

Lehrstuhl für Produktentwicklung
der Technischen Universität München

Anforderungskklärung in interdisziplinärer Entwicklungsumgebung

Christoph Jung

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ. Prof. Dr.-Ing. Hans-Peter Kau
Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. Udo Lindemann
2. Univ.-Prof. Dr.-Ing. Klaus Bender

Die Dissertation wurde am 27.06.2005 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 15.02.2006 angenommen.

Der Unterschied zwischen dem richtigen Wort
und dem beinahe richtigen ist derselbe wie der
zwischen dem Blitz und dem Glühwürmchen.

Mark Twain

VORWORT DES HERAUSGEBERS

Problemstellung

Der Anforderungsklä rung kommt eine besondere Schlüsselstellung für die erfolgreiche Durchführung von Entwicklungsprojekten zu. Um dieser Schlüsselstellung gerecht zu werden, wurden in der Vergangenheit unterschiedliche Methoden und Werkzeuge zur Unterstützung der Anforderungsklä rung entwickelt. Die stark zunehmende Komplexität technischer Produkte führt vermehrt zu einer Integration unterschiedlicher Disziplinen in die Produktentwicklung. Je weiter diese beteiligten Disziplinen fachlich voneinander entfernt sind, desto größer ist der Grad der Interdisziplinarität und desto schwieriger wird die domänenübergreifende Kommunikation und Kooperation. Dies bedingt organisatorische und kommunikative Probleme, führt zum Aufbau von Kommunikationsmauern und erschwert so den Entwicklungsprozess. Davon sind auch die Klärung und Handhabung von Anforderungen betroffen. Dass dies in der Industrie ein großes Problem ist, zeigen zahlreiche Veröffentlichungen zu diesem Thema. Systematische Vorgehensweisen, die den Entwickler bei der Arbeit mit Anforderungen von der Erhebung von Kundenbedürfnissen, der Definition von technischen Anforderungen bis hin zu deren Verwendung besonders in interdisziplinären Entwicklungsprozessen durchgängig anleiten, existieren derzeit jedoch nicht.

Zielsetzung

Durch eine ganzheitliche, entwicklungsbegleitende und systematische Anforderungsklä rung unter kontinuierlicher Kundeneinbindung soll die Anforderungsklä rung verbessert werden. Dabei sollen zwischen den Disziplinen ein gemeinsames Produktverständnis aufgebaut und implizite Wissensbereiche gezielt angestoßen werden, um daraus Anforderungen an das Produkt ableiten zu können. Zur Erreichung dieser Ziele und zur nachvollziehbaren Entwicklung des methodischen Ansatzes werden folgende Fragen behandelt:

- Welche Probleme treten derzeit bei der Anforderungsklä rung in der Praxis auf?
- Wie wird der Prozess der Anforderungsklä rung vonseiten der Entwicklungsmethodik derzeit unterstützt?
- Warum treten in der Praxis bei der Anforderungsklä rung trotz methodischer Unterstützung Probleme auf?
- Wie muss eine Anforderungsklä rung aussehen, um diese Probleme zu vermeiden?
- Wie muss die dazu notwendige entwicklungsmethodische Unterstützung wirken?

Ergebnisse

Es werden ein Vorgehensmodell und eine Methode vorgestellt, die die Anforderungen an die Anforderungsklä rung im interdisziplinären Umfeld erfüllen. Der Schwerpunkt liegt hierbei auf dem eigentlichen Erkennen von Anforderungen. Es wird gezeigt, dass Anforderungen sich

ursächlich auf die Relationen zwischen den Elementen eines Produkts und dessen Umfeld zurückführen lassen. Daraus wird ein allgemeiner Ansatz zum Erkennen von Anforderungen in interdisziplinären Entwicklungsprojekten entwickelt und dann zu einem eigenständigen, ganzheitlichen Vorgehen zur prozessbegleitenden Anforderungsklä rung ausgebaut (relationale iterative Anforderungsklä rung). Diese Methode dient der Klä rung, Festlegung und Verarbeitung von Anforderungen und zur Verbesserung der Kommunikation der unterschiedlichen an einer Entwicklung beteiligten Disziplinen untereinander. Um die Anwendung der Methode in der Praxis zu testen und zu unterstützen wird ein Softwaretool auf der Basis einer relationalen Datenbank als Demonstrator entwickelt und vorgestellt.

Folgerungen für die industrielle Praxis

Durch die ganzheitliche und interdisziplinäre Betrachtung der Anforderungsklä rung werden viele der dabei in der industriellen Praxis auftretenden Probleme thematisiert und zusammengetragen. Die einzelnen Qualitätsmerkmale der Anforderungsklä rung werden herausgearbeitet und in dieser Zusammenstellung vorgestellt. In Verbindung mit der umfangreichen Auswertung des Standes der Technik wird dem Ingenieur in der Praxis zunächst ein Werkzeug an die Hand gegeben seine eigene Situation und sein Vorgehen im Spannungsfeld der Probleme der Anforderungsklä rung zu beurteilen und zu überdenken.

Darüber hinaus wird mit dem vorgestellten Ansatz der „Relationalen Iterativen Anforderungsklä rung“ ein neues Vorgehen für die Anforderungsklä rung zur Verfügung gestellt. Dieses leitet den Ingenieur an, die Anforderungsklä rung aus einer neuen, ganzheitlichen Perspektive zu betrachten und soll ihm helfen unterschiedliche Disziplinen mit ihren eigenen Denkweisen und Begrifflichkeiten zu vereinen und zu einem konstruktiven und zielführenden Dialog zusammenzuführen. Damit wird die Anforderungsklä rung nachhaltig verbessert.

Folgerungen für Forschung und Wissenschaft

Mittels einer umfangreichen Analyse des Standes der Forschung und dem Abgleich mit den in der Praxis auftretenden Problemen wird gezeigt, dass viele der bisherigen Ansätze zur Unterstützung der Anforderungsklä rung in den immer komplexeren und interdisziplinär geprägten Entwicklungsprojekten alleine nicht mehr zielführend sein können.

Der Nachweis, dass Anforderungen auf Relationen zwischen Systemelementen zurückgeführt werden können und das daraus entwickelte Vorgehen der „Relationalen Iterativen Anforderungsklä rung“ stellen eine neue Herangehensweise an die Anforderungsklä rung dar. Mit diesem Ansatz wird der Erforschung der Entwicklungsmethodik im Bereich der Anforderungsklä rung und deren assoziierten Bereichen ein neuer Impuls gegeben, der Raum für weitere Forschungsaktivitäten lässt.

Garching, im Mai 2006

Prof. Dr.-Ing. Udo Lindemann

Lehrstuhl für Produktentwicklung
Technische Universität München

DANKSAGUNG

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Produktentwicklung der Technischen Universität München von Juli 2000 bis Juni 2005.

Mein größter Dank gilt meinem Doktorvater Professor Dr.-Ing. Udo Lindemann für das in mich gesetzte Vertrauen und das Engagement sowie die Unterstützung, mit der er meine Arbeit begleitete. Die mir entgegengebrachte stets konstruktive Kritik gepaart mit der Gewährung gestalterischer Freiheit bildete die Grundlage für das Gelingen der Arbeit.

Professor Dr.-Ing. Klaus Bender danke ich für die Übernahme der Zweitberichterstattung. Für die Übernahme des Vorsitzes der Prüfungskommission und die damit einhergehende organisatorische Abwicklung der Dissertation danke ich Professor Dr.-Ing. Hans-Peter Kau.

Die Erforschung und Entwicklung Medizintechnischer Produkte bildete den Schwerpunkt meiner Forschungstätigkeit am Lehrstuhl. Diese Tätigkeit gab auch den Anstoß zu dieser Arbeit. Daher möchte ich allen Forschungspartnern vom deutschen Herzzentrum München, dem Klinikum Rechts der Isar danken. Besonderer Dank gilt Herrn Dr. Bernhard Voss, Herrn Dr. Rainer Burgkart und Herrn Dipl.-Ing. Tobias Obst für die tolle Projektarbeit und die zahlreichen Beispiele aus der Praxis. Natürlich gilt mein Dank auch allen Industriepartnern: Alle an dieser Stelle zu nennen würde leider den Rahmen sprengen.

Ein besonderer Dank gilt meinen ehemaligen Kolleginnen und Kollegen am Lehrstuhl für die gute Zusammenarbeit in Projekten, Forschung und Lehre. Insbesondere die Zusammenarbeit mit Jens Gramann und Nadja Pecquet auf dem Gebiet der Medizintechnik hat wertvollen Input für diese Arbeit geliefert. Für die zahlreichen hilfreichen Kommentare und Diskussionen während der Erstellung dieser Arbeit möchte ich besonders Christoph Baumberger, Philipp Hutterer und Hans Stricker danken. Auch möchte ich mich bei allen Diplomanden, Semestranten und studentischen Hilfskräften für die zahlreiche und wertvolle Unterstützung bedanken, stellvertretend besonders bei Wolfgang Lauer und Iris Markfort.

Mein abschließender Dank gilt meiner Familie und meinen Freunden, die mir auf meinem Weg stets Unterstützung und Förderung aber vor allem in der heißen Phase Geduld und Verständnis entgegenbrachten. Besonders möchte ich mich hierfür bei meiner Frau Nikola bedanken, die mir nicht nur den nötigen Freiraum geschaffen hat, sondern es vor allem als Germanistin auf sich nahm, die literarischen Ergüsse eines Ingenieurs akribisch zu korrigieren.

Eching, im Mai 2006

Christoph Jung

Anmerkung:

Einige in dieser Abhandlung verwendete Begriffe sind zum besseren Verständnis im Glossar (Kapitel 11) definiert. Im Text sind diese Begriffe bei ihrer ersten Nennung innerhalb eines Teilkapitels kursiv dargestellt.

INHALTSVERZEICHNIS

1	Einleitung	1
1.1	Ausgangssituation	1
1.2	Zielsetzung der Arbeit	5
1.3	Einschränkung des Themengebietes	6
1.4	Grundlagen und Basis der Arbeit	7
1.5	Aufbau der Arbeit	8
2	Warum neue Methoden zur Anforderungsklä rung?	9
2.1	Beispiele aus der Praxis	9
2.1.1	Entwicklung eines medizintechnischen Gerä ts	9
2.1.2	Entwicklung eines Druckers	15
2.2	Beobachtungen in der Projektarbeit	16
2.3	Auswertung der Beobachtungen	17
2.3.1	Makrobetrachtung	17
2.3.2	Mikrobetrachtung	20
2.4	Zusammenfassung	23
3	Anforderungsklä rung in der Entwicklungsmethodik – eine Bestandsaufnahme	25
3.1	Anforderungsklä rung in umfassenden Modellen der Entwicklungsmethodik	25
3.1.1	Klä rung des Konstruktionsauftrags nach Breiing & Flemming	25
3.1.2	“Setting Requirements” nach Cross	27
3.1.3	Aufgabenklä rung und -strukturierung nach Ehrlenspiel	29
3.1.4	Methoden zur Anforderungsklä rung nach Lindemann	31
3.1.5	Klä ren und Präzisieren der Aufgabenstellung nach Pahl et al.	34
3.1.6	„Target Specifications“ nach Ulrich & Eppinger	36
3.1.7	Ermittlung der Kundenbedürfnisse nach Urban & Hauser	39
3.2	Anforderungsklä rung in unabhängigen Ansätzen	40
3.2.1	Erfassen und Handhaben von Produktanforderungen nach Ahrens	40
3.2.2	Ganzheitliches Anforderungsmanagement nach Danner	42

3.2.3	Anforderungskklärung nach Franke & Roth	43
3.2.4	Erfassung und Weiterverwendung von Anforderungen nach Gröger	45
3.2.5	Anforderungskklärung nach Größer	46
3.2.6	Objektorientiertes Anforderungsmodell nach Humpert	49
3.2.7	Gewinnung von Anforderungen nach Kruse	53
3.3	Requirements Engineering	57
3.4	Analyse der vorgestellten Ansätze	60
3.4.1	Ursachenanalyse	61
3.4.2	Extraktion der eingesetzten Methoden	72
3.4.3	Ableitung von Handlungsschwerpunkten	75
3.5	Zusammenfassung	76
4	Formulierung des Forschungsbedarfs	77
5	Methode zur Unterstützung der entwicklungsbegleitenden Anforderungskklärung	81
5.1	Über das Entstehen und Erkennen von Anforderungen	81
5.2	Vorgehen zur relationalen iterativen Anforderungskklärung	85
5.2.1	Iterativer Systemaufbau	85
5.2.2	Relationenanalyse	89
5.2.3	Ableitung von Anforderungen	95
5.2.4	Vorgehensmodell der relationalen iterativen Anforderungskklärung	97
5.3	Beschreibung eines ganzheitlichen Ansatzes zur Anforderungskklärung	98
5.3.1	Kernwirkung der relationalen iterativen Anforderungskklärung	98
5.3.2	Nebenwirkung der relationalen iterativen Anforderungskklärung	100
5.4	Werkzeug für die relationale iterative Anforderungskklärung	108
5.4.1	Anforderungen an ein unterstützendes Tool	108
5.4.2	Systemauswahl	108
5.4.3	Beschreibung des Werkzeugs RIA	109
5.5	Zusammenfassung	120
6	Fallbeispiele	121
6.1	Anforderungskklärung für eine Applikationsspritze	121
6.1.1	Projektbeschreibung	121

6.1.2	Erfahrungen	129
6.2	Entwicklung einer Templatevorrichtung für chirurgische Gelenkkorrekturen	130
6.2.1	Projektbeschreibung	130
6.2.2	Erfahrungen	136
6.3	Auswertung des Methodeneinsatzes	137
7	Zusammenfassung und Ausblick	141
7.1	Zusammenfassung	141
7.2	Ausblick	143
7.2.1	Methodenoptimierung	143
7.2.2	Erweiterte Verifikation und Methodenimplementierung	145
8	Literaturverzeichnis	147
9	Anhang	161
9.1	Leitlinie mit Relationsattributen	161
9.2	Funktionale Anforderungen an RIA	162
10	Dissertationsverzeichnis des Lehrstuhls für Produktentwicklung	165
11	Glossar	175

1 Einleitung

Beim Bau einer Rheinbrücke zwischen dem deutschen und dem schweizer Teil der Stadt Laufenburg ist eine peinliche Panne passiert: Zwischen den beiden Seiten bestand ein Höhenunterschied von 54 Zentimetern. Dies hatte zur Folge, dass der Straßenanschluss auf der deutschen Seite entsprechend tiefer gelegt werden musste. Wie kam es zu einem solchen fatalen Fehler? Die Ursache lag in der grundsätzlich anderen Höhenberechnung auf den beiden Seiten der Grenze. Während die Schweiz das Niveau des Mittelmeeres zu Grunde legt, orientiert sich Deutschland an der Nordsee. Die dazwischen liegende Differenz von 27 Zentimetern ist zwar bekannt, statt aber das Niveau der Brücke auf der schweizerischen Seite um 27 Zentimetern anzuheben, ist sie um denselben Betrag abgesenkt worden, so dass es zu den genannten 54 Zentimetern Höhendifferenz kam. Aufgefallen ist der Fehler anscheinend erst, als der Brückenbogen sich nicht mehr weit vom deutschen Ufer befand und der Höhenunterschied mit bloßem Auge zu sehen war. [STUTTGARTER ZEITUNG 2004]

1.1 Ausgangssituation

Der Erfolg eines Produkts am Markt hängt von vielfältigen Faktoren ab: Vor allem sind dies die Kosten und die Qualität eines Produkts sowie die bis zum Markteintritt benötigte Zeit im Vergleich zu Konkurrenzprodukten [LINDEMANN 2005, S. 7]. Im Spannungsfeld dieser Faktoren ist es das Ziel eines jeden Unternehmens, die Kosten bei einer gleichzeitigen Steigerung der Qualität und Funktionalität zu reduzieren [OTTO & AHRENS 1997, S. 19]. In diesem Zusammenhang ist eine frühzeitige, sorgfältige und umfassende Analyse der zu erfüllenden Aufgabe notwendig, um alle relevanten *Anforderungen* zu erfassen. Das Vergessen auch nur einer Anforderung kann schwere, oft kostspielige Konsequenzen zur Folge haben. Die nachfolgende Abbildung verdeutlicht diesen Zusammenhang. Hat man in den frühen Phasen der Produktentwicklung noch viele Änderungsmöglichkeiten bei vergleichsweise geringen Kosten, nehmen diese mit fortschreitender Konkretisierung des Produkts bei gleichzeitig steigenden Änderungskosten rapide ab [EHRENSPIEL ET AL. 2000, S. 10].

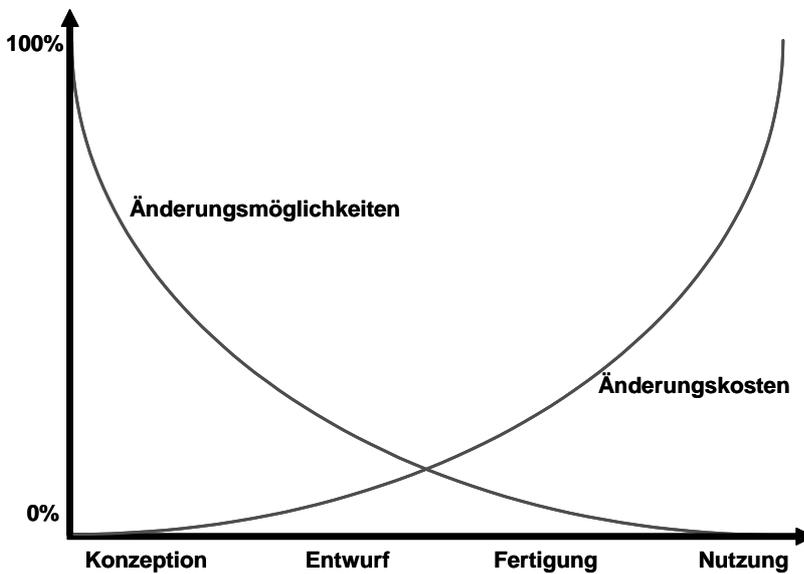


Abb. 1-1: Möglichkeiten der Kostenbeeinflussung und Kostenfestlegung während der Produktentwicklung
[nach LINDEMANN 2005, S. 145]

In diesem Zusammenhang ist es leicht ersichtlich, dass der *Anforderungsklä rung* eine besondere Schlüsselstellung für die erfolgreiche Durchführung von Entwicklungsprojekten zukommt. Werden die Anforderungen, also der Leistungsumfang eines Produkts zu Beginn der Entwicklung nicht ausreichend und sorgfältig genug geklärt oder während des weiteren Produkterstellungsprozesses nicht kontinuierlich ergänzt und aktualisiert, kann dies zu schwerwiegenden Problemen führen. Diese können sich dann zum Beispiel in Form von Budget- und Zeitüberschreitungen äußern [BARRENSCHEEN ET AL. 1989, S. 84; GRÖBER 1992, S. 1; WARD ET AL. 2003; KARCHER ET AL. 1998; VDI-RICHTLINIE 2222 1997, S. 12]. Im schlimmsten Fall kann dies dazu führen, dass ein Produkt am Markt keinen Erfolg hat oder dessen Entwicklung sogar vor Markteinführung eingestellt wird. Neben vielen anderen Faktoren ist also auch die Anforderungsklä rung in den frühen Phasen des Entwicklungsprozesses für den Erfolg eines Produktes von entscheidender Bedeutung.

Oftmals ist jedoch der größte Teil der Anforderungen in der Form, wie er in das Projekt hineingetragen wird, nur bedingt zu nutzen. Es obliegt dann dem Entwickler oder einem Projektteam, diese Anforderungen gewinnbringend aufzubereiten [KRUSE 1996, S. 67]. Zum Beispiel äußert der *Kunde* als Anforderung an einen Sportwagen „gute Kurvenlage“, welche technischen Anforderungen dafür notwendig sind, muss der Entwickler herausarbeiten. Hierbei kommt dem Einsatz von Methoden zur Klärung von Anforderungen eine besondere Rolle zu. Unterschiedliche Methoden, deren Wirkungsweise zum Beispiel auf Abstraktion und Assoziation beruht, wirken hier zusammen, um Unvollständigkeiten, Unklarheiten, Widersprüche und Fehler in der Aufgabenstellung zu vermeiden [BAUMANN 1982, S. 24].

Seit Elektronik- und Softwarekomponenten im klassischen Maschinenbau Einzug gehalten haben, führt die damit zunehmende Komplexität technischer Produkte vermehrt zu einer Integration unterschiedlicher Disziplinen in die Produktentwicklung. Im Vordergrund stehen dabei heute Maschinenbau, Elektrotechnik und Informatik, die unter dem Kunstwort „Mechatronik“ zusammengefasst werden [HARASHIMA ET AL. 1996]. Entwicklungsprozesse in einer solchen interdisziplinären Umgebung bedürfen eines noch höheren Maßes an Unterstützung durch geeignete Methoden, als „klassische Maschinenbauprojekte“ [LINDEMANN ET AL. 2003, S. 102]. Solche Prozesse erfordern „domänenübergreifende Kommunikation und Kooperation zwischen den beteiligten Fachdisziplinen, um eine gemeinsame Vorstellung des zukünftigen Produkts zu gewinnen und eine gesamtoptimierte Lösung herbeizuführen.“ [VDI 2206 2004, S. 4] Diese domänenübergreifende Kommunikation und Kooperation wird aber dadurch begünstigt, dass sich Elektrotechniker, Maschinenbauer und Informatiker in technischen Welten bewegen. Dies hat ähnliche Begriffswelten und vergleichbare Denkstrukturen zur Folge und begünstigt den Prozess der Integration der Disziplinen, auch wenn dabei Hürden zu überwinden waren und auch noch zu überwinden sind.

Geht man nun aber einen Schritt weiter und fügt den Disziplinen der „klassischen“ Mechatronik noch eine weitere, nichttechnische Disziplin hinzu, wie es zum Beispiel bei der Entwicklung von medizintechnisch-mechatronischen Produkten der Fall ist, entsteht eine neue Dimension der interdisziplinären Zusammenarbeit. Dann nimmt die zusätzliche Disziplin, zum Beispiel die Medizin als produktbestimmende Disziplin, da sie die Kernanforderungen an das zu entwickelnde Produkt stellt, maßgeblichen Einfluss auf alle anderen, produktschaffenden Disziplinen. Je weiter diese beiden Gruppen fachlich voneinander entfernt sind, desto größer ist der Grad der Interdisziplinarität. Bei hochgradig interdisziplinären Projekten kann man dann von fremddisziplinär bestimmten Projekten sprechen. Wenn im Rahmen dieser Arbeit von interdisziplinären Projekten gesprochen wird, sind genau solche *fremddisziplinär bestimmte* Projekte gemeint.

