und ressourcenorientierte Anforderungen an die Dienstleistungskomponenten.
4. *Funktionsebene.* Bei der Funktionsmodellierung (s. Kapitel 3.3.1) werden Funktionsstrukturen für Sach- und Dienstleistungen des PSS entworfen. Die Funktionsstrukturen werden im Entwicklungsartefakt „Funktionsstrukturentwurf“ zusammengefasst. Die Anforderungen an Sach- und Dienstleistungskomponenten der Eigenschaftsebene werden konkretisiert, indem sie durch die Funktionen gebündelt werden, wenn sie logisch ein und dieselbe Funktionalität beschreiben, und weiter bezüglich der Funktionalität der Sach- und Dienstleistungskomponenten des PSS präzisiert. Diese Anforderungen werden in den Anforderungsartefakten „detaillierte Produkthanforderungen“, die den Funktionen der Sachleistungskomponenten zugeordnet sind, und „detaillierte ergebnisorientierte Anforderungen“, „detaillierte prozessorientierte Anforderungen“ und „detaillierte ressourcenorientierte Anforderungen“, die den Funktionen der Dienstleistungskomponenten zugeordnet sind, zusammengefasst.
5. *Komponentenebene.* Die konzipierten Komponenten werden im Entwicklungsartefakt „Grobentwurf“ zusammengefasst. Im Rahmen der Prinzipmodellierung und der anschließenden Prozess- und Gestaltmodellierung (s. Kapitel 3.3.1) werden die domänenspezifischen Komponenten basierend auf den Funktionsstruktur-Anforderungen entworfen. Darauf aufbauend werden die Funktionsstruktur-Anforderungen bzgl. der domänenspezifischen Komponenten konkretisiert und den Komponenten zugeordnet. Diese Ebene umfasst die Anforderungsartefakte „Anforderungen an die Produktentwicklung“, „Anforderungen an die Softwareentwicklung“ und „Anforderungen an die Dienstleistungsentwicklung“ an die domänenspezifischen Komponenten des PSS.

Die Anforderungsartefakte beeinflussen einander, indem sie bspw. einander einschränken oder konkretisieren (Pohl 2007, 303). Deswegen werden mehrere Beziehungen zwischen den Artefakten unterschieden (Pohl 2007, 303-305; Geisberger 2005).

Die Beziehung „*Anforderungskonkretisierung beim Übergang zw. Abstraktionsebenen*“ (blauer Pfeil) bedeutet die Konkretisierung von Anforderungen, die in Anforderungsartefakten zusammengefasst sind, beim Übergang von einer Abstraktionsebene i zur Abstraktionsebene $(i+1)$. Die Konkretisierung von Anforderungen erfolgt, indem die Anforderungen unter Einbeziehung der Entwicklungsinformationen mehr technische Details bekommen und somit in Bezug auf das zu entwickelnde PSS konkreter werden (s. Kapitel 6.1.3.5).

Die Beziehung „*Anforderungskonkretisierung einer Abstraktionsebene*“ (weißer Pfeil) bedeutet eine gegenseitige Konkretisierung von Anforderungen in einer Abstraktionsebene. Diese Art der Konkretisierung erfolgt solange, bis alle Anforderungen den von der Abstraktionsebene verlangten Detaillierungsgrad besitzen, indem die gegenseitigen Einschränkungen und die gegenseitigen Ergänzungen durch die kategoriespezifischen Details bei der Anforderungsdefinition berücksichtigt werden (s. Kapitel 6.1.3.5).

Beispiel: Die Anforderungen an die Dienstleistungen können nicht ohne die Anforderungen an die dazugehörigen Sachleistungen betrachtet werden. Die Anforderungen an die Wartung einer Waschmaschine (Dienstleistung) sollen die Variante der Waschmaschine berücksichtigen, damit das zuständige Personal die richtigen Werkzeuge zur Durchführung der Wartung mitnimmt.

Die Beziehung „*Basiert auf*“ verbindet zwei Artefakte, wenn ein Artefakt zur Erstellung des anderen Artefakts durch die Bereitstellung von den zur Erstellung des Artefakts relevanten Informationen beigetragen hat. Diese Art der Beziehung wird im Artefaktmodell beim Erstellen der Entwicklungsartefakte „Systementwurf“, „Funktionsstrukturentwurf“ und „Grobentwurf“ benötigt. Zur Erstellung des Systementwurfs werden die Anforderungen an das PSS benötigt. Um den Funktionsstrukturentwurf zu erstellen, werden die Design-Anforderungen der Eigenschaftsebene und der Systementwurf benötigt. Der Grobentwurf basiert auf dem Funktionsstrukturentwurf. Ebenso werden die Design-Anforderungen der Eigenschaftsebene durch den Systementwurf beeinflusst, da dieser zur Strukturierung der Anforderungen der Eigenschaftsebene verwendet wird. Auf dieselbe Weise fließt der Funktionsstrukturentwurf in die Erstellung der Funktionsstruktur-Anforderungen an. Die Funktionsstruktur-Anforderungen werden aus den Anforderungen der Eigenschaftsebene abgeleitet, indem sie unter Zuhilfenahme des Funktionsstrukturentwurfs aufgeteilt und konkretisiert werden.

6.3.1 Abstraktionsebene 1: Zielebene

Der Entwicklung von PSS als integriertes Bündel aus technischen Produkten und Dienstleistungen liegt die Idee zugrunde, dass „Kunden kein Interesse an Produkten oder Dienstleistungen per se haben, sondern sie erwarten vielmehr die Lösung eines Problems, mit dem sie konfrontiert sind, oder die Erfüllung eines Bedarfes, den sie haben“ (Leimeister/Glauner 2008, 248). Demzufolge sind die Bedürfnisse des Kunden bzgl. des PSS entscheidend für die Konzeption und Entwicklung der Lösung.

Bevor mit der eigentlichen Anforderungsermittlung und -verwaltung angefangen werden kann, sollen die Ziele und Erwartungen des Kunden bzgl. des PSS ermittelt werden (vgl. (Rupp 2004, 86)). Nur, wenn die Ziele bekannt sind, können die richtigen Stakeholder identifiziert und somit die richtigen Anforderungen ermittelt werden (vgl. (Rupp 2004, 46; Leveson

2000; Wiegers 2005, 81)). Ziele definieren die Sicht und die Intentionen des Kunden und des Auftragnehmers bzgl. des zu entwickelnden PSS (vgl. (Robinson et al. 2003; Lindemann 2006, 335)). Ziele abstrahieren von der Nutzung des Systems als auch von den Aspekten der Systemrealisierung (Cox et al. 2009; Pohl 2007, 48; Lamsweerde 2009, 262-265). Sie beschreiben Zustände, die in der Zukunft liegen und die nicht automatisch eintreten (Gernert/Ahrend 2002, 23). Sie schränken den Lösungsraum ein, indem sie voraussetzen, dass nur solche Lösungen zulässig sind, die die Ziele erfüllen. Sie geben aber auch einen Gestaltungsfreiraum für die Kunden zur Definition konkreterer Anforderungen an die Lösung (Pohl 2007, 48). Somit ist ein Ziel „die intentionale Beschreibung eines charakteristischen Merkmals des zu entwickelnden Systems“ (Pohl 2007, 91).

Ziele resultieren aus den Bedürfnissen der Kunden (Grady 2006, 53). Sie sind die *Soll-Vorstellungen* des Kunden oder Auftragnehmers. Sie haben einen Aufforderungscharakter und können unpräzise sein. Sie sind immer unabhängig vom Produkt oder vom Konzept auszudrücken (Ulrich/Eppinger 2003, 54-56). Durch die Konkretisierung entstehen aus den Zielen die Anforderungen, die die *Soll-Eigenschaften* des Systems beschreiben (Ehrlenspiel 2002, 90). Somit sind Ziele unpräzise, abstrahierte und lösungsneutral formulierte Anforderungen an das PSS (Aurum/Wohlin 2005b).

Abbildung 6-5 zeigt die Zielebene und ihre Artefakte in Form eines Objektdiagramms. Die externen Artefakte werden aus den Gründen der Übersichtlichkeit nicht angegeben. Die Anforderungsartefakte der Zielebene werden durch das Artefaktbündel „Geschäftskontext“ zusammengefasst. Die beiden Anforderungsartefakte „Geschäftsziele des Kunden“ und „Geschäftsziele des Auftragnehmers“ werden durch die Spezifikationsartefakte „Zielspezifikation“ oder „Und-Oder-Baum“ dargestellt.

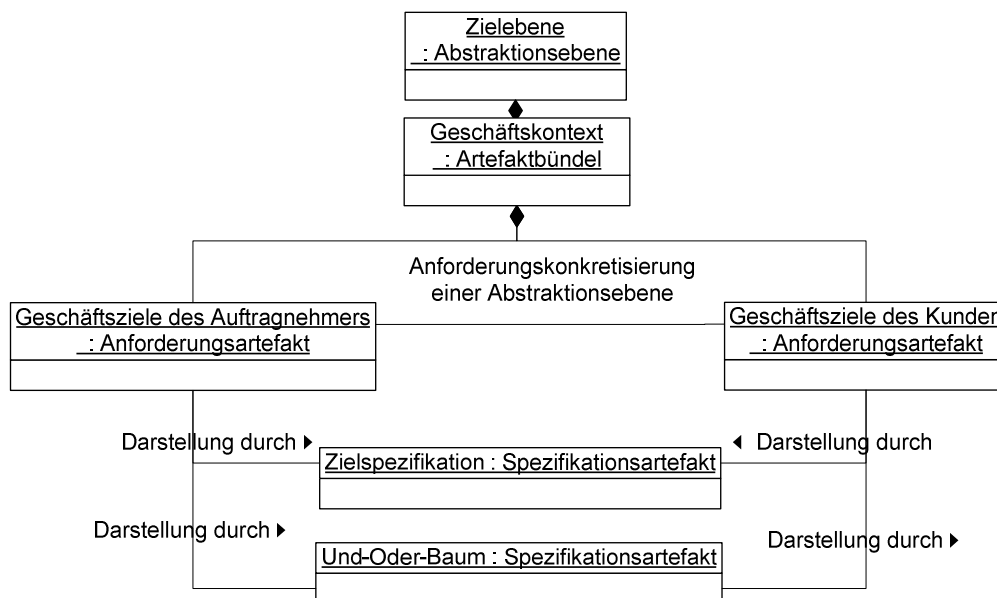


Abbildung 6-5: Artefakte der Zielebene

Quelle: Eigene Darstellung

6.3.1.1 Anforderungsartefakte der Zielebene

Die Zielebene besitzt keine Entwicklungsartefakte, weil sie noch außerhalb des eigentlichen Entwicklungsprozesses liegt. Die Ziele werden in zwei Anforderungsartefakten zusammengefasst.

Die Ziele des Kunden bilden das Artefakt „*Geschäftsziele des Kunden*“. Diese Ziele beschreiben den Zweck des PSS für den Kunden und beantworten somit Fragen wie „Welches Problem haben wir und welche Lösung erwarten wir?“, „Was wird das PSS bei uns verändern?“, „Warum ist das PSS für uns nötig?“, „Wer wird vom PSS profitieren?“, „Welche Kosten und Risiken sind wir bereit zu tragen?“, „Kann der Lösungsanbieter eine geeignete Lösung für uns konzipieren?“ (vgl. (Rupp 2004, 90-91)). Ein Beispiel soll verdeutlichen, was ein Ziel ist. Eines der Ziele einer Hotelanlage an den Lösungsanbieters ist: „Lieferung der gereinigten Wäsche an das Hotel“. Dabei werden in diesem Ziel keine Angaben zur Größe der Hotelanlage, Lage, Kosten oder Umfang der Wäsche abgebildet und somit die Nutzung oder die Realisierung des PSS beschrieben. Das Ziel gibt keine Angaben zur Gestaltung der Lösung, sondern beschreibt nur den Wunsch des Kunden.

Desweiteren hat auch der Auftragnehmer (Lösungsanbieter) bestimmte Ziele, die mit dem bereitzustellenden PSS zusammenhängen (Grady 2006, 86). Diese Ziele werden im Anforderungsartefakt „*Geschäftsziele des Auftragnehmers*“ zusammengefasst. Diese Ziele beschreiben den Zweck des PSS für den Lösungsanbieter aber auch die Erwartungen des Lösungsanbieters bzgl. des PSS und beantworten somit Fragen wie „Welches Problem hat der Kunde und welche Lösung erwartet er von uns?“, „Können wir das Ziel des Kunden erfüllen?“, „Warum ist das PSS für den Kunden nötig?“, „Wer wird vom PSS profitieren?“, „Welche Kosten und Risiken sind wir und der Kunde bereit zu tragen?“ (vgl. (Rupp 2004, 90-91)). Somit wird ersichtlich, dass die Ziele des Kunden und die Ziele des Auftragnehmers nicht voneinander unabhängig sind, sondern einander stark beeinflussen. Um die Konfliktfreiheit der Ziele zu gewährleisten, soll der Auftragnehmer die Ziele des Kunden berücksichtigen. Aber auch der Kunde soll die Ziele des Auftragnehmers kennen, um seine Ziele realistisch zu gestalten. Dabei ist eine gute Kommunikation mit dem Kunden zum Verstehen seiner Ziele und zum Formulieren und Anpassen der Ziele des Auftragnehmers entscheidend (Ebert 2005, 89-90).

6.3.1.2 Externe Artefakte der Zielebene

Die externen Artefakte der Zielebene stellen Informationen dar, die zur Erstellung der Artefakte „*Geschäftsziele des Kunden*“ und „*Geschäftsziele des Auftragnehmers*“ verwendet werden. Zur Ermittlung von Zielen sollen weitere Informationen herangezogen werden, die in den externen Artefakten zusammengefasst werden. Diese Informationen lassen sich in Anlehnung an Husen (2007, 92) in fünf Kategorien einordnen: strategische Faktoren, ökonomische Faktoren, Rahmenbedingungen, rechtliche Faktoren und gesellschaftliche Faktoren.

Um das PSS beim Kunden strategisch richtig zu positionieren und dadurch die strategischen Ziele des Kunden maximal zu erfüllen, ist es wichtig, die Unternehmensstrategie des Kunden aber auch des Lösungsanbieters, die sich bspw. durch den Preis oder die Wettbewerbssituation ausdrückt, zu verstehen. Um das PSS erfolgreich in die Landschaft des Kunden zu integrieren, soll die Verträglichkeit des PSS mit dem bestehenden Leistungsprogramm des Kunden überprüft werden (Böhm/Krcmar 2007, 244). Ebenso sollen die Marketingziele bzgl. der zu erbringenden Leistung durch das PSS betrachtet werden. Die Betrachtung der Wettbewerbsprodukte lässt die Ziele an das zu entwickelnde PSS ableiten, die sicherstellen sollen,

dass das zu entwickelnde PSS die Kundenwünsche besser erfüllt als die Konkurrenzprodukte. Die ökonomischen Faktoren wie Umsatz, Gewinn, Deckungsbeitrag, Investitionsvolumen, Kapazitätsauslastung oder Preisbereitschaft des Kunden sind Zielgrößen, die den wirtschaftlichen Rahmen für die Bereitstellung und Nutzung des PSS vorgeben und einen ökonomischen Erfolg des PSS sichern sollen (vgl. (Husen 2007, 91)). Die Rahmenbedingungen an die Bereitstellung von PSS werden durch die folgenden Gegebenheiten beeinflusst:

- Geschäftsprozesse des Kunden, die zur Erbringung des PSS relevant sind und in deren Landschaft das PSS integriert werden soll.
- Personal und die Technologien zur Entwicklung des PSS.
- Vorhandene Technologien beim Kunden, die für den Einsatz des PSS relevant sind.
- Rechtliche und gesellschaftliche Faktoren, zu denen Richtlinien, Gesetze oder ökologische Faktoren gehören.

Somit lassen sich folgende externe Artefakte (Tabelle 6-1) aufstellen (vgl. (Lindemann 2006, 66; Ebert 2005, 89-90; Lay/Jung 2002; Husen 2007, 87-95; Schweiger/Müller 2004; Krcmar 2010, 32-40; Hungenberg 2004, 84ff.; Schreiner/Strauß 2004; Kotonya/Sommerville 1998, 57f.)):

Kategorie	Externe Artefakte
Strategische Faktoren	<ul style="list-style-type: none"> • Unternehmensstrategien des Kunden und des Auftragnehmers, • Chancen und Risiken des PSS Angebots, • Kundenzufriedenheit, • Innovationsgrad des PSS.
Ökonomische Faktoren	<ul style="list-style-type: none"> • Wettbewerb, • Umsatz, • Gewinn, • Deckungsbeitrag, • Investitionsvolumen, • Kapazitätsauslastung, • Marktanteil, • Liquidität, • Unternehmensfortbestand, • Preisbereitschaft des Kunden, • Verfügbare Ressourcen, wie Zeit oder Kosten.
Rahmenbedingungen	<ul style="list-style-type: none"> • Geschäftsprozesse des Kunden • Personal, • Technologien, • Sonstige Betriebsmittel.
Rechtliche Faktoren	<ul style="list-style-type: none"> • Prozessbezogene Faktoren, • Informationsbezogene Faktoren, • Kundenbezogene Faktoren, • Mitarbeiterbezogene Faktoren.
Gesellschaftliche Faktoren	<ul style="list-style-type: none"> • Ökologische Faktoren, • Infrastruktur, • Branchenstandards.

Tabelle 6-1: *Externe Artefakte der Zielebene*

Quelle: Eigene Darstellung

6.3.1.3 Spezifikationsartefakt der Zielebene

Ziele müssen dokumentiert werden. Sie müssen dabei vollständig, korrekt, konsistent gegenüber anderen Zielen und in sich, testbar, verständlich für den Kunden und den Auftragnehmer, realisierbar, notwendig, eindeutig, aktuell und positiv formuliert sein (Rupp 2004, 100). Eine wichtige Einschränkung bei der Formulierung von Zielen ist ihre Lösungsneutralität. Die Ziele sollen keine Lösungsinformationen enthalten, sondern den Lösungsraum offen lassen.

Nachdem die Ziele des Kunden und des Auftragnehmers ermittelt und abgestimmt sind, so dass sie keine Widersprüche aufweisen, werden sie dokumentiert. Die dokumentierten Ziele werden im Spezifikationsartefakt „*Zielspezifikation*“ (in Anlehnung an „Market Requirements Document“ (MRD) von Wiegers (2005, 81)) zusammengefasst.

Da Ziele unkonkrete und lösungsneutrale Anforderungen an das PSS darstellen, besitzen sie ähnlich wie Anforderungen Attribute, die sie eindeutig identifizieren (s. Kapitel 6.2.2): Name des Ziels, eindeutiger Bezeichner (ID), Verantwortlicher für die Verwaltung der Ziele, Begründung zur Aufnahme des Ziels, Version, Erläuterung des Ziels, Abstraktionsebene, zu der das Ziel gehört, das Anforderungsartefakt, zu dem das Ziel gehört, Nachfolgeranforderungen, Autoren, Änderungshistorie, Gültigkeitsdauer, Spezifikationsform, Validierungsstatus, Übereinstimmungsstatus, Risiko. Ein Beispiel für eine Spezifikation von Zielen, in Anlehnung an Wiegers (2005, 81) und angepasst an PSS, ist in Tabelle 10-4 (Anhang) zu finden.

Falls die vorliegenden Ziele des Kunden zu grob sind, um daraus anschließend die Anforderungen an das PSS abzuleiten, sind die Ziele zu verfeinern. Dafür können die Ziele durch Und-Oder-Bäume dargestellt werden (Pohl 2007, 103-107). Ein Und-Oder-Baum ist ein Baum, dessen Knoten Ziele und dessen Kanten die Beziehungen zwischen den Zielen zeigen (Pohl 2007, 104). Durch die Und- und Oder-Beziehungen zwischen den Zielen lassen sich die Ziele durch Teilziele konkretisieren, indem die Ziele eine hierarchische Struktur und Vorgaben zu ihrer Erfüllung erhalten.

Somit hat das Spezifikationsartefakt der Zielebene zwei Ausprägungen: eine natürlichsprachliche Zielspezifikation anhand von Schablonen und ein Spezifikation in Form eines Und-Oder-Baums.

6.3.2 Abstraktionsebene 2: Systemebene

Nachdem die Ziele des Kunden und des Auftragnehmers ermittelt und geklärt sind, sollen die Anforderungen an das zu entwickelnde PSS (PSS-Anforderungen) ermittelt und strukturiert werden. Die Ziele geben dabei die Richtung für die Identifikation von Anforderungsquellen und die Ermittlung von Anforderungen vor, indem sie den Lösungsraum einschränken (Pohl 2007, 48). Diese Abstraktionsebene beschreibt die Anforderungen, die in Anlehnung an die Phase der hybriden Anforderungsklä rung des Entwicklungsprozesses von PSS (s. Kapitel 3.3.1). Dabei werden die Anforderungsquellen identifiziert und die Anforderungen an die gesamte Lösung ermittelt.

Abbildung 6-6 zeigt das Objektdiagramm der Artefakte der Systemebene. Die externen Artefakte werden aus den Gründen der Übersichtlichkeit in der Abbildung nicht angegeben. Die Anforderungsartefakte der Systemebene werden durch das Artefaktbündel „*PSS-Anforderungen*“ zusammengefasst. Die Anforderungsartefakte „*Kunden- und Stakeholderanforderungen*“, „*Geschäftsprozessanforderungen*“, „*Umgebungsanforderungen*“ und „*Auftragnehmeranforderungen*“ werden durch die Spezifikationsartefakte „*Use Cases*“, „*Szena-*

rio“, „EPK“, „ER-Diagramm“, „Aktivitätsdiagramm“, „Sequenzdiagramm“, „Zustandsdiagramm“, „Klassendiagramm“ und „Prosa“ dargestellt.

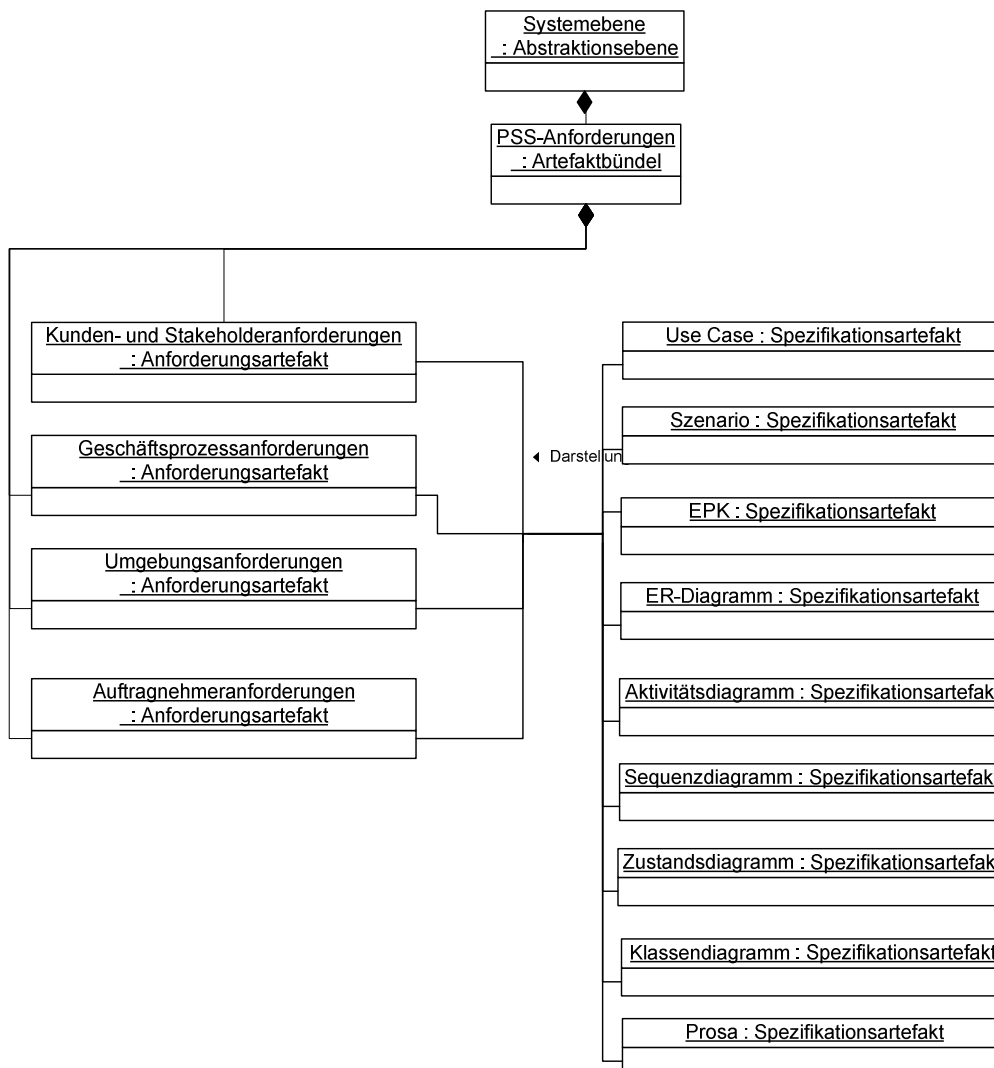


Abbildung 6-6: Artefakte der Systemebene
Quelle: Eigene Darstellung

Da es sich bei einem PSS um ein komplexes System handelt, dessen Ziel es ist, das vorhandene Kundenproblem zu lösen, drücken der Kunde und weitere Stakeholder ihre Anforderungen, nicht an ein konkretes technisches Produkt oder eine Dienstleistung, sondern an eine Lösung aus. Die Systemebene umfasst lösungsneutrale Anforderungen an das gesamte PSS, die keine Lösung voraussetzen und nicht auf die einzelnen materiellen oder immateriellen Komponenten des PSS und auf ihre Ausgestaltung eingehen. Die PSS-Anforderungen der Systemebene geben keinen Aufschluss darüber, um welches technische Produkt oder welche Dienstleistungen es sich handelt. Sie geben die Erwartungen und Vorstellungen der Stakeholder bzgl. der Bereitstellung und Nutzung des PSS wieder. Dadurch wird der Detaillierungsgrad der PSS-Anforderungen definiert. Die PSS-Anforderungen sind aus der Sicht des Kunden oder des Auftragnehmers formuliert und beziehen sich auf das PSS selbst und die allgemeinen Funktionalitäten des PSS, damit das PSS einen möglichst großen Nutzen für den Kunden bereitstellt. Ebenso sollen die Einschränkungen bzgl. der Bereitstellung des PSS, die bspw. von Gesetzen oder Richtlinien vorgegeben werden, berücksichtigt werden.

Wenn die Stakeholder auch detaillierte, lösungsorientierte Anforderungen oder die Anforderungen an die domänenspezifischen Komponenten des PSS äußern, sind diese Anforderungen aufzunehmen und entsprechend ihrem Detaillierungsgrad der richtigen Abstraktionsebene zuzuordnen (s. Kapitel 6.1.3). Da aber keine Anforderung auf einer Abstraktionsebene existieren kann, die nicht eine Vorgängeranforderung auf der Abstraktionsebene mit einem niedrigeren Detaillierungsgrad hat, sind die entsprechenden Vorgängeranforderungen zu ergänzen. Die Aktivitäten dazu sind im Rahmen des Vorgehensmodells (s. Kapitel 7.3) erläutert.

Jede PSS-Anforderung der Systemebene ist aus einem oder mehreren Zielen abgeleitet. Zwischen Zielen und PSS-Anforderungen besteht eine n:m-Beziehung. Jede PSS-Anforderung der Systemebene besitzt eine direkte Verbindung zur Zielebene. Dadurch ist die Existenz dieser Anforderung und somit ihre weitere Realisierung begründet. Das heißt, dass ein Ziel der Zielebene durch mehrere PSS-Anforderungen der Systemebene konkretisiert wird und eine Anforderung der Systemebene mehrere Ziele der Zielebene konkretisiert.

Wenn die für das PSS relevanten Stakeholder identifiziert sind, werden ihre Anforderungen ermittelt und in den Anforderungsartefakten zusammengetragen. Falls Anforderungen, wie Kunden- und Stakeholderanforderungen, Umgebungsanforderungen, Auftragnehmeranforderungen oder Geschäftsprozessanforderungen, aus früheren Projekten existieren und wiederverwendet werden sollen, ist ihr Detaillierungsgrad mit dem Detaillierungsgrad der PSS-Anforderungen der Systemebene zu überprüfen und, falls keine Konflikte zwischen den neuen und existierenden Anforderungen entstehen, sind diese Anforderungen aufzunehmen und in die jeweiligen Artefakte zu platzieren.

6.3.2.1 Anforderungsartefakte Systemebene

Die Anforderungen des Kunden und weiterer Stakeholder seitens des Kunden, zu denen bspw. Investoren, Management, Marketing oder Käufer gehören, sind grundlegend für die Entwicklung eines Systems (vgl. (Halen et al. 2005, 57; Lindemann 2006, 95; Ramaswamy 1996, 18; Ramesh et al. 2008)). Die Erfüllung dieser Anforderungen stellt die Kundenzufriedenheit sicher und trägt somit zum Erfolg eines Produktes oder eines Systems bei (Kotonya/Sommerville 1998, 55). Das Anforderungsartefakt „*Kunden- und Stakeholderanforderungen*“ fasst diese Anforderungen zusammen. Diese Anforderungen drücken Wünsche, Vorstellungen und Erwartungen des Kunden und weiterer Stakeholder seitens des Kunden bzgl. des PSS und des Nutzens, den das PSS dem Kunden bringt, aus. Den Anforderungen liegen die Fragen „Was soll das PSS leisten?“, „Welche Vor- und Nachteile bringt das PSS mit sich?“, „Gibt es Einschränkungen bzgl. der Bereitstellung und Nutzung des PSS?“ zu Grunde (vgl. (Robertson/Robertson 2007, 55-58; Ehrlenspiel 2002, 364)). Im Gegensatz zu den Geschäftszielen des Kunden aus der Zielebene, die die Soll-Vorstellungen des Kunden ausdrücken, drücken die Kunden- und Stakeholderanforderungen konkrete Eigenschaften des PSS (Soll-Eigenschaften) aus. Als Grundlage zur Ableitung der Kunden- und Stakeholderanforderungen dienen die Ziele der Zielebene, die den Lösungsraum bestimmen und somit die inhaltliche Richtung der zu definierenden Anforderungen angeben.

Das fertige PSS soll in den Wertschöpfungsprozess des Kunden, d. h. in bestehende Systemlandschaften und Prozesse aber auch in die Nutzungs-, Entwicklungs- und Betriebsprozesse des Kunden, eingebunden sein (Böhm/Krcmar 2007, 244; Grönroos 2000). Der Anbieter des PSS muss sich mit den Abläufen des Kunden genau auskennen, um die Integration des PSS in die Leistungserstellung des Kunden erfolgreich zu gestalten (Sturm/Bading 2008,

176). Dafür müssen die Umgebung, in die das fertige PSS eingesetzt wird, und die relevanten Geschäftsprozesse des Kunden betrachtet werden (Kotonya/Sommerville 1998, 54-55). Ein Geschäftsprozess setzt sich aus den Teilprozessen zusammen, die für eine Kundenleistung erforderlich sind (Gernert/Köppen 2006). Geschäftsprozesse sind „betriebliche Abläufe, die sich entlang einer Wertschöpfungskette identifizieren lassen, unmittelbar auf den Erfolg am Markt ausgerichtet und durch einen messbaren Input, eine Wertschöpfung und einen messbaren Output gekennzeichnet“ (Gierhake 1998, 14). Die Anforderungen, die im Anforderungsartefakt „*Geschäftsprozessanforderungen*“ zusammengefasst sind, werden aus den für die spätere Integration des PSS relevanten Geschäftsprozessen ermittelt. Diesen Anforderungen liegt die Frage zu Grunde „Welche Bedingungen soll das PSS erfüllen, um in die Welt des Kunden integriert zu werden?“ (vgl. (Clements/Bass 2010; Wiegers 2005, 33; Schienmann 2001, 66)). Ähnlich wie bei Kunden- und Stakeholderanforderungen sind die Geschäftsprozessanforderungen aus einem oder mehreren Zielen abgeleitet und somit begründet.

Die Entwicklung und die spätere Nutzung eines Systems werden oft von den Anforderungen beeinflusst, die aus der Umgebung des Systems stammen. Laut IEEE Std 1233 (1998) wird die Umgebung eines Systems von Markt, Gesetzen, technischen Standards, Organisation und Gesellschaft beeinflusst. Demzufolge werden die Anforderungen ermittelt, die durch Gesetze und Richtlinien, Verbraucherverbände, Gesellschaft, Geschäftspartner wie Lieferanten, vorhandene Technologien, Markt oder Wettbewerb vorgegeben werden (vgl. (Maxwell/Anton 2009; Glinz/Wieringa 2007; Schreiner/Strauß 2004; Halen et al. 2005, 57; Husen 2007, 90; Lindemann 2006, 95)). Diese Anforderungen sind im Artefakt „*Umgebungsanforderungen*“ zusammengefasst. Diese Anforderungen können sowohl die Bereitstellung oder die anschließende Nutzung des PSS durch die Stakeholder einschränken als auch den Lösungsraum erweitern. Diesen Anforderungen liegen die Fragestellungen zu Grunde wie „Was soll das PSS nicht leisten?“, „Was verlangt die Richtlinien / Gesetze vom PSS?“ oder „Was soll das PSS besser leisten als der Wettbewerb, um für den Kunden attraktiv zu sein?“ (vgl. (Wiegers 2005, 162)). Die Umgebungsanforderungen sind aus einem oder mehreren Zielen abgeleitet.

Auch der Auftragnehmer (der Lösungsanbieter) hat Anforderungen an das zu entwickelnde PSS. Diese Anforderungen sind im Artefakt „*Auftragnehmeranforderungen*“ zusammengefasst. Ähnlich wie die Kunden- und Stakeholderanforderungen drücken die Auftragnehmeranforderungen allgemeine Wünsche, Vorstellungen, Erwartungen aber auch Einschränkungen des Auftragnehmers bzgl. der Bereitstellung und Nutzung des PSS aus. Zu den Stakeholdern des Auftragnehmers gehören bspw. Entwickler, Marketing und Vertrieb, Management, Standardisierungsgremien, F&E-Abteilungen und Wartung- und Reparaturdienste seitens des Auftragnehmers (Hull et al. 2011, 96-98; Grady 2006, 85-87). Den Anforderungen des Auftragnehmers liegen Fragestellungen zu Grunde wie „Was soll das PSS leisten?“, „Können wir das PSS nach den Kundenwünschen bereitstellen?“ oder „Gibt es Einschränkungen bzgl. der Bereitstellung und Nutzung des PSS?“ (vgl. (Robertson/Robertson 2007, 55-58; Ehrlenspiel 2002, 364)). Die Anforderungen basieren auf den ermittelten Zielen der Zielebene, die im Zusammenhang mit der Bereitstellung und Nutzung des PSS stehen.

6.3.2.1.1 Taxonomien

Die erläuterten Anforderungsartefakte der Systemebene lassen sich hierarchisch zerlegen und durch eine Taxonomie darstellen (s. Kapitel 6.1.3.1). Diese Taxonomie gilt für alle Anforderungsartefakte. Organisatorische Anforderungen, zu denen die Anforderungen an die Bereitstellung des PSS an den Kunden gehören, sind Teil des Projektmanagements und werden im

Rahmen dieser Arbeit nicht betrachtet. Abbildung 6-7 stellt die Taxonomie der Anforderungsartefakte der Systemebene dar.

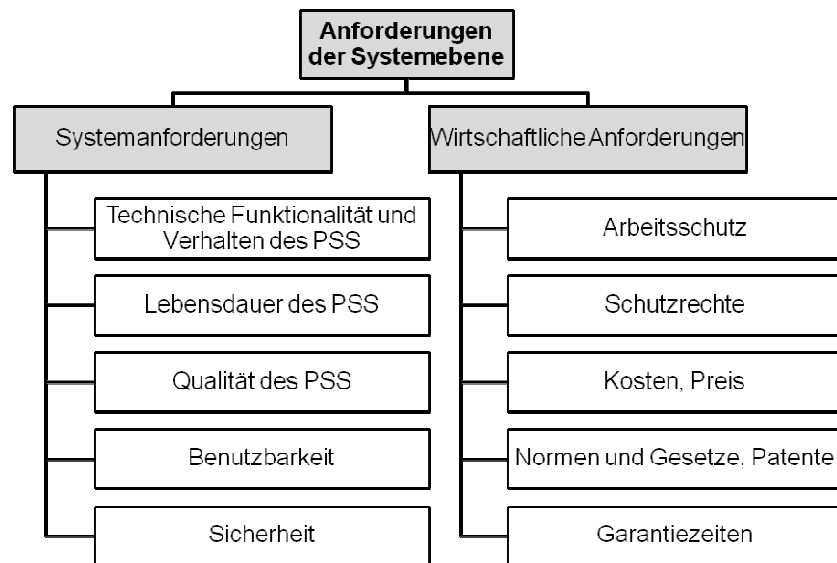


Abbildung 6-7: Taxonomie der Anforderungsartefakte der Systemebene
Quelle: Eigene Darstellung

Die Taxonomie für die Anforderungsartefakte stellt eine Art der Checkliste zur Ermittlung von Anforderungen dar. Die Checkliste soll sicherstellen, dass keine Anforderungen, die nach Kategorien geordnet sind, außer Acht gelassen werden. Im Folgenden werden die einzelnen Kategorien dieser Taxonomien detailliert beschrieben (vgl. (Rupp 2004, 260f.; Husen 2007, 90; Wiegers 2005, 98ff.; Sommerville 2004, 122; Ebert 2005, 11; Ehrlenspiel 2002, 372)).

Die *Systemanforderungen* beschreiben die Anforderungen an das PSS, indem das PSS im Sinne eines komplexen Systems, das aus mehreren von verschiedenen Domänen entwickelten Komponenten besteht, aufgefasst wird. Die Anforderungen an die *technische Funktionalität und das Verhalten des PSS* drücken aus, was das PSS aus technischer Sicht leisten und welche Aufgaben es ausführen soll, damit die definierten Ziele des Kunden und des Auftragnehmers erfüllt sind. Diese Anforderungen gehen nicht der Frage nach, wie das PSS die von ihm erwarteten Leistungen erfüllen soll (vgl. (Hull et al. 2004, 118)). Die Anforderungen an die *Lebensdauer* des PSS beschreiben die Gebrauchsdauer des PSS, die Betriebszeit und die Einsatzdauer (vgl. (Breiing/Flemming 1993, 21)). *Qualitätsanforderungen* definieren die Qualität des PSS, wie bspw. Performance, Verfügbarkeit des PSS, Zuverlässigkeit, Effizienz, Flexibilität des Einsatzes von PSS oder Fehlerfreiheit (vgl. (Lamsweerde 2009, 24)). Die Anforderungen an die *Benutzbarkeit* des PSS beschreiben die Interaktion zwischen dem PSS als ganzer Lösung und dem Kunden. Diese Anforderungen geben an, wie benutzerfreundlich das PSS selbst und die vom PSS angebotenen Funktionen sein sollen (vgl. (Wiegers 2005, 224)). Die Anforderungen an die *Sicherheit* beschreiben, wie sicher das PSS sein soll, damit keine Gefahr für die Nutzer und weitere Stakeholder sowohl aus technischer Sicht als auch aus Informationssicht bestehen (vgl. (Lamsweerde 2000; Sutcliffe 2002, 11-13)).

Die *wirtschaftlichen Anforderungen* definieren die ökonomischen, gesellschaftlichen oder rechtlichen Rahmenbedingungen an das PSS. Dazu gehören die Anforderungen an den *Arbeitsschutz* des an der Bereitstellung oder der Nutzung des PSS beteiligten Personals sowohl seitens des Auftragnehmers als auch des Kunden. Die Anforderungen an die *Schutzrechte*

definieren gewerbliche Schutzrechte, die das geistige Eigentum an der Idee des PSS schützen und die Nachahmung verhindern sollen (vgl. (Ehrlenspiel 2002, 372)). Die Anforderungen an die *Kosten* und den *Preis* des PSS legen die ökonomischen Rahmen für die Bereitstellung des PSS sowohl für den Kunden als auch für den Auftragnehmer fest. Dazu gehören die Anforderungen an das Kostenziel oder an die Inbetriebnahmekosten. Aus *Normen* und *Gesetzen* ergeben sich ebenso Anforderungen, die sowohl die Bereitstellung als auch die Nutzung des PSS bestimmen (vgl. (Sommerville 2004, 122; Halen et al. 2005, 57; Penzenstadler/Leuser 2008)). Dazu gehören bspw. Anforderungen zur Umweltentlastung (Breiing/Flemming 1993, 23) oder die Anforderungen aus dem Produkthaftpflichtrecht (Kullmann 2010). Die Anforderungen an die *Garantiezeiten* geben Angaben zur Wartung, Reparatur und Instandhaltung.

Wie aus oben dargelegten Definitionen ersichtlich wird, beeinflussen sich die Anforderungen der verschiedenen Kategorien stark. Deswegen soll bei der Definition der Anforderungen aller Kategorien überprüft werden, ob sie sich gegenseitig einschränken oder ob weitere Details ergänzt werden können.

6.3.2.2 Externe Artefakte der Systemebene

Zu den externen Artefakten der Systemebene (Tabelle 6-2) gehören verschiedene Anforderungsquellen (vgl. (Halen et al. 2005, 57; Husen 2007, 90; Atlee/Wieringa 2006; Hull et al. 2004; Lindemann 2006, 95; Lamsweerde 2009, 30; Wiegers 2005, 98ff.; Pohl 2007, 66-68)). Die angegebenen externen Artefakte dienen als Checkliste zur Identifikation der potentiellen Anforderungsquellen. Die Checkliste enthält Typen von Anforderungsquellen, für die überprüft wird, in wieweit sie im Kontext des zu entwickelnden Systems relevant sind.

Anforderungsquellen	Beispiele
Kunden und die vom Kunden als wichtig und relevant für die Bereitstellung des PSS erachteten Stakeholder	<ul style="list-style-type: none"> • Nutzer, die das PSS in Anspruch nehmen, • Einkäufer, die die Kaufverträge für das PSS abschließen, • Beeinflusser, die als Experten zur Meinungsbildung über das PSS herangezogen werden, • Entscheider, die Entscheidungskompetenz aufgrund ihrer Position im Unternehmen besitzen, • Endkunden, die von der Nutzung des PSS direkt profitieren.
Mitarbeiter des Kunden und des Auftragnehmers (Lösungsanbieters)	<ul style="list-style-type: none"> • Betroffene Mitarbeiter: Requirements Engineering Analysten, Entwickler, Architekten, Tester, Qualitätsverantwortlicher, Wartung und Reparatur, • Betriebsrat.
Management des Kunden und des Auftragnehmers (Lösungsanbieters)	<ul style="list-style-type: none"> • Marketing, • Vertrieb, • Qualitätsmanagement, • Personal, • IuK, • Finanzen und Controlling, • Verwaltung, • Fertigung, • Entwicklung.

Geschäftspartner	<ul style="list-style-type: none"> • Lieferanten, • Berater.
Investoren	<ul style="list-style-type: none"> • Banken, • Kapitalgeber.
Technologien	<ul style="list-style-type: none"> • Produkttechnologie, • Softwaretechnologie.
Öffentlichkeit	<ul style="list-style-type: none"> • Staat, • Gesellschaft, • Gesetzgeber, • Markt, • Standards, • Normen.
Dokumente, die als Anforderungsquellen dienen	<ul style="list-style-type: none"> • Lasten- und Pflichtenhefte von Vorgängersystemen (falls relevant), • Architekturdokumente von Vorgängersystemen (falls relevant), • Beschreibungen von Geschäftsprozessen des Kunden, • Fehlerberichte von Vorgängersystemen (falls relevant).
Existierende Systeme, falls sie für die Bereitstellung des PSS relevant sind.	<ul style="list-style-type: none"> • Benutzerdokumente, • Modelle, • Handbücher.

Tabelle 6-2: Externe Artefakte der Systemebene

Quelle: Eigene Darstellung

6.3.2.3 Spezifikationsartefakte der Systemebene

Initiale Anforderungen an ein System werden aus der Sicht der Stakeholder erläutert. Meistens sind sie unpräzise und unstrukturiert. Sie können in natürlicher Sprache oder mit Hilfe informeller Diagramme dargestellt werden (Sommerville/Sawyer 1997a, 7; Pohl 2007, 239). Zur Dokumentation von Anforderungen können Attribute verwendet werden, die die Anforderungen eindeutig charakterisieren. Die für die Anforderungen an PSS relevanten Attribute sind in Kapitel 6.2.2 beschrieben.

Zwei gängige Formen zur natürlichsprachlichen Dokumentation im deutschsprachigen Raum stellen Lastenhefte und Pflichtenhefte dar. Das Lastenheft beschreibt die Forderungen bzgl. der Bereitstellung und weiterer Leistungen an den Auftragnehmer (Pohl/Rupp 2009, 46). Somit enthält das Lastenheft die Ziele der Zielebene und die Anforderungen der Systemebene, die die lösungsneutralen Anforderungen an das zu entwickelnde PSS beschreiben (vgl. (Alpar et al. 2008, 73-75; Schäppi et al. 2005, 15-18; Schienmann 2001, 141-150)). Das Pflichtenheft enthält lösungsorientierte Anforderungen, d. h. Anforderungen, die eine bestimmte Lösung voraussetzen. Das sind Realisierungsvorgaben an den Auftragnehmer (Pohl/Rupp 2009, 46). Pflichtenhefte werden im Rahmen der Domänen-Anforderungen der Komponentenebene behandelt (s. Kapitel 6.3.5.4).

6.3.2.3.1 Anforderungsdokumente

Ein Anforderungsdokument soll vollständig, korrekt, klassifizierbar, konsistent, prüfbar, eindeutig, verständlich und verfolgbar sein (IEEE 1998, 4). Ebenso lassen sich Qualitätskriterien auf einzelne dokumentierte Anforderungen anwenden. So soll eine Anforderung folgende

Eigenschaften aufweisen (vgl. (IEEE 1998; Wiegers 2005, 22-26; Hsia et al. 1993; Pohl/Rupp 2009, 52)):

- *Abgestimmt*: eine Anforderung soll für alle Stakeholder korrekt sein und von allen Stakeholdern akzeptiert sein.
- *Bewertet*: die Stakeholder müssen die Anforderungen z. B. nach ihrer Wichtigkeit oder Relevanz bewerten.
- *Eindeutig*: eine eindeutig dokumentierte Anforderung kann von den Stakeholdern nur auf eine Art und Weise verstanden werden.
- *Gültig und aktuell*: eine Anforderung muss im Rahmen des betrachteten Systemkontextes gültig sein.
- *Korrekt*: eine Anforderung muss den Wünschen der Stakeholder entsprechen.
- *Konsistent*: eine Anforderung muss gegenüber allen anderen Anforderungen widerspruchsfrei sein.
- *Prüfbar*: die Funktionalität, die eine Anforderung beschreibt, muss testbar und messbar sein.
- *Realisierbar*: die Anforderung soll innerhalb der definierten Rahmenbedingungen umsetzbar sein.
- *Verfolgbar*: der Ursprung einer Anforderung, deren Umsetzung und die Interdependenzen zu den anderen Anforderungen sollen nachvollziehbar sein.
- *Vollständig*: eine Anforderung soll die geforderte Funktionalität vollständig beschreiben.
- *Verständlich*: die Anforderungen sollen für alle Stakeholder und Entwickler verständlich sein.

6.3.2.3.2 Szenarien

Eine weitere Form der Anforderungsdokumentation in natürlicher Sprache bieten Szenarien an. Ein Szenario ist definiert, als eine Beschreibung eines oder mehrerer Ereignisse, die stattfinden können (vgl. (Gough et al. 1995; Jarke et al. 1999; Sutcliffe 2002, 106-109)). Dabei soll die Beschreibung der Ereignisse aus der Sicht der Stakeholder erfolgen (Hui/Ohnishi 2003). Die Anforderungen lassen sich in Form von Szenarien darstellen, indem konkrete, aus den Anforderungen resultierende Situationen, in Form von Interaktionsfolgen beschrieben werden, die bei der Bereitstellung und der Nutzung des PSS eintreten können. Eine Ausprägung von Szenarien stellen Use Cases (Anwendungsfälle) dar (vgl. (Jacobson et al. 1992; Alspaugh et al. 2007; Jakimi et al. 2007)). Ein Use Case kann Funktionen des PSS beschreiben, die dem Benutzer zur Verfügung gestellt werden. Desweiteren wird beschrieben, wie diese Funktionen mit vorhandenen weiteren externen Interaktionspartnern in Verbindung stehen (vgl. (Pohl/Rupp 2009, 44)). Ein Use Case ist eine Gruppierung aller möglichen Szenarien, die beim Erreichen eines bestimmten Ziels beitragen (Wiegers 2005, 134). Use Cases werden in Prosa und in Form konzeptueller Modelle (Use Case-Diagramme) präsentiert. Use Cases sind besonders zur Darstellung der Anforderungen aus der Nutzersicht, geeignet. Ebenso können Use Cases verwendet werden, um die Anforderungen an die Geschäftsprozesse und die Geschäftsprozesse selbst innerhalb einer Geschäftseinheit darzustellen. Dazu gehören bspw. geschäftliche Abläufe oder Benutzerinteraktionen (Rupp 2004, 182-185).

6.3.2.3.3 Konzeptionelle Modelle

Oft werden die Anforderungen in Form von konzeptuellen Modellen dargestellt. Eine graphische Repräsentation eines Modells wird als Diagramm bezeichnet (Maciaszek 2007, 68). Mit Hilfe von Entity-Relationship-Diagrammen (ER-Diagramme) und Klassendiagrammen können die Anforderungen dargestellt werden, die die statische Struktur des PSS oder die statischen Eigenschaften des PSS beschreiben (vgl. (Booch et al. 2007; Santos/Machado ; Wiegers 2005, 200; Vidgen 2003)). ER-Diagramme stellen Objekte (Entity) der realen Welt und Relationen (Relationship) zwischen ihnen dar und werden eingesetzt, um reale Gegenstandsbereiche zu beschreiben. Durch die Erweiterung der ER-Diagramme um objekt-orientierte Konzepte, wie Vererbung und Aggregation, entstanden Klassendiagramme (Breu 2001, 12). Zu den wesentlichen Komponenten eines Klassendiagramms gehören Klassen, die Objekte der realen Welt abbilden, Attribute, die die Eigenschaften der Objekte präsentieren, und Methoden zur Beschreibung der Funktionalität einer Klasse (Rumpe 2004, 10-12). ER-Diagramme und Klassendiagramme ermöglichen, den strukturellen Aufbau des zu entwickelnden PSS darzustellen, wie die an der Bereitstellung und Nutzung des PSS beteiligten Akteure und Gegenstände oder dafür benötigte Ressourcen.

Beispiel: Die Anforderungen „Bereitstellung eines individuellen Trainingsplans für die Durchführung sportlicher Aktivitäten“ (in Anlehnung an das Beispiel „*Personal Health Manager als IT-gestütztes, gerätebasiertes und personalisiertes Gesundheitsförderungsprogramm*“, Kapitel 3.2.2) kann bspw. mit Hilfe eines Klassendiagramms dargestellt werden. In einem Klassendiagramm werden der Sportler, der Trainingsplan und die sportlichen Aktivitäten als Klassen modelliert. Die Relation zwischen Sportler und Trainingsplan beschreibt, dass jeder Sportler genau einen Trainingsplan besitzt und dass dieser Trainingsplan nur ihm individuell zugeordnet ist. Die Relation zwischen dem Trainingsplan und den sportlichen Aktivitäten beschreibt, dass der Trainingsplan aus einer Reihe von sportlichen Aktivitäten besteht und dass jede dieser Aktivitäten jedoch in verschiedenen Trainingsplänen vorkommen darf.

Aktivitätsdiagramme, Zustandsdiagramme sowie ereignisgesteuerte Prozessketten (EPK) eignen sich gut zur Darstellung von Anforderungen, die Abläufe und Verhalten jeglicher Art beschreiben (vgl. (Rupp 2004, 205ff.; Lamsweerde 2009, 160ff.; Gross/Doerr 2009)). Diese Diagramme eignen sich gut, um die Anforderungen an die Geschäftsprozesse und die Geschäftsprozesse selbst zu modellieren (Brugger 2005, 251f.). Ein Aktivitätsdiagramm stellt die Bearbeitungsschritte (Aktivitäten) eines Gesamtablaufs dar, bei denen Verzweigungen und Nebenläufigkeit der einzelnen Schritte erlaubt sind. Zustandsdiagramme stellen eine Folge von Zuständen dar, die ein Objekt einnehmen kann. Wenn ein Ereignis auftritt, das zu einer Änderung eines Zustands und dem Eintreten eines anderen Zustandes führt, wird dies durch Zustandsübergänge dargestellt. Zur Modellierung betrieblicher Abläufe werden oft EPK eingesetzt, indem betriebliche Funktionen in der Reihenfolge ihrer Ausführung modelliert werden. EPK ermöglichen, ausgehend von einem Startereignis, Funktionen mit Ereignissen zu verbinden (Krcmar 2010, 143-149). Mit Hilfe von Aktivitätsdiagrammen, Zustandsdiagrammen und EPK lassen sich somit die Interaktionen der Nutzer mit dem PSS, das Verhalten des PSS oder die Geschäftsprozesse des Kunden, in die das PSS integriert werden sollen, beschreiben.

Beispiel: Die Anforderung „Bei der individuellen Betreuung des Sportlers durch den Trainer soll der Trainer dem Sportler die Funktionsweise des Crosstrainers erklären, indem die Anzeigetafeln des Geräts erläutert werden und der Trainer das Gerät vorführt“ (in Anlehnung an

das Beispiel „*Personal Health Manager als IT-gestütztes, gerätebasiertes und personalisiertes Gesundheitsförderungsprogramm*“, Kapitel 3.2.2) kann durch ein Aktivitätsdiagramm dargestellt werden. Dafür wird die Aktivität „Crosstrainer erklären“ hierarchisch in die folgenden zwei Aktivitäten zerlegt: „Anzeigetafeln erläutern“, „Gerät vorführen“.

Konzeptuelle Modelle ermöglichen die Visualisierung spezifischer Aspekte eines Gegenstandsbereiches und dadurch die Reduzierung der Komplexität bei der Betrachtung dieser Aspekte (Pohl 2007, 280). Sie bieten im Vergleich zur natürlichsprachlichen Dokumentation an, den zu untersuchenden Gegenstand aus einer Blickwinkelperspektive zu betrachten und dadurch eine bessere Fokussierung auf die relevanten Aspekte des Gegenstandes zu bekommen (Nuseibeh/Easterbrook 2000).

6.3.2.3.4 Auswahl der geeigneten Spezifikationsform

Die Wahl der Dokumentationsform, sei es die natürlichsprachliche Dokumentation oder die Dokumentation in Form konzeptueller Modelle, hängt von unterschiedlichen Faktoren ab. Die Form der Anforderungsdokumentation soll von allen Stakeholdern akzeptiert werden. Die Wahl einer ungeeigneten Dokumentationsform kann dazu führen, dass die Stakeholder die Anforderungen ablehnen oder ignorieren. Ebenso muss bei der Wahl der Form der Anforderungsdokumentation überprüft werden, in wieweit die Stakeholder die Dokumentationsform verstehen. Die Komplexität des zu entwickelnden PSS ist ebenso für die Dokumentationsform entscheidend, da sich bestimmte Formen eher für triviale Systeme oder für abstrakte Anforderungen eignen.

Die Anforderungsdokumente sollen immer konsistent gehalten werden, d. h. die Dokumentation in Prosa muss immer das Gleiche beschreiben wie die konzeptuellen Modelle. Zudem sollen immer alle Dokumente auf dem aktuellen Stand gehalten werden. Die Anforderungen in diesen Dokumenten sollen immer eindeutig sein (Rupp 2004, 217-219). Die Dokumentation in Prosa ist besonders für konkrete Szenarien geeignet, allerdings kann diese Art der Dokumentation Mehrdeutigkeiten oder Redundanzen enthalten. Konzeptuelle Modelle neigen oft dazu, Designentscheidungen vorwegzunehmen, bieten aber weniger Interpretationsraum für Anforderungen als die Dokumentation in Prosa.

Konzeptuelle Modelle werden ebenso eingesetzt, um die Anforderungen zu analysieren, indem die Modelle helfen, die Anforderungen zu konkretisieren und die Interdependenzen zwischen den Anforderungen zu identifizieren (Pohl 2007, 299). Somit werden konzeptuelle Modelle auch im Rahmen von Requirements Engineering Techniken eingesetzt, auf die in Kapitel 7.3 bei der Diskussion des Vorgehensmodells eingegangen wird.

Durch die Angabe von Verweisen auf Ziele, aus denen die PSS-Anforderungen der Systemebene abgeleitet wurden, und auf Design-Anforderungen der Eigenschaftsebene, die aus den PSS-Anforderungen abgeleitet werden, wird die Pre-Traceability, die den Ursprung einer Anforderung umfasst, und die Post-Traceability, die die nachgelagerten Artefakte umfasst, gewährleistet (Dahlstedt/Persson 2005).

6.3.3 Abstraktionsebene 3: Eigenschaftsebene

Die *Eigenschaftsebene* identifiziert und beschreibt die Sach- und Dienstleistungskomponenten des PSS basierend auf den PSS-Anforderungen. Die Eigenschaftsebene strukturiert das PSS unter Zuhilfenahme von PSS-Anforderungen der Systemebene in Funktionen und bündelt sie in Funktionsstrukturen. Der Zweck der Eigenschaftsebene ist die Aufteilung von PSS-

Anforderungen auf die materiellen (technisches Produkt) und immateriellen (Dienstleistungen) Komponenten des PSS. Abbildung 6-8 zeigt die Artefakte der Eigenschaftsebene in Form eines Objektdiagramms.

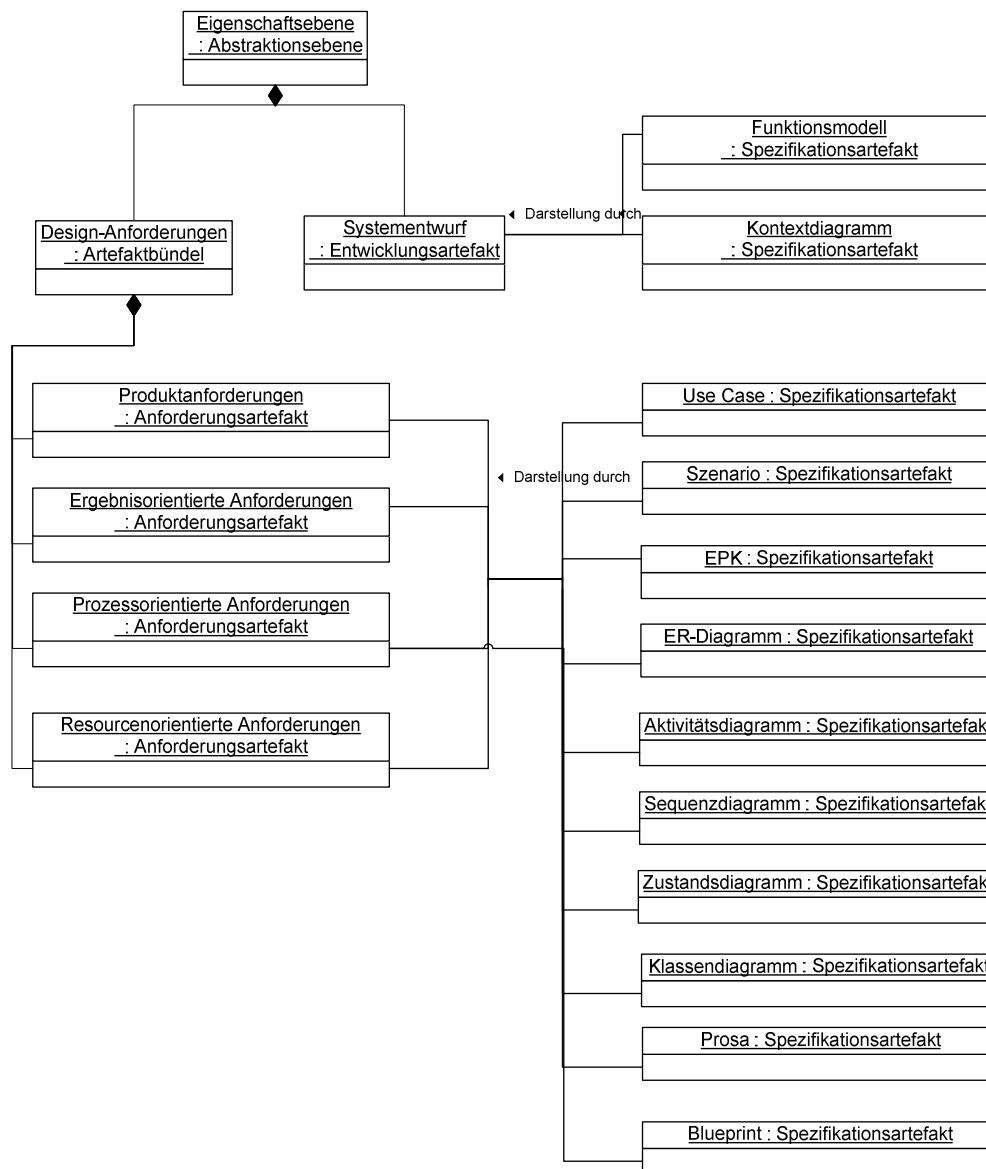


Abbildung 6-8: Artefakte der Eigenschaftsebene

Quelle: Eigene Darstellung

Die Eigenschaftsebene enthält fünf Artefakte: vier Anforderungsartefakte und ein Entwicklungsartefakt, den *Systementwurf*. Zur Darstellung des Systementwurfs können Kontextdiagramme und Funktionsmodelle eingesetzt werden. Die Anforderungsartefakte, „*Produktanforderungen*“, „*Ergebnisorientierte Anforderungen*“, „*Prozessorientierte Anforderungen*“ und „*Ressourcenorientierte Anforderungen*“, sind in einem Anforderungsartefaktbündel „*Design-Anforderungen*“ zusammengefasst. Die Anforderungsartefakte werden ähnlich wie die PSS-Anforderungen der Systemebene durch die Spezifikationsartefakte „*Use Cases*“, „*Szenario*“, „*EPK*“, „*ER-Diagramm*“, „*Aktivitätsdiagramm*“, „*Sequenzdiagramm*“, „*Zustandsdiagramm*“, „*Klassendiagramm*“ und „*Prosa*“ dargestellt. Die Anforderungen an den Prozess der Dienstleistungserbringung können durch *Blueprints* dokumentiert werden.

Nachdem die Anforderungen an das PSS ermittelt und dokumentiert sind, wird darauf basierend im Rahmen der Phase „Produktmodellierung“ des Entwicklungsprozesses der Teil der Umgebung des PSS definiert, der für das Verständnis von Anforderungen an das PSS relevant ist (s. Kapitel 3.3.1). Anschließend werden die materiellen und immateriellen Bestandteile des PSS identifiziert, die zum ausgewählten Teil der Umgebung gehören, und in einem Systementwurf festgehalten. Der *Systementwurf* stellt das domänenübergreifende Konzept zur Realisierung der Anforderungen an das PSS dar, bei dem die wesentlichen Komponenten und die Funktionalitäten des PSS beschrieben sind (vgl. (Bertsche et al. 2009, 21-23; Laudon et al. 2009, 13-15)). Der Systementwurf gibt demzufolge an, aus welchem technischen Produkt, das eine Zusammensetzung aus Hardware und Software darstellt, und aus welchen Dienstleistungen das PSS besteht, ohne auf die eigentliche Ausgestaltung der Komponenten, wie bspw. die genauen Angaben zu der Funktionsweise der Software- oder Hardwarekomponenten oder der Realisierung des Dienstleistungserbringungsprozesses, einzugehen.

Wenn das technische Produkt und die Dienstleistungen des PSS identifiziert sind, sind die initialen PSS-Anforderungen zu analysieren und bzgl. der identifizierten Bestandteile zu konkretisieren. Dafür werden die Anforderungen dem technischen Produkt und den Dienstleistungen des PSS zugeordnet. Anschließend erhalten die Anforderungen weitere konkretere Details bzgl. der Eigenschaften dieser Bestandteile.

Jede Design-Anforderung der Eigenschaftsebene ist direkt auf eine oder mehrere PSS-Anforderungen der Systemebene zurückzuführen. Zwischen PSS-Anforderungen und Design-Anforderungen besteht eine n:m-Beziehung. Das heißt, dass eine PSS-Anforderung der Systemebene durch mehrere Design-Anforderungen der Eigenschaftsebene konkretisiert wird und eine Anforderung der Eigenschaftsebene mehrere PSS-Anforderungen der Systemebene konkretisiert. Wie im Falle der Anforderungen der Zielebene und der Systemebene beeinflussen sich die Design-Anforderungen gegenseitig. Dies soll bei der Konkretisierung der Anforderungen berücksichtigt werden, indem die schon ermittelten Design-Anforderungen der Eigenschaftsebene zur Ermittlung weiterer Anforderungen der Eigenschaftsebene einbezogen werden sollen (s. Kapitel 6.1.3.5).

6.3.3.1 Entwicklungsartefakte der Eigenschaftsebene

Der Systementwurf wird nicht durch die Aktivitäten des Requirements Engineering, sondern von der Entwicklung erstellt. Die Erstellung des Systementwurfs ist ein Entwicklungsschritt, der unter Zuhilfenahme der PSS-Anforderungen aus der Systemebene erfolgt. Deswegen wird die Konzeption des Systementwurfs im Rahmen dieser Arbeit nur skizziert und nicht ausführlich erläutert.

Der Systementwurf besteht aus zwei wesentlichen Teilen, die an dieser Stelle eingeführt werden, um einen Überblick zu geben. Deren detaillierte Beschreibung folgt in zwei Unterkapiteln.

1. Systemkontext: Dieser beschreibt die Umgebung des PSS und definiert die Systemgrenze, die das eigentliche PSS von seiner direkten, durch den Systemkontext vorgegebenen Umgebung separiert.
2. Funktionsstrukturen: Diese legen die Funktionalität des PSS in verschiedenen Granularitätsstufen fest. Die Gesamtfunktionalität des PSS wird durch Hauptfunktionen ausgedrückt, die in dieser Abstraktionsebene (Eigenschaftsebene) festgelegt werden. Diese Hauptfunktionen werden in der nächsten Abstraktionsebene (Funktionsebene) wei-

ter zerlegt und detailliert. Für die Erstellung der Funktionsstrukturen kommt die Funktionsmodellierung (s. Kapitel 7.3.3 und Kapitel 7.3.4) zum Einsatz.

6.3.3.1.1 Systemkontext

Der *Systementwurf* definiert den *Systemkontext* (siehe Abbildung 6-9), der den für das Verständnis der Anforderungen relevanten Teil der Umgebung des zu entwickelnden PSS festlegt (vgl. (Pohl 2007, 55ff.; Hull et al. 2011, 118-121; Ehrlenspiel 2002, 22-25)). Somit wird erreicht, dass die Teile der Umgebung, die für die Bereitstellung und Nutzung des PSS unwichtig sind, nicht betrachtet werden. Zum Systemkontext gehören Prozesse oder Systeme, die das PSS direkt oder indirekt beeinflussen. Ein weiterer Bestandteil des Systementwurfs ist die *Systemgrenze*, die das eigentliche PSS von seiner direkten durch den Systemkontext vorgegebenen Umgebung separiert (vgl. (Pohl 2007, 97)). Dabei wird der Systemkontext durch die Bereitstellung oder Nutzung des PSS nicht verändert. Durch die Identifikation des Systemkontextes und die Angabe der Systemgrenze wird genau definiert, welche Objekte (Personen, Informationen oder Systeme) zum PSS gehören, das PSS beeinflussen, und welche Objekte für die Bereitstellung und Nutzung des PSS nicht relevant sind. Diese Informationen werden zu einer weiteren Konkretisierung von Anforderungen benötigt, damit die Anforderungen entsprechend der vorliegenden Situation richtig interpretiert werden (Pohl 2007, 55ff.).

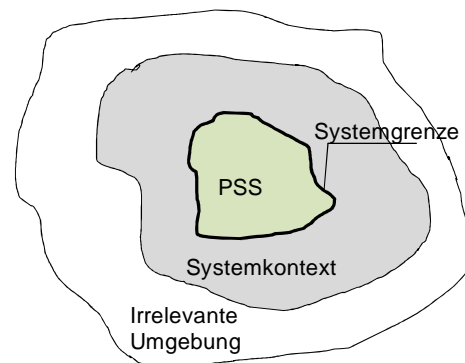


Abbildung 6-9: Systemkontext und Systemgrenze eines PSS

Quelle: In Anlehnung an Pohl (2007, 56)

Die Identifikation der Systemgrenze und der Umgebung des PSS gibt keinen Aufschluss auf die Struktur oder die Funktionalitäten des PSS. Durch die Betrachtung des PSS als eine Black-Box werden die Wechselwirkungen des PSS mit seiner Umgebung verdeutlicht, um dadurch das Systemverständnis zu unterstützen (Lindemann 2006, 116-118; Ehrlenspiel 2002, 376). Die Darstellung als Black-Box ermöglicht das PSS auf einer höheren Abstraktionsebene zu modellieren und dabei auf den wesentlichen Zweck des PSS, aber nicht auf seine Struktur einzugehen (Ponn/Lindemann 2008, 387). Eine Black-Box hat demnach Ein- und Ausgänge an den Systemgrenzen, über die das PSS mit der Umgebung kommuniziert (Green et al. 2004). Durch die Vernachlässigung des inneren Aufbaus des PSS können sich die Entwickler auf die Interaktion des PSS mit der Umgebung konzentrieren und dadurch die Komplexität des Sachverhaltes reduzieren.

6.3.3.1.2 Funktionsstrukturen: Hauptfunktionen

Die Gesamtfunktion, die die gesamte Leistung, d.h. die zu erfüllende Aufgabe, des zu entwickelnden PSS präsentiert, wird in Teilfunktionen zerlegt, wodurch eine *Funktionsstruktur* entsteht. Die Funktionsstruktur beschreibt die kausalen Abhängigkeiten zwischen den Teil-

funktionen (Ponn/Lindemann 2008, 239f.; Buede 2009, 175ff.; Pahl et al. 2006, 24-26). Eine Funktionsstruktur ist also eine logische Architektur, die angibt, was ein System leisten soll, und die Eingabeinformationen in Ausgabeinformationen unter Verwendung von Kontrollflussinformationen transformiert (Ehrlenspiel 2002, 698-700). Funktionsstrukturen stellen funktionelle Zusammenhänge in Form einer Hierarchie von Funktionen dar (Gausemeier/Feldmann 2006, 159-161).