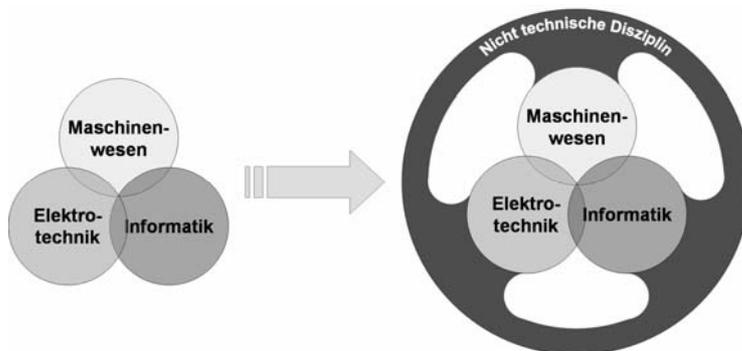


Abb. 1-2: Von der klassischen zur fremddisziplinär bestimmten Mechatronik [LINDEMANN ET AL. 2003, S. 103]

Bei den Beteiligten herrschen meist unterschiedliche Begriffswelten und Denkstrukturen, weshalb sie „unterschiedliche Sprachen“ – sowohl in fachlicher als auch in

zwischenmenschlicher Hinsicht – sprechen [KRUSE 1996, S. 18]. Auch ist die Herangehensweise an eine Aufgabenstellung, sowie die Sichtweise und Beurteilung derselben eine andere. Dies bedingt organisatorische und kommunikative Probleme, führt zum Aufbau von Kommunikationsmauern und erschwert so den kompletten Entwicklungsprozess von der Problemstellung bis zum fertigen Produkt. Davon sind auch die Klärung und Handhabung von Anforderungen betroffen. Zum einen wird das Spektrum, also die Vielzahl und Vielfalt der zu berücksichtigenden Anforderungen noch komplexer und ist aufgrund der fachlichen Unterschiede nur schwer zu überblicken, zum anderen ist die Gefahr aufgrund der Kommunikationsmauern Anforderungen zu vergessen, zu verfälschen oder falsch zu gewichten noch größer [AHRENS 2000, S. 33; KRUSE 1996, S. 9; SCHWANKL 2002, S. 42; LINDEMANN ET AL. 2003, S. 102 ff].

In unterschiedlichen Entwicklungsprojekten konnte immer wieder Folgendes beobachtet werden: Der Entwickler versuchte die Anforderungen an das zu entwickelnde Produkt durch den Einsatz entsprechender Methoden möglichst vollständig und korrekt zu klären. Viele der Anforderungen konnten vom Ingenieur durch entsprechende Recherchen im Selbststudium ermittelt werden. Für viele andere Anforderungen war aber der direkte Kontakt zum Kunden notwendig. Dem Kunden waren aber aufgrund seiner anderen Denk- und Begriffswelt die Auswirkungen fehlender oder unvollständiger Anforderungen oft nicht bewusst, was zu Fehlentwicklungen führte. Erschwerend kam hinzu, dass viele Informationen bei den Beteiligten nur implizit¹ vorhanden waren.

Trotz des konsequenten Einsatzes von Methoden zur Anforderungsklä rung musste immer wieder festgestellt werden, dass Anforderungen nicht korrekt geklärt oder vergessen worden waren [LINDEMANN ET AL. 2003]. Dies hatte unterschiedliche Konsequenzen, führte aber meist zu ärgerlichen und teils langwierigen Verzögerungen in denen die Konzepte überarbeitet und angepasst werden mussten. Hätte es sich bei diesen Entwicklungen um Serienentwicklungen gehandelt, wären das Budget, der Termin der Markteinführung, die Qualität und damit der mögliche Markterfolg des Produkts in Frage gestanden. Dass dies in der Industrie nach wie vor ein großes Problem ist, zeigen zahlreiche Veröffentlichungen zu diesem Thema [siehe ALMEFELT ET AL. 2003; HAVELKA 2002; SALUSTRI & PARMAR 2004; WARD ET AL. 2003].

Zusammenfassend lässt sich festhalten, dass den Anforderungen und damit einer sorgfältigen Anforderungsklä rung eine hohe Bedeutung im Entwicklungsprozess zukommt, da sie die Qualität des späteren Produkts in entscheidender Weise mitbestimmen [DGQ 1995, S. 7 ff]. Gerade in interdisziplinären Projekten gestaltet sich diese oftmals schwierig und birgt somit ein großes Risiko aufgrund falscher oder fehlender Anforderungen Entwicklungsrückschlä ge zu erfahren [siehe WARD ET AL. 2003]. Systematische Vorgehensweisen, die den Entwickler bei der Arbeit mit Anforderungen von der Erhebung von Kundenbedürfnissen, der Definition von technischen Anforderungen bis hin zu deren Verwendung besonders in interdisziplinären Entwicklungsprozessen durchgängig anleiten, existieren derzeit jedoch nicht [siehe auch AHRENS 2000, S. 2].

¹ *Implizites Wissen ist Wissen, welches unbewusst vorhanden ist, aber aus unterschiedlichsten Gründen nicht expliziert wird oder werden kann [HERBIG 2001, S. 24].*

1.2 Zielsetzung der Arbeit

Im Rahmen der vorliegenden Arbeit soll gezeigt werden, dass gerade die in interdisziplinären Projekten immer wieder auftretenden Probleme oftmals auf eine fehlerhafte Anforderungsklä rung zurückzuführen sind. Die Ursachen dafür sind in einer unzureichenden Unterstützung dieser Projekte durch das vorhandene methodische Instrumentarium zu suchen:

- Vorhandene Methoden wirken vor allem in frühen Phasen des Entwicklungsprozesses. Viele *Anforderungen* entstehen aber erst im späteren Verlauf der Entwicklung und werden oft nicht mehr erfasst.
- Das Produkt wird bei der *Anforderungsklä rung* in frühen Phasen der Entwicklung in seiner Gesamtheit betrachtet; eine Förderung der systematischen Betrachtung der Systembestandteile über den Entwicklungsprozess hinweg findet aber nicht statt. So gehen Anforderungen auf höherer Detaillierungsstufe verloren. Die von vielen Autoren [EHRENSPIEL 2003; PAHL ET AL. 2003; KOLLER 1985; ROTH 2000 etc.] geforderte Pflege und Ergänzung der Anforderungen über den Entwicklungsprozess wird von methodischer Seite her nicht gesondert betrachtet.
- Eine Anforderung kann als die Umschreibung eines Problems gesehen werden. Ein Problem kann man nur lösen, wenn man es als solches identifiziert hat. Gleiches gilt auch für Anforderungen: Erst wenn man die Notwendigkeit einer Anforderung als solche erkannt hat, kann man diese auch klären. Die vorhandenen Methoden zur Anforderungsklä rung können dies aber nur bedingt unterstützen, da durch sie meist nur die offensichtlichen und bekannten „Probleme“ angesprochen werden.
- Gerade in interdisziplinären Projekten ist es von großer Bedeutung, den *Kunden* als *Experten* in die Entwicklung einzubinden [DRAKE ET AL. 1993, S. 177; GERHARD 1998, S. 49; KRUSE ET AL. 1997; OTTO & AHRENS 1997; LINDEMANN ET AL. 2003]. Dabei treten typische Probleme der Interdisziplinarität auf, wie zum Beispiel Kommunikationsprobleme [REINICKE 2004, S. 47], Vorbehalte gegenüber der anderen Disziplin [GRÖBER 1992, S. 2; DOHMEN 2002, S. 58] und die fehlenden Nennungen (Explikation) *impliziter Anforderungen*. In diesem Kontext wird die Kundeneinbindung von der Entwicklungsmethodik für die Anforderungsklä rung nur bedingt gefordert, die methodische Unterstützung zur Vermeidung der angesprochenen Schwierigkeiten fehlt.
- Die bisher eingesetzten *Methoden* ermöglichen keine Abschätzung der Anforderungen auf weitgehende Vollständigkeit [MONTAU 1996; S. 68].

Aufgrund der oben aufgezeigten großen Bedeutung der Anforderungen im Entwicklungsprozess und der angesprochenen Schwierigkeiten mit den vorhandenen Methoden erscheint der Aufwand notwendig und angemessen, das Risiko fehlerhafter Anforderungsklä rung zu vermindern. Die vorliegende Arbeit soll einen Beitrag zur Verbesserung der Anforderungsklä rung leisten, indem Methoden und Werkzeuge für die Anforderungsklä rung speziell in interdisziplinären Projekten entwickelt und angepasst werden. In den Augen des Verfassers kann durch eine ganzheitliche, entwicklungsbegleitende und systematische Anforderungsklä rung unter kontinuierlicher Kundeneinbindung die

Anforderungsklärung verbessert werden. Dabei muss zwischen den Disziplinen ein gemeinsames Produktverständnis aufgebaut und implizite Wissensbereiche gezielt angestoßen werden, um daraus Anforderungen an das Produkt ableiten zu können. Zur Erreichung dieses Ziels und zur nachvollziehbaren Entwicklung des methodischen Ansatzes sollen folgende Fragen behandelt werden:

- Welche Probleme treten derzeit bei der Anforderungsklärung in der Praxis auf?
- Wie wird der Prozess der Anforderungsklärung von Seiten der Entwicklungsmethodik derzeit unterstützt?
- Warum treten in der Praxis bei der Anforderungsklärung trotz methodischer Unterstützung Probleme auf?
- Wie muss eine Anforderungsklärung aussehen, um diese Probleme zu vermeiden?
- Wie muss die dazu notwendige entwicklungsmethodische Unterstützung wirken?

1.3 Einschränkung des Themengebietes

Diese Arbeit will nicht die Entwicklungsmethodik hinsichtlich der *Anforderungsklärung* vollkommen neu aufrollen und gestalten. Vielmehr soll ein Beitrag geleistet werden, die durch die stark zunehmende Integration unterschiedlicher Disziplinen in Entwicklungsteams gesteigerte Komplexität der Entwicklung und die dadurch entstehenden Probleme bei der Anforderungsklärung [siehe auch AHRENS 2000, S. 1 ff; WARD ET AL. 2003] zu verringern. Deswegen liegt ein Fokus dieser Arbeit auf hochgradig interdisziplinären (*fremddisziplinär bestimmten*) Entwicklungsprojekten wie sie zum Beispiel bei der Entwicklung medizintechnischer [LINDEMANN ET AL. 2003] oder verfahrenstechnischer [KRUSE 1996] Produkte vorliegen².

Des Weiteren soll die Art der betrachteten Entwicklungsprojekte hinsichtlich der zu Beginn einer Entwicklung vorhandenen Informationen eingeschränkt werden. Das Spektrum der Entwicklungsprojekte kann sich generell von Wiederholteilentwicklungen über Anpassungsentwicklungen, Variantenentwicklungen, Diversifikationen [WAHRIG 2002] bis hin zu Neuentwicklungen erstrecken. Dabei bieten erstere den Vorteil, dass man hier in unterschiedlicher Intensität auf vorhandene Informationen aus vorangegangenen Projekten wie Anforderungslisten, Zeichnungen, Berechnungen etc. zurückgreifen kann und so eine im hohen Maße abgesicherte Anforderungsklärung durchführen kann [siehe auch PAHL ET. AL. 2003, S. 200]. Im Gegensatz dazu stehen die Neuentwicklungen und Diversifikationen. Hier liegen zu Projektbeginn meist wenige Informationen vor, es kann nicht oder nur bedingt auf vergangene Entwicklungen zurückgegriffen werden. Nach Kruse [KRUSE 1996, S. 76] zeigen sich gerade dort besondere Schwierigkeiten bei der Anforderungsklärung.

² Im Folgenden wird in diesem Zusammenhang nur noch von interdisziplinären Projekten gesprochen.

Da die beschriebenen Schwierigkeiten bei der Anforderungsklä rung vor allem bei der Klärung *technischer Anforderungen* in interdisziplinären Projekten und dort vor allem bei Neuentwicklungen und Diversifikationen beobachtet wurden, soll der hier entwickelte und vorgestellte methodische Ansatz besonders in diesen Situationen den Entwickler unterstützen, *technische Anforderungen* systematisch und möglichst vollständig unter Einbindung des *Kunden* oder sonstiger *Experten* zu klären [siehe auch WARD ET AL. 2003, S. 5].

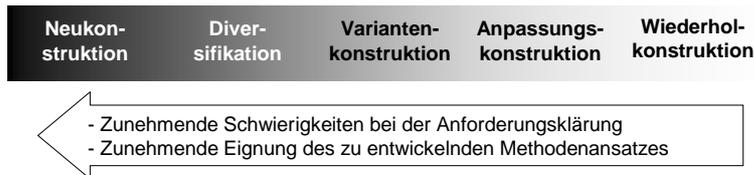


Abb. 1-3: Abgrenzung des Einsatzbereichs des vorgestellten Methodenansatzes

1.4 Grundlagen und Basis der Arbeit

Die Grundlagen der vorliegenden Arbeit wurden in unterschiedlichen Forschungsprojekten des Lehrstuhls für Produktentwicklung der Technischen Universität München erarbeitet. Dies waren insbesondere:

- „Hartwirbeln von Präzisions-Innengewinden“; Förderer: Bayerische Forschungsförderung; Laufzeit: 01.01.2001-31.12.2001
- „Optimierung der Kopplung zwischen extrakorporaler Pumpe und Organismus: Die Herz-Lungen-Maschine als autonomer Roboter.“; Förderer und Projektpartner: Deutsches Herzzentrum München; Laufzeit: 01.01.2001 – 31.12.2002
- „MoKeyS – Mobile Keyhole Surgery“: Entwicklung eines neuartigen, kompakten und günstigen minimalinvasiven Chirurgieroboters; Laufzeit: ab 01.01.2003
- Verschiedene Kooperationen im Bereich der gynäkologischen und orthopädischen Simulation; Projektpartner: Klinikum München Rechts der Isar (MRI)
- „Implicit knowledge in the product innovation process“. Förderer: Volkswagenstiftung; Projektpartner: Lehrstuhl für Psychologie, Technische Universität München; Laufzeit: 01.11.2004 – 30.11.2006 (Der Autor wirkte bis 30.06.2005 an diesem Projekt mit).

Darüber hinaus fanden intensive Diskussionen dieser Thematik mit Kooperationspartnern in der Industrie und mit Mitarbeitern des Lehrstuhls für Produktentwicklung statt. Dabei wurden deren Erfahrungen im Bereich der Anforderungsklä rung aufgenommen und ergänzend den eigenen hinzugefügt, um sich so ein umfassendes Bild von der angesprochenen Problematik schaffen zu können.

1.5 Aufbau der Arbeit

Um einen besseren Überblick über die Arbeit zu bekommen, ist nachfolgend der grundsätzliche Aufbau der Arbeit (Kapitel 2 bis Kapitel 7) beschrieben.

In **Kapitel 2** wird eine Analyse unterschiedlicher durchgeführter Entwicklungsprojekte vorgenommen. Zur weiteren Vertiefung werden grundsätzliche Betrachtungen zu den bei der Anforderungsklä rung ablaufenden Prozessen und den darin verborgenen Fehlermöglichkeiten angestellt. Die Ergebnisse werfen die Frage auf, wie es zu diesen Problemen kommt und wie man diesen entgegenwirken kann.

Um diese Frage umfassend beantworten zu können, wird in **Kapitel 3** der Stand der Technik hinsichtlich der Anforderungsklä rung genauer analysiert. Hierzu werden zum einen Ansätze zur Anforderungsklä rung in umfassenden Modellen der Entwicklungsmethodik, zum anderen spezielle, rein auf die Anforderungsklä rung fokussierte Ansätze betrachtet. Dies dient zum einen für die sich anschließende Analyse der Ursachen der geschilderten Probleme bei der Anforderungsklä rung, zum anderen soll damit für den interessierten Leser ein umfassender Überblick über die zur Verfügung stehenden Methoden zur Anforderungsklä rung gegeben werden. Die Ergebnisse aus Kapitel 2 und 3 werden zusammengeführt und ausgewertet. Es wird herausgearbeitet, warum die Anforderungsklä rung gerade in interdisziplinärer Projektumgebung derzeit nicht ausreichend unterstützt wird.

In **Kapitel 4** werden ausgehend von den erarbeiteten Ursachen Anforderungen an den Prozess der Anforderungsklä rung und die dafür bereitgestellten Methoden erarbeitet und ein grundsätzliches Vorgehensmodell für die Anforderungsklä rung entwickelt.

In **Kapitel 5** wird eine Methode entwickelt, die die erarbeiteten Anforderungen an die Anforderungsklä rung erfüllt und sich in das vorgestellte Vorgehensmodell einordnen lässt. Diese Methode dient der Klä rung, Festlegung und Verarbeitung von Anforderungen und zur Verbesserung der Kommunikation der unterschiedlichen Disziplinen untereinander. Für die Verifikation der Methode wird ein einfaches Softwarewerkzeug vorgestellt.

Die in Kapitel 5 erarbeitete Methode wird in **Kapitel 6** einer ersten Verifikation unterzogen. Dazu wurden die Methode und das dazu entwickelte Softwaretool in unterschiedlichen interdisziplinären Entwicklungsprojekten eingesetzt. Der Methodeneinsatz wird einer Auswertung unterzogen um die Vor- und Nachteile und Optimierungspotential herauszuarbeiten.

Kapitel 7 fasst das Vorgehen und die Ergebnisse nochmals zusammen und gibt einen Ausblick über den weiteren Handlungsbedarf zur Verbesserung der Methode und deren Eingliederung in die bestehende Methodenlandschaft.

2 Warum neue Methoden zur Anforderungsklärung?

In diesem Kapitel wird anhand von unterschiedlichen Beispielen gezeigt werden, dass es Probleme in der Praxis der Anforderungsklärung gibt. Dazu werden Praxisbeispiele herausgegriffen und detaillierter erläutert. Im ersten Teil wird dies anhand eines Forschungsprojektes aus der Herzchirurgie, an dem der Autor maßgeblich beteiligt war, dargestellt. Anforderungen wurden aufgrund von Wissensdefiziten und Kommunikationsschwierigkeiten vergessensweise falsch geklärt, was weitreichende Folgen hatte. Der zweite Teil dieses Kapitels befasst sich mit einem Projekt aus dem Bereich der Entwicklung von Druckern. Aufgrund von zu eng gefassten beziehungsweise ungenügend geklärten Anforderungen konnte das eigentliche Projektziel, nämlich die Entwicklung von grundsätzlichen Lösungen, nicht erreicht werden. Der dritte Abschnitt fasst Erfahrungen aus weiteren Industrieprojekten und die dort festgestellten Probleme zusammen. Anschließend werden die gesammelten Erfahrungen zusammengefasst und einer ersten Auswertung unterzogen.

2.1 Beispiele aus der Praxis

2.1.1 Entwicklung eines medizintechnischen Geräts

Das hier beschriebene Projekt wurde gemeinsam mit der Klinik für Herz- und Gefäßchirurgie und der Experimentellen Chirurgie des Deutschen Herzzentrums München durchgeführt. Weiterführende Informationen sind in unterschiedlichen Veröffentlichungen [LINDEMANN ET AL. 2003; JUNG & GRAMANN 2004; VOSS ET AL. 2005] und im Projektbericht [JUNG ET AL. 2003] zu finden.

Die extrakorporale Zirkulation (EKZ)³ gehört zu den Routineverfahren der offenen Herzchirurgie. Konventionelle Herz-Lungen-Maschinen⁴ liefern einen nahezu konstanten Blutdruck und -fluss, der mit den physiologischen Gradienten des Blutdrucks- und Flusses des Menschen nicht vergleichbar ist. Dies führt zu einer Zunahme des Widerstands der peripheren Blutgefäße und einer Verschlechterung der kapillaren Durchblutung. Folge dieser Minderdurchblutung verschiedener Organsysteme ist die Bildung von Gewebshormonen

³ EKZ=Extrakorporale Zirkulation: Blutumleitung außerhalb des Körpers zur zeitweiligen Ausschaltung des Herzens oder eines Kreislauf- beziehungsweise Gefäßabschnittes bei Operationen am eröffneten Herzen beziehungsweise an großen Blutgefäßen. Die Funktionen des Herzens und der Lunge übernimmt eine Herz-Lungen-Maschine außerhalb des Körpers.

⁴ Gerät, das als apparativer extrakorporaler Kreislauf vorübergehend die Herz- u. Lungenfunktion zu übernehmen vermag.

(Akutphasenmediatoren), die eine umfangreiche Entzündungsreaktion auslösen können, in einzelnen Fällen bis hin zum Organversagen während oder nach der Operation. Bisher gibt es keine Angaben darüber, ob durch eine laufende Modifikation und Anpassung des von der Maschine erzeugten Perfusionsmusters⁵ eine verbesserte Kopplung des Systems Maschine-Mensch erzielt und damit eine „physiologischere“ Perfusion⁶ geschaffen werden könnte [BIRNBAUM ET AL. 1990; RUNGE ET AL. 1992].

Ziel des Forschungsprojekts war die Erarbeitung eines Systems, das durch eine verbesserte Anpassung an den menschlichen Organismus eine physiologischere EKZ ermöglicht. Dadurch sollen sowohl die Risiken, als auch die Nachwirkungen einer Operation am offenen Herzen nachhaltig verringert werden. Deswegen wurde im Rahmen dieses Projektes ein System entwickelt und in Betrieb genommen, das es auf Basis einer konventionellen Herz-Lungen-Maschine ermöglicht, den Puls des Patienten unter EKZ weitestgehend nachzubilden. Anhand des Prototyps (im weiteren Verlauf Pulsator⁷ genannt) wurde dann in Tierversuchen mit Schweinen der eigentlichen medizinischen Fragestellung nachgegangen.

Im Verlauf der ersten Entwicklungsschritte arbeitete sich das Entwicklungsteam anhand von Fachliteratur intensiv in die medizinischen Belange dieser Fragestellung ein. Die Bedeutung einer umfassenden Anforderungsanalyse zu Beginn der Entwicklung war allen Beteiligten klar und wurde anhand von Recherchen und vor allem Interviews mit den Partnern (hier im Sinne von *Kunde* oder Auftraggeber) der medizinischen Disziplin und Entwicklern eines Herstellers für Herz-Lungen-Maschinen durchgeführt. Aufgrund fehlender ähnlicher Vorgängerprojekte bei den beteiligten Institutionen konnte auf keine vorhandenen Informationen zurückgegriffen werden. Zur methodischen Unterstützung kamen vor allem Checkliste [EHRENSPIEL 2003, S. 350] und Hauptmerkmalsliste [PAHL ET AL. 2003, S. 194] zum Einsatz.

Da sich während der EKZ das gesamte systemische Blutvolumen zeitweise außerhalb des Organismus befindet und somit vollständig mit Fremdmaterialien in Kontakt kommt, wurde die Frage der zu verwendenden Werkstoffe als eine besonders wichtige *Anforderung* identifiziert. In den entsprechenden Vorschriften [MPG §7; MPR 93/42/EWG ANHANG I, 7.1] wird hierzu geschrieben, dass „die wechselseitige Verträglichkeit zwischen den eingesetzten Werkstoffen und den Geweben, biologischen Zellen sowie Körperflüssigkeiten, und zwar unter Berücksichtigung der Zweckbestimmung des Produkts“ gegeben sein muss. Dies bedeutet im Einzelnen, dass durch den verwendeten Werkstoff weder der Organismus geschädigt werden darf, noch dürfen vom Organismus Schäden an den medizintechnischen Geräten hervorgerufen werden. Zum Beispiel ist Blut hochgradig korrosiv, was etwa beim Einsatz von Baustahl als Werkstoff zu Korrosion an den Teilen und dadurch wieder zu Schäden am Organismus führen würde.

⁵ Durchblutungsmuster (Kombination aus Pulsfrequenz, Druckverlauf und Flussgeschwindigkeit)

⁶ Durchblutung

⁷ Aggregat zur Erzeugung eines Pulses während der Extrakorporalen Zirkulation

Bei dem letztendlich umgesetzten Konzept des Pulsators handelt es sich um eine Pumpe, bestehend aus einer Zylinder-/Kolbenkombination, die von einem flexibel steuerbaren Antrieb angetrieben wird. Der Zylinder wird von dem kontinuierlichen Grundfluss, der von der Rollerpumpe der Herz-Lungen-Maschine geliefert wird, durchflossen. Diesem kontinuierlichen Grundfluss kann mittels des Kolbens ein in gewissen Grenzen frei wählbarer Puls aufgeprägt werden. Der Puls lässt sich hinsichtlich Frequenz, Druckanstiegsgeschwindigkeit, minimalen und maximalen Druck mittels einer eigens dafür entwickelten Software regeln. Der prinzipielle Aufbau der Anwendung kann Abb. 2-1 entnommen werden.

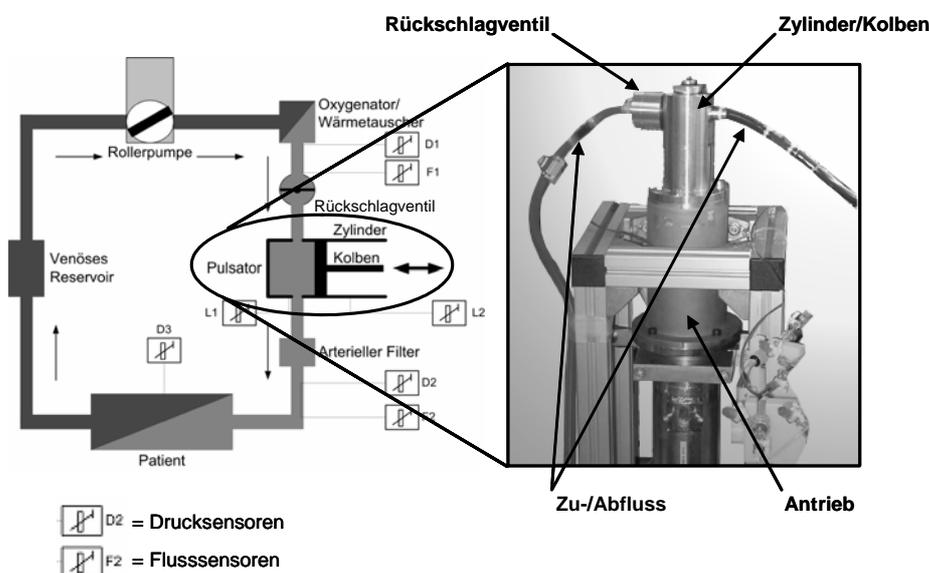


Abb. 2-1: Schema der pulsatilen Perfusion

Im Verlauf der Entwicklung sind unterschiedliche Störungen aufgetreten, die alle ihre Ursache in der *Anforderungskklärung* hatten. Drei davon sollen beispielhaft dargestellt werden.

Ungenügende Biokompatibilität

Der wohl schwerwiegendste Fehler war die Festlegung des falschen Werkstoffs für die blutführenden Teile des Pulsators (Kolben, Zylinder, Ventilgehäuse, div. Anschlüsse). Als Maßgabe für den Werkstoff galt die physiologische Verträglichkeit in Bezug auf den Organismus. Nach Recherchen in medizinischer Fachliteratur, Werkstoffdatenbanken u. v. m. wurde in Absprache mit dem medizinischen Kooperationspartner der Edelstahl 1.4301 [HEINZLER ET AL. 2002] ausgewählt. Diese Anforderungsfestlegung erfolgte etwa sechs Monate nach Beginn der Vorarbeiten. Im Anschluss wurden die notwendigen Teile und Komponenten beschafft beziehungsweise gefertigt, der Prototyp aufgebaut und nach der Programmierung der Steuerungssoftware in Betrieb genommen.

Etwa ein Jahr nach der Fertigstellung des Prototypen, kurz vor Beginn der eigentlichen Tierversuche erkannte man in einem Gespräch mit einem Mitarbeiter der Experimentellen Chirurgie zufällig, dass diese Werkstofffestlegung sehr ungünstig sei, da die mikroskopische Oberfläche des Edelstahls und die im Blut befindlichen Thrombozyten zueinander nicht oberflächenkompatibel seien (ungenügende Biokompatibilität). Dies würde zu einer Zerstörung eben dieser Blutkörperchen führen. In einer Krisensitzung der Projektpartner wurde dieses Problem erörtert. Darauf angesprochen äußerten die beteiligten Mediziner unabhängig voneinander, dass dies korrekt sei, sie aber im Moment der Festlegung der Anforderung und auch danach nicht an diesen Zusammenhang gedacht hätten.

Für den weiteren Projektverlauf wurde dieser Fehler „in Kauf genommen“, da die Versuchsergebnisse durch die Zerstörung der Thrombozyten gar nicht oder nur unwesentlich beeinflusst werden. Hätte es sich bei dieser Entwicklung aber um eine Serienentwicklung gehandelt, wäre das Projekt weit zurückgeworfen worden (der Verfasser schätzt hier eine Verzögerung des Projekts um etwa einem Jahr), da das dargestellte Lösungskonzept nicht ohne weiteres aus einem anderen Werkstoff hätte gefertigt werden können. Außerdem war der Prototyp bereits gefertigt und in Betrieb, so dass zumindest der Beginn der Versuche sich um einige Monate verzögert hätte.

Um mangelhafte Sorgfalt bei der Anforderungskklärung auszuschließen, wurden die Ergebnisse der Literaturrecherche aufgrund dieser neuen Informationen nochmals untersucht, konnten aber kein anderes Ergebnis bringen. Erst eine wesentlich weiter gefasste Recherche zu dem Thema brachte im Umfeld der Implantation von anorganischen Materialien Hinweise auf die ungenügende Biokompatibilität [BECK 2001, S. 4].

In diesem Fall wurde das *Merkmal* der *Anforderung* sehr früh erkannt, die *Ausprägung* aber falsch definiert. Der Fehler wurde erst spät bemerkt.

Unzureichende Funktionalität der Steuerung

Da im Rahmen dieses Projektes vor allem ein Aggregat zum wissenschaftlichen Einsatz in Tierversuchen und nicht zum operativen Einsatz bei Humaneingriffen entwickelt werden sollte, wurden bereits bei der Anforderungskklärung die besonderen Randbedingungen der durchzuführenden Versuche und des Operationsraums für Tierversuche (Tier-OP) einbezogen. Mit den medizinischen Kooperationspartnern wurde deshalb der Ablauf eines solchen Pulsationsversuchs detailliert besprochen, um daraus ableitbare Anforderungen erfassen zu können. Abb. 2-2 stellt den Ablauf im Überblick dar. Um die erhoffte Verbesserung der pulsatilen Perfusion gegenüber der herkömmlichen Perfusion bewerten zu können, waren Referenzversuche geplant, die unter Auslassung der Pulsation nach dem gleichen Muster durchgeführt werden sollten.

Um das nachfolgend geschilderte Problem besser verstehen zu können, muss kurz auf die im Versuchsplan genannte Reperfusion eingegangen werden: Während der EKZ ist der Durchfluss durch die Aorta⁸ unterbrochen. Das Blut wird dann zum Beispiel aus dem rechten Vorhof des Herzens über eine Kanüle einem Reservoir der Herz-Lungen-Maschine zugeführt.

⁸ vom Herzen weggehende Hauptschlagader

Nach der Aufbereitung des Blutes (Gashaushalt, Filterung, Temperatur) wird das Blut mittels einer Rollerpumpe über die Aorta wieder dem Organismus zugeführt (siehe Abb. 1-1).

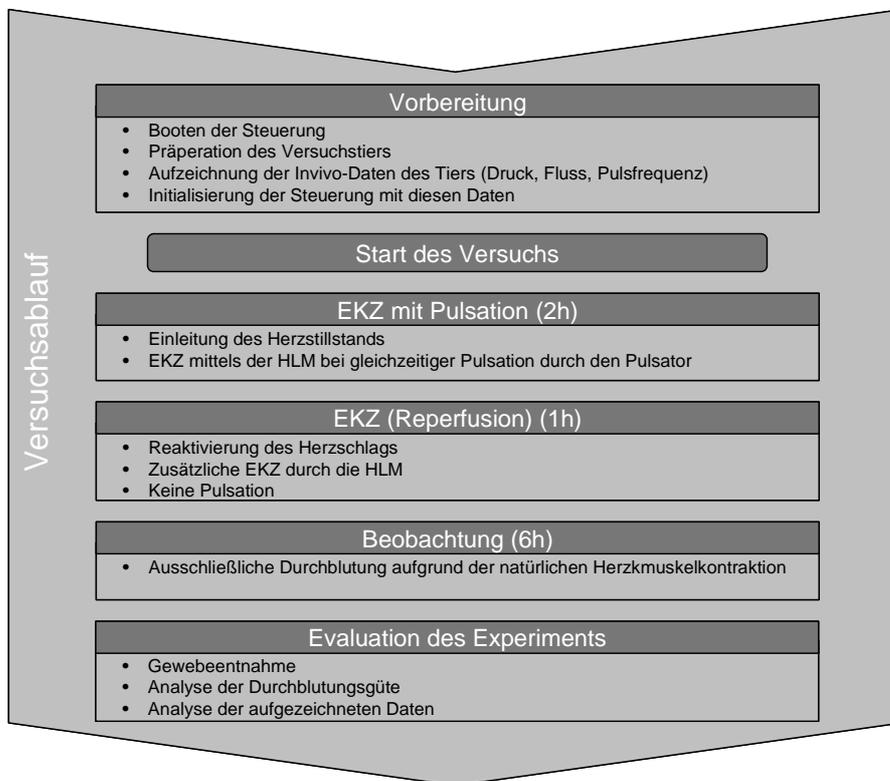


Abb. 2-2: Standardisierter Ablauf eines Pulsationsversuchs

Das Herz wird in dieser Phase nicht durchblutet. Währenddessen wird der Herzschlag mittels einer kardioplegischen Lösung⁹ unterbunden. In der Reperfusionsphase wird zum einen keine weitere kardioplegische Lösung zugeführt, zum anderen wird die Unterbrechung der Aorta wieder aufgehoben, so dass das Herz wieder durchblutet wird und seinen Herzschlag wieder aufnimmt. Um aber dem Kreislauf und dem Herzen die Möglichkeit zur Regeneration und Stabilisierung zu geben, wird der Kreislauf zur Entlastung weiterhin von der Herz-Lungen-Maschine perfundiert [siehe auch WEITKEMPER ET AL. 1999, S. 365 ff].

Nach der Durchführung erster technischer Versuche wurde der erste für die Versuchsreihe gültige Pulsationsversuch durchgeführt. Als die Reperfusionsphase gemäß Versuchsplan eingeleitet werden sollte, wurde der Pulsator außer Betrieb genommen. Hierüber zeigte sich der den Versuch vornehmende Chirurg sehr verwundert: Für eine wirkliche Vergleichbarkeit der beiden Versuchsreihen, hätte der Pulsator seiner Meinung nach weiter in Betrieb bleiben müssen. Es stellte sich aber schnell heraus, dass dies mit der realisierten Anwendung nicht

⁹ Kardioplegische Lösungen, zum Beispiel nach Bretschneider [TROITZSCH ET AL. 1999, S. 181ff], unterbinden die Erzeugung und Weiterleitung eines Signals zur Auslösung der Kontraktion des Herzmuskels.

möglich gewesen wäre. Da in dieser Phase das Herz bereits wieder eigenaktiv schlägt, müsste die Pulsation mit dem Herzschlag synchronisiert werden. Ansonsten bestünde die Gefahr, dass der Pulsator gegen die Herzkontraktion einen Puls erzeugen würde und so das Herz nur weiter geschädigt werden würde. Eine solche Synchronisation ist technisch aber nur sehr schwer und unter großem Aufwand zu realisieren. In den bestehenden Prototypen war eine nachträgliche Integration nicht möglich.

Das Entwicklungsteam versuchte zu analysieren, warum diese Anforderung trotz Einbeziehung des Versuchverlaufs in die Anforderungskklärung und der zahlreichen Gespräche mit den Projektpartnern nicht erfasst worden war. Es konnte rekonstruiert werden, dass die Unterschiede zwischen den Pulsationsversuchen und den Referenzversuchen nicht ausreichend berücksichtigt worden waren. Man war der Meinung, dass „diese Versuche ja nichts Neues wären und man sie wie immer durchführen könne“, nicht bedenkend, dass sie in diesem Fall eben nicht wie immer durchgeführt werden konnten. Eine Gegenüberstellung der Versuchsabläufe, bei der dieser neue Zusammenhang vielleicht aufgefallen wäre, fand nicht statt.

Hier wurde eine Anforderung sehr früh vollständig übersehen. Auch dieser Fehler wurde erst sehr spät bemerkt.

Energieversorgung

Für die Realisierung des Prototypen wurden der Antrieb und alle dazugehörigen Komponenten wie Leistungselektronik, Regler, speicherprogrammierbare Steuerung (SPS) und Software (im Weiteren zusammengefasst unter Steuerung) zugekauft. Die Auslegung des Motors und der übrigen Komponenten fand in Zusammenarbeit mit Mitarbeitern des Antriebsherstellers statt. Aufgrund der zur Verfügung stehenden Energie im Tier-OP (230V AC), wurde mit den Mitarbeitern des Herstellers mehrfach besprochen, dass eine Anpassung der Steuerung an diese Randbedingung notwendig sei. Diese Steuerungen arbeiten üblicherweise mit 360V Drehstrom, der aber im Tier-OP nicht vorhanden ist. Deswegen wurde die Anforderung „Anschluss der Leistungselektronik an einen 230V AC Hausanschluss“ formuliert.

Nach Lieferung der Anlage und Integration in den Prototypen wurde die Anlage für die erste Inbetriebnahme im Labor des Lehrstuhls für Produktentwicklung ans Hausnetz angeschlossen. Nach dem Einschalten der SPS und der Leistungselektronik wurde der Antrieb gestartet, was aber sofort zu einem totalen Stromausfall im Labor führte. Wie sich herausstellte, war der FI-Schutzschalter aktiviert worden, ein Fehler in der Anlage konnte aber nicht gefunden werden. Weitere Einschaltversuche führten zu keinem anderen Ergebnis.

Bei Rücksprache mit dem Hersteller stellte sich heraus, dass bei Betrieb des Motors von der Leistungselektronik Strom zurückgespeist wird. Bei der hier gewählten Anschlussvariante wird dieser auf den Schutzleiter (früher Erde) geleitet, was sofort zu einer Aktivierung des FI-Schutzschalters und somit zur Stromabschaltung führt. Das für diesen Zusammenhang notwendig Fachwissen konnte auf Seiten der Entwickler nicht vorhanden sein, seitens des Herstellers sagte man, dass es gerade aufgrund dieser Problematik unüblich sei, Werkstätten und Laboreinrichtungen mittels Schutzschalter abzusichern, weshalb auch hier nicht extra darauf eingegangen worden ist. Um das Problem zu lösen, musste die Steuerung nochmals

mit einem eigenen Transformator ausgestattet werden. Die Beseitigung dieses Fehlers war zwar möglich und kostete „nur“ wenige hundert Euro, brachte aber eine Projektverzögerung von knapp zwei Monaten mit sich.

Auch in diesem Fall wurde eine Anforderung übersehen. Diese Fehlleistung geschah bereits zu einem relativ frühen Zeitpunkt im Projekt, die Auswirkungen machten sich aber erst spät bemerkbar.

2.1.2 Entwicklung eines Druckers

Im Rahmen des hier beschriebenen Projektes [MEITINGER 2001; PERBANDT 2001] sollten Lösungen für eine neue Generation von Druckern für den professionellen Einsatz entworfen werden. Diese würden neu entwickelte Druckköpfe besitzen und mit einer bisher nicht da gewesenen Präzision und Auflösung farbige Druckerzeugnisse produzieren. Um dies erreichen zu können, waren an die Genauigkeit der Lage der Farbpigmente auf dem Papier sehr hohe Toleranzanforderungen gestellt worden (Abweichung $< 4\mu\text{m}$). Damit dieses Ziel bei gleichzeitig vertretbaren Kosten auch eingehalten werden konnte, sollte ein so genannter „Intelligenter Druckkopf“ entwickelt werden. Die zugrunde liegende Idee ist einfach: Eine Kombination aus Sensorik, Datenverarbeitung und Aktorik erkennt und kompensiert, möglichst während des Druckvorgangs, mechanische Abweichungen, die zu Bildfehlern führen. So können die Genauigkeitsanforderungen an die mechanischen Bauteile und der damit verbundene Kostenaufwand erheblich gesenkt werden. Für die dafür notwendige Aktorik und Sensorik sollten in dem hier beschriebenen Projekt Lösungen systematisch entwickelt und hinsichtlich Kosten und Realisierbarkeit miteinander verglichen werden.

Bei intensiverer Betrachtung der Aufgabenstellung zeigte sich aber bald, dass die Aufgabe mit den bisher vorhandenen Informationen nicht zu lösen sein würde. Das Systemverhalten und der Systemaufbau des Druckers waren vom Auftraggeber noch nicht weit genug durchdrungen worden, um zu erkennen, dass mit den bis dahin bekannten *Anforderungen* hinsichtlich Toleranzen eine Lösungssuche nicht möglich gewesen wäre. Da der im Druckbild messbare Fehler das Ergebnis einer Verkettung von unterschiedlichen Fehlern und Toleranzen im gesamten vorgeschalteten System (Papiertransport, Druckkopf, Führung des Druckkopfes, Antriebe etc.) ist, musste man sich zuerst mit diesen Einflüssen auf das Druckergebnis auseinandersetzen. Wie sich aber herausstellte, war die Entwicklung des eigentlichen Druckers noch gar nicht weit genug vorangeschritten, man befand sich hier noch in der Konzeptphase, um schon konkrete Angaben über möglicher Fehler und Toleranzen machen zu können. Diese Daten wären aber wiederum notwendige gewesen, um Anforderungen an den „Intelligenten Druckkopf“ klären zu können.

Das eigentliche Projektziel konnte also zu diesem Zeitpunkt noch gar nicht erreicht werden, man hatte aber wertvolle Erkenntnisse gewonnen. Deswegen wurde auch der Auftrag umformuliert: Das Projektteam setzte sich jetzt mit der Frage auseinander, welche Fehler beziehungsweise Toleranzen im System denn alleine oder in Kombination zu dem zulässigen beziehungsweise unerwünschten Druckergebnis führen könnten. Hierzu wurde ein abstraktes aber dadurch sehr allgemeines Modell aufgestellt, auf dessen Basis ein Werkzeug zur Toleranzkalkulation entwickelt wurde. Mittels dieses Werkzeugs sollte es dem Auftraggeber

später möglich sein, ausgehend von einem Konzept für den Drucker die zulässigen beziehungsweise auftretenden Toleranzen im System zu berechnen, um daraus Anforderungen an den „Intelligenten Druckkopf“ ableiten zu können.

Das Projektziel konnte in Folge einer sehr ungenauen *Anforderungsklä rung* im Vorfeld und einem ungenügenden Systemverständnis nicht erreicht werden. Es ist zwar die Anforderung an das Gesamtsystem (Abweichung $< 4\mu\text{m}$) bekannt gewesen, die für die Entwicklungsaufgabe notwendigen Anforderungen an das Teilsystem konnten zu diesem frühen Entwicklungszeitpunkt aber noch nicht abgeleitet werden. Dennoch wurden wichtige Erfahrungen gesammelt, aus denen ein für die spätere Klärung dieser Anforderungen bedeutendes Werkzeug entstand.

2.2 Beobachtungen in der Projektarbeit

Über die oben ausgeführten Beispiele hinaus wurden noch weitere Projekte aus unterschiedlichen Branchen untersucht. Unter anderem waren dies Projekte aus der Medizintechnik [BERNHART 2002, BOKODI 2003, ENG 2005, ENGSTLER 2004, LAUER 2003, PECQUET 2002, PETERMANN 2003], Fotolabortechnik [DIEHL 2001, GERL 2001, KÖCK 2002, TILKE 2001, WAGNER 2001], Sport- und Rehabilitationsgerätetechnik [HERBST 2001, HEYNEN 2003, HÜNECKE 2003]. Überall tauchten immer wieder kleinere oder größere Schwierigkeiten bei der *Anforderungsklä rung* auf. Für die spätere Auswertung hinsichtlich der Definition eines verbesserten Vorgehens zur Anforderungsklä rung und der dazu notwendigen *Methode* sind nachfolgend die in diesen Projekten zur Anforderungsklä rung eingesetzten Methoden und Werkzeuge und die festgestellten Fehler aufgeführt.

Überwiegend wurden Checklisten [EHRENSPIEL 2003, S. 350] und Hauptmerkmalslisten [PAHL ET AL. 2003, S. 194] eingesetzt. Bei umfangreicheren Projekten beziehungsweise bei wiederkehrenden Projekten wurden diese Checklisten teilweise modifiziert und an die besonderen Belange des Projektes angepasst. Unterstützend wurden unterschiedliche Formen der Funktionsstrukturen zur Abstraktion des Problems eingesetzt. Bei Neuentwicklungen ist hier vor allem die ablauforientierte [EHRENSPIEL 2003, S. 368 ff] und bei Weiterentwicklungen (zur Optimierung, Anpassung o. ä.) die relationsorientierte Funktionsstruktur [ALTSCHULLER 1984; LINDEMANN 2005; TERNINKO 1998] zu nennen. In allen Fällen wurde eine Anforderungsliste zur Strukturierung und Dokumentation eingesetzt. In unterschiedlicher Häufigkeit sind die nachstehenden Fehler aufgetreten. Zu einem ähnlichen Schluss kommt Ward [WARD ET AL. 2003] in einer von ihm in der Industrie durchgeführten Studie.

- *Anforderungen* wurden nicht erkannt, weshalb diese auch nicht geklä rt wurden und später fehlten.
- Anforderungen wurden zwar erkannt und geklä rt, aber (teilweise trotz Anforderungsliste) im weiteren Projektverlauf vergessen.
- Es wurden vereinzelt Anforderungen erkannt, die eigentlich gar nicht notwendig gewesen wären und somit die Lösungssuche unnötig und irreführend einschränkten.

- Anforderungen wurden falsch (zum Beispiel nicht lösungsneutral, mehrdeutig, oder unvollständig) formuliert. Oftmals wurde gerade bei globaleren Anforderungen, die nicht quantifiziert werden konnten, die *Ausprägung* mit in das *Merkmal* formuliert. Zum Beispiel wurde die Anforderung „gute Sterilisierbarkeit“ formuliert, die Ausprägung wurde dann nicht mehr angegeben, da sie sich ja in Form von „gut“ schon in dem Merkmal verbarg.
- Unspezifische Anforderungen zu Beginn der Entwicklung wurden dem im Verlauf der Entwicklung zunehmenden Kenntnisstand über das zukünftige Produkt nicht angepasst, so dass eine Überprüfung der Zielerreichung nicht oder nur schwer möglich war.
- Die Intensität der Anforderungsklärung variiert sehr stark.
- Anforderungen wurden zwar erkannt, die Formulierung des Merkmals und/oder die Quantifizierung aber fehlerhaft angefertigt.

Die Folgen dieser unterschiedlichen Versäumnisse bei der Anforderungsklärung können sehr verschieden sein. Verzögerungen im Projekt durch zusätzliche *Iterationen* [HUTTERER 2005, S. 86 ff], finanzielle Einbußen und Misserfolg am Markt seien hier nur stellvertretend genannt [siehe auch ALMEFELT ET AL. 2003; WARD ET AL. 2003].