Auf dieser Abstraktionsebene wird die Gesamtfunktion des PSS in Teilfunktionen zerlegt, die die Funktionalität des technischen Produktes und der Dienstleistungen beschreiben. Diese Teilfunktionen werden als Black-Boxen beschrieben und werden *Hauptfunktionen* genannt. Durch die Hauptfunktionen wird definiert, welche Funktionalitäten das technische Produkt und die Dienstleistungen aufweisen sollen, aber nicht, wie diese Funktionalitäten genau realisiert werden, d. h. die Komponenten des technischen Produktes und der Dienstleistungen werden nicht angegeben. Basierend auf den identifizierten Funktionalitäten sollen von der Entwicklung das eigentliche technische Produkt und die Dienstleistungen ausgewählt werden. Durch die Funktionsmodellierung für PHM (s. Kapitel 3.2.2) wird bspw. definiert, dass das System die Fortschritte des Sportlers analysieren, beobachten und an den Trainer weitergeben sowie eine Selbstdokumentation der während der Sportaktivitäten eingesetzten Geräte (Selbstdokumentation) erlauben soll. Die eingesetzten Geräte sollen in der Lage sein, die Ergebnisse der Sportaktivitäten an das System mitzuteilen. Das System, die eingesetzten Geräte und die Dienstleistungen, wie die Durchführung der Selbstdokumentation oder die Analyse der Sportaktivitäten durch das System, werden aus der Black-Box-Sicht dargestellt. Die Angaben zu den Komponenten des Systems, ihren Interaktionen untereinander und die Angaben zur genauen Durchführung der Dienstleistungen, der dafür benötigten Ressourcen oder dem Grad der Automatisierung werden nicht gemacht. Das detaillierte Vorgehen zur Funktionsmodellierung wird in Kapiteln 7.3.3 und 7.3.4 erläutert.

Durch die Darstellung des PSS in Form einer Black-Box kann die Funktionalität des PSS auf einem abstrakten Niveau beschrieben werden. Um die Funktionalitäten des PSS konkreter zu beschreiben, kann die *Funktionsmodellierung* verwendet werden (Ulrich/Eppinger 2003, 102-105; Galvao/Sato 2005; Meerkamm 2003; Gorschek/Wohlin 2006; Pohl 2007, 563-588; Ramaswamy 1996, 125ff.). Die Funktionsmodellierung stellt in allen Ansätzen das zentrale Bindeglied zwischen Anforderungen und Design dar. Die Funktionsmodellierung soll den Entwickler unterstützen, sich mit dem Zweck des zu entwickelnden Systems zu beschäftigen, bevor er mit der eigentlichen Lösungssuche beginnt (Ehrlenspiel et al. 2005, 55). Sie abstrahiert von konkreten Lösungen und hilft dabei die Aufmerksamkeit auf die zu realisierende Funktion des Produktes zu richten. Ehrlenspiel (2002, 396) definiert eine Funktion als „lösungsneutrale Formulierung des gewollten (geplanten, bestimmungsgemäßen) Zwecks eines technischen Gebildes“. Durch Funktionen ist es möglich, Anforderungen in abstrakter, lösungsunabhängiger Weise darzustellen. Somit wird eine Spezifikation des Produktes in einer einheitlichen Art und Weise erreicht, ohne Entscheidungen über die Komponenten des Produktes vorwegzunehmen. Durch die lösungsneutrale Darstellung der Funktionalitäten des PSS wird der Wettbewerbsvergleich vereinfacht und die Ideensuche durch sinnvolle Abstraktion der Problemstellung unterstützt (vgl. (Engeln 2006, 76)). Zudem ist es nicht möglich, alle Anforderungen unabhängig von Entwurfsentscheidungen zu konkretisieren. Durch die Zuordnung von Funktionen zu Komponenten werden frühe Entwurfsentscheidungen in die Anforderungsanalyse aufgenommen und festgehalten.

6.3.3.1.3 Zusammenfassung

Zusammenfassend lässt sich der Systementwurf als eine Zusammensetzung aus dem Systemkontext und den Hauptfunktionen des PSS definieren. Der Systemkontext gibt die Systemgrenze an, d.h. er definiert, was zum PSS gehört und mit welchen Teilen der Umgebung das PSS interagiert. Die Hauptfunktionen definieren die Funktionalitäten des PSS als Black-Box. Das bedeutet, es werden lösungsneutrale Angaben zum technischen Produkt und zu den Dienstleistungen gemacht. Es wird also die Art und die Funktionalität der Dienstleistungen definiert, aber keine genaue Implementierung des Dienstleistungserbringungsprozesses, die Ergebnisse des Dienstleistungserbringungsprozesses oder die dafür benötigten Ressourcen angegeben. Ebenso definieren die Hauptfunktionen die Art und die Funktionalität des technischen Produktes, aber nicht ihre Bestandteile und Details ihrer technischen Realisierung.

6.3.3.2 Anforderungsartefakte der Eigenschaftsebene

Aufbauend auf der Aufteilung des PSS in das technische Produkt und die Dienstleistungen sollen die PSS-Anforderungen der Systemebene analysiert, konkretisiert und dem technischen Produkt und den Dienstleistungen zugeordnet werden, um dadurch eine Grundlage zum Aufbau des Funktionsstrukturentwurfs der Funktionsebene anzubieten.

Das Anforderungsartefakt „**Produktanforderungen**“ beschreibt die Anforderungen an das technische Produkt des PSS. Diese Anforderungen sind noch lösungsneutral in der Hinsicht, dass sie die Funktionalität, das Verhalten und weitere rechtliche, qualitative und ökonomische Aspekte des technischen Produktes und nicht seine Hardware- oder Softwarekomponenten beschreiben, die zum Zeitpunkt der Entwicklung noch nicht definiert sind. Somit beantworten diese Anforderungen die Fragen „Was soll das technische Produkt leisten?“, „Welche Vor- und Nachteile bringt das technische Produkt mit sich?“, „Gibt es Einschränkungen bzgl. der Bereitstellung und Nutzung des technischen Produktes?“, „Welche Anforderungen ergeben sich an das technische Produkt seitens der Dienstleistungen von PSS?“. Diese Anforderungen gehen nicht auf die genaue Ausgestaltung des technischen Produktes, wie Ästhetik, Geometrie oder Akustik ein.

Es ergibt sich eine Taxonomie von Anforderungsartefakten für Produkthanforderungen (vgl. (Husen 2007, 90; Wiegers 2005, 98ff.; Sommerville 2004, 122; Pahl et al. 2006, 192-195; Ehrlenspiel 2002, 370ff.; Breiing/Flemming 1993, 15-28; Maciaszek 2007, 80f.)). Abbildung 6-10 stellt die Taxonomie der Produkthanforderungen der Eigenschaftsebene dar. Diese Taxonomie ist detaillierter als die der vorhergehenden Ebene, weil hier auch Bezug auf die technischen Eigenschaften des Produktes gemacht werden kann.

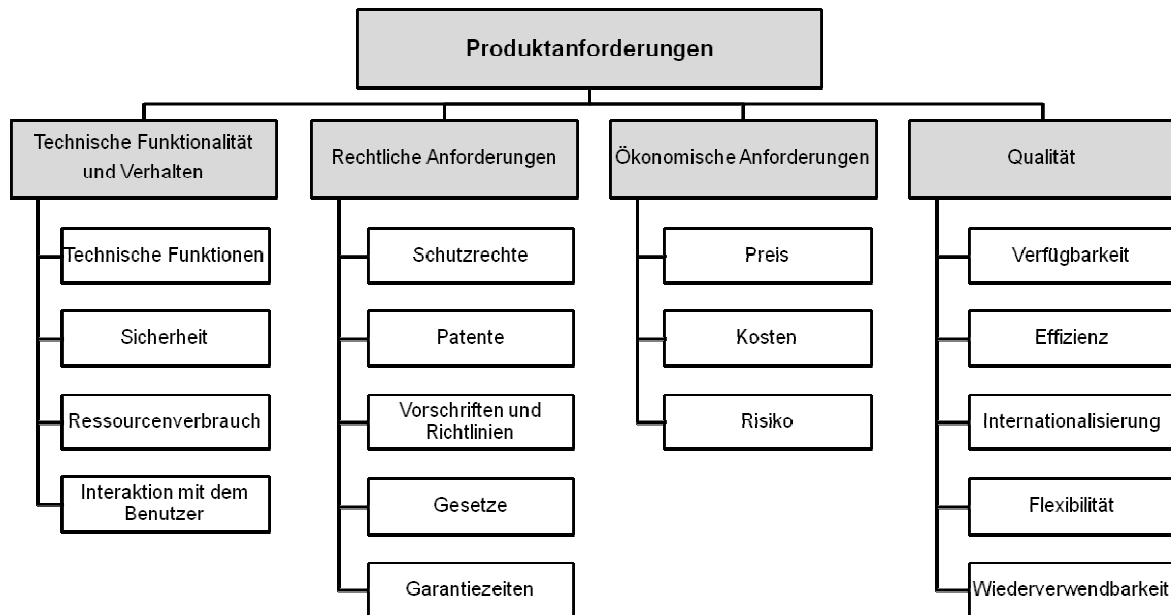


Abbildung 6-10: Taxonomie der Produkthanforderungen der Eigenschaftsebene

Quelle: Eigene Darstellung

Die Anforderungen der Kategorie „*Technische Funktionalität und Verhalten*“ beschreiben das erwartete Verhalten des technischen Produktes während der Bereitstellung oder der Nutzung des PSS aus der technischen Sicht. Das sind Aufgaben, die das technische Produkt erfüllen soll, damit die Stakeholder mit dem gesamten PSS zufrieden sind. Zu dieser Kategorie gehören die Anforderungen, die die Funktionalitäten (*Technische Funktionen*) des technischen Produktes beschreiben, d. h. die Frage beantworten „Was soll das technische Produkt leisten?“ oder „Was darf das technische Produkt nicht leisten?“. Es werden keine „Wie“-Fragen gestellt (vgl. (Hull et al. 2004, 118; Breiing/Flemming 1993, 21; Maier et al. 2007)). Desweiteren gehören zu der Kategorie die Anforderungen an die *Sicherheit* des technischen Produktes, wie die unmittelbare Sicherheitstechnik, Schutzsysteme oder Betriebs-, Arbeits- und Umweltsicherheit (vgl. (Ehrlenspiel 2002, 74-75)). Die Anforderungen an den *Ressourcenverbrauch* beschreiben, welche Ressourcen, wie bspw. Mitarbeiterqualifikationen, Meß- und Prüfmöglichkeiten oder Leistung, benötigt werden, um das technische Produkt zu nutzen (vgl. (Ehrlenspiel 2002, 74-75; Dodrige 1999; Graham 2003; Maiden 2010)). Im Unterschied zu den ressourcenorientierten Dienstleistungsanforderungen werden hier die Anforderungen an die Ressourcen zum Betreiben des technischen Produktes beschrieben. Die Anforderungen an die *Interaktion mit dem Benutzer* beschreiben den Ablauf der Nutzung des technischen Produktes, bspw. die Tätigkeiten der Stakeholder beim Nutzen des technischen Produktes (vgl. (Rupp 2004, 257; Galvao/Sato 2005; Pahl et al. 2006, 405-410)).

Die Anforderungen der Kategorie „*Rechtliche Anforderungen*“ beschreiben die Vorgaben an das technische Produkt, die sich aus den *Gesetzen*, *Vorschriften* und *Richtlinien* ergeben (vgl. (Sommerville 2004, 122; Halen et al. 2005, 57; Penzenstadler/Leuser 2008)). Dazu gehören bspw. das Medizinproduktegesetz an die Entwicklung von medizinischen Geräten (Mauro et al. 2009) oder Anforderungen zur Umweltentlastung durch das technische Produkt (Breiing/Flemming 1993, 23). Die Anforderungen an die *Schutzrechte* und *Patente* definieren gewerbliche Schutzrechte, die das geistige Eigentum an der Idee des technischen Produktes schützen und die Nachahmung verhindern sollen (vgl. (Ehrlenspiel 2002, 372)). Die Anforderungen an die *Garantiezeiten* beschäftigen sich mit dem Anspruch des Kunden an die Garan-

tie des technischen Produktes (vgl. (Breiing/Flemming 1993, 21)). Ebenso gehören die Anforderungen an den Arbeitsschutz des an der Bereitstellung oder der Nutzung des technischen Produktes beteiligten Personals sowohl seitens des Auftragnehmers als auch des Kunden zu der betrachteten Kategorie der Anforderungen.

Die Anforderungen der Kategorie „*Ökonomische Anforderungen*“ beschäftigen sich mit dem *Preis*, *Kosten* und *Risiken* der Bereitstellung und der Nutzung des technischen Produktes. Somit werden die wirtschaftlichen Rahmenbedingungen für die Bereitstellung des technischen Produktes sowohl für den Kunden als auch für den Auftragnehmer festgelegt. Dazu gehören Vereinbarungen über die Herstellkosten, Selbstkosten oder Lebenslaufkosten, der zu erwartende Schadensumfang durch das Eintreten eines Schadensfalls oder die Vereinbarungen bzgl. des Preises des technischen Produktes (vgl. (Ehrlenspiel et al. 2005, 5ff.; Berenbach et al. 2009, 276-283; Breiing/Flemming 1993, 21)).

Die Anforderungen der Kategorie „*Qualität*“ geben Angaben zur Qualität des technischen Produktes, wie *Verfügbarkeit*, *Effizienz*, *Internationalisierung*, *Flexibilität* des Einsatzes von PSS oder *Wiederverwendbarkeit* (vgl. (Lamsweerde 2009, 24; Schäppi et al. 2005, 8ff.; Deubel et al. 2005; Boehm 1996)). Die Anforderungen an die Verfügbarkeit gibt die Zeitintervalle an, an welchen das technische Produkt ordnungsgemäß zur Verfügung steht (Pahl et al. 2006, 301). Ebenso beschreiben die Anforderungen dieser Kategorie die Zuverlässigkeit, die aussagt, wie verlässlich das technische Produkt die von ihm geforderten Funktionen in einem Zeitintervall erfüllt (vgl. (Pahl et al. 2006, 300; Bowen et al. 2002)). Die Angaben zur Robustheit in der Unterkategorie „Verfügbarkeit“ beschreiben die Unfallfähigkeit des technischen Produktes (Regius 2005, 15). Die Anforderungen an die Effizienz geben die Angaben an, wie wirkungsvoll das technische Produkt ist (vgl. (Schäppi et al. 2005, 35ff.; Wiegers 2005, 220-221)). Die Anforderungen an die Internationalisierung zeigen, welche Eigenschaften das technische Produkt besitzen soll, um auch international angeboten zu werden. Die Anforderungen an die Flexibilität geben an, welche Eigenschaften das technische Produkt besitzen muss, um, falls erforderlich, neue Eigenschaften zu erhalten (Wiegers 2005, 221). Ebenso gehören zu den Anforderungen an die Flexibilität die Angaben zum Anpassungsvermögen des technischen Produktes bei sich wechselnden Bedingungen, wie bspw. geänderte Anforderungen der Kunden (Pahl et al. 2006, 63). Die Anforderungen an das Wiederverwenden beschreiben, welche Teile des technischen Produktes wiederverwendbar sein sollen oder welche schon existierenden Elemente im technischen Produkt wiederverwendet werden sollen (vgl. (Wiegers 2005, 226-227; Hardt et al. 2002; Schäppi et al. 2005, 538f.)).

Die Artefakte „*Ergebnisorientierte Anforderungen*“, „*Prozessorientierte Anforderungen*“ und „*Ressourcenorientierte Anforderungen*“ sind drei Anforderungsartefakte, die die Anforderungen an die Dienstleistungen zusammenfassen. Eine Dienstleistung kann anhand der potenzial-, prozess- und ergebnisorientierte Dimensionen charakterisiert werden (Corsten 2001, 26; Scholl 2009, 42-44; Scheer et al. 2003, 25). Husen (2007, 33ff.) fordert die Dienstleistungsdimensionen nicht nur bei der Gestaltung von Dienstleistungen, sondern schon bei der Anforderungsdefinition zu berücksichtigen. Die Erläuterung zu den einzelnen Dimensionen ist in Kapitel 3.1.1.2 zu finden.

Die Anforderungen, die im Anforderungsartefakt „*Ergebnisorientierte Anforderungen*“ zusammengefasst sind, beschreiben das Ergebnis der Dienstleistungserbringung. Das kann ein materielles oder immaterielles Ergebnis sein, das für den Kunden einen gegenständlichen oder zeitlichen Nutzen stiftet (Scholl 2009, 43; Hoeth/Schwarz 2002, 15-16). Das immaterielle

Ergebnis eines Zahnarztbesuchs bedeutet für den Patienten die Verbesserung seines Gesundheitszustandes, da der die Schmerzen verursachende Zahn eine Plombe erhält. Die Plombe ist ein materielles Ergebnis der Dienstleistungserbringung, d. h. der ärztlichen Behandlung. Da das Ergebnis einer Dienstleistung von den individuellen Kundenanforderungen abhängt und diese auf spezifische Weise ausdrückt, ist es nicht möglich, eine Taxonomie für ergebnisorientierte Anforderungen anzugeben.

Die Anforderungen der Kategorie „**Prozessorientierte Anforderungen**“ geben Angaben zur Gestaltung des Dienstleistungserbringungsprozesses (vgl. (Scholl 2009, 43; Scheer et al. 2003, 25; Engelke 1997, 48-54)). Diese Anforderungen beschreiben den Ablauf der Dienstleistungserbringung, bspw. wie die Behandlung eines Patienten bei einem Zahnarztbesuch abläuft welche Aktivitäten dabei zu verrichten sind. Der Kunde ist dabei das prozessauslösende und prozessbegleitende Element für die Umsetzung der Dienstleistung. Für die Anforderungsartefakte der prozessorientierten Anforderungen an die Dienstleistungen, wurde die Taxonomie die in Abbildung 6-11 dargestellt wird, entwickelt. Sie lehnt sich an die Kriterien für prozessorientierte Anforderungen von (Husen 2007, 185) an.

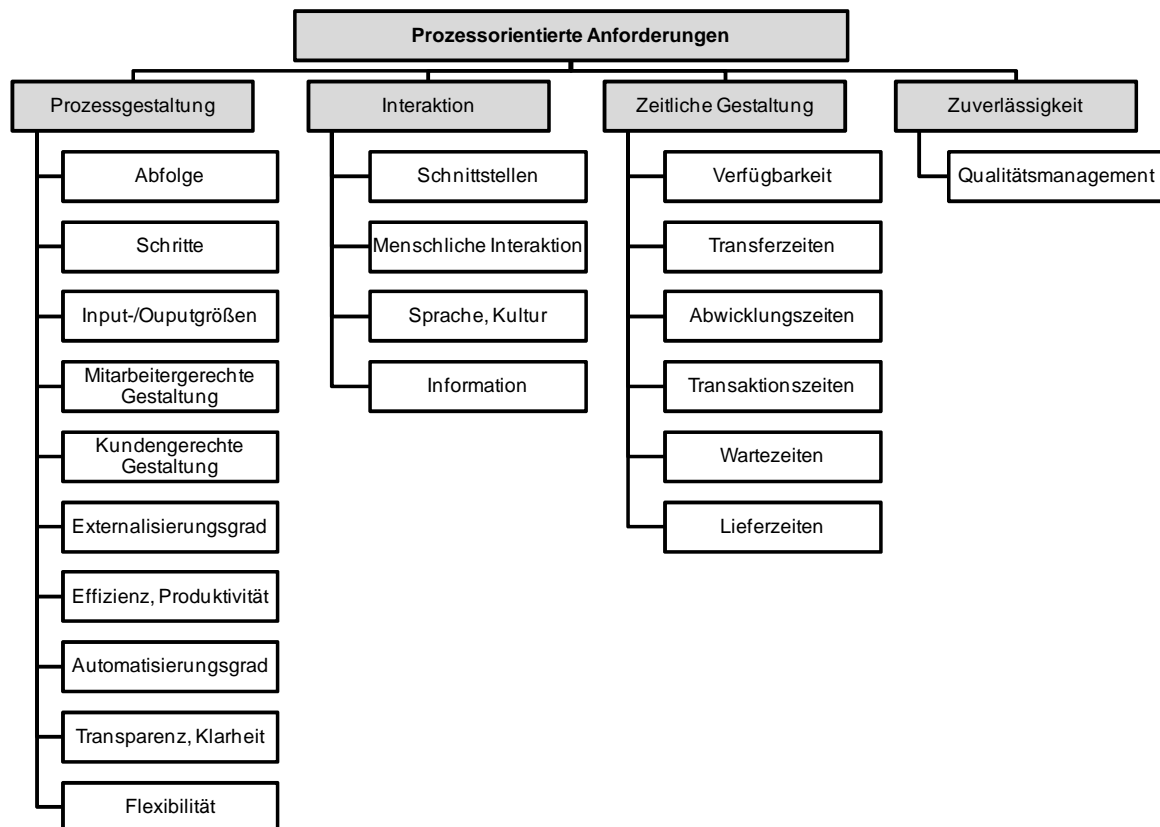


Abbildung 6-11: Taxonomie der prozessorientierten Anforderungen der Eigenschaftsebene

Quelle: Eigene Darstellung

Die Anforderungen der Kategorie „**Prozessgestaltung**“ beschreiben die Aktivitäten der Dienstleistungserbringung. Dazu gehören die *Abfolge* der einzelnen Aktivitäten, die genaue Beschreibung der *Schritte* sowie der *Input-/Outputgrößen*, die zur Durchführung der Aktivitäten benötigt werden (vgl. (Husen 2007, 93; Shostack/Kingman-Brundage 1991; Karni/Kaner 2007)). Die *mitarbeitergerechte Gestaltung* des Dienstleistungserbringungsprozesses steht für eine sichere und einfache Erbringung der Dienstleistungen für die Mitarbeiter des Dienstleis-

tungsanbieters (vgl. (Cook et al. 1999; Meiren/Liestmann 2002, 35-36)). Die *kundengerechte Gestaltung* steht für eine sichere und einfache gestaltete Erbringung der Dienstleistungen für den Kunden (Klein 2007, 20f.). Der *Externalisierungsgrad* zeigt, welchen Umfang von den Aktivitäten zur Dienstleistungserbringung an den Kunden externalisiert wird. Weitere Kategorien bilden die Prozesseffizienz und die Produktivität (*Effizienz, Produktivität*), die Angaben zu den erwarteten Prozessleistungen der Dienstleistungserbringung enthalten (Husen 2007, 93). Die Anforderungen an die Automatisierung beschreiben den *Automatisierungsgrad* des Dienstleistungserbringungsprozesses, welche Teilprozesse automatisiert werden sollen und ob die Automatisierung bspw. durch die Software erfolgt (vgl. (Esch et al. 2010a; Esch et al. 2010b)). Der Dienstleistungserbringungsprozess soll so gestaltet sein, dass er sowohl vom Kunden als auch vom Auftragnehmer nachvollzogen werden kann (*Transparenz, Klarheit*). Die Anforderungen an die *Flexibilität* beschreiben den Anpassungsgrad der Dienstleistungen an die Bedingungen, unter denen sie eingesetzt werden (vgl. (Herrmann et al. 2005, 202)).

Eine weitere Kategorie ist „*Interaktion*“. Dazu gehören die Angaben zu den *Schnittstellen* zu den Geschäftsprozessen des Kunden und des Auftragnehmers bei der Erbringung der Dienstleistungen (vgl. (Stauss/Bruhn 2007; Karwowski/Salvendy 2010, 86ff.)). Die *menschliche Interaktion* bedeutet die Interaktion der am Dienstleistungserbringungsprozess Beteiligten untereinander (Freiling/Gersch 2007). So ist der Kunde gleichzeitig der Co-Produzent der Dienstleistungen und soll seine Wünsche und Vorstellungen über die Erbringung der Dienstleistungen an die zuständigen Mitarbeiter weitergeben (vgl. (Edvardsson/Olsson 1996; Husen 2007, 94)). Mit der Interaktion sind die Angaben zur *Sprache* und *Kultur* sowie die Angaben zur Form der vorliegenden *Information* während der Interaktion verbunden.

Die Kategorie „*Zeitliche Gestaltung*“ enthält die Anforderungen an die *Verfügbarkeit* der Dienstleistungserbringung, wie bspw. Gewährleistung der Reparaturdienste für Waschmaschinen (vgl. (Pahl et al. 2006, 301; Freiling/Gersch 2007)). Die Anforderungen an die *Transferzeiten* (räumliche Entfernung zum Ort der Dienstleistungserbringung), *Abwicklungszeiten* (erforderliche Aktivitäten für die spätere Bereitstellung der Dienstleistung), *Transaktionszeiten* (Zeitintervalle zur eigentlichen Dienstleistungserbringung) und *Wartezeiten* (Dauer bis zur Erbringung der Dienstleistung) liefern Angaben zu den kundenrelevanten Zeitaspekten, die mit der Bereitstellung der Dienstleistung im Zusammenhang stehen (Dreyer 2003; Husen 2007, 94). Wenn zur Erbringung der Dienstleistung entsprechende materielle Komponenten benötigt werden, die dem Kunden erst geliefert werden sollen, werden die Anforderungen an die *Lieferzeiten* definiert (vgl. (Meffert/Bruhn 2006, 596)).

Die Kategorie „*Zuverlässigkeit*“ enthält die Anforderungen an das *Qualitätsmanagement* der Dienstleistungserbringung. Diese Anforderungen definieren die Voraussetzungen für die Dienstleistungserbringung, damit der Kunde die Dienstleistung als Mehrwert wahrnimmt und dadurch mit der Dienstleistung zufrieden ist (vgl. (Bullinger/Schreiner 2003; Edvardsson/Olsson 1996; Zangemeister 2008, 70ff.; Ramaswamy 1996, 291ff.))

Zur Erbringung einer Leistung werden verschiedene Potenziale, wie Humanressourcen oder Maschinen benötigt (Corsten 2001, 21-22; Hoeth/Schwarz 2002, 15; Scholl 2009, 43). Die Anforderungen der Kategorie „**Ressourcenorientierte Anforderungen**“ geben Angaben zu den zur Erbringung der Dienstleistungen erforderlichen Ressourcen. Abbildung 6-12 zeigt eine Taxonomie der Anforderungsartefakte für die ressourcenorientierten Anforderungen.

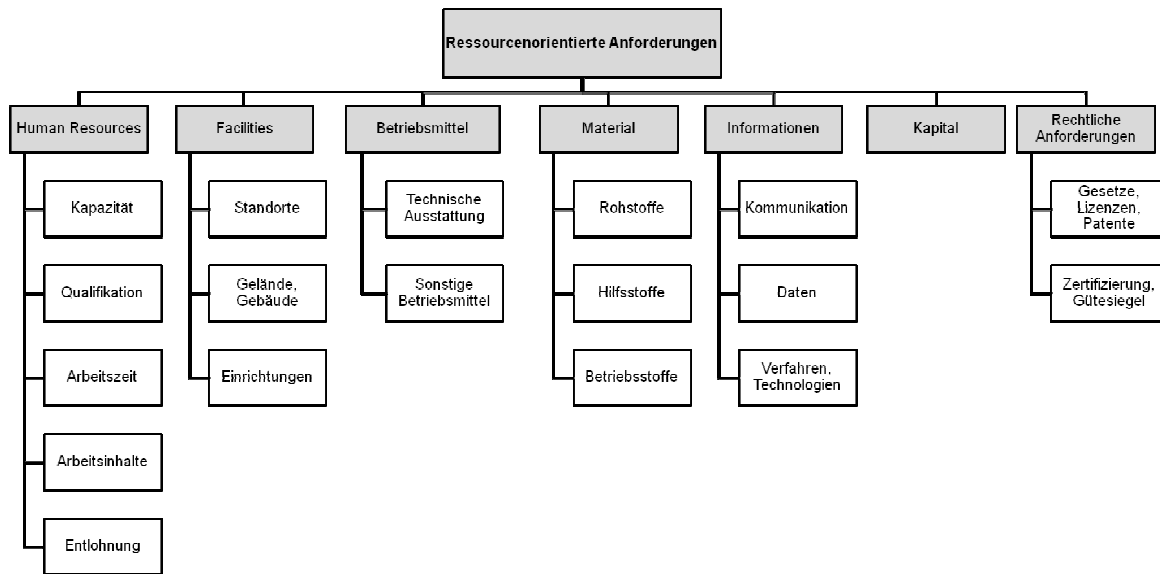


Abbildung 6-12: Taxonomie der ressourcenorientierten Anforderungen der Eigenschaftsebene

Quelle: Eigene Darstellung

Humankapital ist ein wichtiger Faktor bei der Erbringung von Dienstleistungen (Bullinger/Schreiner 2003; Olemotz 1995, 253; Steffen 2006, 77-83). Die Kategorie „*Human Resources*“ enthält die Anforderungen, die die Eigenschaften des zur Erbringung der Dienstleistungen benötigten Humankapitals definieren. Dazu gehören die quantitativ ausgedrückte Personalkapazität (*Kapazität*), die geforderte Qualifizierung (*Qualifikation*) der Mitarbeiter, der durch die Fach- und Sozialkompetenzen und das Verhalten ausgedrückt wird, die zur Erbringung der Dienstleistungen benötigte *Arbeitszeit*, die vereinbarten Inhalte der Dienstleistungserbringung (*Arbeitsinhalte*) und die *Entlohnung* (vgl. (Schmid 2005, 105; Zangemeister 2008, 87-88; Haller 2005, 101ff.)).

Die Kategorie „*Facilities*“ beinhaltet die Anforderungen an die zur Dienstleistungserbringung benötigten *Einrichtungen*, *Gelände* und *Gebäude* oder *Standorte*, an denen die Dienstleistungen erbracht werden. Dazu gehören bspw. die Angaben zur Umgebung, Landschaft, Größe oder Erreichbarkeit (vgl. (Karni/Kaner 2006; Husen 2007, 94-95)).

Zur Erbringung der Dienstleistungen werden verschiedene Betriebsmittel, wie Anlagen, Geräte, Maschinen und weitere Arbeitsmittel benötigt. Die Kategorie „*Betriebsmittel*“ enthält die Anforderungen an die technische Ausstattung der zur Erbringung der Dienstleistungen erforderlichen Ressourcen (vgl. (Zarnekow 2007, 87ff.)). Ein typisches Beispiel für Betriebsmittel ist die Informations- und Kommunikationstechnologie (IuK) (Maleri 1997, 61-62). Die Anforderungen dieser Kategorie können die Angaben an die Ausstattung, Handhabung, Verfügbarkeit oder die Funktionalität der Betriebsmittel (*technische Ausstattung*) enthalten (Husen 2007, 94-95). Imageaspekte und Layout (*sonstige Betriebsmittel*) sind weitere Aspekte, die bei der Planung der Betriebsmittel zu berücksichtigen sind (Bodendorf 1999, 73). Eng mit den Betriebsmitteln verbunden steht die Kategorie „*Material*“, die die Anforderungen an die *Hilfsstoffe* oder Verbrauchsmaterial (*Betriebsstoffe*) enthält. Das Material geht dabei in die Dienstleistung ein (Bodendorf 1999, 72-74). Diese Anforderungen beschreiben bspw. das Material des Kunden, das bei der Erbringung der Dienstleistungen verwendet werden soll, oder Arbeitsmittel (*Rohstoffe*), wie Energieträger (Erdgas, Benzin).

Die Kategorie „*Informationen*“ enthält Anforderungen an die Gestaltung der *Kommunikation* zwischen den an der Dienstleistungserbringung beteiligten Akteuren, den Austausch an *Daten*, wie bspw. Berichte, und an die einzusetzenden *Technologien* und *Verfahren* (vgl. (Karni/Kaner 2006; Husen 2007, 94-95)). Die Regelung der Art und des Umfangs des Know-How-Austausches soll ebenso bei den Anforderungen berücksichtigt werden (Hauck 2009).

„*Kapital*“ ist eine weitere Kategorie, die die Anforderungen an das zur Erbringung der Dienstleistungen verfügbare Kapitel und die mit der Erbringung der Dienstleistung verbundenen Kosten beschreiben (vgl. (Fließ 2009, 65; Maleri 1997, 61)). Die „*Rechtlichen Anforderungen*“ stammen aus *Gesetzen*, *Lizenzen*, *Patenten* oder *Zertifizierungen* (Scheer et al. 2003). Gewerbliche Schutzrechte zum Schutz des Know-How oder Eigentümerverhältnisse sind typische Beispiele für die Gesetze und Normen, die bei der Definition von Anforderungen zu berücksichtigen sind (Hauck 2009).

Wie aus oben dargelegten Definitionen ersichtlich wird, beeinflussen sich die Anforderungen der verschiedenen Kategorien stark. Daher muss bei der Erstellung von Anforderungen, die zu diesen Kategorien gehören, immer überprüft werden, ob sie sich gegenseitig einschränken oder auf andere Weise beeinflussen. Dabei kann es vorkommen, dass aus einer Anforderung zusätzliche Details für eine andere Anforderung abgeleitet werden können.

6.3.3.3 Externe Artefakte der Eigenschaftsebene

Zur Auswahl und späteren Konzeption der Dienstleistungen, die im Zusammenhang mit dem technischen Produkt angeboten werden, kann ein *Katalog zur Systematisierung von Dienstleistungen* verwendet werden (vgl. (Husen 2007, 23; Oguachuba 2009, 47; Bruhn 2008, 20ff.)). Tabelle 10-5 (Anhang) gibt einen Überblick über die Dienstleistungen. Die Systematisierung von Dienstleistungen soll dem Lösungsanbieter ermöglichen, die für sein Lösungsangebot passenden Dienstleistungen bereitzustellen. So kann er zwischen kaufmännischen, wie bspw. Wirtschaftlichkeitsanalysen oder Schulung, und technischen, wie bspw. Wartung oder Reparatur, wählen. Eine Einteilung in Pre-Sales Dienstleistungen, die vor dem eigentlichen Nutzen des technischen Produktes angeboten werden können, Sales Dienstleistungen, die während der Bereitstellung des technischen Produktes angeboten werden können, und After-Sales, die angeboten werden können, nachdem das technische Produkt in Betrieb genommen wurde, ist ebenso möglich.

Ähnlich zu den Dienstleistungskatalogen gibt es Konstruktionskataloge für technische Produkte. Das sind Sammlungen bekannter Lösungen für bestimmte konstruktive Aufgaben (Pahl et al. 2006, 129-135). Solche Kataloge können physikalische Effekte, Wirkprinzipien, prinzipielle Lösungen, Maschinenelemente oder Normteile enthalten (vgl. (Roth 1994; Koller 1997)). Konstruktionskataloge werden eingesetzt, um die Suche nach dem technischen Produkt und seinen Funktionalitäten zu unterstützen.

6.3.3.4 Spezifikationsartefakte der Eigenschaftsebene

Zur Spezifikation des Systemkontextes und der Systemgrenze kann ein *Kontextdiagramm* eingesetzt werden (Hull et al. 2004, 48-55; Rupp 2004, 190-193). Das Kontextdiagramm ist eine abstrahierte Form der Datenflussdiagramme. Es veranschaulicht die Systemumgebung und die Schnittstellen der Umgebung zur Umwelt. Dabei werden die Eingangs- und Ausgangsgrößen des Systems dargestellt. Somit wird die Systemgrenze eindeutig definiert (Wiegers 2005, 90-91; Berenbach et al. 2009, 99-101).

Um die Funktionen, die in Funktionsstrukturen zusammengefasst sind, darzustellen, können *Funktionsmodelle* eingesetzt werden. Sowohl die Gesamtfunktion als auch ihre Teilfunktionen (Hauptfunktionen) sind möglichst lösungsneutral zu formulieren (Lindemann 2006, 116-118). Funktionen werden durch Kombination eines Substantivs mit einem Verb beschrieben, wie z.B. *Wäsche trocknen*, (vgl. (Ramaswamy 1996, 125ff.; Ehrlenspiel 2002, 390-401; Spath/Demuß 2003)). Funktionen sind durch Kommunikationspfade miteinander verbunden. Wird die Funktionsmodellierung auf eine graphenbasierte Repräsentationsform angewendet, lassen sich Funktionsstrukturen als Netzwerke von Eingangs- und Ausgangsbeziehungen darstellen. Die Relationen zwischen Funktionen können verschiedenartiger Natur sein, wie z.B. kausale Beziehungen (*Wäsche trocknen benötigt Energiezufuhr gewährleisten*), oder flussorientierte Beziehungen, die zeigen, welche Eingangsgrößen benötigt werden und welche Ausgangsgrößen entstehen, wie zum Beispiel der Materialfluss *Nasse und saubere Wäsche* zwischen den Funktionen *Wäsche waschen* und *Wäsche trocknen* erfolgt. Kausale Beziehungen werden in der relationsorientierten Funktionsmodellierung verwendet, um die wesentlichen Zusammenhänge im System zu identifizieren und dadurch den Fokus auf die wichtigsten technischen Problemstellungen zu lenken (Ponn/Lindemann 2008, 67; Engeln 2006, 74ff.). Flussorientierte Beziehungen werden in der umsatzorientierten Funktionsmodellierung verwendet, um die Übergänge zwischen den verschiedenen Zuständen eines Produktes zu zeigen (Lindemann 2006, 116-118; Schlink 2004, 62-67). In der Anforderungsanalyse für PSS werden diese beiden Relationsarten vereint. Auf die Konzeption von Funktionsmodellen wird im Rahmen des Vorgehensmodells (s. Kapitel 7.3) in Detail eingegangen.

Zur Dokumentation von Anforderungen werden Spezifikationsformen, wie „Use Case“, „Szenario“, „EPK“, „ER-Diagramm“, „Aktivitätsdiagramm“, „Sequenzdiagramm“, „Zustandsdiagramm“, „Klassendiagramm“ und „Prosa“ gewählt. Die genaue Definition dieser Spezifikationsartefakte wurde im Rahmen von Kapitel 6.3.2.3 gegeben. Dabei werden diese Spezifikationsartefakte zur Darstellung der Anforderungen der Eigenschaftsebene verwendet. Für die Dokumentation in Prosa eignet sich die Struktur eines Lastenheftes, die an die Vorgaben der Eigenschaftsebene angepasst wird. Die Dokumentation der konkretisierten Anforderungen und der jeweiligen Attribute dieser Anforderungen in Prosa ermöglichen den Evolutionsstand der Anforderungen nachzuvollziehen.

Durch die Angabe von Verweisen auf PSS-Anforderungen der Systemebene, aus denen die Design-Anforderungen der Eigenschaftsebene abgeleitet wurden, und auf Funktionsstruktur-Anforderungen der Funktionsebene, die aus den Design-Anforderungen abgeleitet werden, wird die Pre-Traceability, die den Ursprung einer Anforderung umfasst, und die Post-Traceability, die die nachgelagerten Artefakte umfasst, gewährleistet (Dahlstedt/Persson 2005).

Zusätzlich können die Anforderungen an die Dienstleistungsprozesse mit *Blueprints* dokumentiert werden (Shostack 1984). Blueprints ermöglichen den Dienstleistungsprozess in die einzelnen Teilprozesse zu zerlegen und mittels der Ablaufdiagramme darzustellen. Durch diese strukturierte Visualisierung von Dienstleistungsprozessen können die Kunden die Wahl der einzelnen Dienstleistungskomponenten besser nachvollziehen, dadurch Mängel an der Dienstleistungserbringung erkennen und somit konstruktive Verbesserungsvorschläge machen (Boghnim/Yannou 2005; Shostack 1984; Bruhn/Georgi 2005, 88-91; Reckenfelderbäumer/Busse 2003). Die Blueprints ermöglichen den Kunden direkt in den Dienstleistungsprozess einzubinden.

6.3.4 Abstraktionsebene 4: Funktionsebene

Laut dem Entwicklungsprozess des PSS findet während der hybriden Produktkonzeption die Funktionsmodellierung für das technische Produkt und die Dienstleistungen statt (s. Kapitel 3.3.1). Das Ziel der Funktionsebene ist die Hauptfunktionen des Systementwurfs der Eigenschaftsebene so zu zerlegen und zu detaillieren, dass sie anschließend den einzelnen Hardware-, Software- oder Dienstleistungskomponenten zugeordnet werden. Abbildung 6-13 zeigt die Artefakte der Funktionsebene in Form eines Objektdiagramms dar.

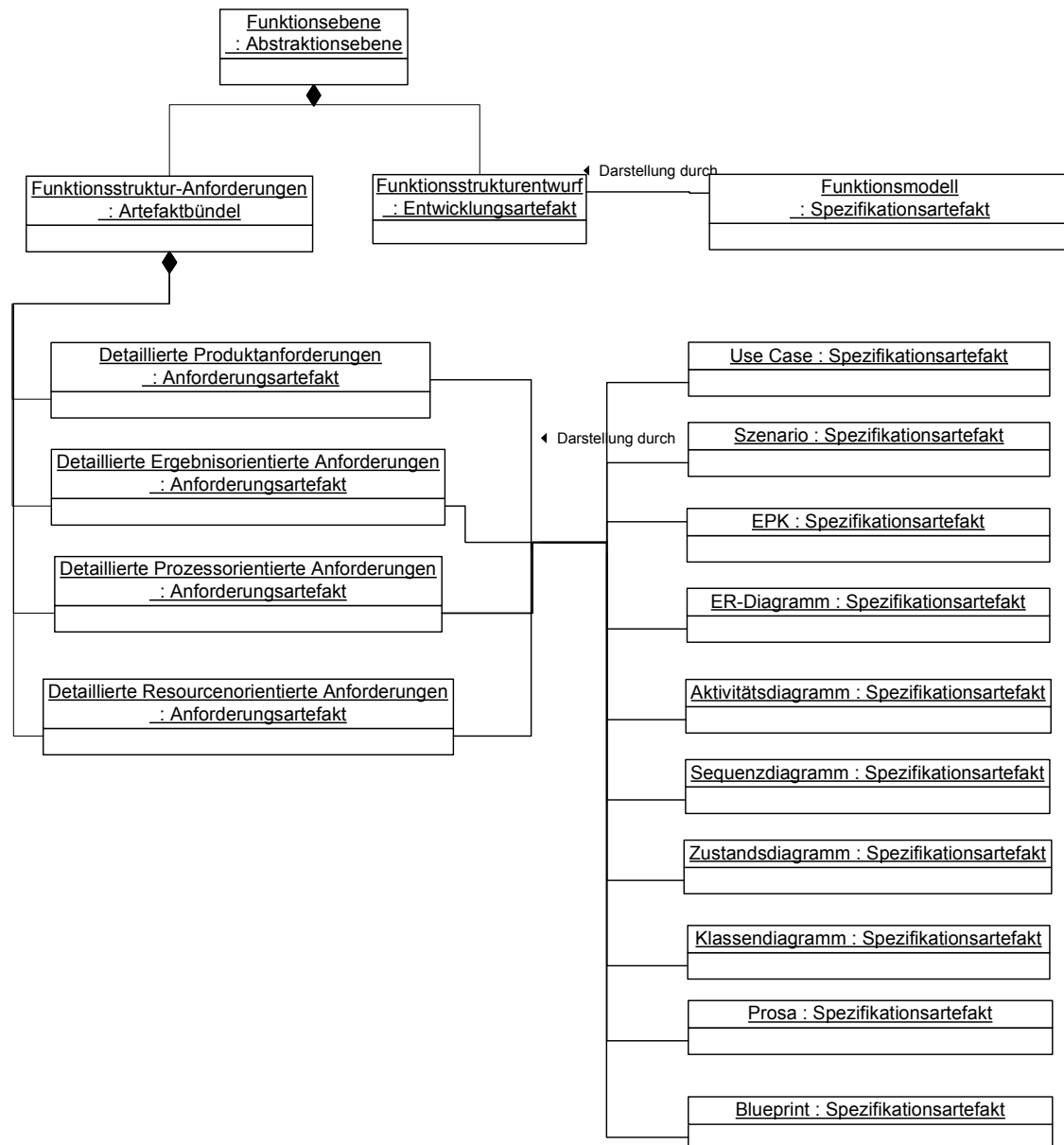


Abbildung 6-13: Artefakte der Funktionsebene

Quelle: Eigene Darstellung

Auf der Funktionsebene werden ein Entwicklungsartefakt „*Funktionsstrukturentwurf*“ und vier Anforderungsartefakte „*Detaillierte Produktanforderungen*“, „*Detaillierte Prozessorientierte Anforderungen*“, „*Detaillierte Ergebnisorientierte Anforderungen*“ und „*Detaillierte Ressourcenorientierte Anforderungen*“ unterschieden. Die Anforderungsartefakte sind im Artefaktbündel „*Funktionsstruktur-Anforderungen*“ zusammengefasst. Der Funktionsstrukturentwurf wird durch das Spezifikationsartefakt „*Funktionsmodell*“ und die Anforderungs-

tefakte durch die Spezifikationsartefakte „Use Case“, „Szenario“, „EPK“, „ER-Diagramm“, „Aktivitätsdiagramm“, „Sequenzdiagramm“, „Zustandsdiagramm“, „Klassendiagramm“, „Prosa“ und „Blueprint“ dokumentiert.

Die *Funktionsebene* strukturiert das technische Produkt und die Dienstleistungen, die zusammen mit dem technischen Produkt in Form einer Lösung angeboten werden, in Funktionen und bündelt sie in Funktionsstrukturen. Der Funktionsstrukturentwurf fasst die Funktionsstrukturen zusammen. Die Strukturierung des technischen Produktes und der Dienstleistungen erfolgt unter Zuhilfenahme der Design-Anforderungen der Eigenschaftsebene. Dabei beschreibt die Funktionsebene die Konkretisierung von Design-Anforderungen der Eigenschaftsebene für die einzelnen Funktionen des Funktionsstrukturentwurfs. Somit beschreiben die Funktionsstruktur-Anforderungen die Funktionalitäten der domänenspezifischen Komponenten, wie Hardware, Software und Dienstleistungen, und legen dadurch die Grundlage zur vollständigen Identifikation dieser Komponenten auf der Komponentenebene.

Jede Funktionsstruktur-Anforderung der Funktionsebene ist direkt auf eine oder mehrere Design-Anforderungen der Eigenschaftsebene zurückzuführen. Das heißt, dass eine Design-Anforderung der Eigenschaftsebene durch mehrere Funktionsstruktur-Anforderungen der Funktionsebene konkretisiert wird und eine Funktionsstruktur-Anforderung der Funktionsebene mehrere Design-Anforderungen der Eigenschaftsebene konkretisiert. Zwischen Design-Anforderungen und Funktionsstruktur-Anforderungen besteht eine n:m-Beziehung. Wie im Falle der Anforderungen der Zielebene, der Systemebene und der Eigenschaftsebene beeinflussen sich die Anforderungen gegenseitig. Das soll bei der Konkretisierung der Funktionsstruktur-Anforderungen berücksichtigt werden, indem die schon ermittelten Anforderungen der Funktionsebene zur Ermittlung weiterer Anforderungen der Funktionsebene einbezogen werden sollen (s. Kapitel 6.1.3.5).

6.3.4.1 Entwicklungsartefakte der Funktionsebene

Der *Funktionsstrukturentwurf* beinhaltet die Funktionen, die das technische Produkt und die Dienstleistungen nicht mehr aus der Black-Box-Sicht betrachten, sondern auf die Funktionalitäten ihrer Komponenten eingehen. Dadurch soll ermöglicht werden, basierend auf diesen Funktionalitäten die eigentlichen Ausgestaltungen der Komponenten zu definieren.

Die Funktionen des Funktionsstrukturentwurfs werden aus den Hauptfunktionen der Eigenschaftsebene unter Einbeziehung der Design-Anforderungen durch eine iterative Zerlegung und die damit verbundene Detaillierung abgeleitet. Die iterative Zerlegung der Funktionen findet solange statt, bis jeder Funktion eine eindeutige Hardwarekomponente, unter welcher auch ein mechatronisches System zu verstehen ist, oder eine Softwarekomponente oder eine Dienstleistung, deren Ergebnis-, Prozess- und Ressourcendimensionen vollständig ermittelt sind, zugeordnet werden kann. Das detaillierte Verfahren zur Funktionszerlegung wird in Kapitel 7.3 im Rahmen des Vorgehensmodells für das Requirements Engineering erläutert.

Durch die Zerlegung der Hauptfunktionen des Systementwurfs, die das technische Produkt und die Dienstleistungen beschreiben, in weitere Teilfunktionen entstehen Funktionsstrukturen, die in ihrer Gesamtheit den Funktionsstrukturentwurf bilden. Die Funktionen des Funktionsstrukturentwurfs beschreiben die Funktionalität der Software- und Hardwarekomponenten, während die Funktionen für die Dienstleistungen auf die Gestaltung des Dienstleistungserbringungsprozesses und der dafür benötigten Ressourcen eingehen. Die Funktionen sind hierarchisch gegliedert. Sie sind durch Kommunikationspfade miteinander verbunden, die kausa-

le und flussorientierte Beziehungen zwischen den Funktionen darstellen. Ähnlich wie bei den im Systementwurf identifizierten Hauptfunktionen (s. Kapitel 6.3.3.4) werden für die Funktionen im Funktionsstrukturentwurf die relationsorientierte und die umsatzorientierte Funktionsmodellierung verwendet. Dabei werden kausale Beziehungen zwischen den Funktionen verwendet, um die Abhängigkeiten zwischen den Funktionen darzustellen, die Systemzusammenhänge zu identifizieren und dadurch aus einer technischen Funktionsstruktur eine Problemstellung zur weiteren Bearbeitung zu erstellen (Ponn/Lindemann 2008, 67; Engeln 2006, 74ff.). Flussorientierte Beziehungen werden in der umsatzorientierten Funktionsmodellierung verwendet, um die Übergänge zwischen den verschiedenen Zuständen eines Produktes zu zeigen (Lindemann 2006, 116-118; Schlink 2004, 62-67). Weitere Details zu Funktionen und Funktionsstrukturen sind in Kapitel 6.3.3.1 zu finden.

6.3.4.2 Anforderungsartefakte der Funktionsebene

Die Design-Anforderungen der Eigenschaftsebene werden zur Zerlegung der Hauptfunktionen des Systementwurfs verwendet, indem sie detaillierte Informationen über das technische Produkt und die Dienstleistungen, wie Funktionalitäten oder die Qualität des technischen Produktes oder der Dienstleistungen, einbringen. Diese Informationen werden verwendet, um die Hauptfunktionen in Teilfunktionen, die die von den Komponenten des PSS, wie Hardware, Software oder Dienstleistungen, zu erfüllenden Teilaufgaben beschreiben, zu zerlegen. Somit beschreiben die durch die Zerlegung gewonnenen Funktionen die Funktionalitäten der domänenspezifischen Komponenten, ohne auf die Ausgestaltung und die Bestandteile der Komponenten einzugehen. Durch die Zerlegung der Hauptfunktionen werden die Design-Anforderungen der Eigenschaftsebene konkretisiert und sie erhalten weitere Details bzgl. der Ausgestaltung und Funktionalitäten der einzelnen domänenspezifischen Komponenten. Daraus ergeben sich detaillierte Anforderungen an die Komponenten des PSS.

Die detaillierten Anforderungen an das technische Produkt, die die Funktionalitäten der Hardware- und Softwarekomponenten beschreiben, werden im Anforderungsartefakt „*Detaillierte Produktanforderungen*“ zusammengefasst. Die detaillierten Anforderungen an die Dienstleistungen, die die Funktionalitäten der Dienstleistungen beschreiben, werden in den Anforderungsartefakten „*Detaillierte prozessorientierte Anforderungen*“, „*Detaillierte ergebnisorientierte Anforderungen*“ und „*Detaillierte ressourcenorientierte Anforderungen*“ zusammengefasst.

Ein Beispiel für eine Hauptfunktion ist „Terminverwaltung durch das IT-System“ (in Anlehnung an das Beispiel „*Personal Health Manager als IT-gestütztes, gerätebasiertes und personalisiertes Gesundheitsförderungsprogramm*“, Kapitel 3.2.2). Die Produktanforderung, die sich aus der Hauptfunktion ergibt, lautet: „Ein zentraler Kalender wird verwendet, um alle Termine der Teilnehmer und Trainer zu managen“. Durch die Zerlegung der Hauptfunktion ergeben sich zwei Teilfunktionen: „Vereinbarung der Termine“ und „Überblick über die Termine eines Monats geben“. Daraus ergeben sich eine detaillierte Produktanforderung „Es besteht die Möglichkeit, Termine bezüglich des Trainingsplans zu vereinbaren, bei welchen sowohl der Trainer als auch der Teilnehmer anwesend sind“ und eine prozessorientierte Anforderung an die Dienstleistung „Dem Teilnehmer wird ein Überblick über alle Termine eines Monats gegeben“.

Für die detaillierten Dienstleistungsanforderungen bleiben die Taxonomien, die im Rahmen der Eigenschaftsebene definiert wurden (s. Kapitel 6.3.3.2), bestehen. Bei der Konkretisierung

werden die Anforderungen bzgl. dieser Kategorien der Taxonomien unter Zuhilfenahme von Funktionen der Funktionsebene, die die konkreten Funktionalitäten der Dienstleistungen beschreiben, detailliert.

Für die detaillierten Produkthanforderungen wird die bestehende Taxonomie aus der Eigenschaftsebene um eine weitere Kategorie erweitert. Abbildung 6-14 zeigt die Taxonomie für die detaillierten Produkthanforderungen.

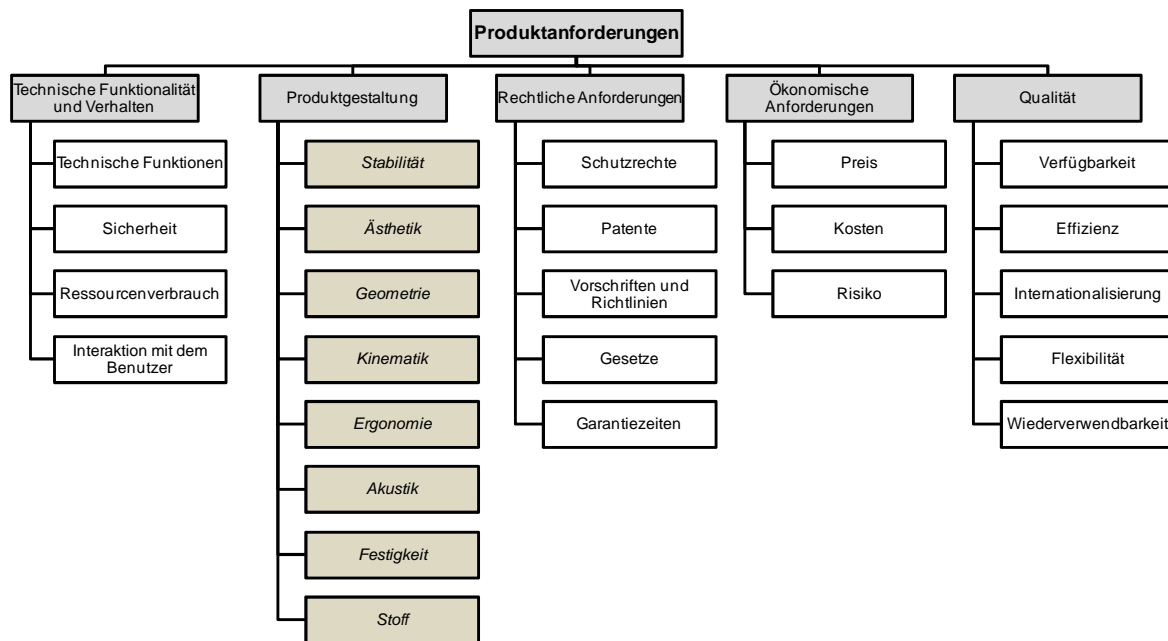


Abbildung 6-14: Taxonomie der detaillierten produktorientierten Anforderungen der Funktionsebene

Quelle: Eigene Darstellung

Eine neue Anforderungskategorie in der Taxonomie ist „*Produktgestaltung*“ (in der Abbildung ist diese Kategorie kursiv). Die Anforderungen, die dieser Kategorie zugeordnet sind, beschreiben die physikalischen und haptischen Eigenschaften des technischen Produktes unter Berücksichtigung der Funktionen, die die Funktionalität der Komponenten des technischen Produktes definieren.

Die Anforderungen an die *Stabilität* definieren, zu welchem Grad das technische Produkt und somit die Hardware- und Softwarekomponenten des PSS frei von Schwankungen sein soll. Ein technisches Produkt ist von verschiedenen Rückkopplungen, wie bspw. Beschallungssysteme, beeinflusst. Die Anforderungen an die *Stabilität* beschreiben, wie ein technisches Produkt auf eine beschränkte Erregung, die durch die Rückkopplungen ausgelöst ist, reagieren soll (Lunze 2003, 161f.). Die Anforderungen an die *Ästhetik* definieren die äußeren haptischen Merkmale des technischen Produktes, die für die Wahrnehmung des technischen Produktes durch den Kunden wichtig sind. So kann sich ein Kunde bspw. eine Waschmaschine wünschen, die farblich zur Umgebung des Badezimmers passt. Die Anforderungen an die *Geometrie* definieren die Größe, Höhe, Länge, Durchmesser, Raumbedarf, Anordnung, Anschluss, Ausbau und Erweiterung des technischen Produktes. Die Anforderungen an die *Kinematik* beschreiben die Bewegungsart, die Bewegungsrichtung, die Geschwindigkeit und die Beschleunigung des technischen Produktes oder seiner Teile. Die Anforderungen an die *Ergonomie* definieren die Mensch-Maschine-Interaktion, wie Bedienung, Bedienungsart, Über-

sichtigkeit, Beleuchtung und Formgestaltung (vgl. (Ehrlenspiel 2002, 375; Pahl et al. 2006, 51; Dul/Weerdmeester 2008, 15ff.; Dix et al. 2003, 123ff.)). Die Anforderungen an die *Akustik* beschreiben den Geräuschpegel, die Lautstärke und den Schalldruckverlauf (Maute 2006, 24). Die Anforderungen an die *Festigkeit* geben den mechanischen Widerstand eines Werkstoffes an. Die Anforderungen an den *Stoff* beschreiben physikalische, chemische, biologische Eigenschaften des Eingangs- und Ausgangsproduktes, vorgeschlagene Werkstoffe, Materialfluss und Materialtransport (Ehrlenspiel 2002, 375).

6.3.4.3 Externe Artefakte der Funktionsebene

Als externes Artefakt dient auf der Funktionsebene ähnlich wie auf der Eigenschaftsebene (s. Kapitel 6.3.3.3) ein *Katalog zur Systematisierung von Dienstleistungen* (Tabelle 10-5 im Anhang), um einen Überblick über die Art der Dienstleistungen zu geben und dadurch die passenden Dienstleistungen und ihre Ausgestaltung zu wählen. Das weitere externe Artefakt stellen die *Konstruktionskataloge* dar, die zur Suche nach den Funktionalitäten und Gestaltungseigenschaften der Komponenten des technischen Produktes verwendet werden können (s. Kapitel 6.3.3.3).

6.3.4.4 Spezifikationsartefakte der Funktionsebene

Zur Spezifikation der Funktionsstrukturen werden *Funktionsmodelle* (s. Kapitel 6.3.3.4) eingesetzt. Ähnlich wie bei den Funktionsmodellen der Eigenschaftsebene werden die Funktionsstrukturen als Netzwerke von Eingangs- und Ausgangsbeziehungen dargestellt.

Zur Dokumentation von Anforderungen der Funktionsebene werden dieselben Spezifikationsartefakte wie auf der Eigenschaftsebene (s. Kapitel 6.3.3.4) eingesetzt: „*Use Cases*“, „*Szenario*“, „*EPK*“, „*ER-Diagramm*“, „*Aktivitätsdiagramm*“, „*Sequenzdiagramm*“, „*Zustandsdiagramm*“, „*Klassendiagramm*“, „*Prosa*“ und *Blueprints* gewählt. Diese Spezifikationsartefakte werden hier zur Darstellung der Anforderungen der Funktionsebene verwendet. Mit *Blueprints* werden die Dienstleistungen dokumentiert. Die Dokumentation der Funktionsstruktur-Anforderungen und der jeweiligen Attribute dieser Anforderungen in *Prosa* ermöglichen den Status der Anforderungen nachzuvollziehen. Für die Dokumentation in *Prosa* eignet sich die Struktur des Lastenheftes, die an die Vorgaben der Funktionsebene angepasst wird. Durch die Angabe von Verweisen auf *Design-Anforderungen* der Eigenschaftsebene, aus denen die Funktionsstruktur-Anforderungen abgeleitet wurden, und auf *Domänen-Anforderungen* der Komponentenebene, die aus den Funktionsstruktur-Anforderungen abgeleitet werden, wird die *Pre-Traceability*, die den Ursprung einer Anforderung umfasst, und die *Post-Traceability*, die die nachgelagerten Artefakte umfasst, gewährleistet.

6.3.5 Abstraktionsebene 5: Komponentenebene

Das Ziel der Komponentenebene ist es, basierend auf den Funktionsstruktur-Anforderungen, die die Funktionalitäten der Komponenten des PSS beschreiben, und den Funktionen des Funktionsstrukturentwurfs, das PSS in domänenspezifischen Komponenten zu strukturieren und in einem Grobentwurf abzubilden. Desweiteren beschreibt die Komponentenebene die Konkretisierung von Funktionsstruktur-Anforderungen für die einzelnen Komponenten des PSS. Die Komponentenebene stimmt mit den Vorgaben des Entwicklungsprozesses von PSS zur Prinzipmodellierung überein (s. Kapitel 3.3.1).

Auf der Komponentenebene werden ein Entwicklungsartefakt „*Grobentwurf*“ und drei Anforderungsartefakte „*Anforderungen an die Produktentwicklung*“, „*Anforderungen an die*

Softwareentwicklung“ und „Anforderungen an die Dienstleistungsentwicklung“ unterscheiden. Die Anforderungsartefakte sind in einem Anforderungsartefaktbündel „Domänen-Anforderungen“ zusammengefasst. Der Grobentwurf wird durch das Spezifikationsartefakt „Komponentendiagramm“ und die Anforderungsartefakte werden durch die Spezifikationsartefakte „Use Cases“, „Szenario“, „EPK“, „ER-Diagramm“, „Aktivitätsdiagramm“, „Sequenzdiagramm“, „Zustandsdiagramm“, „Klassendiagramm“, „Prosa“ und „Blueprint“ dokumentiert. Abbildung 6-15 zeigt die Artefakte der Komponentenebene in Form eines Objektdiagramms dar.

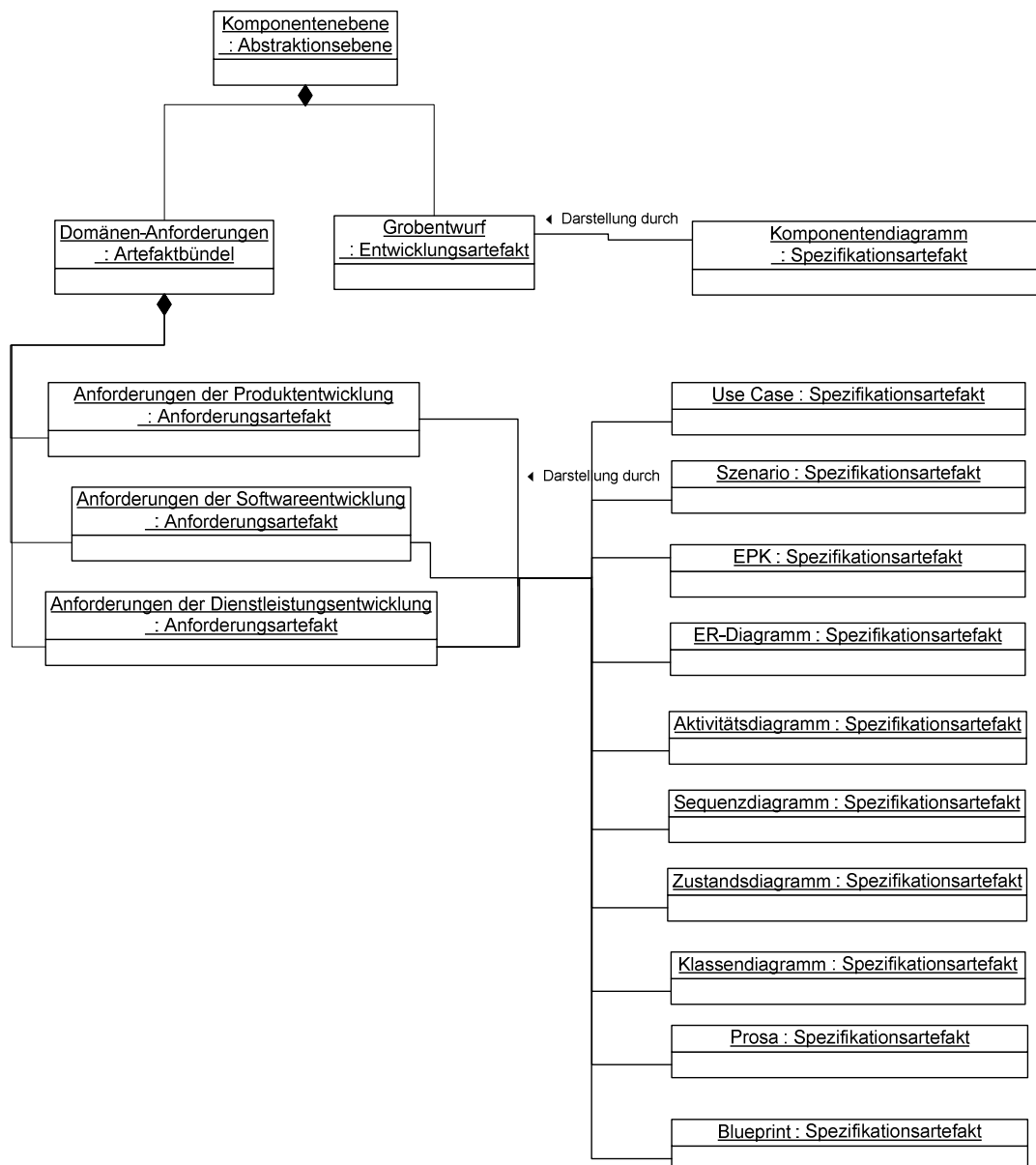


Abbildung 6-15: Artefakte der Komponentenebene

Quelle: Eigene Darstellung

Jede Domänen-Anforderung der Komponentenebene ist direkt auf eine oder mehrere Funktionsstruktur-Anforderungen der Funktionsebene zurückzuführen. Das heißt, dass eine Funktionsstruktur-Anforderung der Funktionsebene durch mehrere Domänen-Anforderungen der Komponentenebene konkretisiert wird und eine Domänen-Anforderung der Komponentenebene mehrere Funktionsstruktur-Anforderungen der Funktionsebene konkretisiert. Zwischen

Domänen-Anforderungen und Funktionsstruktur-Anforderungen besteht eine n:m-Beziehung. Wie im Falle der Anforderungen der Zielebene, der Systemebene, der Eigenschaftsebene und der Funktionsebene beeinflussen sich die Anforderungen gegenseitig. Das soll bei der Konkretisierung der Domänen-Anforderungen berücksichtigt werden, indem die schon ermittelten Anforderungen der Komponentenebene zur Ermittlung weiterer Anforderungen der Komponentenebene einbezogen werden sollen (s. Kapitel 6.1.3.5).

6.3.5.1 Entwicklungsartefakte der Komponentenebene

Der *Grobentwurf* stellt die Aufteilung des PSS in Hardware-, Software- und Dienstleistungskomponenten dar. Der Grobentwurf detailliert den Systementwurf, der auf der Systemebene definiert wurde, indem der Funktionsstrukturentwurf mit den Funktionen und die lösungsorientierten Funktionsstruktur-Anforderungen der Funktionsebene berücksichtigt werden. Die domänenspezifischen Komponenten, die im Grobentwurf festgehalten sind, sind abstrakte, d.h. logische, Komponenten. Der Grobentwurf ist somit eine logische Architektur des PSS (Schätz 2006). Er definiert ebenso die Schnittstellen zwischen den logischen Komponenten, durch welche die Kommunikation zwischen den Komponenten verläuft, und die Eingangs- und Ausgangsdaten, die während der Kommunikation zwischen den Komponenten ausgetauscht werden.

Der Grobentwurf stellt eine Grundlage für die nächsten Entwicklungsschritte bereit, in denen die logischen Komponenten auf die physikalischen Komponenten abgebildet werden (Pohl 2007). Der Grobentwurf selbst definiert jedoch noch keine konkreten physikalischen Hardwarekomponenten (wie z.B. Aktuatoren), Softwarekomponenten (wie z.B. konkretes Echtzeitbetriebssystem), oder Dienstleistungen (wie z.B. den konkreten Ablauf der Wartung für eine Waschmaschine der Premiumvariante).

6.3.5.2 Anforderungsartefakte der Komponentenebene

Die Funktionsstruktur-Anforderungen der Funktionsebene werden bzgl. der Komponenten iterativ konkretisiert und auf die Domänen-Anforderungen der Komponentenebene übertragen. Die Domänen-Anforderungen beschreiben die Ausgestaltung und Bestandteile der domänenspezifischen Komponenten, die im Grobentwurf definiert sind. Somit beschreiben die Domänen-Anforderungen die physikalische Struktur des PSS.

Es werden drei Anforderungsartefakte unterschieden: „Anforderungen an die Produktentwicklung“, „Anforderungen an die Softwareentwicklung“ und „Anforderungen an die Dienstleistungsentwicklung“. Diese Anforderungen beschreiben die einzelnen Komponenten des PSS. Die Anforderungen an die Produktentwicklung beschreiben die physikalische Struktur der Hardwarekomponenten, wie bspw. Anforderungen an die Aktuatoren. Die Anforderungen an die Softwareentwicklung beschreiben die technische Struktur der Softwarekomponenten, wie bspw. die Anforderungen an das Betriebssystem. Die Anforderungen an die Dienstleistungsentwicklung beschreiben die Struktur der Dienstleistungen, wie bspw. die Anforderungen an den konkreten Ablauf der Wartung für eine Waschmaschine der Premiumvariante. Dabei können die Anforderungen an die Softwarekomponente weiterhin nach funktionalen und nicht-funktionalen Anforderungen aufgeteilt werden (Sommerville 2004, 121-122; Berenbach et al. 2009, 22). Abbildung 10-1 (Anhang) gibt einen Überblick über die Kategorisierung von nicht-funktionalen Anforderungen an die Software-Komponenten. Die Anforderungen an die Hardware-Komponenten können basierend auf der Hauptmerkmalliste von (Pahl et al. 2006, 225) charakterisiert werden (Tabelle 10-6 im Anhang).

6.3.5.3 Externe Artefakte der Komponentenebene

Zu den externen Artefakten gehören Konstruktionskataloge zur Gestaltung der Komponenten des technischen Produktes des PSS im Rahmen des Grobentwurfs (s. Kapitel 6.3.3.3). Ebenso wird ein *Katalog zur Systematisierung von Dienstleistungen* (Tabelle 10-5 im Anhang) zur Gestaltung der Dienstleistungskomponenten des PSS eingesetzt (Kapitel 6.3.3.3).

6.3.5.4 Spezifikationsartefakte der Komponentenebene

Zur Spezifikation des Grobentwurfs werden *Komponentendiagramme* eingesetzt. Ein Komponentendiagramm zeigt die internen Elemente eines Systems und die Beziehungen zwischen denen (Booch et al. 2007, 126). Das Komponentendiagramm zeigt die Struktur der Komponenten des PSS und die Struktur der Abhängigkeitsbeziehungen zwischen den Komponenten.

Zur Dokumentation von Anforderungen der Komponentenebene werden dieselben Spezifikationsartefakte wie auf der Eigenschaftsebene verwendet: (s. Kapitel 6.3.3.4) „Use Cases“, „Szenario“, „EPK“, „ER-Diagramm“, „Aktivitätsdiagramm“, „Sequenzdiagramm“, „Zustandsdiagramm“, „Klassendiagramm“, „Prosa“ und *Blueprints*. Im Gegensatz zu den Spezifikationsartefakten der System-, Eigenschafts- und Funktionsebene, beziehen sich diese Spezifikationsartefakte auf die Anforderungen der Komponentenebene. Mit *Blueprints* werden die Dienstleistungen dokumentiert. Als Ergebnis der Spezifikation von Anforderungen liegt das Pflichtenheft vor, das die analysierten und konkretisierten Anforderungen an die domänenspezifischen Komponenten des PSS detailliert beschreibt. Das Pflichtenheft enthält somit die Realisierungsvorgaben an die Komponenten und definiert, wie und womit die Anforderungen des Lastenheftes umgesetzt werden sollen (vgl. (Ponn/Lindemann 2008, 35; Balzert 1998, 447-448; Krcmar 2010, 167)). Die Dokumentation der Domänen-Anforderungen und der jeweiligen Attribute dieser Anforderungen in Prosa ermöglichen den Evolutionsstand der Anforderungen zu jedem Zeitpunkt der Entwicklung nachzuvollziehen. Durch die Angabe von Verweisen auf Funktionsstruktur-Anforderungen, aus denen die Domänen-Anforderungen abgeleitet wurden, wird die Pre-Traceability, die den Ursprung einer Anforderung umfasst, gewährleistet.

6.4 Zusammenfassung

In diesem Kapitel wurde das Artefaktmodell für die Anforderungen an IT-gestützte Product Service Systems vorgestellt. Das Artefaktmodell basiert auf den im Kapitel 4.7 erarbeiteten Anforderungen an einen Ansatz zum Requirements Engineering für PSS. Das Artefaktmodell stellt die Grundlage dar, um die schrittweise Konkretisierung der Anforderungen in Übereinstimmung mit dem Entwicklungsprozess zu gewährleisten. Zudem unterstützt die Strukturierung der Anforderungen durch das Artefaktmodell den systematischen Umgang mit den Zyklen die in der Entwicklung von PSS auftreten.

In diesem Kapitel wurden als Erstes begriffliche Grundlagen, wie die Definition des Begriffes „Artefakt“ im Requirements Engineering eingeführt. Anschließend wurden existierende Ansätze aus der Literatur in Bezug zum entwickelten Artefaktmodell gestellt und dessen Abgrenzung diskutiert. Darauf folgend wurden die wesentlichen Prinzipien, die im Artefaktmodell umgesetzt werden erläutert. Dazu gehören die Definition von Artefakten und Beziehungen zwischen ihnen, die Definition von Abstraktionsebenen, die zur Strukturierung der Artefakte genutzt werden, sowie die Integration von Entwicklungsinformationen in die Anforderungsstruktur. Ein weiteres wesentliches Konzept stellt die Trennung der Beschreibung von

Inhalten (in Modellartefakten) und Darstellungsformen (in Spezifikationsartefakten) dar. Im darauf folgenden Unterkapitel wird ein Meta-Modell des Artefaktmodells eingeführt. Dieses definiert die Modellelemente aus denen das Artefaktmodell besteht und ermöglicht dadurch eine klare und einheitliche Definition des Artefaktmodells.

Darauffolgend werden alle Artefakte und ihre Beziehungen im Detail erklärt. Dazu wird für jede Abstraktionsebene ein Unterkapitel eingeführt, in denen die Modellartefakte, Taxonomien und Spezifikationsartefakte erläutert werden.

In der ersten Abstraktionsebene, der Zielebene, werden die allgemeinen Ziele bzgl. des zu entwickelnden PSS beschreiben. Diese Abstraktionsebene liegt noch außerhalb des eigentlichen Entwicklungsprozesses von PSS. Die darauffolgende Systemebene umfasst die initialen Anforderungen an das PSS (PSS-Anforderungen). Anschließend wird in der Eigenschaftsebene der Systementwurf definiert, der den Systemkontext und die Aufteilung des PSS in das technische Produkt und die Dienstleistungen beschreibt. Die Eigenschaftsebene umfasst die Anforderungen, die die Sach- und Dienstleistungskomponenten beschreiben. In der darauffolgenden Funktionsebene werden die Funktionsstrukturen für Sach- und Dienstleistungen des PSS sowie die bzgl. der Funktionen konkretisierten Anforderungen an die Sach- und Dienstleistungskomponenten, zu denen Hardware, Software und Dienstleistungen gehören, beschrieben. Die Komponentenebene beschreibt die physikalische Struktur des PSS und die Anforderungen an die einzelnen physikalischen Komponenten des PSS, die an die Domänen zur weiteren Entwicklung übergeben werden können.

7 Vorgehensmodell für das Requirements Engineering für Product Service Systems

In Kapitel 7 wird das Vorgehensmodell für das Requirements Engineering für PSS vorgestellt. In Kapitel 7.1 werden die für das Verständnis der Konzeption des Vorgehensmodells relevanten Elemente eingeführt. In Kapitel 7.2 wird das Metamodell des Vorgehensmodells, das die Bestandteile des Vorgehensmodells definiert, erläutert. Anschließend wird in Kapitel 7.3 das eigentliche Vorgehensmodell für das Requirements Engineering für PSS vorgestellt.

7.1 Begriffliche Grundlagen

Um den Begriff „Vorgehensmodell“ eindeutig zu definieren, ist es wichtig ihn von den Begriffen „Methode“, „Technik“, „Werkzeug“ und „Vorgehen“ abzugrenzen.

Der Methodenbegriff wird oft im unterschiedlichen Kontext verwendet. Eine Methode ist „eine planmäßige Art und Weise des Handelns mit überprüfbaren Ergebnissen“ (Greiffenberg 2004). Eine Methode ist ein Ansatz, der die Vorgaben und Regeln zur Durchführung eines Entwicklungsprojektes in Form von Aktivitäten und Entwicklungsergebnissen definiert (Brinkkemper 1996). In der Software-Technik versteht man unter einer Methode einen Oberbegriff von Konzepten, Notationen und methodischer Vorgehensweise.

Eng verbunden mit dem Begriff „Methode“ ist auch der Begriff „Technik“. „A technique prescribes how to perform one particular activity - and, if necessary, how to describe the product of that activity in a particular notation. A method provides a prescription for how to perform a collection of activities, focusing on how a related set of techniques can be integrated, and providing guidance on their use“ (Nuseibeh/Easterbrook 2000). Eine Technik ist eine Prozedur, die über eine definierte Notation verfügt, um eine Entwicklungsaktivität durchzuführen (Brinkkemper 1996). Zu den wichtigsten Bestandteilen einer Technik gehört das Konzept, das ihr zu Grunde liegt, die Notation, die Schritte, die ausgeführt werden und die Anleitung, wie die Technik anzuwenden ist (Nuseibeh et al. 1996).

Ein Vorgehensmodell definiert „die zeitlich logische Folge von Schritten in einem Problemlösungsprozess“ (Brugger 2005, 155). Das Vorgehensmodell definiert den Input, der zur Durchführung einer Aktivität notwendig ist, den Output, der das Ergebnis einer Aktivität darstellt, und die Aktivität selbst (Versteegen 2002, 29; Bunse/von Knethen 2002, 3ff.). Ebenso stellen Vorgehensmodelle Techniken bereit, die bei der Durchführung der Aktivitäten eingesetzt werden (Brandt-Pook/Kollmeier 2008, 3ff.).

Dem Vorgehensmodell für das Requirements Engineering für PSS liegt die Definition von Aktivitäten und Techniken zu Grunde.

7.2 Metamodells des Vorgehensmodells

Während das Artefaktmodell ein Datenmodell für die Anforderungen an das PSS darstellt, soll das Vorgehensmodell für das Requirements Engineering für PSS die Aktivitäten und ihre Reihenfolge zur Erstellung und Verwaltung von Anforderungen an PSS bereitstellen. Aus den in Kapitel 7.1 angeführten Definitionen wird ebenso ersichtlich, dass ein Vorgehensmodell für das Requirements Engineering für PSS zusätzlich zu den Aktivitäten auch die Techniken anbieten soll, um die Aktivitäten durchzuführen. Abbildung 7-1 zeigt das Metamodell des Vorgehensmodells für das Requirements Engineering für PSS.

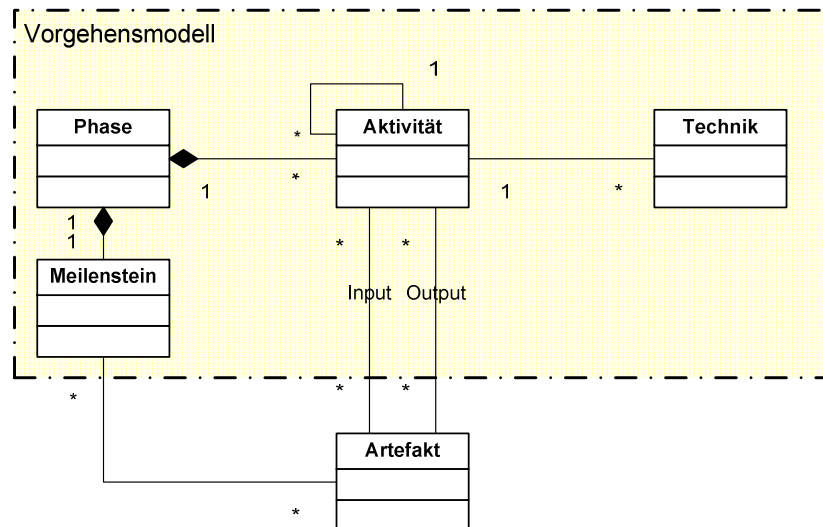


Abbildung 7-1: Metamodell des Vorgehensmodell für das Requirements Engineering für PSS

Quelle: Eigene Darstellung (In Anlehnung an (Méndez Fernández et al. 2010))

Eine Aktivität definiert, auf welche Weise und durch welche Arbeitsschritte ein Ergebnis erstellt wird (Höhn/Höppner 2008, 27). Dem Vorgehensmodell für das Requirements Engineering liegen Aktivitäten zu Grunde, die die Artefakte erstellen, verwalten und modifizieren. Zur Durchführung einer Aktivität werden Eingangsinformationen (Input) benötigt. Eine solche Eingangsinformation ist immer ein Artefakt des Artefaktmodells. Das Ergebnis einer Aktivität des Requirements Engineering (Output) sind Anforderungs- oder Entwicklungsartefakte, die durch Spezifikationsartefakte dokumentiert werden. Aktivitäten können in Teilaktivitäten zerlegt werden, wodurch eine Hierarchie von Aktivitäten entsteht. Auf diese Weise können die Aktivitäten im Vorgehensmodell gegliedert werden.

Die Anwendung richtiger Techniken ist die Voraussetzung für den Erfolg eines Projektes (Jiang et al. 2005). Zur Durchführung einer Aktivität werden eine oder mehrere Techniken benötigt. Das Vorgehensmodell gibt Techniken für die entsprechenden Aktivitäten an. Bei der Auswahl der Techniken wurde insbesondere auf die Eignung der Techniken für die Anforderungen an PSS geachtet (Berkovich et al. 2011c).

Die Artefakte des Artefaktmodells werden in Anlehnung an die Phasen der Entwicklung erstellt. Die Aktivitäten des Vorgehensmodells werden während dieser Phasen durchgeführt. Der Zeitpunkt der Fertigstellung eines Artefakts wird durch Ereignisse, die als Meilensteine bezeichnet werden, angegeben. Ein Meilenstein wird also dadurch definiert, dass bestimmte Artefakte fertiggestellt worden sind (Méndez Fernández et al. 2010).

7.3 Vorgehensmodell für das Requirements Engineering für Product Service Systems

Das Vorgehensmodell für das Requirements Engineering für PSS ist an die Abstraktionsebenen des Artefaktmodells und demzufolge an die Phasen der Entwicklung von PSS angelehnt. Um die Artefakte einer bestimmten Abstraktionsebene zu erstellen, werden folgende Aktivitäten des Vorgehensmodells definiert:

1. Ermittlung und Analyse von Zielen an PSS,
2. Ermittlung und Analyse von PSS-Anforderungen,
3. Konzeption eines Systementwurfs, Ermittlung und Analyse von Design-Anforderungen,
4. Konzeption eines Funktionsstrukturentwurfs, Ermittlung und Analyse von Funktionsstruktur-Anforderungen,
5. Konzeption eines Grobentwurfs, Ermittlung und Analyse von Domänen-Anforderungen.

Abbildung 7-2 stellt eine schematische Darstellung des Vorgehensmodells für das Requirements Engineering für PSS dar.

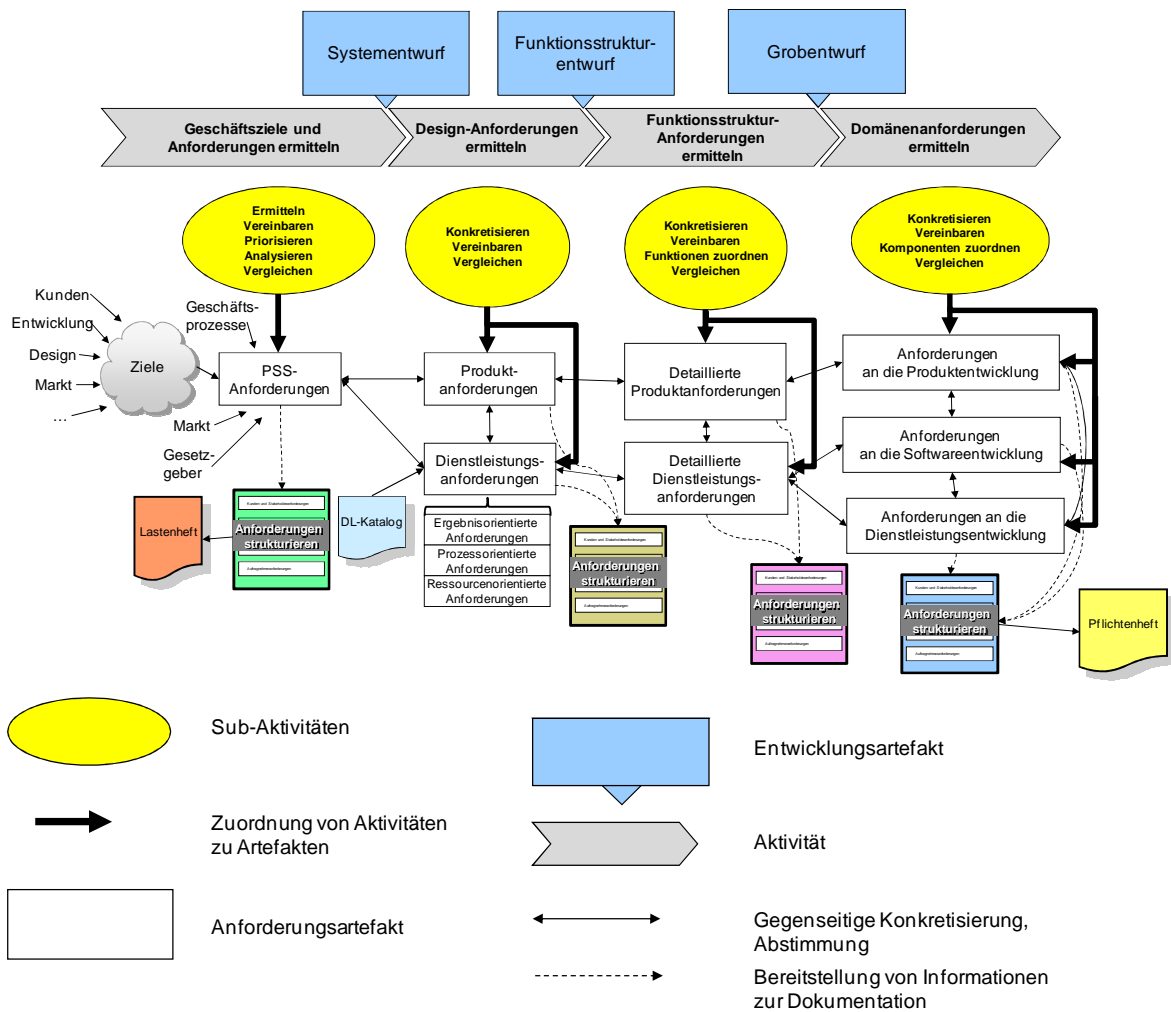


Abbildung 7-2: Schematische Darstellung des Vorgehensmodells für das Requirements Engineering für PSS

Quelle: Eigene Darstellung

Da die Entwicklungsartefakte nicht im Rahmen des Requirements Engineering erstellt werden, sondern zur Ermittlung und Konkretisierung von Anforderungen benötigt werden, sind sie oberhalb der eigentlichen Aktivitäten des Requirements Engineering platziert. Die Aktivitäten des Requirements Engineering (graue Pfeile) können in weitere Sub-Aktivitäten (gelbe Kreise) zerlegt werden.

Die Aktivitäten werden iterativ durchgeführt. Die Anforderungsartefakte, die gegenseitig konkretisiert oder abgestimmt werden müssen, sind durch den durchgezogenen Doppelpfeil miteinander verbunden. Die Produkthanforderungen (Eigenschaftsebene) leiten sich bspw. aus den PSS-Anforderungen (Systemebene) ab. Wenn durch die Konkretisierung neue oder zusätzliche Eigenschaften zu den Produkthanforderungen hinzugefügt werden, die eine Änderung oder Anpassung an den PSS-Anforderungen verlangen, sollen die davon betroffenen PSS-Anforderungen erneut analysiert und konkretisiert werden. Das kann wiederum zu einer erneuten Analyse und Konkretisierung von entsprechenden Produkthanforderungen führen. Die Aktivitäten sind solange durchzuführen, bis der Detaillierungsgrad aller Anforderungen dem von der Abstraktionsebene definierten Detaillierungsgrad entspricht und die Anforderungen konsistent und konfliktfrei sind. Abbildung 7-3 zeigt schematisch das Vorgehen bei der iterativen Analyse und Konkretisierung von zwei Anforderungen.

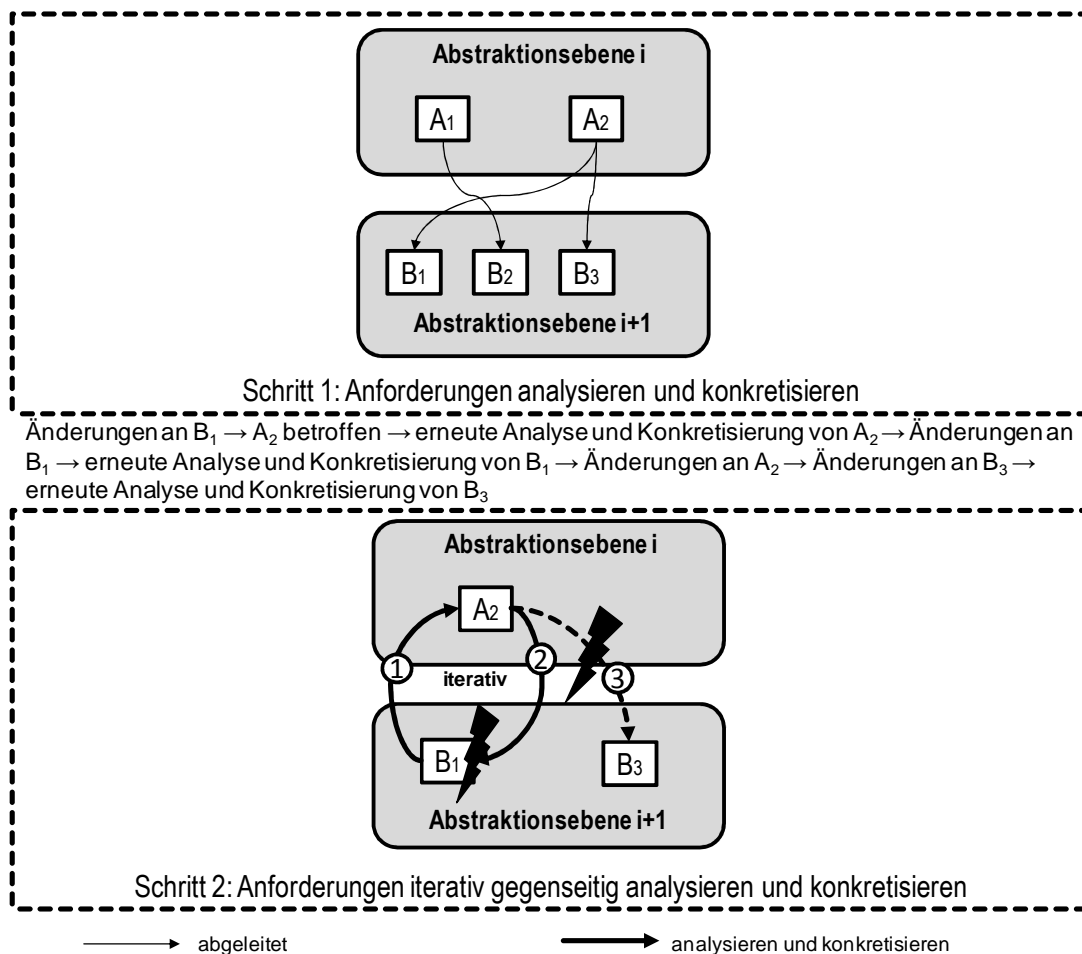


Abbildung 7-3: Iterative gegenseitige Analyse und Konkretisierung von Anforderungen

Quelle: Eigene Darstellung

Die Anforderung B_1 ist aus der Anforderung A_2 abgeleitet. Durch die Änderungen an B_1 ist auch A_2 betroffen und muss erneut analysiert und konkretisiert werden. Dadurch können sich aber Änderungen an B_1 ergeben, was zu einer erneuten Analyse und Konkretisierung von B_1 führt. Desweiteren ist die Anforderungen B_3 aus der Anforderungen A_2 abgeleitet. Die Änderungen an A_2 können somit zu Änderungen an B_3 führen. In diesem Fall muss B_3 erneut analysiert und konkretisiert werden. Falls B_3 aus weiteren Anforderungen abgeleitet ist, sind diese Anforderungen ebenfalls erneut zu analysieren.

Die im Rahmen des Vorgehensmodells angegebenen Techniken beziehen sich nur auf die Erstellung und Modifikation von Anforderungsartefakten. Die Techniken zur Erstellung und Modifikation der Entwicklungsartefakte sind kein Teil des Requirements Engineering Ansatzes und werden daher im Rahmen dieser Arbeit nicht betrachtet.

7.3.1 Ermittlung und Analyse von Zielen an PSS

Die Aktivitäten zur Ermittlung und Analyse von Zielen an PSS finden vor dem eigentlichen Entwicklungsprozess statt und sollen die Ziele der Kunden und des Auftragnehmers bzgl. des PSS ermitteln, analysieren und entsprechend den Spezifikationsvorschriften dokumentieren.

7.3.1.1 Aktivitäten zur Ermittlung und Analyse von Zielen an PSS

Ziele definieren die Intentionen des Kunden und des Auftragnehmers bzgl. der Nutzung und Bereitstellung des PSS (s. Kapitel 6.3.1). Sie stellen die Soll-Vorstellungen des Kunden und des Auftragnehmers dar, während die Anforderungen die Soll-Eigenschaften des PSS sind, d. h. die Anforderungen sind konkreter als Ziele und werden aus den Zielen abgeleitet. Bei der Ermittlung und Analyse von Zielen werden fünf wesentliche Schritte unterschieden (vgl. (Lindemann 2006, 63ff.; Husen 2007, 77-80; Berenbach et al. 2009, 49-51)):

1. Identifikation der relevanten Stakeholder, die von der Nutzung und Bereitstellung des PSS betroffen sind;
2. Durchführung einer neutralen und wertungsfreien Erhebung der Ist-Situation beim Kunden und dem Auftragnehmer,
3. Bewertung der Ist-Situation und Ableitung der Ziele seitens des Kunden und des Auftragnehmers (Geschäftsziele des Kunden und Geschäftsziele des Auftragnehmers),
4. Analyse von Zielen auf mögliche Konflikte, redundante Ziele und Auflösung von Konflikten, Elimination redundanter Ziele,
5. Validierung von Zielen,
6. Dokumentation von Zielen.

Abbildung 7-4 zeigt die Aktivitäten zur Ermittlung und Analyse von Zielen an das PSS

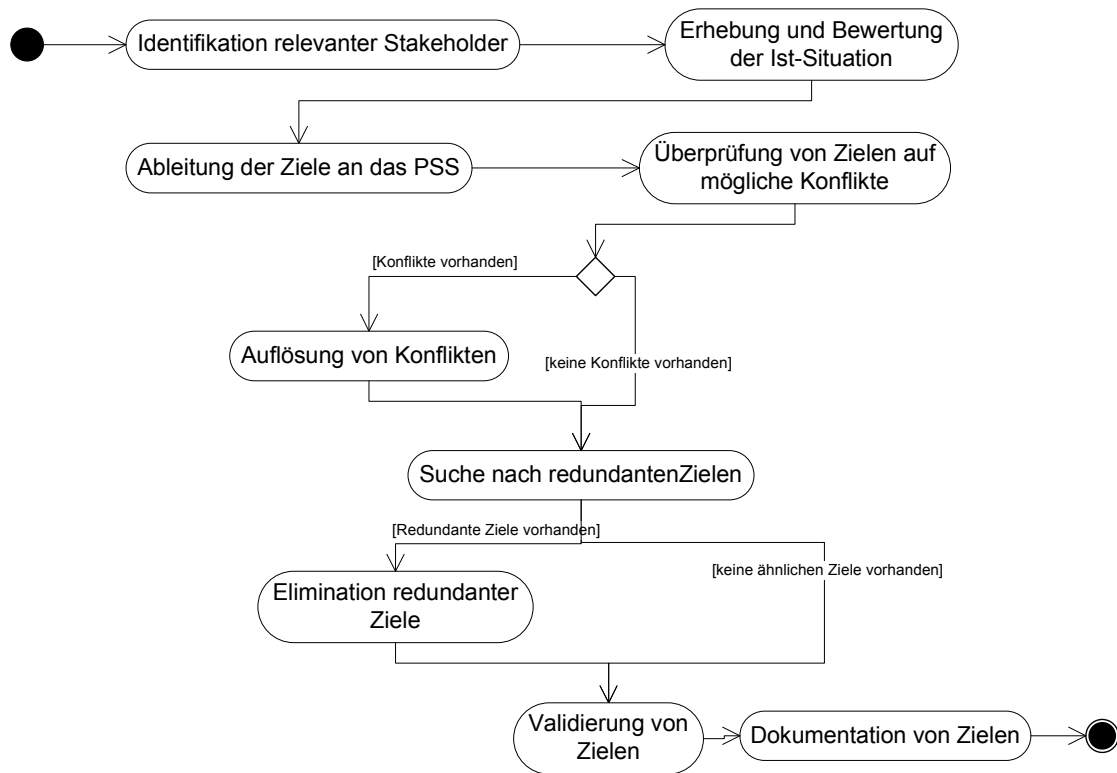


Abbildung 7-4: Aktivitäten der Ermittlung und Analyse von Zielen an das PSS

Quelle: Eigene Darstellung

Der Output der Durchführung der genannten Aktivitäten stellen die Anforderungsartefakte „Geschäftsziele des Kunden“ und „Geschäftsziele des Auftragnehmers“ dar. Der erste Schritt ist die Identifikation relevanter Stakeholder für das PSS. Zur Identifikation der relevanten Stakeholder können folgende Fragen als Anleitung dienen: „Wer und was sind von der Entwicklung des PSS betroffen?“, „Wer und was sind von der Nutzung des PSS betroffen?“, „Welche Rahmenbedingungen, wie Standards, Normen oder Gesetze, haben einen Einfluss auf die Bereitstellung und Nutzung des PSS?“. Tabelle 6-1 (s. Kapitel 6.3.1.2) stellt eine Liste an Faktoren dar, die die Ziele bei der Bereitstellung und Nutzung des PSS beeinflussen. Bei jedem identifizierten Stakeholder soll der Frage nachgegangen werden, in wieweit er Informationen zur Identifikation weiterer Stakeholder geben kann.

Anschließend soll die Ist-Situation beim Kunden und beim Auftragnehmer analysiert werden. Dabei soll festgestellt werden, in wieweit der Auftragnehmer in der Lage ist, das PSS gemäß den Zielen des Kunden bereitzustellen. Ebenso müssen die relevanten Geschäftsprozesse des Kunden analysiert werden, um die betrieblichen Vorgänge und die Zusammenhänge zwischen ihnen zu verstehen und, falls sich die Betriebsbedingungen ändern, die Änderungen am PSS richtig und effizient vorzunehmen (vgl. (Rupp 2004, 89)). Desweiteren sollen die technischen Altsysteme oder die schon vorhandenen Dienstleistungsprozesse des Kunden identifiziert werden, um sie wiederzuverwenden oder zu ersetzen. Die Ist-Analyse ermöglicht die bestehenden Probleme und Herausforderungen des Kunden, die mit der Einführung des PSS gelöst werden sollen, aufzudecken. Die so identifizierten Aspekte können verwendet werden, um bei der Bereitstellung des PSS explizit auf die Lösung der Probleme einzugehen und dadurch ein System zu entwickeln, das sich von der bestehenden Lösung beim Kunden positiv abhebt. Durch die Analyse der Ausgangssituation bei der Definition der Ziele können realistische Ziele an das PSS definiert werden (vgl. (Rupp 2004, 89)). Desweiteren ermöglicht eine Ist-

Analyse dem Auftragnehmer die eigene Leistungsfähigkeit zu erkennen und den Stand der Technik sowohl seitens des Kunden als auch seitens des Auftragnehmers zu erkennen (Pahl et al. 2006, 96-98).

Um die Ziele zu identifizieren, müssen den relevanten Stakeholdern die Fragen „Warum?“, „Was?“, „Wie?“ und „Wie noch?“ gestellt werden (Anwer/Ikram 2006). Die Frage nach „Warum?“ hilft nicht die Ziele selbst, sondern die Begründung für die Ziele aufzudecken. Typische Beispiele für die „Warum?“-Fragen könnten sein: „Warum soll das PSS entwickelt werden?“, „Was will der Kunde mit der Einführung des PSS erreichen / verbessern?“. Die beiden „Wie?“-Fragen helfen die unkonkreten „high level“-Ziele zu konkretisieren und eventuell auch alternative Ziele zu finden (De Landtsheer et al. 2003). Dazu gehören die Fragen: „Wie soll das PSS genutzt werden?“, „Wie soll das PSS entwickelt und bereitgestellt werden?“. Bei der Formulierung von Fragen zur Erhebung von Zielen sowie bei der Formulierung von Zielen steht die Lösungsneutralität im Vordergrund. Die Fragen sollen den Stakeholdern keine Lösungen vorgeben. Aber auch die gefundenen Ziele sollen keine Lösungen voraussetzen.

Die Ziele des Kunden sind grundlegend für die Ziele des Auftragnehmers bzgl. des PSS. Die Ziele des Auftragnehmers entscheiden darüber, ob die Ziele des Kunden erfüllt werden können. Zwischen den Geschäftszielen des Kunden und Auftragnehmers bestehen wechselseitige Abhängigkeiten. Das resultiert in der Notwendigkeit, dass bei der Konkretisierung der Geschäftsziele des Kunden die Geschäftsziele des Auftragnehmers berücksichtigt werden müssen und vice versa. Diese gegenseitige Konkretisierung ermöglicht es, die Ziele zu detaillieren oder sie einzuschränken, falls sich die Ziele widersprechen.

Die ermittelten Ziele sind auf mögliche Konflikte zu überprüfen, die entstehen, wenn sich zwei oder mehrere Ziele widersprechen. Die identifizierten Konflikte sind aufzulösen (Ahmad 2008; Easterbrook 1991). Wenn die Konflikte nicht aufgelöst werden, kann dadurch die Akzeptanz der Lösung bei den Stakeholdern beeinträchtigt werden. Das liegt daran, dass im Falle von widersprüchlichen Zielen, nicht alle Ziele erfüllt werden können.

Anschließend soll nach redundanten Zielen gesucht werden (Lindemann 2006, 100). Ziele heißen redundant, wenn zwei oder mehrere Ziele inhaltlich ähnliche Aspekte, bspw. denselben Sachverhalt, beschreiben, aber unterschiedlich genannt werden. Solche Ziele sollen zusammengefasst werden (s. Kapitel 6.1.3.5).

Die ermittelten Ziele werden validiert, indem sie den Kunden und Stakeholdern vorgelegt und auf Richtigkeit und Konsistenz überprüft werden. Anschließend sind die Ziele entsprechend den Spezifikationsvorschriften zu dokumentieren. Dazu werden die in Kapitel 6.3.1.3 angegebenen Spezifikationsartefakte, wie z.B. die „Zielspezifikation“ (in Anlehnung an „Market Requirements Document“ von Wiegers (2005, 81)) verwendet. Ebenso können die Ziele durch Und-Oder-Bäume (Pohl 2007, 103-107) dokumentiert werden (s. Kapitel 6.3.1.3).

7.3.1.2 Techniken zur Ermittlung und Analyse von Zielen an PSS

Zur Ermittlung von Zielen (Geschäftsziele des Kunden und Geschäftsziele des Auftragnehmers) eignet sich besonders die *vergangenheitsorientierte Technik* (vgl. (Humpert 1995; Husen 2007, 175; Pahl et al. 2006, 109-111; Rupp 2004, 89; Berkovich et al. 2011c)). Mit Hilfe dieser Technik, die auch als „Systemarchäologie“ bezeichnet wird, lassen sich bestehende Systeme oder Projekte und Situationen analysieren. Die vergangenheitsorientierte Technik ermöglicht Verbesserungsvorschläge an das neue System zu ermitteln. Bei der Systemarchäologie werden existierende Systeme und dazugehörige Unterlagen, wie Benutzerhandbücher,

bzgl. des Verhaltens existierender Systeme analysiert. Ebenso sollen Dokumente aus existierenden, verwandten Projekten analysiert werden, um daraus die schon ermittelten Ziele wiederzuverwenden. Diese Technik ist gut geeignet, wenn verwandte Lösungen angeboten werden sollen. Sie ist einfach anzuwenden, setzt aber voraus, dass entsprechende Unterlagen existieren. Wenn keine vergleichbaren Projekte oder System existieren, ist diese Technik nicht einsetzbar. Ebenso sollen die Rahmenbedingungen beim Einsatz dieser Technik überprüft werden. Wenn sich die Rahmenbedingungen schnell ändern, können sich die Ziele oder die Ist-Situation ebenso ändern, wodurch die Ergebnisse der Analyse schnell veralten.

Als weitere Möglichkeit zur Ermittlung und Analyse der Ist-Situation werden die Techniken aus dem Projektmanagement eingesetzt. Im Rahmen von *Benchmarking* wird die Ist-Situation des eigenen Unternehmens mit anderen erfolgreichen Unternehmen verglichen (vgl. (Hagemann 2009, 50; Lindemann 2006, 70; Krcmar 2010, 558ff.; Gorschek/Svahnberg 2005)). Die Technik „*Balanced Scorecard*“ unterstützt die Unternehmensziele darzustellen und dabei die Ziele ganzheitlich zu betrachten. Dabei werden die Ziele aus den relevanten Unternehmensbereichen gesammelt, geclustert und ausgewählt (vgl. (Lindemann 2006, 76; Young 2003, 169; Durraui et al. 2000)).

Desweiteren werden für die Ermittlung und Analyse von Zielen die Techniken eingesetzt, die ebenso für die Anforderungsermittlung eingesetzt werden (s. Kapitel 7.3.2.2). *Traditionelle Techniken* (bspw. Interviews), „*Group Elicitation*“-Techniken (bspw. Workshops, Prototyping und Simulation), *Modell-basierte Techniken* (bspw. Szenarien), *kognitive Techniken* (bspw. Card Sorting), und *kontextuelle Techniken* (bspw. Beobachtungen), sind ebenso zur Ermittlung von Zielen an das PSS geeignet (s. Kapitel 7.3.2.2). Beim Einsatz dieser Techniken soll der Unterschied zwischen den Zielen und den Anforderungen berücksichtigt werden. Die Ziele sind nur Anregungen für die PSS-Anforderungen. Sie sind lösungsneutral und stellen die Absichten des Kunden oder des Auftragnehmers dar.

Die Techniken zur Identifikation und Auflösung von Konflikten zwischen den Anforderungen werden in Kapitel 7.3.2.2 erläutert. Obwohl diese Techniken auf die Anforderungen zugeschnitten sind, können sie auch für die Ziele verwendet werden. Die Techniken zur Validierung von Zielen werden in Kapitel 7.3.4.2 beschrieben, da die Validierung als letzter Schritt vor der Übergabe der Anforderungen an die Entwicklung durchgeführt wird.

7.3.2 Ermittlung und Analyse von PSS-Anforderungen

Basierend auf den Zielen bzgl. der Bereitstellung und Nutzung des PSS, die zur Zielebene gehören, werden die Anforderungen an das PSS (PSS-Anforderungen der Systemebene) abgeleitet. Im ersten Schritt werden die Anforderungsquellen gesucht, und die PSS-Anforderungen ermittelt. Anschließend werden die Anforderungen bzgl. Konsistenz, Vollständigkeit, Korrektheit und mögliche Konflikte analysiert und entsprechend den Spezifikationsvorschriften dokumentiert. Das Ergebnis der Aktivitäten stellen analysierte, dokumentierte Anforderungen an das PSS (Systemebene) dar.

7.3.2.1 Aktivitäten zur Ermittlung und Analyse von PSS-Anforderungen

Abbildung 7-5 zeigt die Aktivitäten bei der Ermittlung und Analyse von PSS-Anforderungen.

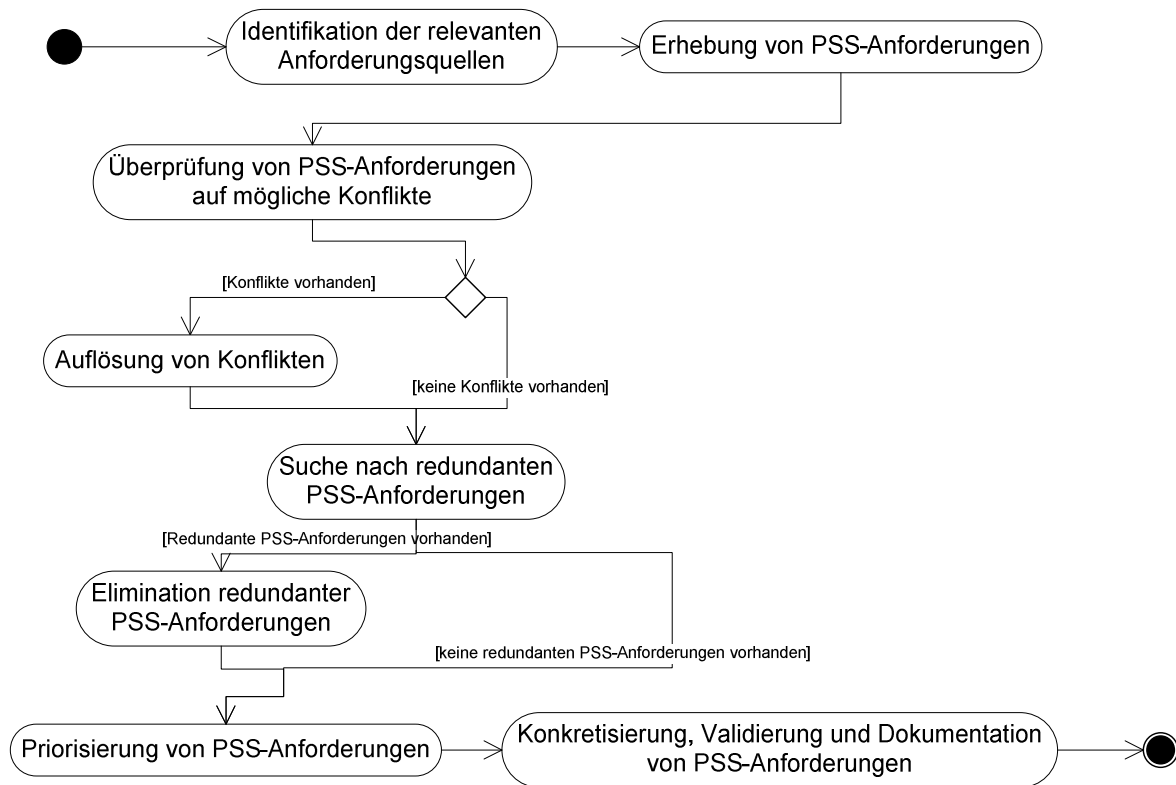


Abbildung 7-5: Aktivitäten der Ermittlung und Analyse von PSS-Anforderungen
Quelle: Eigene Darstellung

Zur Ermittlung und Analyse von PSS-Anforderungen werden folgende Aktivitäten durchgeführt:

1. Identifikation der relevanten Anforderungsquellen,
2. Erhebung von PSS-Anforderungen (Kunden- und Stakeholderanforderungen, Geschäftsprozessanforderungen, Umgebungsanforderungen und Auftragnehmeranforderungen) aus den Zielen (Geschäftsziele des Kunden und Geschäftsziele des Auftragnehmers),
3. Identifikation und Auflösung von Konflikten zwischen den Anforderungen (Anforderungsvereinbarung),
4. Identifikation und Elimination redundanter Anforderungen,
5. Priorisierung von Anforderungen,
6. Konkretisierung, Validierung und Dokumentation von Anforderungen.

Der Input zur Durchführung der oben genannten Aktivitäten stellen die Anforderungsartefakte „Geschäftsziele des Kunden“ und „Geschäftsziele des Auftragnehmers“ der Zielebene dar. Der Output der Durchführung der oben genannten Aktivitäten stellen die Anforderungsartefakte der Systemebene dar: „Kunden- und Stakeholderanforderungen“, „Geschäftsprozessanforderungen“, „Umgebungsanforderungen“ und „Auftragnehmeranforderungen“.

Die Anforderungsermittlung wird definiert als „learning, uncovering, extracting, surfacing, or discovering needs of customers, users, and other potential stakeholders“ (Hickey/Davis 2004, 67). Die Aktivitäten der Anforderungsermittlung sollen ein möglichst vollständiges Verständnis der Anforderungen an das zu entwickelnde System erzielen (Pohl 2007, 311; Cheng/Atlee 2007). Im Rahmen der Erhebung von Anforderungen soll ein Dialog mit den Stakeholdern aufgebaut werden, um sowohl implizite als auch explizite Anforderungen zu erheben

(Lindemann 2006, 94ff.). Dabei sollen bei der Anforderungsermittlung folgende Herausforderungen berücksichtigt werden (Sommerville/Sawyer 1997a, 64; Coughlan/Macredie 2002):

- Mangelhafte Kommunikationen zwischen den an der Bereitstellung und Nutzung des PSS beteiligten Personen, wie Entwickler, Kunden, Manager oder Anforderungsanalysten über die Erwartungen bzgl. des zu entwickelnden Systems,
- Das Fehlen eines gemeinsamen Verständnisses über das zu lösende Problem,
- Unangemessene, unvollständige oder unpräzise Beschreibung von Anforderungen und weiterer Entwicklungsinformationen in bestehenden Dokumenten,
- Die Stakeholder wissen oft nicht genau, was sie vom System wollen. Desweiteren können die Stakeholder ihre Anforderungen oft nicht präzise ausdrücken,
- Die Stakeholder drücken ihre Anforderungen in ihrer „Sprache“ aus und setzen dabei viele Aspekte implizit voraus.

Während der Anforderungsermittlung werden die Anforderungsquellen identifiziert und die Anforderungen erhoben (Lindemann 2006, 94ff.; Wiegers 2005, 115ff.; Husen 2007, 77ff.; Ramaswamy 1996, 51ff.). Als Grundlage zur Identifikation der Anforderungsquellen dienen die in Kapitel 6.3.2.2 definierten externen Artefakte der Systemebene. Die identifizierten Stakeholder sollen nach weiteren möglichen Anforderungsquellen gefragt werden, die anschließend auf ihre Relevanz für die Bereitstellung der Anforderungen überprüft werden sollen.

Basierend auf den ermittelten Zielen der Zielebene bzgl. der Bereitstellung und der Nutzung des PSS werden die PSS-Anforderungen ermittelt. Die Ziele geben dabei die thematische Richtung zur Formulierung der Fragen bzgl. der Anforderungen vor. Aus jedem vorhandenen Ziel werden eine oder mehrere Anforderungen abgeleitet. Die Ziele, aus denen die Anforderungen abgeleitet werden, werden bei der anschließenden Dokumentation der Anforderungen, bspw. im Lastenheft, durch die dafür vorgesehenen Attribute (s. Kapitel 6.2.2) festgehalten. Nur so kann sichergestellt werden, dass jede Anforderung begründet existiert (s. Kapitel 6.1.3).

Es werden die Kunden- und Stakeholderanforderungen, Auftragnehmeranforderungen, Geschäftsprozessanforderungen und Umgebungsanforderungen erhoben, die in den Anforderungsartefakten zusammengefasst sind. Als Checkliste zur Ermittlung von Anforderungen dient die Taxonomie der Anforderungen (s. Kapitel 6.3.2.1). Um die Anforderungen an das PSS aus den Geschäftsprozessen abzuleiten, sollen die relevanten Geschäftsprozesse analysiert werden. Das Ziel der Analyse von Geschäftsprozessen ist das Aufdecken der Zusammenhänge zwischen den Prozessen, das Erkennen von Problemen und Herausforderungen im Ist-Zustand, um mit der Einführung des PSS dagegen vorzugehen. Durch die Erfassung der für die Einführung des PSS relevanten Aspekte der Geschäftsprozesse, sollen zusätzliche Anforderungen an das PSS abgeleitet werden. Durch die Einführung des PSS können sich auch Änderungen an den Geschäftsprozessen ergeben. Ebenso ist zu überprüfen, in wieweit die Geschäftsprozesse vom PSS betroffen sind und durch die Einführung des PSS geändert bzw. angepasst werden müssen (vgl. (Rupp 2004, 183)).

Da sich alle Anforderungen gegenseitig beeinflussen, sollte dies bei der Anforderungsermittlung berücksichtigt werden. Die Anforderungen aus den Gesetzen können bspw. als Eingabe zur Ermittlung der Kundenanforderungen verwendet werden, indem bestimmte Rahmenbedingungen, die von den Gesetzen vorgegeben werden, in der Ermittlung der Kundenanforderungen aufgegriffen werden. So soll die Anforderungsermittlung iterativ zwischen den ver-

schiedenen Anforderungsartefakten der Systemebene stattfinden, da sich alle diese Anforderungen gegenseitig beeinflussen können.

Der Anforderungserhebung liegen folgende Fragestellungen zu Grunde (vgl. (Sommerville/Sawyer 1997a, 67-68; Husen 2007, 85)), die im Rahmen der Techniken (Kapitel 7.3.2.2) abgefragt werden sollen:

- Was soll das PSS leisten?
- Welche Vor- und Nachteile bringt das PSS für den Kunden und für den Auftragnehmer mit sich?
- Gibt es Einschränkungen bzgl. der Bereitstellung und Nutzung des PSS?
- Brauchen der Kunde / der Auftragnehmer das PSS?
- Was sind die Konsequenzen, wenn das PSS nicht entwickelt wird?
- In welcher direkten oder indirekten Weise unterstützt das PSS die Geschäftsprozesse des Kunden?
- Welche kritische Prozesse oder Objekte sollen bei der Entwicklung eines PSS berücksichtigt werden?
- Welchen Einfluss hat das PSS auf die Prozesse und Systeme des Kunden?
- Welche Anforderungen ergeben sich aus den Geschäftsprozessen an das PSS?
- Welche Assoziationen hat der Kunde mit der Inanspruchnahme des PSS?
- Welche Kriterien berücksichtigt der Kunde beim Erwerb des PSS?

Die ermittelten PSS-Anforderungen sollen anschließend auf Konsistenz und Konfliktfreiheit überprüft werden (Kotonya/Sommerville 1998, 59-60). Dafür sollen die Interdependenzen zwischen den Anforderungen identifiziert werden. Es können strukturelle Interdependenzen und einschränkende Interdependenzen unterschieden werden. Strukturelle Interdependenzen liegen vor, wenn zur Ableitung einer Anforderung andere Anforderungen benötigt werden. Einschränkende Interdependenzen liegen vor, wenn die Erfüllung einer Anforderung die Erfüllung einer anderen Anforderung voraussetzt oder wenn eine Anforderung einen Konflikt mit einer anderen Anforderung hat (Pohl 2007, 508-511; Dahlstedt/Persson 2005). Ein Konflikt zwischen den Anforderungen besteht, „wenn die Bedürfnisse und Wünsche verschiedener Stakeholder (oder Gruppen von Stakeholdern) an das geplante System sich widersprechen oder nicht alle Bedürfnisse und Wünsche berücksichtigt werden können“ (Pohl 2007, 394). Die Aktivitäten der Anforderungsvereinbarung sollen die Konflikte zwischen den Anforderungen identifizieren, analysieren, indem nicht nur die Konflikte selbst, sondern auch die Konfliktursachen analysiert werden, und schließlich die Konflikte auflösen (Lindemann 2006, 100-102; Wiegers 2005, 109-111; Ahmad 2008; Grünbacher/Seyff 2005). Konflikte zwischen den Anforderungen können durch Mangel an Informationen oder unterschiedliche Interpretation von Informationen (Sachkonflikte) verursacht werden. Auch verschiedene Interessen und Ziele von Stakeholdern (Interessenkonflikte) oder die Aufstellung von verschiedenen Kriterien zur Bewertung von Anforderungen durch die Stakeholder (Wertekonflikte) können Konflikte verursachen (in Anlehnung an (Moore 2003) zitiert in (Pohl 2007, 398)).

Wenn es redundante PSS-Anforderungen gibt, d. h. Anforderungen, die inhaltliche Ähnlichkeiten aufweisen, aber unterschiedlich benannt werden, sind diese Anforderungen durch eine sinnvoll konsolidierte Anforderung zu ersetzen (Lindemann 2006, 100; Rupp 2004, 166-167; Pohl 2007, 508-511).

Anschließend sind die PSS-Anforderungen zu priorisieren. Die Priorisierung von Anforderungen unterstützt (Berander/Andrews 2005; Beiter et al. 2006; Ramaswamy 1996, 28-29):

- Die Entscheidungen der Stakeholder bzgl. der wichtigen Anforderungen, die auf jeden Fall zu realisieren sind, und Anforderungen, deren Realisierung für den Kunden vorrangig ist,
- Die Abschätzung der Zufriedenheit des Kunden durch den Auftragnehmer,
- Festlegung der Reihenfolge und der Sammlung der Anforderungen zur Realisierung.

Zur Priorisierung von PSS-Anforderungen kann eine Skala verwendet werden, die die Anforderungen nach ihrer Wichtigkeit charakterisiert. Die Anforderungen können eine sehr hohe, mittlere und niedrigere Priorität bekommen (Wiegers 2005, 251). Der Priorisierung können solche Faktoren, wie Wichtigkeit für die Stakeholder, Zeitaufwand, Kostenaufwand, Risiko oder eine Kombination von diesen Faktoren zu Grunde liegen (Berander/Andrews 2005).

Im nächsten Schritt sind die PSS-Anforderungen zu priorisieren. Die Priorisierung von Anforderungen unterstützt (Berander/Andrews 2005; Beiter et al. 2006; Ramaswamy 1996, 28-29):

- Die Entscheidungen der Stakeholder bzgl. der wichtigen Anforderungen, die auf jeden Fall zu realisieren sind, und Anforderungen, deren Realisierung für den Kunden vorrangig ist,
- Die Abschätzung der Zufriedenheit des Kunden durch den Auftragnehmer,
- Festlegung der Reihenfolge und der Sammlung der Anforderungen zur Realisierung.

Zur Priorisierung von PSS-Anforderungen kann eine Skala verwendet werden, die die Anforderungen nach ihrer Wichtigkeit charakterisiert. Die Anforderungen können eine sehr hohe, mittlere und niedrigere Priorität bekommen (Wiegers 2005, 251). Der Priorisierung können solche Faktoren, wie Wichtigkeit für die Stakeholder, Zeitaufwand, Kostenaufwand, Risiko, Unbeständigkeit oder eine Kombination von diesen Faktoren zu Grunde liegen (Berander/Andrews 2005).

Im nächsten Schritt wird die Anforderungsanalyse durchgeführt, dessen Ziel es ist, die PSS-Anforderungen so zu konkretisieren, dass sie im nächsten Schritt der Entwicklung als Input zum Erstellen von Entwicklungsartefakten und zur Ableitung von Design-Anforderungen der Eigenschaftsebene verwendet werden können. Desweiteren soll die Anforderungsanalyse sicherstellen, dass die Anforderungen korrekt und vollständig sind.

Zur Durchführung der Anforderungsanalyse kann eine Checkliste eingesetzt werden (Sommerville/Sawyer 1997a, 117-120):

- Enthält die PSS-Anforderung vorzeitige Design- oder Implementierungsinformationen?
- Beschreibt die PSS-Anforderung exakt eine Anforderung oder könnte sie in mehrere Anforderungen aufgeteilt werden?
- Beschreibt die PSS-Anforderung einen kosmetischen Zusatz zum System, der nicht wirklich wichtig ist?
- Ist die PSS-Anforderung konsistent zu ihren Zielen in der Zielebene?
- Ist die PSS-Anforderung zweideutig? D.h. kann sie von unterschiedlichen Personen auf unterschiedliche Weise aufgefasst werden?
- Ist die PSS-Anforderung testbar? D.h. Können aus der Anforderungen Testfälle abgeleitet werden, mit denen die Erfüllung der Anforderung geprüft werden kann?

Die gewonnenen PSS-Anforderungen sind noch in „der Sprache der Stakeholder“ ausgedrückt. Sie sind noch unkonkret und können nicht direkt an die Entwicklung weitergegeben

werden, d. h. aus denen können noch keine Design-Anforderungen der Eigenschaftsebene abgeleitet werden. Das Ziel der Konkretisierung ist es, die PSS-Anforderungen aus verschiedenen Blickwinkeln so darzustellen, dass die Entwickler daraus Vorgaben an die Realisierung von Anforderungen ableiten können. Um die Konkretisierung zu unterstützen, werden die Anforderungen und die, für die Realisierung der Anforderungen relevante Umgebung, modelliert. Hull et al. (2004, 17) weisen darauf hin: „Modelling supports the design activity, and is where most of the creative work takes place. It assists the engineer in understanding enough of the system to decompose the requirements at a particular level into the next level down. The requirements themselves are a complete snapshot of what is required at each level in increasing levels of detail“. Durch das Einsetzen von Modellen werden verschiedene Aspekte der Anforderungen fokussiert beleuchtet. Dazu gehören bspw. das Verhalten oder die Struktur des PSS. Aus den Modellen der Anforderungen und ihrer Umgebung können konkretere Anforderungen, die bestimmte Aspekte des PSS fokussiert beschreiben, abgeleitet werden. Die Modelle, die durch Diagramme oder Prosa präsentiert sind, stellen Spezifikationsartefakte der Systemebene dar (s. Kapitel 6.3.2.3). Die Diagramme werden neben Prosa zur Dokumentation von Anforderungen verwendet. Die Anforderungsverfolgung soll sichergestellt werden, indem für jede PSS-Anforderung die Anforderungsquelle und das Ziel aus der Zielebene, aus dem die PSS-Anforderung abgeleitet wurde, festgehalten und im Lastenheft dokumentiert wird. Bei möglichen Änderungen an PSS-Anforderungen ist es dadurch möglich, die Ziele zu identifizieren, die von den Änderungen betroffen werden können, um die Änderungen entsprechend zu integrieren. Somit wird die Pre-Traceability gewährleistet (die Definition der Traceability ist in Kapitel 4.4.2.2 gegeben). Ebenso sollen im Lastenheft die Hinweise auf die Design-Anforderungen enthalten sein, die aus den PSS-Anforderungen abgeleitet sind, um damit die Post-Traceability zu gewährleisten.

7.3.2.2 Techniken zur Ermittlung von Anforderungen an PSS

Die Qualität von Anforderungen wird davon beeinflusst, welche Techniken zur Anforderungsermittlung gewählt wurden (Hickey/Davis 2003). Die Auswahl der Technik oder einer Kombination mehrerer Techniken hängt von den vorhandenen Zeit- und Kostenressourcen ab (Nuseibeh/Easterbrook 2000). Die Techniken der Anforderungsermittlung lassen sich in sechs Gruppen einteilen (Kotonya/Sommerville 1998; Nuseibeh/Easterbrook 2000; Rupp 2004, 110): traditionelle Techniken, „Group Elicitation“-Techniken, Prototyping und Simulation, Modell-basierte Techniken, Kognitive Techniken, Kontextuelle Techniken. Durch die Anwendung dieser Techniken werden die PSS-Anforderungen, die in den Anforderungsartefakten „Kunden- und Stakeholderanforderungen“, „Geschäftsprozessanforderungen“, „Umgebungsanforderungen“ und „Auftragnehmeranforderungen“ zusammengefasst sind, ermittelt.

7.3.2.2.1 Traditionelle Techniken

Traditionelle Techniken stellen Techniken zur Sammlung generischen Daten dar. Dazu gehören bspw. Interviews, Analyse existierender Dokumente und schriftliche Befragungen (Ulrich/Eppinger 2003, 56f.; Pohl 2007, 323ff.; Byrd et al. 1992; Kotonya/Sommerville 1998, 62ff.; Coughlan/Macredie 2002; Hickey/Davis 2003; Hengst et al. 2004; Hull et al. 2004, 107; Lindemann 2006, 68f.; Ahrens 2000, 122; Breiing/Flemming 1993, 12; Aurum/Wohlin 2005b; Ramaswamy 1996, 51ff.; Ehrlenspiel 2002, 344; Humpert 1995, 38; Zowghi/Coulin 2005; Tuunanen 2003; Berkovich et al. 2011c).

Interviews sind „essentially human based social activities“ (Zowghi/Coulin 2005). Die Effektivität der Interviews bei der Anforderungsermittlung hängt von der Qualität der Interaktionen zwischen den Teilnehmern ab (Zowghi/Coulin 2005). Interviews sind besonders dann gut geeignet, wenn die Informationen schnell beschafft werden sollen. Bevor ein Interview durchgeführt wird, ist es wichtig, das Ziel des Interviews zu bestimmen, das die Vorbereitung und die Durchführung des Interviews beeinflusst. Es können drei Arten von Interviews unterschieden werden: unstrukturierte (explorativ), standardisierte und semi-standardisierte Interviews. Explorative Interviews werden in Form eines „offenen“ Gesprächs geführt, ohne vorbereitete Fragen. Explorative Interviews sind besonders in den Situationen geeignet, wenn es darum geht, einen tiefen Einblick in einen Bereich zu bekommen. Standardisierte Interviews werden anhand von Fragen geführt. Der Erfolg von standardisierten Interviews hängt davon ab, ob es sich um „richtige“ Fragen handelt, durch die man die nötige Information bekommen kann. Die standardisierten Interviews sind besonders dann geeignet, wenn die Meinungen von Stakeholdern zum gleichen Sachverhalt abgefragt werden sollen. Die Ergebnisse der Interviews sind nachzubereiten, indem ein Protokoll des Interviews erstellt wird und die gewonnenen Anforderungen festgehalten werden.

Schriftliche Befragungen werden oft in den frühen Phasen der Anforderungsermittlung eingesetzt. Die Voraussetzung zur Durchführung einer schriftlichen Befragung ist, dass ein gemeinsames Verständnis über den befragten Bereich bei allen beteiligten Stakeholdern besteht. Die Fragen sind so zu formulieren, dass keine redundanten Informationen abgefragt und gesammelt werden. Basierend auf den Zielen der Befragung werden die Stakeholder identifiziert und die Fragen formuliert. Es können sowohl offene als auch geschlossene Fragen formuliert werden. Bei geschlossenen Fragen sind die vorgegebenen Antworten zu definieren, bei offenen Fragen ist die Dokumentationsform für die Antworten festzulegen.

Der *Analyse existierender Dokumente* liegt die Idee zu Grunde, bestehende Dokumente aus einer bestimmten Perspektive, bspw. aus der Perspektive des Anwenders, zu lesen. Der erste Schritt bei der Analyse ist die Definition des Ziels der Analyse existierender Dokumente, sowie der erwarteten Ergebnisse. Basierend darauf können geeignete Perspektiven zum Lesen gewählt werden. Aus den vorhandenen Anforderungsquellen werden die Dokumente ausgewählt, aus denen durch das perspektivenbasierte Lesen neue Anforderungen gewonnen werden können. Das perspektivenbasierte Lesen ermöglicht die Identifikation von Zielen, die das zu entwickelnde System erfüllen soll, Szenarien, die vom zu entwickelnden System umzusetzen sind, und lösungsorientierten Anforderungen, die in einer konkreten für die Umsetzung geeigneten Form vorliegen. Das Analysieren von existierenden Dokumenten erlaubt das Domänenwissen, das für die Entwicklung des Systems relevant ist, zu erweitern sowie Konzepte und Komponenten zu identifizieren, die wiederverwendet werden können. Als die zu analysierenden Dokumente können Entwurfsdokumente, Bedienungsanleitungen für existierende Systeme sowie weitere Dokumente, die für die Entwicklung und die Verwendung des Systems genutzt werden, dienen.

Die traditionellen Techniken eignen sich gut, um die initialen PSS-Anforderungen schnell und einfach zu gewinnen. Die Techniken bieten eine gute Möglichkeit, die Anforderungen möglichst vollständig zu erheben, vorausgesetzt die richtige Art der Technik wurde ausgewählt und die richtigen Fragen wurden gestellt. Durch die Unterstützung der Kommunikation durch die Techniken wird das Verständnis zwischen den an der Entwicklung von PSS beteiligten Domänen gefördert. Zu den Vorteilen dieser Techniken gehört ihre einfache Anwendung. Zu den Nachteilen der Techniken gehört die Gefahr, viele unwichtige Informationen zu sammeln,

aus denen es problematisch ist, die Anforderungen abzuleiten. Desweiteren eignen sich die traditionellen Techniken besonders gut für Prozess-Informationen, da Stakeholder gut aussagen können, welche Aktivitäten sie in ihrer täglichen Arbeit ausführen. Die Techniken berücksichtigen aber keine impliziten Anforderungen der Stakeholder, die nicht direkt ausgedrückt werden (Maiden/Rugg 1996). Die Techniken sind nicht geeignet, wenn die Stakeholder schlechte kommunikative Fähigkeiten aufweisen. Die Techniken sind gut geeignet, wenn die Stakeholder ein geringes Abstraktionsvermögen besitzen, bei einer hohen Kritikalität des zu entwickelnden Systems, bei einer geringen Beobachtbarkeit des zu entwickelnden Systems, sowie zur Ermittlung detaillierter Anforderungen (Rupp 2004, 110). Interviews und Befragungen unterstützen die Identifikation von Anforderungsquellen, die Gewinnung existierender sowie die Entwicklung neuer und innovativer Anforderungen. Die Analyse existierender Dokumente ist nur für die Gewinnung existierender Anforderungen geeignet (Pohl 2007, 324).

Die durch diese Techniken ermittelten Anforderungen können in einem Lastenheft (s. Kapitel 6.3.2.3) dokumentiert werden.

7.3.2.2.2 „Group Elicitation“-Techniken

„*Group Elicitation*“-Techniken zielen darauf ab die Übereinstimmung zwischen den Stakeholdern und die Zustimmung der Stakeholder zu den Anforderungen zu fördern, indem gruppendynamische Effekte ausgenutzt werden, um ein besseres Verständnis der Bedürfnisse zu erreichen. Zu diesen Techniken zählen bspw. Brainstorming, Focus Groups und Workshops, (Ehrlenspiel 2002, 413f.; Fritzsche 2007, 73-75; Ramaswamy 1996, 51ff.; Coughlan/Macredie 2002; Hengst et al. 2004; Pohl 2007, 323ff.; Kotonya/Sommerville 1998, 62ff.; Robertson/Robertson 2007, 116ff.; Grube/Schmid 2008; Lindemann 2006, 139ff.; Aurum/Wohlin 2005a; Humpert 1995, 40f.; Ahrens 2000, 122; Menten et al. ; Berenbach et al. 2009, 55-56; Osborn 1979; Berkovich et al. 2011c).

Das Ziel von *Brainstorming* ist das Sammeln von Ideen. Die Teilnehmer aus verschiedenen Stakeholdergruppen stellen im Rahmen einer Sitzung Ideen vor, die zuerst ohne Kommentare gesammelt und notiert werden. Die gesammelten Ideen werden von den Teilnehmern der Sitzung zur Entwicklung eigener Ideen genutzt. Zu vermeiden sind dabei jegliche Kritiken der gesammelten Ideen. Während der Sitzung sollen möglichst viele Ideen gesammelt werden. Auf die Qualität der Ideen wird nicht direkt eingegangen. Wenn die Ideen gesammelt sind, werden sie analysiert und nach drei Kategorien, wie unmittelbar verwendbare Ideen, potenziell verwendbare Ideen und unbrauchbare Ideen, aufgeteilt. Die unbrauchbaren Ideen werden nicht mehr betrachtet. Anschließend werden die Ideen den Teilnehmern präsentiert.

Focus Group ist eine Form der Gruppendiskussion. Der Aufwand für die Organisation einer Focus Group ist gering. Focus Groups sind effektiv, weil die Stakeholder in den Prozess der Anforderungsermittlung direkt einbezogen werden können. An einer Focus Group sollten Stakeholder verschiedener Gruppen sowie ein Moderator als neutrale Person teilnehmen. Beim Eröffnen der Focus Group wird das Thema der Diskussion vorgestellt. Es sollte keiner der Teilnehmer dominieren. Außerdem ist darauf zu achten, dass jedes Gruppenmitglied zu Wort kommt. Wenn die Ideen und Vorschläge gesammelt sind, werden sie analysiert, kommentiert, erneut überarbeitet und kombiniert.

In einem *Workshop* entwickeln die Stakeholder gemeinsam abgestimmte Anforderungen an das zu entwickelnde System. Somit stellen die Anforderungen das Ergebnis der gemeinsamen

Arbeit, die im Rahmen einer Sitzung erfolgt, dar. Ein Workshop hat immer einen festgelegten Ablauf und klare Regeln. Wenn es schon Anforderungen gibt, können sie in einem Workshop nach ihrer Relevanz geordnet und konkretisiert werden. Oft werden in einem Workshop weitere Techniken zur Unterstützung der Anforderungserhebung eingesetzt. Zur Identifikation der Anforderungsquellen kann Brainstorming verwendet werden. Die Durchführung von Workshops ist oft mit einem hohen Ressourcenaufwand verbunden, der sich aus den Aktivitäten zur Organisation und Nachbereitung des Workshops zusammensetzt. Die Vorteile eines Workshops liegen in der direkten Kooperation und der Förderung der Kompromissbereitschaft der Stakeholder. Vor der Durchführung eines Workshops ist es wichtig, die Ziele, die im Workshop erzielt werden sollen, genau zu definieren.

Die „Group Elicitation“-Techniken sind geeignet, um die Anforderungsquellen zu identifizieren und neue, innovative Anforderungen zu sammeln. Workshops unterstützen auch die Erhebung von existierenden Anforderungen. Ähnlich wie die traditionellen Techniken eignen sich diese Techniken gut, um Prozess-Informationen zu ermitteln. Das Erheben der impliziten Anforderungen wird nur bedingt unterstützt. Bei divergierenden Stakeholdermeinungen und einer problematischen Gruppendynamik ist Brainstorming schlecht geeignet. Ebenso können mit Hilfe von Brainstorming keine detaillierten Anforderungen erhoben werden. Workshops sind geeignet, wenn es um eine problematische Gruppendynamik geht, das zu entwickelnde System eine hohe Kritikalität und einen großen Umfang besitzt (Rupp 2004, 110).

Die durch diese Techniken ermittelten Anforderungen können in einem Lastenheft (s. Kapitel 6.3.2.3) dokumentiert werden.

7.3.2.2.3 Prototyping

Prototyping ist eine analytischengetriebene Technik, die die Entwicklung einer ausführbaren Pilot-Version des Systems beinhaltet (Byrd et al. 1992; Alavi 1984; Kotonya/Sommerville 1996, 73-75; Pohl 2007, 367ff.; Andriole 1994; Hull et al. 2004, 106ff.; Tuunanen 2003). In der Produktentwicklung wird dafür der Begriff „Simulation“ verwendet (Seiffert/Rainer 2008, 345-347; Pelz 2001, 5-8; Berkovich et al. 2011c).