2.3 Auswertung der Beobachtungen

Die vorangehend geschilderten Beobachtungen aus unterschiedlichen Entwicklungsprojekten werfen die Frage auf, warum die eingesetzten Methoden nicht ausreichend funktioniert haben. Um diese Frage mittels einer Gegenüberstellung mit den bekannten Methoden zur *Anforderungsklärung* beantworten zu können, werden die Beobachtungen im Folgenden unter zwei Gesichtspunkten analysiert und abstrahiert:

- Betrachtung der gesamten Anforderungsklärung im Projektverlauf (Makrobetrachtung): Da die beschriebenen Fehler zu unterschiedlichen Zeitpunkten im Entwicklungsprozess sichtbar geworden sind und deren Ursachen ebenso in unterschiedlichen Phasen zu suchen waren, wird der Verlauf der Anforderungsklärung im Prozess untersucht.
- Betrachtung der Klärung einer einzelnen *Anforderung* (Mikrobetrachtung): Der Prozess, der zur Klärung einer einzelnen Anforderung notwendig ist, wird betrachtet und abstrahiert, um die auftretenden Fehler schematisieren zu können.

2.3.1 Makrobetrachtung

Wie bereits erwähnt, traten die geschilderten Probleme zu unterschiedlichen Zeitpunkten im Entwicklungsprozess auf. Um deren Ursachen aufzudecken, wurden die Entwicklungsprojekte hinsichtlich folgender Kriterien qualitativ untersucht:

- Intensität der aktiven, bewussten Anforderungsklä rung im Projektverlauf
- Menge der dokumentierten Anforderungen
- Menge der festgelegten technischen Anforderungen
- Detaillierungsgrad des Produkts

Aus der Auswertung konnten die folgenden Schlüsse gezogen werden, die in Abb. 2-3 zusammengefasst sind:

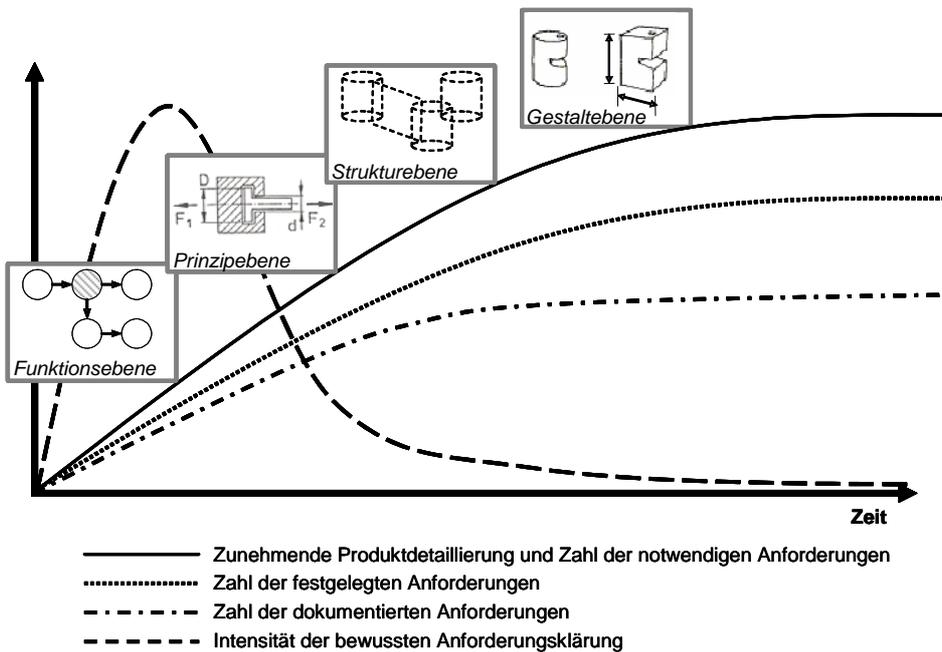


Abb. 2-3: Qualitative Darstellung des Verlaufs der Anforderungsklä rung im Projekt (Makrobetrachtung)

- Anforderungen werden vor allem am Anfang der Entwicklung unter Einsatz unterschiedlicher Methoden geklä rt. Aufgrund des geringen Kenntnisstandes bei Neuentwicklungen handelt es sich dann vor allem um unkonkrete, globale Anforderungen.
- Die Intensität dieser aktiven und bewussten Anforderungsklä rung lässt im weiteren Projektverlauf meist stark nach, während die Detaillierung des Produkts stetig zunimmt.
- Mit der zunehmenden Detaillierung des Produkts wächst auch der Kenntnisstand über das Produkt. Dies führt zu neuen, oftmals lösungsbezogenen Anforderungen beziehungsweise zur Detaillierung oder Änderung vorhandener Anforderungen [siehe GIPOULIS ET AL. 1995, S. 478]. Diese faktisch durchgeführte Anforderungsklä rung findet aber meist unbewusst und ohne Methodeneinsatz statt.

Oft werden die daraus resultierenden Anforderungen gar nicht oder zu mindest nicht in der Anforderungsliste festgehalten [siehe KARCHER ET AL. 1998]. Zum Beispiel wurde bei der Entwicklung eines Chirurgie-Telemanipulators nach der Auswahl des zuzukaufenden Antriebs zwar dieser und der daraus resultierende Flansch in Konzeptzeichnungen festgehalten, der entsprechende Eintrag in der Anforderungsliste wurde aber nicht vorgenommen [siehe auch ENG 2005]. Der Bearbeiter der untergeordneten Baugruppe verließ sich anfangs auf die Anforderungsliste und nahm an, dass er den Flansch frei gestalten könnte.

Um diesen Sachverhalt unter kontrollierten Bedingungen zu überprüfen, wurde vom Verfasser die Arbeit einer Forschergruppe des Lehrstuhls für Produktentwicklung beobachtet. Ziel dieser Forschergruppe, der der Verfasser angehörte, war, anhand der Entwicklung eines Möbelscharniers unterschiedliche Entwicklungssituationen zu simulieren, um vorhandene Methoden anwenden und/oder üben und neue Ansätze untersuchen zu können.

Für die vorliegende Arbeit wurde die Anforderungsklä rung beobachtet. Hierzu wurden die Teilnehmer in zwei Gruppen unterteilt, die mittels unterschiedlicher Methoden [EHRENSPIEL 2003; CONRAD ET AL. 1977] Anforderungen an das Scharnier klären sollten. Die Ergebnisse dieser beiden Gruppen wurden anschließend zusammengeführt und strukturiert. Die so entstandene Anforderungsliste stellte dann die Basis für die weitere Entwicklung dar. Die Anforderungsliste enthielt eine Vielzahl genereller Anforderungen, die gar nicht oder nur sehr allgemein quantifiziert waren. Trotzdem wurde auch hier im weiteren Verlauf der Entwicklung nur bedingt auf die Anforderungsliste zurückgegriffen beziehungsweise die vorhandenen Anforderungen praktisch nicht erweitert beziehungsweise konkretisiert.

Beispielsweise wurde die Justierbarkeit der Tür (Höhe und Neigung um 2 Achsen) als wichtige Funktion und Anforderung erkannt und in die Anforderungsliste übernommen. Bei der Lösungssuche und -auswahl für das Scharnier wurde diese Anforderung aber nicht weiterverarbeitet. Je nach Lösung kann die geforderte Justagemöglichkeit der Tür nur mittels des Scharniers vorgenommen werden. In einem solchen Fall hätte die Anforderung an die Tür vollständig oder teilweise auf das Scharnier übertragen werden müssen und zum Beispiel zu den Anforderungen „Justagerichtungen beziehungsweise -achsen des Scharniers“ und „Justagebereich des Scharniers“ und deren individuellen Quantifizierungen detailliert werden müssen.

Im Fall der eingangs geschilderten Entwicklung eines Druckers hatte man zwar für Teilbereiche der Entwicklung detaillierte Konzepte entwickelt, die Anforderungsklä rung aber nicht konsequent weitergeführt, so dass die Definition notwendiger Schnittstellen zur Entwicklung des „Intelligenten Druckkopfs“ nicht möglich gewesen ist.

Zusammenfassend kann festgehalten werden, dass Ursachen für Probleme mit Anforderungen nicht nur in den frühen Phasen der Produktentwicklung zu suchen sind, sondern auch in den späteren Phasen, in denen die Intensität der bewussten Anforderungsklä rung stark abnimmt, obwohl lösungsbezogene Anforderungen sowie Schwachstellen erst im Laufe des Entwicklungsprozesses auftreten. Diese These, die besonders für Neuentwicklungen gültig ist, wird auch durch eine Untersuchung von Giapoulis gestützt [GIAPOULIS ET AL. 1995, S. 478].

2.3.2 Mikrobetrachtung

Um den Gründen für eine fehlerhafte Anforderungsklä rung näher zu kommen, soll der Versuch der Schematisierung des Prozesses der Anforderungsklä rung vorgenommen werden. Dazu wird die Klä rung einer **einzelnen** Anforderung betrachtet. Um die Frage des Ablaufs einer Anforderungsklä rung zu beantworten, beobachtete der Verfasser Studenten und sich selbst bei der Anforderungsklä rung und verglich die Beobachtungen mit Beschreibungen in der Literatur [zum Beispiel BREIING & FLEMMING 1993; CONRAD ET AL. 1977; CROSS 1996; PAHL ET AL. 2003].

Zusammenfassend lässt sich festhalten, dass die Klä rung einer einzelnen Anforderung ein mehrstufiger Prozess ist [siehe auch CROSS 1996, S. 77 ff; PAHL ET AL. 2003, S. 188 ff]. Abb. 2-4 stellt die Klä rung einer einzelnen Anforderung schematisch dar.

- Eventuell mit Hilfe einer Checkliste wird eine mögliche Anforderung oder ein *Anforderungsfeld* identifiziert.
- Man stellt sich die Frage, ob diese im Falle des zu entwickelnden Produkts notwendig ist. Sollte man *erkennen*, dass eine solche *Anforderung* nicht notwendig ist, fallen die weiteren Arbeitsschritte natürlich weg.
- Das *Merkmal* der *Anforderung* wird lösungsneutral *formuliert* und in die Anforderungsliste übernommen.
- Die bis zu diesem Zeitpunkt vorhandenen Informationen über die Entwicklungsaufgabe, dies können auch erste Lösungen etc. sein, werden hinsichtlich der betrachteten Anforderung analysiert.
- Man stellt sich die Frage, ob die *Ausprägung* der Anforderung mit den vorhandenen Informationen zu *definieren* ist oder nicht. Ist dies nicht der Fall, muss die fehlende Information eingeholt werden oder das entstehende Produkt zu einem späteren Zeitpunkt, wenn mehr und detailliertere Informationen vorliegen, nochmals hinsichtlich dieser Anforderung analysiert werden.
- Sind genügend Informationen vorhanden, wird die *Ausprägung* der Anforderung *definiert* und in die Anforderungsliste übernommen.

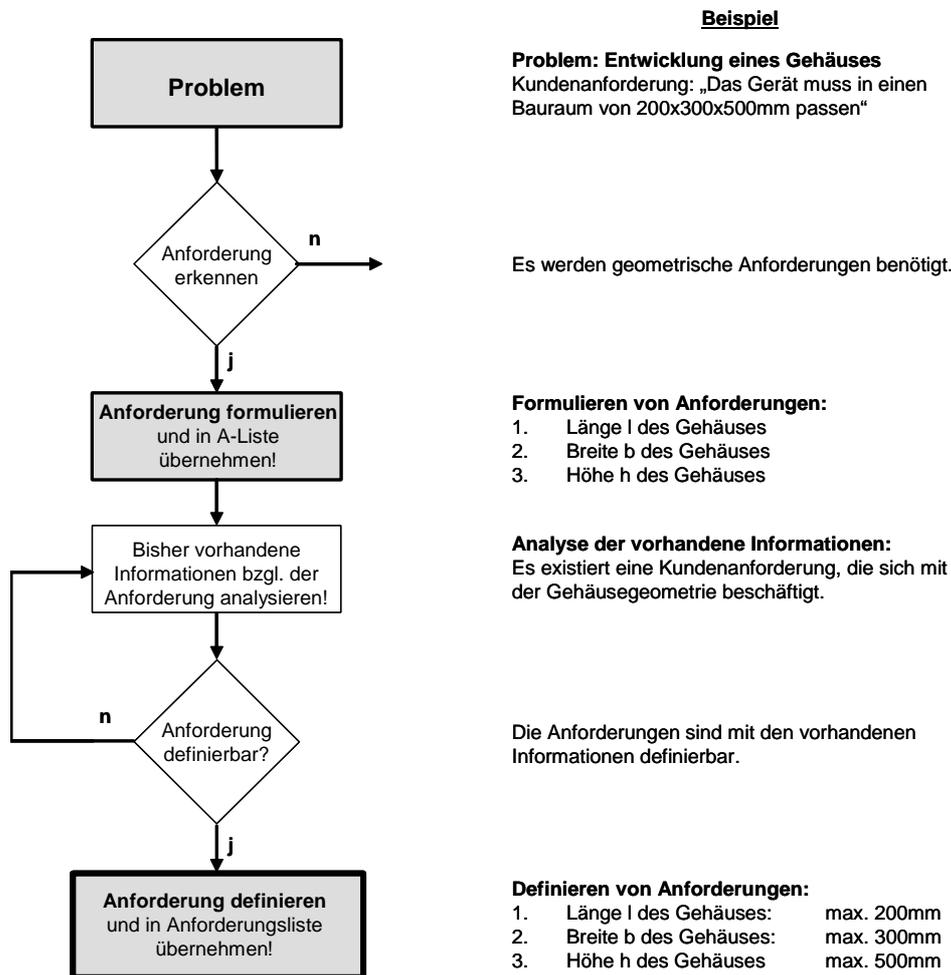
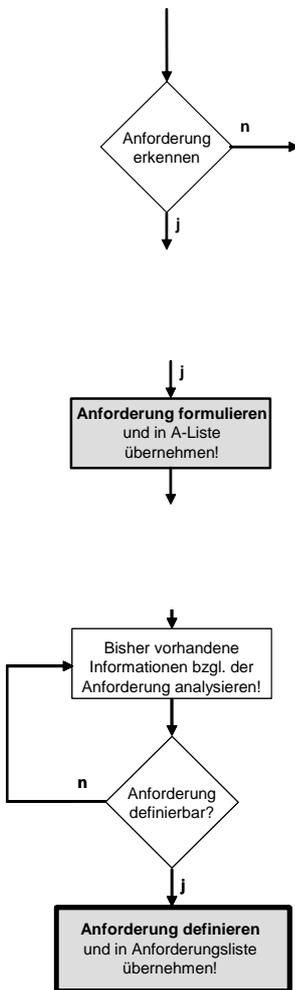


Abb. 2-4: Mehrstufiger Ablauf einer Anforderungsklä rung (Mikrobetrachtung)¹⁰

Innerhalb dieses mehrstufigen Prozesses sind nun natürlich unterschiedliche Fehler denkbar, die nachfolgend näher betrachtet werden sollen [siehe auch SCHWANKL 2002, S. 41].

¹⁰ Diese stufenweise Vervollständigung und Konkretisierung einer einzelnen Anforderung und global gesehen einer ganzen Anforderungsliste kann mit der zum Beispiel von Bichlmaier [BICHLMAIER 2000, S. 97 ff] vorgestellten Eigenschaftsliste verglichen werden: Hier werden schrittweise Eigenschaften gesammelt, durch ihre Soll- und Istausprägungen konkretisiert und mit einem Status (erfüllt/nicht erfüllt) gekennzeichnet.



- Eine notwendige Anforderung wird nicht als solche erkannt und die Anforderungsklä rung wird abgebrochen.
- Eine nicht notwendige Anforderung wird als notwendig erkannt und die Anforderungsklä rung nicht abgebrochen.
- Das *Merkmal* einer **Anforderung** wird falsch, missverständlich etc. oder nicht lösungsneutral **formuliert**.
- Die Analyse der vorhandenen Informationen ist unzureichend, so dass die **Ausprägung** einer Anforderung falsch **definiert** wird.

Zusammengefasst lassen sich die beschriebenen Fehler in einem Baumdiagramm wie in Abb. 2-5 darstellen. Die in Kapitel 2.1 geschilderten Probleme bei der Anforderungsklä rung lassen sich in dieses Schema einordnen: Zum Beispiel wurde im Falle des Pulsators die Anforderung „physiologisch verträglicher Werkstoff“ richtig erkannt und formuliert, die Ausprägung wurde aber falsch definiert.

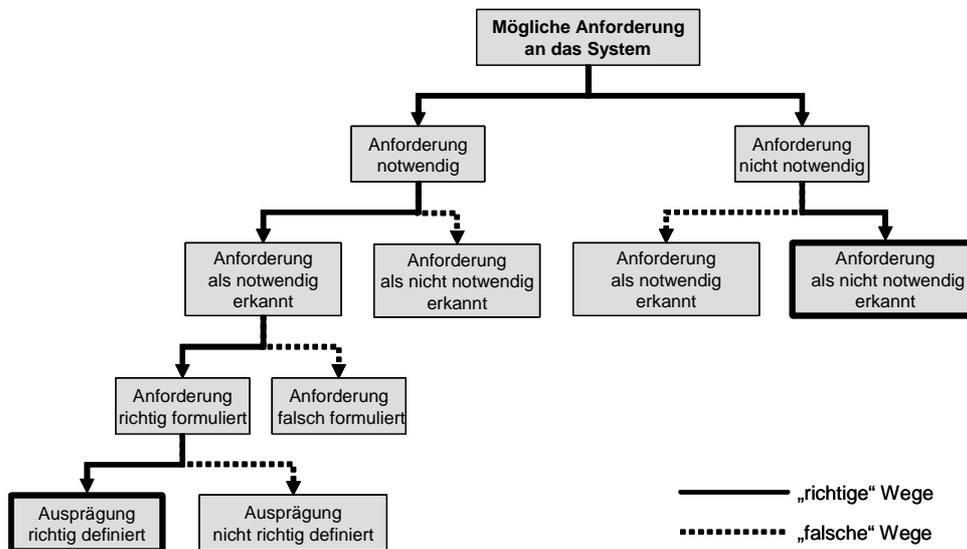


Abb. 2-5: Mögliche Fehler bei der Klärung *einer* Anforderung

Mit der dargestellten Sichtweise des Ablaufs der Klärung einer Anforderung will der Verfasser nicht den Anspruch der Allgemeingültigkeit erheben. Andere Sichtweisen sind denkbar und genauso richtig. Diese Darstellung ist das Ergebnis der Erfahrungen und Beobachtungen des Autors und wird als Grundlage für die Ausführungen im weiteren Verlauf dieser Arbeit herangezogen werden. In Verbindung mit der Betrachtung des Verlaufs der Anforderungsklä rung im Projekt steht ein Hilfsmittel zur Verfügung mit dem später gezielt nach Ursachen für die herausgearbeiteten Fehler bei der Anforderungsklä rung gesucht werden kann.

2.4 Zusammenfassung

Anhand unterschiedlicher Entwicklungsbeispiele konnte verdeutlicht werden, dass trotz des Einsatzes von unterstützenden Methoden Probleme bei der Anforderungsklä rung auftreten. Aus diesen Beispielen und der Erfahrungen aus weiteren interdisziplinären Entwicklungsprojekten wurden grundsätzliche Fehler bei der Anforderungsklä rung herausgearbeitet und schematisiert. Diese Schematisierung der bei der Anforderungsklä rung auftretenden Fehler bildet zusammen mit den im nächsten Kapitel dargestellten bekannten Methoden zur Anforderungsklä rung die Grundlage für die gezielte Beantwortung der Frage nach den Ursachen und dient gleichzeitig als Ausgangspunkt für die Verbesserung des methodischen Instrumentariums.

3 Anforderungskklärung in der Entwicklungsmethodik – eine Bestandsaufnahme

Um ergründen zu können, warum die angesprochenen Probleme bei der Anforderungskklärung auftreten, wird in diesem Kapitel der Stand der Technik hinsichtlich der Anforderungskklärung analysiert. Es wird untersucht, wie die Anforderungskklärung in den Augen verschiedener Autoren durchgeführt werden sollte, um möglichst effizient und effektiv zu sein. Im ersten Abschnitt dieses Teilkapitels finden sich dazu Vorgehensweisen, die in einen größeren Kontext, meist einer Betrachtung und Beschreibung des gesamten Produktentwicklungsprozesses, eingebettet sind. Der Zweite Teil beschreibt davon weitgehend unabhängige Ansätze, die sich ausschließlich mit der Anforderungskklärung beschäftigen. Hier finden sich auch unterschiedliche rechnergestützte Ansätze. Des Weiteren wird eine Strömung aus der Softwareentwicklung, das so genannte Requirements Engineering, vorgestellt. Dies beschäftigt sich mit der Ermittlung von Anforderungen vor allem an Individualsoftware und der Erstellung von daraus abgeleiteten Spezifikationen. Die Ergebnisse dieser Untersuchung werden abschließend zusammengefasst und in Verbindung mit den Ergebnissen aus Kapitel 2 analysiert, um daraus Ursachen für die Problem bei der Anforderungskklärung abzuleiten und Handlungsschwerpunkte zu definieren.

3.1 Anforderungskklärung in umfassenden Modellen der Entwicklungsmethodik

3.1.1 Klärung des Konstruktionsauftrags nach Breiing & Flemming

Breiing & Flemming [BREIING & FLEMMING 1993] beschreiben in „Theorie und Methoden des Konstruierens“ folgende Untergliederung des Konstruktionsprozesses:

- Entstehung eines Konstruktionsauftrages
- Eingang des Konstruktionsauftrages
- Klärung des Konstruktionsauftrages
- Entwurfsphase
- Ausarbeitungsphase

Die Autoren vertreten keine ganzheitliche Sicht auf die Entwicklung eines Produkts, sondern legen ihre Schwerpunkte auf die lösungsschaffenden und detaillierenden Phasen. Deswegen haben wohl auch die frühen Phasen, in denen das Produkt zum Beispiel geplant wird, nur sehr geringes Gewicht und werden von den Autoren auch nicht zum eigentlichen Konstruktionsprozess gezählt. Dieser beginnt in ihren Augen mit dem Eingang des Konstruktionsauftrages.

Der Konstruktionsauftrag enthält eine fertige Anforderungsliste [BREIING & FLEMMING 1993, S. 9], der hier eine große Bedeutung beigemessen wird [BREIING & FLEMMING 1993, S. 14]. Trotzdem wird aber nicht näher auf die Entstehung beziehungsweise Erstellung dieser Liste eingegangen. Nach Breiing & Flemming entsteht die Anforderungsliste beim Auftraggeber und wird nicht im Dialog zwischen Auftragnehmer und -geber entwickelt. Dennoch wird eingeräumt, dass die Anforderungsliste ein lebendes Dokument ist und „von Fall zu Fall an den Entwicklungsgegenstand angepasst werden muss“ [BREIING & FLEMMING 1993, S. 10].

Die sich daran anschließende Klärung des Konstruktionsauftrags wird in die Teilphasen

- Aufgabenstellung analysieren und überprüfen,
- Anforderungsliste bereinigen oder erstellen,
- Kostenvorgaben überprüfen,
- Terminvorgaben überprüfen und
- Konstruktionsablauf planen untergliedert [BREIING & FLEMMING 1993, S. 14].

Von besonderer Bedeutung sind für die Autoren dabei die ersten zwei Teilphasen:

Beim **Analysieren und Überprüfen** der Aufgabenstellung wird der Konstrukteur aufgefordert zum einen durch Rücksprache mit dem *Kunden*, zum anderen durch wissenschaftliche Einzeluntersuchungen sowie Recherchen das Ziel und die Realisierbarkeit der Aufgabe zu überprüfen und die Informationen zu vertiefen. Als besonderes Hilfsmittel wird hierzu die Abstraktion empfohlen.

Beim **Bereinigen oder Erstellen der Anforderungsliste** wird auf mögliche Fehler in Anforderungslisten hingewiesen. Dies können vor allem vergessene, nicht einwandfrei formulierte (zum Beispiel wenn der Auftraggeber dazu fachlich nicht in der Lage ist) oder unvollständige *Anforderungen* sein. Hier wird dann sehr detailliert auf unterschiedliche Klassen von Anforderungen und verschiedene Strukturierungsmöglichkeiten eingegangen.

Es gibt so genannte *explizite* und *implizite Anforderungen*. Explizite Anforderungen sind solche, die ausdrücklich an ein zu entwickelndes technisches System gestellt werden, wogegen die impliziten Anforderungen „aus den naturwissenschaftlichen Gesetzen und ihren Randbedingungen herleitbar sind und nicht gesondert genannt werden“ [BREIING & FLEMMING 1993, S. 14]. Sie weichen damit deutlich von der in der vorliegenden Abhandlung verwendeten, weiter gefassten Definition für implizite Anforderungen ab.

Die von Breiing & Flemming vorgestellten Möglichkeiten der Strukturierung von Anforderungen orientieren sich zum einen an der Bedeutung von Anforderungen (Wunsch/Forderung), wobei in einer freigegebenen Anforderungsliste keine Wünsche mehr vorkommen dürfen [BREIING & FLEMMING 1993, S. 15]. Zum anderen schlagen sie vor, die Anforderungsliste anhand ihrer inhaltlichen Merkmale (zum Beispiel technisch, wirtschaftlich etc.) zu strukturieren. Letztere Strukturierungsmöglichkeit wird intensiv diskutiert und abschließend eine nach Anforderungsfamilien geordnete Musteranforderungsliste entwickelt. Diese entspricht in ihrer Art einer Check- oder Hauptmerkmalsliste [PAHL ET AL. 2003],

deren Einsatz auch bei der Erstellung von Anforderungslisten empfohlen wird, wobei die Autoren eine Allgemeingültigkeit für alle Produktbereiche verneinen und eine entsprechende Anpassung empfehlen.

Abschließend wird aufgezeigt, dass zwischen Anforderungen meist eine Vernetzung unterschiedlicher Intensität existiert, was ein Bereinigen der Anforderungsliste mittels der Überprüfung der möglichen Anforderungsrelationen „unabhängig“, „unterstützend“, „gegenläufig“, und „widersprüchlich“ notwendig macht. Zur Überprüfung der Anforderungsrelationen schlagen sie den Einsatz einer Relationsprüfmatrix vor. Hierbei handelt es sich um eine Einflussmatrix, in der die Anforderungen einander gegenübergestellt werden und auf die oben genannten Relationen hin überprüft werden. Alternativen zur Lösung der daraus möglicherweise resultierenden Konflikte werden nicht angeboten [BREIING & FLEMMING 1993, S. 24 f].

3.1.2 “Setting Requirements” nach Cross

Cross [CROSS 1996] beschreibt einen wesentlich detaillierteren Entwicklungsprozess, den er in die nachfolgenden Phasen unterteilt, deren ersten vier sich mit der *Anforderungsklä rung* beschäftigen.

- Klä rung der Zielvorstellungen (clarifying objectives)
- Ermitteln der Funktionen (establishing functions)
- Festsetzen der *Anforderungen* (setting requirements)
- Ermittlung der Eigenschaften (determining characteristics)
- Entwicklung von Alternativen (generating alternatives)
- Beurteilung von Alternativen (evaluating alternatives)
- Verbessern von Details (improving details)

Unter der **Klä rung der Zielvorstellung** [CROSS 1996, S. 49 ff] versteht Cross die erste Sammlung und Festlegung von allgemeinen Zielen, die das zu entwickelnde Produkt erfüllen muss. Auch wenn sich diese Ziele im weiteren Verlauf der Entwicklung ändern können, ist es wichtig, von Beginn an im Entwicklungsteam eine gemeinsame Vorstellung von der zu erfüllenden Aufgabe zu haben. Hierzu wird ein Zielbaum (objective tree [CROSS 1996, S. 50]) als unterstützende *Methode* vorgeschlagen. Dabei wird davon ausgegangen, dass zu Beginn der Entwicklung sehr vage formulierte Ziele, wie zum Beispiel „das Produkt muss sicher sein“ vorliegen. Um dies zu spezifizieren soll das Ziel mittels W-Fragen detaillierter ergründet werden. Da sich die hierbei ermittelten Ziele und Teilziele auf unterschiedlichen Ebenen befinden, werden diese in einem Baumdiagramm hierarchisch angeordnet. Die schrittweise Erstellung eines solchen Zielbaums, und die dafür stattfindenden Gespräche mit dem *Kunden* helfen das Problemverständnis zu schä rfen und zu vertiefen und so im Entwicklungsteam und mit dem Kunden einen Konsens zu erzielen. Diese Darstellung kann aber immer nur eine temporäre Darstellung des Problems sein, die sich mit dem Entwicklungsprozess verändern wird. Als eigentliches Ziel der Methode nennt Cross nicht

den entstehenden Zielbaum, sondern die während dessen Erstellung stattfindenden Diskussionen und gedankliche Auseinandersetzung mit dem Problem [CROSS 1996, S. 53]. Denn gerade dadurch wird das Problemverständnis erhöht.

Um bei einer Entwicklung funktional auf dem richtigen Detaillierungsgrad anzusetzen beziehungsweise durch die Variation dieses Detaillierungsgrades Anstoß für radikalere, innovative Lösungen zu erhalten, empfiehlt Cross die **Ermittlung der Funktionen** [CROSS 1996, S. 65 ff] mittels einer Funktionsanalyse. Cross empfiehlt die Definition einer oder mehrerer Black-Boxen mit deren Ein- und Ausgängen in Verbindung mit einer Systemgrenzenverschiebung. Davon ausgehend soll dann das System weiter untergliedert werden, um so letztendlich eine Funktionsstruktur zu entwickeln. Cross betont in diesem Zusammenhang besonders, dass in dieser Phase aber auch in allen anderen Phasen, die sich mit Anforderungen auseinandersetzen, immer die Frage „Was?“ das Produkt leisten soll und nicht „Wie?“ es das Produkt leisten soll im Vordergrund stehen muss.

Im Gegensatz zu den bisher ermittelten Funktionen und Zielen, die nur festlegen, was ein Produkt prinzipiell können muss, werden beim **Festsetzen der Anforderungen** [CROSS 1996, S. 77 ff] genaue, quantifizierte Grenzen gesetzt. Um eine Liste der Anforderungen erzeugen zu können, schlägt Cross vor, ausgehend von den Zielen und Funktionen das Produkt beschreibende Attribute (Merkmale) herzuleiten. Aber auch hier gilt wieder der Grundsatz, dass die Anforderungen die benötigte Performance und nicht das benötigte Produkt beschreiben sollen. Anschließend werden die Attribute mit Werten (*Ausprägungen*) versehen. Damit die Werte nicht einfach von vorhandenen Produkten übernommen oder geschätzt werden, sind sorgfältige Recherchen notwendig. Cross räumt ein, dass es oftmals notwendig sein kann, bei den Anforderungen zwischen Wünschen und Forderungen zu unterscheiden.

Hinter der **Ermittlung der Eigenschaften** [CROSS 1996, S. 91 ff] verbirgt sich nichts anderes als die Anwendung des Quality Function Deployment (QFD) beziehungsweise eines Teilbereichs daraus. Mittels dieser Methode sollen Kundenanforderungen in *technische Anforderungen* (engineering requirements) übersetzt werden. Zur Sammlung der dafür notwendigen Kundenanforderungen und deren Bedeutung werden nur sehr allgemein Methoden des Marketings angesprochen. Diesen Anforderungen werden technischen Merkmale gegenübergestellt. Cross spricht hier von „engineering characteristics“ [CROSS 1996, S. 95], die vom Ingenieur direkt beeinflusst werden können. Die Trennung zwischen Kundenanforderungen, den bereits ermittelten Attributen und den technischen Anforderungen wird aber nicht klar. Anhand dieser Gegenüberstellung soll dann herausgearbeitet werden, welche technischen Anforderungen welche Kundenanforderung in welchem Maße beeinflussen.

Cross sieht keine feste Verknüpfung der beschriebenen Phasen zur Anforderungsklä rung, geht aber auf deren Vernetzung nicht näher ein. An die letzte Phase der Anforderungsklä rung schließt Cross die Suche nach Alternativen an.

3.1.3 Aufgabenklärung und -strukturierung nach Ehrlenspiel

Ehrlenspiel [EHRENSPIEL 2003] stellt den Entwicklungsprozess in seinem Vorgehenszyklus als eine Kombination aus mehreren selbstähnlichen TOTE-Schemata dar [EHRENSPIEL 2003, S. 82]. Dieser Vorgehenszyklus beschreibt, ausgehend von einer Aufgabe beziehungsweise einem Problem, eine Abfolge von drei Arbeitsschritten, die zu einer Lösung führen. Durch den grafischen Aufbau des Vorgehenszyklus wird die von Ehrlenspiel geforderte Erzeugung der Lösungsvielfalt innerhalb der ersten zwei Arbeitsschritte und die Einschränkung der Lösungsvielfalt im dritten Arbeitsschritt hin zu einer Lösung gut dargestellt. Des Weiteren zeigt es die zulässigen, oftmals notwendigen beziehungsweise sinnvollen *Iteration* und Rekursionen zwischen den einzelnen Arbeitsschritten.

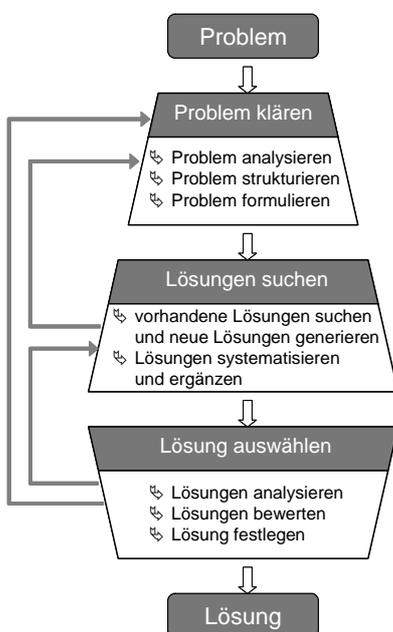


Abb. 3-1: Vorgehenszyklus nach Ehrlenspiel [EHRENSPIEL 2003, S. 83]

Ein Problem besteht aus der Summe aller *Anforderungen*, wobei unter Anforderung die knappe, präzise Formulierung eines gewünschten Sachverhalts in der Sprache des Konstrukteurs verstanden werden soll [EHRENSPIEL 2003, S. 345]. Die Problemklärung nimmt deswegen in diesem Vorgehensmodell als eigener Arbeitsschritt eine wichtige Position ein und wird in die drei Teilschritte Problem analysieren, Problem strukturieren und Problem formulieren unterteilt.

Das grundsätzliche Dilemma der Problemklärung sieht Ehrlenspiel in der Tatsache, dass dem *Kunden* meist nicht klar ist, welche Informationen die Konstruktion im Detail benötigt, und der Konstruktion oftmals nicht klar ist, auf was es dem Kunden wirklich ankommt. Da in den Anforderungen aber der Kern des zukünftigen Produkts liegt, muss dieses Schnittstellenproblem effektiv gelöst werden. Zu Projektbeginn ist es deswegen unerlässlich,

eine möglichst gute Kommunikation mit dem Kunden aufzubauen und mittels intensiver Gespräche, unterstützt durch Check- und Fragelisten, Vergleiche zu früheren Erfahrungen und zur Konkurrenz das **Problem zu analysieren** und Anforderungen zu ermitteln. Ehrlenspiel empfiehlt dringend, dem Entstehen von Kommunikationsmauern zwischen den unterschiedlichen Lieferanten von Anforderungen entgegenzuwirken, da man sonst Gefahr läuft, Anforderungen zu vergessen, zu verfälschen oder falsch zu priorisieren. Wichtigster Lieferant für Anforderungen ist der Endkunde als Nutzer des späteren Produkts. Zusätzlich sind aber auch so genannte „Zwischenkunden“ zu berücksichtigen (Händler, Service, Monteure etc.). Auch diese können wichtigen Input für die später zu erstellende Anforderungsliste liefern.

Im Einzelnen werden nachfolgende Hilfsmittel für das Ermitteln von Anforderungen vorgeschlagen [EHRENSPIEL 2003, S. 353 ff]:

- Um das Vergessen von Anforderungen zu vermeiden, sollte man Checklisten und Leitlinien einsetzen. Vor allem bei neuen Produkten und ungewohnten Situationen können aber Lücken in den Anforderungen nicht mit Sicherheit ausgeschlossen werden.
- Zusammenstellung von interdisziplinären Teams (Entwickler, Service, Monteure etc.).
- Fragelisten
- Ausgabe von Fragebögen an den Endkunden
- Gespräche mit Endkunden, Nutzern etc.
- Systemabgrenzung mittels Black Box: Bei konzeptionellen Konstruktionsproblemen sind die Funktionsanforderungen am wichtigsten. Um diese zu ermitteln, benutzt man die Black-Box-Darstellung. Sie hilft, die Formulierung des Kernproblems zu erleichtern.
- Problemanalyse durch Systemgrenzenverschiebung: Dies kann dann hilfreich sein, wenn es darum geht, ein Problem klarer zu erkennen oder einen ins Stocken geratenen Lösungsprozess wieder zu aktivieren.
- Aufgabenanalyse durch Abstraktion: Je enger und konkreter die Problemstellung und letztlich sämtliche Forderungen formuliert sind, desto kleiner wird die mögliche Lösungsvielfalt. Abstraktion kehrt diesen Prozess um und ermöglicht mehrere Lösungen. Neben der bereits genannten Systemgrenzenverschiebung dient auch die oft damit verbundene Abstraktion zur Klärung des Kernproblems und zur Festlegung des Suchfeldes.

Die so analysierte **Aufgabe** soll anschließend **strukturiert** werden, um die Bearbeitbarkeit zu verbessern. Eine Strukturierung der Aufgabe kann natürlich auch zu einer Strukturierung der Anforderungen führen, weshalb an dieser Stelle auch darauf eingegangen werden soll. Ehrlenspiel stellt dazu unterschiedliche Vorgehensweisen zur Verfügung:

- Gliederung nach der Wichtigkeit von Anforderungen (Wünsche/Forderungen): Ehrlenspiel geht detailliert auf die unterschiedlichen Ausprägungsformen von Anforderungen ein (Fest-, Intervall-, Mindest-, Maximalforderungen und Wünsche).
- Strukturierung nach organisatorischen Merkmalen (Bearbeiter, Entwicklungsphasen etc.).
- Strukturierung nach Funktionen mittels Funktionsstrukturen.
- Strukturierung nach Modulen (zum Beispiel Baugruppen).
- Strukturierung nach inhaltlichen Klassen oder Produktlebensphasen.

Ehrlenspiel geht kurz darauf ein, dass es unterschiedliche Vernetzungen zwischen Anforderungen geben kann (Zielkonflikt, Zielunabhängigkeit, Zielunterstützung) [EHRENSPIEL 2003, S. 350] nennt aber wenig Möglichkeiten, wie diese zu ermitteln und vor allem zu handhaben sind. Er verweist hier vor allem auf die Arbeit von Eiletz [EILETZ 1999], der sich intensiv mit der Behandlung von Zielkonflikten auseinandergesetzt hat.

Alle gesammelten Informationen lässt Ehrlenspiel in eine Anforderungsliste einfließen, die er als eines der wichtigsten Dokumente beim methodischen Konstruieren bezeichnet, das gelebt und immer wieder aktualisiert werden muss. Methodische Unterstützung speziell für die Pflege der Anforderungsliste wird nicht gegeben.

Ehrlenspiel betont, dass die vorgestellten Methoden nicht zwangsläufig zum Ziel führen, vielmehr werden dem Anwender Anregungen gegeben, mit welchen Fragestellungen und Überlegungen Klarheit in eine Aufgabe zu bringen ist [EHRENSPIEL 2003, S. 345].

3.1.4 Methoden zur Anforderungsklä rung nach Lindemann

Lindemann [LINDEMANN 2005] geht mit seinem aus dem Vorgehenszyklus nach Ehrlenspiel hervorgegangenen Münchener Vorgehensmodell (MVM) neue Wege.

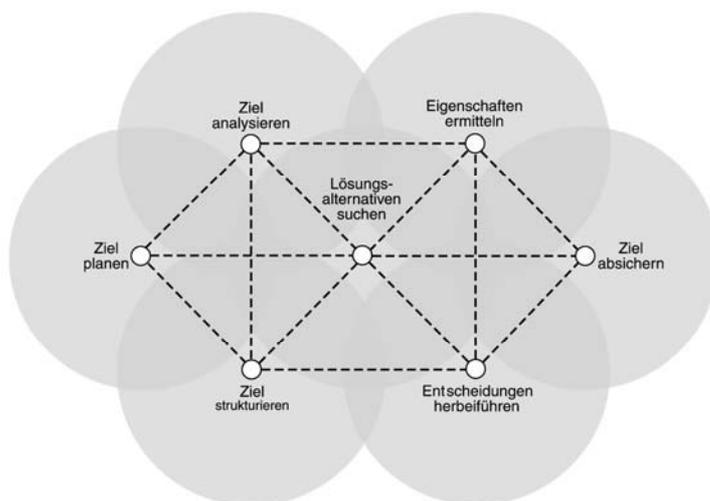


Abb. 3-2: Münchener Vorgehensmodell [LINDEMANN 2005, S. 40]

Das MVM unterteilt den Entwicklungsprozess in sieben gleichwertig nebeneinander stehende Elemente, die nicht wie bisher üblich in einer sequentiellen Abfolge stehen, sondern in Form eines Netzwerks dargestellt sind. Es wird ein Standardweg durch dieses Modell angeboten, das Modell will aber noch viel mehr leisten, um damit realen Prozessen mit ihren sprunghaften Verläufen näher zu kommen. Das Modell vermittelt, dass es, wenn notwendig, möglich ist, nicht wie bisher zu einer „früheren Phase zurückzukehren“, sondern ein, eventuell auf anderer Konkretisierungsebene, schon einmal bearbeitetes Element nochmals „aufzusuchen“, es zu wiederholen, und es dabei zu konkretisieren. Das bedeutet, dass es zum Beispiel zulässig wäre, von dem Element „Eigenschaften ermitteln“ wieder das Element „Ziel analysieren“ aufzusuchen und von dort wieder zu ersterem zurückzugehen ohne dabei nochmals die Elemente „Ziel strukturieren“ und „Lösungsalternativen suchen“ zu durchlaufen.

Innerhalb des MVM beschreiben vor allem die Elemente „Ziel analysieren“ und „Ziel strukturieren“ die Maßnahmen zur *Anforderungsklä rung*. Lindemann untergliedert die beiden Elemente nochmals in einzelne Teilelemente. Hierzu formuliert er Fragen, die dann in Form von Empfehlungen bezüglich des Vorgehens und möglicher einsetzbarer Methoden beantwortet werden. Auch hier geht Lindemann einen neuen Weg, indem er sich von der stringenten Zuordnung der Methoden zu bestimmten Phasen des Entwicklungsprozesses löst. Vielmehr schlägt er vor, Methoden der Situation entsprechend auszuwählen und bei Bedarf auch durch Modifikationen der Situation anzupassen.

Das Element „**Analyse des Ziels**“ dient zur Klärung konkreter und detaillierter *Anforderungen* an das zu entwickelnde Produkt. Zielkonflikte werden bereits an dieser Stelle intensiv herausgearbeitet. Parallel dazu werden die Anforderungen zum Beispiel in einer Anforderungsliste dokumentiert. Die einzelnen Teilelemente sind:

- Wie können wir Anforderungen ermitteln? [LINDEMANN 2005, S. 84 ff]: Es werden verschiedene konkrete Maßnahmen vorgeschlagen, um Anforderungen zu ermitteln (Fragetechniken, Textanalyse, Mind Mapping, Wirkungsnetz, Ursache-Wirkungsdiagramm, Recherche, Checklisten, Fragebögen/Interviews, Benchmarking, Reverse Engineering, Funktionsbetrachtungen). Auf die einzelnen Methoden soll hier nicht näher eingegangen werden. Wichtiger erscheint, dass Lindemann großen Wert auf die Zusammenstellung interdisziplinärer Teams schon bei der Anforderungsklä rung legt. Dabei soll, und dies wird auch durch die Methodenauswahl deutlich, der *Kunde* stets berücksichtigt und soweit möglich frühzeitig eingebunden werden. Dies kann anhand des Begriffs „implizites Wissen“ verdeutlicht werden, der an dieser Stelle eingeführt wird. Viele Informationen bezüglich Anforderungen sind gerade beim Kunden unbewusst vorhanden und werden nicht *explizit* ausgesprochen. Für den Erfolg eines Produkts sind aber gerade diese Anforderungen wichtig und können nur durch den engen Kontakt mit dem Kunden herausgearbeitet werden.
- Wie können wir Zusammenhänge zwischen den Anforderungen ermitteln? [LINDEMANN 2005, S. 89 ff]: Lindemann spricht hier von den Möglichkeiten der sich widersprechenden, unterstützenden und voneinander unabhängigen Anforderungen, also von der Identifikation möglicher Zielkonflikte. Nach der Ermittlung der

Doppelnennungen wird vorgeschlagen, weitere Untersuchungen mittels einer Konsistenzmatrix, einer Funktionsmodellierung oder eines Mind Maps beziehungsweise eines Wirkungsnetzes durchzuführen. Identifizierte Zielkonflikte sind rechtzeitig im Prozess aufzulösen, geeignete Maßnahmen oder Kriterien dazu werden aber nicht vorgeschlagen.

- Wie können wir Anforderungen gewichten? [LINDEMANN 2005, S. 92 ff]: Lindemann betont die Notwendigkeit einer Gewichtung, da sich daraus erst die Schwerpunkte für die Entwicklung ableiten lassen. Er greift nicht auf die bisher meist übliche Gliederung in Wünsche und Forderungen zurück und führt stattdessen eine numerische Gewichtung ein, um so den Grad der Übererfüllung einzelner Anforderungen differenziert bewerten zu können. Hinweise auf die Gewichtung kann man zum Beispiel vom Kunden mittels Interviews oder Fragebögen oder durch die Diskussion der betrachteten Merkmale anhand eines Kano-Modells bekommen. Durchführen lässt sich eine Gewichtung beispielsweise anhand des gewichtenden Anteils von Bewertungsmethoden.
- Wie können wir Anforderungen dokumentieren? [LINDEMANN 2005, S. 95 ff]: Die Bedeutung der kontinuierlichen Dokumentation der Anforderungen für eine erfolgreiche Produktentwicklung wird hervorgehoben. Nur so sind die notwendigen Informationen stets und aktuell für jeden verfügbar und können für die Lösungssuche, für die Eigenschaftsanalyse oder die Bewertung als wichtige Informationsquelle herangezogen werden. Die Dokumentation findet in einer Anforderungsliste statt, die in ihrem grundsätzlichen Aufbau der zum Beispiel auch von Ehrlenspiel [EHRLENSPIEL 2003, S. 359] benutzten entspricht.