Ein Prototyp ist ein Gegenstand, eine initiale Version des zu entwickelnden Systems, die den Stakeholdern bestimmte Konzepte demonstrieren soll. Ein Prototyp ist oft eine partielle Repräsentation des Gesamtsystems, die nur für das Klären von Anforderungen geeignet ist. Das Ziel dabei ist es, durch Prototypen inadäquate Anforderungen zu identifizieren (Byrd et al. 1992). Die Stakeholder können den Prototyp real wahrnehmen, wodurch ihr Feedback konkreter wird. Ein Prototyp stellt eine Grundlage für die Diskussion zwischen den Entwicklern und den Stakeholdern dar, um Probleme sowie weitere Möglichkeiten beim zu entwickelnden System zu identifizieren. Zu den Nachteilen von Prototyping gehören die Fokussierung und der Abstraktionsmangel seitens der Stakeholder. Oft verbinden die Stakeholder mit dem Prototypen unerfüllte Wünsche, die auch vom System nicht realisiert werden können. Prototypen sind schwer zu verwalten und zu kontrollieren. Für große Systeme sind sie nicht praktikabel. Das liegt daran, dass es kaum möglich ist, durch einen Prototypen die komplette Funktionalität des Systems darzustellen. Prototypen sind besonders dann gut geeignet, wenn es um das Verdeutlichen von schon ermittelten Anforderungen geht. Prototypen können auch dann eingesetzt werden, wenn es um die Ermittlung von Anforderungen geht und die Stakeholder nur vage Vorstellungen von ihren Wünschen haben. Ihr Einsatz für die Ermittlung von Anforderungen an Dienstleistungen erweist sich als problematisch.

7.3.2.2.4 Modell-basierte Techniken

Modell-basierte Techniken leiten die Erhebung auf Basis eines Modells, das die zu erhebenden Daten darstellt. Dazu gehört bspw. Anwendungsfall-Modellierung (Use-Case) und Szenariotechnik (Pahl et al. 2006, 219-221; Lindemann 2006, 78; Alexander 2003; Anderson/Durney 1993; Jarke et al. 1999; Sutcliffe 2002, 48-51; Wright 1992; Rupp 2004, 197-205; Jakimi et al. 2007; Wiegers 2005, 121ff.).

Anwendungsfälle (Use-Cases) beschreiben typische Interaktionen eines Anwenders mit einem System. Ein Anwendungsfall stellt das Verhalten des Systems aus der Sicht des Anwenders dar und beschreibt somit die Anforderungen, die das System erfüllen soll. Ein Use-Case-Diagramm umfasst mehrere Anwendungsfälle, ihre Beziehungen untereinander und zu den beteiligten Akteuren. Zu den Vorteilen von Anwendungsfalldiagrammen gehört, dass sie einen für die Stakeholder gut verständlichen Überblick über die Anwendungsfälle, Akteure und deren Beziehungen untereinander geben. Auch die Beschreibung der Anwendungsfälle selbst ist für die Stakeholder gut verständlich. Zu den Nachteilen der Beschreibung von Anforderungen in Form von Anwendungsfällen gehört, dass sie nur die Interaktion zwischen System und Akteur abbilden können. Daher können sie insbesondere nicht-funktionale Anforderungen nicht erfassen. Zudem kommt es in der Praxis oft vor, dass mehrere Anwendungsfälle ähnliche Sachverhalte beschreiben, was zu Redundanzen in den Anwendungsfällen führt.

Szenarien beschreiben konkrete Beispiele für die Erfüllung oder Nicht-Erfüllung einer oder mehrerer Anforderungen. Sie beschreiben Situationen, in denen das zu entwickelnde System genutzt wird. Szenarien werden meist in natürlicher Sprache erstellt. Zur Erhebung von Anforderungen werden oft positive, negative und Missbrauchsszenarien eingesetzt. Ein positives Szenario beschreibt eine Interaktion, die zur Erfüllung der mit dem Szenario im Zusammenhang stehenden Anforderungen führt. Ein negatives Szenario beschreibt eine Interaktion, die zur Nicht-Erfüllung der mit dem Szenario im Zusammenhang stehenden Anforderungen führt. Ein Missbrauchsszenario beschreibt eine Interaktion, die nicht vom System erlaubt ist (Pohl 2007, 127ff.).

Anwendungsfälle und Szenarien sind Spezifikationsartefakte für PSS-Anforderungen, Design-Anforderungen, Funktionsstruktur-Anforderungen und Domänen-Anforderungen (s. Kapitel 6.3.2.3, 6.3.3.4, 6.3.4.4, 6.3.5.4).

7.3.2.2.5 Kognitive Techniken

Kognitive Techniken wurden ursprünglich zur Identifikation und Sammlung von Wissen entwickelt. Dazu gehören bspw. Card Sorting und Repertory Grids (Nurmuliani et al. 2004; Niu/Easterbrook 2006; Nuseibeh/Easterbrook 2000; Byrd et al. 1992; Sutcliffe 2002, 48-51; Davis et al. 2006a; Berkovich et al. 2011c).

Als Voraussetzung für die Anwendung von *Card Sorting* muss bereits ein Kategoriensystem vorhanden sein. So ein Kategoriensystem beinhaltet im Requirements Engineering beispielsweise eine initiale Auswahl an Domänenobjekten, möglichen Softwarefunktionen, oder Softwarepaketen. Der Requirements Analyst schreibt ca. 60 dieser Begriffe auf Kärtchen, die den Stakeholdern vorgelegt werden. Die Stakeholder müssen diese Kärtchen dann in Kategorien anordnen, den Kategorien Namen geben, sowie Begründungen für die Wahl der Kategorien und Begründungen für die Zuordnung der einzelnen Begriffe zu den Kategorien erläutern. Diese Technik eignet sich sehr gut, um einen Eindruck des Verständnisses und der Wahrneh-

mung des Problems durch den Stakeholder zu bekommen. Weiterhin kann auf diese Weise eine Unterscheidung von high-level und low-level Problemen vorgenommen werden. Die Kategorien, die der Stakeholder erstellt, können auch als Input für weitere Techniken, oder für die Kategorisierung der Anforderungen verwendet werden. Ein weiterer Vorteil von Card Sorting ist, dass es schnell durchgeführt werden kann und nur wenig Aufwand verursacht. Jedoch ist Card Sorting nur dann anwendbar, wenn bereits das Kategoriensystem vorgegeben ist. Zudem erzeugt es keine komplett neuen Anforderungen, sondern erleichtert nur das Verständnis von bereits bestehenden Anforderungen.

Die Technik *Repertory Grid* geht auf die Personal Construct Theory von Kelly (1998) zurück. Diese Theorie geht davon aus, dass die Art, in der Personen mit ihrer Umgebung interagieren, auf deren Repertoire von Konstrukten der Welt schließen lässt. Die Repertory Grid Technik zielt darauf ab, es einer Person zu ermöglichen diese Konstrukte explizit zu äußern. Bei der Anwendung dieser Technik werden die Stakeholder aufgefordert, Unterscheidungen zwischen Konstrukten und deren Erfahrungen (genannt Elemente) zu machen. Diese werden in einer Tabelle (genannt Grid) festgehalten und vom Stakeholder bewertet. Das Ziel ist es, dass der Stakeholder möglichst viele Konstrukte äußert, diese in das Grid einordnet und bewertet. Dadurch lässt sich dann das Bild, das der Stakeholder von der analysierten Domäne hat, ableiten. Repertory Grids können zur Identifikation von allgemeinen Wünschen und Einstellungen des Stakeholders verwendet werden, sowie um unterschiedliche Einstellungen zwischen verschiedenen Stakeholdern zu erheben.

Die kognitiven Techniken sind gut geeignet, um das implizite Wissen zu sammeln. Die Techniken Card Sorting und Repertory Grids werden vor allem in den frühen Phasen der Anforderungsermittlung eingesetzt. Sie sind dazu geeignet das Wissen der Stakeholder über das zu lösende Problem abzufragen und zu strukturieren. Die kognitiven Techniken haben allerdings den Nachteil, dass sie nicht geeignet sind, um spezifische Fragen an die Stakeholder zu klären, denn sie beruhen darauf, dass der Stakeholder selbst die ihm als wichtig erscheinenden Informationen liefert.

7.3.2.2.6 Kontextuelle Techniken

Kontextuelle Techniken stellen eine Kombination aus den traditionellen und kognitiven Techniken zur Anforderungsermittlung dar (Goguen/Linde 1993). Dazu gehören bspw. Beobachtungen (Robertson/Robertson 2007, 101-105; Zave 1997; Pohl 2007, 346-352; Kotonya/Sommerville 1998, 69f.; Haumer et al. 1998; Berkovich et al. 2011c).

Im Rahmen einer *Beobachtung*, im Kontext des Requirements Engineering, werden die Anforderungen aus den Tätigkeitsfeldern der Stakeholder, während ihrer Durchführung, abgeleitet. Die Ergebnisse der Beobachtung sollen dokumentiert werden. Bei einer direkten Beobachtung werden die Stakeholder bei der Durchführung ihrer Tätigkeiten beobachtet. Wenn die Anforderungsquellen Systeme sind, werden die Systemnutzung oder der Systembetrieb beobachtet. Die ethnographische Beobachtung setzt voraus, dass der Beobachter über einen längeren Zeitraum die Stakeholder beobachtet, um ihre Arbeitsweisen zu lernen. Für jede Beobachtung soll ein Ziel definiert werden, das die Art der Ergebnisse der Beobachtung vor der eigentlichen Durchführung festlegt. Aus einer Beobachtung können nicht nur Ziele oder Anforderungen abgeleitet werden, sondern auch Szenarien. Während der Beobachtungen können neue Anforderungsquellen identifiziert werden, die weitere Anforderungen an das PSS liefern. Beobachtungen sind aber nicht dafür geeignet, um neue, innovative Anforderungen zu

erheben. Die Durchführung von Beobachtungen ist oft mit einem hohen Ressourcenaufwand verbunden (Pohl 2007, 351). Beobachtungen sind gut geeignet, wenn die zu beobachtenden Stakeholder über schlechte kommunikative Fähigkeiten verfügen. Dennoch müssen die Stakeholder für eine Beobachtung aufgeschlossen sein und die Kooperation mit dem Beobachter zulassen (Rupp 2004, 110).

7.3.2.3 Techniken zur Anforderungsvereinbarung

Die Techniken zur Anforderungsvereinbarung sollen in der Lage sein die Konflikte zwischen den Anforderungen zu identifizieren und aufzulösen (Berkovich et al. 2011c; Grünbacher/Seyff 2005; Jarke et al. 1996; Boehm/In 1996).

Zur Identifikation von Konflikten werden *Konsistenzmatrizen* eingesetzt (Husen 2007, 97; Lindemann 2006, 100; Pohl 2007, 409-410; Kotonya/Sommerville 1998, 78-80). Eine Konsistenzmatrix stellt die Anforderungen einander gegenüber, indem die Interdependenzen zwischen den Anforderungen dargestellt werden. Die Zeilen und Spalten einer Konsistenzmatrix stellen Anforderungen dar. Wenn die Anforderung in der Zeile i mit der Anforderung in der Spalte j durch eine Interdependenzbeziehung verbunden ist, wird das in der Matrix durch den Eintrag (i,j) vermerkt (Pohl 2007, 514-515). Somit gibt die Konsistenzmatrix an, welche Anforderungen einander beeinflussen und welche Anforderungen voneinander unabhängig sind. Die in der Konsistenzmatrix festgehaltenen Abhängigkeiten zwischen den Anforderungen werden zur Identifikation von Konflikten verwendet, denn wenn zwei oder mehrere Anforderungen einander beeinflussen, kann dies zu Widersprüchen zwischen den Anforderungen führen. Die Konsistenzmatrix ermöglicht die Identifikation der Auswirkungen von Anforderungsänderungen auf weitere Anforderungen (Jönsson/Lindvall 2005). Wenn sich eine Anforderung ändert, können sich Änderungen an weiteren Anforderungen ergeben, mit denen sie durch die Interdependenzbeziehungen verbunden ist. Ebenso unterstützt die Konsistenzmatrix die gegenseitige Konkretisierung von Anforderungen (s. Kapitel 6.1.3.5), indem in die Konkretisierung einer Anforderung die durch sie beeinflussende Anforderung einbezogen wird. Zur Identifikation der Interdependenzbeziehung eignen sich die Techniken der Anforderungsermittlung (s. Kapitel 7.3.2.2) und der Anforderungsanalyse (s. Kapitel 7.3.2.4).

Zur Auflösung von Konflikten können *Verhandlungen*, *kreative Lösungen* und *Entscheidungen* eingesetzt werden. Der Konfliktauflösung durch Verhandlungen liegt der Meinungsaustausch zwischen den Stakeholdern zu Grunde, die die konfliktären Anforderungen initiieren. Es soll ein Dialog zwischen den beteiligten Parteien aufgebaut werden, um die Argumente bzgl. des Konfliktes und möglicher Lösungen auszutauschen und einen Kompromiss zu finden. Kreative Lösungen werden zur Konfliktauflösung verwendet, wenn eine dritte, nicht die durch den Konflikt betroffene Partei eine Lösung des Konflikts vorschlägt. Wenn sich die Parteien nicht auf eine Lösung des Konfliktes einigen können, kann ein Entscheidungsträger, der eine übergeordnete Rolle bei den beteiligten Parteien spielt, die Entscheidung bzgl. der Auflösung des Konfliktes treffen (Pohl 2007, 401ff.; Easterbrook 1991).

Der *Win-Win-Ansatz* unterstützt die Konfliktauflösung, indem drei Situationen, Win-Lose-, Loose-Lose- und Win-Win-Situationen, unterschieden werden (Pohl 2007, 407-409). Die Win-Win-Situation bezeichnet eine Art der Konfliktauflösung, bei der alle beteiligten Parteien mit der Lösung zufrieden sind. Um diese Situation zu erreichen, sollen bei den Stakeholdern, vom Beginn des Projektes an, angemessene Erwartungen bzgl. der Anforderungen und ihrer Realisierung erzeugt werden. Zu erreichen ist dabei, dass kein Interessengegensatz zwi-

schen den beteiligten Parteien vorliegt. Für den Win-Win-Ansatz wurde eine IT-Unterstützung entwickelt (Briggs/Grünbacher 2002; Boehm et al. 1994; Boehm et al. 2001; Boehm/In 1996).

7.3.2.4 Techniken zur Anforderungskonkretisierung

Im Rahmen der Anforderungskonkretisierung werden Anforderungen aus verschiedenen Perspektiven betrachtet, um dadurch den Anforderungen konkretere und detailliertere Eigenschaften zuzuordnen.

Die Darstellung von Anforderungen mit Hilfe von *konzeptionellen Modellen* stellt eine Technik zur Konkretisierung von Anforderungen dar. Dabei werden nur bestimmte Facetten von Anforderungen betrachtet. Die konzeptionellen Modelle stellen die Anforderungen in Form eines Diagramms dar und sind daher geeignet, um die Anforderungen zu dokumentieren. Die resultierenden Diagramme wurden daher als Spezifikationsartefakte (s. Kapitel 6.3.2.3, Kapitel 6.3.3.4, Kapitel 6.3.4.4, Kapitel 6.3.5.4) eingeführt: Die Klassendiagramme und Entity-Relationship-Diagramme (ER-Diagramme) ermöglichen die Struktur des PSS und somit die statischen Eigenschaften des PSS darzustellen (Booch et al. 2007; Santos/Machado ; Wieggers 2005, 200; Vidgen 2003; Nuseibeh/Easterbrook 2000; Breu 2001, 12). Aktivitätsdiagramme, Zustandsdiagramme sowie ereignisgesteuerte Prozessketten (EPK) beschreiben die Anforderungen an das Verhalten des PSS, Interaktionen der Benutzer und die Abläufe im Rahmen des PSS (vgl. (Rupp 2004, 205ff.; Lamsweerde 2009, 160ff.; Gross/Doerr 2009; Krcmar 2010, 143-149; Berkovich et al. 2011c)).

Zur Konkretisierung von Anforderungen kann ebenso die Technik „*Quality Function Deployment*“ (QFD) eingesetzt werden. QFD ist eine weit verbreitete Methode aus dem Bereich der Entwicklung technischer Produkte, die eingesetzt wird, um Kundenanforderungen in Eigenschaften des technischen Produktes zu übersetzen (Akao 1990; An et al. 2008; Chahadi et al. 2007; Hauser/Clausung 1988; Möhrle/Spilgies 2005). Das Hauptelement von QFD ist eine Matrix, genannt House-of-Quality. In den Zeilen dieser Matrix werden die Kundenanforderungen eingetragen, in den Spalten werden die Produkt-Eigenschaften eingetragen. In den Feldern der Matrix wird dann festgehalten, welchen Einfluss die Produkt-Eigenschaften auf die Kundenanforderungen haben. Zusätzlich können in der Matrix auch die Gewichtung der Kundenwünsche festgehalten und über Zeilen- und Spaltensummen, die daraus abgeleitete Wichtigkeit der Produkt-Eigenschaften errechnet werden. In der von Möhrle (2005) durchgeführten Erweiterung von QFD für PSS wird in den Zeilen- und Spalten der Matrix ein eigener Bereich für die Dienstleistungswünsche und Dienstleistungseigenschaften eingeführt. Dadurch wird es ermöglicht, systematisch zu ermitteln, wie sich Produkt- und Dienstleistungseigenschaften gegenseitig beeinflussen. QFD bietet also eine Hilfe bei der gemeinsamen Detaillierung von Anforderungen an Dienstleistungen und Produkten.

Die Techniken zur Validierung von Anforderungen werden in Kapitel 7.3.4.2 beschrieben, da die Validierung als letzter Schritt vor der Übergabe der Anforderungen an die Entwicklung durchgeführt wird.

7.3.2.5 Techniken zur Anforderungsverfolgung

Um die Anforderungsverfolgung zu gewährleisten, können textuelle Referenzen, Hyperlinks und Traceability-Matrizen eingesetzt werden (Berkovich et al. 2011c; Pohl 2007, 513ff.; Kotonya/Sommerville 1998, 128-134).

Die *Traceability-Matrizen* sind ähnlich aufgebaut wie die Konsistenzmatrizen (s. Kapitel 7.3.2.3). Die Anforderungen, die in den Zeilen stehen, sind aus den Anforderungen, die in den Spalten enthalten sind, abgeleitet.

In der Dokumentation der Anforderungen werden die Vorgängeranforderungen, aus denen die Anforderungen abgeleitet wurden, und die Nachfolgeranforderungen, die aus den vorliegenden Anforderungen abgeleitet wurden, durch *textuelle Referenzen* angeben. Die so angegebenen Nachvollziehbarkeitsinformationen können in Form eines Graphen visualisiert werden. Die Referenzen können auch in Form von *Hyperlinks* dargestellt werden, so dass die Vorgängeranforderungen auf die Nachfolgeranforderungen verweisen.

7.3.3 Ermittlung und Analyse von Design-Anforderungen

Basierend auf den PSS-Anforderungen aus der Systemebene werden die Design-Anforderungen (Eigenschaftsebene) an materielle und immaterielle Bestandteile des PSS definiert und beschrieben (s. Kapitel 6.3.3). Da die Erstellung des Systementwurfs und der Funktionsstrukturen im Rahmen des Entwicklungsprozesses und nicht im Rahmen des Requirements Engineering erfolgen, werden die dazu erforderlichen Techniken nicht betrachtet.

7.3.3.1 Aktivitäten zur Ermittlung und Analyse von Design-Anforderungen

Folgende Aktivitäten werden zur Ermittlung und Analyse von Design-Anforderungen durchgeführt:

1. Ermittlung der Systemgrenze,
2. Erstellung des Systementwurfs,
3. Wechselseitige Anforderungskonkretisierung und Funktionsverfeinerung. Diese Aktivität ist insbesondere für die iterative und gegenseitig abgestimmte Konkretisierung von Anforderungen und Entwicklungsinformationen wichtig. Daher wird das Vorgehen hierzu in den Unterkapiteln 7.3.3.1.1, 7.3.3.1.2 und 7.3.3.1.3 beschrieben. Die Anforderungskonkretisierung in dieser Aktivität schließt die Ermittlung von Design-Anforderungen mit ein,
4. Identifikation und Auflösung von Konflikten zwischen den Design-Anforderungen (Anforderungsvereinbarung),
5. Identifikation und Elimination redundanter Design-Anforderungen,
6. Konkretisierung, Validierung und Dokumentation von Design-Anforderungen.

Abbildung 7-6 zeigt die oben beschriebenen Aktivitäten.

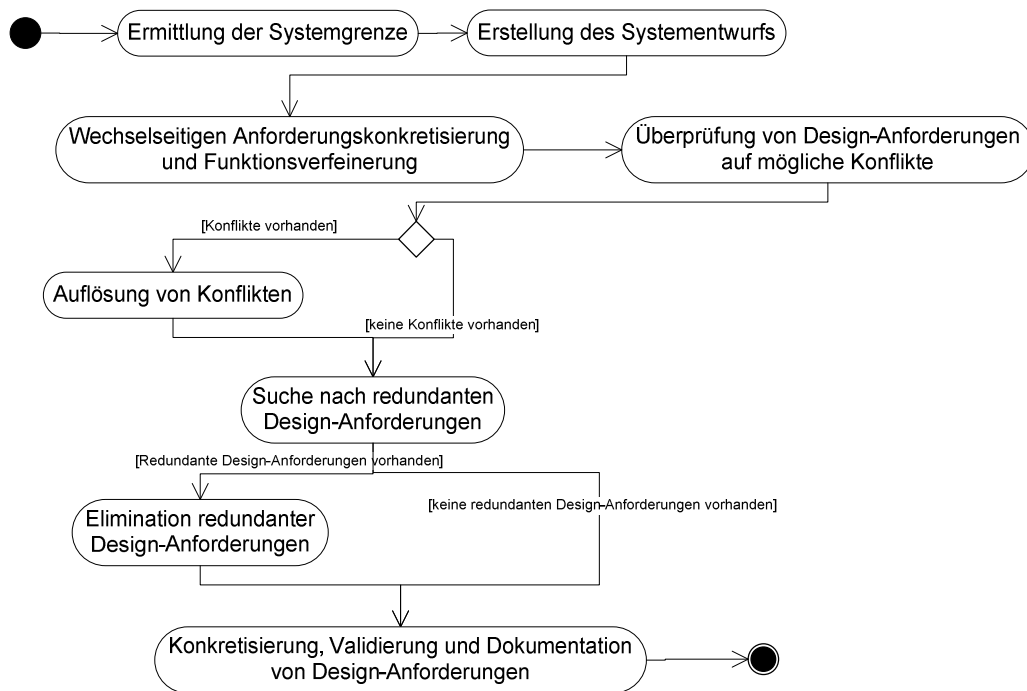


Abbildung 7-6: Aktivitäten der Ermittlung und Analyse von Design-Anforderungen
Quelle: Eigene Darstellung

Als Input zur Durchführung der oben genannten Aktivitäten dienen die PSS-Anforderungen der Systemebene. Der Output der Durchführung der Aktivitäten stellen der Systementwurf und die Design-Anforderungen der Eigenschaftsebene, dar.

Zur Identifikation der Systemgrenze und dem Separieren des Systems von seiner Umgebung empfiehlt Pohl (2007, 59) im ersten Schritt die Objekte zu identifizieren, die sich innerhalb der Systemgrenze befinden. Anschließend werden die Objekte identifiziert, die außerhalb der Systemgrenze liegen. Daraus lässt sich die Struktur des Systems, d. h. des gesamten PSS ableiten, das sich aus Objekten und zwischen ihnen vorhandenen Relationen zusammensetzt. Das so definierte System und die Systemgrenze werden den relevanten Stakeholdern präsentiert und mit ihnen diskutiert. Die Systemgrenze soll in bestimmten Intervallen auf Korrektheit überprüft werden. Wenn sich die Systemgrenze verschiebt, soll das in den Entwicklungsprozess einbezogen werden.

Anschließend wird der Systementwurf erstellt, der sich aus der Systemgrenze, dem Systemkontext und der Struktur des PSS zusammensetzt. Die Definition der Funktionalitäten der materiellen und immateriellen Bestandteile des PSS, die ebenso Teile des Systementwurfs sind, findet im Rahmen der iterativen und wechselseitigen Konkretisierung von Anforderungen und der damit in Verbindung stehenden Verfeinerung von Funktionen statt.

Das Vorgehen zur iterativen und wechselseitigen Konkretisierung von Anforderungen und Verfeinerung von Funktionen für PSS konkretisiert die Design-Anforderungen über die Funktionsstruktur-Anforderungen bis zu den Domänen-Anforderungen. Dabei werden die Entwicklungsinformationen, die während des Entwicklungsprozesses von PSS (s. Kapitel 3.3.1) entstehen, in die Konkretisierung von Anforderungen einbezogen. Ebenso wird darauf eingegangen, wie die zur Konkretisierung von Anforderungen relevanten Entwicklungsinformationen entstehen. Das Vorgehen besteht aus vier Phasen, die in den folgenden Kapiteln 7.3.3.1.2, 7.3.3.1.3, 7.3.4.1.1, 7.3.4.1.2 beschrieben werden.

Das *Vorgehen zur iterativen und wechselseitigen Konkretisierung von Anforderungen und Verfeinerung von Funktionen für PSS* basiert auf dem Vorgehen von Kortler et al. (2010), das die Vereinfachung des durch Iterationen geprägten Prozesses der stetigen Anforderungskonkretisierung und Funktionsverfeinerung durch eine Analyse der Zuordnung von Anforderungen zu Funktionen realisiert. Unter der Verfeinerung von Funktionen versteht man eine Dekomposition von abstrakten zu konkreten Funktionen (Buede 2009). Das Vorgehen von Kortler et al. (2010) wurde für innovative technische Produkte vorgeschlagen. Es beschreibt eine iterative, parallele Detaillierung von Anforderungen und Funktionen. Nach jeder Iteration werden durch matrixbasierte Analysen Funktionen mit besonders hohem Dekompositionsbedarf und Anforderungen mit besonders hohem Konkretisierungsbedarf identifiziert und verfeinert bzw. konkretisiert. Durch das abwechselnde Einnehmen der Anforderungs- und Funktionsperspektive werden die Anforderungen konkretisiert und die Funktionen hinreichend zerlegt, damit sie in ihrer Gesamtheit die Architektur des zu entwickelnden Produktes ergeben. Das Ergebnis des Vorgehens ist eine hierarchische Gliederung von konkretisierten Anforderungen und Funktionen, die an die Entwicklung übergeben werden können. Das Prinzip von Kortler et al. (2010) ist allgemein gehalten und geht nicht auf eine konkrete Ausgestaltung des Produktes ein. Das Prinzip stellt eine Sammlung an Aktivitäten bereit, um die Anforderungen den Funktionen zuzuordnen, die Funktionen zu verfeinern und die Anforderungen zu konkretisieren. Dieses Prinzip wurde für PSS angepasst, indem die Spezifika von PSS integriert wurden. Dabei wird zum einen zwischen Anforderungen an Sachleistungen und Dienstleistungen unterschieden und zum anderen eine abgestimmte und parallele Entwicklung der Anforderungen und der Architektur des PSS unterstützt, um dadurch eine nahtlose Einbindung des Requirements Engineering in den Entwicklungsprozess zu schaffen.

7.3.3.1.1 Prinzip der wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung

In diesem Kapitel wird das allgemeine Prinzip der wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung erläutert (Kortler et al. 2010). Anschließend wird in den Kapiteln 7.3.3.1.2 und 7.3.3.1.3 das an PSS angepasste Vorgehen beschrieben. Aufgrund der iterativen Durchführung der einzelnen Schritte des Verfahrens wird anhand einer Iteration gezeigt, wie das Prinzip der wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung abläuft. Abbildung 7-7 stellt das Vorgehen dar (Kortler et al. 2010).

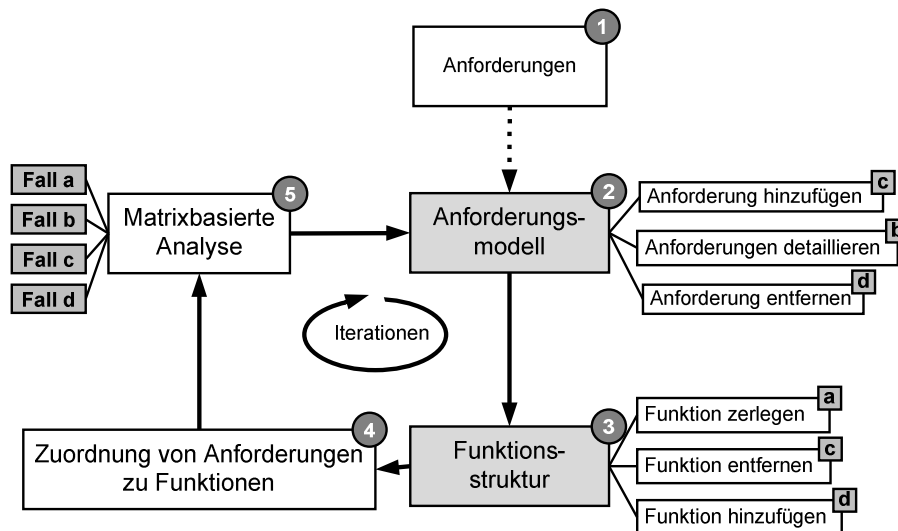


Abbildung 7-7: Prinzip der wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung

Quelle: (Kortler et al. 2010)

Folgende Schritte werden dabei ausgeführt (Kortler et al. 2010):

Schritt 1: Als Input für das Anforderungsmodell dienen die Anforderungen, die für die erste Iteration zu ermitteln sind.

Schritt 2: Im zweiten Schritt wird ein *Anforderungsmodell* erstellt, das die Anforderungen und die Beziehungen zwischen ihnen hierarchisch präsentiert (Ponn/Lindemann 2008, 31ff.). Um ein Anforderungsmodell zu erstellen, werden die Anforderungen konkretisiert, d. h. sie werden detaillierter dargestellt und beschreiben weitere konkretere Eigenschaften des zu entwickelnden Systems (Fall b), und, falls erforderlich, werden neue Anforderungen hinzugefügt (Fall c) oder bestehende Anforderungen entfernt (Fall d). In der ersten Iteration findet die Konkretisierung bei Bedarf statt.

Schritt 3: Im dritten Schritt wird eine *Funktionsstruktur* erstellt. Eine Funktionsstruktur wird definiert als eine logische Architektur, die angibt, welche Funktionen das System erbringen soll und welche Eingabeinformationen es in Ausgabeinformationen transformiert. Sie entspricht der Funktionsstruktur, die während des Entwicklungsprozess von PSS in der Phase der Produktkonzeption entsteht (s. Kapitel 6.3.3.1).

Funktionsstrukturen stellen funktionelle Zusammenhänge in Form einer Hierarchie von Funktionen dar (Ponn/Lindemann 2008, 239f.; Buede 2009, 175ff.; Lindemann 2006, 117-123; Ehrlenspiel 2002, 390-400). Eine Funktionsstruktur entsteht, wenn die *Gesamtfunktion*, die die gesamte Leistung des zu entwickelnden Systems präsentiert, in weitere *Teilfunktionen* zerlegt wird (Ehrlenspiel 2002, 390-391). Die Betrachtung einzelner Teilfunktionen bietet die Möglichkeit, nur Teilausschnitte des Systems zu analysieren. Um eine vollständige Funktionsstruktur zu erstellen, können Funktionen verfeinert, d. h. in weitere Teilfunktionen zerlegt (Fall a), entfernt (Fall c), sowie hinzugefügt (Fall d) werden. Eine Funktion bündelt dabei die Anforderungen, die logisch zu einer „Aufgabe“ gehören (Pohl 2007). Zum genauen Identifizieren und Formulieren von Funktionen sei auf weitere Quellen verwiesen (vgl. (Ramaswamy 1996, 125ff.; Ehrlenspiel 2002, 390-401; Spath/Demuß 2003)). In der Anforderungsanalyse für PSS werden die relationsorientierte Funktionsmodellierung, die die

kausalen Abhängigkeiten zwischen den Funktionen beschreibt, und die umsatzorientierte Funktionsmodellierung, die die Eingangs- und Ausgangsgrößen für die Funktionen beschreibt, vereint (s. Kapitel 6.3.3.4). In der nachfolgenden Iteration werden die Funktionen in ihre Teilfunktionen zerlegt.

Schritt 4: Der vierte Schritt beinhaltet die Zuordnung von Anforderungen zu den Funktionen, welche diese Anforderungen umsetzen. Die Zuordnung erfolgt basierend auf der Erfahrung der Entwickler durch Interviews und Workshops (Lindemann 2006, 115ff.). Die Unterstützung des Verfahrens der Zuordnung von Anforderungen zu den Funktionen erfolgt durch gezieltes Fragen, ob die Realisierung einer Funktion von einer Anforderung abhängt. Hierbei können für gewöhnlich in den ersten Iterationen keine 1:1 Zuordnungen gemacht werden. Vielmehr befinden sich die Funktionsstruktur, sowie das Anforderungsmodell in einer derart groben Phase, dass sowohl jede Anforderung nur durch mehrere Funktionen erfüllt wird, als auch jede Funktion bei der Erfüllung mehrerer Anforderungen beteiligt ist. Für die Zuordnung von Anforderungen zu den Funktionen können matrixbasierte Ansätze verwendet werden, die zur Modellierung und Analyse von Systemen eingesetzt werden (Steward 1981). Eine Design Structure Matrix (*DSM*) ist eine quadratische Matrix, die die Möglichkeit bietet, Elemente und ihre Relationsarten aufeinander abzubilden (Guenov/Barker 2004; Lindemann et al. 2008, 55f.; Lai/Gershenson 2009; Lee 2008, 205ff.). Typische Elemente können Produktkomponenten, Funktionen, Anforderungen, Produkteigenschaften oder Prozessschritte sein. Für die Zuordnung wird eine Domain-Mapping-Matrix (*DMM*) verwendet, bei welcher Anforderungen und Funktionen zeilen- bzw. spaltenweise aufgetragen sind (Abbildung 7-8). DMM ist eine Weiterentwicklung von DSM, die Abhängigkeiten zwischen verschiedenen Elementen binär (entweder hängen zwei Elemente zusammen oder nicht) darstellt (Lindemann et al. 2008, 54). Die DMM ermöglicht es, die Relationen und Abhängigkeiten zwischen verschiedenen vernetzten Domänen zu identifizieren, und in quantitativer Form zu modellieren (Danilovic/Sandkull 2005). Einträge in der betreffenden Zelle der Matrix stellen eine Relation zwischen den beiden Elementen dar. Eine 1 in Zelle (x,y) signalisiert, dass eine Funktion in Zeile x bei der Erfüllung von der Anforderung in Zeile y beteiligt ist.

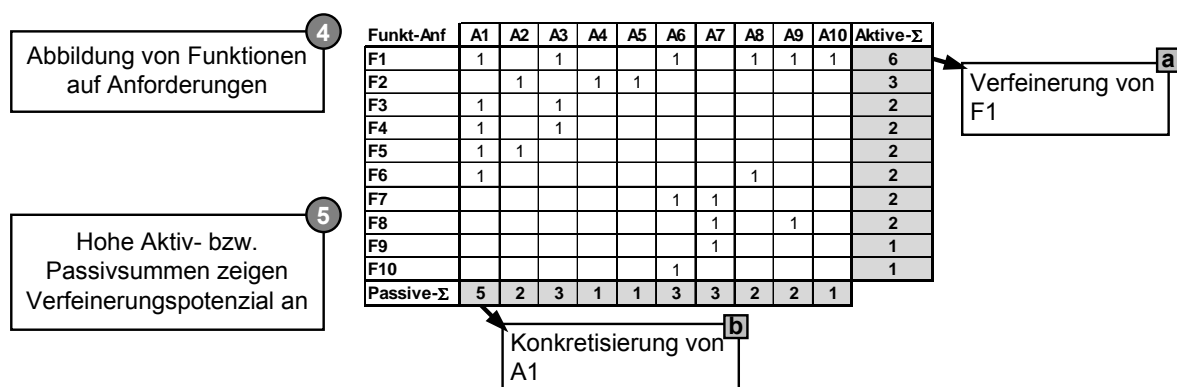


Abbildung 7-8: Zuordnung von Anforderungen zu den Funktionen

Quelle: (Kortler et al. 2010)

Schritt 5: Im 5.ten Schritt findet die matrixbasierte Analyse statt, die entscheidet, welche Anforderungen detailliert/hinzugefügt/entfernt werden sollen und welche Funktionen zerlegt/hinzugefügt/entfernt werden sollen. Dafür werden die Spaltensummen für Anforderungen (*Passivsummen*) und Zeilensummen für Funktionen (*Aktivsummen*) berechnet.

Die Passivsumme gibt an, zu wie vielen Funktionen eine Anforderung zugeordnet werden kann. Die Aktivsumme gibt an, wie viele Anforderungen einer Funktion zugeordnet werden können. Darauf basierend konnten folgende Schlussfolgerungen getroffen werden. Je höher die Anzahl der Anforderungen ist, welche eine Funktionen erfüllt (Aktivsumme), desto abstrakter ist die Funktion formuliert worden (Fall a). Somit indiziert die Anzahl von Anforderungen A, welche Funktion F zugeordnet ist, das Verfeinerungspotenzial der Funktion F. Im Umkehrschluss konnte festgestellt werden, dass die Anzahl an Funktionen, welche benötigt werden, um eine Anforderung zu erfüllen (Passivsumme), ein Indiz für das Konkretisierungspotenzial der Anforderung A darstellt (Fall b). Neben Fall a und Fall b können nach der Analyse weitere Fälle auftreten. Werden Funktionen mit einer Aktivsumme=0 identifiziert, so weisen diese auf fehlende Anforderungen bzw. nutzlose Funktionen hin (Fall c). Anforderungen mit Passivsumme=0 zeigen an, dass die nötige Funktion im Funktionsmodell noch nicht vorhanden und somit noch zu erstellen ist (Fall d). Andernfalls wird die entsprechende Anforderung verfehlt. Während Fall c und Fall d eine eindeutige Handlungsanweisung nach sich ziehen, geben die Verfeinerungspotenziale der Funktionen und Konkretisierungspotenziale der Anforderungen lediglich einen Hinweis darauf, welche Funktionen und Anforderungen innerhalb der nächsten Iteration verfeinert bzw. konkretisiert werden sollten. Hierbei soll darauf geachtet werden, im Anforderungsmodell und in der Funktionsstruktur durch gezielte Konkretisierung bzw. Verfeinerung ein homogenes Level im Grad der Abstraktheit der Funktionen und Anforderungen untereinander zu halten. Hierdurch wird ein stetiges Verfeinern der beiden Modelle erreicht. So ist es möglich die Iterationen zwischen Funktions- und Anforderungsmodellierung zu steuern.

Das Vorgehen zur iterativen und wechselseitigen Konkretisierung von Anforderungen und Verfeinerung von Funktionen für PSS, im Rahmen dessen die Design-Anforderungen der Eigenschaftsebene über die Funktionsstruktur-Anforderungen der Funktionsebene bis zu den Domänen-Anforderungen der Komponentenebene konkretisiert werden, besteht aus mehreren Phasen (Berkovich et al. 2011b). Die einzelnen Schritte des Vorgehens von Kortler et al. (2010) werden in den einzelnen Phasen verwendet. Die Beschreibung der Vorgehens ist unter (Berkovich et al. 2011b)¹⁸ zu finden.

7.3.3.1.2 Phase 1: Aufbau der Funktionsstruktur

Das Verfahren zur iterativen Anforderungsanalyse an PSS besteht aus vier Phasen (Berkovich et al. 2011b). Die ersten zwei Phasen, in denen das PSS in materielle und immaterielle Bestandteile zerlegt wird und darauf basierend die Anforderungen konkretisiert werden, finden im Rahmen der Ermittlung und Analyse der Design-Anforderungen (Eigenschaftsebene) statt. Die zwei Phasen zur Erstellung von Funktionsstrukturen im Funktionsentwurf und der Ermittlung von Funktionsstruktur-Anforderungen (Funktionsebene) sowie zum Aufbau des Grobentwurfs und der Konkretisierung von Anforderungen zu Domänen-Anforderungen (Komponentenebene) finden im Rahmen Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen statt (s. Kapitel 7.3.4).

Um eine Zerlegung des PSS in materielle und immaterielle Bestandteile vorzunehmen (s. Kapitel 6.3.3.1), soll als erstes, basierend auf den ermittelten PSS-Anforderungen der Systemebene, die Gesamtfunktion des PSS geklärt werden (Berkovich et al. 2011b). Die so

¹⁸ submitted

ermittelte Gesamtfunktion des PSS kann in weitere Funktionen (genannt *Hauptfunktionen*) zerlegt werden, ohne auf die genaue Ausgestaltung und die Verbindungen zwischen den Hauptfunktionen einzugehen. Die Hauptfunktionen beschreiben die primären Funktionen, die das PSS erfüllt. Sie können später zur Modulbildung verwendet werden (s. Kapitel 3.1.2). Dazu kann die SCORE-Methode eingesetzt werden (Böhm et al. 2008), die das PSS in kundenindividuelle oder standardisierte Module aufteilt.

Das Ziel der ersten Phase ist das Erstellen einer Funktionsstruktur, die angibt, welche Funktionen durch eine Sachleistung und welche durch eine Dienstleistung realisiert werden können. Abbildung 7-9 stellt die einzelnen Schritte beim Aufbau der Funktionsstruktur dar.

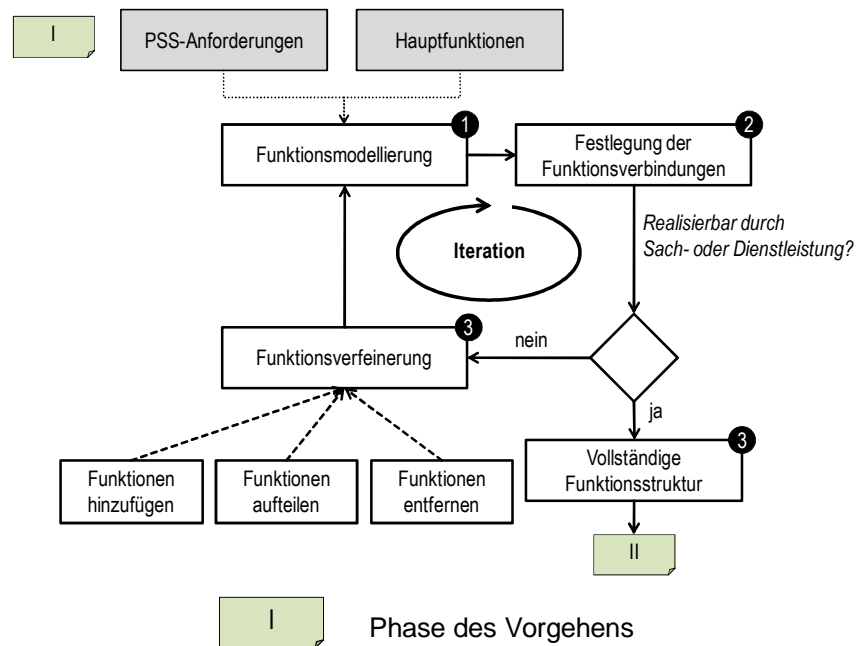


Abbildung 7-9: Phase 1: Aufbau der Funktionsstruktur

Quelle: (Berkovich et al. 2011b)

Die PSS-Anforderungen und die Hauptfunktionen sind die Eingangsgrößen beim Aufbau der Funktionsstruktur. Während der Iterationen werden die Hauptfunktionen hierarchisch in Funktionen zerlegt und ergeben zum Schluss eine vollständige Funktionsstruktur. Während der Funktionsmodellierung (1) werden alle Funktionen aufgestellt und graphenbasiert repräsentiert. Um die kausalen Abhängigkeiten zwischen den Funktionen zu identifizieren und festzustellen, welche Funktionen zur Durchführung anderer Funktionen benötigt werden (2), wird eine DSM eingesetzt. Falls zwischen den Funktionen kausale Abhängigkeiten bestehen, werden sie in der Matrix angegeben, um die optimale Reihenfolge der Funktionen zu bestimmen.

Die initialen PSS-Anforderungen sowie die bestehende Funktionsstruktur geben der Entwicklung die Funktionalitäten der Sach- und Dienstleistungen vor, ohne auf die genauen Komponenten und Ausgestaltung der materiellen und immateriellen Bestandteile einzugehen. Darauf basierend soll für jede Funktion beschlossen werden, ob sie eindeutig durch eine Sach- oder Dienstleistung realisiert wird. Bei der Identifikation von Dienstleistungen sollen die Potenzial-, Prozess- und Ergebnisdimensionen beachtet werden. Die Ressourcen, die zur Erbringung der Dienstleistungen benötigt werden, sind nicht mit den Sachleistungskomponenten des PSS zu vermischen, die im Zentrum der PSS-Entwicklung

stehen. In diesem Fall entsteht eine vollständige Funktionsstruktur, die die Funktionalität des PSS beschreibt, ohne auf die genaue Ausgestaltung einzugehen, und die Funktionen gemäß ihrer Realisierung nach Sach- und Dienstleistungen unterteilt (3). Falls von der Entwicklung nicht eindeutig beschlossen werden kann, ob eine Funktion durch die Sach- oder Dienstleistungen realisiert wird, soll die Funktion der Funktionsstruktur verfeinert werden, indem sie in weitere Funktionen zerlegt wird oder neue Funktionen der Funktionsstruktur hinzugefügt oder vorhandene Funktionen entfernt werden (3). Die Funktionsaufteilung bedeutet in diesem Fall keine hierarchische Zerlegung einer Funktion sondern das Ersetzen der vorhandenen Funktion durch detailliertere Funktionen, die den Sinn der zu zerlegenden Funktion konkreter wiedergeben. Für die so gewonnene Funktionsstruktur werden die Schritte 1 bis 3 wiederholt, bis eine vollständige Funktionsstruktur besteht.

7.3.3.1.3 Phase 2: Aufteilung von Anforderungen nach materiellen und immateriellen Bestandteilen des PSS

Anschließend werden die PSS-Anforderungen der Eigenschaftsebene nach materiellen und immateriellen Komponenten aufgeteilt (s. Kapitel 6.3.3.2). Abbildung 7-10 stellt die einzelnen Schritte des Vorgehens dar (Berkovich et al. 2011b).

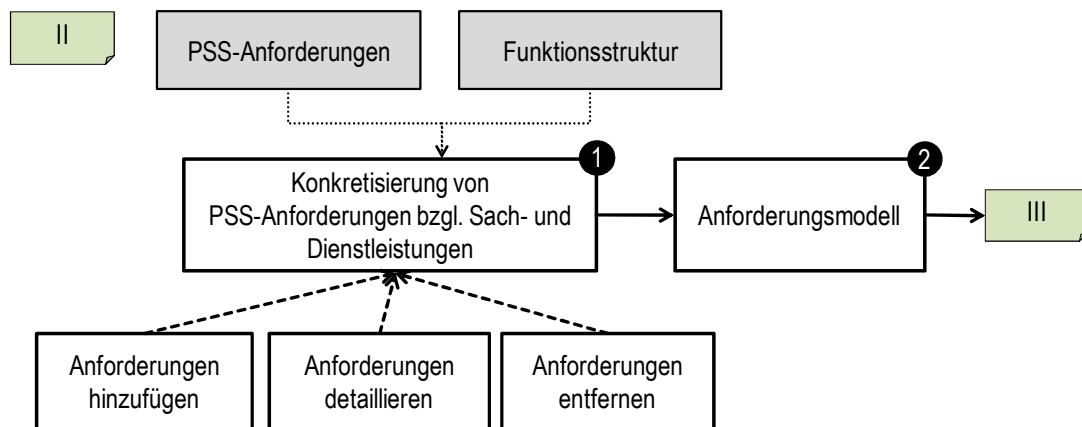


Abbildung 7-10: Phase 2: Aufteilung von Anforderungen nach materiellen und immateriellen Leistungsbündeln

Quelle: (Berkovich et al. 2011b)

Die Konkretisierung kann zum Beispiel durch Interviews, Use Cases, Szenarien oder formale Modelle unterstützt werden (Nuseibeh/Easterbrook 2000). Somit steht ein Anforderungsmodell mit konkretisierten Design-Anforderungen als Ergebnis von Phase 2 bereit (2).

Unter Einbeziehung der Informationen aus der Funktionsstruktur, die in Phase 1 erstellt wurde, wird ein Anforderungsmodell aufgebaut, indem die PSS-Anforderungen konkretisiert werden (1). Das bedeutet, die Information, ob eine Funktion als Sach- oder Dienstleistung realisiert wird, geht in die Anforderung mit ein, wodurch diese konkreter beschrieben wird. Ähnlich wie in Schritt 2 von Kapitel 7.3.3.1.1 beschrieben, werden die Anforderungen bei der Konkretisierung detailliert, entfernt oder durch neue Anforderungen ergänzt. Es entsteht ein Anforderungsmodell (2), das aus den Design-Anforderungen (Eigenschaftsebene) besteht.

Wenn die konkretisierten Design-Anforderungen gegliedert nach Entwicklungsdimensionen an Dienstleistungen vorliegen, werden für die Anforderungen, die quantifizierbar sind, Ziel-

werte bestimmt (Husen 2007, 99-101). Dafür kann ein indirektes Verfahren nach Ramaswamy (1996, 51ff.) eingesetzt werden. In diesem Fall werden erst die Leistungs-/Zufriedenheitsfunktionen bestimmt, die die Kundenzufriedenheit abhängig von der Ausprägung der Leistung definieren (Husen 2007, 99-101; Ramaswamy 1996, 51ff.). Anschließend wird die Zufriedenheitsdimension der Design-Anforderungen bestimmt. Zuletzt werden die Zielwerte und Toleranzen für die Anforderungen an die Dienstleistungen bestimmt.

Wenn die Design-Anforderungen feststehen, werden die Anforderungen auf mögliche Konflikte überprüft. Die identifizierten Konflikte werden aufgelöst. Anschließend wird nach redundanten Anforderungen gesucht. Die identifizierten redundanten Anforderungen werden eliminiert. Die so gewonnenen Anforderungen werden konkretisiert und entsprechend den Spezifikationsvorschriften dokumentiert.

7.3.3.2 Techniken zur Ermittlung und Analyse von Design-Anforderungen

Zur Erstellung des Systementwurfs wird ein Kontextdiagramm verwendet, das die Umgebung des PSS und das PSS modelliert. Dazu zeigt es das System und alle mit ihm in Verbindung stehenden externen Objekte auf (Hull et al. 2004, 48-55; Rupp 2004, 190-193). Kontextdiagramme stellen ihrerseits Spezifikationsartefakte der Eigenschaftsebene dar (s. Kapitel 6.3.3.4).

Zur Ermittlung von Design-Anforderungen im Rahmen der iterativen und wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung werden die Techniken der Ermittlung von PSS-Anforderungen (s. Kapitel 7.3.2.2) verwendet.

Zur Konkretisierung von Design-Anforderungen können die Techniken, die in Kapitel 7.3.2.4 beschrieben sind, eingesetzt werden. In diesem Fall werden nicht die Anforderungen an das PSS, sondern die Anforderungen an das technische Produkt und an die Dienstleistungen dargestellt.

Zur Konkretisierung von Design-Anforderungen kann die Technik „QFD“ eingesetzt werden (s. Kapitel 7.3.2.4). QFD hat den Vorteil, dass es ein strukturiertes Verfahren anbietet, wie Kundenanforderungen zu Produkteigenschaften detailliert werden können. Eine der Schwächen von QFD ist jedoch der große Aufwand, der benötigt wird, um die Technik anzuwenden. Zudem müssen bereits vor der Anwendung von QFD die Anforderungen ermittelt worden sein. Die Brauchbarkeit des Ergebnisses der QFD hängt dabei maßgeblich von den vorher ermittelten Anforderungen ab (Zacharias 2003). Ein weiterer Nachteil von QFD ist auch, dass die Produkteigenschaften im Rahmen der QFD ermittelt werden müssen. Dafür greift QFD vor allem auf Vorgängerprodukte zurück, die bei Neuentwicklungen nicht vorhanden sind. Die Methode QFD, die für technische Produkte und PSS entwickelt wurde (s. Kapitel 7.3.2.4) wurde von Ramaswamy (1996) und anschließend Husen (2007) für Dienstleistungen erweitert. Dabei werden drei Matrizen erstellt, wobei für jede der drei Dimensionen von Dienstleistungen eine Matrix angewendet wird. Auf diese Weise können die Anforderungen an Ergebnisse, an die Prozesse und letztendlich an die zugrunde liegenden Potenziale sukzessiv detailliert werden. Für die QFD für Dienstleistungen treffen dieselben Vor- und Nachteile zu, wie für technische Produkte. Jedoch verschärft sich das Problem, dass die Produkteigenschaften nur schwer ermittelbar sind, da bei Dienstleistungen nicht auf ein Vorgängerprodukt zurückgegriffen werden kann (Husen 2007, 103ff.). Zudem setzt QFD auch voraus, dass Dienstleistungseigenschaften von den Requirements Analysten und Entwicklern gefunden werden müssen. Als konkrete Vorgehensweise dazu wird jedoch nur Brainstorming vorgeschlagen. In der

Praxis stellt jedoch vor allem das Finden von Dienstleistungsanforderungen, die oft implizit geäußert werden, und von Dienstleistungseigenschaften eine große Herausforderung dar.

Zur Gewährleistung der Anforderungsverfolgung in der Eigenschaftsebene sollen dieselben Techniken wie in der Systemebene eingesetzt werden, die in Kapitel 7.3.2.5 beschrieben wurden. Zur Identifikation und Auflösung von Konflikten werden die in Kapitel 7.3.2.3 beschriebenen Techniken eingesetzt. Die Techniken zur Validierung von Anforderungen werden in Kapitel 7.3.4.2 beschrieben.

7.3.4 Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen

Basierend auf den Design-Anforderungen und dem Systementwurf der Eigenschaftsebene, werden die Funktionsstrukturen und die Funktionsstrukturanforderungen ermittelt. Diese beiden Elemente stellen die Funktionsebene des Artefaktmodells dar. Anschließend werden aus den Funktionsstrukturen und den Funktionsstrukturanforderungen der Grobentwurf und die Domänenanforderungen der Komponentenebene abgeleitet. Da die Erstellung des Funktionsstrukturentwurfs, der Funktionsstrukturen und des Grobentwurfs im Rahmen des Entwicklungsprozesses und nicht im Rahmen des Requirements Engineering erfolgen, werden die dazu erforderlichen Techniken nicht betrachtet.

7.3.4.1 Aktivitäten zur Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen

Folgende Aktivitäten werden im Rahmen der Konzeption von Funktionsstrukturentwurf, Ermittlung und Analyse von Funktionsstruktur-Anforderungen, Konzeption von Grobentwurf, Ermittlung und Analyse von Domänen-Anforderungen durchgeführt:

1. Ermittlung des Funktionsstrukturentwurfs,
2. Wechselseitige Anforderungskonkretisierung und Funktionsverfeinerung. Wie in der vorhergehenden Säule werden dieser Aktivität eigene Unterkapitel (s. Kapitel 7.3.4.1.1, Kapitel 7.3.4.1.2) gewidmet, in denen das Vorgehen im Detail erläutert wird. Diese Aktivität schließt die Ermittlung von Funktionsstruktur-Anforderungen mit ein,
3. Identifikation und Auflösung von Konflikten zwischen den Funktionsstruktur-Anforderungen (Anforderungsvereinbarung),
4. Identifikation und Elimination redundanter Funktionsstruktur-Anforderungen,
5. Konkretisierung und Dokumentation von Funktionsstruktur-Anforderungen,
6. Ermittlung des Grobentwurfs,
7. Konkretisierung und Aufteilung von Funktionsstruktur-Anforderungen nach Komponenten, Ermittlung von Domänen-Anforderungen,
8. Identifikation und Auflösung von Konflikten zwischen den Domänen-Anforderungen (Anforderungsvereinbarung),
9. Identifikation und Elimination redundanter Domänen-Anforderungen,
10. Konkretisierung und Dokumentation von Domänen-Anforderungen,
11. Validierung von Domänen-Anforderungen bzgl. ihrer Übereinstimmung mit den initialen Zielen und den initialen Anforderungen an das PSS.

Abbildung 7-11 zeigt die Aktivitäten zur Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen.

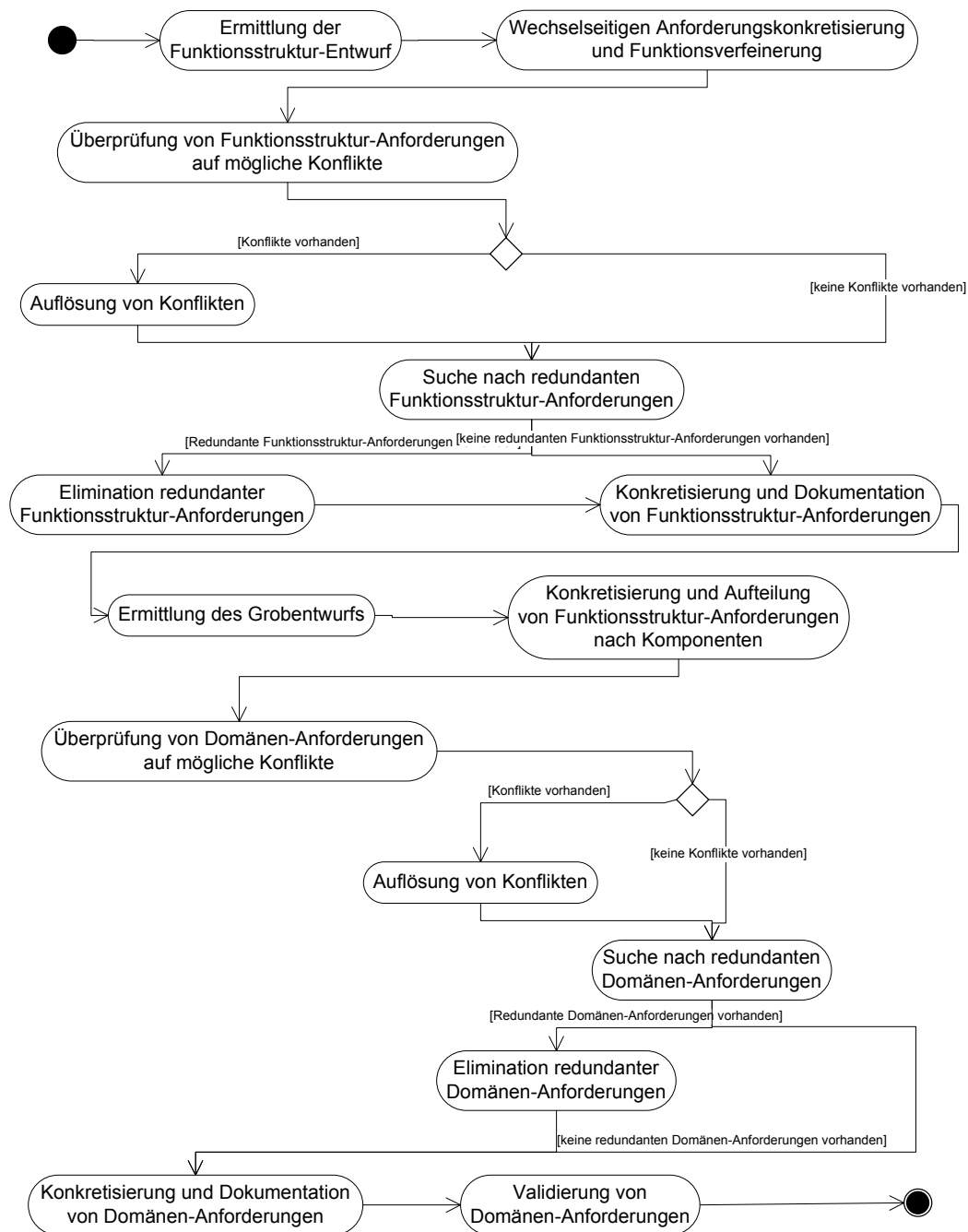


Abbildung 7-11: Aktivitäten zur Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen

Quelle: Eigene Darstellung

Die wechselseitige Konkretisierung und Funktionsverfeinerung (Berkovich et al. 2011b) basiert auf dem von Kortler et al. (2010) entwickelten Vorgehen (s. Kapitel 7.3.3.1.1) und stellt eine logische Fortsetzung des in Kapiteln 7.3.3.1.2 und 7.3.3.1.3 beschriebenen Vorgehens dar, indem das Vorgehen zur Erstellung von Entwicklungs- und Anforderungsartefakten auf der Funktionsstruktur-Ebene und der Komponentenebene angewendet wird. Um die Verwechslung zwischen dem Vorgehen im vorigen Kapitel und in diesem Kapitel zu verhindern, werden die Phasen in diesem Kapitel fortlaufend bzgl. der Phasen des Vorgehens in Kapiteln 7.3.3.1.2 und 7.3.3.1.3 nummeriert.

7.3.4.1.1 Phase 3: Iterative Konkretisierung von Anforderungen

In den Phasen 3 bis 5 findet eine iterative Konkretisierung von Design-Anforderungen, die durch die Konkretisierung in die Funktionsstruktur-Anforderungen münden, und Verfeinerung von Funktionen statt (Berkovich et al. 2011b). Das Ziel dabei ist, die Anforderungen so zu konkretisieren und die Funktionen zu verfeinern, dass darauf basierend eine Architektur für das PSS aufgebaut werden kann (s. Kapitel 6.3.5.1).

Das Anforderungsmodell aus Phase 2 (s. Kapitel 7.3.3.1.3) und die Funktionsstruktur aus Phase 1 (s. Kapitel 7.3.3.1.2) bilden die Grundlage zur iterativen Konkretisierung von Anforderungen in Phase 3. Abbildung 7-12 stellt die einzelnen Schritte der Phase 3 dar (Berkovich et al. 2011b). Im ersten Schritt werden die Anforderungen den Funktionen in Relation n:m zugeordnet (1). Darauf basierend wird eine DMM erstellt, die diese Relationen quantitativ darstellt. Für weitere Details sei auf Schritt 4 der wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung in Kapitel 7.3.3.1.1 verwiesen.

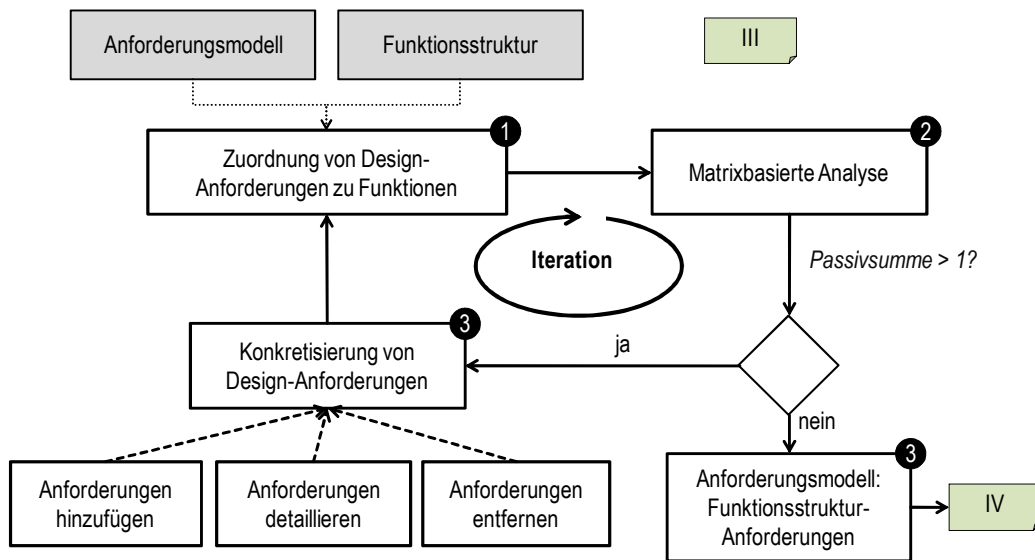


Abbildung 7-12: Phase 3: Konkretisierung von Anforderungen

Quelle: (Berkovich et al. 2011b)

Im nächsten Schritt findet die matrixbasierte Analyse statt, die die Passivsummen (*Spaltensummen von Anforderungen*) und Aktivsummen (*Zeilensummen von Funktionen*) berechnet (2). Falls eine Design-Anforderung mehreren Funktionen zugeordnet ist (Passivsumme einer Anforderung > 1), sind die Anforderungen zu konkretisieren (3), indem sie detailliert oder entfernt werden oder neue Anforderungen hinzugefügt bekommen (vgl. Schritt 5 des Kapitels 7.3.3.1.1). Anschließend werden die so konkretisierten Anforderungen erneut auf die bestehenden Funktionen abgebildet und die Schritte wiederholt. Die Anforderungen werden solange konkretisiert, bis die Passivsumme = 1 ist, was bedeutet, dass eine konkretisierte Design-Anforderung genau einer Funktion zugeordnet werden kann. Das Ergebnis der Phase ist ein Anforderungsmodell, das konkretisierte Anforderungen (Funktionsstruktur-Anforderungen) hierarchisch gliedert enthält. Eine Funktionsstruktur-Anforderung kann genau einer Funktion zugeordnet werden, während einer Funktion mehrere Funktionsstruktur-Anforderungen entsprechen können. Diese Anforderungen verlaufen vom Abstrakten zum Spezifischen und beziehen sich auf die einzelnen Funktionen des PSS (Berkovich et al. 2011b). Somit münden die Design-Anforderungen durch eine iterative

Konkretisierung in Funktionsstruktur-Anforderungen (die Erläuterung zu Funktionsstruktur-Anforderungen ist in Kapitel 6.3.4.2 zu finden).

Schematisch lässt sich das Ergebnis der Anforderungskonkretisierung so beschreiben (Berkovich et al. 2011b): Anforderung A kann den Funktionen F₁ und F₂ zugeordnet werden. Nach der iterativen Konkretisierung von A zu A₁ und A₂ wird A₁ der F₁ und A₂ der F₂ zugeordnet (Abbildung 7-13).

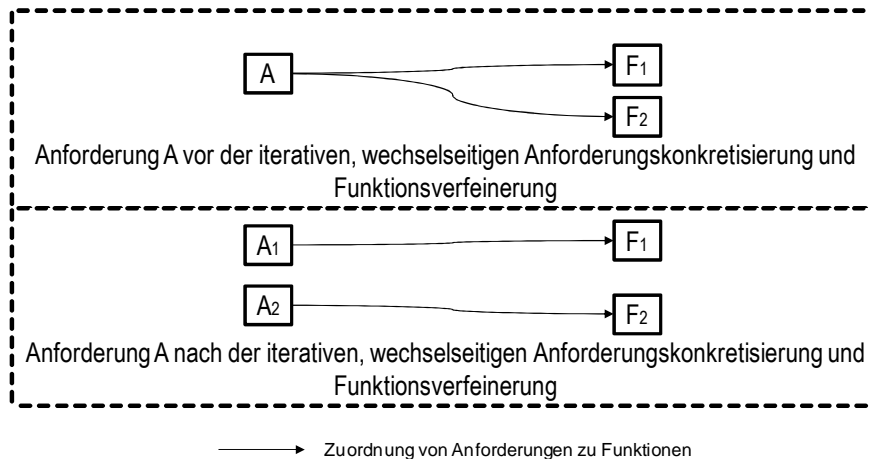


Abbildung 7-13: Iterative Zuordnung von Anforderungen zu Funktionen

Quelle: Eigene Darstellung

Die Verbindungen zwischen den Funktionen der Funktionsstruktur unterstützen die *Konfliktsuche* zwischen den Anforderungen. Wenn zwei Funktionen miteinander verbunden sind, kann es zwischen den Anforderungen, die diesen Funktionen zugeordnet sind, zu Widersprüchen kommen. Der Grad der Verbindung zwischen den Funktionen beeinflusst die Intensivität der möglichen Konflikte. Eine Strukturanalyse von Anforderungen (Eben/Lindemann 2010; Browning 2001) kann dafür verwendet werden. Dadurch können verschiedene Arten von Zusammenhängen zwischen konkretisierten Anforderungen identifiziert werden, wie bspw. gegenseitige Beeinflussung.

Wenn die Passivsumme einer Anforderung = 0 bedeutet dies, dass eine Anforderung überflüssig ist, d.h. es existiert keine Funktion, der die Anforderung zugeordnet wird. Wenn die Aktivsumme einer Funktion = 0 bedeutet dies, dass keine Anforderungen existieren, die der Funktion zugeordnet werden. Beide Fälle weisen darauf hin, dass Anforderungen falsch konkretisiert oder Funktionen falsch verfeinert wurden. So soll in beiden Fällen von der Entwicklung analysiert werden, ob die Vorgänger-Anforderungen erneut zu konkretisieren oder die Vorgänger-Funktionen erneut zu verfeinern sind.

7.3.4.1.2 Phase 4: Iterative Verfeinerung von Funktionen

Nachdem das Anforderungsmodell mit Funktionsstruktur-Anforderungen aus Phase 3 feststeht, sind die Funktionen iterativ zu verfeinern (Berkovich et al. 2011b). Abbildung 7-14 stellt die einzelnen Schritte des Vorgehens dar.

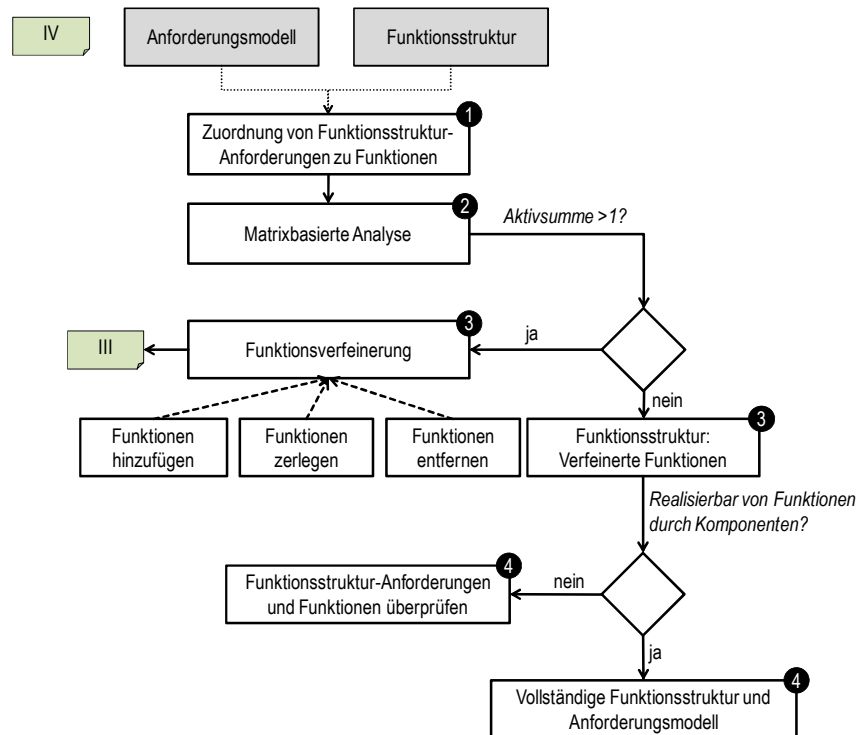


Abbildung 7-14: Phase 4: Iterative Verfeinerung von Funktionen

Quelle: (Berkovich et al. 2011b)

Dabei werden die in Phase 3 gewonnenen Anforderungen, die im Anforderungsmodell festgehalten werden, den bestehenden Funktionen zugeordnet (1), sowie eine matrixbasierte Analyse durchgeführt (2). Wenn einer Funktion mehrere Anforderungen zugeordnet werden können (Aktivsumme einer Funktion > 1), soll diese Funktion so verfeinert werden, damit ein einheitlicher Abstraktionsgrad der Funktionen und Anforderungen erreicht wird. Für die Details der Funktionsverfeinerung sei auf Schritt 3 (s. Kapitel 7.3.3.1.1) verwiesen. Durch die Funktionsverfeinerung entsteht eine Hierarchie von Funktionen, die die Funktionalitäten des PSS, angefangen bei abstrakten, bis hin zu immer spezifischeren beschreibt. Anschließend wird Phase 3 mit den verfeinerten Funktionen und dem bestehenden Anforderungsmodell mit konkretisierten Anforderungen wiederholt. Bei der Durchführung der Anforderungskonkretisierung in der erneut durchgeführten Phase 3 werden die Anforderungen den neu gewonnenen Funktionen zugeordnet.

Ein Beispiel soll schematisch veranschaulichen, wie die Anforderungskonkretisierung und Funktionsverfeinerung funktionieren (siehe Abbildung 7-15). Die konkretisierte Anforderung A ist der Funktion F zugeordnet, die hierarchisch in F_1 und F_2 verfeinert wird. Die Anforderung A wird daraufhin F_1 und F_2 zugeordnet und basierend auf den Ergebnissen der matrixbasierten Analyse in A_1 und A_2 konkretisiert, so dass schließlich A_1 zu F_1 und A_2 zu F_2 zugeordnet sind (Berkovich et al. 2011b).

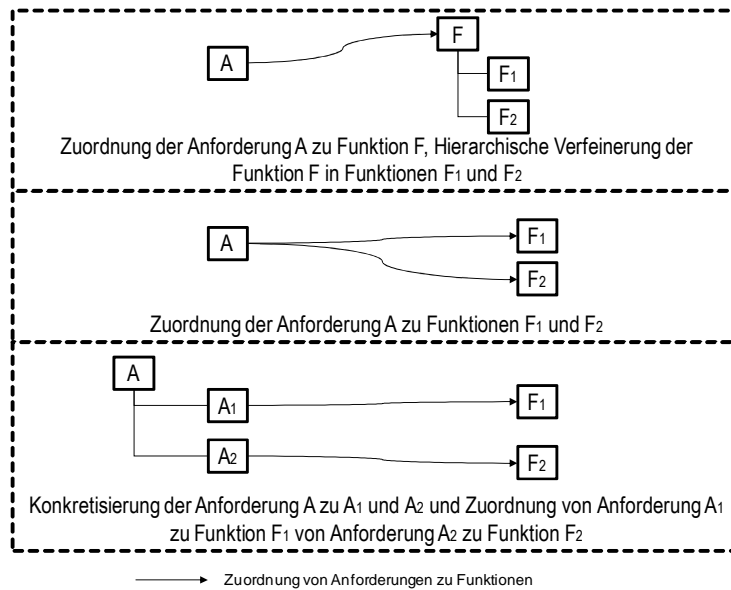


Abbildung 7-15: Schematische Darstellung der iterativen wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung

Quelle: Eigene Darstellung

Während dieser iterativen Verfeinerung der Funktionsstruktur und der anschließenden Anforderungskonkretisierung wird bei den Funktionen, die Sachleistungen beschreiben, die flussorientierte Funktionsmodellierung durchgeführt, bis schließlich ein Netzwerk entstanden ist, welches ausschließlich aus Energie-, Signal- und Materialflüssen besteht (Lindemann 2006, 116ff.). Im Falle von Dienstleistungen entsteht ein Netzwerk von Funktionen, die verfeinert werden, bis die Prozessaktivitäten für die Dienstleistungserbringung z.B. in Form von Blueprinting (Shostack 1984; Ramaswamy 1996, 125ff.) bestehen.

Wenn die Funktionsstruktur feststeht (3), soll geklärt werden, ob jede Funktion dieser Funktionsstruktur durch eine Komponente realisierbar ist. Eine Komponente ist eine abstrakte logische Einheit, die noch keine konkrete Hardware-, Software- oder Dienstleistungsumsetzung darstellt (Pohl 2007, 568-569). Es ist jedoch möglich die Komponenten direkt einer der jeweiligen Domänen zuzuordnen. Somit münden die iterative wechselseitige Anforderungskonkretisierung und Funktionsverfeinerung schließlich in einer groben Produktarchitektur des PSS, welche mögliche Komponenten und Serviceanteile darstellt und somit die Untergliederung in die beteiligten Domänen abbildet. Die Produktarchitektur wird als Grobentwurf bezeichnet (s. Kapitel 6.3.5). Zum Ableiten von Hardware- und Softwarekomponenten aus den Funktionen sei auf die Arbeiten von Schätz (2006) hingewiesen. Zur Ableitung von Dienstleistungskomponenten aus den Funktionen sei auf Ramaswamy (1996, 125ff.) hingewiesen.

Die Funktionsstruktur-Anforderungen werden auf Konflikte und redundante Anforderungen überprüft. Die bestehenden Konflikte sollen aufgelöst und redundante Anforderungen eliminiert werden.

Anschließend werden die Funktionsstruktur-Anforderungen der Funktionsebene bzgl. der definierten domänenspezifischen Komponenten des PSS konkretisiert. Die Anforderungen an die Hardware-Komponenten erhalten konkretere Eigenschaften in Anlehnung an die Hauptmerkmalliste (Tabelle 10-6). Die Anforderungen an die Software-Komponenten erhalten konkretere Eigenschaften für funktionale Anforderungen, indem auf die eigentlichen Funktionen

der Komponenten eingegangen wird, und die nicht-funktionalen Anforderungen werden gemäß den Kategorien für nicht-funktionale Anforderungen (Abbildung 10-1) kategorisiert. Die Anforderungen an die Dienstleistungen erhalten konkretere Eigenschaften bzgl. der Angaben zur Gestaltung des Dienstleistungserbringungsprozesses, der dafür benötigten Ressourcen und der erwarteten Ergebnisse der Dienstleistungserbringung. Die Domänen-Anforderungen werden konkretisiert und somit in die für die weitere Umsetzung benötigte Form gebracht, und dokumentiert.

Wenn die Anforderungen und der Grobentwurf vollständig definiert sind, müssen sie validiert werden. Die Validierung wird als eine Aktivität definiert, die sicherstellt, dass die analysierten konkreten Anforderungen den initialen Anforderungen der Stakeholder entsprechen (Nuseibeh/Easterbrook 2000). Somit soll durch die Validierung die Frage „Am I building the right product?“ beantwortet werden (Boehm 1984). In jeder Abstraktionsebene werden die Anforderungen validiert, indem überprüft wird, ob sie mit den initialen Zielen und PSS-Anforderungen übereinstimmen. Besonders wichtig ist aber die Validierung auf der Komponentenebene, denn die Domänen-Anforderungen der Komponentenebene werden direkt an die Entwicklung weitergegeben. In diesem Zusammenhang soll geklärt werden, ob die Domänenanforderungen die initialen Ziele und PSS-Anforderungen erfüllen. Deswegen werden die Techniken der Validierung im Rahmen der Aktivitäten zur Ermittlung und Analyse von Domänen-Anforderungen beschrieben.

7.3.4.2 Techniken zur Ermittlung und Analyse von Funktionsstruktur-Anforderungen und Domänen-Anforderungen

Die Ermittlung von Funktionsstruktur-Anforderungen und Domänen-Anforderungen findet im Rahmen des in Kapiteln 7.3.3.1.2, 7.3.4.1.1 und 7.3.4.1.2 beschriebenen Verfahren zur iterativen, wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung statt. Zur Unterstützung des Vorgehens können die Techniken zur Anforderungsermittlung (s. Kapitel 7.3.2.2) verwendet werden. Zur Anforderungsvereinbarung, Anforderungskonkretisierung und Anforderungsverfolgung können die in Kapiteln 7.3.2.3, 7.3.2.4, 7.3.2.5 beschriebenen Techniken eingesetzt werden. Dabei sollen die für die Abstraktionsebene spezifischen Informationen, wie bspw. die das PSS beschreibenden Funktionen oder der Grobentwurf, der die Komponenten des PSS angibt, in die Verwendung der Techniken einbezogen werden.

Die Techniken der Validierung sollen Fehler in den Anforderungen entdecken und somit die Qualität der Anforderungen sicherstellen. Zur Anforderungvalidierung können Inspektionen, Reviews, Walkthroughs oder Prototyping eingesetzt werden (Pohl/Rupp 2009, 99ff.; Denger/Olsson 2005; Pohl 2007, 441ff.), die im Folgenden im Detail erklärt werden.

Im Rahmen der *Inspektion* sollen Fehler in Anforderungsdokumenten entdeckt werden. Das Ziel ist dabei, die Qualität der Anforderungsdokumente und der Anforderungen zu verbessern. Der Anforderungsinspektion liegt die Idee der Analyse der Anforderungsdokumente durch Inspektionsteams zu Grunde, die versuchen, Fehlen in Anforderungsdokumenten festzustellen und sie anschließend zu überarbeiten (Rupp 2004, 312-313; Pohl 2007, 442-448). Inspektionen eignen sich gut, um ein gemeinsames Verständnis über das zu lösende Problem zu gewinnen. Ein Nachteil der Inspektion ist, dass der Erfolg ihres Einsatz stark von der Erfahrung und dem Können des Inspektoren abhängt (Rupp 2004, 313). Die Technik kann gut zur Überprüfung der Übereinstimmung der Domänen-Anforderungen mit den initialen Anforderungen an das PSS eingesetzt werden. Eine vereinfachte Form der Inspektion stellt ein *Re-*

view dar, wobei der Ablauf des Reviews flexibler gehalten wird. Im Rahmen eines Reviews werden meist mehr Dokumente überprüft, als bei einer Inspektion. Dafür ist der Detaillierungsgrad der Überprüfung geringer als bei einer Inspektion (Pohl 2007, 448-451). Ein *Walkthrough* basiert auf den Prinzipien einer Inspektion und hat keinen festgelegten Ablauf. Das Ziel eines Walkthrough ist ein gemeinsames Verständnis über die Anforderungen bei allen Stakeholdern und weiteren an der Entwicklung des Systems beteiligten Personen zu fördern (Rupp 2004, 312)..

Ein Prototyp des zu entwickelnden Systems kann ebenso zur Validierung von Anforderungen verwendet werden. Durch den Prototypen können die Stakeholder ihre Anforderungen testen und dadurch sehen, wie die Anforderungen realisiert sind oder realisiert werden sollen (Sommerville/Sawyer 1997a, 203-204; Hickey/Dean 1998; Pohl 2007, 457-462). Der Einsatz eines Prototypen für die Validierung von Anforderungen an PSS kann sich aufgrund der Immaterialität von Dienstleistungen und der damit verbundenen komplexen Darstellung als schwierig erweisen.

7.4 Zusammenfassung

In diesem Kapitel wurde ein Vorgehensmodell zum Requirements Engineering für PSS vorgestellt, das auf dem Artefaktmodell aus Kapitel 6 basiert. Das Vorgehensmodell bietet Aktivitäten und Techniken an, um die im Artefaktmodell definierten Artefakte zu erstellen und zu verwalten.

Als Erstes wurden die begrifflichen Grundlagen zu Vorgehensmodellen, Aktivitäten, Methoden und Techniken geklärt. Anschließend wurde ein Meta-Modell für das Vorgehensmodell eingeführt. Das Vorgehensmodell beschreibt hierarchisch zerlegbare Aktivitäten, dessen Durchführung durch die Angabe von konkreten Techniken unterstützt wird. Die Aktivitäten werden zudem in Phasen organisiert, die an den Abstraktionsebenen des Artefaktmodells ausgerichtet sind. Jede Aktivität benötigt Eingaben (Input) und Ausgaben (Output) für ihre Durchführung. Die Inputs und Outputs stellen immer Artefakte des Artefaktmodells dar.

Basierend auf diesem Meta-Modell wurde das Vorgehen für jede Phase des Vorgehensmodells in einem eigenen Unterkapitel erläutert. Jedes dieser Unterkapitel setzt sich aus einer Beschreibung der durchzuführenden Aktivitäten und der Techniken, die für die jeweiligen Aktivitäten eingesetzt werden können, zusammen. Ein wichtiger Teil des Vorgehensmodells stellt die Beschreibung der wechselseitigen Konkretisierung von Funktionsstrukturen und Anforderungen dar. Dieses Verfahren realisiert die Integration von Entwicklungsinformationen in das Requirements Engineering und trägt dadurch wesentlich zur Realisierung der Anforderungen an den Ansatz bei.

Das Meta-Modell des gesamten Requirements Engineering Ansatzes für PSS wird in Abbildung 7-16 dargestellt. Die Verbindung zwischen dem Artefaktmodell und dem Vorgehensmodell ist hier ersichtlich: Artefakte dienen als Input und Output der Aktivitäten. Zudem sind die Phasen (*Phase*) des Vorgehensmodells an die Abstraktionsebenen (*Abstraktionsebene*) des Artefaktmodells angelehnt. Zur Erstellung der Artefakte einer bestimmten Abstraktionsebene werden die Aktivitäten der dazugehörigen Phase verwendet.

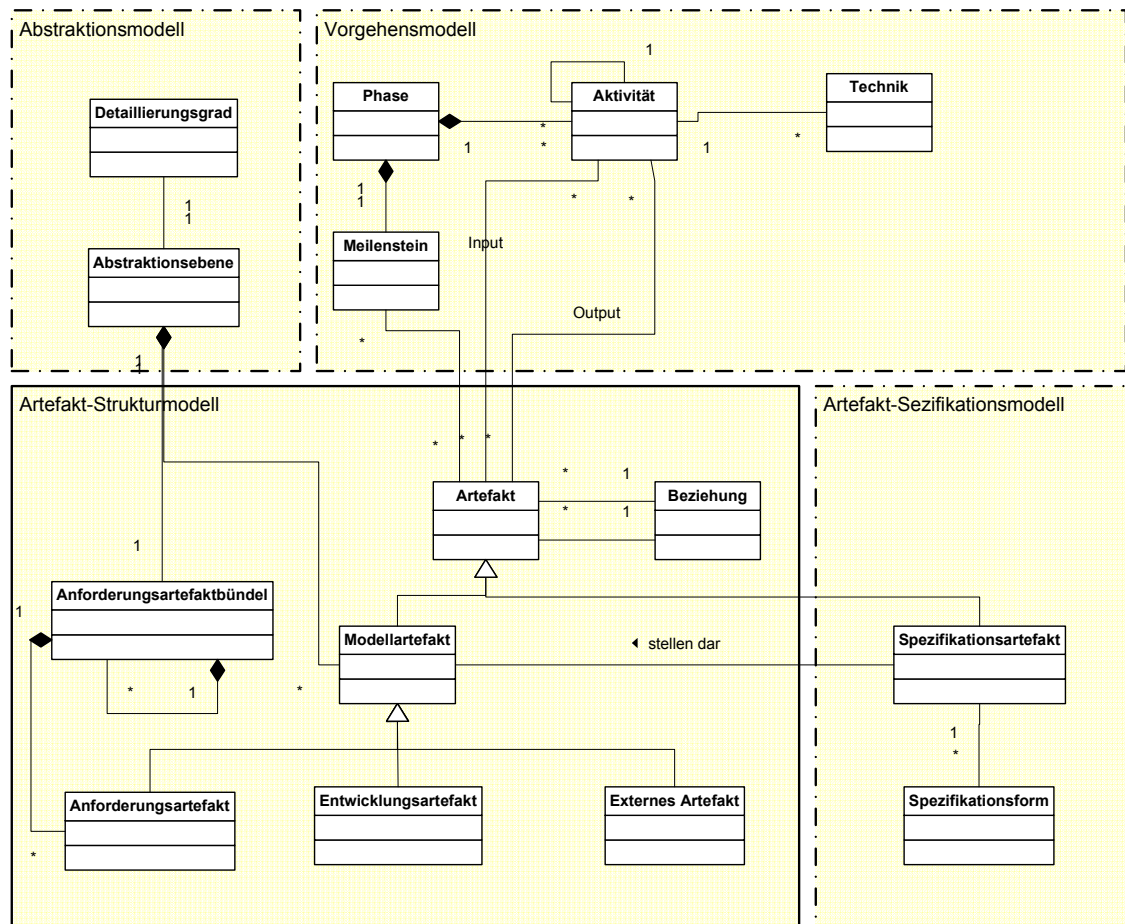


Abbildung 7-16: Metamodell des Requirements Engineering Ansatzes für PSS

Quelle: In Anlehnung an Berkovich et al. (2011a), Berkovich (2010) und Méndez Fernández (2010)

8 Evaluation des Requirements Engineering Ansatzes für Product Service Systems

In Kapitel 8 wird die Evaluation des Requirements Engineering Ansatzes für PSS vorgestellt. In Kapitel 8.1 werden die Ziele der Evaluation, die Evaluationskriterien und die gewählten Evaluationsmethoden beschrieben. In Kapitel 8.2 wird auf analytische Weise gezeigt, dass der entwickelte Ansatz die an ihn gestellten Anforderungen erfüllt. In Kapitel 8.3 wird anhand einer exemplarischen Anwendung des Requirements Engineering Ansatzes seine Machbarkeit gezeigt. Darauf folgend wird in Kapitel 8.4 die merkmalsbasierte Evaluation des Requirements Engineering Ansatzes für PSS beschrieben. Dazu werden Kriterien zur Bewertung der aus der Anwendung des Ansatzes resultierenden Anforderungen definiert. Diese Kriterien basieren auf dem „IEEE recommended practice for software requirements specifications“ (1998) Standard für gut dokumentierte Anforderungen. In Kapitel 8.5 wird eine Evaluation des Requirements Engineering Ansatzes anhand von exemplarischen Anwendungen des Ansatzes in zwei realen Fallstudien durchgeführt. In diesen Fallstudien wurden sich ergänzende Studienobjekte ausgewählt. In der ersten Fallstudie wurde der Ansatz auf ein PSS aus dem IT-Bereich angewendet. Dort steht das Softwareprodukt im Zentrum, das die Automatisierung von Dienstleistungen ermöglicht. In der zweiten Fallstudie wurde ein PSS aus dem Bereich der technischen Produkte gewählt. Im Zentrum steht dort also ein technisches Produkt, das durch die Integration von Dienstleistungen zum PSS ausgebaut wird.

8.1 Ziele, Kriterien und Methoden der Evaluation

Evaluation ist eine Beurteilung von beliebigen Objekten bezogen auf ein Ziel auf der Grundlage eines Systems von Beurteilungskriterien im Feld oder im Labor (Schmidt/Häntschel 2000; Kromrey 2000, 102-107). Die wesentlichen Ziele einer Evaluation sind die Sammlung von der für die Steuerungsentscheidungen relevanten Informationen (Erkenntnisfunktion), die Überprüfung der Erreichbarkeit der im Zusammenhang mit dem zu evaluierenden Objekt festgelegten Ziele (Kontrollfunktion), die Förderung des Dialogs zwischen verschiedenen Stakeholdern bzgl. der Anforderungen an das zu evaluierende Objekt (Lernfunktion) und der Nachweis über die erreichten Ziele (Stockmann 2007; Bortz/Döring 2006, 96-98). Die Evaluation im Rahmen dieser Arbeit hat das Ziel, den entwickelten Requirements Engineering Ansatz auf die Erfüllung der festgelegten Ziele zu überprüfen.

Die Evaluation im Rahmen der gestaltungsorientierten Forschung soll die Artefakte, d. h. Konstrukte, Methoden, Modelle und ihre Instanziierungen (March/Smith 1995) bzgl. der Nützlichkeit, Qualität und Effizienz für die Lösung des bestehenden Problems bewerten (Hevner et al. 2004; Takeda et al. 1990). Dafür stellen Hevner et al. (2004) eine Reihe an Methoden zur Evaluation von Artefakten vor. Tabelle 8-1 gibt einen Überblick über die Methoden der Artefaktevaluation in der gestaltungsorientierten Forschung.

Art	Methode	Beschreibung
Beobachtend	Fallstudie	Untersuchung des Artefakts im jeweiligen Geschäftsumfeld
	Feldstudie	Beobachtung der Nutzung des Artefakts in Projekten
Analytisch	Statische Analyse	Untersuchung der Struktur des Artefakts
	Architekturanalyse	Untersuchung der Integrationsfähigkeit des Artefaktes in die technische Infrastruktur
	Optimierung	Demonstration der Optimalität des Artefaktes für den Zweck oder Aufzeigen der Grenzen der Optimierung
	Dynamische Analyse	Untersuchung des Laufzeitverhaltens des Artefakts
Experimentell	Kontrolliertes Experiment	Untersuchung von Artefakteigenschaften unter kontrollierten Bedingungen
	Simulation	Ausführen des Artefakts unter Nutzung nicht-realer Daten
Testend	Funktionale Tests	Verwenden der Schnittstellen des Artefakts um Fehler zu finden (Black Box Tests)
	Strukturelle Tests	Testen der inneren Funktionsweise des Artefakts um Fehler zu finden (White Box Tests)
Beschreibend	Expertenwissen	Begründen der Nützlichkeit des Artefakts durch Informationen aus der Wissensbasis
	Szenarios	Erstellung von Verwendungsszenarios, um die Nützlichkeit des Artefakts zu zeigen

Tabelle 8-1: Methoden der Artefaktevaluation in der gestaltungsorientierten Forschung

Quelle: (Hevner et al. 2004) zitiert in (Hoffmann 2010, 245)

Der vorliegenden Arbeit liegt die gestaltungsorientierte Forschung zu Grunde, deren Ziel es ist, das Artefakt - im Rahmen der Arbeit stellt das Artefakt im Sinne von Hevner et al. (2004) das Artefaktmodell (s. Kapitel 6) und das Vorgehensmodell (s. Kapitel 7) dar - zu gestalten und zu evaluieren, um dadurch ein Problem in der Praxis zu lösen. Das übergeordnete Ziel eines jeden Entwicklungsprozesses ist es ein erfolgreiches Produkt bzw. PSS zu erstellen, das für die Kunden einen Nutzen bringt. Der in dieser Arbeit entwickelte Ansatz ist im Bereich der ersten Phasen des Entwicklungsprozesses angesiedelt und muss daher einen Beitrag zum Gesamtentwicklungsprozess leisten. Daher wird in der Evaluation die Frage der Nützlichkeit des Ansatzes für das Erreichen des Gesamtziels der Entwicklung beantwortet. Dies beinhaltet die Frage nach den Vor- und Nachteilen des Ansatzes.

Die Evaluation soll den im Rahmen dieser Arbeit entwickelten Ansatz anhand der Methoden und Kriterien zur Evaluation auf seine Eignung zur Entwicklung von PSS und somit zur Lösung des initialen Problems, überprüfen und somit die Einsatzpotenziale des erarbeiteten Ansatzes zeigen. Die Evaluation des entwickelten Requirements Engineering Ansatzes soll die Frage „Erfüllt das Artefakt die Aufgaben, die es erfüllen soll?“ beantworten.

Einer systematischen Evaluation liegt die Idee zu Grunde, zunächst, noch vor der eigentlichen Entwicklung des Artefakts die Anforderungen, die das Artefakt erfüllen soll, festzulegen und anschließend das Artefakt gegen sie zu evaluieren. Basierend auf den Anforderungen werden die Kriterien, die bzgl. der Erfüllung von Anforderungen wichtig sind, ermittelt (Frank 2000).

Die Kriterien sind Eigenschaften des Evaluationsobjektes, die unter Berücksichtigung des Ziels der Evaluation aus der Bedeutung des Evaluationsobjektes abgeleitet werden (Heinrich 2000). Anschließend wird das Artefakt ebenso gegen die Kriterien evaluiert. In Anlehnung an die Methoden der Evaluation (Tabelle 8-1) werden **analytische Verfahren (statische Analysen) und beobachtende Verfahren (Fallstudien)** zur Evaluation des Requirements Engineering Ansatzes für PSS verwendet. Somit gestaltet sich die Evaluation des Requirements Engineering Ansatzes in vier Schritten, die in Abbildung 8-1 dargestellt sind. Jedem dieser Schritte ist ein eigenes Unterkapitel gewidmet, in dem der jeweilige Schritt im Detail erklärt wird. Im ersten Schritt wird die analytische Evaluation anhand einer natürlichsprachlichen Beschreibung der Erfüllung der Evaluationskriterien gegeben. Im zweiten Teil wird eine exemplarische Anwendung des Ansatzes auf ein Beispiel vorgestellt, um dadurch die Durchführbarkeit des Ansatzes zu diskutieren. Im dritten Teil wird der Requirements Engineering Ansatz von PSS anhand von Merkmalen, die den Ansatz charakterisieren, bewertet. Im vierten Teil werden zwei Fallstudien aus der Praxis vorgestellt, in denen der Ansatz retrospektiv angewendet wurde. Dieser Teil enthält auch eine Expertenbewertung des Ergebnisses der Fallstudien und eine Diskussion des Ansatzes basierend auf den Merkmalen aus dem dritten Teil der Evaluation durch die Experten. Zudem wurden die Experten vergangenheitsbezogen gefragt, welche Nützlichkeit des Ansatzes sie sich erwarten, wenn sie ihm zu Beginn der Entwicklung eingesetzt hätten.

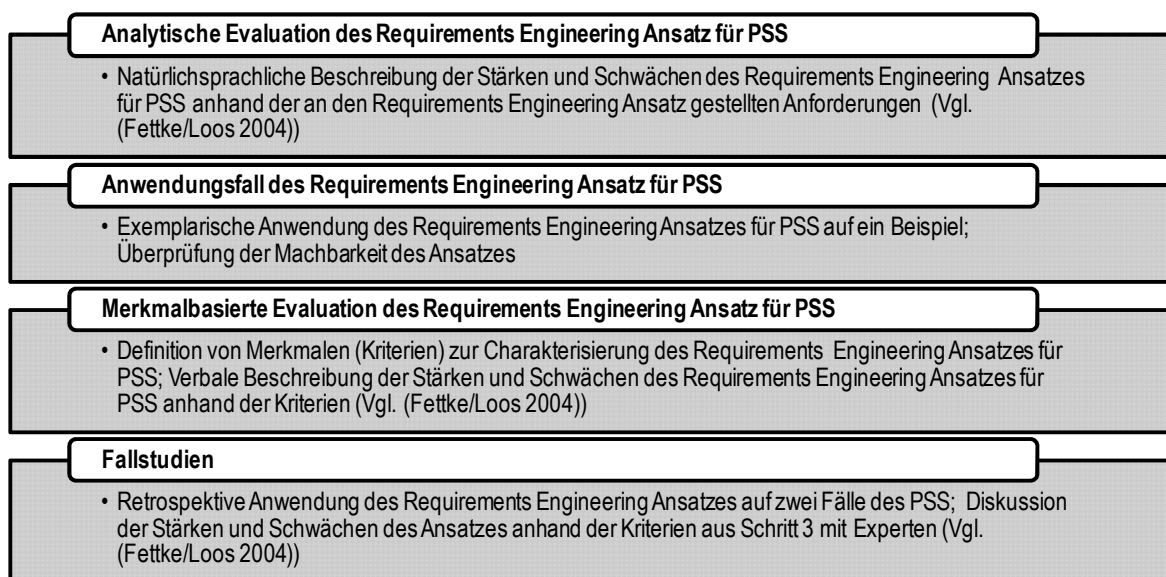


Abbildung 8-1: Aufbau der Evaluation des Requirements Engineering Ansatzes für PSS

Quelle: Eigene Darstellung

8.2 Analytische Evaluation des Requirements Engineering Ansatzes für PSS

Einer analytischen Evaluation liegt die Idee zu Grunde, die Stärken und Schwächen eines zu evaluierenden Objektes in natürlicher Sprache und basierend auf logischen Schlüssen (analytisch) zu beschreiben (Fettke/Loos 2004). Zur Beurteilung des Requirements Engineering Ansatzes für PSS eignen sich die Anforderungen, die an den Ansatz gestellt wurden (s. Kapitel 4.7). So kann festgestellt werden, ob und zu welchem Grad der Ansatz die Anforderungen erfüllt. Die Bewertung der Anforderungserfüllung erfolgt argumentativ.

Anforderung 1: Domänenübergreifende Kooperation. Die Konkretisierung von Anforderungen in Anlehnung an die Schritte der Entwicklung ermöglicht den beteiligten Domänen über die Anforderungen nicht nur im Rahmen von Lasten- und Pflichtenheften, sondern bei jedem Konkretisierungsschritt zu diskutieren. Durch die Vorgabe von Artefakten und Taxonomien für Anforderungen an das technische Produkt und an die Dienstleistungen (s. Kapitel 6.3.3.1) werden Anknüpfungspunkte zur Kommunikation zwischen den beteiligten Domänen geschaffen. Die beteiligten Domänen können dadurch ihre Diskussionen über die Anforderungen gezielt gestalten. Durch die Zuordnung von Anforderungen zu den Funktionen und eine anschließende Konkretisierung von Anforderungen bzgl. der domänenspezifischen Komponenten werden die Interdependenzen zwischen den domänenspezifischen Anforderungen klar herausgestellt, so dass die beteiligten Domänen die Abhängigkeiten zwischen den Anforderungen innerhalb einer Domäne und zwischen den Anforderungen, die zu verschiedenen Domänen gehören, in ihre Diskussionen aufnehmen können.

Anforderung 2: Durchgängige Kommunikation der Anforderungen an den Kunden und die Stakeholder. Die Strukturierung von Anforderungen in Artefakte und in die damit im Zusammenhang stehenden Taxonomien ermöglichen es, die Anforderungen den Kunden und weiteren Stakeholdern geordnet nach relevanten Themenbereichen zu präsentieren und dadurch gezielte Diskussionen zu führen. Durch die schrittweise Konkretisierung von Anforderungen und ihre Dokumentation in Anlehnung an die Phasen der Entwicklung von PSS besteht die Möglichkeit, die Anforderungen in jedem Entwicklungsschritt mit den Kunden und Stakeholdern zu vereinbaren und zu validieren. Damit kann das Feedback der Kunden und der Stakeholder iterativ über die Phasen der Entwicklung in kurzen Abständen eingeholt und berücksichtigt werden. Auch wird das Verständnis der Kunden und Stakeholder über die Anforderungen durch ihre Einbeziehung in Anlehnung an die Entwicklungsschritte gefördert, indem die Kunden und Stakeholder den Konkretisierungsprozess der Anforderungen nachvollziehen können.

Anforderung 3: Bereitstellung von Vorgehen zur Anforderungsermittlung. Die Anforderungsartefakte und die im Zusammenhang mit ihnen stehenden Taxonomien der Systemebene (s. Kapitel 6.3.2) geben die Kategorien von Anforderungen (thematische Strukturierung) an das PSS vor. Die Orientierung an die vorgeschlagene Struktur für die Anforderungen an das PSS ermöglicht auch Fragen zu impliziten Anforderungen zu stellen. Die externen Artefakte geben die relevanten Anforderungsquellen an, die die Anforderungen an das PSS liefern. Die entsprechenden Aktivitäten, die im Vorgehensmodell abgebildet sind, helfen bei der Anforderungsermittlung und der anschließenden Anforderungsdokumentation (s. Kapitel 7.3.2.1). Die in Kapitel 7.3.2.2 beschriebenen Techniken unterstützen die Anforderungsermittlung.

Anforderung 4: Bereitstellung von Vorgehen zur Analyse von Anforderungen an PSS. Durch die Vorgabe von Abstraktionsebenen wird die Analyse und Konkretisierung von Anforderungen in Anlehnung an die Entwicklungsschritte des PSS unterstützt. In jedem Konkretisierungsschritt werden die entsprechenden Entwicklungsinformationen einbezogen, wodurch eine nahtlose Integration der Anforderungen in den Entwicklungsprozess geschaffen wird. Die Abstraktionsebenen verhindern die Vermischung mit unterschiedlichen Detaillierungsgraden. Sie geben genau an welche Anforderungen in welchem Entwicklungsschritt erstellt werden und ermöglichen den Detaillierungsgrad in jeder Abstraktionsebene einheitlich zu halten. Durch die Vorgabe von Anforderungsartefakten ist es möglich, die Anforderungen nach lösungsorientierten Kategorien zu strukturieren und dadurch detailliertere und konkre-

re Angaben zu ihnen zu machen. Die entsprechenden Aktivitäten und Techniken unterstützen die Analyse und Konkretisierung von Anforderungen (s. Kapitel 7.3.2, 7.3.3, 7.3.4).

Anforderung 5: Bereitstellung von Vorgehen zur Anforderungsvereinbarung. Die Anforderungsartefakte geben die Strukturierung von Anforderungen nach Kategorien vor, die es ermöglicht Konflikte themenorientiert zu suchen. Die kausalen Abhängigkeiten zwischen den Funktionen der Funktionsebene (s. Kapitel 6.3.4, 7.3.3, 7.3.4) unterstützen die Konfliktidentifikation, indem nach Konflikten zwischen den Anforderungen gesucht wird, die den miteinander verbundenen Anforderungen zugeordnet sind. Somit ist es möglich die Konflikte sowohl zwischen den domänenspezifischen Anforderungen als auch zwischen den Anforderungen, die zu unterschiedlichen Domänen gehören, zu identifizieren. Desweiteren verfügen die Anforderungen derselben Abstraktionsebene über einen vergleichbaren Detaillierungsgrad (s. Kapitel 6.1.3.3), so dass die Identifikation und die Auflösung von Konflikten erfolgen können. Zur Unterstützung der Konfliktsuche und Konfliktauflösung können die entsprechenden Techniken (s. Kapitel 7.3.2.3) eingesetzt werden.

Anforderung 6: Bereitstellung von Vorgehen zur Anforderungsdokumentation. Die Anforderungen werden sowohl natürlichsprachlich als auch in Form von konzeptuellen Modellen dokumentiert. Die natürlichsprachliche Dokumentation in jeder Abstraktionsebene ermöglicht die schrittweise Konkretisierung von Anforderungen nachzuvollziehen und dabei einen nahtlosen Übergang vom Lastenheft ins Pflichtenheft zu schaffen. Die domänenspezifischen Informationen können ebenso durch die natürlichsprachliche Dokumentationsform berücksichtigt werden. Die Dokumentation von Anforderungen mittels konzeptueller Modelle ermöglicht die Anforderungen konkreter und fokussiert auf bestimmte Aspekte darzustellen.

Anforderung 7: Bereitstellung von Vorgehen zur Anforderungsverfolgung. Jede Anforderung der Abstraktionsebene i wird aus einer oder mehreren Anforderungen der Abstraktionsebene $(i-1)$ abgeleitet. Es existieren keine Anforderungen, außer den Zielen, die keine Vorgängeranforderung haben. Bei den Zielen wird stattdessen der Stakeholder angegeben, der das Ziel ursprünglich geäußert hat. Ebenso werden aus den Anforderungen der Abstraktionsebene i eine oder mehrere Anforderungen der Abstraktionsebene $(i+1)$ abgeleitet (s. Kapitel 6.1.3). Somit kann über alle Schritte der Entwicklung nachvollzogen werden, aus welcher Anforderung welche weiteren Anforderungen abgeleitet wurden. Desweiteren bieten die Funktionen und die ihnen zugeordneten Anforderungen die Möglichkeit durch die kausalen Abhängigkeiten zwischen den Funktionen auch die entsprechenden Interdependenzen zwischen den Anforderungen festzustellen. Ebenso werden entsprechende Techniken zur Anforderungsverfolgung eingesetzt (s. Kapitel 7.3.2.5).

Anforderung 8: Bereitstellung von Vorgehen zum Änderungsmanagement. Eng mit der Anforderungsverfolgung ist das Änderungsmanagement verbunden. Durch die Gewährleistung der Anforderungsverfolgung ist es möglich, die Auswirkungen von Änderungen einer Anforderung auf die Vorgänger- und die Nachfolgeranforderungen zu bestimmen. Bei den Änderungen von Anforderungen, die den Funktionen der Funktionsstrukturen zugeordnet sind (s. Kapitel 6.3.4), können die kausalen Abhängigkeiten zwischen den Funktionen die Identifikation von Auswirkungen von Anforderungsänderungen auf weitere Anforderungen unterstützen.

Anforderung 9: Bereitstellung von Vorgehen zur Anforderungvalidierung. Durch die Strukturierung von Anforderungen in Artefakte besteht die Möglichkeit, die Anforderungen in jedem Konkretisierungsschritt, der in Anlehnung an die Entwicklungsschritte stattfindet, zu

validieren. Die Anforderungsartefakte und Taxonomien geben dabei die Kategorien für die Abstimmung der Anforderungen vor. Somit ist es auch möglich für jede Kategorie von Anforderungen relevante Stakeholder für die Validierung zu finden. Zur Unterstützung der Validierung werden entsprechende Techniken (s. Kapitel 7.3.4.2) eingesetzt.

Anforderung 10: Bereitstellung von Vorgehen zur Wiederverwendung. Die Anforderungs- und Entwicklungsartefakte gruppieren die Anforderungen nach thematischen Kategorien, die es ermöglichen die Anforderungen und die Entwicklungsinformationen bei Bedarf wiederzuverwenden. Die Abstraktionsebenen, die die Anforderungs- und Entwicklungsinformationen in Anlehnung an die Entwicklungsschritte strukturieren, geben den Detaillierungsgrad der Anforderungen an (s. Kapitel 6.1.3.3), der als Unterstützung zur Identifikation der zur Wiederverwendung geeigneten Anforderungen verwendet wird. So können bspw. während der Erstellung des Funktionsstruktur-Entwurfs Funktionsstruktur-Anforderungen der Funktionsebene wiederverwendet werden, die nach entsprechenden Anforderungsartefakten gegliedert sind.

Anforderung 11: Festlegung des Geltungsbereichs für legislative Anforderungen. Jede Anforderung besitzt Attribute, die die Anforderung eindeutig identifizieren und beschreiben (s. Kapitel 6.2.2). Jeder Anforderung wird eine Gültigkeitsdauer zugewiesen, die ein Zeitintervall für die Gültigkeit der Anforderungen angibt. Die Gültigkeitsdauer soll immer aktuell gehalten werden und angeben, wann eine Anforderung bzgl. ihrer Aktualität geprüft werden soll.

8.3 Evaluation des Requirements Engineering Ansatzes für PSS anhand eines Anwendungsfalls: Clean Laundry on Demand

Der Anwendungsfall „Clean Laundry on Demand“ erhebt nicht den Anspruch einer Evaluation des Requirements Engineering Ansatzes im Rahmen einer Fallstudie, sondern dient zur Verdeutlichung der Anwendung des Artefakt- und Vorgehensmodells für die Anforderungen an das PSS (Berkovich et al. 2011b). Desweiteren wird im Rahmen des Anwendungsfalls auf die an den Requirements Engineering Ansatz gestellten Anforderungen (s. Kapitel 4.7) eingegangen und anhand des vorliegenden Anwendungsfalls ihre Erfüllung gezeigt. Auf die genaue Beschreibung der angewendeten Techniken zur Erstellung, Modifikation und Verwaltung der Artefakte wird verzichtet. Hierzu sei auf die entsprechenden Erläuterungen im Rahmen des Vorgehensmodells verwiesen.

Dem Anwendungsfall „*Clean Laundry on Demand*“ liegen die Anforderungen zu Grunde, die in diversen Projekten zur Entwicklung von PSS, technischen Produkten und Dienstleistungen am Lehrstuhl für Wirtschaftsinformatik¹⁹ und am Lehrstuhl für Produktentwicklung²⁰ der Technischen Universität München bekannt geworden sind. Weitere Anforderungen wurden aus der Literatur abgeleitet (Regius 2005, 35ff.; Weigert 2005, 168-170).

Im Folgenden wird der Hintergrund des Anwendungsfalls erläutert. Die Wäsche eines Hotels, zu welcher Bettwäsche, Wäsche aus drei Restaurants, und Wäsche aus dem eigenen Spa-Bereich des Hotels gehören, wurde immer hausintern gereinigt. Um sich besser auf ihre Kernkompetenzen, zu welchen das Betreiben des Hotels und der Restaurants gehören, zu konzentrieren und gleichzeitig die Qualität der Reinigung zu verbessern, sucht das Hotel nach

¹⁹ <http://www.winbase.de/>

²⁰ <http://www.pe.mw.tum.de/>

einer Lösung, die es ermöglicht, die Wäsche entsprechend den Rahmenbedingungen des Hotels zu reinigen. Im nachfolgenden werden die Anforderungen an diese Lösung beschrieben (Berkovich et al. 2011b).

Es wurden insgesamt acht Ziele an das PSS seitens des Kunden und des Lösungsanbieters identifiziert. Aus diesen Zielen wurden 28 PSS-Anforderungen (Systemebene) abgeleitet, von denen acht Kunden- und Stakeholderanforderungen, acht Geschäftsprozessanforderungen, drei Umgebungsanforderungen und drei Auftragnehmeranforderungen sind. Aus den PSS-Anforderungen wurden insgesamt 37 Design-Anforderungen (Eigenschaftsebene) abgeleitet, von denen sechs produktorientierte, 15 prozessorientierte, 10 ressourcenorientierte und fünf ergebnisorientierte Anforderungen sind. Es wurden 70 Funktionsstruktur-Anforderungen aus den Design-Anforderungen abgeleitet. Schließlich wurden aus den Funktionsstruktur-Anforderungen 98 Domänen-Anforderungen abgeleitet, von denen sich 42 Anforderungen auf die Hardware-Komponenten, 31 Anforderungen auf die Software-Komponenten und 25 Anforderungen auf die Dienstleistungen beziehen. Exemplarisch wird anhand von einem Ziel an die Bereitstellung und Nutzung des PSS die Anwendung des Artefakt- und Vorgehensmodells gezeigt. Tabelle 8-2 zeigt das Ziel Z_1 , das zu den Geschäftszielen des Kunden gehört.

Ziel
Z_1 (<i>Geschäftsziele des Kunden</i>): Der Kunde ist ein Hotel mit 200 Zimmern im Zentrum Münchens mit guter Verkehrsanbindung und benötigt saubere Wäsche nach Bedarf.

Tabelle 8-2: *Geschäftsziel des Kunden an das PSS „Clean Laundry on Demand“*
Quelle: Eigene Darstellung

Aus dem Geschäftsziel der Kunden wurden vier Anforderungen abgeleitet. Dafür wurden entsprechende Geschäftsprozesse beobachtet, die mit dem Umgang mit der Wäsche, wie Bettwäsche, Tisch- und Küchenwäsche und Wäsche aus dem Spa-Bereich, zu tun haben. Zu solchen Geschäftsprozessen gehören bspw. Abholung und Rückgabe der Wäsche durch das Hotelpersonal an die Reinigung und der damit im Zusammenhang stehende Zimmerreinigungsprozess, der Prozess des Kochens oder der Prozess der Bedienung im Spa-Bereich. Zu den wichtigen Stakeholdern gehören:

- Stakeholder des Kunden: Hotelmanager, Hotelfachfrau, Reinigungskraft, Küchenbedienstete, Schwimmbadbedienstete,
- Stakeholder des Auftragnehmers: Chef, Prozessentwickler, Softwareentwickler, Zuständiger für die Reinigung,
- Zuständiger für die Lieferung der Wäsche an das Hotel.

Darauf basierend und unter Berücksichtigung der identifizierten Ziele wurden die PSS-Anforderungen abgeleitet.

Tabelle 8-3 zeigt die PSS-Anforderungen an „Clean Laundry on Demand“, die aus dem Ziel Z_1 (Tabelle 8-2) abgeleitet wurden.

PSS-Anforderungen
PA_1 (<i>Kunden- und Stakeholderanforderung</i>): Folgende Wäsche muss täglich ausgetauscht und gewaschen werden: Bettwäsche, Tischdecken, Handtücher und Bademäntel des Schwimmbades.
PA_2 (<i>Kunden- und Stakeholderanforderung</i>): Die gewaschene Wäsche muss ab 08:00 Uhr im Lager abholbar sein

PA₃ (<i>Geschäftsprozessanforderung</i>): Im Lager des Hotels soll die Wäsche sortiert gehalten werden.
PA₄ (<i>Geschäftsprozessanforderung</i>): Die zu waschende Wäsche kann ab 16:00 Uhr im Lager abgeholt werden

Tabelle 8-3: PSS-Anforderungen an „Clean Laundry on Demand“
Quelle: Eigene Darstellung

Die Anforderungen PA₁ und PA₃ gehört laut der Taxonomie der Anforderungen der Systemebene (s. Kapitel 6.3.2.1) zu der Kategorie „Technische Funktionalität und Verhalten des PSS“. Die Anforderung PA₂, PA₄ gehören zur Kategorie „Benutzbarkeit“.

Anschließend wurde basierend auf den PSS-Anforderungen der Systementwurf definiert. Es wurden drei Hauptfunktionen identifiziert, „Wäsche reinigen“, „Reinigung planen“ und „Abrechnung durchführen“. Diese Hauptfunktionen geben keine Angaben zu den Komponenten des PSS. Die Hauptfunktionen wurden in weitere Funktionen iterativ verfeinert und in einer Funktionsstruktur zusammengefasst. Die Hauptfunktionen wurden solange verfeinert, bis für jede Funktion beschlossen werden konnte, ob sie durch ein technisches Produkt oder eine Dienstleistung realisierbar ist. Abbildung 8-2 zeigt die Funktionsstruktur für das PSS „Clean Laundry on Demand“. Durch die iterative Verfeinerung der Hauptfunktionen „Wäsche reinigen“, „Reinigung planen“ und „Abrechnung durchführen“ sind folgende Hauptfunktionen entstanden: „F1: Wäsche abholen“, „F2: Wäsche waschen“, „F3: Wäsche trocknen“, „F4: Wäsche bügeln“, „F5: Wäsche sortieren“, „F6: Wäsche zurückliefern“, „F7: Infos abrufen“, „F8: Lieferzeit planen“, „F9: Prozessschritte planen“, „F10: Mengen protokollieren“, „F10: Monatsbericht erstellen“, „F10: Monatsbericht erstellen“, „F12: Rechnung erstellen“ und „F13: Dokumente verschicken“ (Berkovich et al. 2011b).

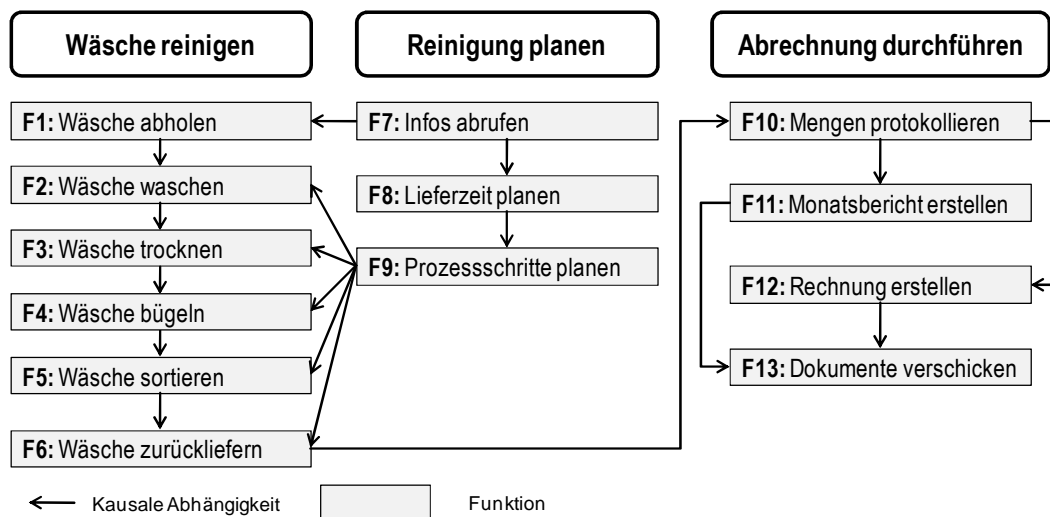


Abbildung 8-2: Funktionsstruktur des PSS „Laundry on Demand“
Quelle: (Berkovich et al. 2011b)

Zur Bestimmung der kausalen Abhängigkeiten zwischen den Hauptfunktionen wurde eine DSM ermittelt (siehe Vorgehensmodell, s. Kapitel 7.3.3 und Kapitel 7.3.4), die in Abbildung 8-3 dargestellt ist. Somit wird für jede Hauptfunktion bestimmt, ob sie zu einer oder mehreren anderen Hauptfunktionen eine kausale Abhängigkeit hat.

F	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
F1	1												
F2		1											
F3			1										
F4				1									
F5					1								
F6						1				1			
F7	1						1						
F8								1					
F9		1	1	1	1	1			1				
F10										1	1		
F11												1	
F12													1
F13													

Abbildung 8-3: DSM zur Bestimmung der kausalen Abhängigkeiten für die Hauptfunktionen des PSS „Laundry on Demand“
 Quelle: (Berkovich et al. 2011b)

Basierend auf den Hauptfunktionen wurden die in Abbildung 8-4 schematisch dargestellten Sach- und Dienstleistungen identifiziert. Zu den technischen Produkten gehören eine Waschmaschine, ein Bügeleisen und ein Trockner. Um das PSS in die Landschaft des Kunden (Hotels) zu integrieren, braucht der Lösungsanbieter einen Zugang zur Software des Kunden, um genau zu wissen, wann, wie viel und welche Art der Wäsche zur Verfügung steht. Die Software stellt dabei das Bindeglied zwischen den eigentlichen technischen Produkten und Dienstleistungen dar. Zu den Dienstleistungen gehören die Durchführung der Abrechnung für die gereinigte Wäsche, die Planung der Reinigung und die Bereitstellung der Lösung an den Kunden, wie Transport der Wäsche zum Kunden, das Abholen der Wäsche vom Kunden oder das Sortieren der Wäsche.

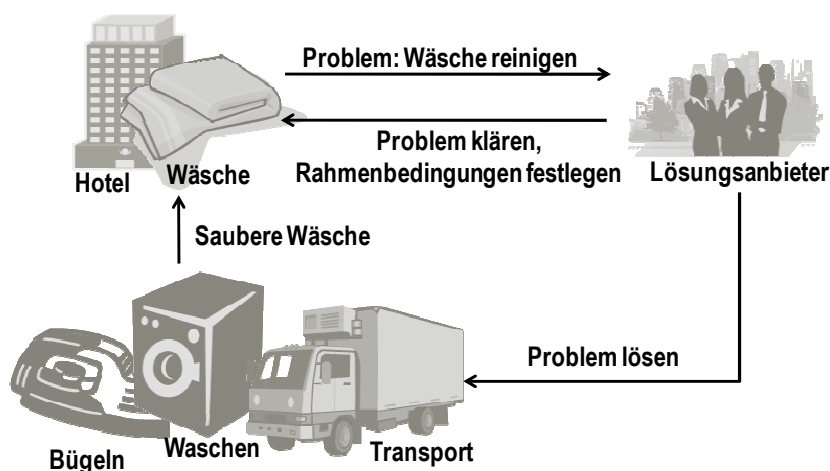


Abbildung 8-4: Clean Laundry on Demand
 Quelle: Eigene Darstellung

Nach der Definition des Systementwurfs wurden anschließend die Design-Anforderungen ermittelt. Aus vier in Tabelle 8-3 dargestellten PSS-Anforderungen wurden acht Design-Anforderungen abgeleitet, indem die PSS-Anforderungen bzgl. der Realisierung der für sie zuständigen Funktionen durch Sach- oder Dienstleistungen konkretisiert wurden. Exemplarisch wird die Konkretisierung der PSS-Anforderungen PA₁ und PA₂ gezeigt (Tabelle 8-4).

PSS-Anforderungen (Vorgängeranforderungen)	Design-Anforderungen (Nachfolgeranforderungen)
PA ₁	DA ₁ (Prozessorientierte Anforderung): Es muss immer ausreichend viel Wäsche zusammen verarbeitet werden, um die Kosteneffizienz zu gewährleisten.
PA ₁	DA ₂ (Produktanforderung): Das Waschen muss alle Arten von Wäsche unterstützen.
PA ₁	DA ₃ (Produktanforderung): Tischdecken und Küchentücher müssen besonders intensiv gereinigt werden.
PA ₁	DA ₄ (Produktanforderung): Seidenwäsche muss besonders schonend gewaschen werden.
PA ₁	DA ₅ (Prozessorientierte Anforderung): Der Planung müssen alle nötigen Informationen zur Verfügung gestellt werden.
PA ₂	DA ₆ (Prozessorientierte Anforderung): Die Planung muss dafür sorgen, dass die Lieferung der Wäsche pünktlich stattfindet.

Tabelle 8-4: Design-Anforderungen an „Clean Laundry on Demand“
Quelle: Eigene Darstellung

Die Anforderung DA₁ gehört laut der Taxonomien der Design-Anforderungen (s. Kapitel 6.3.3.2) zur Kategorie „Effizienz, Produktivität“. Die Anforderungen DA₂, DA₃ und DA₄ gehören zur Kategorie „Technische Funktionen“. Die Anforderung DA₅ gehört der Kategorie „Input-/Outputgrößen“ an. Die Anforderung DA₆ gehört zur Kategorie „Lieferzeiten“.

Anschließend wird die „Wechselseitige Anforderungs- und Funktionskonkretisierung“ (s. Kapitel 7.3.3 und Kapitel 7.3.4) durchgeführt. Dazu werden die Design-Anforderungen den Hauptfunktionen der Funktionsstruktur zugeordnet und iterativ, gegenseitig konkretisiert, während die Hauptfunktionen verfeinert werden.

Zur Analyse, welche Funktionen verfeinert und welche Anforderungen konkretisiert werden müssen, wird eine DMM erstellt (Abbildung 8-5). Darauf basierend werden die Anforderungen den Funktionen zugeordnet. Anschließend werden die Zeilen- und Spaltensummen berechnet, die Funktionen verfeinert und die Anforderungen konkretisiert. Diese Schritte finden iterativ solange statt, bis jede Anforderung genau einer Funktion zugeordnet ist und jede Funktion durch genau eine domänenspezifische Komponente realisierbar ist (s. Kapitel 7.3.4.1).

F-A	DA1	DA2	DA3	DA4	DA5	DA6	DA7	DA8	DA9	DA10	DA11	DA12	DA13	DA14	DA15	DA16	DA17
F1																	
F2		1	1	1						1	1	1	1	1	1	1	1
F3		1		1							1	1	1			1	1
F4		1		1							1	1	1			1	1
F5		1		1			1						1			1	
F6		1		1		1	1						1			1	
F7																	
F8																	
F9	1				1	1											
F10					1												
F11					1		1		1								
F12					1												
F13					1												

Abbildung 8-5: Ausschnitt aus der DMM der Zuordnung von Anforderungen zu Funktionen

Quelle: Eigene Darstellung

Das Ergebnis ist eine detailliertere Funktionsstruktur, die aus verfeinerten und durch eine Komponente zu realisierenden Funktionen besteht. Der Ausschnitt der Funktion *F2: Wäsche waschen*, ist in Abbildung 8-6 dargestellt. Die einzelnen Pfeile stellen die kausalen Abhängigkeiten zwischen den Sub-Funktionen dar.

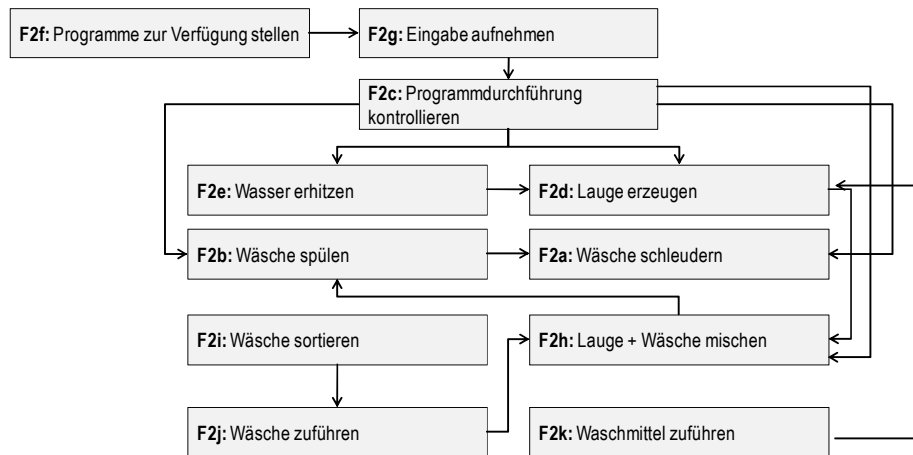


Abbildung 8-6: Detaillierte Funktionsstruktur der Funktion „Wäsche waschen“
 Quelle: (Berkovich et al. 2011b)

Zur Bestimmung der kausalen Abhängigkeiten zwischen den Funktionen wird eine DSM ermittelt, die in Abbildung 8-7 dargestellt wird. Für jede Funktion wird dadurch bestimmt, ob sie zu einer oder mehreren anderen Funktionen eine kausale Abhängigkeit hat.

Wäsche waschen	F2a	F2b	F2c	F2d	F2e	F2f	F2g	F2h	F2i	F2j	F2k
F2a	1										
F2b	1	1									
F2c	1	1	1	1				1			
F2d				1				1			
F2e				1	1						
F2f						1					
F2g			1				1				
F2h		1						1			
F2i									1		
F2j							1			1	
F2k				1							1

Abbildung 8-7: DSM zur Bestimmung der kausalen Abhängigkeiten für die Sub-Funktionen der Funktion „Wäsche waschen“
 Quelle: Eigene Darstellung

Abbildung 8-8 zeigt die Funktionsstruktur der Funktion „Prozessschritte planen“. Da die Funktion „Prozessschritte Planen“ durch eine Dienstleistung realisiert wird, wird sie hier in Form eines EPK-Diagramms dargestellt. Auch in diesem Fall wurde eine DSM zur Bestimmung kausaler Abhängigkeiten erstellt.

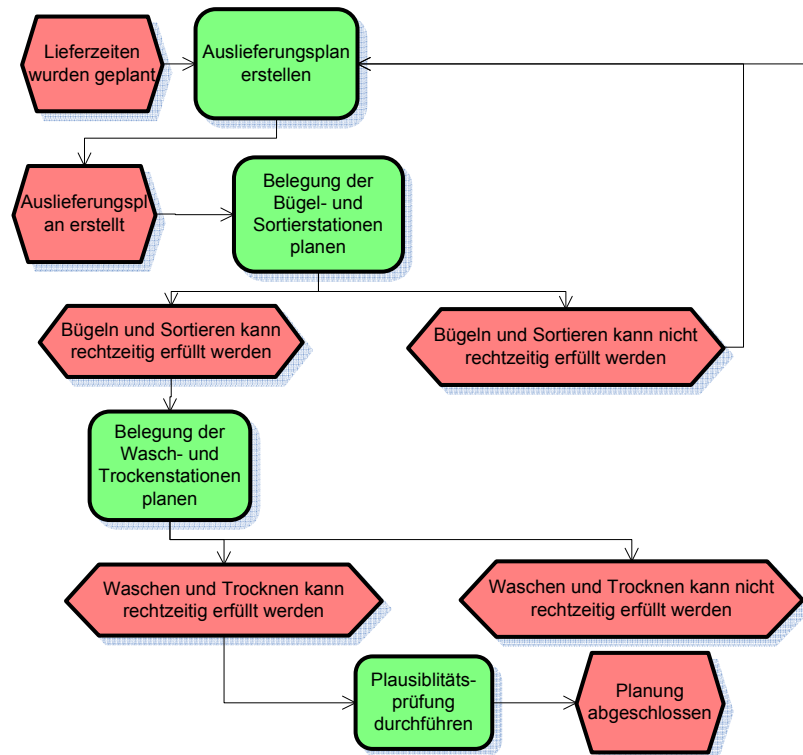


Abbildung 8-8: Detaillierte Funktionsstruktur der Funktion „Prozessschritte Planen“
Quelle: Eigene Darstellung

Exemplarisch werden anhand der Design-Anforderungen **DA₄** und **DA₆** die aus ihnen abgeleiteten Funktionsstruktur-Anforderungen gezeigt (Tabelle 8-5).

Design-Anforderungen (Vorgängeranforderungen)	Funktionsstruktur-Anforderungen (Nachfolgeranforderungen)
DA₄	FA₁ (Detaillierte Produktanforderung): Das Waschmittel muss gleichmäßig auf die Seidenwäsche aufgetragen werden.
DA₄	FA₂ (Detaillierte Produktanforderung): Die Seidenwäsche darf nicht mit mehr als 500U/min geschleudert werden.
DA₄	FA₃ (Detaillierte Produktanforderung): Die Seidenwäsche darf nicht mit mehr als 30°C gewaschen werden.
DA₄	FA₄ (Detaillierte Produktanforderung): Für das Waschen von Seidenwäsche muss Seidenwaschmittel verwendet werden.
DA₆	FA₅ (Detaillierte Dienstleistungsanforderung): Die Planung muss die Kapazitäten und die Durchführungsdauer der Funktionen „Abholen“ und „Zurückliefern“ berücksichtigen.
DA₆	FA₆ (Detaillierte Dienstleistungsanforderung): Die Zulieferung darf an maximal 1 Tag pro Jahr ausfallen.

Tabelle 8-5: Funktionsstruktur-Anforderungen an „Clean Laundry on Demand“
Quelle: Eigene Darstellung

Die Anforderungen FA₁, FA₂, FA₃ und FA₄ gehören laut der Taxonomien der Funktionsstruktur-Anforderungen (s. Kapitel 6.3.4.2) zur Kategorie „Technische Funktionen“. Die Anforderung FA₅ gehört der Kategorie „Schritte“. Die Anforderung FA₆ gehört zur Kategorie „Verfügbarkeit“.

Basierend auf dem Funktionsstrukturentwurf und den Funktionsstruktur-Anforderungen wurden die domänenspezifischen Komponenten und die Domänen-Anforderungen identifiziert. Dafür wurden den Funktionen des Funktionsstrukturentwurfs Wirkprinzipien und Realisierungsvorschläge zugeordnet. Abbildung 8-9 stellt exemplarisch eine detaillierte Darstellung der Funktion „Lauge erzeugen“ mit Eingangs- und Ausgangsinformationen dar.

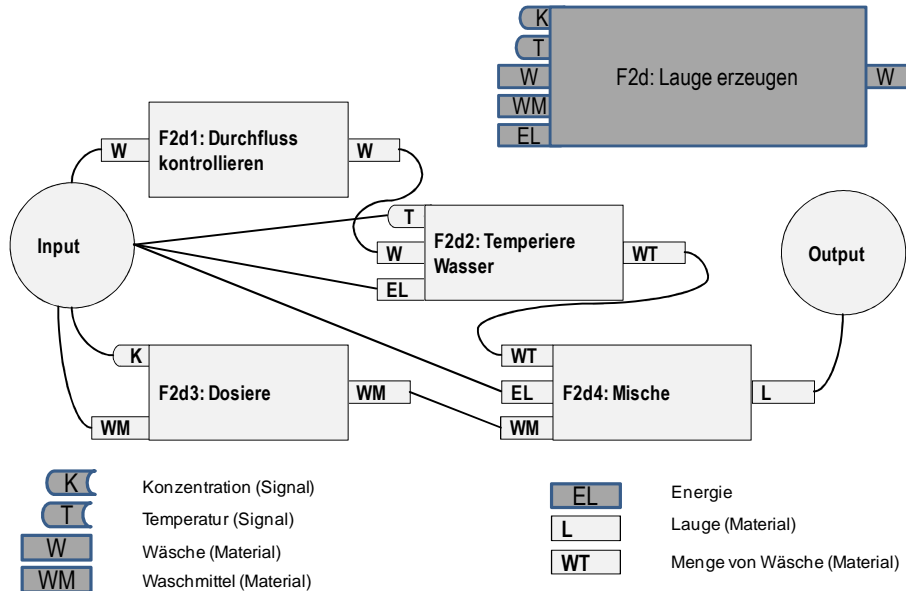


Abbildung 8-9: Detaillierte Darstellung der Funktion „Lauge erzeugen“

Quelle: Eigene Darstellung

Nachdem die einzelnen Funktionen so verfeinert wurden, dass es möglich ist, sie auf die domänenspezifischen Komponenten abzubilden, werden die Funktionsstruktur-Anforderungen bzgl. der Komponenten konkretisiert. Exemplarisch wird das für die Anforderungen FA₃ und FA₂ gezeigt (Tabelle 8-6).

Funktionsstruktur-Anforderungen (Vorgängeranforderungen)	Komponente	Domänen-Anforderungen (Nachfolgeranforderungen)
FA ₃	Temperatursteuerung	DoA ₁ : Die Heizung für das Wasser muss eine Zieltemperatur von 30°C unterstützen.
FA ₃	Temperatursteuerung	DoA ₂ : Das Wasser muss innerhalb von 3min auf eine Temperatur von 30°C erhitzt werden können.
FA ₂	Trommel	DoA ₃ : Die Schleudergeschwindigkeit muss für verschiedene Waschprogramme unterschiedlich einstellbar sein.
FA ₂	Trommel	DoA ₄ : Die Trommel muss eine Schleudergeschwindigkeit von 500U/min unterstützen.

Tabelle 8-6: Domänen-Anforderungen an „Clean Laundry on Demand“

Quelle: Eigene Darstellung

Die Anforderungen DoA₁, DoA₂ gehören laut der Hauptmerkmalliste (Tabelle 10-6) zur Kategorie „Energie“. Die Anforderungen DoA₃ und DoA₄ gehören laut der Hauptmerkmalliste (Tabelle 10-6) zur Kategorie „Kinematik“.

Eine schematische Ableitung von Anforderungen an das PSS „Clean Laundry on Demand“ über die Phasen der Entwicklung wird in Abbildung 8-10 gezeigt.

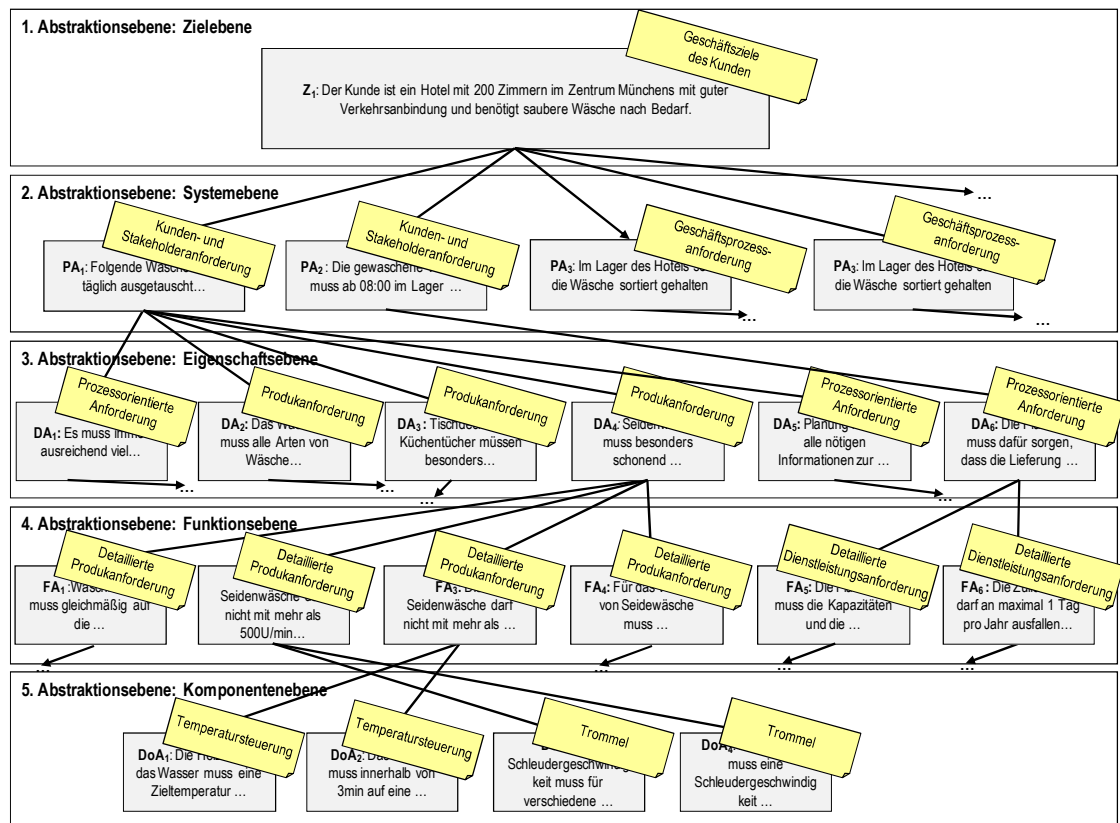


Abbildung 8-10: Schematische Ableitung von Anforderungen an das PSS „Clean Laundry on Demand“

Quelle: Eigene Darstellung

Der Anwendungsfall „Clean Laundry on Demand“ hat gezeigt, dass der Requirements Engineering Ansatz für PSS anwendbar ist und eine strukturierte, mit den Phasen der Entwicklung abgestimmte Konkretisierung von Anforderungen an das PSS ermöglicht und Anforderungen erzeugt, die den domänenspezifischen Komponenten zugeordnet werden können (Berkovich et al. 2011b).