Damit Anforderungen auch vernünftig bearbeitet, abgearbeitet und für die Entwicklung die richtigen Schwerpunkte gesetzt werden können, wird großer Wert auf eine gute **Strukturierung** der Entwicklungsaufgabe gelegt. Zusätzlich kann durch diese Maßnahmen oftmals auf weitere Anforderungen geschlossen werden. Das entsprechende Element des MVM gliedert sich in die unten stehenden Teilfragen. An dieser Stelle werden aber nur jene Fragen behandelt, die für die eigentliche Anforderungsklä rung von Bedeutung sind.

- Wie können wir wichtige Anforderungen und Produktmerkmale verknüpfen? [LINDEMANN 2005, S. 105 ff]: Um die Bedeutung einzelner Produktmerkmale beziehungsweise Anforderungen für die Kundenwünsche herauszuarbeiten, wird der Einsatz einer Verknüpfungsmatrix empfohlen. In dieser werden die Kundenanforderungen den Produktmerkmalen gegenübergestellt und deren Vernetzung untersucht. Um deutlichere Schwerpunkte setzen zu können, kann dies auch gewichtet durchgeführt werden.
- Wie können wir das Problem auf abstrahiertem Niveau beschreiben? [LINDEMANN 2005, S. 107 ff]: Oftmals wird die Durchdringung eines Problems aufgrund seiner Komplexität erschwert oder sogar verhindert. In solchen Situationen kann es hilfreich sein, durch abstrahierende Maßnahmen die Sichtweise auf ein Problem zu verändern, um sich so Zugang zum Problem zu verschaffen und daraus

auch Anforderungen ableiten zu können. Unterstützen können den Entwickler hierbei Methoden wie zum Beispiel Blackbox, Abstraktion oder Funktionsmodellierungen.

- Wie können wir Stärken und Schwächen ermitteln? [LINDEMANN 2005, S. 109]: Viele Projekte beschäftigen sich mit der Weiterentwicklung von vorhandenen Produkten, so dass eine Analyse der Schwachstellen der eigenen oder fremden existierenden Produkte eine wichtige *Quelle* für Anforderungen sein kann.

Zusammenfassend kann man sagen, dass eine sehr viel differenziertere Sicht auf die Anforderungsklä rung vorliegt, nicht zuletzt dadurch, dass durch die sehr freie Gestaltungsmöglichkeit des Entwicklungsprozesses regelmäßige „Sprünge“ zur Zielanalyse möglich sind.

3.1.5 Klären und Präzisieren der Aufgabenstellung nach Pahl et al.

Pahl et al. [PAHL ET AL. 2003] orientieren das von ihnen vorgeschlagene generelle Vorgehen beim Entwickeln und Konstruieren an dem in der VDI 2221 vorgeschlagenen Modell. Dies gibt einen sequentiellen Ablauf vor, in dem *Iterationen* und Rekursionen immer ein Zurückkehren zu früheren, bereits abgeschlossenen Arbeitsschritten und dem nochmaligen Durchlaufen der sich daran anschließenden ebenfalls bereits absolvierten Arbeitsschritte darstellen. Ein bewusstes Wiederaufsuchen von unterschiedlichen Elementen aus Gründen der Konkretisierung wie im *MVM* ist nur bedingt vorgesehen. Dies schlägt sich auch in dem von ihnen vorgeschlagenen Vorgehen zum Klären und Präzisieren der Aufgabenstellung nieder.

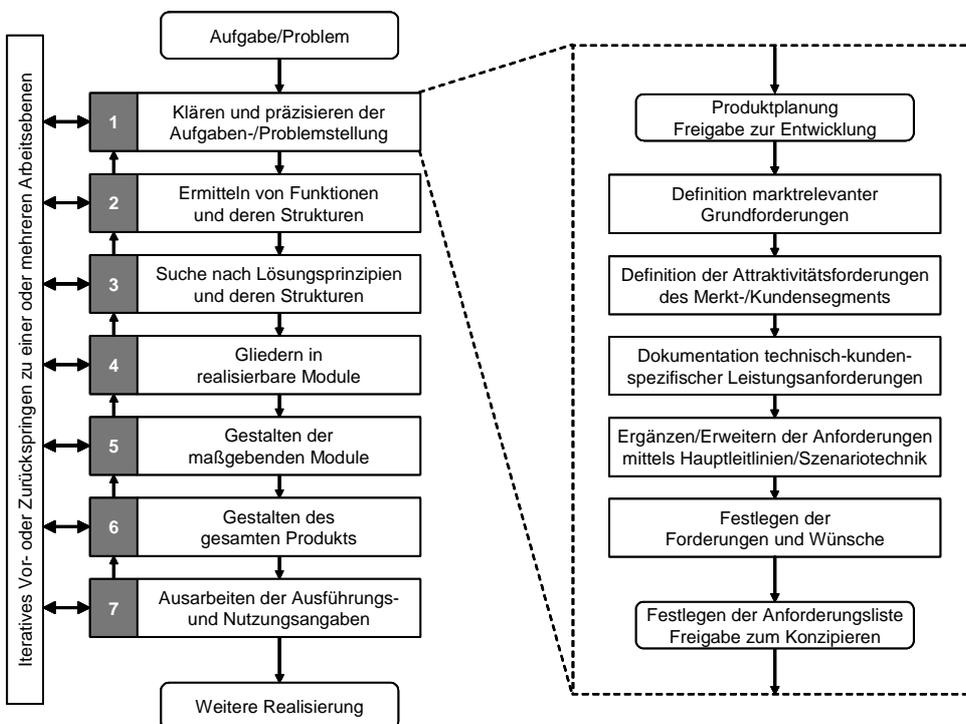


Abb. 3-3: Vorgehensmodell nach VDI 2221 [VDI 2221 1997] und Hauptarbeitsschritte zum Aufstellen einer Anforderungsliste [nach PAHL ET AL. 2003, S. 188]

Es handelt sich um ein sequentiell abzuarbeitendes Vorgehen, das aufgrund der sequentiellen Anordnung des Vorgehensmodells nur bedingt noch einmal aufgerufen und zur weiteren Konkretisierung der *Anforderungen* durchlaufen wird. Besonders der letzte Arbeitsschritt im Vorgehensplan zur *Anforderungsklä rung* vermittelt den Gedanken einer in den frühen Phasen der Entwicklung einmal erstellten, freigegebenen und dann nicht mehr veränderten Anforderungsliste.

Für Pahl et al. wird die Aufgabenstellung zur Entwicklung eines Produkts meist als

- Entwicklungsauftrag (extern oder intern durch die Produktplanung in Form eines Produktvorschlags),
- als konkrete Bestellung eines *Kunden* oder
- als Anregung aufgrund von zum Beispiel Verbesserungsvorschlägen und Kritik durch Verkauf, Versuch, Prüffeld, Montage oder aus benachbarten beziehungsweise eigenem Konstruktionsbereich

an die Entwicklungsabteilung herangetragen [PAHL ET AL. 2003, S. 187]. Diese Entwicklungsaufträge enthalten meist nicht nur Angaben zu Funktionalität und Leistungsumfang des Produkts, sondern üblicherweise auch noch wirtschaftliche und organisatorische Angaben. Dem Entwickler obliegt es nun, die lösungs- und gestaltbeeinflussenden *Produktanforderungen* zu *erkennen* und sie mit quantitativen Angaben versehen in einer Anforderungsliste zu dokumentieren.

Eine wichtige Vorarbeit für die Erstellung einer Anforderungsliste ist das Erkennen und Sammeln von Kundenwünschen und deren Übersetzung in Anforderungen an das zu entwickelnde Produkt. Als Unterstützung für diese Übersetzung wird die *Methode QFD* genannt, eine Erläuterung des notwendigen Vorgehens fehlt aber [PAHL ET AL. 2003, S. 188]. Des Weiteren sind in Zusammenarbeit mit dem Auftraggeber die Fragen

- Welchen Zweck muss die beabsichtigte Lösung erfüllen?
- Welche Eigenschaften muss sie aufweisen?
- Welche Eigenschaften darf sie nicht haben?

zu klären. Nach diesen ersten Schritten zu einer Anforderungsliste schlagen die Autoren ein zweistufiges Vorgehen vor.

- Zunächst soll der Entwickler die offensichtlichen Anforderungen definieren und dokumentieren.
- In einem zweiten Schritt werden diese Anforderungen mit Hilfe entsprechender Methoden ergänzt und erweitert.

In diesem Zusammenhang wird auf eine grundsätzliche Problematik bei der Anforderungsklä rung hingewiesen: Leider wird oftmals nur ein Teil der notwendigen Anforderungen geliefert. Pahl et al. sprechen in diesem Zusammenhang von *expliziten Anforderungen*, die festgelegt und beschrieben sind [PAHL ET AL. 2003, S. 191]. Dies können

zum Beispiel Informationen aus einem Vertrag oder ähnlichem sein, die vom Ingenieur in produktrelevante Größen umgesetzt werden können. Viele andere *Anforderungen* werden vom Kunden unausgesprochen, also *implizit* erwartet. Gerade diese sind aber für den Erfolg eines Produkts meist ausschlaggebend. Um diese impliziten Anforderungen klären zu können, ist es unbedingt notwendig, entweder den Kunden sehr genau zu kennen oder das entsprechende Marktsegment genau zu kennen¹¹. Methodische Unterstützung zur Ermittlung dieser Anforderungen wird nicht gegeben [PAHL ET AL. 2003, S. 192].

Die bisher zusammengetragenen Anforderungen sollen ergänzt und erweitert werden. Für diese Tätigkeiten werden der Einsatz von Leitlinien mit Hauptmerkmalslisten (Checklisten) und die Szenariotechnik vorgeschlagen. Durch den Einsatz von Hauptmerkmalslisten können mittels Assoziation weitere Erkenntnisse gewonnen werden, die dann wiederum zu neuen Anforderungen führen können. Im Rahmen der Szenariotechnik wird das zu entwickelnde Produkt entlang seines gesamten Produktlebenszyklus von der Entwicklung bis zur Entsorgung durchdacht, um so weitere Anregungen für Anforderungen zu erhalten. Es wird eingeräumt, dass die so aufgedeckten Anforderungen meist sehr unspezifische sind und oftmals nicht direkt in lösungsbestimmende oder gestalterische Produktparameter umgesetzt werden können [PAHL ET AL. 2003, S. 193].

Die Anforderungen sollen dann hinsichtlich ihrer Bedeutung (Wunsch/Forderung) untersucht und in einer Anforderungsliste dokumentiert werden. In diesem Zusammenhang wird kurz auf Aufbau und Strukturierungsmöglichkeiten von Anforderungslisten eingegangen.

Interessant ist, dass trotz des Eindrucks den der anfänglich vorgestellte deterministischen Vorgehensplan zur Anforderungsklä rung geweckt hat, eine deutlich differenziertere Betrachtung der Anforderungen über den Produktentwicklungsprozess stattfindet. Die Autoren räumen ein, dass eine Anforderungsliste zu Beginn einer Entwicklung niemals vollständig sein kann, sondern im Verlauf der Entwicklung wächst und sich ändert. Eine Anforderungsliste kann zu Beginn nicht alle notwendigen Anforderungen enthalten, sondern jeweils nur die, die unbedingt zur Bearbeitung des gerade anstehenden Arbeitsschrittes notwendig sind, weshalb der Inhalt einer Anforderungsliste nicht nur produkt- sondern auch arbeitsschrittabhängig ist [PAHL ET AL. 2003, S. 198].

3.1.6 „Target Specifications“ nach Ulrich & Eppinger

Ulrich & Eppinger [ULRICH & EPPINGER 2003] vertreten einen sechsphasigen generischen Entwicklungsprozess, der mit der Produktplanung beginnt und mit der Produktion endet. Die *Anforderungsklä rung* stellt in diesem Prozessmodell nicht eine eigene Phase dar, sondern erstreckt sich vor allem über die ersten zwei Phasen.

¹¹ *Anm. d. Verfassers: Dies kann bedeuten, dass man in einer fremden Disziplin fundierte Kenntnisse aufweisen muss beziehungsweise einen entsprechenden Experten zu Seite stehen hat.*

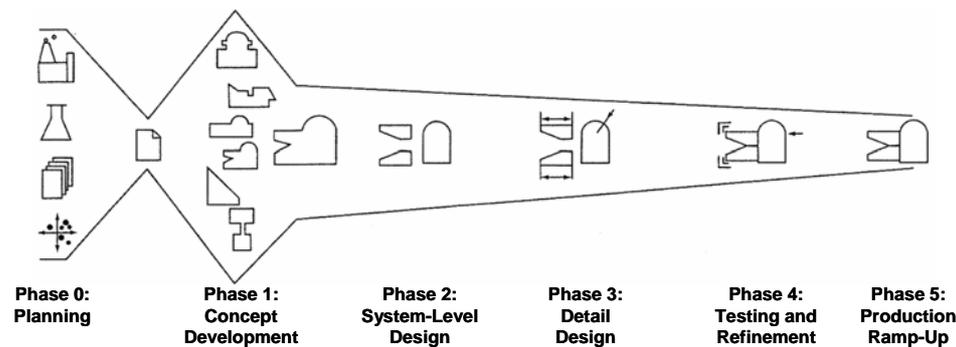


Abb. 3-4: Generischer Produktentwicklungsprozess nach Ulrich und Eppinger [Ulrich & Eppinger 2003, S. 14]

In der Planungsphase (Phase 0) werden die Geschäftsziele, die Kernhypothesen und Randbedingungen (business goals, key assumptions, and constraints) festgelegt. Diese Phase entspricht teilweise dem Element „Ziel planen“ im *MVM* nach Lindemann [LINDEMANN 2005, S. 40]. Der Kern der Anforderungsklärung findet in der Phase der Konzeptentwicklung (Phase 1) statt. Es werden die Bedürfnisse des Marktes beziehungsweise Kunden identifiziert, erste alternative Produktkonzepte generiert und bewertet und endgültige *Anforderungen* („Setting final specifications“) festgelegt.

Zunächst werden die Bedürfnisse des *Kunden* analysiert, die in einer hierarchischen Liste geordnet und gewichtet werden. Ziele dieses Vorgehens sind:

- Sicherstellen, dass das Produkt auf die Kundenbedürfnisse fokussiert ist.
- Latente und versteckte Bedürfnisse genauso wie explizite identifizieren.
- Faktenbasis erstellen, um die Produkthanforderungen zu begründen.
- Sicherstellen, dass keine wichtigen Kundenbedürfnisse vergessen worden sind.
- Entwicklung eines einheitlichen Produktverständnisses unter den Mitgliedern des Entwicklungsteams.

Um diese zu erreichen, muss eine hochqualitative Kommunikationsverbindung zwischen dem Kunden im Zielmarkt und dem Entwickler hergestellt werden. Diese Forderung basiert auf der Annahme, dass Innovation nur möglich ist und die Kundenwünsche nur dann vollständig getroffen werden können, wenn die, die direkt Details des Produkts bestimmen (Entwickler), mit dem Kunden interagieren und die Einsatzumgebung (use environment) des Produkts erleben. Dabei muss der Entwickler berücksichtigen, dass Kundenwünsche unabhängig von einem etwaigen Produkt sind. Man muss diese Wünsche adressieren können ohne die Art der Erfüllung dieser Wünsche zu kennen. Dagegen sind viele *technische Anforderungen* abhängig von dem ausgewählten Konzept [ULRICH & EPPINGER 2003, S. 55]. Zur Ermittlung der

Kundenbedürfnisse schlagen Ulrich & Eppinger Interviews, Focus Groups¹² und Beobachtung der Kunden bei der Nutzung des Produkts vor, wobei nach einer Studie von Griffin und Hauser [GRIFFIN & HAUSER 1993], auf die sich Ulrich & Eppinger stützen, die Effektivität eines Interviews wesentlich höher als die einer Focus Group ist. Von schriftlichen Befragungen wird aufgrund der systembedingten Starre abgeraten. Es wird empfohlen, zu Beginn möglichst alle verfügbaren Informationen aufzunehmen und diese erst später zu filtern, da man prinzipiell aus jeder Aussage des Kunden auf Bedürfnisse schließen kann. Aus diesen Rohdaten sollen anschließend die Anforderungen an das Produkt herausgefiltert werden. Ein spezielles Vorgehen wird hierzu aber nicht angegeben.

Im Anschluss daran werden die „target specifications“ ermittelt. Sie sollen eine präzise Beschreibung dessen sein, was das Produkt zu leisten hat und sind die Übersetzung der Kundenwünsche in technische Begriffe. Die Autoren differenzieren bei dieser Tätigkeit zwischen dem *Merkmal* und der *Ausprägung* einer Anforderung [ULRICH & EPPINGER 2003, S. 73]. Dazu werden die Kundenbedürfnisse in technische Beschreibungen übersetzt und mit Einheiten und Gewichtungen versehen. Da aus einer Kundenbeschreibung meist mehrere technische Beschreibungen, also Anforderungen resultieren können, schlagen Ulrich & Eppinger zur Unterstützung den Einsatz von QFD vor, um sie die vielfältigen Einflüsse der Kundenanforderungen auf technische Anforderungen analysieren zu können. Da Anforderungen bei technisch anspruchsvollen Produkten nicht endgültig zu Beginn geklärt werden können, sondern immer eine oder mehrere Reflektionen im Projektverlauf stattfinden müssen, sollen die Ausprägungen folgendermaßen definiert werden: Jede Anforderung ist mit einem Idealwert und einem Grenzwert, mit dem sich das Produkt am Markt gerade noch kommerziell behaupten würde, zu versehen. Zur Ermittlung dieser Ausprägungen wird die Durchführung zum Beispiel einer Marktrecherche oder eines vereinfachten Benchmarkings vorgeschlagen.

An diese schließt sich die Konzeptsuche an. Zu den bisher gefundenen Anforderungen sollen mögliche Lösungskonzepte gesucht, analysiert, bewertet und ausgewählt werden. Hat man ein Konzept ausgewählt, werden die Anforderungen und vor allem deren Ausprägungen nochmals reflektiert. Dies dient weniger dazu, weitere neu hinzugekommene Anforderungen zu klären, sondern vielmehr dazu, die endgültigen Ausprägungen der vorhandenen Anforderungen unter Bezugnahme auf das entwickelte Konzept einzugrenzen und festzulegen. Anfangs eventuell vorhandene Konflikte sollen aufgelöst werden (Finden des optimalen Kompromisses durch sog. Competitive Maps). Anschließend werden die Anforderungen auf die zu entwickelnden Subsysteme heruntergebrochen. Die Anforderungen an die Subsysteme müssen die Anforderungen an das Gesamtprodukt widerspiegeln. Anforderungen wie Gewicht oder Kosten können „einfach“ aufgeteilt werden und durch Summation überprüft werden. Bei anderen Anforderungen ist eine komplexere Betrachtung notwendig. Hier wird auf das Systems Engineering verwiesen (siehe Kapitel 5.1).

¹² moderierte Diskussionsrunde mit Kunden

3.1.7 Ermittlung der Kundenbedürfnisse nach Urban & Hauser

Urban & Hauser [URBAN & HAUSER 1993] beleuchten die Produktentwicklung vor allem aus der Sicht des Marketings beziehungsweise der Unternehmensführung und beschreiben aus dieser Perspektive den Prozess von der Entwicklung von Innovationsstrategien bis hin zur Produkteinführung.

In Bezug auf die *Anforderungsklä rung* befassen sie sich hauptsächlich mit der Ermittlung der Kundenbedürfnisse. Hierzu teilen sie die Kundenbedürfnisse in „basic needs“, „articulated needs“ und „exciting needs“ [URBAN & HAUSER 1993, S. 223 f] ein. Diese Gliederung entspricht der Untergliederung nach Kano [KANO ET AL. 1984] in Grundforderungen, Leistungsforderungen und Begeisterungsmerkmale. Um die sich dahinter verbergenden *Anforderungen* zu ermitteln, schlagen Urban & Hauser vor allem Interviews und Focus Groups vor, halten aber Interviews für die effizientere *Methode*. Die Autoren betonen, dass sowohl alle Beteiligten des Entwicklungsteams in die Erhebung der Kundenanforderungen eingebunden werden sollen, als auch alle Nennungen der *Kunden* aufgenommen werden sollen. Ersteres ist für ein gemeinsames Verständnis der Entwicklungsaufgabe, letzteres für ein umfassendes Verständnis der Kundenbedürfnisse notwendig [URBAN & HAUSER 1993, S. 224 ff]. Für ein strukturiertes Vorgehen werden die Repetory-Grid Methode [nach KELLY 1955] und die Echo technique [nach BARTHOL & BRIDGE 1968] vorgeschlagen. Bei der Repetory-Grid Methode wird der Kunde aufgefordert aus einer Menge vergleichbarer Produkte jeweils drei hinsichtlich eines Kriteriums zu vergleichen. Aus der dabei entstehenden Vielzahl an Einordnungen können Schwerpunkte für die Entwicklung abgeleitet werden. Bei der Echo technique beurteilt der Kunde vorhandene Produkte hinsichtlich „guter Dinge“ und „schlechter Dinge“, „nützlicher Dinge“ und „unnützer Dinge“ etc. [URBAN & HAUSER 1993, S. 226 ff].

An die Analyse der Kundenbedürfnisse schließt sich die Hierarchisierung und Clusterung der gewonnenen Informationen an. Die Notwendigkeit und Bedeutung dieses Schritts wird zwar sehr ausführlich diskutiert, die Empfehlung beziehungsweise Vorstellung konkreterer und praktikabler Vorgehensweisen fehlt aber. Zumindest unter dem Gesichtspunkt, dass aus diesen Informationen *technische Anforderungen* abgeleitet werden sollen und den Kundenanforderungen mittels *QFD* gegenübergestellt werden sollen. Ebenso fehlen Hilfsmittel, um im weiteren Vorgehen die technischen Anforderungen vollständig ermitteln zu können [URBAN & HAUSER 1993, S. 227-252; siehe auch AHRENS 2000, S. 69 ff].

Insgesamt besteht nur ein geringer Bezug zur technischen Seite einer Produktentwicklung.

3.2 Anforderungsklä rung in unabhängigen Ansätzen

Unter „unabhängigen Ansätzen“ werden im Sinne dieser Arbeit solche verstanden, die sich ausschließlich mit der *Anforderungsklä rung* befassen und nicht in ein eigenes umfassendes Vorgehensmodell für Produktentwicklungsprozesse eingebettet sind.

3.2.1 Erfassen und Handhaben von Produk tanforderungen nach Ahrens

Ahrens [AHRENS 2000] entwickelt ein iterativ anzuwendendes Vorgehensmodell zur Anforderungsklä rung. Innerhalb dieses Modells greift sie auf bekannte Methoden zurück. Da der *Kunde* in Ihren Augen herausragende Bedeutung für die Qualität der *Anforderungen* hat, baut ihr Vorgehen auf den Kundenbedürfnissen auf. Sie identifiziert 4 Dimensionen, die die Anforderungsklä rung aktiv beeinflussen: Kunden-, Transformations-, Konkretisierungs- und Handhabungsdimension.