8.4 Merkmalbasierte Evaluation des Requirements Engineering Ansatzes für PSS

Der merkmalsbasierten Evaluation liegt die Idee zu Grunde, eine Menge von Merkmalen, anhand derer der Requirements Engineering Ansatz charakterisiert werden kann, zu definieren (Fettke/Loos 2004). Bei einem merkmalsbasierten Vergleich ist die Auswahl von Merkmalen subjektiv, soll aber begründbar sein. In diesem Kapitel werden die Merkmale definiert und analytisch bewertet. Im anschließenden Kapitel 8.5 wird die Erfüllung der Merkmale in einer praxisnahen Umgebung anhand von Fallstudien und Experteninterviews gezeigt.

Das Ziel des Requirements Engineering Ansatzes für PSS sind die Ermittlung und die Aufbereitung von Anforderungen an ein zu entwickelndes PSS, die dann nahtlos an die Entwicklung übergeben werden können. Das Ergebnis der Anwendung des Ansatzes stellen dokumentierte Anforderungen an das PSS dar. Laut „IEEE recommended practice for software requirements specifications“ (1998) sollen die dokumentierten Anforderungen abgestimmt, bewertet, ein-

deutig, gültig und aktuell, korrekt, konsistent, prüfbar, realisierbar, verfolgbar, vollständig und verständlich sein. Diese Kriterien werden auf die Anforderungen an das PSS angewendet, die mithilfe des Requirements Engineering Ansatzes ermittelt, für die Entwicklung aufbereitet und konkretisiert werden. Die Bewertung der Erfüllung der Kriterien erfolgt argumentativ. Das Ziel der Bewertung ist festzustellen, welche Kriterien vom Ansatz zu welchem Grad erfüllt werden. Die Erfüllung von diesen Kriterien bestimmt die Qualität der dokumentierten Anforderungen.

Eine Anforderung ist **abgestimmt**, wenn sie von allen Stakeholdern als korrekt angesehen und akzeptiert wird. Abgestimmt heißt, dass alle Stakeholder mit allen Anforderungen einverstanden sind und dass zwischen den Anforderungen aus Sicht der Stakeholder keine Konflikte auftreten. Der Requirements Engineering Ansatz ermöglicht die Anforderungen schrittweise in jeder Abstraktionsebene, die an die Schritte der Entwicklung angelehnt sind, zu ermitteln und zu dokumentieren. Im Unterschied zu Lasten- und Pflichtenheften, bei dem eine Abstimmung nur zwei Mal stattfindet, ermöglicht es der Requirements Engineering Ansatz in jedem Schritt die Anforderungen zwischen den Stakeholdern abzustimmen. Die Stakeholder haben somit die Möglichkeit in mehreren detaillierten Schritten über die Anforderungen zu diskutieren. Falls die Stakeholder dabei mit einer Anforderung nicht einverstanden sind, oder Konflikte zwischen Anforderungen finden, kann damit folgendermaßen umgegangen werden: Durch die Nachvollziehbarkeit zwischen den Anforderungsartefakten der Abstraktionsebenen können die Anforderungen, aus der die betroffene Anforderung abgeleitet wurde, identifiziert werden. Als Reaktion kann dann die übergeordnete Anforderung in Frage gestellt werden, oder die konkretisierte Anforderung kann geändert werden. Der Requirements Engineering Ansatz unterstützt die Abstimmung weiterhin dadurch, dass die Anforderungen in verschiedenen Anforderungsartefakten gegliedert werden. Durch diese themenbasierte Kategorisierung der Anforderungen wird es unterstützt, dass sich die Stakeholder nur auf bestimmte Aspekte konzentrieren, z.B. diskutieren sie über die Ressourcen der Dienstleistungen, ohne dabei auf den Prozess der Dienstleistungserbringung einzugehen. Durch diese inhaltliche Strukturierung wird die Abstimmung erleichtert.

Eine Anforderung ist **eindeutig**, wenn sie von den Stakeholdern nur auf eine Art und Weise verstanden werden kann. Der Requirements Engineering Ansatz beschreibt die Abstraktionsebenen, Anforderungsartefakte und deren Verbindung zu den Phasen des Entwicklungsprozesses von PSS. Eine Abstraktionsebene gibt Inhalte vor, die durch den Fortschritt der Entwicklung entstehen, und anhand derer die Anforderungen konkretisiert werden. Die einzelnen Anforderungsartefakte geben auch Inhalte vor, die jedoch auf verschiedenen Kategorien von Anforderungen beruhen. Auch diese Kategorien ermöglichen das Konkretisieren der Anforderungen.

Eine Anforderung ist **korrekt**, wenn sie die Wünsche der Stakeholder adäquat repräsentiert. Der Requirements Engineering Ansatz bietet die Möglichkeit eine Anforderung von ihrem Ursprung (in der ersten Abstraktionsebene) über die Phasen des Entwicklungsprozesses bis hin zu den domänenspezifischen Anforderungen an die Komponenten (in der letzten Abstraktionsebene) zu verfolgen. Die Korrektheit der Anforderungen kann somit in jedem Schritt der Entwicklung überprüft werden. Desweiteren bieten die Anforderungsartefakte Strukturierungs- und Konkretisierungsunterstützung für die Anforderungen, so dass die Ermittlung und die Validierung von Anforderungen durch die thematische Ausrichtung der Anforderungsartefakte gelenkt werden kann. Die Korrektheit wird zudem durch das Vorgehensmodell unter-

stützt, indem es Aktivitäten und Techniken definiert, die zur Validierung der Anforderungen eingesetzt werden.

Eine Anforderung ist **konsistent**, wenn sie gegenüber allen anderen Anforderungen widerspruchsfrei ist. Der Requirements Engineering Ansatz ermöglicht es die initialen Anforderungen (Zielebene) bis hin zu den Domänen-Anforderungen an die Komponenten des PSS über die Phasen der Entwicklung anhand der Anforderungsartefakte zu verfolgen und vice versa. Bei der Identifikation eines Konfliktes zwischen zwei Anforderungen ist es möglich diese Anforderungen bis zu den konfliktfreien Anforderungen, aus denen sie abgeleitet wurden, zu verfolgen. Dadurch wird die Identifikation der Ursache des Konfliktes unterstützt, um darauf basierend den Konflikt aufzulösen. Durch die Zuordnung von Anforderungen zu den Funktionen, die die Funktionalität des zukünftigen Produktes beschreiben und durch die Angabe der Abhängigkeiten zwischen den Funktionen kann die Suche nach Konflikten unterstützt werden, indem die abhängigen Funktionen implizieren, dass denen zugeordnete Anforderungen möglicherweise konfliktär sind.

Eine Anforderung ist **realisierbar**, wenn sie innerhalb der definierten Rahmenbedingungen umsetzbar ist. Durch die Strukturierung von Anforderungen in Anforderungsartefakte, in Anlehnung an die Entwicklungsschritte, können die Anforderungen bzgl. ihrer möglichen Realisierung schrittweise bei den Stakeholdern und Entwicklern validiert werden. Durch die Einbeziehung der Entwicklungsinformationen in jedem Schritt der Anforderungskonkretisierung können die Anforderungen bzgl. der weiteren Entwicklung überprüft werden.

Eine Anforderung ist **verfolgbar**, wenn der Ursprung einer Anforderung, ihre Umsetzung und die Interdependenzen zu den anderen Anforderungen nachvollziehbar sind. Durch die schrittweise Konkretisierung der Anforderungen von den Zielen zu den Domänen-Anforderungen an die Komponenten, wird sichergestellt, dass die Beziehungen von jeder Anforderung zu ihren Vorgänger- und Nachfolgeranforderungen festgehalten werden. Zudem legt der Requirements Engineering Ansatz fest, dass beim Einfügen einer neuen Anforderung in das Artefaktmodell, auch entsprechende vorgelagerte Anforderungen in die Abstraktionsebene davor eingefügt werden. Außerdem muss die neu eingefügte Anforderung auch bis hin zu den Komponentenanforderungen verfeinert werden. Auf diese Weise wird sichergestellt, dass für jede Anforderung alle Beziehungen zu vor- und nachgelagerten Anforderungen vorhanden sind.

Eine Anforderungsspezifikation ist **vollständig**, wenn sie die geforderte Funktionalität des PSS vollständig beschreibt. Der Requirements Engineering Ansatz strukturiert die Anforderungen in Anforderungsartefakte, die die Kategorien für die Anforderungen vorgeben. Dadurch wird ermöglicht, dass die Anforderungsartefakte als Richtlinie zur strukturellen Ermittlung und Konkretisierung von Anforderungen verwendet werden. Durch die iterative Konkretisierung der Anforderungen unter Einbeziehung der Informationen aus dem Entwicklungsprozess wird sichergestellt, dass bei der Konkretisierung keine Anforderungen vergessen werden. Die vorgegebenen Artefakte und Taxonomien können zudem als Checkliste zur Überprüfung der Vollständigkeit genutzt werden.

Eine Anforderung ist **bewertbar**, wenn die Stakeholder sie nach ihrer Wichtigkeit oder Relevanz bewerten können. Die Anforderungsartefakte und die Taxonomien unterstützen die Priorisierung von Anforderungen durch die Vorgabe der thematischen Kategorien, die es ermöglichen, dass sich die Stakeholder auf bestimmte Aspekte bei der Priorisierung konzentrieren.

Eine Anforderung ist **gültig** und **aktuell**, wenn sie im Rahmen des betrachteten Systemkontextes gültig ist. Eine Anforderung ist **prüfbar**, wenn die Funktionalität, die eine Anforderung beschreibt, testbar und messbar ist. Der Requirements Engineering Ansatz gibt keine Angaben zu den beiden Merkmalen. So kann der Ansatz nicht überprüfen, ob eine Anforderung gültig und aktuell ist. Ebenso werden keine Angaben zur Testbarkeit von Anforderungen gegeben. Jedoch unterstützt die Konkretisierung der Anforderungen bis hin zu Domänenanforderungen, dass initiale abstrakte Anforderungen in konkrete Anforderungen überführt werden, die von der Entwicklung verstanden werden.

8.5 Evaluation des Requirements Engineering Ansatzes anhand von Fallstudien

In diesem Kapitel wird die Anwendung des Requirements Engineering Ansatzes in der Praxis anhand von zwei Fallstudien retrospektiv gezeigt (s. Kapitel 8.5.1 und 8.5.2). Die erste Fallstudie „Personal Health Manager“ stammt aus einem Forschungsprojekt, während die zweite Fallstudie „Waschlösung“ im Zusammenhang mit der Untersuchung des Requirements Engineering bei Unternehmen *Alpha* (s. Kapitel 4.5.2) entstanden ist.

8.5.1 Fallstudie 1: Retrospektive Anwendung des Requirements Engineering Ansatzes auf das internetgestützte Bewegungsprogramm „Personal Health Manager“

In Kapitel 3.2.2 wurde das im Rahmen des Projektes SPRINT²¹ entwickelte IT-gestützte Bewegungsprogramm – *Personal Health Manager (PHM)* – beschrieben. PHM ist ein internetgestütztes Bewegungsprogramm, das ein zeit- und ortsunabhängiges Bewegungstraining unterstützt (Esch et al. 2010b; Esch et al. 2010a; Knebel et al. 2009; Knebel et al. 2007). Das Ziel von PHM, welches aus der Kombination aus Hardware-, Software- und Dienstleistungskomponenten besteht, ist die Steigerung der sportlichen Leistungsfähigkeit inaktiver Menschen. Dabei liegt die Hauptschwierigkeit in der richtigen Balance zwischen automatisierten (durch IT bereitgestellten) und persönlichen Dienstleistungen, die in Abhängigkeit zu den Bedürfnissen der Anwender stehen. In diesem Zusammenhang stellt der Requirements Engineering Ansatz eine essentielle Grundlage für die zielgerichtete Ermittlung und Umsetzung der Anforderungen dar. In einer retrospektiven Analyse wurden deshalb die Probleme, die bei der Anforderungsstrukturierung des abgeschlossenen PHM-Projektes aufgetreten sind, identifiziert und es wurde untersucht, ob diese mit Hilfe des entwickelten Ansatzes gelöst werden können (Berkovich et al. 2011a).

8.5.1.1 Evaluationsvorgehen

Der Evaluation lagen drei Schritte zu Grunde:

Schritt 1: Zunächst erfolgte eine Definition der Evaluations-Kriterien. Als Basis hierfür wurden die dafür entwickelten Kriterien der merkmalsbasierten Evaluation (s. Kapitel 8.4) herangezogen. Anschließend fanden zwei Interviews mit den an der Entwicklung des PHM beteiligten Entwicklern statt, die darauf abzielten, Probleme und Herausforderungen im Entwicklungsprozess von PHM aufzudecken. Dadurch konnten verschiedene subjektive Eindrücke und Informationen bzgl. des Entwicklungsprozesses gewonnen werden, welche für eine Fallstudie essenziell sind (Fettke/Loos 2004). Aus diesen subjektiven Informationen wurden

²¹ <http://www.projekt-sprint.de>

Probleme, die bei der Entwicklung aufgetreten sind kondensiert. Daraus wurden Kriterien abgeleitet, die beschreiben was der Ansatz leisten müsste, um diese Probleme zu vermeiden.

Schritt 2: In einem gemeinsamen Workshop mit den Entwicklern wurde der Requirements Engineering Ansatz retrospektiv anhand 29 initialer Anforderungen exemplarisch angewendet. Dadurch sollte die Anwendbarkeit des Ansatzes sowie seine Stärken und Schwächen veranschaulicht werden.

Schritt 3: Im letzten Schritt wurden die Stärken und Schwächen des Ansatzes mit den Entwicklern von PHM diskutiert. Dazu wurde den Entwicklern ein Fragebogen zur Bewertung des Ansatzes vorgelegt. Die Fragen des Fragebogens wurden in Anlehnung an den Kriterien der merkmalsbasierten Evaluation des Requirements Engineering Ansatzes (s. Kapitel 8.4) entwickelt (Tabelle 10-7). Es wurde ein unstrukturiertes Interview mit den Entwicklern geführt, in dem die Fragen des Fragebogens diskutiert und von den Entwicklern beantwortet wurden. Dadurch konnten sowohl qualitative Eindrücke der Entwickler festgehalten werden, als auch eine strukturierte Bewertung des Ansatzes anhand des Fragebogens vorgenommen werden. Abschließend wurden die Entwickler nach der Nützlichkeit des Ansatzes gefragt: Wie schätzen Sie die Nützlichkeit des Ansatzes ein, falls sie ihn bereits zu Beginn der Entwicklung eingesetzt hätten? Welche Vor- und Nachteile glauben Sie hätte der Ansatz mit sich bringen können? Welche Probleme- und Herausforderungen hätten durch den Ansatz gelöst werden können?

8.5.1.2 Ergebnisse der Evaluation

8.5.1.2.1 Schritt 1: Kriterien festlegen

In den Interviews des Schrittes 1, haben die Entwickler berichtet dass dem Entwicklungsprozess des PHM das klassische V-Modell zu Grunde lag. Im ersten Schritt wurden die Anforderungen der Stakeholder erhoben und dokumentiert. Dabei sind folgende Probleme aufgetreten:

- (1) *Erreichen einer Konsistenz zwischen Anforderungen an Hardware-, Software- und Service-Komponenten.* Dies stellte eine große Herausforderung dar, da die verwendeten Requirements Engineering Modelle die Identifikation und Beschreibung der Beziehungen zwischen den Anforderungen an Hardware-, Software- und Dienstleistungskomponenten kaum unterstützten (vgl. auch Kapitel 4.4.2.5)).
- (2) *Schaffen eines einheitlichen Detaillierungsgrads der Anforderungen und Sicherstellen einer ausreichenden Konkretisierung von initialen Anforderungen, die ziemlich abstrakt sind und in „der Sprache des Kunden“ ausgedrückt wurden.* Die verwendeten Requirements Engineering Ansätze bieten keine klaren Kriterien für die Konkretisierung von Anforderungen, sondern legen lediglich fest, dass die Anforderungen solange konkretisiert werden müssen, bis sie einen ausreichenden Detaillierungsgrad erreicht haben, um sie im Pflichtenheft festzuhalten und an die Entwicklung übergeben zu können (vgl. auch Kapitel 4.4.2.5)).
- (3) *Unsystematisches Einbeziehen von Annahmen über die Lösung in die Spezifikation.* Diese Problematik betrifft insbesondere die Fragestellung, welche Funktionen als Service, Software, oder Hardware realisiert werden sollen. Wurden Entscheidungen getroffen, erfolgte keine Dokumentation ihrer Gründe sowie ihrer Informationsquellen.
- (4) *Änderungsmanagement in einer iterativen Entwicklung.* Da die Anforderungen häufig modifiziert wurden, insbesondere durch die Transformation zu automatischen Dienst-

leistungen, mussten iterativ komponentenübergreifende Anpassungen vorgenommen werden. Dies umfasste vor allem Software- sowie Dienstleistungsanforderungen. Dabei stellten die Übersichtlichkeit aller Anforderungen sowie die Verfolgung der Änderungen große Herausforderungen dar.

- (5) *Einbeziehen aller Stakeholder und Vollständigkeit der Anforderungen.* Die Koordination aller Projektbeteiligten sowie die Ermittlung und Umsetzung ihrer Bedürfnisse gestalteten sich aufgrund der zahlreichen Stakeholder (z.B. Anwender, IT-Dienstleister, Verantwortliche für das Gesundheitsmanagement) mit unterschiedlichem Hintergrund als schwierig.

8.5.1.2.2 Schritt 2: Anwenden des Requirements Engineering Ansatzes

Im Folgenden werden die Anforderungen an PHM und ihre Konkretisierungen gezeigt, ohne detailliert auf die Entwicklungsartefakte einzugehen.

Im Workshop wurden zwei Geschäftsziele und 20 daraus abgeleitet Stakeholder-Anforderungen entlang der Abstraktionsebenen des Artefakt-Modells in einem iterativen Prozess konkretisiert, so dass 67 neue Anforderungen gewonnen werden konnten. Exemplarisch wird die Konkretisierung eines Geschäftsziels des Kunden gezeigt (Tabelle 8-7), der die Lösung (PHM) anbietet. Die Anforderungen wurden iterativ und in Anlehnung an die Schritte des Vorgehensmodells abgeleitet.

Ziel
Z₁ (<i>Geschäftsziele des Kunden</i>): Der Sportler soll ein internetgestütztes Bewegungsprogramm, das ein zeit- und ortsunabhängiges Bewegungstraining unterstützt, bekommen. Das Programm soll ebenso eine skalierbare Infrastruktur zur Bereitstellung der Betreuung des Sportlers durch den Fitnesstrainer ermöglichen.

Tabelle 8-7: *Geschäftsziel des Kunden an PHM*
Quelle: Eigene Darstellung

Anschließend wurden folgende Stakeholder identifiziert: Teilnehmer des Bewegungsprogramms (Sportler), Betreuer des Bewegungsprogramms (Ärzte und Trainer), Unternehmen, die ihren Mitarbeitern ein innovatives Bewegungsprogramm anbieten wollen, Trainingsdienstleister, die PHM einsetzen, IT-Betrieb, der die PHM-Plattform betreibt, und der Gesetzgeber.

Aus dem Ziel **Z₁** wurde unter Berücksichtigung der relevanten Geschäftsprozesse, wie Bereitstellung der Informationen über Gesundheit und Bewegung an die Teilnehmer des Bewegungsprogramms, Erstellung des Bewegungsplans mit den Angaben zu den durchzuführenden Übungen, Angabe des Feedback der Teilnehmer und Beantwortung der Anfragen an das Bewegungsprogramm, sieben PSS-Anforderungen abgeleitet. Tabelle 8-8 stellt exemplarisch eine aus **Z₁** abgeleitete PSS-Anforderung **PA₁** dar.

PSS-Anforderung
PA₁ (<i>Kunden- und Stakeholderanforderungen</i>): Die Sportler sollen Informationen über ihre körperlichen Aktivitäten sowie über ihren Trainingsplan erhalten, die der Unterstützung der Leistungssteigerung dienen.

Tabelle 8-8: *PSS-Anforderung an PHM*
Quelle: Eigene Darstellung

Die Anforderung PA₁ gehört laut der Taxonomie für PSS-Anforderungen (s. Kapitel 6.3.2.1) zur Kategorie „Technische Funktionalität und Verhalten des PSS“.

Bei der Erstellung des Systementwurfs wurden acht Hauptfunktionen identifiziert: Teilnehmermanagement, Fitnessüberwachung, Terminverwaltung, Content Management, Kommunikation, Trainingsplan Management, Trainingskurs Management und Gesundheitsuntersuchungsmanagement. Diese Hauptfunktionen, welche das PSS ganzheitlich beschreiben, wurden iterativ zu 25 Funktionen verfeinert. Im Folgenden werden lediglich diejenigen Funktionen erläutert, die eine Beziehung zu den vorliegenden Anforderungen aufweisen.

Die PSS-Anforderungen werden anschließend zu Design-Anforderungen (DA) konkretisiert. Sie zeigen die verschiedenen Aspekte der Bereitstellung des Trainingsplans für die Teilnehmer auf. In Tabelle 8-9 sind die Anforderungen, die aus PA₁ abgeleitet worden sind, abgebildet. Dabei werden zwei Kategorien unterschieden: Produktanforderungen (DA₁ und DA₂) und prozessorientierte (DA₃) Anforderungen.

PSS-Anforderungen (Vorgängeranforderungen)	Design-Anforderungen (Nachfolgeranforderungen)
PA₁	DA₁ (Produktanforderung): Ein zentraler Kalender soll verwendet werden, um alle Termine der Teilnehmer (Sportler) und Trainer zu verwalten.
PA₁	DA₂ (Produktanforderung): Der Trainingsplan muss so gestaltet sein, dass die körperliche Aktivität der Teilnehmer gesteigert wird.
PA₁	DA₃ (Prozessorientierte Anforderung): Der Trainingsplan wird in Zusammenarbeit mit den Teilnehmern und Trainern ausgearbeitet, um sicherzustellen, dass er für die Anwender zweckerfüllend ist.

Tabelle 8-9: Design-Anforderungen an PHM

Quelle: Eigene Darstellung

Die Anforderungen DA₁ und DA₂ gehören laut den Taxonomien für Design-Anforderungen (s. Kapitel 6.3.3.2) zur Kategorie „Technische Funktionen“, die Anforderung DA₃ gehört zur Kategorie „Schnittstellen“.

Die Design-Anforderungen der Eigenschaftsebene wurden konkretisiert, indem sie den entsprechenden Funktionen zugeordnet wurden (Funktionsstruktur-Anforderungen). Dabei kann eine Design-Anforderung aus der Eigenschaftsebene durch mehrere Funktionsstruktur-Anforderungen der Funktionsebene konkretisiert werden. Beispielsweise wurde DA₁ durch FA₁ (Funktion F3: “Termine vereinbaren” zugeordnet und durch das Produkt realisiert) und FA₂ (Funktion F2: “Terminübersicht” zugeordnet und durch das Produkt realisiert) konkretisiert (Tabelle 8-10).

Design-Anforderungen (Vorgängeranforderungen)	Funktionsstruktur-Anforderungen (Nachfolgeranforderungen)
DA ₁	FA ₁ (Detaillierte Produktanforderung): Es besteht die Möglichkeit, Termine bezüglich des Trainingsplans zu vereinbaren, bei welchen sowohl der Trainer als auch der Teilnehmer (Sportler) anwesend sind.
DA ₁	FA ₂ (Detaillierte Produktanforderung): Dem Teilnehmer (Sportler) wird ein Überblick über alle Termine eines Monats gegeben.

Tabelle 8-10: Funktionsstruktur-Anforderungen an PHM

Quelle: Eigene Darstellung

Die Anforderungen FA₁ und FA₂ gehören laut den Taxonomien für Funktionsstruktur-Anforderungen (s. Kapitel 6.3.4.2) zur Kategorie „Technische Funktionen“.

Basierend auf der Funktionsstruktur und den Funktionsstruktur-Anforderungen werden der Grobentwurf und die Domänen-Anforderungen an die Komponenten definiert. Dabei werden die Funktionsstruktur-Anforderungen konkretisiert und den entsprechenden Komponenten zugeordnet.

Funktionsstruktur-Anforderungen (Vorgängeranforderungen)	Komponente	Domänen-Anforderungen (Nachfolgeranforderungen)
FA ₁	Software: Kalender → Termine erstellen	DoA ₁ : Die Terminvereinbarungsfunktion der Software muss in der Lage sein, Teilnehmer und Trainer einzuladen.
FA ₁	Hardware: Erinnerungsfunktion für Termine	DoA ₂ : Die Pulsuhr muss akustische Signale aussenden, um dem Anwender seine Termine mitzuteilen.

Tabelle 8-11: Domänen-Anforderungen an PHM

Quelle: Eigene Darstellung

Die Anforderungen DoA₂ gehört laut der Hauptmerkmalliste (Tabelle 10-6) zur Kategorie „Akustik“. Die Anforderung DoA₁ ist eine funktionale Anforderung.

Abbildung 8-11 zeigt einen Auszug den beschriebenen Abstraktionsebenen und den dazugehörigen Anforderungen. Die Pfeile symbolisieren die Konkretisierungsschritte.

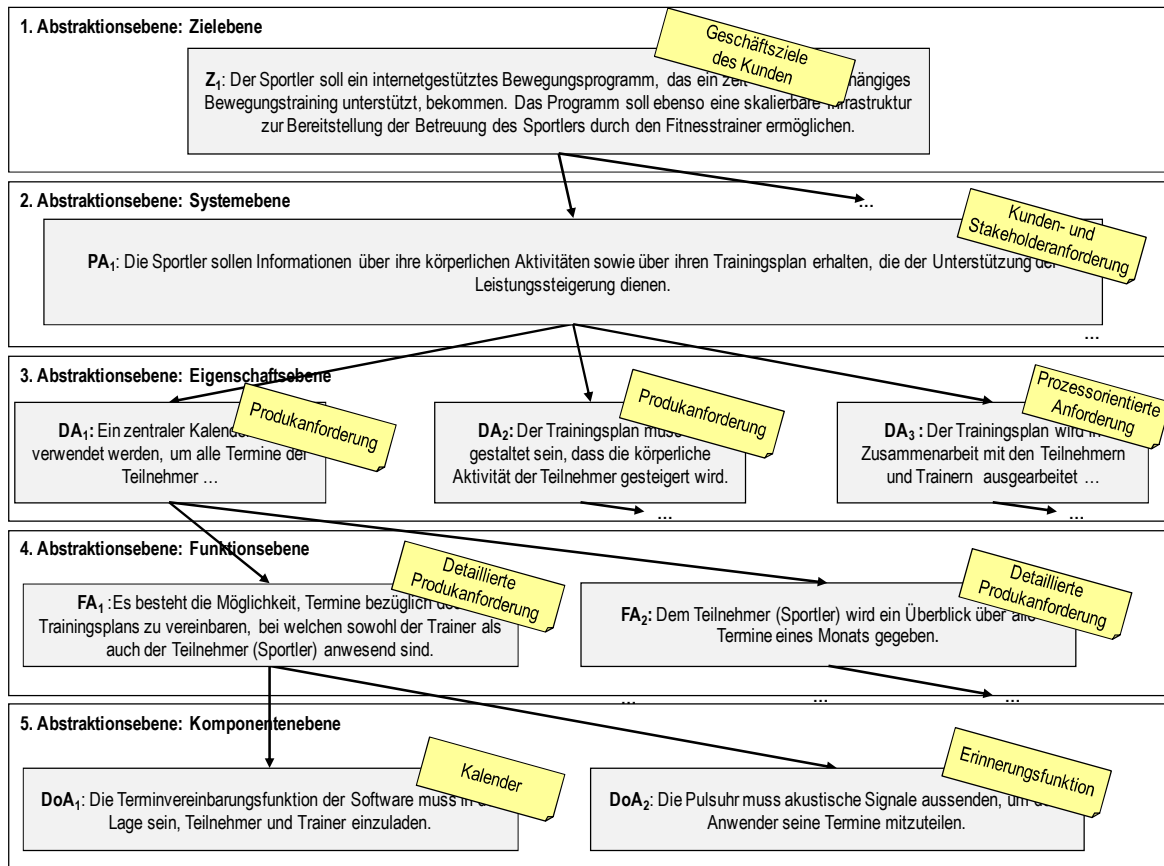


Abbildung 8-11: Schematische Ableitung von Anforderungen an PHM

Quelle: Eigene Darstellung

8.5.1.3 Diskussion der Ergebnisse der Evaluation

Nach der Anwendung des Requirements Engineering Ansatzes auf die Anforderungen, wurden den beiden Entwicklern, die PHM konzipiert und entwickelt haben, Fragebögen vorgelegt, um ihre Einschätzung bzgl. der Unterstützung der Entwicklung des PHM durch den Requirements Engineering Ansatz abzufragen. Die einzelnen Fragen wurden auf einer Skala von „stimme überhaupt nicht zu“, über „stimme nicht zu“, „teils / teils“, „stimme zu“ bis „stimme voll und ganz zu“ beantwortet. Da für die Befragung nur zwei Experten gewonnen werden konnten, ist es nicht möglich die Daten durch statistische Verfahren auszuwerten. Stattdessen werden die Daten in einem Balkendiagramm veranschaulicht (Abbildung 8-12). Für jede Frage, werden die Antworten der beiden Experten dargestellt. Man erkennt, dass außer in einem Fall, die Antworten der beiden Experten maximal um Eins voneinander abweichen. Daran ist deutlich zu erkennen, dass die Experten eine ähnliche Einschätzung der abgefragten Aspekte vorgenommen haben.

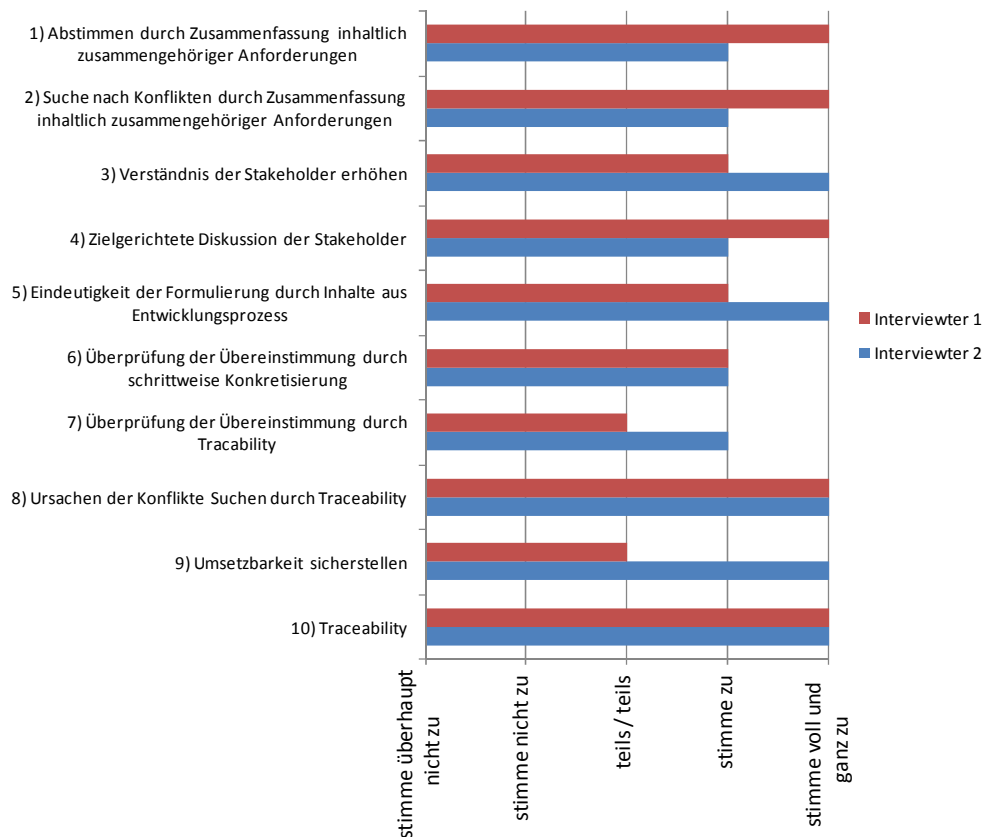


Abbildung 8-12: Ergebnis der Fragebogenumfrage zur Bewertung des Requirements Engineering Ansatzes für PSS anhand von PHM
Quelle: Eigene Darstellung

Zusätzlich zur Beantwortung der Fragebögen wurde ein unstrukturiertes Interview mit den Entwicklern von PHM durchgeführt, um ihre Einschätzung des Requirements Engineering Ansatzes festzustellen. Die Ergebnisse davon werden nachfolgend diskutiert:

Den Fragen 8 und 10 des Fragebogens stimmten beide Interviewten voll und ganz zu. Bei diesen Fragen geht es um die Traceability der resultierenden Anforderungen und um die Konfliktsuche unter Zuhilfenahme der Traceability. Ähnlich hohe Zustimmung erhielten die Fragen zur Unterstützung der zielgerichteten Diskussionen der Stakeholder und der damit in Zusammenhang stehenden Verbesserung des Verständnisses der Anforderungen durch die Stakeholder. Am geringsten war zu Zustimmung zur Überprüfung der Übereinstimmung der Anforderungen, sowohl durch die schrittweise Konkretisierung als auch durch die Traceability.

In den Interviews wurde von den Entwicklern auch intensiv diskutiert, auf welche Weise die zuvor identifizierten Probleme in der Entwicklung durch den Ansatz gelöst werden können:

Die beiden interviewten Entwickler bestätigten, dass der Requirements Engineering Ansatz eine Unterstützung zur Strukturierung und Konkretisierung von Anforderungen liefert. Ausgehend von der Definition von Anforderungsartefakten werden die Wechselwirkungen der einzelnen materiellen und immateriellen Bestandteile des PSS aufgezeigt. Dabei ermöglichen die unterschiedlichen Abstraktionsebenen und Funktionsstrukturen eine anschauliche Darstellung der konkretisierten Anforderungen, die in Hardware-, Software- und Dienstleistungsanforderungen resultieren.

Darüber hinaus ist die Konkretisierung der Anforderungen mit dem Entwicklungsprozess und den Entwicklungsartefakten abgestimmt. Folglich wird das Co-Design der Anforderungen und der Entwicklungsartefakte unterstützt. Die Entwickler stellten fest, dass der Requirements Engineering Ansatz aufgrund der vordefinierten Kategorien (Anforderungsartefakte im Artefaktmodell) eine unstrukturierte Vermischung der Anforderungen unterschiedlicher Detaillierungsgrades verhindert. Dies gilt für die in Schritt 1 der Evaluation (s. Kapitel 8.5.1.2.1) identifizierten Entwicklungsprobleme (1) und (2) bei der Entwicklung des PHM. Das Einbeziehen von Stakeholderanforderungen wird durch das Artefakt-Modell erleichtert. Äußert ein Stakeholder detaillierte Anforderungen, werden diese in eine niedrigere Abstraktionsebene eingeordnet. So sieht der Requirements Engineer jederzeit die Notwendigkeit, Anforderungen auf höherer Ebene zu erheben, um eine Begründung für Anforderungen auf niedrigerer Abstraktionsebene zu liefern. Detaillierte Anforderungen werden daher hinterfragt und eine frühzeitige Fokussierung auf Realisierungsprobleme wird folglich vermieden.

In der Fallstudie ordneten die Entwickler zahlreiche Anforderungen auf eine niedrigere Abstraktionsebene ein und stellten erst dann fest, dass die zugehörigen Anforderungen auf höheren Ebenen fehlten. Dabei sahen sie die Entwicklerprobleme 2, 3 und 5 (s. Kapitel 8.5.1.2.1) durch das Artefakt- und Vorgehensmodell des Requirements Engineering Ansatzes als gelöst an. Durch die Verwendung von Anforderungsartefakten kann das Artefakt-Modell auf verschiedene PSS angewendet werden. Die explizite Definition der Anforderungsartefakte für technische Produkte und Dienstleistungen auf der Eigenschaftsebene sowie die Beschreibung ihrer Konkretisierung in der Funktions- und Komponentenebene ermöglichen, dass die Anforderungen an die gesamte Lösung spezifiziert und konkretisiert werden können. Hierbei werden insbesondere das Entwicklerproblem 5 (s. Kapitel 8.5.1.2.1) angesprochen.

Das Artefaktmodell definiert die Interdependenzen zwischen den Anforderungsartefakten in Anlehnung an die Entwicklungsschritte. Somit ist es möglich die Vorgänger- und Nachfolgeranforderungen zu identifizieren, womit die Anforderungsverfolgung unterstützt wird. Für jede Anforderung einer Abstraktionsebene i kann ihre Konkretisierung auf der Abstraktionsebene $(i+1)$ sowie ihre Begründung auf der Abstraktionsebene $(i-1)$ identifiziert werden. Diese Information schafft eine effiziente Wirkungsanalyse bei Anforderungsänderungen. Die Verfolgung stellt dabei eine Grundvoraussetzung für das Änderungsmanagement dar, welches im Entwicklerproblem 4 (s. Kapitel 8.5.1.2.1) adressiert wird. Die Anforderungen, die im Artefakt-Modell beschrieben werden, sind so eng mit den Funktionsstrukturen verbunden, dass letztere verwendet werden können, um standardisierte und wiederverwendbare Module festzulegen.

Abschließend wurde die Frage nach dem Nutzen des Ansatzes gefragt, wenn er bereits zur Beginn der Entwicklung angewendet worden wäre. Die Entwickler fassten zusammen, dass der Ansatz dabei geholfen hätte konkretere und detailliertere Anforderungen zu spezifizieren. Auch hätten die Anforderungen vollständiger und konsistenter festgelegt werden können. Die Entwickler wiesen jedoch auch darauf hin, dass der Ansatz mehr Aufwände in der Anforderungserhebung und -analyse verursacht hätte. Die Entwickler wiesen jedoch darauf hin, dass sich diese Mehraufwände letztendlich gelohnt hätten, da sich durch die allgemein bessere Anforderungsspezifikation Iterationen, Abstimmungen und nachträgliche Änderungen in späteren Phasen der Entwicklung hätten vermeiden lassen.

8.5.2 Fallstudie 2: Retrospektive Anwendung des Requirements Engineering Ansatzes im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha²²

Das Unternehmen *Alpha* ist ein Hersteller von Haushaltsgeräten in Deutschland und wurde in Kapitel 4.5.2 vorgestellt. Zu den größten Herausforderungen im Requirements Engineering im *Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha* zählen (s. Kapitel 4.5.2.2):

- Fehlende Verknüpfung zwischen den Anforderungen der Lasten- und Pflichtenhefte. Die Anforderungen aus dem Pflichtenheft haben oft keine Begründung für ihre Existenz in Form einer Anforderung aus dem Lastenheft. Die Anforderungen im Pflichtenheft basieren auf dem Wissen der Entwickler und werden nicht aus den Anforderungen des Lastenhefts abgeleitet. Somit sind die Konkretisierung und die Übernahme von Anforderungen aus dem Lasten- ins Pflichtenheft unklar. Die Konkretisierung von Anforderungen aus dem Lastenheft ist dadurch nicht immer nachvollziehbar. Aus einer weitgehend fehlenden Unterstützung der Verfolgung zwischen den Anforderungen resultiert eine komplexe Überprüfbarkeit der Umsetzung von Anforderungen.
- Unterschiedliche Detaillierungsgrade von Anforderungen. Sowohl in Lasten- als auch Pflichtenheften wurden Anforderungen identifiziert, deren Detaillierungsgrad nicht zu dem Konzept des Dokuments oder zu den anderen Anforderungen passte.
- Das Requirements Engineering soll eine Art „Wissensdatenbank“ für Anforderungen sein, indem alle Anforderungen und ihre Konkretisierungen über die Phasen der Entwicklung festgehalten werden.
- Das Requirements Engineering soll die Wiederverwendbarkeit der Anforderungen ermöglichen.

In diesem Fall erschien der Einsatz des Requirements Engineering Ansatzes als sinnvoll, um die oben genannten Herausforderungen anzugehen.

8.5.2.1 Evaluationsvorgehen

Die Evaluation wurde in drei Schritten durchgeführt:

Schritt 1: Definition der Analysekriterien: Für die Analyse in dieser Fallstudie werden die Kriterien der merkmalsbasierten Evaluation (s. Kapitel 8.4) verwendet. Basierend auf diesen Kriterien wurde ein Fragebogen für ein semi-strukturiertes Interview der Experten abgeleitet. Zusätzlich wurden die Herausforderungen, die in Kapitel 4.5.2.1 identifiziert wurden in den Fragebogen aufgenommen.

Schritt 2: Anwenden des Requirements Engineering Ansatzes: Der Requirements Engineering Ansatz für PSS wurde retrospektiv auf die Anforderungen aus einem Lastenheft, das aus der Fallstudie zum Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bekannt wurde (s. Kapitel 4.5.2), angewendet. Dabei beschrieben die Anforderungen des Lastenhefts das PSS „Waschlösung“, das den Waschvorgang von Textilien unterstützt und dabei automatisch gewartet wird. Um die vier identifizierten Herausforderungen im Requirements Engineering bei *Alpha* (s. Kapitel 8.5.2) anzugehen, wurden die Anforderungen aus dem Lastenheft entsprechend ihrem Detaillierungsgrad den richtigen Abstraktionsebenen zugeordnet und bei Bedarf konkretisiert, indem

²² Das Unternehmen *Alpha* (anonymisiert) wurde in Kapitel 4.5.2 vorgestellt.

zusätzliche durch den relevanten Entwicklungsschritt vorgegebene Informationen ergänzt wurden. Desweiteren wurden die Verbindungen zwischen den Anforderungen im Lasten- und Pflichtenheft über die Phasen der Entwicklung, die durch die Abstraktionsebenen dargestellt sind, hergestellt. Ebenso wurden die fehlenden Vor- und Nachfolgeranforderungen laut den Prinzipien des Artefaktmodells (s. Kapitel 6.1.3) und in Anlehnung an die Schritte des Vorgehensmodells (s. Kapitel 7.3) ergänzt und konkretisiert.

Schritt 3: Die Ergebnisse der retrospektiven Anwendung wurden zwei Experten, die an der Fallstudie (Kapitel 4.5.2.1) beteiligten waren, vorgelegt und im Rahmen eines semi-strukturierten Interviews (Interviewleitfaden in Tabelle 10-9) diskutiert. Die Fragen des Interviewleitfadens basieren auf dem Fragebogen (Tabelle 10-7) zur merkmalsbasierten Evaluation des Requirements Engineering Ansatzes. Zusätzlich wurde das Interview um weitere vier Fragen ergänzt (Tabelle 10-8), die direkt auf die Herausforderungen des Requirements Engineering bei *Alpha* eingehen. Zur Evaluierung der merkmalsbasierten Kriterien wurde während des Interviews zusätzlich ein Fragebogen von der Forscherin ausgefüllt und anschließend den Experten zur Validierung vorgelegt. Das Interview hat ca. 56 Minuten gedauert.

8.5.2.2 Ergebnisse der Evaluation

Exemplarisch wird anhand von zwei Anforderungen aus den untersuchten Lastenheften die Anwendung des Requirements Engineering Ansatzes gezeigt, indem die Konkretisierung von Anforderungen über die Phasen der Entwicklung dargestellt wird, ohne auf die Schritte des Vorgehens zur Strukturierung und Konkretisierung und auf die entsprechenden Techniken einzugehen. Da die Anforderungen im Lastenheft charakteristische Abkürzungen und bereichsspezifische Formulierungen für den Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei *Alpha* enthielten, wurden sie im nachfolgenden Beispiel entsprechend aufbereitet und zu einem besseren Verständnis umformuliert.

Tabelle 8-12 stellt die zwei Anforderungen aus dem Lastenheft dar.

Anforderungen aus dem Lastenheft an die „Waschlösung“
A ₁ : Die Waschmaschine muss alle Safety-Vorschriften erfüllen, auch wenn sie eingebaut oder untergebaut wird.
A ₂ : Die Waschmaschine muss für die Standards verschiedener Märkte (Sprache, Stromspannung, usw.) geeignet sein.

Tabelle 8-12: *Initiale Anforderungen aus dem Lastenheft des Unternehmensbereichs zur Herstellung von Waschmaschinen und Trocknern bei Alpha*

Quelle: Eigene Darstellung

Die Anforderungen beschreiben das technische Produkt und gehören somit zur Eigenschaftsebene. Es sind Anforderungen an das technische Produkt (Waschmaschine), die im Anforderungsartefakt „Produktanforderungen“ zusammengefasst und durch Design-Anforderungen gebündelt werden. Die Anforderung A₁ gehört zur Kategorie „Sicherheit“. Die Anforderung A₂ gehört zur Kategorie „Internationalisierung“ (Kapitel 6.3.3.2). Diese Anforderungen sind technisch und konkret. Die genaue Begründung für ihre Existenz, und somit, welche Ziele und Anforderungen der Kunden sie erfüllen, fehlen dabei. Es ist unklar, durch welche Anforderungen an das gesamte PSS diese Anforderungen zu begründen sind. Da aber das Prinzip der Anforderungsverfolgung (Kapitel 6.1.3.6) gewährleistet werden soll, sollen die fehlenden Anforderungen in die vorgelagerten System- und Zielebene und in die nachgelagerten Funkti-

ons- und Komponentenebene ergänzt werden. Die entsprechenden PSS-Anforderungen, Ziele, Funktionsstruktur-Anforderungen und Komponenten-Anforderungen wurden iterativ und in Anlehnung an die Schritte des Vorgehensmodells (Kapitel 7) abgeleitet. Dabei wurde die Anforderung A_1 von den Anforderungen PA_1 und PA_2 abgeleitet. Die Anforderung A_2 wurde aus der Anforderung PA_3 abgeleitet. Tabelle 8-13 zeigt die Vorgängeranforderungen (PSS-Anforderungen) an die Lösung.

Design-Anforderungen aus dem Lastenheft an die „Waschlösung“ (Nachfolgeranforderungen)	Anforderungen an das PSS „Waschlösung“ (Vorgängeranforderungen)
A_1	Z_1 (Geschäftsziel des Kunden): Das Anwenden der Lösung soll einfach und sicher sein.
A_1	PA_2 (Umgebungsanforderung): Die Lösung soll alle Sicherheitsbestimmungen der EU einhalten.
A_2	PA_3 (Auftragnehmeranforderung): Die Lösung soll weltweit angeboten werden.

Tabelle 8-13: Anforderungen an das PSS „Waschlösung“

Quelle: Eigene Darstellung

Die Anforderung PA_1 gehört zur Kategorie „Sicherheit“, die Anforderung PA_2 gehört zur Kategorie „Normen und Gesetze, Patente“ und die Anforderung PA_3 gehört zur Kategorie „Qualität des PSS“ (Kapitel 6.3.2.1).

Basierend auf den PSS-Anforderungen wurden die Ziele an das PSS abgeleitet. Dabei wurde PA_1 aus Z_1 , PA_2 aus Z_2 und PA_3 aus Z_3 abgeleitet. Tabelle 8-14 stellt die Ziele an das PSS dar.

Anforderungen an das PSS „Waschlösung“ (Nachfolgeranforderungen)	Ziele an das PSS „Waschlösung“ (Vorgängeranforderungen)
PA_1	Z_1 (Geschäftsziel des Kunden): Das Anwenden der Lösung soll einfach sein.
PA_2	Z_2 (Geschäftsziele des Auftragnehmers): Alle gesetzlichen Regelungen der EU müssen eingehalten werden.
PA_3	Z_3 (Geschäftsziele des Auftragnehmers): Die Lösung soll in verschiedenen Marktsegmenten erfolgreich sein.

Tabelle 8-14: Ziele an das PSS „Waschlösung“

Quelle: Eigene Darstellung

Desweiteren wurden für die initialen Anforderungen aus dem Lastenheft (Tabelle 8-12) die Nachfolgeranforderungen (Funktionsstruktur-Anforderungen und Komponenten-Anforderungen) abgeleitet. Dabei stellen die Anforderungen FA_1 und FA_4 die initialen Anforderungen aus dem Lastenheft dar, die aber aufgrund ihres Detaillierungsgrades und ihrer Beschreibung zur Funktionsebene gehören und durch die Anforderungen aus dem Lastenheft, die in Tabelle 8-12 beschrieben sind, begründet werden. Die Anforderungen FA_3 und FA_2 wurden durch die Konkretisierung der Anforderung A_2 in der Funktionsebene ergänzt. Tabelle 8-15 stellt die Funktionsstruktur-Anforderungen dar.

Anforderungen aus dem Lastenheft an die „Waschlösung“ (Vorgängeranforderungen)	Funktionsstruktur-Anforderungen an das PSS „Waschlösung“ (Nachfolgeranforderungen)
A ₁	FA₁ (Detaillierte Produktanforderung): Die Waschmaschine muss gesperrt werden können, um Kinder daran zu hindern sie zu verwenden.
A ₂	FA₂ (Detaillierte Produktanforderung): Eine Bedienungsanleitung für die Waschmaschine muss für jede Region vorhanden sein.
A ₂	FA₃ (Detaillierte Produktanforderung): Die Programmwahl und Steuerung soll in allen Sprachen verfügbar sein.
A ₂	FA₄ (Detaillierte Produktanforderung): Die Waschmaschine muss für die Standards verschiedener Märkte (Sprache, Stromspannung, usw.) geeignet sein.

Tabelle 8-15: Funktionsstruktur-Anforderungen an das PSS „Waschlösung“

Quelle: Eigene Darstellung

Die Anforderung FA₁ gehört zur Kategorie „Sicherheit“, die Anforderungen FA₂, FA₃ und FA₄ gehören zur Kategorie „Internationalisierung“. Die Konkretisierung der Anforderungen aus dem Lastenheft basierte auf den vorhandenen Funktionen. Die Anforderung FA₁ bezieht sich auf die Funktion „Kindersicherung“. Die Anforderung FA₂ bezieht sich auf die Funktion „Waschen“. Die Anforderung FA₃ bezieht sich auf die Funktion „Programm wählen und steuern“. Die Anforderung FA₄ bezieht sich auf die Funktion „Waschen“.

Diese Anforderungen wurden anschließend konkretisiert und den Komponenten zugeordnet. So wurde die Anforderung FA₁ durch die Anforderung DoA₁ an das Gehäuse konkretisiert. Die Anforderung FA₂ wurde durch die Anforderung DoA₂ an die Bedienungsanleitung konkretisiert. Die Anforderung FA₃ wurde durch die Anforderung DoA₃ an das Bedienelement konkretisiert. Die Anforderung FA₄ wurde durch die Anforderung DoA₄ an die Waschmaschine konkretisiert. Die Anforderungen DoA₂, DoA₃ und DoA₄ stammen aus dem vorhandenen Lastenheft und wurden entsprechend ihrem Detaillierungsgrad und ihren Verbindungen zu den Anforderungen der Funktionsebene platziert, die in Tabelle 8-16 dargestellt werden.

Funktionsstruktur-Anforderungen an das PSS „Waschlösung“ (Vorgängeranforderungen)	Domänen-Anforderungen an das PSS „Waschlösung“ (Nachfolgeranforderungen)
FA ₁	DoA₁ (Komponente: Gehäuse): Die Verriegelung der Tür soll automatisch durch das Software-Programm erfolgen.
FA ₂	DoA₂ (Komponenten: Bedienungsanleitung): Es wird ein Benutzerhandbuch angeboten, das die regionalen Standards erfüllt.
FA ₃	DoA₃ (Komponente: Bedienelement): Der Bildschirm der Waschmaschine soll in der Lage sein verschiedene Sprachen anzuzeigen.
FA ₄	DoA₄ (Komponente: Waschmaschine): Kein Ausfall bei "unsauberer" Einspeisung.

Tabelle 8-16: Domänen-Anforderungen an das PSS „Waschlösung“

Quelle: Eigene Darstellung

Abbildung 8-13 zeigt einen Auszug der beschriebenen Abstraktionsebenen und den dazugehörigen Anforderungen. Die Pfeile symbolisieren die Konkretisierungsschritte. Die existierenden Anforderungen aus dem Lastenheft sind schwarz dargestellt.

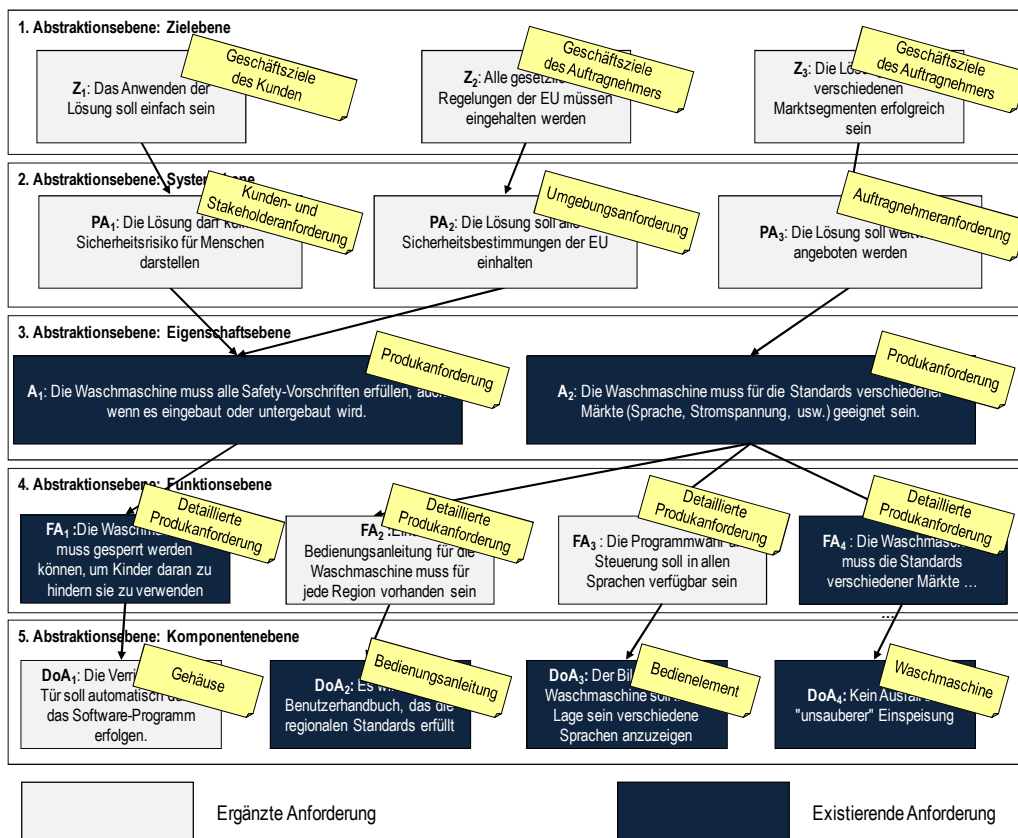


Abbildung 8-13: Schematische Ableitung von Anforderungen an das PSS „Waschlöschung“

Quelle: Eigene Darstellung

8.5.2.3 Diskussion der Ergebnisse der Evaluation

8.5.2.3.1 Ergebnisse der Experteninterviews

Nach der Anwendung des Requirements Engineering Ansatzes auf die Anforderungen, wurde mit den beiden Experten (Kapitel 4.5.2) ein semi-strukturiertes Interview (Tabelle 10-9) geführt. Das Ziel des Interviews war die Einschätzung des Requirements Engineering Ansatzes durch die Experten.

Zu Beginn des Interviews bestätigten beide Experten, dass die in Kapitel 4.5.2.2 dargelegten Herausforderungen für den Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern für Alpha zutreffend sind.

Desweiteren wurde von den Interviewten angegeben, dass der Requirements Engineering Ansatz für PSS zur Strukturierung von Anforderungen und daher zur Unterstützung der Stakeholder beim Abstimmen der Anforderungen gut geeignet ist. Dazu sagte **Experte II**: „Allgemein entwickelt sich [bei Alpha] gerade ein Bewusstsein, dass man Anforderungen strukturieren, in bestimmte Phasen gliedern und verfolgen muss. Folglich sind wir relativ dicht an Ihrem Ansatz dran“.

Die beiden Interviewten bestätigten, dass der Requirements Engineering Ansatz die Suche nach Konflikten zwischen den Anforderungen unterstützt. Es wurde verdeutlicht, dass der Requirements Engineering Ansatz dazu geeignet ist, um Konflikte zwischen den Anforderungen besser zu finden, indem man sie kategorisiert. **Experte I** bestätigte bzgl. der Unterstützung der Identifikation von Konflikten zwischen den Anforderungen: „*In diese Richtung werden wir uns auch orientieren. Das wird tatsächlich in der Zukunft genau so kommen*“. Das Finden der allgemeinen Konflikte im Anforderungsmanagement, die bspw. durch Missverständnisse zwischen den Stakeholdern entstehen, wird dabei nicht unterstützt. So sagte **Experte II**: „*Ich habe ein Gegenbeispiel. Bei den Kühlschränken haben wir den ganz klassischen Konflikt zwischen Energieeffizienz, d.h. möglichst viel Isolierung und möglichst hohes Nutzvolumen, d.h. möglichst wenig Isolierung. Diese Gegensätze werden völlig unterschiedlichen Kategorien zugeordnet, wobei das eine das andere ausschließt. Ein solches Beispiel gibt es bestimmt in jedem Gerät. Diesen Konflikt kann man durch die Kategorisierung nicht herausfinden. Der Requirements Engineering Ansatz ist also kein Allheilmittel*“. Ebenso leistet der Requirements Engineering Ansatz keine Unterstützung zur Identifikation der Konflikte, die erst während der Umsetzung von Anforderungen auftreten, denn der Ansatz konzentriert sich auf die ersten Phasen der Entwicklung (**Experte II**).

Die Konkretisierung von Anforderungen in mehreren Schritten in Anlehnung an den Entwicklungsprozess und die Verbindungen zwischen den Artefakten, die die Kategorien für die Anforderungen angeben, verbessert kaum das Verständnis über die Anforderungen bei den Stakeholdern. Dazu sagte **Experte II**: „*[...] da die Voraussetzung gelten müsste, dass sämtliche Stakeholder jederzeit den kompletten Prozess und Ablauf kennen und im Auge haben. Erfahrungsgemäß ist das nicht so*“. Allerdings wies er hin: „*Insofern glaube ich eher, theoretisch wäre das möglich, dass der Requirements Engineering Ansatz das Verständnis bei den Stakeholdern unterstützt. Praktisch ist das eher nicht*“. Der Requirements Engineering Ansatz kann aber helfen, durch die Vorgabe an Kategorien konstruktive Diskussionen mit den Stakeholdern zu führen und somit das Verständnis von ihnen bzgl. der Anforderungen zu unterstützen. **Experte I** sagte dazu: „*Hat man ein Kategoriengerüst, das so vollständig ist wie Ihres, besteht eine deutliche höhere Chance, ein vollständiges Lastenheft zu erstellen*“. Allerdings soll dafür der Übergang aus der theoretischen Perspektive in die praktische genau erläutert werden, so dass die Verbesserung des Verständnisses über die Anforderungen bei den Stakeholdern auch methodisch unterstützt wird.

Weiterhin unterstützt der Requirements Engineering Ansatz durch die Angabe von Kategorien und Abstraktionsebenen die Anforderungen eindeutiger zu formulieren, so dass keine Vermischung zwischen den verschiedenen Arten von Anforderungen entsteht. So wies **Experte I** hin: „*Sicherlich wird uns der Ansatz dabei helfen, Informationen darüber zu gewinnen, wer für welchen Inhalt zuständig ist. Dieser Aspekt ist in unserem Unternehmen bis heute ein großes Problem. Ferner unterstützt uns der Ansatz, dass wir uns Gedanken über den Detaillierungsgrad bei der Anforderungsformulierung machen*“. Somit können die Kategorien von Anforderungen (Anforderungsartefakte) durch die Vorgabe an inhaltlichen Aspekten die eindeutige Formulierung unterstützen.

Auf die Frage, ob durch die schrittweise Konkretisierung die Übereinstimmung der lösungsorientierten Anforderungen, d.h. die Anforderungen an die einzelnen Komponenten, mit den initialen Anforderungen unterstützt wird, antwortete **Experte I**: „*Da bin ich auf jeden Fall der Meinung, dass die Übereinstimmung gewährleistet wird. Ohne das Artefakt-Modell würde man diesen Vorteil nicht so leicht erreichen können*“.

Die Verfolgbarkeit der Anforderungen über die Phasen des Entwicklungsprozesses unterstützt die Überprüfung der Übereinstimmung der Komponentenanforderungen mit den initialen Anforderungen durch die Stakeholder und die Entwickler, indem man im Artefaktmodell genau nachverfolgen kann, wenn zwei oder mehrere Anforderungen konfliktär sind, auf welche initiale Anforderungen und Ziele die Konflikte zurückzuführen sind. Interviewter I wies darauf hin, dass es zu unterscheiden ist, was das Artefaktmodell und was reine Verbindungstechniken unterstützen. Der Requirements Engineering Ansatz unterstützt die Nachvollziehbarkeit der Ableitung von Anforderungen. Auf die Frage, ob es beim Auftretens eines Konfliktes zwischen den Anforderungen möglich ist, herauszufinden, welche andere Anforderungen davon betroffen sind, antwortet **Experte I**: „*Ja, wenn das Artefakt-Modell die Überführung von initialen Anforderungen in domänenspezifischen Anforderungen unterstützt und dadurch eine Ableitung möglich ist*“.

Auf die Frage, ob der Requirements Engineering Ansatz die Umsetzbarkeit der Anforderungen sicherstellen kann, sodass man am Ende weiß, ob die Anforderungen umsetzbar sind oder nicht, antwortete **Experte I**: „*Da wäre ich mir nicht so sicher. Viele dieser Entscheidungen ergeben sich erst, wenn das Produkt konkreter ausformuliert ist oder Versuche durchgeführt worden sind. [...] In diesem Fall repräsentiert der Ansatz eher eine Wissensbasis*“. Das Wissen aus der Vergangenheit ist durch die Verknüpfungen, die zwischen den Anforderungen bestehen, besser erreichbar und daher für zukünftige Projekte nutzbar. Somit können im Nachhinein die Gründe wie z.B. das Verhältnis zwischen Preis und Nutzen für eine gescheiterte Umsetzung gefunden werden. Eine direkte Bewertung der Umsetzbarkeit im aktuellen Projekt ist jedoch nicht möglich. **Experte II** fügte hinzu: „*Das Modell hilft dabei, die Anforderungen genauer zu spezifizieren, d.h. detaillierter zu formulieren und somit die Anforderungen zielgerichteter umzusetzen. Ob dieser theoretische Effekt tatsächlich in der Praxis auftritt, ist ungewiss*“.

Die Frage, ob der Requirements Engineering Ansatz ermöglicht, durch die schrittweise Konkretisierung der Anforderungen die Beziehungen von jeder Anforderung zu ihren Vorgängern und Nachfolgern festzuhalten, wurde von beiden Interviewten bejaht.

Zu der Frage, ob durch die Angabe der Vor- und Nachfolgeranforderungen durch den Requirements Engineering Ansatz eine Begründung für die Existenz jeder Anforderung und somit für den Grund ihrer Realisierung angegeben werden, äußerten sich beide Experten positiv. Die Vollständigkeit von Anforderungen durch die Vorgabe von Anforderungskategorien (Artefakten) wird nach Meinung der Experten nur bedingt gewährleistet, was von der vorliegenden Situation abhängt.

Die Frage, ob aufgrund der unterschiedlichen Detaillierungsgrade von Anforderungen Verbesserungen durch den Requirements Engineering Ansatz erzielt werden, so dass die Anforderungen bereits lösungsorientiert ins Lastenheft aufgenommen werden, wurde von den beiden Interviewten bejaht. So sagte **Experte I**: „*Wird eine feste Struktur vorgegeben, dann müssen die Anforderungen auf diese Art und Weise inhaltlich beschrieben werden. Ansonsten kommt bei jeder zweiten Anforderung die Diskussion auf, ob es sich um eine initiale oder detaillierte Anforderung handelt*“. Allerdings müsste der Ansatz entsprechend auf die Praxisbedürfnisse angepasst werden. Das richtige Erkennen des Detaillierungsgrades einer Anforderung steht direkt im Zusammenhang mit dem Verfasser und Leser dieser Anforderung.

Die abschließende Frage zur Unterstützung der Wiederverwendung von Anforderungen durch den Ansatz wurde von den beiden Experten bejaht.

8.5.2.3.2 Ergebnisse des Fragebogens zu den merkmalsbasierten Kriterien

Während des Interviews wurde von der Wissenschaftlerin ein Fragebogen ausgefüllt, der eine umfassende Zusammenstellung der im Interview gestellten Fragen enthält. Die Experten haben diesen Fragebogen anschließend zur Validierung erhalten und Korrekturen vorgenommen. Die Ergebnisse der Fragebogenumfrage sind in Abbildung 8-14 dargestellt.

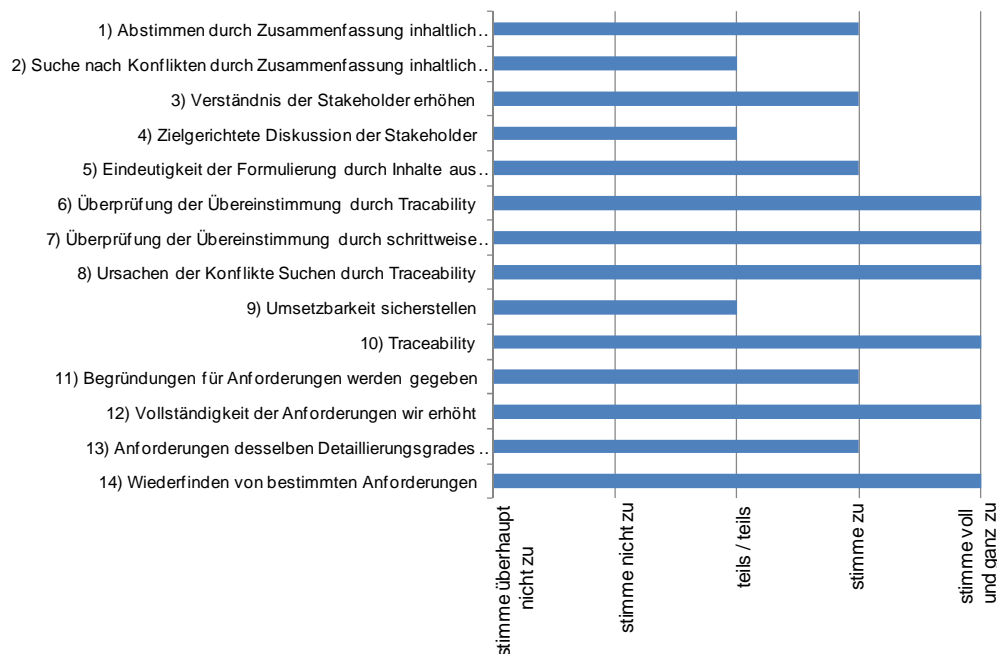


Abbildung 8-14: Ergebnis der Fragebogenumfrage zur Bewertung des Requirements Engineering Ansatzes für PSS im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha

Quelle: Eigene Darstellung

Das Ergebnis des Fragebogens stimmt mit den Aussagen im semi-strukturierten Interview überein. Die Experten sehen vor allem die Traceability und die damit in Verbindungen stehenden Themen am besten durch den Ansatz erfüllt. Auch die Strukturierung der Anforderungen und das Verhindern der Vermischung von Anforderungen mit unterschiedlichem Detaillierungsgrad werden hervorragend durch den Ansatz unterstützt. Die größten Schwächen sehen die Experten in der Sicherstellung der Umsetzbarkeit der Anforderungen, sowie in den Themen die mit der direkten Kommunikation mit den Stakeholdern zu tun haben.

8.5.2.3.3 Fazit der Expertenbewertung

Zusammenfassend lässt sich feststellen, dass der Requirements Engineering Ansatz für PSS gut geeignet ist, um die Herausforderungen im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern bei Alpha zu adressieren. Der Ansatz kann ebenso gut bei der Strukturierung und Konkretisierung von Anforderungen über die Phasen des Entwicklungsprozesses helfen, sowie bei der Übereinstimmung von Anforderungen eingesetzt werden. Die Fragen zur Förderung des Verständnisses von Anforderungen bei den Stakeholdern und zur Identifikation von Konflikten zwischen den Anforderungen konnten von den Experten bzgl. der Vorteile des entwickelten Ansatzes nicht eindeutig beantwortet werden. Die Sicherstellung der Umsetzbarkeit von Anforderungen kann nur in den ersten Phasen der Entwicklung durch den Ansatz unterstützt werden. Der Ansatz hilft aber die Anforderungen ge-

nauer zu spezifizieren, d.h. detaillierter zu formulieren und somit die Anforderungen zielgerichteter umzusetzen. Ebenso unterstützt der Requirements Engineering Ansatz die Vollständigkeit von Anforderungen, die Strukturierung von Anforderungen gemäß ihrem Detaillierungsgrad und die Wiederverwendung von Anforderungen. Als ein wichtiger Aspekt der von den Experten angemerkt wurde ist dass der Requirements Engineering Ansatz mehr an die spezifischen Bedürfnisse der Praxis angepasst werden sollte, indem genau angegeben wird, welche Anforderungs- und Entwicklungsartefakte benötigt werden. Dafür müssen aber die Struktur und die Ausrichtung des Unternehmens bei der Herstellung der Lösungen genau untersucht und in die Konzeption des individuellen Requirements Engineering Ansatzes, dem der entwickelte Requirements Engineering Ansatz zu Grunde liegt, einbezogen werden.

8.6 Zusammenfassung

In diesem Kapitel wurde der Requirements Engineering Ansatz für PSS evaluiert. Die Evaluation wurde in mehreren Schritten durchgeführt. Im ersten Schritt wurden die an den Requirements Engineering Ansatz gestellten Anforderungen (Kapitel 4.7) durch eine analytische Evaluation überprüft. Somit wurde argumentativ gezeigt, dass der Ansatz die an ihn gestellten Anforderungen erfüllt.

Im zweiten Schritt wurde der Ansatz exemplarisch auf ein Beispiel angewendet, um seine Durchführbarkeit zu überprüfen. Als Beispiel wurde der Anwendungsfall „Clean Laundry on Demand“ gewählt, der in verschiedenen Forschungsprojekten zum Einsatz kam. Durch dieses Beispiel konnte die Durchführbarkeit des Ansatzes gezeigt werden.

Der dritte Schritt bestand in einer merkmalsbasierten Evaluation des Requirements Engineering Ansatzes. Die zu Grunde liegenden Kriterien wurden aus den „IEEE recommended practice for software requirements specifications“ (1998) abgeleitet, die vorgeben, welche Charakteristiken „gut“ dokumentierte Anforderungsspezifikationen aufweisen müssen. Es konnte argumentativ gezeigt werden, dass der Ansatz diese Kriterien erfüllt.

Im letzten Schritt der Evaluation wurden zwei Fallstudien durchgeführt. Der Requirements Engineering Ansatz wurden retrospektiv auf zwei Fälle aus dem wissenschaftlichen Umfeld und der Industrie angewendet. Zur Bewertung der Erfüllung der Kriterien der merkmalsbasierten Evaluation wurden Interviews mit Experten aus beiden Fällen geführt. Diese Experten konnten zusätzlich diskutieren, welche Probleme, die während der Entwicklung aufgetreten sind, durch den Ansatz gelöst worden wären. In der Fallstudie aus dem wissenschaftlichen Umfeld wurde vor allem die Unterstützung verschiedener Requirements Aktivitäten durch die Zusammenfassung inhaltlich zusammengehöriger Anforderungen erwähnt. Zudem haben die Entwickler erwähnt, dass die Abstraktionsebenen und die damit vorgegebene schrittweise Konkretisierung dabei hilft einen geeigneten Detaillierungsgrad der Anforderungen zu wählen. In der Fallstudie beim Unternehmen *Alpha* haben die Experten darauf hingewiesen, dass der Ansatz gut bei der Strukturierung und Konkretisierung von Anforderungen über die Phasen des Entwicklungsprozesses helfen kann. Die Strukturierung der Anforderungen und die Sicherstellung der Verfolgbarkeit durch das Artefaktmodell helfen vor allem bei der Suche nach Konflikten und der Übereinstimmung der Anforderungen. Zudem wird die Vollständigkeit durch die Vorgabe der Artefakte und Taxonomien unterstützt. Zusammenfassend, haben die Experten festgestellt, dass der Ansatz für den Einsatz in der Praxis geeignet ist, jedoch auf die entsprechenden Bedürfnisse des jeweiligen Unternehmens angepasst werden muss. Trotz der unterschiedlichen Ausrichtung der beiden betrachteten PSS wurde der entwickelte Ansatz

in beiden Fallstudien als nützlich eingestuft. Der Ansatz lässt sich also sowohl auf PSS anwenden bei denen Software im Zentrum steht, als auch auf solche bei denen Hardware im Zentrum steht.

9 Fazit und Ausblick

9.1 Zusammenfassung der Problemstellung und Ergebnisse der Arbeit

Immer mehr Unternehmen bieten integrierte Bündel von Sach- und Dienstleistungen an, die für den Kunden nicht als einzelne Komponenten erkennbar sind, um dadurch ihre Differenzierung von den Wettbewerbern und ihre Innovationsfähigkeit auszubauen. Diese integrierten Bündel haben das Ziel die Kundenanforderungen möglichst vollständig zu erfüllen und dadurch dem Wechsel von der angebotsorientierten zur nachfrageorientierten Marktsituation gerecht zu werden. Somit entwickeln sich die Unternehmen zu sogenannten „Lösungsanbietern“, die individuelle Lösungen für Kundenprobleme anbieten. Die einzelnen Hardware-, Software- und Dienstleistungskomponenten der Lösung, die auch unter den Namen *Hybride Produkte* oder *Product Service Systems (PSS)* bekannt ist, werden von Produkt-, Software- und Dienstleistungsentwicklung in integrierter Weise hergestellt. Dabei spielt die Software eine wichtige Rolle, indem sie das Bindeglied zwischen Hardware und Dienstleistungen darstellt. Im diesem Zusammenhang wurden im Rahmen dieser Arbeit IT-gestützte PSS betrachtet, bei denen Softwarekomponenten eine entscheidende Rolle bei der Bereitstellung von Basisfunktionalitäten, wie Speicherung und Kommunikation zur Verfügung stehender Ressourcen spielen.

Ein wichtiger Bestandteil der Entwicklung von PSS ist das Requirements Engineering, das die Anforderungen an das PSS ermittelt, dokumentiert und verwaltet. Ein erfolgreiches Requirements Engineering stellt sicher, dass die Anforderungen der Stakeholder an das PSS als Ganzes richtig verstanden und in geeigneter Form an die Entwicklung übergeben werden. Dabei müssen die an der Entwicklung von PSS beteiligten Domänen die Anforderungen richtig interpretieren, um sie korrekt umzusetzen. Da die Domänen oft ein unterschiedliches Verständnis über die Anforderungen haben und unterschiedliche Vorgehensweisen zur Ermittlung und Verwaltung von Anforderungen anwenden, stellt das Erarbeiten einer gemeinsamen Vorstellung über das zu lösende Problem eine Herausforderung im Requirements Engineering dar. Eine weitere Herausforderung stellt die konzeptionelle Lücke zwischen den Anforderungen und der Entwicklung dar. Um diese Herausforderung zu bewältigen, müssen die initialen unkonkreten Anforderungen an das PSS „in die Sprache des Entwicklers“ übersetzt werden. Aufgrund dieser Herausforderungen entstehen oft Iterationen beim Klären von Anforderungen zwischen den Entwicklern, Anforderungsanalysten und Stakeholdern, die auch als Zyklen bezeichnet werden. Diese Zyklen führen zu Verzögerungen beim Bereitstellen des PSS, sinkender Kundenzufriedenheit und erhöhten Kosten und Ressourcenaufwand.

Um die zuvor genannten Herausforderungen zu lösen, wurde in dieser Arbeit ein integrierter und domänenübergreifender Ansatz zum Requirements Engineering entwickelt, der speziell auf PSS zugeschnitten ist. Der Konzeption des Requirements Engineering Ansatzes liegt die Aufbereitung einer theoretischen Basis zu Grunde, indem der Stand der Wissenschaft und der Technik im Requirements Engineering für PSS analysiert wurden. Basierend auf den Ergebnissen einer Literaturanalyse und empirischer Untersuchungen in der Praxis wurden entsprechende Anforderungen an einen Requirements Engineering Ansatz für PSS abgeleitet. Die Zyklen, die während des Requirements Engineering auftreten, wurden bei der Aufbereitung des Standes der Wissenschaft und Technik identifiziert und in die Anforderungen an einen Requirements Engineering Ansatz mit einbezogen.

Der entwickelte Ansatz zum Requirements Engineering für IT-gestützte PSS setzt sich aus einem Artefaktmodell und einem Vorgehensmodell zusammen. Ein Artefakt fasst die Anforderungen an PSS oder Entwicklungsinformationen, basierend auf definierten Kategorien, in Anlehnung an den Vorgaben des Entwicklungsprozesses, zusammen. Das Artefaktmodell beschreibt Artefakte und definiert Strukturierungsprinzipien und Relationen zwischen ihnen. Damit gibt das Artefaktmodell ein Strukturierungs- und Konkretisierungsschema für Anforderungen an PSS vor. Das Artefaktmodell ermöglicht die initialen Kunden- und Stakeholderanforderungen an das PSS als Ganzes festzuhalten und die Anforderungen schrittweise, über die Phasen der Entwicklung und integriert von den beteiligten Domänen zu konkretisieren. Als Ergebnis entstehen konkretisierte Anforderungen, die den jeweiligen domänenspezifischen Komponenten der Produkt-, Software- und Dienstleistungsentwicklung zugeordnet und zur weiteren Entwicklung übergeben werden können. Das Vorgehensmodell ist an das Artefaktmodell angelehnt und beschreibt die Aktivitäten und Techniken zur Erstellung, Modifikation und Verwaltung von den einzelnen Artefakten. Jede Aktivität erhält dabei Artefakte des Artefaktmodells als Eingabe und erzeugt oder verändert Artefakte als Ausgabe. Das Vorgehensmodell definiert insbesondere das Vorgehen zur integrierten und von allen beteiligten Domänen abgestimmten Konkretisierung der Anforderungen. Dabei gibt es vor, auf welche Weise Entwicklungsinformationen in die Anforderungskonkretisierung einfließen.

Der Requirements Engineering Ansatz wurde anhand einer analytischen Evaluation bewertet. Als Kriterien für die Evaluation wurden die an den Ansatz gestellten Anforderungen verwendet, die basierend auf der Literaturrecherche und den Fallstudien in der Praxis erstellt wurden. Zudem wurde der Requirements Engineering Ansatz auf ein fiktives Beispiel, das jedoch an realen Anforderungen angelehnt ist, angewandt. Dieser Anwendungsfall hat gezeigt, dass der Ansatz anwendbar ist und eine strukturierte Konkretisierung von Anforderungen an das PSS, mit ihrer abschließenden Zuordnung zu den domänenspezifischen Komponenten, ermöglicht. Im zweiten Teil der Evaluation wurde die Praxistauglichkeit des Ansatzes anhand einer merkmalsbasierten Evaluation und von Fallstudien untersucht. In der merkmalsbasierten Evaluation wurden Kriterien für die Bewertung des Ansatzes auf Basis von unabhängigen Standards erstellt. Durch ein analytisches Vorgehen wurde dann gezeigt, dass die Kriterien vom Ansatz erfüllt werden. Weiterhin sind diese Kriterien in die Fallstudien eingeflossen. In zwei Fallstudien wurde der Ansatz retrospektiv auf ein PSS angewandt und bzgl. der identifizierten Kriterien bewertet. Das erste PSS stammt aus einem Forschungsprojekt, während das zweite aus einer Industriekooperation hervorgegangen ist. Im zweiten Fall war es auch möglich auf Experten der Industrie zurückzugreifen und die Ergebnisse in semi-strukturierten Interviews zu diskutieren. Diese Fallstudien haben gezeigt, dass der Ansatz dazu geeignet ist die Anforderungen schrittweise zu verfeinern und in Abstimmung mit dem Entwicklungsprozess zu bringen. Das Artefaktmodell ermöglicht es insbesondere explizit über den Detaillierungsgrad von Anforderungen zu sprechen und die Rückverfolgbarkeit von detaillierten technischen Anforderungen zu den initialen Kundenwünschen sicherzustellen. Durch die Vorgabe der Artefakte und der Beschreibung deren Inhalts wird insbesondere auch die Vollständigkeit der Anforderungsspezifikationen sichergestellt.

Die retrospektive Anwendung des Ansatzes zeigte, dass der Ansatz auf die Bedürfnisse des jeweiligen Unternehmens angepasst werden kann, indem die benötigten Anforderungs- und Entwicklungsartefakte sowie die Abstraktionsebenen in Anlehnung an den Entwicklungsprozess des Unternehmens vorgegeben werden. Diese Anpassung wird durch die lose Kopplung von Artefakten und Abstraktionsebenen unterstützt, indem die Artefakte, die dazugehörigen

Taxonomien und die Abstraktionsebenen abhängig von den Entwicklungsvorgaben gewählt werden können.

9.2 Implikationen für die Praxis und die Forschung

Der entwickelte Requirements Engineering Ansatz trägt zum Verständnis der frühen Phasen der Entwicklung von PSS bei, indem er definiert, welche Anforderungen zu ermitteln sind, wie sie mit den Zielen der Kunden und des Lösungsanbieters bezüglich des PSS verbunden sind und wie die Anforderungen schrittweise in Abstimmung mit den Entwicklungsphasen konkretisiert werden. Durch die schrittweise Konkretisierung der Anforderungen über die Phasen der Entwicklung hinweg ermöglicht der Ansatz die verschiedenen Sichten von den beteiligten Domänen in das Requirements Engineering und somit in die Strukturierung und Konkretisierung von Anforderungen zu integrieren. Dadurch wird das Verständnis für die interdisziplinären Zusammenhänge gefördert, die aufgrund der Beteiligung von Domänen mit unterschiedlichem Hintergrundwissen entstehen. Somit leistet die vorliegende Arbeit einen Beitrag zur Unterstützung der domänenübergreifenden Zusammenarbeit bei der integrierten Entwicklung von Hardware-, Software- und Dienstleistungskomponenten des PSS.

Ein weiterer Beitrag sowohl für die Forschung als auch für die Praxis leistet die Arbeit bei der iterativen und wechselseitigen Konkretisierung von Anforderungen und Verfeinerung von Funktionen. Dabei wurden existierende Ansätze in der Literatur zur Anforderungskonkretisierung und Funktionsverfeinerung für PSS erweitert. Anhand einer exemplarischen Anwendung der iterativen und wechselseitigen Anforderungskonkretisierung und Funktionsverfeinerung wurde verdeutlicht, welche Interdependenzen zwischen den Anforderungen und den Funktionen bestehen und wie die Konkretisierung von Anforderungen schrittweise unter Einbeziehung der Entwicklungsinformationen erfolgen kann.

Das Verständnis von Zyklen und Iterationen im Requirements Engineering wird durch diese Arbeit verbessert. Die Ermittlung und Verwaltung von Anforderungen an das PSS ist von zyklischen Wechselwirkungen geprägt, die zu steigenden Kosten, Zeit- und Ressourcenaufwand führen. Die vorliegende Arbeit gibt einen umfassenden Überblick über mögliche Zyklen im Requirements Engineering für PSS. Desweiteren wurden die Probleme, die auf Grund der Zyklen auftreten, in die Anforderungen an den Requirements Engineering Ansatz aufgenommen und in der Entwicklung des Ansatzes berücksichtigt. Somit trägt die vorliegende Arbeit dazu bei, dass die Zyklen im Requirements Engineering schneller erkannt und entsprechend behandelt werden können.

Zuletzt trägt das Artefaktmodell, das im Rahmen des Requirements Engineering Ansatzes entwickelt wurde, zur Referenzmodellierung bei. Ein Referenzmodell ist „ein für einen ganzen Wirtschaftszweig erstelltes Modell, das allgemeingültigen Charakter haben soll. Es dient als Ausgangslösung zur Entwicklung unternehmensspezifischer Modelle“ ((Becker/Schütte 1999) zitiert in (Krcmar 2010, 121)). Ähnlich, wie die in der Literatur vorgeschlagenen Artefaktmodelle, bspw. Requirements Engineering Reference Model (REM) (Geisberger et al. 2006; Geisberger 2005), Requirements Abstraction Model (RAM) (Gorschek/Wohlin 2006) und Requirements Engineering und Management für Business Informations-Systeme (REMBIS) (Méndez Fernández/Kuhrmann 2009), stellt das Artefaktmodell für die Anforderungen an PSS ein Referenzmodell dar. Es beschreibt nämlich, losgelöst von unternehmensspezifischen Erfordernissen, eine allgemeingültige Struktur von Anforderungen für PSS. Daher kann es als Grundlage für die Entwicklung zugeschnittener Artefaktmodelle in spezifischen An-

wendungsumgebungen eingesetzt werden, wie bspw. für Software-as-a-Service das ein PSS darstellt, das sich vor allem aus Software und Dienstleistungen zusammensetzt.

9.3 Grenzen der Arbeit und weiterer Forschungsbedarf

Der in dieser Arbeit entwickelte Requirements Engineering Ansatz für PSS ermöglicht eine ganzheitliche Betrachtung von Anforderungen an PSS sowie eine integrierte Betrachtung der domänenspezifischen Anforderungen an die einzelnen Komponenten des PSS. Der Ansatz beschränkt sich aber auf die früheren Phasen des Lebenszykluses von PSS, nämlich bis die Anforderungen an die Entwicklung übergeben werden. Forschungsbedarf besteht in der Analyse der Umsetzung von Anforderungen, indem die Traceability der Umsetzung der einzelnen domänenspezifischen Anforderungen bis zur Bereitstellung des PSS an den Kunden gewährleistet wird. Traceability bedeutet somit die Verfolgung und die Nachvollziehbarkeit des Lebenszyklus jeder Anforderung, samt allen Änderungen und Anpassungen über die Phasen ihrer Umsetzung, bis sie in die Ist-Eigenschaften des PSS mündet.

Die Verwaltung und Pflege der Anforderungen während der Umsetzung des PSS ist ein weiterer Bereich, der detaillierter untersucht werden muss. Während der Umsetzung der Anforderungen treten verschiedene Zyklen auf, wie bspw. Produkttechnologiewechsel oder Gesetzesänderungen. Diese führen dazu, dass bestimmte Lösungsanforderungen nicht umsetzbar sind, oder dass die Lösungsanforderungen während der Umsetzung geändert werden müssen. Die Anforderungsänderungen ziehen eine Reihe von Iterationen zum Klären der Änderungen zwischen den Unternehmensbereichen nach sich. Dazu müssen die Auswirkungen der Änderungen auf die initialen Anforderungen und die von den anderen Domänen umzusetzenden Domänen-Anforderungen identifiziert werden. An dieser Stelle müssen Methoden zum Abschätzen der Auswirkungen der Änderungen entwickelt werden, die Aufschluss über mögliche Zusatzkosten und Verzögerungen bei der Bereitstellung des PSS an den Kunden geben.

Ein weiterer nicht in dieser Arbeit adressierter Punkt ist das Variantenmanagement von PSS. Verschiedene Ausprägungen des PSS, die den gleichen Zweck erfüllen, sich aber in mindestens einer Eigenschaft unterscheiden, werden als Varianten bezeichnet. Die Existenz von verschiedenen Varianten sollte bereits im Requirements Engineering berücksichtigt werden, um die Varianten in den Anforderungen abbilden zu können. Die Unterscheidung von verschiedenen Varianten bereits in der Anforderungsphase ist für die Modularisierung und Wiederverwendung von Komponenten eines PSS von entscheidender Rolle. Angelehnt an die bestehenden Ansätze zur Modularisierung und Standardisierung von Dienstleistungen könnten ähnliche Verfahren für PSS entwickelt werden.

Ein weiterer Forschungsbedarf besteht in der Anpassung des entwickelten Requirements Engineering Ansatzes an weitere Formen von PSS. Beispielsweise existieren PSS, die keine Hardwarekomponenten enthalten, sondern nur eine Kombination von Software und Dienstleistungen darstellen und in Form von Software-as-a-Service (SaaS) angeboten werden. In diesem Fall müssen bspw. die Abstraktionsebenen des Artefaktmodells an den Entwicklungsprozess von SaaS angepasst werden und das dazugehörige Vorgehensmodell dementsprechend adaptiert werden. Desweiteren müssen geeignete Anforderungs- und Entwicklungsartenfakte und die dazugehörigen Taxonomien ausgewählt werden, die den Betrachtungsgegenstand von SaaS abdecken. Entsprechend ist der Requirements Engineering Ansatz unter geeigneten Bedingungen für den Einsatz von SaaS zu evaluieren. Basierend auf den Erfahrungen der Anpassung des Requirements Engineering Ansatzes an SaaS, kann ein generischer

Ansatz zum Adaption des Artefakt- und Vorgehensmodells an verschiedene Gegebenheiten, wie bspw. Branchen, Projektgrößen, etc. entwickelt werden.

Der entwickelte Ansatz bietet im Rahmen des Vorgehensmodells Techniken an, die zur Durchführung der Aktivitäten des Vorgehensmodells eingesetzt werden. Weiterer Forschungsbedarf besteht darin, diese Techniken gezielter auf einzelne Projekte zuzuschneiden. Somit soll bspw. geklärt werden, welche Technik bei welchem Typ von PSS (bspw. ohne die Hardware-Komponente wie SaaS) eingesetzt werden sollen. Das schließt mit ein, dass untersucht wird mit welchen Vor- und Nachteilen die einzelnen Techniken bei ihrem Einsatz zur Ermittlung und Verwaltung von Anforderungen an PSS verbunden sind. Dabei sollen der Aufwand beim Einsatz der Technik und das zu erwartende Ergebnis mit in die Analyse einbezogen werden. Das Ergebnis der Analyse stellt ein Technikbaukasten dar, der in der Lage ist, für jede Aktivität des Requirements Engineering, unter Berücksichtigung der genannten Gegebenheiten, eine geeignete Technik anzubieten.

Eine weitere zukünftige Forschungsrichtung für das Requirements Engineering für PSS ist die Integration von formalen Methoden. Die Anforderungen der einzelnen Artefakte könnten durch formale Beschreibungstechniken ausgedrückt werden. Dadurch erschließt sich die Möglichkeit Verifikationstechniken, wie beispielsweise Model Checking anzuwenden. Im Bereich der Software-Anforderungen kann auf diese Weise ein nahtloser Übergang in die modellbasierte Softwareentwicklung geschaffen werden. Im Bereich der Produkthanforderungen können dadurch Simulationstechniken aus dem FEM Bereich angewandt werden.

Der Requirements Engineering Ansatz in dieser Arbeit wurde in drei Fallstudien angewendet, um seine Vor- und Nachteile zu zeigen. Zwei der Fallstudien bestanden in einer retrospektiven Anwendung auf realen Fällen aus der Praxis. Zukünftig könnten weitere intensive Fallstudien durchgeführt werden, um die Anwendbarkeit und Nützlichkeit des Ansatzes in der Praxis weiter zu untermauern. Dabei sollen auch Verbesserungspotentiale des Ansatzes bezüglich der Lösung von praxisrelevanten Problemen bei der Entwicklung von PSS aufgedeckt werden. Auf diese Weise kann der Ansatz weiter verbessert und an die Bedürfnisse der Praxis angepasst werden.

Literaturverzeichnis

- Abramovici, M.; Schulte, S. (2005):** Lifecycle Management von Produkt-Service-Systemen (PSS) für einen maximierten Kundennutzen. *Gemeinsames Kolloquium Konstruktionstechnik*. Magdeburg
- Abramovici, M.; Schulte, S. (2007):** Optimising customer satisfaction by integrating the customer's voice into product development. *International Conference on Engineering Design, ICED'07*. Paris, France.
- Adler, T.R. (1994):** The innovation process: interpreting customer requirements. In: IEEE Aerospace and Electronic Systems Magazine, Vol. 9 (1994) No. 6, pp. 17 - 25.
- Ahmad, S. (2008):** Negotiation in the Requirements Elicitation and Analysis Process. Paper presented at the 19th Australian Conference on Software Engineering, ASWEC 2008, Perth, Australia, pp. 683-689.
- Ahrens, G. (2000):** Das Erfassen und Handhaben von Produkthanforderungen - methodische Voraussetzungen und Anwendung in der Praxis. Dissertation, Technische Universität Berlin 2000.
- Akao, Y. (1990):** Quality Function Deployment: Quality Function Deployment - Integrating Customer Requirements into Product Design, Productivity Press 1990.
- Alavi, M. (1984):** An Assessment of the Prototyping Approach to Information Systems. In: Communications of the ACM, Vol. 27 (1984) No. 4, pp. 556-563.
- Alexander, I. (2003):** Misuse cases: use cases with hostile intent. In: Software, IEEE, Vol. 20 (2003) No. 1, pp. 58-66.
- Almefelt, L.; Andersson, F.; Nilsson, P.; Malmqvist, J. (2003):** Exploring requirements management in the automotive industry. *International Conference On Engineering Design, ICED 03*. Stockholm.
- Almefelt, L.; Berglund, F.; Nilsson, P.; Malmqvist, J. (2006):** Requirements management in practice: findings from an empirical study in the automotive industry. In: Research in Engineering Design, Vol. 17 (2006) No. 3, pp. 113-134.
- Alpar, P.; Grob, H.L.; Weimann, P.; Winter, R. (2008):** Anwendungsorientierte Wirtschaftsinformatik: Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen. (5 ed.), Vieweg+Teubner 2008.
- Alspaugh, T.A.; Sim, S.E.; Winbladh, K.; Diallo, M.H.; Naslavsky, L.; Ziv, H.; Richardson, D.J. (2007):** Clarity for Stakeholders: Empirical Evaluation of ScenarioML, Use Cases, and Sequence Diagrams. Paper presented at the Fifth International Workshop on Comparative Evaluation in Requirements Engineering, CERE '07, New Delhi, pp. 1-10.
- Alter, S. (2008):** Defining information systems as work systems: implications for the IS field. In: European Journal of Information Systems, Vol. 17 (2008) No. 5, pp. 448-469.
- An, Y.; Lee, S.; Park, Y. (2008):** Development of an integrated product-service roadmap with QFD: A case study on mobile communications. In: International Journal of Service Industry Management, Vol. 19 (2008) No. 5, pp. 621 - 638.

- Anderson, J.S.; Durney, B. (1993):** Using scenarios in deficiency-driven requirements engineering. Paper presented at the International Symposium on Requirements Engineering, San Diego, USA, pp. 134-141.
- Andriole, S.J. (1994):** Fast, cheap requirements: Prototype, or else! In: IEEE Software, Vol. 11 (1994) No. 2, pp. 85-87.
- Anwer, S.; Ikram, N. (2006):** Goal Oriented Requirement Engineering: A Critical Study of Techniques. Paper presented at the 13th Asia Pacific Software Engineering Conference, APSEC 2006 Bangalore, India, pp. 121-130.
- Arnold, V.; Dettmering, H.; Engel, T.; Karcher, A. (2005):** Product Lifecycle Management beherrschen. (1 ed.), Springer, Berlin 2005.
- Atlee, J.; Wieringa, R. (2006):** RE 05: Engineering Successful Products. In: Software, IEEE, Vol. 23 (2006) No. 3, pp. 16-18.
- Atteslander, P. (2008):** Methoden der empirischen Sozialforschung. (12 ed.), Schmidt (Erich), Berlin 2008.
- Aurich, J.C.; Schweitzer, E.; Fuchs, C. (2007):** Life Cycle Management of Industrial Product-Service Systems. In Takata, S.; Umeda, Y. (Eds.), *Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses: Proceedings of the 14th CIRP Conference on Life Cycle Engineering, Waseda University, Tokyo, Japan*, (pp. 171-176): Springer.
- Aurum, A.; Wohlin, C. (2005a):** Engineering and Managing Software Requirements (1 ed.), Springer, Berlin 2005a.
- Aurum, A.; Wohlin, C. (2005b):** Requirements Engineering: Setting the Context. In: Engineering and Managing Software Requirements. Eds.: Aurum, A.; Wohlin, C., 1 (ed.). Springer, Berlin 2005b, pp. 1-15.
- Backhaus, K.; Weiber, R. (1993):** Das industrielle Anlagengeschäft - ein Dienstleistungsgeschäft? In: Industrielle Dienstleistungen. Eds.: Simon, H., Stuttgart 1993, pp. 67-84.
- Baines, T.S.; Lightfoot, H.W.; Evans, S.; Neely, A.; Greenough, R.; Peppard, J.; Roy, R.; Shehab, E.; Braganza, A.; Tiwari, A.; Alcock, J.R.; Angus, J.P.; Bastl, M.; Cousens, A.; Irving, P.; M Johnson; Kingston, J.; Lockett, H.; Martinez, V.; Michele, P.; Tranfield, D.; Walton, I.M.; Wilson, H. (2007):** State-of-the-art in product-service systems. In: Journal of Engineering Manufacture, Vol. 221 (2007), pp. 1543-1552.
- Balzert, H. (1998):** Lehrbuch der Software-Technik, Software-Management, Software-Qualitätssicherung und Unternehmensmodellierung, Spektrum Akademischer Verlag 1998.
- Balzert, H. (2000):** Lehrbuch der Software-Technik, Software-Entwicklung. (2 ed.), Spektrum Akademischer Verlag 2000.
- Baskerville, R. (2008):** What design science is not. In: European Journal of Information Systems, Vol. 17 (2008) No. 5, pp. 441-443.
- Becker, J.; Beverungen, D.; Knackstedt, R. (2009):** Wertschöpfungsnetzwerke von Produzenten und Dienstleistern. In: Wertschöpfungsnetzwerke: Konzepte für das Netzwerkmanagement und Potenziale aktueller Informationstechnologien. Eds.: Becker, J.; Knackstedt, R.; Pfeiffer, D., 1 (ed.). Springer 2009.

- Becker, J.; Holten, R.; Knackstedt, R.; Niehaves, B. (2003):** Forschungsmethodische Positionierung in der Wirtschaftsinformatik – epistemologische, ontologische und linguistische Leitfragen-. Institut für Wirtschaftsinformatik, 2003.
- Becker, J.; Schütte, R. (1999):** Handelsinformationssysteme, Moderne Industrie 1999.
- Beiter, K.A.; Ishii, K.K.; Karandikar, H.H. (2006):** Customer requirements management: methodology selection and Deployment guide. *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Philadelphia, Pennsylvania, USA.
- Benbasat, I.; Goldstein, D.K.; Mead, M. (1987):** The Case Research Strategy in Studies of Information Systems. In: *MIS Quarterly*, Vol. 11 (1987) No. 3, pp. 369-386.
- Berander, P.; Andrews, A. (2005):** Requirements Prioritization. In: *Engineering and Managing Software Requirements* Eds.: Aurum, A.; Wohlin, C., 1 (ed.). Springer, Berlin 2005, pp. 69-94.
- Berenbach, B. (2006):** Introduction to Product Line Requirements Engineering. *10th International Software Product Line Conference*. Baltimore, USA.
- Berenbach, B.; Paulish, D.J.; Kazmeier, J.; Rudorfer, A. (2009):** Software & Systems Requirements Engineering: In Practice, McGraw-Hill Professional 2009.
- Berkovich, M. (2010):** Requirements Engineering for IT-Enabled Product Service Systems. *The 18th IEEE International Requirements Engineering Conference, Doctoral Consortium*. Sydney, Australia.
- Berkovich, M.; Esch, S.; Leimeister, J.M.; Krcmar, H. (2009a):** Requirements engineering for hybrid products as bundles of hardware, software and service elements – a literature review. Paper presented at the 9. Internationale Tagung Wirtschaftsinformatik (WI), Wien, Österreich, pp. 727-736.
- Berkovich, M.; Esch, S.; Leimeister, J.M.; Krcmar, H. (2010a):** Towards Requirements Engineering for “Software as a Service”. Paper presented at the Multikonferenz Wirtschaftsinformatik (MKWI 2010), Göttingen, pp. 517-528.
- Berkovich, M.; Esch, S.; Mauro, C.; Leimeister, J.M.; Krcmar, H. (2011a):** Towards an Artifact Model for Requirements to IT-enabled Product Service Systems. *10. Internationale Tagung Wirtschaftsinformatik, WI2011*. Zürich, Schweiz.
- Berkovich, M.; Helms, B.; Kortler, S.; Leimeister, J.M.; Krcmar, H.; Shea, K.; Lindemann, U. (2011b):** Von Anforderungslisten zum konzeptionellen Design: Funktionsbasierte Analyse von Anforderungen an IT-gestützte Product Service Systems. In: *Informatik Spektrum*, (2011b) - submitted.
- Berkovich, M.; Hoffmann, A.; Leimeister, J.M.; Krcmar, H. (2011c):** Analysis of Requirements Engineering Techniques for IT-enabled Product-Service-Systems. Paper presented at the Workshop on Requirements Engineering for Systems and Systems-of-Systems held in conjunction with the International Requirements Engineering Conference, Trient, Italy (accepted for publication).
- Berkovich, M.; Leimeister, J.M.; Krcmar, H. (2009b):** An empirical exploration of requirements engineering for hybrid products. Paper presented at the XVIIth European Conference on Information Systems (ECIS), Verona, pp. 2570-2582.
- Berkovich, M.; Leimeister, J.M.; Krcmar, H. (2009c):** Suitability of Product Development Methods for Hybrid Products as Bundles of Classic Products, Software and Service

- Elements. *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*. San Diego.
- Berkovich, M.; Leimeister, J.M.; Krcmar, H. (2010b):** Ein Bezugsrahmen für Requirements Engineering hybrider Produkte. Paper presented at the Multikonferenz Wirtschaftsinformatik (MKWI 2010), Göttingen, pp. 2017-2029.
- Berkovich, M.; Leimeister, J.M.; Krcmar, H. (2011d):** Requirements Engineering für Product Service Systems - eine State of the Art Analyse. In: *Wirtschaftsinformatik*, (2011d) - accepted for publication.
- Berkovich, M.; Mauro, C.; Leimeister, J.M.; Weyde, F.; Krcmar, H. (2011e):** Towards Cycle-Oriented Requirements Engineering. *10. Internationale Tagung Wirtschaftsinformatik, WI2011*. Zürich, Schweiz.
- Bertsche, B.; Göhner, P.; Jensen, U.; Schinköthe, W.; Wunderlich, H.-J. (2009):** Zuverlässigkeit mechatronischer Systeme: Grundlagen und Bewertung in frühen Entwicklungsphasen, Springer, Berlin 2009.
- Beus-Dukic, L.; Alexander, I. (2008):** Learning How To Discover Requirements. Paper presented at the Requirements Engineering Education and Training, REET '08, pp. 12-14.
- Beverungen, D.; Knackstedt, R.; Müller, O. (2008):** Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung – Eine Konzeptionsmethode und ihre Anwendung am Beispiel des Recyclings elektronischer Geräte. In: *Wirtschaftsinformatik*, Vol. 50 (2008) No. 3.
- Blinn, N.; Nüttgens, M.; Schlicker, M.; Thomas, O.; Walter, P. (2008):** Lebenszyklusmodelle hybrider Wertschöpfung: Modellimplikationen und Fallstudie an einem Beispiel des Maschinen- und Anlagenbaus. Paper presented at the MKWI, München, pp. 711-722.
- Bodendorf, F. (1999):** *Wirtschaftsinformatik im Dienstleistungsbereich*. (1 ed.), Springer, Berlin 1999.
- Boehm, B.; Basili, V.R. (2001):** Top 10 list [software development]. In: *Computer*, Vol. 34 (2001) No. 1, pp. 135 - 137.
- Boehm, B.; Bose, P.; Horowitz, E.; Lee, M.J. (1994):** Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach. *The 17th International Conference on Software Engineering (ICSE 95)*. Seattle, USA.
- Boehm, B.; Grünbacher, P.; Briggs, R.O. (2001):** Developing Groupware for Requirements Negotiation: Lessons Learned. In: *IEEE Software*, Vol. 18 (2001) No. 3, pp. 46-55.
- Boehm, B.; In, H. (1996):** Identifying quality-requirement conflicts. Paper presented at the Second International Conference on Requirements Engineering, Colorado Springs, CO, USA, pp. 218.
- Boehm, B.W. (1984):** Verifying and Validating Software Requirements and Design Specifications. In: *Software*, IEEE, Vol. 1 (1984) No. 1, pp. 75 - 88.
- Boehm, B.W. (1996):** Identifying quality-requirement conflicts. In: *Software*, IEEE, Vol. 13 (1996) No. 2, pp. 25 - 35.
- Boghnim, N.; Yannou, B. (2005):** Using blueprinting method for developing productservice systems. *International Conference on Engineering Design, ICED'05*. Melbourne.

- Böhm, R.; Fuchs, E. (2002):** System-Entwicklung in der Wirtschaftsinformatik. (5 ed.), Vdf Hochschulverlag 2002.
- Böhm, T. (2003):** Modularisierung von IT-Dienstleistungen - Gegenstand und Konzept einer Methode für das Service Engineering. Dissertation, Universität Hohenheim 2003.
- Böhm, T.; Krcmar, H. (2003):** Modulare Servicearchitekturen. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer 2003, pp. 376-401.
- Böhm, T.; Krcmar, H. (2006):** Komplexitätsmanagement als Herausforderung hybrider Wertschöpfung im Netzwerk. In: Innovative Kooperationsnetzwerke. Eds.: Wojda, F.; Barth, A., 2006.
- Böhm, T.; Krcmar, H. (2007):** Hybride Produkte: Merkmale und Herausforderungen. In: Wertschöpfungsprozesse bei Dienstleistungen: Forum Dienstleistungsmanagement. Eds.: Bruhn, M.; Stauss, B. Gabler 2007, pp. 240-255.
- Böhm, T.; Langer, P.; Schermann, M. (2008):** Systematische Überführung von kundenspezifischen IT-Lösungen in integrierte Produkt-Dienstleistungsbausteine mit der SCORE-Methode. In: Wirtschaftsinformatik, Vol. 50 (2008) No. 3, pp. 196-207.
- Bonnemeier, S.; Ihl, C.; Reichwald, R. (2007):** Wertschaffung und Wertaneignung bei hybriden Produkten - Eine prozessorientierte Betrachtung, 2007.
- Booch, G.; Rumbaugh, J.; Jacobson, I. (2007):** Das UML Benutzerhandbuch, Addison-Wesley, München 2007.
- Born, M.; Holz, E.; Kath, O. (2004):** Softwareentwicklung mit UML 2. Die "neuen" Entwurfstechniken UML 2, MOF 2 und MDA, Addison-Wesley 2004.
- Bortz, J.; Döring, N. (2006):** Forschungsmethoden und Evaluation. für Human- und Sozialwissenschaftler: Für Human- Und Sozialwissenschaftler (4 ed.), Springer, Berlin 2006.
- Botta, C. (2007):** Rahmenkonzept zur Entwicklung von Product-Service Systems, Eul-Verlag 2007.
- Böttcher, M.; Meyer, K. (2004):** IT-basierte Dienstleistungen. In: Computer Aided Engineering für IT-basierte Dienstleistungen. Eds.: Fähnrich, K.-P.; Meiren, T. Eigenverlag der Universität Leipzig, Leipzig 2004.
- Bowen, P.L.; Heales, J.; Vongphakdi, M.T. (2002):** Reliability factors in business software: volatility, requirements and end-users. In: Information Systems Journal, Vol. 12 (2002) No. 3, pp. 185-213.
- Boyta, T.; Harvey, M. (1997):** Classification of industrial services: A model with strategic implications. In: Industrial Marketing Management, Vol. 26 (1997) No. 4, pp. 291-300.
- Brandt-Pook, H.; Kollmeier, R. (2008):** Softwareentwicklung kompakt und verständlich: Wie Softwaresysteme entstehen. (1 ed.), Vieweg+Teubner 2008.
- Breiting, A.; Flemming, M. (1993):** Theorie und Methoden des Konstruierens Springer, Berlin 1993.
- Brereton, P.; Kitchenham, B.A.; Budgen, D.; Turner, M.; Khahil, M. (2007):** Lessons from applying the systematic literature review process within the software engineering domain. In: Journal of Systems and Software, Vol. 4 (2007) No. 80, pp. 571-583.

- Breu, R. (2001):** Objektorientierter Softwareentwurf: Integration mit UML. (1 ed.), Springer, Berlin 2001.
- Briggs, R.O.; Grünbacher, P. (2002):** EasyWinWin: Managing Complexity in Requirements Negotiation with GSS. *The 35th Hawaii International Conference on System Sciences*. Hawaii
- Brinkkemper, S. (1996):** Method engineering: engineering of information systems development methods and tools. In: *Information and Software Technology*, Vol. 38 (1996) No. 4, pp. 275-280.
- Brocke, J.v. (2003):** Referenzmodellierung: Gestaltung und Verteilung von Konstruktionsprozessen. (1 ed.), Logos Berlin, Berlin 2003.
- Brooks, F.P., Jr. (1987):** No Silver Bullet Essence and Accidents of Software Engineering. In: *Computer*, Vol. 20 (1987) No. 4, pp. 10 - 19.
- Browning, T.R. (2001):** Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. In: *IEEE Transaction on Engineering Management*, Vol. 48 (2001) No. 3, pp. 292-306.
- Broy, M.; Feilkas, M.; Grünbauer, J.; Gruler, A.; Harhurin, A.; Judith Hartmann; Penzenstadler, B.; Schätz, B.; Wild, D. (2008):** Umfassendes Architekturmodell für das Engineering eingebetteter Software-intensiver Systeme. Institut für Informatik, TUM, 2008.
- Broy, M.; Geisberger, E.; Kazmeier, J.; Rudorfer, A.; Beetz, K. (2007):** Ein Requirements-Engineering-Referenzmodell In: *Informatik-Spektrum*, Vol. 30 (2007) No. 3, pp. 127-142.
- Brugger, R. (2005):** IT-Projekte strukturiert realisieren: Situationen analysieren, Lösungen konzipieren - Vorgehen systematisieren, Sachverhalte visualisieren - UML und EPKs nutzen. (2 ed.), Vieweg+Teubner 2005.
- Bruhn, M. (2008):** Qualitätsmanagement für Dienstleistungen: Grundlagen, Konzepte, Methoden (7 ed.), Springer, Berlin 2008.
- Bruhn, M.; Georgi, D. (2005):** Services Marketing: Managing the Service Value Chain, Financial Times Prentice Hall, Harlow 2005.
- Budgen, D.; Brereton, P. (2006):** Performing systematic literature reviews in software engineering. Paper presented at the International Conference on Software Engineering; Proceedings of the 28th international conference on Software engineering, Shanghai, China, pp. 1051 - 1052.
- Buede, D.M. (2009):** The Engineering Design of Systems: Models and Methods. (2 ed.), John Wiley & Sons 2009.
- Bulent, G.; Atila, E. (2004):** Requirement management and axiomatic design. In: *Journal of Integrated Design and Process Science*, Vol. 8 (2004) No. 4, pp. 19-31.
- Bullinger, H.-J. (1999):** Entwicklung innovativer Dienstleistungen. In: *Dienstleistungen. Innovation für Wachstum und Beschäftigung. Herausforderungen des internationalen Wettbewerbs*. Eds.: Bullinger, H.-J. Gabler 1999, pp. 49-65.
- Bullinger, H.-J.; Schreiner, P. (2003):** Service Engineering: Ein Rahmenkonzept für die systematische Entwicklung von Dienstleistungen. In: *Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen*. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer, Berlin, Heidelberg, New York 2003, pp. 53-84.

- Bunse, C.; von Kneten, A. (2002):** Vorgehensmodelle kompakt, Spektrum Akademischer Verlag, Heidelberg 2002.
- Burianek, F.; Christoph, I.; Sebastian, B.; Ralf, R. (2007):** Typologisierung hybrider Produkte – Ein Ansatz basierend auf der Komplexität der Leistungserbringung. In: Arbeitsberichte des Lehrstuhls für Betriebswirtschaftslehre – Information, Organisation und Management an der TU München, (2007) No. 1.
- Burianek, F.; Reichwald, R. (2009):** Vertrags- und Preisgestaltung bei hybriden Produkten. In: Hybride Wertschöpfung: Konzepte, Methoden und Kompetenzen für die Preis- und Vertragsgestaltung. Eds.: Reichwald, R.; Krmar, H.; Nippa, M. Eul Verlag, Lohmar 2009.
- Burr, W.; Stephan, M. (2006):** Dienstleistungsmanagement: Innovative Wertschöpfungskonzepte für Dienstleistungssektor, Kohlhammer 2006.
- Buschermöhle, R.; Eckhoff, H.; Josko, B. (2006):** Success - Erfolgs- und Misserfolgskriterien bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland, BIS-Verlag 2006.
- Büttemeyer, W. (1995):** Wissenschaftstheorie für Informatiker, Spektrum Akademischer Verlag, Heidelberg 1995.
- Byrd, T.A.; Cossick, K.L.; Zmud, R.W. (1992):** A synthesis of research on requirements analysis and knowledge acquisition techniques. In: MIS Quarterly, Vol. 16 (1992) No. 1, pp. 117-138.
- Chahadi, Y.; Wäldele, M.; Birkhofer, H. (2007):** QFD – A link between customer requirements and product properties. *International Conference on Engineering Design, ICED'07*. Paris, France.
- Cheng, B.H.C.; Atlee, J.M. (2007):** Research Directions in Requirements Engineering. *Future of Software Engineering (FOSE'07)*. Minneapolis, USA.
- Chmielewicz, K. (1979):** Forschungskonzeption der Wirtschaftswissenschaft. (2 ed.), C. E. Poeschel Verlag, Stuttgart 1979.
- Clements, P.; Bass, L. (2010):** Using Business Goals to Inform a Software Architecture. Paper presented at the 18th IEEE International Requirements Engineering Conference (RE), Sydney, Australia, pp. 69-78.
- Cook, D.P.; Goh, C.-H.; Chung, C.H. (1999):** Service Typologies: A State of the Art Survey. In: Production and Operations Management, Vol. 8 (1999) No. 3, pp. 318 - 338.
- Corsten, H. (2001):** Dienstleistungsmanagement (4 ed.), Oldenbourg Wissensch.Vlg 2001.
- Coughlan, J.; Macredie, R.D. (2002):** Effective Communication in Requirements Elicitation: A Comparison of Methodologies. In: Requirements Engineering Journal, Vol. 7 (2002) No. 2, pp. 47-60.
- Cox, K.; Niazi, M.; Verner, J. (2009):** Empirical study of Sommerville and Sawyer's requirements engineering practices. In: Software, IET, Vol. 3 (2009) No. 5, pp. 339-355.
- Cross, N. (2008):** Engineering Design Methods: Strategies for Product Design. (4 ed.), Wiley & Sons 2008.

- Dahlstedt, A.G.; Persson, A. (2005):** Requirements Interdependencies: State of the Art and Future challenges In: Engineering and Managing Software Requirements. Eds.: Aurum, A.; Wohlin, C., 1 (ed.). Springer, Berlin 2005.
- Damian, D.; Chisan, J. (2006):** An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management. In: IEEE Transactions on Software Engineering, Vol. 32 (2006) No. 7, pp. 433 - 453.
- Dangelmaier, W.; Emmrich, A.; Gajewski, T.; Heidenreich, J. (2008):** Ein Referenzmodell zur Beschreibung der Geschäftsprozesse von After-Sales-Dienstleistungen. In: Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung. Eds.: Becker, J.; Delfmann, P., 1 (ed.). Springer 2008.
- Danilovic, M.; Sandkull, B. (2005):** The use of dependence structure matrix and domain mapping matrix in managing uncertainty in multiple project situations. In: International Journal of Project Management, Vol. 23 (2005) No. 3, pp. 193-203.
- Danner, S. (1996):** Ganzheitliches Anforderungsmanagement. Dissertation, Technische Universität München 1996.
- Dano, B.; Briand, H.; Barbier, F. (1996):** Progressing towards object-oriented requirements specifications by using the use case concept. *IEEE Symposium and Workshop on Engineering of Computer-Based Systems*. Friedrichshafen, Germany.
- Dardenne, A.; Lamsweerde, A.v.; Fickas, S. (1993):** Goal-directed requirements acquisition. In: Science of Computer Programming, Vol. 20 (1993), pp. 3 - 50.
- Darimont, R.; Delor, E.; Massonet, P.; van Lamsweerde, A. (1997):** GRAIL/KAOS: An Environment for Goal-Driven Requirements Engineering. *The 19th International Conference on Software Engineering* (pp. 612-613).
- Darke, P.; Shanks, G. (1996):** Stakeholder viewpoints in requirements definition: A framework for understanding viewpoint development approaches. In: Requirements Engineering, Vol. 1 (1996) No. 2.
- Davies, A. (2004):** Moving base into high-value integrated solutions: a value stream approach. In: Industrial & Corporate Change, Vol. 13 (2004) No. 5, pp. 727-756.
- Davis, A.; Dieste, O.; Hickey, A.; Juristo, N.; Moreno, A.M. (2006a):** Effectiveness of Requirements Elicitation Techniques: Empirical Results Derived from a Systematic Review. Paper presented at the The 14th IEEE International Conference Requirements Engineering, pp. 179-188.
- Davis, A.M. (1992):** Operational prototyping: a new development approach. In: IEEE Software, Vol. 9 (1992) No. 5, pp. 70-78.
- Davis, C.J.; Fuller, R.M.; Tremblay, M.C.; Berndt, D.J. (2006b):** Communication Challenges in Requirements Elicitation and the Use of the Repertory Grid Technique. In: Journal of Computer Information Systems, Vol. 47 (2006b), pp. 78-86.
- Davis, G.B. (1982):** Strategies for information requirements determination. In: IBM Systems Journal, Vol. 21 (1982) No. 1, pp. 4-30.
- De Landtsheer, R.; Letier, E.; van Laamsweerde, A. (2003):** Deriving Tabular Event-Based Specifications from Goal-Oriented Requirements Models. Paper presented at the 11th IEEE International Conference on Requirements Engineering, USA, pp. 200-210.

- Denger, C.; Olsson, T. (2005):** Quality Assurance in Requirements Engineering. In: Engineering and Managing Software Requirements Eds.: Aurum, A.; Wohlin, C., 1 (ed.). Springer, Berlin 2005, pp. 163-185.
- Deubel, T.; Steinbach, M.; Weber, C. (2005):** Requirement- and cost-driven product development process. *International Conference on Engineering Design, ICED'05*. Melbourne, Australia.
- Dix, A.; Finlay, J.; Abowd, G.D.; Beale, R. (2003):** Human Computer Interaction. (3 ed.), Prentice Hall 2003.
- Dix, A.J. (1994):** Computer-supported cooperative work - a framework. In: Design Issues in CSCW. Eds.: Rosenburg, D.; Hutchison, C. Springer Verlag 1994, pp. 23-37.
- Dodgson, M.; Gann, D.; Salter, A. (2008):** The Management of Technological Innovation: Strategy and Practice: The Strategy and Practice, Oxford University Press, 2008.
- Dodrige, M. (1999):** Generic skill requirements for engineers the 21st century. In: 29th ASEE/IEEE Frontiers Education Conference, (1999) No. Vol 3. (1999), pp. 13A9/9-13A9/14.
- Doolin, B.; McLeod, L. (2005):** Towards critical interpretivism in IS research. In: Handbook of Critical Information Systems Research: Theory And Application. Eds.: Howcroft, D.; Trauth, E.M. Edward Elgar Publishing Ltd 2005, pp. 244-271.
- Doster, D.; Roegner, E. (2000):** Setting the pace with solutions. In: Marketing Management, Vol. 9 (2000) No. 1, pp. 51-54.
- Dreyer, A. (2003):** Kundenorientierung in touristischen Destinationen. In: Kundenmanagement als Erfolgsfaktor: Grundlagen des Tourismusmarketing. Eds.: Hinterhuber, H.H.; Pechlaner, H.; Kaiser, M.-O.; Matzler, K. Schmidt (Erich), Berlin 2003, pp. 29-49.
- Dul, J.; Weerdmeester, B. (2008):** Ergonomics for Beginners: A Quick Reference Guide. (3 ed.), Crc Pr Inc 2008.
- Durraui, T.S.; Forbes, S.M.; Carrie, A.S. (2000):** Extending the balanced scorecard for technology strategy development. Paper presented at the Engineering Management Society, 2000. Proceedings of the 2000 IEEE, Albuquerque, NM, USA pp. 120-125.
- Easterbrook, S. (1991):** Handling conflict between domain descriptions with computer-supported negotiation. In: Knowledge Acquisition archive, Vol. 3 (1991) No. 3, pp. 255 - 289.
- Eben, K.G.M.; Lindemann, U. (2010):** Structural Analysis of Requirements – Interpretation of structural Criteria. *The 12th International Dependency and Structure Modelling Conference*. Cambridge, UK.
- Ebert, C. (2005):** Systematisches Requirements Management. (1 ed.), Dpunkt Verlag 2005.
- Edvardsson, B.; Olsson, J. (1996):** Key Concepts for New Service Development. In: Service Industries Journal, Vol. 16 (1996) No. 2, pp. 140-164.
- Ehrlenspiel, K. (2002):** Integrierte Produktentwicklung (2 ed.), Hanser Fachbuchverlag 2002.
- Ehrlenspiel, K.; Kiewert, A.; Lindemann, U. (2005):** Kostengünstig Entwickeln und Konstruieren. Kostenmanagement bei der integrierten Produktentwicklung. (5 ed.), Springer-Verlag GmbH 2005.

- Engelhardt, W.H.; Kleinaltenkamp, M.; Reckenfelderbäumer, M. (1993):** Leistungsbündel als Absatzobjekte. In: Zeitschrift für Betriebswirtschaftliche Forschung, Vol. 45 (1993) No. 5, pp. 394 - 426.
- Engelke, M. (1997):** Qualität logistischer Dienstleistungen, Schmidt (Erich), Berlin 1997.
- Engeln, W. (2006):** Methoden der Produktentwicklung. (1 ed.), Oldenbourg Industrieverlag, München 2006.
- Ericson, Å.; Larsson, A.; Larsson, T.; Larsson, M. (2007):** Need driven product development in team-based projects. *International Conference on Engineering Design, ICED'07*. Paris, France.
- Ernst, G. (2008):** Integration von Produktion und Dienstleistungen - neuere Entwicklungen. In: DL2100.de, (2008).
- Esch, S.; Knebel, U.; Leimeister, J.M. (2010a):** Der Personal Health Manager: Ein IT-gestütztes Bewegungsprogramm als hybrides Produkt. In: *Hybride Wertschöpfung in der Gesundheitsförderung*. Eds.: Leimeister, J.M.; Krcmar, H.; Halle, M.; Möslein, K. Josef EUL Verlag, Köln 2010a.
- Esch, S.; Köbler, F.; Knebel, U.; Leimeister, J.M.; Krcmar, H. (2010b):** Der Personal Health Manager - ein IT-gestütztes Bewegungsprogramm. In Fährnrich, K.; Franczyk, B. (Eds.), *Informatik 2010: Service Science - Neue Perspektiven für die Informatik* (pp. 15-20). Leipzig, Germany: .
- Fähling, J.; Köbler, F.; Leimeister, J.M.; Krcmar, H. (2010):** From products to product-service systems: IT driven transformation of a medical equipment manufacturer to a customer-centric solution provider. *International Conference on Information Systems (ICIS)*. Saint Louis, Missouri.
- Fährnrich, K.-P.; Opitz, M. (2003):** Service Engineering - Entwicklungspfad und Bild einer jungen Disziplin. In: *Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen*. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer, Berlin, Heidelberg, New York 2003.
- Fettke, P.; Loos, P. (2004):** Entwicklung eines Bezugsrahmens zur Evaluierung von Referenzmodellen. Research Group Information Systems & Management, 2004.
- Finkelstein, A.; Dowell, J. (1996):** A comedy of errors: the London Ambulance Service case study Paper presented at the 8th International Workshop on Software Specification and Design, Schloss Velen pp. 2 - 4.
- Fließ, S. (2009):** Dienstleistungsmanagement. (1 ed.), Gabler 2009.
- Frank, U. (2000):** Evaluation von Artefakten in der Wirtschaftsinformatik. In: *Evaluation und Evaluationsforschung in der Wirtschaftsinformatik (Vol. 1)*. Eds.: Heinrich, L.J.; Häntschel, I. Oldenbourg 2000, pp. 35-49.
- Frank, U. (2008):** Herausforderungen der Wirtschaftsinformatik in Zeiten des Wandels. In: *Quo vadis Wirtschaftsinformatik?* Eds.: Jung, R.; Myrach, T., 1 (ed.). Gabler, Wiesbaden 2008.
- Freiling, J.; Gersch, M. (2007):** Kompetenztheoretische Fundierung dienstleistungsbezogener Wertschöpfungsprozesse. In: *Wertschöpfungsprozesse bei Dienstleistungen: Forum Dienstleistungsmanagement*. Eds.: Bruhn, M.; Stauss, B. Gabler 2007.

- Frietzsche, U.; Maleri, R. (2003):** Dienstleistungsproduktion. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer, Berlin, Heidelberg, New York 2003, pp. 195-225.
- Fritzsche, P. (2007):** Innovationsmanagement für Dienstleistungen durch Service Engineering: Bedeutung und Ablauf der systematischen Dienstleistungsentwicklung, Vdm Verlag Dr. Müller 2007.
- Fuchs, M.; Apfelthaler, G. (2008):** Management internationaler Geschäftstätigkeit (Vol. 2), Springer, Wien 2008.
- Galbraith, J.R. (2002):** Organizing to Deliver Solutions. In: Organizational Dynamics, Vol. 31 (2002) No. 2, pp. 194-207.
- Galvao, A.B.; Sato, K. (2005):** Affordances in product architecture: Linking technical functions and users' tasks. *ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*. Long Beach, California USA.
- Gausemeier, J.; Feldmann, K. (2006):** Integrative Entwicklung räumlicher elektronischer Baugruppen. (1 ed.), Hanser Fachbuchverlag 2006.
- Geisberger, E. (2005):** Requirements Engineering eingebetteter Systeme - ein interdisziplinärer Modellierungsansatz (Vol. 1), Shaker 2005.
- Geisberger, E.; Broy, M.; Berenbach, B.; Kazmeier, J.; Paulish, D.; Rudorfer, A. (2006):** Requirements Engineering Reference Model (REM). Technische Universität München, 2006.
- Gericke, A.; Winter, R. (2009):** Entwicklung eines Bezugsrahmens für Konstruktionsforschung und Artefaktkonstruktion in der gestaltungsorientierten Wirtschaftsinformatik. In: Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik. Eds.: Becker, J.; Krcmar, H.; Niehaves, B., 1 (ed.). Physica-Verlag 2009, pp. 195-210.
- Gernert, C.; Ahrend, N. (2002):** IT-Management, System statt Chaos (Vol. 2), Oldenbourg 2002.
- Gernert, C.; Köppen, V. (2006):** Geschäftsprozesse optimal gestalten. In: Handbuch IT in der Verwaltung. Eds.: Kröger, D.; Wind, M., 1 (ed.). Springer, Berlin 2006, pp. 194-223.
- Gervasi, V.; Zowghi, D. (2005):** Reasoning about inconsistencies in natural language requirements. In: ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 14 (2005) No. 3, pp. 277 - 330.
- Gierhake, O. (1998):** Integriertes Geschäftsprozeßmanagement, Vieweg Friedr. + Sohn Ver 1998.
- Glaser, B.G.; Strauss, A.L. (1967):** Discovery of Grounded Theory: The Strategies for Qualitative Research (Vol. 8), Aldine Pub 1967.
- Gläser, J.; Laudel, G. (2004):** Experteninterviews und qualitative Inhaltsanalyse. (1 ed.), Vs Verlag 2004.
- Glinz, M.; Wieringa, R.J. (2007):** Stakeholders in Requirements Engineering. In: Software, IEEE, Vol. 24 (2007) No. 2, pp. 18-20.

- Goedkoop, M.J.; Halen, C.J.G.v.; Harry R.M. te Riele; Rommens, P.J.M. (1999):** Product Service systems - Ecological and Economic Basics. Ministry of VROM and EZ, 1999.
- Goeken, M.; Patas, J. (2010):** Evidenzbasierte Strukturierung und Bewertung empirischer Forschung im Requirements Engineering. In: Wirtschaftsinformatik, Vol. 52 (2010) No. 3, pp. 173-184.
- Goguen, J.; Linde, C. (1993):** Techniques for Requirements Elicitation. *1st IEEE International Symposium on Requirements Engineering (RE'93)*. San Diego, USA.
- Gorschek, T.; Davis, A.M. (2008):** Requirements engineering: In search of the dependent variables. In: Information and Software Technology, Vol. 50 (2008) No. 1-2, pp. 67-75.
- Gorschek, T.; Svahnberg, M. (2005):** Requirements Experience in Practice: Studies of Six Companies. In: Engineering and Managing Software Requirements. Eds.: Aurum, A.; Wohlin, C., 1 (ed.). Springer, Berlin 2005, pp. 405-426.
- Gorschek, T.; Wohlin, C. (2006):** Requirements Abstraction Model. In: Requirements Engineering, Vol. 11 (2006) No. 1, pp. 79-101.
- Gotel, O.C.Z.; Finkelstein, C.W. (1994):** An analysis of the requirements traceability problem. Paper presented at the First International Conference on Requirements Engineering, Colorado Springs, USA, pp. 94-101.
- Gough, P.A.; Fodemski, F.T.; Higgins, S.A.; Ray, S.J. (1995):** Scenarios-an industrial case study and hypermedia enhancements. Paper presented at the Second IEEE International Symposium on Requirements Engineering, pp. 10-17.
- Grady, J.O. (2006):** System Requirements Analysis, Academic Pr Inc 2006.
- Graham, I. (2003):** The Compleat Requirements Analyste. In: IEEE Software, (2003) No. Vol 20 (2003) Nr 6., pp. 99-101.
- Gräble, M.; Dollmann, T. (2010):** Vorgehensmodelle des Product-Service Systems Engineering. In: Hybride Wertschöpfung. Eds.: Thomas, O.; Loos, P.; Nüttgens, M. Springer Berlin Heidelberg 2010, pp. 82-129.
- Green, M.G.; P.K., P.R.; Wood, K.L. (2004):** Product usage context: Improving customer needs gathering and design target setting. *ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Salt Lake City, Utah, USA.
- Greiffenberg, S. (2004):** Methodenentwicklung in Wirtschaft und Verwaltung (Vol. 1), Kovac, J. 2004.
- Grönroos, C. (2000):** Service Management and Marketing: A Customer Relationship Management Approach. (2 ed.), John Wiley and Sons 2000.
- Gross, A.; Doerr, J. (2009):** EPC vs. UML Activity Diagram - Two Experiments Examining their Usefulness for Requirements Engineering. Paper presented at the 17th IEEE International Requirements Engineering Conference, RE '09, pp. 47-56.
- Gross, D.; Yu, E. (2001):** From Non-Functional Requirements to Design through Patterns. In: Requirements Engineering, Vol. 6 (2001) No. 1, pp. 18-36.
- Grube, P.P.; Schmid, K. (2008):** Selecting Creativity Techniques for Innovative Requirements Engineering. Paper presented at the Third International Workshop on

- Multimedia and Enjoyable Requirements Engineering - Beyond Mere Descriptions and with More Fun and Games, MERE '08 pp. 32-36.
- Grünbacher, P.; Seyff, N. (2005):** Requirements Negotiation. In: Engineering and Managing Software Requirements Eds.: Aurum, A.; Wohlin, C., 1 (ed.). Springer, Berlin 2005, pp. 478
- Guenov, M.D.; Barker, S.G. (2004):** Requirements-driven design decomposition: A method for exploring complex system architecture. *ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Salt Lake City, Utah, USA.
- Gulla, J.A. (1996):** A general explanation component for conceptual modeling in CASE environments. In: ACM Transactions on Information Systems Vol. 14 (1996) No. 3, pp. 297-322.
- Gupta, D.; Prakash, N. (2001):** Engineering Methods from Method Requirements Specifications. In: Requirements Engineering, Vol. 6 (2001) No. 3, pp. 135-160.
- Hagemann, G. (2009):** Methodenhandbuch Unternehmensentwicklung: Ist-Situation analysieren, Strategie entwickeln, Marke positionieren. (2 ed.), Gabler 2009.
- Halen, C.V.; Vezzoli, C.; Wimmer, R. (2005):** Methodology for Product Service System Innovation, Koninklijke Van Gorcum BV, PO 2005.
- Hall, A. (1997):** What's the use of requirements engineering? Paper presented at the Third IEEE International Symposium on Requirements Engineering, pp. 2 - 3.
- Hall, T.; Beecham, S.; Rainer, A. (2002):** Requirements problems in twelve software companies: an empirical analysis. In: IEE Proceedings Software, Vol. 149 (2002), pp. 153 - 160.
- Haller, S. (2005):** Dienstleistungsmanagement: Grundlagen - Konzepte - Instrumente (3 ed.), Gabler 2005.
- Hardt, M.; Mackenthun, R.; Bielefeld, R. (2002):** Integrating ECUs in vehicles - requirements engineering in series development. Paper presented at the Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on, pp. 227-236.
- Hauck, R. (2009):** Rechtliche Betrachtung hybrider Produkte. In: Hybride Wertschöpfung: Konzepte, Methoden und Kompetenzen für die Preis- und Vertragsgestaltung. Eds.: Reichwald, R.; Krcmar, H.; Nippa, M. Josef Eul Verlag GmbH 2009, pp. 67-98.
- Haumer, P.; Pohl, K.; Weidenhaupt, K. (1998):** Requirements Elicitation and Validation with Real World Scenes. In: IEEE Trans. Softw. Eng., Vol. 24 (1998) No. 12, pp. 1036-1054.
- Hauschildt, J. (2004):** Innovationsmanagement. (3 ed.), Vahlen, München 2004.
- Hauser, J.R.; Clausing, D. (1988):** The House of Quality. In: Harvard Business Review, Vol. 66 (1988) No. 3, pp. 63-73.
- Heinrich, L.J. (2000):** Bedeutung von Evaluation und Evaluationsforschung in der Wirtschaftsinformatik. In: Evaluation und Evaluationsforschung in der Wirtschaftsinformatik (Vol. 1). Eds.: Heinrich, L.J.; Häntschel, I. Oldenbourg 2000, pp. 7-22.
- Heinzl, A.; König, W.; Hack, J. (2001):** Erkenntnisziele der Wirtschaftsinformatik in den nächsten drei und zehn Jahren. In: Wirtschaftsinformatik, Vol. 43 (2001) No. 3, pp. 223-233.

- Hengst, M.d.; Kar, E.v.d.; Appelman, J. (2004):** Designing Mobile Information Services: User Requirements Elicitation with GSS Design and Application of a Repeatable Process. *The 37th Hawaii International Conference on System Sciences Hawaii*
- Herrmann, T.; Kleinbeck, U.; Kremar, H. (2005):** Konzepte für das Service Engineering: Modularisierung, Prozessgestaltung und Produktivitätsmanagement. (1 ed.), Physica Verlag 2005.
- Hevner, A.R.; March, S.T.; Park, J.; Ram, S. (2004):** Design Science in Information Systems Research. In: *MIS Quarterly*, Vol. 28 (2004) No. 1, pp. 75-105.
- Hickey, A.M.; Davis, A. (2004):** A Unified Model of Requirements Elicitation. In: *Journal of Management Information Systems*, Vol. 20 (2004) No. 4, pp. 65-84.
- Hickey, A.M.; Davis, A.M. (2003):** Elicitation technique Selection: how do experts do it? , *The 11th IEEE international Requirements Engineering Conference*. Monterey Bay.
- Hickey, A.M.; Dean, D.L. (1998):** Prototyping for Requirements Elicitation and Validation: A Participative Prototype Evaluation Methodology. Paper presented at the Americas Conference on Information Systems, Baltimore, pp. 798-200.
- Hill, W.; Ulrich, H. (1979):** Wissenschaftstheoretische Aspekte ausgewählter betriebswirtschaftlicher Konzeptionen. In: *Wissenschaftstheoretische Grundfragen der Wirtschaftswissenschaften*. Eds.: Raffee, H.; Abel, B. Vahlen, München 1979, pp. 161-190.
- Hoeth, U.; Schwarz, W. (2002):** Qualitätstechniken für die Dienstleistung. (2 ed.), Hanser Fachbuch 2002.
- Hoffmann, H. (2010):** Prototyping Automotive Software und Services: Vorgehen und Werkzeug zur nutzerorientierten Entwicklung. (1 ed.), Cuvillier, München 2010.
- Höhn, R.; Höppner, S. (2008):** Das V-Modell XT. Grundlagen, Methodik und Anwendungen. (1 ed.), Springer 2008.
- Hsia, P.; Davis, A.M.; Kung, D.C. (1993):** Status report: requirements engineering. In: *IEEE Software* Vol. 10 (1993) No. 6, pp. 75 - 79.
- Hui, Z.H.; Ohnishi, A. (2003):** Integration and evolution method of scenarios from different viewpoints. Paper presented at the Software Evolution, 2003. Proceedings. Sixth International Workshop on Principles of, pp. 183-188.
- Hull, E.; Jackson, K.; Dick, J. (2004):** Requirements Engineering. (2 ed.), Springer London 2004.
- Hull, E.; Jackson, K.; Dick, J. (2011):** Requirements Engineering, Springer London 2011.
- Humpert, A. (1995):** Methodische Anforderungsverarbeitung auf Basis eines objektorientierten Anforderungsmodells Dissertation, Universität Paderborn Heinz Nixdorf Inst 1995.
- Hungenberg, H. (2004):** Strategisches Management in Unternehmen. Ziele, Prozesse, Verfahren. (3 ed.), Gabler 2004.
- Husen, C.v. (2007):** Anforderungsanalyse für produktbegleitende Dienstleistungen. Dissertation, Universität Stuttgart 2007.
- Hutzschenreuter, T. (2009):** Allgemeine Betriebswirtschaftslehre: Grundlagen mit zahlreichen Praxisbeispielen. (3 ed.), Gabler 2009.

- IEEE (1990):** IEEE standard glossary of software engineering terminology (IEEE 610.12-1990). New York.
- IEEE (1998):** IEEE recommended practice for software requirements specifications, 1998.
- IEEE Std 1233 (1998):** IEEE guide for developing system requirements specifications. In: IEEE Std 1233, 1998 Edition, (1998).
- IfM (2002):** Unternehmensgrößenstatistik 2001/2002 - Daten und Fakten, Bonn 2002.
- Jacobson, I. (1995):** The use-case construct in object-oriented software engineering. In: Scenario-based design. Eds. John Wiley and Sons, Inc. 1995, pp. 309-336.
- Jacobson, I.; Booch, G.; Rumbaugh, J. (1999):** Unified Software Development Process: The complete guide to the Unified Process from the original designers, Addison-Wesley Longman, Amsterdam 1999.
- Jacobson, I.; Christerson, M.; Jonsson, P. (1992):** Object- Oriented Software Engineering. A Use Case Driven Approach: A Use CASE Approach. (4 ed.), Addison-Wesley Longman, Amsterdam 1992.
- Jacsó, P. (2008):** Google Scholar revisited. In: Online Information Review, Vol. 32 (2008) No. 1, pp. 102-114.
- Jakimi, A.; Sabraoui, A.; Badidi, E.; Idri, A.; El Koutbi, M. (2007):** Use Cases and Scenarios Engineering. Paper presented at the 4th International Conference on Innovations in Information Technology, IIT '07, Dubai, pp. 521-525.
- Jarke, M. (1998):** Anforderungsmodellierung: Können wir die Brücke zwischen Anwendung und Entwicklung stabilisieren? , *GI-Workshops Modellierung* 98. Münster.
- Jarke, M. (2009):** Perspektiven der Wirtschaftsinformatik aus Sicht der Informatik. In: Wirtschaftsinformatik, Vol. 51 (2009) No. 1, pp. 82-87.
- Jarke, M.; Bui Tung, X.; Carrol, J.M. (1999):** Scenario Management: An Interdisciplinary Approach. In: Requirements Engineering Journal, Vol. 3 (1999) No. 4.
- Jarke, M.; Gebhardt, M.; Jacobs, S.; Nissen, H.W. (1996):** Conflict analysis across heterogeneous viewpoints: formalization and visualization. Paper presented at the Twenty-Ninth Hawaii International Conference on System Sciences, Hawaii, pp. 199-208.
- Järvinen, P. (2000):** On A Variety of Research Output Types. Paper presented at the 23rd Information Systems Research Seminar (IRIS23), Tampere, pp. 251-265.
- Jiang, L.; Eberlein, A.; Far, B.H. (2005):** Combining requirements engineering techniques - theory and case study. Paper presented at the 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, ECBS '05 Greenbelt, Maryland, USA, pp. 105 - 112.
- Jiao, J.; Tseng, M.M. (1999):** A requirement management database system for product definition. In: Journal of Manufacturing Technology Management, Vol. 10 (1999) No. 3, pp. 146-153.
- Jiao, J.R.; Chen, C.-H. (2006):** Customer Requirement Management in Product Development: A Review of Research Issues. In: Concurrent Engineering September, Vol. 14 (2006) No. 3, pp. 173-185.
- Johansson, J.E.; Krishnamurthy, C.; Schlißberg, H.E. (2003):** Solving the solutions problem. In: McKinsey Quarterly, (2003) No. 3, pp. 116-125.