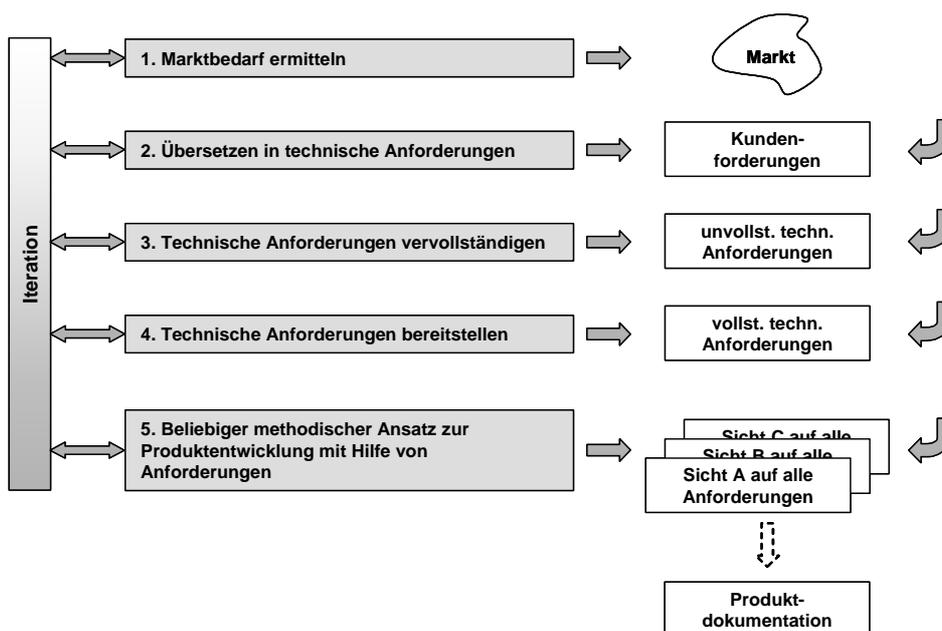


Abb. 3-5: Methodisches Konzept zur Erfassung und Handhabung von Produk tanforderungen
[nach AHRENS 2002, S. 118]

Ahrens bildet dieses Modell in einem informationstechnischen Ansatz ab, der zum einen in Form eines Methodenbaukastens [AHRENS 2000, S. 120] die Methodenauswahl in den unterschiedlichen Phasen unterstützen und zum anderen die Erfassung der Anforderungen hinsichtlich ihrer Dokumentation und Verwaltung ermöglichen soll.

- **Kundendimension** [AHRENS 2000, S. 120]: Die im Rahmen der Kundendimension zur Verfügung gestellten Methoden dienen der Erfassung der Marktbedürfnisse beziehungsweise der Kundenanforderungen. Ahrens fordert hierfür den direkten Kontakt zwischen Kunde und Entwickler und stellt deshalb auch nur solche Methoden zur Verfügung, die dies ermöglichen und erfordern (zum Beispiel Interview, Lead-User, Gebrauchstest etc.). Um die Effizienz des Vorgehens zu steigern und unnötige Kommunikationsschnittstellen zu vermeiden, empfiehlt sie, dass solche Analysen nicht, wie meist üblich, vom Marketing alleine sondern immer in Zusammenarbeit mit Entwicklern durchgeführt werden. Über die reine Methodenaufzählung hinaus vergleicht sie die angegebenen Methoden hinsichtlich Nutzen und Aufwand. Die zu bevorzugende *Methode* ist in ihren Augen das Interview.
- **Transformationsdimension** [AHRENS 2000, S. 122 ff]: Unter Transformation versteht Ahrens die Übersetzung von Kundenanforderungen in *technische Anforderungen*. Sie räumt entgegen ihrer Forderung bezüglich der Erhebung der Kundenanforderungen ein, dass es wohl nicht immer der Fall sein wird, dass ein Entwickler, der die Kundenanforderungen übersetzen soll, diese auch erhoben hat. Deshalb stellt sie mit der Transformationsdimension nicht nur Methoden für die Übersetzung (*QFD*-Matrix) sondern auch für das Verstehen von Kundenanforderungen (Abstraktion, Funktionsanalysen) zur Verfügung. Abstraktion und Funktionsanalyse stellen für Ahrens vorbereitende Maßnahmen zur Anwendung der *QFD*-Matrix dar.
- **Konkretisierungsdimension** [AHRENS 2000, S. 123 ff]: Die in der Konkretisierungsdimension angebotenen Methoden sollen zu Vollständigkeit und weitestgehender Präzision der Anforderungen führen. Des Weiteren fordert Ahrens für diese Methoden die Möglichkeit der informationstechnischen Unterstützung. Zum Erfassen von Anforderungen werden verschiedene Methoden angeboten (Leitlinien/Checkliste, Suchmatrix, Frageliste etc.), die sich in Aufwand und Nutzen stark unterscheiden. Eine eindeutige Empfehlung für eine Methode wird von der Autorin nicht gegeben, da diese Wahl sehr stark von der jeweiligen Situation abhängt. Damit eine Anforderungsliste wirklich von Nutzen ist, ist es von grundlegender Bedeutung, dass die ermittelten technischen Anforderungen eindeutig formuliert sind. Dies schließt ein, dass Anforderungen immer mit einem Zahlenwert, zumindest aber verbal genauer beschrieben werden. Ahrens stellt fest, dass dies in der Literatur zwar generell gefordert wird, dazu aber leider keine methodische Unterstützung angeboten wird.
- **Handhabungsdimension** [AHRENS 2000, S. 126 ff]: Unter diesem Punkt wird von Ahrens die weitere Handhabung von Anforderungen im Entwicklungsprozess verstanden. Dies meint zum einen das Dokumentieren, Gliedern, Verwalten und Ändern von Anforderungen. Die Vorschläge zum Gliedern von Anforderungen orientieren sich an Pahl et al. [PAHL ET AL. 2003] und Ehrlenspiel [EHRENSPIEL 2003]: Gliederung der Anforderungen anhand einer Leitlinie/Checkliste, Funktionsstruktur oder Baustuktur. Bezüglich der

Änderungsverfolgung von Anforderungen empfiehlt sie ein Vorgehen, wie es zum Beispiel auch von Ehrlenspiel vorgeschlagen wird. Den Leistungsumfang der vorhandenen rechnerunterstützten Ansätze hält Ahrens für nicht ausreichend.

Aufbauend auf diesem Modell entwickelt Ahrens ein Konzept zur rechnerunterstützten Erfassung und Handhabung von Anforderungen, um die existierenden theoretischen Vorgehensweisen für eine anwendungsgerechte Nutzung in der industriellen Praxis sinnvoll in bestehende Entwicklungsprozesse zu integrieren [AHRENS 2000, S. 150]. Sie konzentriert sich dabei vor allem auf die verteilte Anforderungsnutzung, die Suche nach Anforderungen und die Sicherheit in Bezug auf Freigaben und Änderungen. Die Strategien, die sie dafür entwickelt, bauen auf den oben vorgestellten Gedanken auf.

3.2.2 Ganzheitliches Anforderungsmanagement nach Danner

Danner [DANNER 1996] baut durch die Verknüpfung von *QFD* mit Methoden der Integrierten Produktentwicklung nach Ehrlenspiel [EHRENSPIEL 2003] einen ganzheitlichen Ansatz zum Anforderungsmanagement auf. Darunter versteht Danner „das methodische, umfassende Klären, Strukturieren, Vernetzen und Pflegen von *Anforderungen* aus externen Quellen – zum Beispiel Markt, Gesetze, Normen – und aufgrund unternehmensinterner Bedingungen sowie die Verknüpfung mit Methoden für die Lösungssuche und Bewertung – als Ausprägung einer Regelung des Erstellungsprozesses mit dem Ziel des aufwandsoptimalen Einsatzes von Unternehmensressourcen.“ [DANNER 1996, S. 6]. Im Gegensatz zu vorgenannten Autoren, die *QFD* nur in Form der Verknüpfungsmatrix aufgreifen, orientiert sich der Ansatz von Danner an dem gesamten Leistungsumfang des *QFD*. Daher geht dieser Ansatz über die reine *Anforderungsklä rung* hinaus und beschreibt auch die Lösungssuche und -auswahl im Sinne des *QFD*. Eine Weiterverwendung der Daten für spätere Projekte ist vorgesehen, wird sogar dringend empfohlen.

Als die wichtigste Aufgabe des *QFD* beschreibt Danner die Übersetzung der Kundenanforderungen. Um diese zu erheben und eine möglichst effektive und effiziente Weiterverarbeitung zu ermöglichen, schlägt er ein Team bestehend aus Marketing- und Entwicklungsfachleuten vor. Dieses Team soll die Kundenanforderungen aufbereiten und in *technische Anforderungen* übersetzen. Die Einbindung des *Kunden* in diesen Prozess ist aber nicht vorgesehen. Danner schlägt stattdessen eine *Methode* zur strukturierten Umformulierung von wörtlichen Kundenaussagen im Team nach King vor [KING 1994, S. 86].

QFD ist eine Methode zur frühzeitigen, umfassenden und strukturierten Anforderungsklä rung, deren Hauptziel die bestmögliche Erfüllung der Marktanforderungen, untergeordnete Ziele verkürzte Entwicklungszeiträume infolge geringerer Änderungsumfänge sind. Die Aufgabenklä rung im *QFD* erfolgt durch die Ermittlung der qualitätsbestimmenden Kriterien. Das Ergebnis sind aber nicht nur die technischen Anforderungen an das zu entwickelnde Produkt, sondern auch eine Priorisierung der notwendigen Prozesse der Planung und Entwicklung [DANNER 1996, S. 102]. Durch die durchgängige Modellierung der Einflussfaktoren, deren Gewichtungen und deren Zusammenhänge wird eine konsequente Anforderungsklä rung maßgeblich unterstützt. Üblicherweise lassen sich diese Faktoren aus Vorläuferprodukten oder ähnlichen, bereits existenten Produkten aus anderen Bereichen

mittels Analysen des Ist-Zustandes und der Fehlerursachen ableiten. Da in der Literatur Hinweise auf eine vollständige Erfassung von Anforderungen bei Neuentwicklungen fehlen [vgl. hierzu auch AKAO 1992, S. 64], schlägt Danner vor, QFD mit weiteren aus der integrierten Produktentwicklung bekannten Methoden zu verknüpfen und so nicht nur Weiterentwicklungen sondern auch Neuentwicklungen mit QFD unterstützen zu können. Als Beispiel werden die Checklisten und Fragelisten [EHRENSPIEL 2003, S. 350-355] angeführt [DANNER 1996, S. 112]. Um ein tieferes Verständnis des Systems und der inneren Systemzusammenhänge zu erlangen, führt Danner als weitere Methoden, die die Anforderungsklä rung mit QFD unterstützen können, die Black-Box, die Systemgrenzenschiebung, das Baumdiagramm und das Relationendiagramm an [DANNER 1996, S. 113 f].

Der Einsatz von QFD zur Anforderungsklä rung bedingt die Strukturierung der Anforderungen in Matrizen. Die unterschiedlichen Ebenen des QFD helfen, die Aufgabenstellung aus verschiedenen Perspektiven mit unterschiedlichem Abstraktionsgrad zu klären und weiter zu strukturieren.

3.2.3 Anforderungsklä rung nach Franke & Roth

Franke [FRANKE 1975] entwickelte einen Ansatz für die systematische *Anforderungsklä rung*, der von Roth [ROTH 2000] in dessen Darstellung des Entwicklungsprozesses übernommen worden ist. Dieser zweistufige Ansatz basiert auf der grundsätzlichen Idee, die Anforderungsklä rung nicht nur an verschiedenen technischen, wirtschaftlichen oder organisatorischen Kriterien zu spiegeln¹³, sondern eine weitere Dimension in Form des Produktlebenslaufs von der Planung bis hin zum Recycling einzubeziehen.

Frankes Ansatz beginnt mit einer Vorklä rung der *Anforderungen*. Diese kann zum Beispiel auf der Analyse von Vorgängerprodukten oder Konkurrenzprodukten beruhen. Im zweiten Schritt werden die gewonnen Informationen gezielt ausgeweitet und gesichert. Hilfsmittel hierfür können Recherchen unterschiedlichster Art sein. Franke hält es für notwendig, die Produktumgebung systematisch zu untersuchen. Das übliche Vorgehen wäre, die Produktumgebung in hinreichend kleine Bereiche zu zerlegen und anschließend gezielt nach die Produktentwicklung beeinflussenden Gesichtspunkten zu suchen. Hierfür ist aber ein hohes Abstraktionsvermögen des Entwicklers Voraussetzung.

Eine mögliche Lösung ist die systematische Untersuchung der Produktumgebung mittels der Suchmatrix nach Franke. In der einen Dimension dieser Matrix wird der Lebenslauf des Produkts abgebildet, in der zweiten die technisch-physikalische Umgebung, der Mensch, die wirtschaftliche Umgebung sowie normative Körperschaften¹⁴.

¹³ Diese Art der Betrachtung ist zum Beispiel in den bereits mehrfach erwähnten Checklisten und Leitlinien zu finden.

¹⁴ Gesetze, Normen, Richtlinien etc.

Die beschriebene Matrix enthält insgesamt 90 Felder und 250 darin eingetragenen Fragen. Zu einigen dieser Felder existieren Checklisten für eine noch detailliertere Erstellung der Anforderungsliste. Anhand der Assoziationshilfen in diesen Feldern kann ein zu entwickelndes Produkt systematisch auf Anforderungen untersucht werden. Dazu wird eine leere Suchmatrix erstellt, in deren Felder dann die Antworten auf die entsprechenden Fragen eingetragen werden. Es ist sinnvoll, sich frühzeitig Gedanken über die Bedeutung einzelner Punkte zu machen und die Suchmatrix dann auch in der Reihenfolge ihrer Wichtigkeit abzuarbeiten. Da man in frühen Phasen nicht immer alle Dinge abschätzen kann, empfiehlt Franke alle Felder der Matrix im Entwicklungsverlauf mehrfach durchzugehen, da sich viele Assoziationen erst zu späteren Zeitpunkten ergeben.

Mit diesem Ansatz werden Anforderungen also nicht nur aus der Sicht des Produkts, sondern aus der Sicht aller Dinge betrachtet, mit denen es in Berührung kommt beziehungsweise aus der Sicht aller Lebensphasen und nicht nur aus der Sicht der Nutzung. Die Suchmatrix soll im Entwicklungsprozess mehrfach durchlaufen werden, da die Anforderungsliste nach Roth immer erst für den gerade folgenden Konstruktionsabschnitt vollständig sein kann [ROTH 2000, S. 71].

Eigenschaften u. Bedingungen Lebenslauf-Phasen		a		Technisch-physikalische		
		b		Technologische u. funktionelle	Physikalische u. naturbezogene	
1	2	Nr.	1	Technisch-physikalische Eigenschaften und Bedingungen		
Herstellung	Produktplanung, Entwicklung, Konstruktion	1	1.1 Stand der Technik, Entwicklungs-Know-how	Herstellung	Produktplanung, Entwicklung, Konstruktion	Stand der Technik, bekannte Pionierkonstruktionen, neuartige Lösungsprinzipie, extreme Leistungen und Bau-Größen, Miniaturisierung; im Betrieb verfügbare und sonstige beschaffbare Erfahrungen, vergleichbare Konkurrenzprodukte; verfügbare Laboreinrichtungen (Prüfstände, Meßgeräte u.ä.); EDV (Hardware, Software), Analogrechner, zeichentechnische Hilfsmittel; Nachbarprodukte, Bau-reihen
	Arbeitsvorbereitung und Teilefertigung	2	2.1 Verfügbare Fertigungs- u. Betriebsmittel, technologisches Know-how	Arbeitsvorbereitung, Teilefertigung	Arbeitsvorbereitung, Teilefertigung	Verfügbare Werkzeugmaschinen, Vorrichtungen und sonstige Betriebsmittel nach Art, Abmessungen und Leistungen (z.B. Drehbankspitzenlänge und -höhe, realisierbare Schnittgeschwindigkeit, Pressenkkräfte, erzielbare Toleranzen und Rauigkeiten); Automatisierungsgrad der Fertigung, verfügbare Steuerungs- und Regelungsmöglichkeiten (NC-Maschinen); verfügbare Lagerteile, Halbzeuge und Materialien; Rücksicht auf Teilefamilien, Wiederholteile u. Gruppenfertigung, Stückzahlen, Losgrößen; Rücksicht auf technologische Bedingungen einzelner Fertigungsverfahren (verarbeitbare Werkstoffe, Bedingungen hinsichtlich Form und Gestaltung: z.B. Ausform- bzw. Aushebbarkeit u. fließgerechte Gestaltung bei Guß- u. Schmiedeteilen, geeignete Werkzeuganschnitte für Bohren, Räumen, Fräsen usw., zulässige Biegeradien, Lage der Gesenk- oder Formteilung, Rücksicht auf Walztexturen u.ä.); Zerspanvolumen, Verschnitt, Zahl der Aufspannungen, Zahl der Arbeitsgänge pro Teil, Fertigungstiefe des Produkts; Qualitätskontrolle, Meßeinrichtungen und -geräte, Prüfwerkzeuge
	Montage	3	3.1 Verfügbare Montagewerkzeuge und Hilfsmittel			

Abb. 3-6: Ausschnitt aus der Suchmatrix mit Checkliste nach Franke [ROTH 2000, S. 72]

Der hier beschriebene Ansatz wurde von Barrenscheen [BARRENSCHEEN ET AL. 1989] informationstechnisch weiterverarbeitet. Da es sich hierbei lediglich um eine softwaretechnische Umsetzung des Frankschen Ansatzes ohne inhaltliche Modifikation handelt, wird an dieser Stelle nicht näher darauf eingegangen werden.

Auf weitere Aspekte der Anforderungskklärung, wie zum Beispiel die Einbindung des Kunden, die Übersetzung von Kundenanforderungen, Zielkonflikte, Vollständigkeit und Interdisziplinarität geht Franke nicht ein.

3.2.4 Erfassung und Weiterverwendung von Anforderungen nach Gröger

Dieser rechnerunterstützte Ansatz von Gröger [GRÖGER 1990] basiert auf dem im Rahmen eines Sonderforschungsbereichs entwickelten Softwaresystem KALEIT, das den ganzen Konstruktionsprozess flexibel und durchgängig unterstützen soll. Dieses System enthält unter anderem ein Modul zur *Anforderungskklärung*, das im Folgenden näher beleuchtet wird.

Das Modul hat vor allem zwei Funktionen: die Unterstützung des Konstrukteurs bei der Aufstellung der Anforderungsliste und der Analyse der Konstruktionsaufgabe. Schnittstelle des Prozesses nach außen ist der Konstruktionsauftrag, mit dessen Eingang die Arbeit des Konstrukteurs beginnt. Dieser Auftrag enthält die mehr oder weniger spezifizierten Aufgaben, Größen und Randbedingungen. Ziel des angesprochenen Moduls ist vor allem die Generierung einer Anforderungsliste mit möglichst genau spezifizierten *Anforderungen*, Beschreibung der Hauptfunktionen und Zuordnung der Bedeutung der Anforderungen in Form von Forderungen und Wünschen. Die Strukturierung der Anforderungen orientiert sich an der Leitlinie mit Hauptmerkmalen nach Pahl et al. [PAHL ET AL. 2003, S. 194].

Die eigentliche Klärung der Anforderungen erfolgt auf der Basis der von Pahl et al. und der VDI 2221 [VDI 2221 1997] beschriebenen Methoden (Checklisten, Leitlinien...). Da die Leitlinie als zu unvollständig und zu grob strukturiert für einen Rechneinsatz angesehen wird [FELDHUSEN 1989, S. 133], hat man die Leitlinie in eine Baumstruktur überführt. Die ursprünglichen Hauptmerkmale bilden jetzt jeweils den ersten Zweig [siehe EHRENSPIEL 2003, S. 352]. Die weiteren Verzweigungen der Hauptmerkmale werden unter dem Gesichtspunkt möglichst alle Aspekte des Hauptmerkmals zu berücksichtigen vorgenommen, um so alle Facetten eines technischen Produkts zu erfassen.

Der Prozess der Klärung einer Anforderung sieht in KALEIT dann wie folgt aus:

- Eingabe der Aufgabenstellung in Prosaform
- Auswahl eines Hauptmerkmals (hier kann durch Assoziation das *Erkennen* von *Anforderungen* angestoßen werden, so wie es auch für den Einsatz von Leitlinien beschrieben ist [PAHL ET AL. 2003, S. 193 f])
- Auswahl der für eine Anforderung notwendigen Zweige des Merkmalbaums
- Eventuell Eingabe eines ergänzenden Textes

Die Forderung nach einem genormten Wellenende wird zum Beispiel durch die Auswahl folgender Attribute definiert: „Geometrie: Schnittstellen: Energie: Drehmoment: Wellenende: zylindrisch (DIN 748): horizontal“.

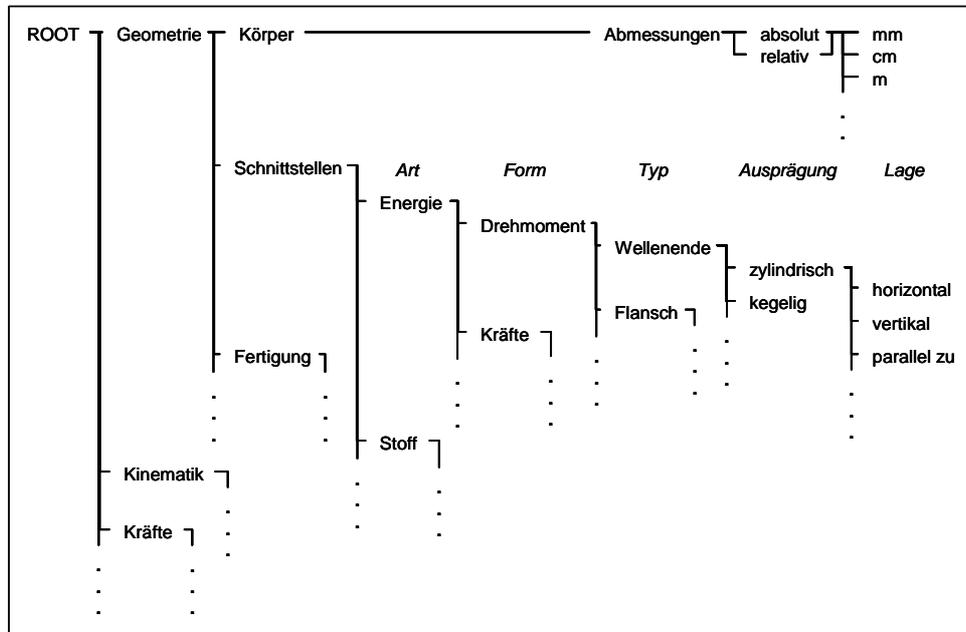


Abb. 3-7: Ausschnitt aus dem Merkmalbaum [FELDHUSEN 1989, S. 134]

Dieser Ansatz versucht vor allem ein Mittel zur Verwaltung, Formulierung und Definition von Anforderungen zur Verfügung zu stellen, eine Betrachtung der Kundeneinbindung oder von Zielkonflikten und ähnlichem findet nicht statt.

3.2.5 Anforderungsklä rung nach Grö ß er

Grö ß er [GRÖBER 1992] sieht die *Anforderungsklä rung* als Bindeglied zwischen der dem Marketing zufallenden Aufgabe, die Bedürfnisse des *Kunden* zu erforschen, und den Aufgaben der Konstruktion, die geforderten Eigenschaften in konkrete technische Lösungen umzusetzen [GRÖBER 1992, S. 1]. Ausgehend von dieser Annahme leitet er typische Probleme der Anforderungsklä rung ab [GRÖBER 1992, S. 2]:

- Unidisziplinär – d.h. technisch/naturwissenschaftlich oder kaufmännisch – orientierte Anforderungsermittler haben oft nur ihre *Anforderungen* an das Produkt im Auge und sind nicht in der Lage, einen anderen Blickwinkel zu akzeptieren.
- Terminologische Barrieren verhindern oder erschweren häufig die Verständigung bei der Ermittlung von Produkthanforderungen zwischen Marketing und Produktplanung auf der einen und Produktentwicklung und Konstruktion auf der anderen Seite.