- Jönsson, P.; Lindvall, M. (2005):** Impact Analysis. In: Engineering and Managing Software Requirements. Eds.: Aurum, A.; Wohlin, C. Springer, Berlin 2005, pp. 117-142.
- Jung, C. (2006):** Anforderungsklärun g in interdisziplinärer Entwicklungsumgebung, Dr. Hut 2006.
- Käkölä, T.; López, J.C.D. (2006):** Software Product Lines. Research Issues in Engineering and Management (Vol. 1), Springer, Berlin 2006.
- Kalmbach, P.; Franke, R.; Knottenbauer, K.; Krämer, H.; Schaefer, H. (2003):** Statistisches Bundesamt: Die Bedeutung einer wettbewerbsfähigen Industrie für die Entwicklung des Dienstleistungssektors - Eine Analyse der Bestimmungsgründe der Expansion industrienahe r Dienstleistungen in modernen Industriestaaten. Institut für Konjunktur- und Strukturfor schung (IKSF), 2003.
- Kamata, M.I.; Tamai, T. (2007):** How Does Requirements Quality Relate to Project Success or Failure? Paper presented at the 15th IEEE International Requirements Engineering Conference, RE '07, New Delhi, pp. 69-78.
- Kaplan, B.; Duchon, D. (1988):** Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study. In: MIS Quarterly, Vol. 12 (1988) No. 4, pp. 571-586.
- Karni, R.; Kaner, M. (2006):** An Engineering Tool for the Conceptual Design of Service Systems. In: Advances in Services Innovations. Eds.: Fähnrich, K.-P.; Spath, D., 1 (ed.). Springer, Berlin 2006, pp. 65-83.
- Karni, R.; Kaner, M. (2007):** An Engineering Tool for the Conceptual Design of Service Systems. In: Advances in Services Innovations. Eds.: Spath, D.; Fähnrich, K.-P. Springer, Berlin 2007, pp. 65-83.
- Karwowski, W.; Salvendy, G. (2010):** Introduction to Service Engineering, John Wiley & Sons 2010.
- Kelly, G.A. (1998):** Die Psychologie der persönlichen Konstrukte, Junfermann 1998.
- Kersten, W.; Zink, T.; Kern, E.-M. (2006):** Wertschöpfungsnetzwerke zur Entwicklung und Produktion hybrider Produkte: Ansatzpunkte und Forschungsbedarf. In: Wertschöpfungsnetzwerke. Eds.: Blecker, T.; Gemünden, H.G. Schmidt (Erich) 2006, pp. 189-201.
- Kienreich, W.; Strohmaier, M. (2006):** Wissensmodellierung - Basis für die Anwendung semantischer Technologien. In: Semantic Web: Wege zur vernetzten Wissensgesellschaft. Eds.: Pellegrini, T.; Blumauer, A., 1 (ed.). Springer, Berlin 2006, pp. 360-371.
- Kitchenham, B.A.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. (2009):** Systematic literature reviews in software engineering - a systematic literature review. In: Information and Software Technology, Vol. 51 (2009) No. 1, pp. 7-15.
- Kitchenham, B.A.; Dyba, T.; Jorgensen, M. (2004):** Evidence-Based Software Engineering. Paper presented at the 26th International Conference on Software Engineering, ICSE 2004, Edinburgh, Scotland, pp. 273 - 281.
- Klein, H.; Myers, M. (1999):** A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. In: MIS Quarterly, Vol. 23 (1999) No. 1, pp. 67-94.

- Klein, R. (2007):** Modellgestütztes Service Systems Engineering: Theorie und Technik einer systemischen Entwicklung von Dienstleistungen (1 ed.), Gabler 2007.
- Knackstedt, R.; Pöppelbuß, J.; Winkelmann, A. (2008):** Integration von Sach- und Dienstleistungen – Ausgewählte Internetquellen zur hybriden Wertschöpfung. In: *Wirtschaftsinformatik*, Vol. 50 (2008) No. 3, pp. 235-247.
- Knebel, U.; Esch, S.; Leimeister, J.M.; Pressler, A.; Krcmar, H. (2009):** Online, Set, Go - Design and empirical Test of an It-based physical Activity Intervention. *17th European Conference on Information Systems*. Verona.
- Knebel, U.; Leimeister, J.M.; Krcmar, H. (2007):** Personal Mobile Sports Companion: Design and Evaluation of IT-supported Product Service-Bundles in the Sports Industry. *15th European Conference on Information Systems (ECIS 2007)*. St. Gallen: University of St. Gallen, Institute of Information Management.
- Koch, V. (2010):** Interaktionsarbeit bei produktbegleitenden Dienstleistungen: Am Beispiel des technischen Services im Maschinenbau, Gabler 2010.
- Koller, R. (1997):** Konstruktionslehre für den Maschinenbau. (3 ed.), Springer 1997.
- Kortler, S.; Helms, B.; Berkovich, M.; Lindemann, U.; Shea, K.; Leimeister, J.M.; Krcmar, H. (2010):** Using mdm-methods in order to improve managing of iterations in design processes. *The 12th International dependency and structure modelling conference, DSM*. Cambridge.
- Kortmann, D. (2007):** Dienstleistungsgestaltung innerhalb hybrider Leistungsbündel. (1 ed.), Shaker 2007.
- Kotonya, G.; Sommerville, I. (1996):** Requirements engineering with viewpoints. In: *Software Engineering Journal*, Vol. 11 (1996) No. 1, pp. 5-18.
- Kotonya, G.; Sommerville, I. (1998):** Requirements Engineering: Processes and Techniques (1 ed.), Wiley & Sons 1998.
- Krcmar, H. (2010):** Informationsmanagement. (5 ed.), Springer, Berlin 2010.
- Kromrey, H. (2000):** Empirische Sozialforschung, UTB, Stuttgart 2000.
- Kruse, P.J. (1996):** Anforderungen in der Systementwicklung: Erfassung, Aufbereitung und Bereitstellung von Anforderungen in interdisziplinären Entwicklungsprojekten, VDI-Verlag, Düsseldorf 1996.
- Kuechler, B.; Vaishnavi, V. (2008):** On theory development in design science research: anatomy of a research project. In: *European Journal of Information Systems*, Vol. 17 (2008) No. 5, pp. 489–504.
- Kujala, S.; Kauppinen, M.; Lehtola, L.; Kojo, T. (2005):** The Role of User Involvement in Requirements Quality and Project Success. *13th IEEE International Conference on Requirements Engineering* (pp. 75 - 84). La Sorbonne, France: IEEE Computer Society.
- Kullmann, H.J. (2010):** ProdHaftG: Gesetz über die Haftung für fehlerhafte Produkte. (6 ed.), Schmidt (Erich), Berlin 2010.
- Lai, X.; Gershenson, J.K. (2009):** DSM-based Product Representation for Retirement Process-based Modularity. *ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE*. San Diego.

- Lamnek, S. (2005):** Qualitative Sozialforschung: Lehrbuch. (4 ed.), Beltz Psychologie Verlags Union 2005.
- Lamsweerde, A. (2000):** Requirements engineering in the year 00: a research perspective. Paper presented at the International Conference on Software Engineering, ICSE, Limerick, Ireland, pp. 5 - 19.
- Lamsweerde, A.v. (2001):** Goal-oriented requirements engineering: a guided tour. Paper presented at the fifth IEEE International Symposium on Requirements Engineering, Toronto, Canada, pp. 249-262.
- Lamsweerde, A.v. (2003):** From System Goals to Software Architecture. In: Springer Berlin / Heidelberg, Vol. 28 (2003) No. 4, pp. 25-43.
- Lamsweerde, A.v. (2009):** Requirements Engineering: From System Goals to UML Models to Software Specifications (Vol. 1), Wiley & Sons 2009.
- Langer, P.; Böhm, T.; Schermann, M.; Krcmar, H. (2009):** Solution Data Management (SDM) Referenzmodell. In: Hybride Wertschöpfung: Konzepte, Methoden und Kompetenzen für die Preis- und Vertragsgestaltung. Eds.: Reichwald, R.; Krcmar, H.; Nippa, M. Josef Eul Verlag GmbH 2009.
- Langer, P.; Köbler, F.; Berkovich, M.; Weyde, F.; Leimeister, J.M.; Krcmar, H. (2010):** Vorgehensmodelle für die Entwicklung hybrider Produkte – eine Vergleichsanalyse. *Multikonferenz Wirtschaftsinformatik, MKWI 2010* (pp. 405-406). Göttingen.
- Langer, S.; Kreimeyer, M.; Müller, P.; Lindemann, U.; Blessing, L. (2008):** Prozessmodellierung für das Zyklusmanagement der Entwicklung hybrider Leistungsbündel. In: Dienstleistungsmodellierung: Methoden, Werkzeuge und Branchenlösungen (Vol. 1). Eds.: Thomas, O.; Nüttgens, M. Physica-Verlag 2008, pp. 71-89.
- Langer, S.; Lindemann, U. (2009):** Managing cycles in development processes - analysis and classification of external context factors. *International conference on engineering design, ICED'09*. Stanford CA, USA.
- Laqua, R. (2007):** Das Projekt ServCASE. In: Entwicklung IT-basierter Dienstleistungen: Co-Design von Software und Services mit ServCASE (Vol. 1). Eds.: Fähnrich, K.P.; Husen, C.v. Physica-Verlag 2007, pp. 338
- Larsen, T.J.; Niederman, F.; Limayem, M.; Chan, J. (2007):** The role of modelling in achieving information systems success: UML to the rescue? In: Information Systems Journal, Vol. 19 (2007) No. 1, pp. 83-117.
- Laudon, K.C.; Laudon, J.P.; Schoder, D. (2009):** Wirtschaftsinformatik: Eine Einführung. (2 ed.), Pearson Studium 2009.
- Lavagno, L.; Passerone, C. (2005):** Design of Embedded Systems. In: Embedded Systems Handbook. Eds.: Zurawski, R. CRC Press Inc 2005.
- Lay, G. (1998):** Dienstleistungen in der Investitionsgüterindustrie. ISI Fraunhofer, 1998.
- Lay, G.; Jung, E. (2002):** Elemente einer Strategieentwicklung für produktbegleitende Dienstleistungen in der Industrie. In: Produktbegleitende Dienstleistungen - Konzepte und Beispiele erfolgreicher Strategieentwicklung. Eds. Lay, G., Jung, E., Berlin 2002, pp. 5-67.
- Lee, R. (2008):** Software Engineering Research, Management and Applications. (1 ed.), Springer, Berlin 2008.

- Leffingwell, D.; Widrig, D. (2003):** Managing Software Requirements: A Unified Approach Addison-Wesley Longman, Amsterdam 2003.
- Lehmann, A. (1995):** Dienstleistungsmanagement. Strategien und Ansatzpunkte zur Schaffung von Servicequalität, Schäffer-Poeschel Verlag 1995.
- Leimeister, J.M.; Glauner, C. (2008):** Hybride Produkte – Einordnung und Herausforderungen für die Wirtschaftsinformatik. In: Wirtschaftsinformatik, Vol. 50 (2008) No. 3, pp. 248-251.
- Leveson, N.G. (2000):** Intent specifications: An approach to building human-centered specifications. In: IEEE Trans. on Soft. Eng., Vol. 26 (2000) No. 1, pp. 15-35.
- Lindahl, M.; Sundin, E.; Sakao, T.; Shimomura, Y. (2007):** Integrated Product and Service Engineering *versus* Design for Environment – A Comparison and Evaluation of Advantages and Disadvantages. In: Advances in Life Cycle Engineering for Sustainable Manufacturing Businesses. Eds.: Takata, S.; Umeda, Y. Springer London 2007, pp. 137-142.
- Lindemann, U. (2006):** Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden (2 ed.), Springer, Berlin 2006.
- Lindemann, U.; Maurer, M.; Braun, T. (2008):** Structural Complexity Management: An Approach for the Field of Product Design (Vol. 1), Springer, Berlin 2008.
- Lönngren, H.-M.; Kolbe, H.; Rosenkranz, C. (2008):** Erfolgsfaktoren für hybride Wertschöpfungsnetze – Eine Fallstudienanalyse. Paper presented at the Multikonferenz der Wirtschaftsinformatik, MKWI, München, pp. 723-734.
- Luczak, H.; Liestmann, V.; Winkelmann, K.; Gill, C. (2003):** Service Engineering industrieller Dienstleistungen. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer, Berlin, Heidelberg, New York 2003.
- Lunze, J. (2003):** Automatisierungstechnik, Oldenbourg 2003.
- Macaulay, L.A. (1996):** Requirements Engineering Springer-Verlag GmbH 1996.
- Maciaszek, L.A. (2007):** Requirements Analysis and System Design. (3 ed.), Addison-Wesley Educational Publishers Inc 2007.
- Maiden, N. (2010):** Trust Me I'm Analyst. In: IEEE Software, (2010) No. Vol 27 (2010) Nr 1, pp. 46-47.
- Maiden, N.A.M.; Rugg, G. (1996):** ACRE: selecting methods for requirements acquisition. In: Software Engineering Journal, Vol. 11 (1996) No. 3, pp. 183-192.
- Maier, J.R.A.; Ezhilan, T.; Fadel, G.M.; Summers, J.D.; Mocko, G. (2007):** A hierarchical requirements modeling scheme to support engineering innovation. *International Conference on Engineering Design, ICED'07*. Paris, France.
- Maleri, R. (1997):** Grundlagen der Dienstleistungsproduktion. (1 ed.), Springer 1997.
- Mannweiler, C.; Möhrer, J.; Fiekers, C. (2010):** Planung investiver Produkt-Service Systeme. In: Produkt-Service Systeme: Gestaltung und Realisierung. Eds.: Aurich, J.C.; Clement, M.H., 1 (ed.). Springer, Berlin Heidelberg 2010, pp. 170.
- March, S.T.; Smith, G.F. (1995):** Design and natural science research on information technology. In: Decision Support Systems, Vol. 15 (1995) No. 4, pp. 251-266.

- Mauro, C.; A., S.; Dünnebeil, S.; Leimeister, J.M.; Krcmar, H. (2009):** Mobile Anwendungen im Kontext des Medizinproduktegesetzes. In Fischer, S.; Maehle, E.; Reischuk, R. (Eds.), *Informatik 2009 - Im Focus das Leben*. Lübeck: Köllen Druck+Verlag GmbH.
- Maute, D. (2006):** Technische Akustik und Lärmschutz. (1 ed.), Hanser Fachbuchverlag 2006.
- Maxwell, J.C.; Anton, A.I. (2009):** Checking Existing Requirements for Compliance with Law Using a Production Rule Model. Paper presented at the Requirements Engineering and Law (RELAW), 2009 Second International Workshop on, pp. 1-6.
- Mayring, P. (2002):** Einführung in die qualitative Sozialforschung. (5 ed.), Beltz 2002.
- Mayring, P. (2007):** Qualitative Inhaltsanalyse. Grundlagen und Techniken Utb 2007.
- Meerkamm, M.K.a.H. (2003):** Universal function-structures in early design stages. *International Conference on Engineering Design, ICED'03*. Stockholm.
- Meffert, H.; Bruhn, M. (2006):** Dienstleistungsmarketing: Grundlagen - Konzepte - Methoden. (5 ed.), Gabler 2006.
- Meier, H.; Kortmann, D.; Golembiewski, M. (2006):** Hybride Leistungsbündel in kooperativen Anbieter-Netzwerken. In: *Industrie Management*, Vol. 22 (2006) No. 4, pp. 25-28.
- Meier, J.J.; Conkling, T.W. (2008):** Google Scholar's Coverage of the Engineering Literature: An Empirical Study. In: *The Journal of Academic Librarianship*, Vol. 34 (2008) No. 3, pp. 196-201.
- Meiren, T.; Liestmann, V. (2002):** Service Engineering in der Praxis, Fraunhofer IRB Verlag, Stuttgart 2002.
- Mellerowicz, K. (1958):** Betriebswirtschaftslehre der Industrie (Vol. 2). (3 ed.), Haufe 1958.
- Méndez Fernández, D.; Kuhrmann, M. (2009):** Artefact-based Requirements Engineering and its Integration into a Process Framework. Institut für Informatik, Technische Universität München, 2009.
- Méndez Fernández, D.; Penzenstadler, B.; Kuhrmann, M.; Broy, M. (2010):** A Meta Model for Artefact-Oriented: Fundamentals and Lessons Learned in Requirements Engineering. Paper presented at the 13th International Conference on Model Driven Engineering Languages and Systems, Oslo, Norway, pp. 183-197.
- Menten, A.; Scheibmayr, S.; Klimpke, L.:** Using audio and collaboration technologies for distributed requirements elicitation and documentation. Paper presented at the Third International Workshop on Managing Requirements Knowledge, MARK, Sydney, Australia, pp. 51-59.
- Meyer, A.; Mattmüller, R. (1987):** Qualität von Dienstleistungen - Entwurf eines praxisorientierten Qualitätsmodells. In: *Marketing: Zeitschrift für Forschung und Praxis*, Vol. 3 (1987), pp. 187-195.
- Meyer, K.; Husen, C.v. (2007):** Die ServCASE-Methode im Überblick. In: *Entwicklung IT-basierter Dienstleistungen: Co-Design von Software und Services mit ServCASE* (Vol. 1). Eds.: Fähnrich, K.P.; Husen, C.v. Physica-Verlag 2007, pp. 338
- Miller, D.; Hope, Q.; Eisenstat, R.; Foote, N.; Galbraith, J. (2002):** The problem of solutions: Balancing clients and capabilities. In: *Business Horizons*, Vol. 45 (2002) No. 2, pp. 3-12.

- Möhrle, M.G.; Spilgies, W.-D. (2005):** QFD für Product Service Systems. In: *Industrie Management*, Vol. 21 (2005) No. 3, pp. 9-12.
- Möller, K.; Cassack, I. (2008):** Prozessorientierte Planung und Kalkulation (kern-) produktbegleitender Dienstleistungen. In: *Zeitschrift für Planung & Unternehmenssteuerung*, Vol. 19 (2008) No. 2, pp. 159-184.
- Mont, O.K. (2002):** Clarifying the concept of product–service system. In: *Journal of Cleaner Production*, Vol. 10 (2002) No. 3, pp. 237-245.
- Moore, C.W. (2003):** *The Mediation Process: Practical Strategies for Resolving Conflict*, Jossey Bass 2003.
- Morelli, N. (2002):** Designing product/service systems - A methodological exploration. In: *Design Issue*, Vol. 18 (2002) No. 3, pp. 3-17.
- Murdick, R.G.; Render, B.; Russe, R.S. (1990):** *Service Operations Management*, Allyn & Bacon 1990.
- Mylopoulos, J. (2006):** Goal-Oriented Requirements Engineering, Part II. Paper presented at the 14th IEEE International Conference Requirements Engineering, Minneapolis / St. Paul, Minnesota, USA, pp. 5-5.
- Nikora, A.P. (2005):** Classifying requirements: towards a more rigorous analysis of natural-language specifications. *16th IEEE International Symposium on Software Reliability Engineering*. Chicago, USA.
- Nikula, U.; Sajaniemi, J.; Kälviäinen, H. (2000):** A state-of-the-practice survey on requirements engineering in small-and medium-sized enterprises. *Telecom Business Research Center*, 2000.
- Nippa, M.; Wienhold, D.; Piezonka, S. (2007):** Vom klassischen Produktgeschäft zum Lösungsgeschäft - Implikationen für eine Neugestaltung des Vergütungssystems im Vertrieb. *Technische Universität Bergakademie Freiberg*, 2007.
- Niu, N.; Easterbrook, S. (2006):** Discovering aspects in requirements with repertory grid. *Workshop on Early Aspects at ICSE: Aspect-Oriented Requirements Engineering and Architecture Design*. Shanghai, China: ACM.
- Nurmuliani, N.; Zowghi, D.; Williams, S.P. (2004):** Using card sorting technique to classify requirements change. Paper presented at the 12th IEEE International Requirements Engineering Conference, Kyoto, Japan, pp. 240-248.
- Nuseibeh, B.; Finkelstein, A.; Kramer, J. (1996):** Method engineering for multi-perspective software development. In: *Information & Software Technology*, Vol. 38 (1996) No. 4, pp. 267-274.
- Nuseibeh, B.A.; Easterbrook, S.M. (2000):** Requirements Engineering: A Roadmap. *22nd International Conference on Software Engineering, ICSE'00*. Limerick, Ireland.
- Oguachuba, J.S. (2009):** Markenprofilierung durch produktbegleitende Dienstleistungen: Eine empirische Untersuchung unter besonderer Berücksichtigung der Kundenmitwirkung, Gabler 2009.
- Olemotz, T. (1995):** Strategische Wettbewerbsvorteile durch industrielle Dienstleistungen, P. Lang 1995.
- Orlikowski, W.J.; Iacono, C.S. (2001):** Research Commentary: Desperately Seeking the "IT" in IT Research--A Call to Theorizing the IT Artifact. In: *Information Systems Research*, Vol. 12 (2001) No. 2, pp. 121-134.

- Osborn, A.F. (1979):** Applied Imagination: Principles and Procedures of Creative Problem-Solving. (3 ed.), Charles Scribner's Sons 1979.
- Pahl, G.; Beitz, W.; Feldhusen, J. (2006):** Konstruktionslehre: Grundlagen Erfolgreicher Produktentwicklung. Methoden Und Anwendung. (7 ed.), Springer, Berlin 2006.
- Pelz, G. (2001):** Modellierung und Simulation mechatronischer Systeme. Vom Chip- zum Systementwurf mit Hardwarebeschreibungssprachen. (1 ed.) 2001.
- Penzenstadler, B.; Koss, D. (2008):** High Confidence Subsystem Modelling for Reuse. Paper presented at the International Conference for Software Reuse, ICSR, Beijing, China, pp. 52-63.
- Penzenstadler, B.; Leuser, J. (2008):** Complying with Law for RE in the Automotive Domain. Paper presented at the Workshop on Requirements Engineering and Law, RELAW '08, Barcelona, Spain, pp. 11-15.
- Peterson, C.; Paasch, R.K.; Ge, P.; Dietterich, T.G. (2007):** Product innovation for interdisciplinary design under changing requirements. *International Conference on Engineering Design, ICED'07*. Paris, France.
- Petkoff, B. (1999):** Die Kybernetik II. Ordnung - eine methodologische Basis der Wirtschaftsinformatik? In: *Wirtschaftsinformatik und Wissenschaftstheorie*. Eds.: Becker, J.; König, W.; Schüüte, R. Dr. Th. Gabler Verlag 1999.
- Pohl, K. (2007):** Requirements Engineering. Grundlagen, Prinzipien, Techniken (1 ed.), Dpunkt Verlag 2007.
- Pohl, K.; Böckle, G.; Linden, F.v.d. (2005):** Software Product Line Engineering. Foundations, Principles, and Techniques. (1 ed.), Springer, Berlin 2005.
- Pohl, K.; Rupp, C. (2009):** Basiswissen Requirements Engineering. (1 ed.), dpunkt Verlag 2009.
- Pohl, K.; Sikora, E. (2007a):** COSMOD-RE: Supporting the Co-Design of Requirements and Architectural Artifacts. *15th IEEE International Requirements Engineering Conference* (pp. 258 - 261). New Delhi, India.
- Pohl, K.; Sikora, E. (2007b):** Eine Methode für das Co-Design von Anforderungs- und Entwurfsartefakten. Paper presented at the Software Engineering 2007, Fachtagung des GI-Fachbereichs Softwaretechnik, Hamburg, pp. 259-260.
- Pohl, K.; Sikora, E. (2007c):** Structuring the Co-design of Requirements and Architecture. Paper presented at the 13th Intl. Working Conference on Requirements Engineering: Foundation for Software Quality, REFSQ 2007, Trondheim, Norway, pp. 48-62.
- Ponn, J.; Lindemann, U. (2008):** Konzeptentwicklung und Gestaltung technischer Produkte: Optimierte Produkte - systematisch von Anforderungen zu Konzepten. (1 ed.), Springer, Berlin 2008.
- Procaccino, J.D.; Verner, J.M.; Shelfer, K.M.; Gefen, D. (2005):** What do software practitioners really think about project success: an exploratory study. In: *J. Syst. Softw.*, Vol. 78 (2005) No. 2, pp. 194-203.
- Procaccino, J.D.; Verner, J.M. (2006):** Software project managers and project success: An exploratory study. In: *Journal of Systems and Software*, Vol. 79 (2006) No. 11, pp. 1541-1551.
- Pulm, U. (2004):** Eine systemtheoretische Betrachtung der Produktentwicklung. Dissertation, Technischen Universität München 2004.

- Rainfurth, C. (2003):** Der Einfluss der Organisationsgestaltung produktbegleitender Dienstleistungen auf die Arbeitswelt der Dienstleistungsakteure; Am Beispiel von KMU des Maschinenbaus. Dissertation, Technischen Universität Darmstadt 2003.
- Ramaswamy, R. (1996):** Design and Management Service Processes: Keeping Customers for Life, Addison Wesley 1996.
- Ramesh, B.; Cao, L.; Baskerville, R. (2008):** Agile requirements engineering practices and challenges: an empirical study In: Information Systems Journal, Vol. 25 (2008) No. 1, pp. 60-67.
- Rashid, A.; Chitchyan, R. (2008):** Aspect-oriented requirements engineering: a roadmap. *13th international workshop on Software architectures and mobility table of contents*. Leipzig, Germany.
- Reckenfelderbäumer, M.; Busse, D. (2003):** Kundenmitwirkung bei der Entwicklung von industriellen Dienstleistungen - eine phasenbezogene Analyse. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer, Berlin, Heidelberg, New York 2003.
- Regius, B.v. (2005):** Qualität in der Produktentwicklung: Vom Kundenwunsch bis zum fehlerfreien Produkt. (1 ed.), Hanser Fachbuchverlag 2005.
- Reichwald, R. (1993):** Kommunikation und Kommunikationsmodelle. In: Handwörterbuch der Betriebswirtschaft. Eds.: v. Wittmann, W., 1993, pp. 2174-2188.
- Robertson, S.; Robertson, J. (2007):** Mastering the Requirements Process, Addison-Wesley Longman, Amsterdam 2007.
- Robinson, W.N.; Pawlowski, S.D.; Volkov, V. (2003):** Requirements interaction management. In: Computing Surveys (CSUR), Vol. 35 (2003) No. 2, pp. 132-190.
- Rosenthal, G. (2005):** Interpretative Sozialforschung: Eine Einführung. (2 ed.), Juventa 2005.
- Rössl, D. (1990):** Die Entwicklung eines Bezugsrahmens und seine Stellung im Forschungsprozess. In: Journal für Betriebswirtschaft, Vol. 40 (1990) No. 1, pp. 99-110.
- Roth, K. (1994):** Konstruieren mit Konstruktionskatalogen I. Konstruktionslehre (Band 1 : Konstruktionslehre). (2 ed.), Springer-Verlag GmbH, Berlin 1994.
- Rumpe, B. (2004):** Agile Modellierung mit UML. (1 ed.), Springer, Berlin 2004.
- Rupp, C. (2004):** Requirements-Engineering und -Management (3 ed.), Hanser Fachbuchverlag 2004.
- Sampaio, A.; Greenwood, P.; Garcia, A.F.; Rashid, A. (2007):** A Comparative Study of Aspect-Oriented Requirements Engineering Approaches. Paper presented at the First International Symposium on Empirical Software Engineering and Measurement, ESEM 2007, Madrid, Spain, pp. 166--175.
- Santos, M.Y.; Machado, R.J. (2010):** On the Derivation of Class Diagrams from Use Cases and Logical Software Architectures. Paper presented at the Fifth International Conference on Software Engineering Advances, ICSEA, Nice, France, pp. 107-113.
- Sawhney, M. (2006):** Going beyond the Product: Defining, Designing and Devlivering Customer Solutions. In: The Service-dominant Logic of Marketing. Eds.: Lusch, R.F.; Vargo, S.L. M. E. Sharpe, New York 2006, pp. 365-380.

- Sawhney, M.; Wolcott, R.C.; Arroniz, I. (2006):** The 12 Different Ways for Companies to Innovate. In: MIT Sloan Management Review, Vol. 47 (2006) No. 3, pp. 75-81.
- Schäppi, B.; Andreasen, M.M.; Kirchgeorg, M.; Radermacher, F.-J. (2005):** Handbuch Produktentwicklung (Vol. 1), Hanser Fachbuchverlag 2005.
- Schätz, B. (2006):** Building Components from Functions. In: Electronic Notes in Theoretical Computer Science, Vol. 160 (2006), pp. 321-334.
- Scheer, A.-W. (2002):** ARIS - Vom Geschäftsprozess zum Anwendungssystem. (4 ed.), Springer 2002.
- Scheer, A.-W.; Grieble, O.; Klein, R. (2003):** Modellbasiertes Dienstleistungsmanagement. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer 2003.
- Schienmann, B. (2001):** Kontinuierliches Anforderungsmanagement . Prozesse - Techniken - Werkzeuge (1 ed.), Addison-Wesley 2001.
- Schlink, H. (2004):** Bestimmung von Funktionskosten. (1 ed.), Gabler 2004.
- Schmid, M. (2005):** Service Engineering: Innovationsmanagement für Industrie und Dienstleister, Kohlhammer, Stuttgart 2005.
- Schmidt, H.; Häntschel, I. (2000):** Verbreitung von Evaluation und Evaluationsforschung in der Wirtschaftsinformatik. In: Evaluation und Evaluationsforschung in der Wirtschaftsinformatik (Vol. 1). Eds.: Heinrich, L.J.; Häntschel, I. Oldenbourg 2000, pp. 450.
- Schneider, K.; Daun, C.; Behrens, H.; Wagner, D. (2003):** Vorgehensmodelle und Standards zur systematischen Entwicklung von Dienstleistungen. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer 2003.
- Scholl, G. (2009):** Marketing nachhaltiger Dienstleistungen. (1 ed.), Metropolis, Marburg 2009.
- Schrage, M. (1990):** Shared minds - the new technologies of collaboration. Eds. Random House, New York 1990.
- Schreiner, P.; Strauß, O. (2004):** Konzeptionelle und informationstechnische Anforderungen an eine systematische Dienstleistungsentwicklung. In: Computer Aided Service Engineering. Eds.: Scheer, A.-W.; Spath, D. Spinger Verlag 2004, pp. 13-41.
- Schweiger, S.; Müller, M. (2004):** Erfolgsfaktoren für das Management industrieller Dienstleistungen in der Investitionsgüterindustrie. In: Service Today, Vol. 3 (2004), pp. 5-10.
- Schweitzer, E. (2010):** Lebenszyklusmanagement investiver Produkt-Service Systeme. In: Produkt-Service Systeme: Gestaltung und Realisierung. Eds.: Aurich, J.C.; Clement, M.H., 1 (ed.). Springer 2010, pp. 170.
- Seiffert, U.; Rainer, G. (2008):** Virtuelle Produktentstehung für Fahrzeug und Antrieb im Kfz: Prozesse, Komponenten, Beispiele aus der Praxis. (1 ed.), Vieweg+Teubner, Wiesbaden 2008.
- Shostack, G. (1984):** Designing services that deliver. In: Harvard Business Review, Vol. 62 (1984) No. 1, pp. 133-139.

- Shostack, G.L.; Kingman-Brundage, J. (1991):** How to Design a Service. In: The AMA Handbook of Marketing for the Service Industries. Eds.: Congram, C. American Management Association, New York 1991, pp. 243-261.
- Simon, H.A. (1996):** Sciences of the Artificial. (3 ed.), MIT Press, Boston 1996.
- Sommerville, I. (2004):** Software Engineering. (7 ed.), Pearson, Boston 2004.
- Sommerville, I. (2005):** Integrated requirements engineering: a tutorial. In: Software, IEEE, Vol. 22 (2005) No. 1, pp. 16-23.
- Sommerville, I. (2006):** Software Engineering: Update. (8 ed.), Addison-Wesley 2006.
- Sommerville, I.; Ransom, J. (2005):** An empirical study of industrial requirements engineering process assessment and improvement. In: ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 14 (2005) No. 1, pp. 85-117.
- Sommerville, I.; Sawyer, P. (1997a):** Requirements Engineering: A Good Practice Guide (Vol. 1), Wiley & Sons 1997a.
- Sommerville, I.; Sawyer, P. (1997b):** Viewpoints: principles, problems and a practical approach to requirements engineering. In: Annals of Software Engineering, Vol. 3 (1997b), pp. 101-130.
- Sommerville, I.; Sawyer, P.; Viller, S. (1998):** Viewpoints for requirements elicitation: a practical approach. Paper presented at the Requirements Engineering, 1998. Proceedings. 1998 Third International Conference on, pp. 74-81.
- Sommerville, I.; Sawyer, P.; Viller, S. (1999):** Managing process inconsistency using viewpoints. In: Software Engineering, IEEE Transactions on, Vol. 25 (1999) No. 6, pp. 784-799.
- Spath, D.; Demuß, L. (2003):** Entwicklung hybrider Produkte – Gestaltung materieller und immaterieller Leistungsbündel. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer 2003, pp. 463-502.
- Spath, D.; Dill, C.; Scharer, M. (2001):** Vom Markt zum Markt, Log_x 2001.
- Stachowiak, H. (1973):** Allgemeine Modelltheorie, Springer, Wien 1973.
- Stauss, B.; Bruhn, M. (2007):** Wertschöpfungsprozesse bei Dienstleistungen - Eine Einführung in den Sammelband. In: Wertschöpfungsprozesse bei Dienstleistungen: Forum Dienstleistungsmanagement. Eds.: Bruhn, M.; Stauss, B. Gabler 2007.
- Stechert, C.; Alexandrescu, I.; Franke, H.-J. (2007):** Modeling of inter-model relations for a customer oriented development of complex products. *International Conference on Engineering Design, ICED'07*. Paris, France.
- Stechert, C.; Bauer, S.; Franke, H.-J.; Meerkamm, H. (2008):** Requirements management in early stages of mechatronic design by visualisation of interdependencies. *International design conference - DESIGN 2008*. Dubrovnik, Croatia.
- Steffen, D. (2006):** Die Potenzialqualität von Dienstleistungen. Konzeptualisierung und empirische Prüfung. (1 ed.), Gabler Verlag 2006.
- Steinbach, M. (2005):** Systematische Gestaltung von Product-Service Systems: Integrierte Entwicklung von Product-Service Systems auf Basis der Lehre von Merkmalen und Eigenschaften. Dissertation, Universität d. Saarlandes Produktionstechnik 2005.

- Steward, D. (1981):** The Design Structure System: A Method for Managing the Design of Complex Systems. In: IEEE Transaction on Engineering Management, Vol. 28 (1981) No. 3, pp. 79-83.
- Stockmann, R. (2007):** Einleitung: Wozu brauchen wir Evaluation? In: Handbuch zur Evaluation: Eine praktische Handlungsanleitung. Eds.: Stockmann, R., 1 (ed.). Waxmann 2007, pp. 367.
- Sturm, F.; Bading, A. (2008):** Investitionsgüterhersteller als Anbieter industrieller Lösungen – Bestandsaufnahme des Wandels anhand einer Umfrage. In: Wirtschaftsinformatik, Vol. 50 (2008) No. 3, pp. 174-186.
- Sturm, F.; Bading, A.; Schubert, M. (2007):** Investitionsgüterhersteller auf dem Weg zum Lösungsanbieter. Institut Arbeitswissenschaft und Technologiemanagement, 2007.
- Sutcliffe, A. (2002):** User-Centred Requirements Engineering (Vol. 1), Springer, Berlin 2002.
- Takeda, H.; Tomiyama, T.; Yoshikawa, H.; Veerkamp, P. (1990):** Modeling Design Processes. Centre for Mathematics and Computer Science, 1990.
- Tan, A.R.; McAlloone, T.C.; Gall, C. (2007):** Product/Service-System Development - An explorative Case Study in a manufacturing Company. *international conference on engineering design, ICED 07*. Paris, France.
- Tanabe, D.; Uno, K.; Akemine, K.; Yoshikawa, T.; Kaiya, H.; Saeki, M. (2008):** Supporting Requirements Change Management in Goal Oriented Analysis. Paper presented at the 16th IEEE International Requirements Engineering, RE '08 Barcelona, Spain, pp. 3-12.
- The Standish Group (1995):** The chaos report.
- The Standish Group (1999):** A recipe for success.
- The Standish Group (2002):** The Standish Group: Extreme Chaos.
- The Standish Group (2004):** The Standish Group: Chaos Demographics.
- Thomas, O.; Walter, P.; Loos, P. (2008):** Product-Service Systems: Konstruktion und Anwendung einer Entwicklungsmethodik. In: Wirtschaftsinformatik, Vol. 50 (2008) No. 3, pp. 208-219.
- Thommen, J.-P.; Achleitner, A.-K. (2006):** Allgemeine Betriebswirtschaftslehre. Umfassende Einführung aus managementorientierter Sicht. (4 ed.), Gabler 2006.
- Tschamler, H. (1996):** Wissenschaftstheorie - Eine Einführung für Pädagogen. (3 ed.), Klinkhardt, Bad Heilbrunn 1996.
- Tukker, A. (2004):** Eight types of product-service system: eight ways to sustainability? Experiences from suspronet. In: Business Strategy and the Environment, Vol. 13 (2004), pp. 246-260.
- Tuli, R.; Kohli, A.; Bharadwaj, S. (2007):** Rethinking Customer Solutions: From product Bundles to Relational Processes. In: Journal of Marketing, Vol. 71 (2007) No. 3, pp. 1-17.
- Tuunanen, T. (2003):** A new perspective on requirements elicitation methods. In: Journal of Information Technology Theory and Application, Vol. 5 (2003) No. 3, pp. 45-72.
- Ulrich, K.; Eppinger, S. (2003):** Product Design and Development (3 ed.), Mcgraw-Hill Professional 2003.

- Vaishnavi, V.; Kuechler, W. (2004):** Design Science in Information Systems. <http://www.isworld.org/Researchdesign/drisISworld.htm>, accessed at 08.09.2007.
- Verabschiedete Fassung der MKWI-Sitzung vom 2008-02-27 in München (2008):** WI-Orientierungslisten. Wirtschaftsinformatik.
- Verein Deutschr Ingenieure (1993):** VDI 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, 1993.
- Versteegen, G. (2002):** Software-Management. (1 ed.), Springer 2002.
- Versteegen, G. (2007):** Neues im Anforderungsmanagement. In: OBJEKTSpektrum/Requirements Engineering - online Ausgabe, (2007).
- Vidgen, R. (2003):** Requirements analysis and UML use cases and class diagrams. In: Computing & Control Engineering Journal, Vol. 14 (2003) No. 2, pp. 12-17.
- Vogel, O.; Arnold, I.; Chughtai, A.; Ihler, E.; Kehrer, T.; Mehlig, U.; Zdun, U. (2009):** Software-Architektur. (2 ed.), Spektrum Akademischer Verlag 2009.
- Volz, T. (1997):** Management ergänzender Dienstleistungen für Sachgüter. Der schwierige Weg vom Sachgut- Hersteller zum Problemlöser, Vandenhoeck + Ruprecht Gm 1997.
- Walsham, G. (1993):** Interpreting Information Systems in Organizations, John Wiley and Sons, Chichester 1993.
- Weber, C.; Steinbach, M.; Botta, C.; Deubel, T. (2004):** Modelling of Product-Service Systems (PSS) Based on the PDD Approach. *International design Conference - DESIGN 2004*. Dubrovnik.
- Webster, J.; Watson, R.T. (2002):** Analyzing the Past to Prepare for the Future: Writing a Literature Review. In: MIS Quarterly, Vol. 26 (2002) No. 2, pp. xiii-xxiii.
- Weigert, J. (2005):** Hygienemanagement und Infektionsprophylaxe: Ein praktischer Leitfaden für teil- und vollstationäre Pflegeeinrichtungen. (1 ed.), Schlütersche, Hannover 2005.
- Wieggers, K.E. (2005):** Software Requirements. (1 ed.), Microsoft Press Deutschland 2005.
- Wieringa, R.; Heerkens, H. (2008):** Design Science, Engineering Science and Requirements Engineering. Paper presented at the International Requirements Engineering Conference, RE '08, pp. 36 - 48.
- Wieringa, R.; Maiden, N.; Mead, N.; Rolland, C. (2006):** Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. In: Requirements Engineering, Vol. 11 (2006) No. 1, pp. 102-107.
- Wilde, T.; Hess, T. (2007):** Forschungsmethoden der Wirtschaftsinformatik. In: Wirtschaftsinformatik, Vol. 49 (2007) No. 4, pp. 280-287.
- Wimmer, R.; Kang, M.J.; Tischner, U.; Verkuil, M.; Fresner, J.; Möller, M. (2008):** Erfolgsstrategien für Produkt-Dienstleistungssysteme. Bundesministerium für Verkehr, Innovation und Technologie, 2008.
- Winter, R. (2009):** Was ist eigentlich Grundlagenforschung in der Wirtschaftsinformatik? In: Wirtschaftsinformatik, Vol. 51 (2009) No. 2, pp. 223-231.
- Wright, P. (1992):** What's in a scenario? In: ACM SIGCHI Bulletin, Vol. 24 (1992) No. 4, pp. 11 - 12.
- Yin, R.K. (2009):** Case Study Research. Design and Methods. (5 ed.), B&T 2009.

- Young, R.R. (2003):** The Requirements Engineering Handbook, Artech House Inc 2003.
- Zacharias, A. (2003):** Quality Function Deployment im Kreditkartengeschäft - Anwendung, Nutzen und Grenzen der Methode bei der Entwicklung von Komponenten in der Finanzdienstleistung. In: Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. Eds.: Bullinger, H.-J.; Scheer, A.-W. Springer 2003.
- Zaeh, M.F.; Reinhart, G.; Pohl, J.; Schindler, S.; Karl, F.; Rimpau, C. (2009):** Modelling, Anticipating and Managing Cyclic Behaviour in Industry. *3rd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2009)*. München.
- Zangemeister, C. (2008):** Innovationsmanagement für Dienstleistungen: Leitfaden, Instrumente, Umsetzungshilfen (Vol. 1) 2008.
- Zarnekow, R. (2007):** Produktionsmanagement von IT-Dienstleistungen: Grundlagen, Aufgaben und Prozesse. (1 ed.), Springer, Berlin 2007.
- Zaugg, R. (2002):** Bezugsrahmen als Heuristik der explorativen Forschung: Grundlagen - Bezugsrahmen - Forschungsstrategien - Forschungsmethode, Universität Bern IOP, Bern 2002.
- Zave, P. (1997):** Classification of Research Efforts in Requirements Engineering. In: ACM Computing Surveys, Vol. 29 (1997) No. 4, pp. 315-321.
- Zellner, G. (2008):** Gestaltung hybrider Wertschöpfung mittels Architekturen – Analyse am Beispiel des Business Engineering. In: Wirtschaftsinformatik, Vol. 50 (2008) No. 3, pp. 187-195.
- Zerbe, S. (1998):** Globale Teams; Ein Ansatz zur Formulierung von Gestaltungsvorschlägen für verteilte, teamorientierte Organisationsformen. Dissertation, Universität Hohenheim 1998.
- Zowghi, D.; Coulin, C. (2005):** Requirements Elicitation: A Survey of Techniques, Approaches. In: Engineering and Managing Software Requirements Eds.: Aurum, A.; Wohlin, C., 1 (ed.). Springer, Berlin 2005, pp. 17-47.

10 Anhang

Interviewleitfaden zum Ist-Zustand des Requirements Engineering
Unternehmen: Interviewte: Rolle des Interviewten: Branche: Datum: Ort:
1. Allgemeine Informationen zum Unternehmen <ul style="list-style-type: none"> • Umsatz gesamt, Deutschland, in Mio. € (2007) • Mitarbeiter gesamt (2007)
2. Produkte des Unternehmens <ul style="list-style-type: none"> • Welche Produkte bietet das Unternehmen an? Bieten Sie auch Dienstleistungen an? Falls ja, werden Dienstleistungen unabhängig vom Produkt angeboten? • Wie lassen sich diese Produkte hinsichtlich folgender Kriterien klassifizieren: <ul style="list-style-type: none"> ▪ Standardsoftware (nicht für einzelnen Kunden, Kundengruppe mit gleichen Problemstellungen) vs. Individualsoftware ▪ Kommerzielle Software vs. Open-Source-Software • Wie würden sie das Vorgehen bei der Entwicklung neuer Produkte beschreiben; folgt ihr Unternehmen einem bestimmten Vorgehensmodell? Und in Bezug auf das Requirements Engineering?
3. Projekte des Unternehmens (ein bestimmtes Projekt) <ul style="list-style-type: none"> • Wie lange dauert ein Projekt durchschnittlich? • Wie entsteht ein Projekt? Werden die Vorgaben von Kunden gemacht? • Wie viele und welche Personen sind am Projekt beteiligt? Es geht um die Rollen der Personen im Unternehmen. • Wurden Kunden in das Projekt eingebunden? Wenn ja, in wieweit?
4. Requirements Engineering <ul style="list-style-type: none"> • Werden die Anforderungen kontinuierlich verwaltet oder nur in der Anfangsphase des Projektes? • Was wird beim Prozess der Verwaltung von Anforderungen genau gemacht? Anforderungserhebung, Interpretation und Strukturierung, Verifikation und Validation, Änderungsmanagement, Anforderungsverfolgung? • Woher kommen die Anforderungen? (Kunde, intern, Berater...) Werden die Anforderungen alle gleich behandelt oder wird beim Umgang mit den Anforderungen einen Unterschied hinsichtlich der Herkunft gemacht? • Wer ist am Prozess der Verwaltung von Anforderungen beteiligt? • Wer übersetzt den Bedarf des Kunden in die Anforderungen? Wie wird das gemacht? • Wie viele Anforderungen sind schon am Anfang der Entwicklung bekannt? Wie viele Anforderungen kommen erst während der Entwicklung dazu? • Wie wird damit umgegangen, wenn Anforderungen erst während der Entwicklung dazukommen? Werden sie alle umgesetzt? • Welche Art von Anforderungen kommt später hinzu? (Funktionalität, Qualität, Nutzeroberfläche,...) • Welcher Aufwand entsteht im Projekt durch später hinzukommende bzw. sich ändernde Anforderungen? • Wenn sich die Anforderungen ändern, was wird dann gemacht? Werden die Änderungen dokumentiert?

<ul style="list-style-type: none"> • Von wem kommen die Änderungen? • Werden Lasten- und Pflichtenhefte erstellt? Falls ja, wann und von wem? • Welche Informationen sind im Pflichtenheft enthalten? Welche Informationen sind im Lastenheft enthalten?
<p>5. Anforderungsermittlung und Anforderungsanalyse</p> <ul style="list-style-type: none"> • Wie werden die Anforderungen erhoben (Es geht um die Techniken, die eingesetzt werden, um die Anforderungen zu ermitteln, z. B. Interviews, Workshops, Beobachtungen, schriftliche Befragungen, Brainstorming, Prototypen, Kartenabfrage, usw.)? • Was sind die häufigsten Probleme, die bei der Ermittlung von Anforderungen auftreten? • Wie werden die Anforderungen analysiert? (Unter Analyse versteht man die Konkretisierung von Anforderungen) • Werden die Anforderungen priorisiert? Falls ja, wie?
<p>6. Anforderungsdokumentation</p> <ul style="list-style-type: none"> • Wie werden die Anforderungen dokumentiert? Gibt es ein Muster (Vorlage), um Anforderungen zu dokumentieren? • Welche Informationen werden dokumentiert? • Wird die Verwaltung von Anforderungen mit Werkzeugen unterstützt? Falls ja, mit welchen? Falls nein, warum nicht?
<p>7. Konflikte im Requirements Engineering</p> <ul style="list-style-type: none"> • Falls es zu Konflikten zwischen den Anforderungen kommt, wie wird damit umgegangen? Unter Konflikten zwischen den Anforderungen ist gemeint, dass zwei oder mehrere Anforderungen sich widersprechen und somit nicht gemeinsam umsetzbar sind. • Werden verschiedene Arten von Konflikten unterschieden? (Zum Beispiel Konflikte, die auf Grund des Mangels an Informationen oder Fehlinformationen verursacht werden, beziehungsweise Interessenkonflikte, die durch verschiedene Interessen oder Ziele von Stakeholdern verursacht werden etc.) • Gibt es Strategien, solche Konflikte aufzulösen?
<p>8. Verifikation und Validierung</p> <ul style="list-style-type: none"> • Wer trifft die Entscheidung, welche Anforderungen umgesetzt werden? • Wie wird überprüft, ob die Eigenschaften des entwickelten Produktes mit definierten Anforderungen übereinstimmen?
<p>9. Prototypen</p> <ul style="list-style-type: none"> • Welchen Zweck oder welcher Funktion erfüllen Prototypen im Rahmen des Anforderungsmanagements bei Ihnen? (Kommunikation, Dokumentation, Verifikation und Validierung, ...) • Für welche Zielgruppe (Stakeholder) werden Prototypen entwickelt? • Wie werden diese Prototypen entwickelt? (Werkzeuge, Sprachen, Frameworks, ...) • Wie schätzen Sie Aufwand/Kosten und Erfolg von Prototypen im Anforderungsmanagement ein? • Was spricht aus Ihrer Sicht für oder gegen den Einsatz von Prototypen im Anforderungsmanagement?
<p>10. Abschließende Fragen zum Thema „Requirements Engineering“</p> <ul style="list-style-type: none"> • Was würden Sie im Bereich der Anforderungsverwaltung in zukünftigen Projekten verbessern? • Welche Aspekte würden Sie als wichtig für die Anforderungsverwaltung betrachten?

Tabelle 10-1: Interviewleitfaden zum Ist-Zustand des Requirements Engineering in Vorstudie I

Quelle: Eigene Darstellung

Leitfaden zum Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern in Unternehmen Alpha
Interviewte: Rolle des Interviewten: Unternehmensbereich: Datum: Ort:
Welche Aktivitäten werden im Rahmen des Requirements Engineering durchgeführt? Was sind die Ergebnisse dieser Aktivitäten? Wer ist an der Durchführung der Aktivitäten beteiligt?
Anforderungsermittlung: <ul style="list-style-type: none"> • Vorgehen • Quellen • Art und Form der ermittelten Anforderungen • Beteiligte Unternehmensbereiche / Personen
Anforderungsanalyse: <ul style="list-style-type: none"> • Vorgehen • Art und Form der analysierten Anforderungen • Beteiligte Unternehmensbereiche / Personen
Anforderungsvereinbarung: <ul style="list-style-type: none"> • Vorgehen • Beteiligte Unternehmensbereiche / Personen
Anforderungsdokumentation: <ul style="list-style-type: none"> • Vorgehen • Art und Form der Dokumentation • Beteiligte Unternehmensbereiche / Personen
Anforderungsverfolgung: <ul style="list-style-type: none"> • Vorgehen • Beteiligte Unternehmensbereiche / Personen
Änderungsmanagement: <ul style="list-style-type: none"> • Vorgehen • Beteiligte Unternehmensbereiche / Personen
Probleme und Herausforderungen im Requirements Engineering
Zyklische Wechselwirkungen im Requirements Engineering

Tabelle 10-2: Leitfaden zum Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern von Unternehmen Alpha in Vorstudie II

Quelle: Eigene Darstellung

Fragebogen zum Requirements Engineering

Dieser Fragebogen dient dazu Ihre Einschätzung zum Requirements Engineering zu erheben.							
		stimme voll und ganz zu	stimme zu	teils / teils	stimme nicht zu	stimme überhaupt nicht zu	nicht zutreffend / keine
1	Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) ist im Allgemeinen zufriedenstellend.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) führt dazu, dass thematisch ähnliche Anforderungen gruppiert abgelegt werden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) ermöglicht es auf einfache Weise Anforderungen zu einem bestimmten Thema wiederzufinden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) ermöglicht es auf einfache Weise, bei der Änderung einer Anforderung, davon betroffene Anforderungen zu finden.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) gibt beim Einfügen von neuen Anforderungen eine eindeutige Stelle vor, an der die Anforderung abgelegt wird.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) ermöglicht eine Nachvollziehbarkeit zwischen Lasten- und Pflichtenheft.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	Die Struktur der Anforderungsdokumente (Lasten- und Pflichtenhefte) hilft bei der Koordination der Arbeit verschiedener beteiligter Bereiche/Abteilungen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	Die Anforderungen im Pflichtenheft sind ausreichend detailliert, so dass ersichtlich ist, wie sie realisiert werden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	Im Lastenheft wird klar festgehalten, warum eine Anforderung vorhanden ist	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tabelle 10-3: Fragebogen zum Ist-Stand des Requirements Engineering im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern von Unternehmen Alpha in Vorstudie II

Quelle: Eigene Darstellung

Spezifikation von Zielen an PSS

1. Ziele an PSS
 - 1.1 Hintergrundinformationen
 - 1.2 Geschäftschancen des Kunden und des Auftragnehmers
 - 1.3 Unternehmensziele des Kunden und des Auftragnehmers
 - 1.4 Beschreibung des Kundenproblems, das mit dem PSS gelöst werden soll
 - 1.5 Kundenziele
 - 1.6 Auftragnehmerziele
 - 1.7 Mögliche Risiken
2. Vision der Lösung
 - 2.1 Abstrakte Beschreibung der Lösung
 - 2.2 Eigenschaften der Lösung
 - 2.3 Annahmen und Abhängigkeiten
3. Abgrenzung und Einschränkungen
 - 3.1 Umfang des initialen Releases
 - 3.2 Umfang folgender Releases
 - 3.3 Einschränkungen

Tabelle 10-4: *Spezifikation von Zielen an PSS*
Quelle: in Anlehnung an (Wiegers 2005, 81)

Systematisierung von Dienstleistungen		
	Kaufmännische Leistungen	Technische Leistungen
Pre-Sales	<ul style="list-style-type: none"> • Wirtschaftlichkeitsanalysen • Beratung • Bestelldienst • Marktinformationen 	<ul style="list-style-type: none"> • Techn. Beratung • Informationsveranstaltungen • F&E-Dienstleistungen • Bedarfsanalyse • Systemanalysen • Feasibility-Studien • Produktdemonstrationen • Testgeräte • Planung und Projektierung • Bereitstellung von • Planungssoftware
Sales	<ul style="list-style-type: none"> • Generalunternehmerschaft • Finanzierung • Leasing • Vermietung • Logistikkonzepte • Absatzgarantien • Rücknahmegarantien 	<ul style="list-style-type: none"> • Engineering • Softwareentwicklung • Transport • Montage • Techn. Anpassungen • Inbetriebnahme • Techn. Schulung • Probefertigung • Techn. Dokumentation
After-Sales	<ul style="list-style-type: none"> • Schulung • Managementberatung • Personalvermittlung • Absatzhilfen 	<ul style="list-style-type: none"> • Wartung • Inspektion • Reparatur • Reinigung • Service-Verträge • Service-Hotline • Ferndiagnose • Tele-Service • Ersatzteilservice • Maschinenverleih/-vermietung, Tauschgeräte • Modernisierung • Rücknahme • Gebrauchtmaschinen-Vermittlung • Demontage • Recycling, Entsorgung

Tabelle 10-5: Systematisierung von Dienstleistungen

Quelle: In Anlehnung an (Backhaus/Weiber 1993; Koch 2010)

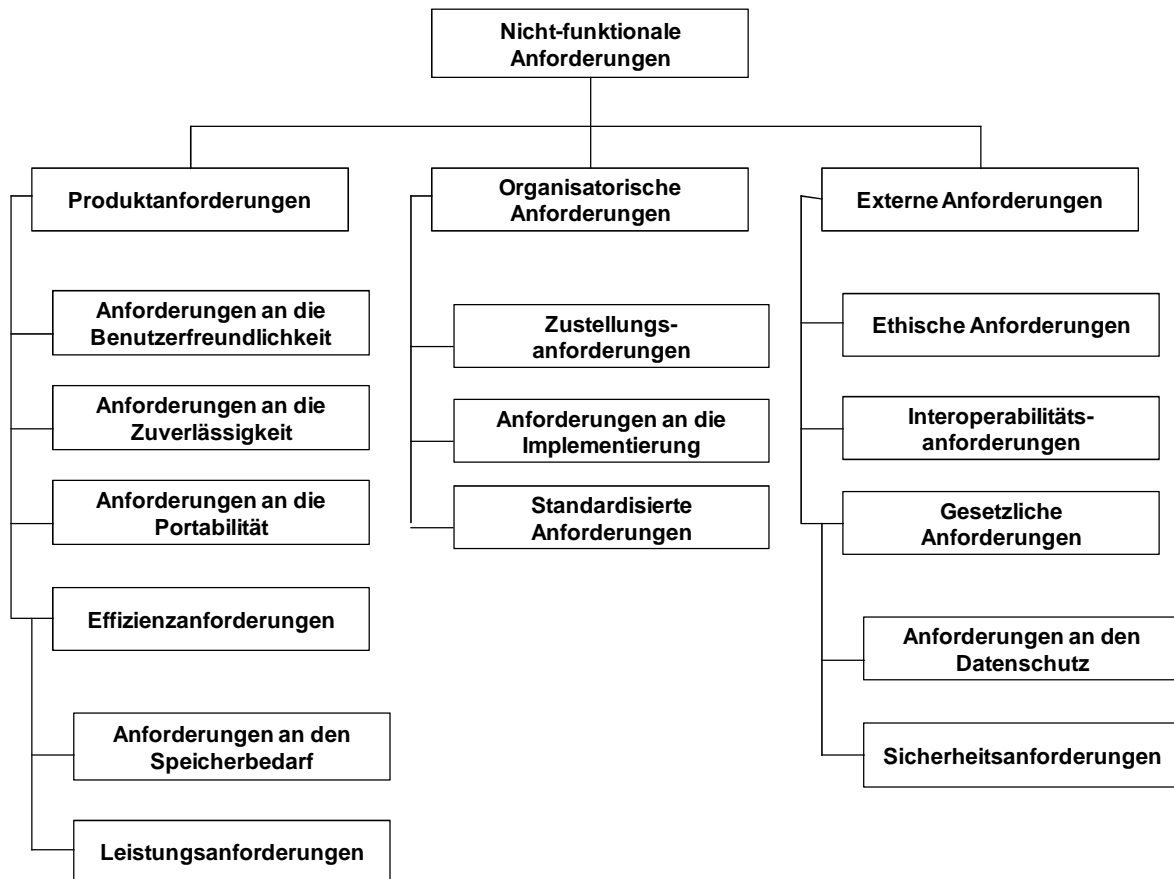


Abbildung 10-1: Kategorisierung nicht-funktionaler Anforderungen

Quelle: In Anlehnung an Sommerville (2006, 124-126)

Hauptmerkmal	Beispiele
Geometrie	Größe, Höhe, Breite, Länge, Durchmesser, Raumbedarf, Anzahl, Anordnung, Anschluss, Ausbau und Erweiterung
Kinematik	Bewegungsart, Bewegungsrichtung, Geschwindigkeit, Beschleunigung
Kräfte	Kraftgröße, Kraftrichtung, Krafthäufigkeit, Gewicht, Last, Verformung, Steifigkeit, Federeigenschaften, Stabilität, Resonanzen
Energie	Leistung, Wirkungsgrad, Verlust, Reibung, Ventilation, Zustandsgrößen wie Druck, Temperatur, Feuchtigkeit, Erwärmung, Kühlung, Anschlussenergie, Speicherung, Arbeitsaufnahme, Energieumformung
Stoff	Physikalische und chemische Eigenschaften des Eingangs- und Ausgangsprodukts, Hilfsstoffe, vorgeschriebene Werkstoffe (Nahrungsmittelgesetz u.ä.), Materialfluss und -transport
Signal	Eingangs- und Ausgangssignale, Anzeigeart, Betriebs- und Überwachungsgeräte, Signalform
Sicherheit	Unmittelbare Sicherheitstechnik, Schutzsysteme, Betriebs-, Arbeits- und Umweltsicherheit
Ergonomie	Mensch-Maschine-Beziehung: Bedienung, Bedienungsart, Übersichtlichkeit, Beleuchtung, Formgestaltung
Fertigung	Einschränkung durch Produktionsstätte, größte herstellbare Abmessung, bevorzugtes Fertigungsverfahren, Fertigungsmittel, mögliche Qualität und Toleranzen
Kontrolle	Mess- und Prüfmöglichkeit, besondere Vorschriften (TÜV, ASME, DIN, ISO, AD-Merkblätter)
Montage	Besondere Montagevorschriften, Zusammenbau, Einbau, Baustellenmontage, Fundamentierung
Transport	Begrenzung durch Hebezeuge, Bahnprofil, Transportwege nach Größe und Gewicht, Versandart und -bedingungen
Gebrauch	Geräuscharmheit, Verschleißrate, Anwendung und Absatzgebiet, Einsatzort (z.B. schwefelige Atmosphäre, Tropen,...)
Instandhaltung	Wartungsfreiheit bzw. Anzahl und Zeitbedarf der Wartung, Inspektion, Austausch und Instandsetzung, Anstrich, Säuberung
Recycling	Wiederverwendung, Wiederverwertung, Entsorgung, Endlagerung, Beseitigung
Kosten	Max. zulässige Herstellkosten, Werkzeugkosten, Investition und Amortisation

Tabelle 10-6: Hauptmerkmalliste zur Charakterisierung von Anforderungen an Hardware-Komponenten

Quelle: Pahl (2006, 220)

4.	<p>Die inhaltliche Anordnung von Anforderungen in Artefakte unterstützt die Stakeholder bei der zielgerichteten Diskussion über die Anforderungen und ihre Umsetzung, da jedes Artefakt nur Anforderungen nach definierten inhaltlichen Kriterien enthält.</p> <p>Kriterium: abgestimmt</p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	<p>Die Strukturierung von Anforderungen in Artefakte und Abstraktionsebenen unterstützt die Eindeutigkeit der Formulierung von Anforderungen durch die Vorgabe von Inhalten aus dem Entwicklungsprozess.</p> <p>Kriterium: eindeutig</p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	<p>Durch die schrittweise Konkretisierung der Anforderungen können die Stakeholder die Übereinstimmung der Anforderungen während der Entwicklung mit den initialen Anforderungen laufend überprüfen, da für jede Anforderung nachvollzogen werden kann, aus welcher Anforderung sie abgeleitet ist.</p> <p>Kriterium: korrekt</p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	<p>Die Verfolgung der Anforderungen über die Phasen des Entwicklungsprozesses, von den initialen bis hin zu den Domänen-Anforderungen, unterstützt die Stakeholder und die Entwickler bei der Überprüfung der Übereinstimmung der komponentenspezifischen Domänen-Anforderungen mit den initialen Anforderungen, da für jede Anforderung nachvollzogen werden kann, aus welcher Anforderung sie abgeleitet ist.</p> <p>Kriterium: korrekt</p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	<p>Durch die Zurückverfolgbarkeit von konfliktären Anforderungen bis zu ihren konfliktfreien Vorgängern ist es möglich die Identifikation der Ursache des Konfliktes in jeder Phase der Entwicklung zu unterstützen, da für jede Anforderung nachvollzogen werden kann, aus welcher Anforderung sie abgeleitet ist.</p> <p>Kriterium: konsistent</p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

9.	<p>Die zu Beginn der Entwicklung erhobenen Anforderungen werden schrittweise gemäß den Phasen der Entwicklung, unter einer ständigen Einbeziehung der Entwicklungsinformationen konkretisiert. Dadurch kann während des Entwicklungsprozesses laufend sichergestellt werden, dass die Anforderungen umsetzbar sind.</p> <p>Kriterium: realisierbar</p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	<p>Die schrittweise Konkretisierung der Anforderungen ermöglicht die Beziehungen von jeder Anforderung zu ihren Vorgängern und Nachfolgern festzuhalten.</p> <p>Kriterium: verfolgbar</p>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tabelle 10-7: Fragebogen zur Evaluation des Requirements Engineering Ansatzes für PSS

Quelle: Eigene Darstellung

Fragebogen zur Evaluation des Requirements Engineering Ansatzes für PSS (Fortsetzung)

		stimme voll und ganz zu	stimme zu	teils / teils	stimme nicht zu	stimme überhaupt nicht zu	nicht zutreffend
11.	Durch die Angabe der Vor- und Nachfolgeranforderungen ist für jede Anforderung eine Begründung für ihre Existenz und somit für den Grund ihrer Realisierung angegeben. Kriterium: Weitgehend keine Unterstützung der Anforderungsverfolgung, Kapitel 4.5.2.2.3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.	Durch die Vorgabe von Artefakten auf verschiedenen Abstraktionsebenen werden inhaltliche Richtlinien für die Erstellung der Anforderungen gegeben, wodurch die Vollständigkeit der Anforderungen unterstützt wird. Kriterium: vollständig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13.	Die Strukturierung von Anforderungen in Abstraktionsebenen, die an die Entwicklungsschritte angelehnt sind, ermöglicht es die Anforderungen desselben Detaillierungsgrades zusammenzufassen. Dadurch erfolgt keine Vermischung von Anforderungen unterschiedlichen Detaillierungsgrades in Lasten- und Pflichtenheften. Kriterium: Unterschiedliche Detaillierungsgrade, Kapitel 4.5.2.2.3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14.	Die Strukturierung von Anforderungen in Artefakte ermöglicht bei Bedarf die Anforderungen wiederzuverwenden. Kriterium: Wiederverwendung von Anforderungen, Kapitel 4.5.2.2.5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tabelle 10-8: Fragebogen zur Evaluation des Requirements Engineering Ansatzes für PSS

Quelle: Eigene Darstellung

Interviewleitfaden zur Evaluation des Requirements Engineering Ansatzes für PSS in Unternehmen im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern <i>Alpha</i>
Interviewte: Rolle des Interviewten: Unternehmensbereich: Datum: Ort:
Unterstützt die Strukturierung der Anforderungen in Artefakten und in Abstraktionsebenen das Abstimmen (korrekte Anforderungen) der Anforderungen zwischen den Stakeholdern? Wenn ja, in wie weit?
Unterstützt die Strukturierung der Anforderungen in Artefakten und in Abstraktionsebenen die Suche nach Konflikten zwischen den Anforderungen? Wenn ja, in wie weit?
Unterstützt die schrittweise Konkretisierung von Anforderungen entlang der Schritte des Entwicklungsprozesses das Verständnis der Stakeholder bezüglich der Anforderungen und ihrer Realisierung? Wenn ja, in wie weit?
Unterstützt die inhaltliche Anordnung von Anforderungen in Artefakte (Anforderungskategorien) die Stakeholder bei der zielgerichteten Diskussion über die Anforderungen? Wenn ja, in wie weit?
Unterstützt die Strukturierung von Anforderungen in Artefakte und Abstraktionsebenen die Eindeutigkeit der Formulierung von Anforderungen? Wenn ja, in wie weit?
Unterstützt die schrittweise Konkretisierung der Anforderungen die laufende Überprüfung der Übereinstimmung der Anforderungen mit den initialen Anforderungen? Wenn ja, in wie weit?
Unterstützt die Verfolgbarkeit der Anforderungen über die Phasen des Entwicklungsprozesses, von den initialen bis hin zu den Domänen-Anforderungen, die Stakeholder und die Entwickler bei der Überprüfung der Übereinstimmung der komponentenspezifischen Domänen-Anforderungen mit den initialen Anforderungen? Wenn ja, in wie weit?
Unterstützt die Zurückverfolgbarkeit von konfliktären Anforderungen bis zu ihren konfliktfreien Vorgängern die Identifikation der Ursache des Konfliktes? Wenn ja, in wie weit?
Unterstützt die schrittweise Konkretisierung der Anforderungen, unter einer ständigen Einbeziehung der Entwicklungsinformationen die Sicherstellung der Umsetzbarkeit der Anforderungen? Wenn ja, in wie weit?
Ermöglicht die schrittweise Konkretisierung der Anforderungen die Beziehungen von jeder Anforderung zu ihren Vorgängern und Nachfolgern festzuhalten (Verfolgung)? Wenn ja, in wie weit?
Wird durch die Angabe der Vor- und Nachfolgeranforderungen für jede Anforderung eine Begründung für ihre Existenz und somit für den Grund ihrer Realisierung angegeben? Wenn ja, in wie weit?
Unterstützt die Vorgabe von Anforderungskategorien (Artefakten) die Vollständigkeit der Anforderungen, indem eine inhaltliche Strukturierung für die Anforderungen vorgegeben wird? Wenn ja, in wie weit?
Erkenntnisse aus der Fallstudie bei Alpha: Unterschiedliche Detaillierungsgrade von Anforderungen in Lasten- und Pflichtenheften. Die Strukturierung von Anforderungen in Abstraktionsebenen, die an

die Entwicklungsschritte angelehnt sind, ermöglicht es die Anforderungen desselben Detaillierungsgrades zusammenzufassen. Dadurch erfolgt keine Vermischung von Anforderungen unterschiedlichen Detaillierungsgrades in Lasten- und Pflichtenheften. Wie würden Sie diese Behauptung bewerten?

Erkenntnisse aus der Fallstudie bei Alpha: Wiederverwendung von Anforderungen: Die Strukturierung von Anforderungen in Artefakte ermöglicht bei Bedarf die Anforderungen in den Artefakten wiederzuverwenden. Wie würden Sie diese Behauptung bewerten?

Table 10-9: *Interviewleitfaden zur Evaluation des Requirements Engineering Ansatzes für PSS im Unternehmensbereich zur Herstellung von Waschmaschinen und Trocknern in Unternehmen Alpha*

Quelle: Eigene Darstellung