- Heutige Unterstützungsformen der Anforderungsklärung (Suchmatrix, Frageliste, Checkliste etc.) berücksichtigen keine problem- beziehungsweise bearbeiterspezifischen Aspekte [GRÖßER 1992, S. 19].

Die Lösung für dieses häufig vorkommende Dilemma sieht Größer in der Entwicklung eines interdisziplinär anwendbaren methodischen Instrumentariums, um Anforderungen zu finden, zu analysieren und zu dokumentieren [GRÖßER 1992, S. 2]. Grundlage dieses Instrumentariums bildet ein Methodeninformationssystem, das aus einem Situationsanalysator, einem Methodenselektor und einem Methodenspeicher besteht.

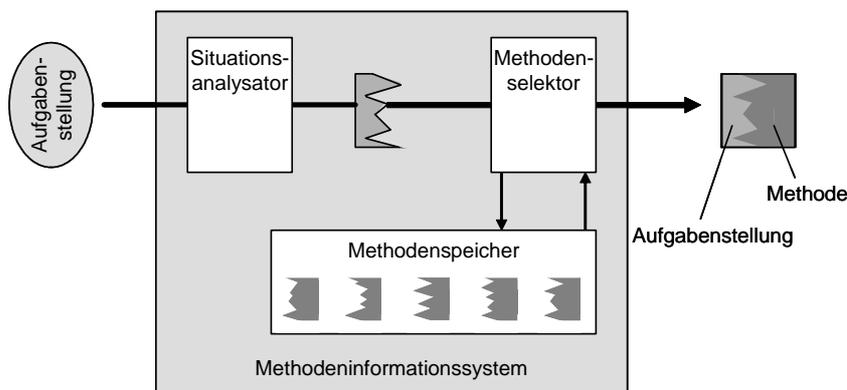


Abb. 3-8: Prinzipielle Struktur des Methodeninformationssystems [GRÖßER 1992, S. 41]

Eine Aufgabenstellung wird innerhalb des Systems unter Berücksichtigung problem- und bearbeiterspezifischer Randbedingungen profiliert. Der Bearbeiter erhält als Output die profilierte, also präzierte, Aufgabenstellung und die passende(n) Methode(n) zur Unterstützung der Anforderungsklärung. Erst durch die anschließende Anwendung der angebotenen Methode(n) ergeben sich konkreten Anforderungen.

Für diesen grundsätzlichen theoretischen Ansatz hat Größer [GRÖßER 1992, S. 43] zum einen das Methodeninformationssystem (AE-MIS) für die Ermittlung von Produkthanforderungen und zum anderen das System zur Analyse und Dokumentation von Produkthanforderungen (AE-DOS), das auf morphologischen Systematiken von Anforderungen und der Beziehung zwischen Anforderungen aufbaut, entwickelt.

Das Methodeninformationssystem AE-MIS berücksichtigt besonders die problem- und bearbeiterspezifischen Aspekte bei der Auswahl einer Methode. Um das dafür notwendige Instrumentarium entwickeln zu können, analysiert Größer sehr detailliert das von ihm so bezeichnete „Handlungssystem Anforderungsermittlung“ (HSAE) [GRÖßER 1992, S. 44], indem er die auf das System wirkenden Randbedingungen und die eingehenden Informationen den benötigten Informationen und Nebenwirkungen gegenüberstellt. Auf diese Weise stellt er Kriterien auf, die zum einen die problemspezifischen Aspekte und zum anderen die bearbeiterspezifischen Aspekte einer Anforderungsklärung ausmachen und als Eingangsdaten für den Situationsanalysator dienen.

Merkmale	Ausprägungen							
Aufgabensteller	externer Auftraggeber	Unternehmensleitung	Produktplanungsabteilung	andere Abteilung	eigene Abteilung	selbst		
Aufgabenumfang	Produktprogramm	Produkt						
Darstellungsart	verbal-mündlich	verbal-schriftlich	graphische	gegenständlich	symbolisch			
Vollständigkeit	nahezu vollständig	unvollständig						
Neuheitsgrad	Marktinnovation	Unternehmensinnovation	keine Innovation					
Innovationstyp	Strukturinnovation	Funktionsinnovation	Systeminnovation	keine Innovation				
Entwicklungsstand	Produktidee	ausgewählte Produktidee	Produktvorschlag	Anforderungssystem	Funktionsstruktur	Prinziplösung	Grobgestalt	Feingestalt
Komplexität	komplex	simplex						

Tab. 3-1: Morphologische Systematik der Aufgabenstellung für die Anforderungsermittlung [GRÖBER 1992, S. 59]

Innerhalb des Situationsanalysators werden zu den einzelnen Merkmalen Fragen formuliert und Antwortmöglichkeiten vorgegeben. Auf der Basis aller Antworten erfolgt dann die Auswahl einer Methode aus dem Methodenspeicher.

Der dabei mögliche Output beschränkt sich nicht nur auf die auch von Gröber aufgegriffenen typischen Methoden der Anforderungsklä rung, sondern erstreckt sich über ein wesentlich weiteres Methodenspektrum. Auch Methoden, die im klassischen Sinne nicht der Anforderungsklä rung zugeordnet werden, sind im Methodenspeicher zu finden: ABC-Analyse, Bionik etc. [GRÖBER 1992, S. 180]. Um über die angesprochenen situationsdefinierenden Kriterien die Methoden ansprechen zu können, hat Gröber mittels eines Methodenmodells eine Klassifikation der Methoden hinsichtlich Input, Output, Kosten etc. vorgenommen.

Gröber erhebt nicht den Anspruch, dass ein einmaliges Durchlaufen des Methodenselektors und die anschließende Anforderungsklä rung für eine endgültige Klä rung der Anforderungen ausreichend wären. Je nach Situation, die ja für die Methodenauswahl abgefragt wird, kann es notwendig sein, den Selektor mehrfach zu durchlaufen. Da sich aber nach jedem Durchlauf die Situation in die eine oder andere Richtung geändert haben sollte, kann dem Anwender bei jedem weiteren Schritt eine andere Methode situationsbezogen angeboten werden. Auf diese Weise ist es Gröber möglich, ein wesentlich differenzierteres und individuelleres Vorgehen für die Anforderungsklä rung zu entwickeln und dem Entwickler zur Verfügung zu stellen [GRÖBER 1992, S. 194 ff].

Das Analyse- und Dokumentationssystem für Produkthanforderungen AE-DOS dient der Unterstützung der Transformation eines mental vorhandenen Produktmodells in ein möglichst

vollständiges Anforderungssystem¹⁵. Dieses soll dem Anforderungsermittler helfen, mental gespeicherte Informationen abzurufen und extern zu dokumentieren. Der Ansatz basiert auf der morphologischen Systematik von Anforderungen und der Beziehungen zwischen Anforderungen. Die Software bildet diese Morphologie ab und die Abfrage von Anforderungen orientiert sich daran [GRÖßER 1992, S. 128 f]. Dieser Ansatz setzt aber nicht die oben angesprochenen Methoden des Methodenspeichers softwaretechnisch um, sondern dient nur der Dokumentation und Verwaltung von Anforderungen. Zwischen dem Einsatz des AE-MIS und dem Einsatz des AE-DOS muss also ein selbstständiger Methodeneinsatz durch den Anforderungsermittler stattfinden. Eine gewisse Grundkenntnis der Methoden und die Fähigkeit und Bereitschaft, sich in Methoden einzuarbeiten, muss also vorhanden sein.

Größer geht mit seinen hier vorgestellten Ansätzen vollkommen neue Wege und verbessert so, ohne neue Methoden einzuführen, durch die Hinzunahme von ursprünglich nicht der Anforderungsklä rung zugeordneten Methoden und die Berücksichtigung der stets unterschiedlichen Entwicklungssituation die Anforderungsklä rung maßgeblich. Wendet man diesen Ansatz innerhalb einer Entwicklung mehrfach an, bieten sich ganz neue Möglichkeiten der methodischen Unterstützung der Pflege und Erweiterung von Anforderungen im Projektverlauf.

3.2.6 Objektorientiertes Anforderungsmodell nach Humpert

Humpert [HUMPERT 1995] kritisiert, dass

- bestehende Entwicklungsmethoden nur mangelhaft in die Anwendungsgebiete Mechanik, Elektronik und Software integriert sind [HUMPERT 1995, S. 60],
- die zunehmend komplexer werdenden Produkte den Einsatz von leistungsfähigen Informationssystemen erfordern [HUMPERT 1995, S. 60],
- kein einheitlicher Formalismus für eine ingenieurdomänenübergreifende Beschreibung von *Anforderungen* existiert [HUMPERT 1995, S. 61] und
- keiner der bekannten Ansätze (unter anderem Fragelisten, Interviews, Analyse der Produktumgebung, Suchmatrix, *QFD*) den Prozess der Anforderungsverarbeitung durchgängig von der Planung und Durchführung der Anforderungserhebung über die Anforderungsevaluierung und -modellierung bis hin zur fortlaufenden Überprüfung der Anforderungen auf Erfüllung beziehungsweise Weiterverarbeitung im Produktentwicklungsprozess unterstützt [HUMPERT 1995, S. 7 ff; S. 57].

Aus diesen Gründen ist es das Ziel von Humpert ein durchgängiges Vorgehensmodells für die Anforderungsverarbeitung in der Produktentwicklung zu entwickeln.

¹⁵ Größer unterscheidet zwischen Anforderungsliste und Anforderungssystem. Anforderungsliste ist in seinen Augen die rein listenmäßige Dokumentation der Anforderungen. Die Gesamtheit der Anforderungen und deren die Vernetzung untereinander bezeichnet er dagegen als Anforderungssystem [GRÖßER 1992, S. 39].

Grundlage des Ansatzes von Humpert ist die Annahme, dass die Entwicklung von Produkten auf der Basis unterschiedlicher Partialmodelle erfolgt, von denen jedes eine eigene Sicht auf das Produkt beschreibt. Neu ist, dass von Humpert der Begriff des Anforderungsmodells als eines dieser Partialmodelle eingeführt wird. Neben der eigentlichen Abbildung von Informationen innerhalb des Anforderungsmodells muss zum einen der Präzisionsprozess der Anforderungen über den Entwicklungsprozess hinweg abgebildet und zum anderen die Vernetzung des Anforderungsmodell mit anderen Partialmodellen dargestellt und verarbeitet werden können [HUMPERT 1995, S. 65 ff]. Das grundlegende Vorgehen, das die geschilderten Forderungen erfüllen soll, ist nachfolgend beschrieben.

Zu Beginn steht die Transformation der in der Realität durch den *Kunden* oder beliebige am Produktentwicklungsprozess beteiligte Personenkreise umgangssprachlich formulierten Anforderungen in Objekte und Beziehungen des Anforderungsmodells.

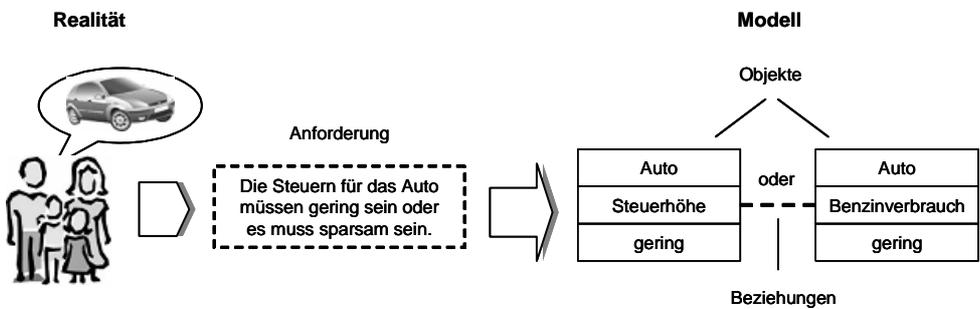


Abb. 3-9: Relevante und formalisierte Anforderung [nach HUMPERT 1995, S. 65]

Die Erhebung von Anforderungen wird mit Hilfe von Methoden wie Fragelisten, Einzelinterview, Gespräche etc. durchgeführt. Die anschließende Initialisierung des Anforderungsmodells hat zum Ziel, Anforderungen systematisch zu erheben und in ein formales Anforderungsmodell umzusetzen.

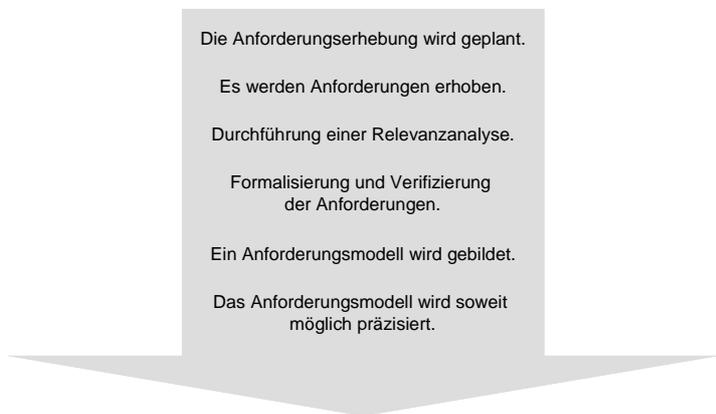


Abb. 3-10: Initialisierung des Anforderungsmodells [nach HUMPERT 1995, S. 65]

Im Anschluss an die Initialisierung des Anforderungsmodells erfolgen in einem iterativen Prozess die Vernetzung des Anforderungsmodells mit weiteren Partialmodellen sowie die Präzisierung der Anforderungen im Anforderungsmodell. Zunächst wird für die Vernetzung und die Präzisierung die bisher bekannte Produktstruktur initialisiert. Aufgrund dieser Initialisierung und der Untersuchung des Anforderungsmodells werden neue Partialmodellobjekte der Produktstruktur identifiziert und mit dem Anforderungsmodell vernetzt. Während dieser kontinuierlichen Weiterentwicklung der Produktstruktur werden die Ergänzung der Vernetzung sowie die Integration zusätzlich identifizierter Anforderungen in das Anforderungsmodell durchgeführt. Dieser stetig ablaufende Prozess präzisiert das Anforderungsmodell.

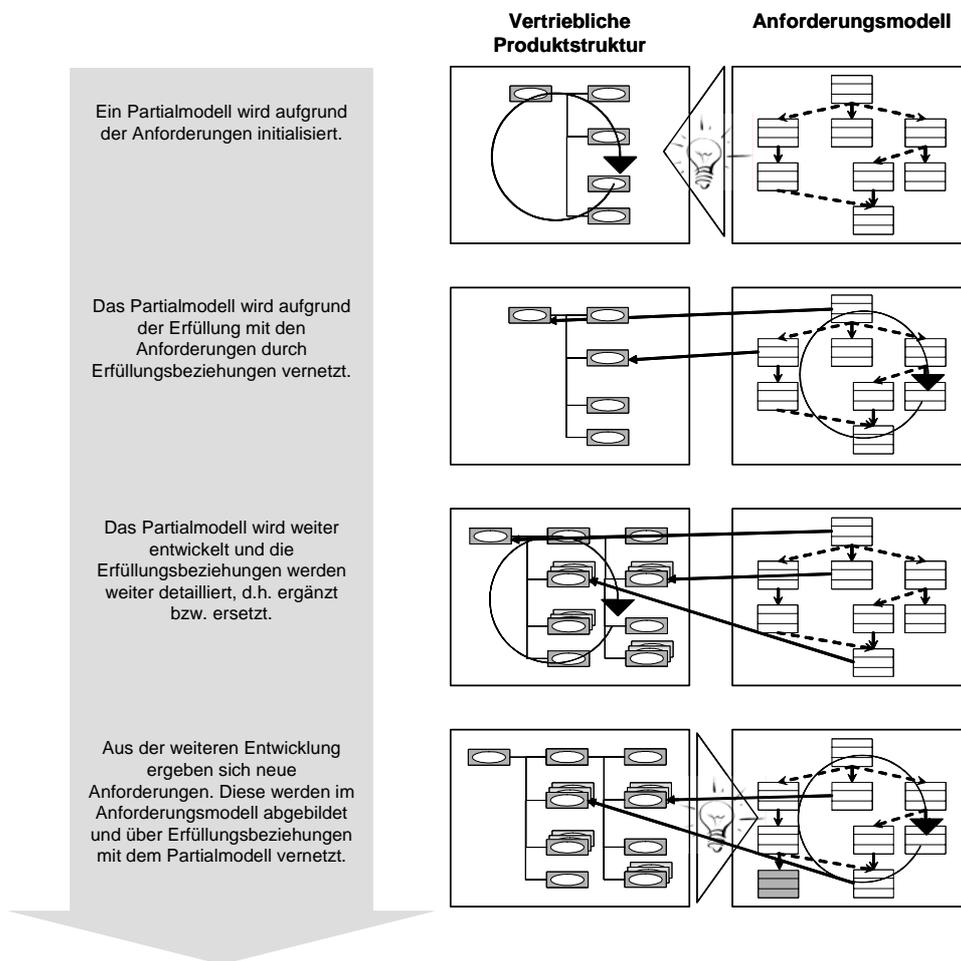


Abb. 3-11: Vernetzung und weitere Präzisierung des Anforderungsmodells [nach HUMPERT 1995, S. 70]

Die Präzisierung von Anforderungen aufgrund deren unterschiedlicher Beziehungen wird in die Eigenschaftspräzisierung (*Merkmalspräzisierung*) und die Ausprägungspräzisierung unterteilt. Die Eigenschaftspräzisierung ist eine Beziehung zwischen Anforderungsobjekten, welche die Präzisierung von Anforderungen durch Subanforderungen zum Ausdruck bringt.

Dabei beschreiben die durch die Subanforderungen geforderten Bezugseigenschaften die von den Topanforderungen geforderte Bezugseigenschaft präziser [HUMPERT 1995, S. 86]. Zum Beispiel kann sich aus der Anforderung „geringe Kosten“ die Anforderung „billiges Material“ und/oder „geringe Komplexität“ ergeben. Die Ausprägungspräzisierung ist eine Beziehung zwischen Anforderungsobjekten, welche die Präzisierung von Anforderungen durch Subanforderungen zum Ausdruck bringt. Dabei beschreiben die durch die Subanforderungen geforderten Bezugsausprägungen die von den Topanforderungen geforderte Bezugsausprägung präziser. Für die Ausprägungspräzisierung können drei Arten angegeben werden: Ordinalskaliert (zum Beispiel Geschwindigkeit hoch), Übergang durch Bereichseingrenzung (zum Beispiel Geschwindigkeit 150 - 240 km/h) oder absolutskaliert (zum Beispiel Geschwindigkeit 220 km/h) [HUMPERT 1995, S. 88].

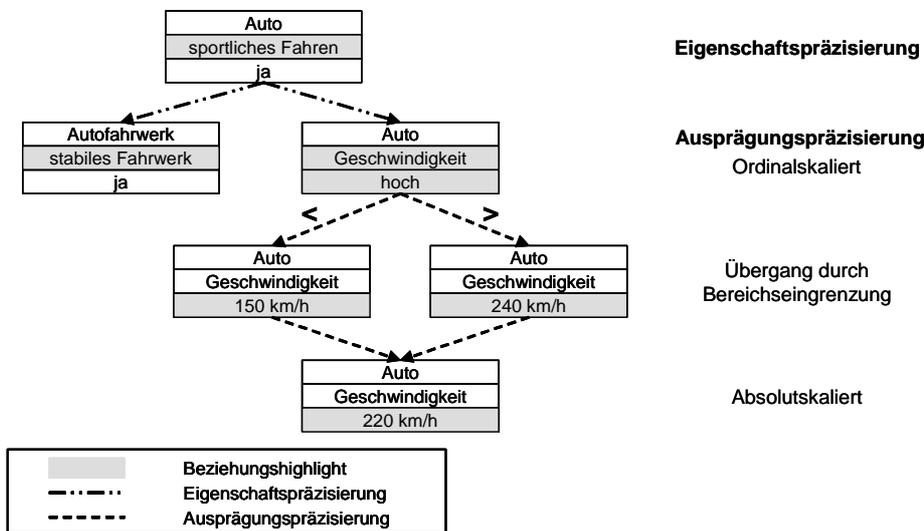


Abb. 3-12: Beispiel einer Eigenschafts- und Ausprägungspräzisierung [nach HUMPERT 1995, S. 88]

Um das hier beschriebene Vorgehen in der Praxis realisieren zu können, hat Humpert ein an dieser Stelle nicht näher betrachtetes Softwaretool entwickelt, das die notwendigen Partialmodelle erfassen, darstellen und verarbeiten kann. Interessanter scheint aber die Methodenklassifizierung zu sein, die Humpert entwickelt, um die Durchführung der bereits genannten Anforderungserhebung zu unterstützen. Er greift dabei auf bereits vorhandene Methoden zurück, ordnet diese aber einem Schema zu, das sich an den Produktlebensphasen, dem Detaillierungsniveau und dem Wissen sowohl des Anforderungserhebers als auch des Anforderungsträgers (zum Beispiel der Kunde) orientiert.

Erhebungsmethoden und -werkzeuge	Planungsfaktoren												
	Standardfragebogen	Individualfragebogen	Frageliste	Inventur	Prototyping	Einzelinterview	Fachgruppeninterview	Gruppeninterview	Schätzen	Beobachten	Eigenschaftsliste	Produktumfeldanalyse	Produktlebensphasenanalyse
Produktlebensphasen													
Produktdesign & -entwicklung													
Neuentwicklung				X	X	X	X				X	X	X
Anpassungsentwicklung				X	X	X	X				X	X	X
Variantenentwicklung	X	X	X							X	X		X
Entwicklung nach festem Prinzip	X	X	X							X			X
Prozessplanung & -entwicklung				X	X	X	X	X	X				
Beschaffung				X	X		X	X	X	(X)			
Produktion oder Dienstleistungserstellung				X	X		X	X	X	(X)			
Verifizierung/Prüfung								X					
Verpackung & Lagerung		X	X			X	X	X	X	X			
Verkauf & Vertrieb	X	X	X	X		X	X	X	X	(X)			
Montage & Inbetriebnahme		X	X	X		X	X	X	X	X			
Technische Unterstützung & Wartung	X	X	X			X	X	X	X	X			
Benutzung & Wiederverwendung	X	X	X			X				X			
Marketing & Marktforschung	X	X	X	X		X	X		X	X			
Detaillierungsniveau													
hoch		X		X	X	X	X				X	X	X
tief	X		X				X	X	X	X			
Produktwissen (Erheber)													
hoch		X		X	X	X	X	X	X	X		X	X
tief	X		X							X			
Produktwissen (Träger)													
hoch	X	X	X	X	X	X	X				X	X	X
tief								X	X	X			

Tab. 3-2: Einsatztabelle für Anforderungserhebungsmethoden und -werkzeuge [nach HUMPERT 1995, S. 114]

3.2.7 Gewinnung von Anforderungen nach Kruse

Kruse [KRUSE 1996] stellt zu Beginn seiner Ausführungen fest, dass das Spektrum der *Anforderungen* noch komplexer wird, wenn ein Erzeugnis durch Zusammenarbeit von Entwicklungsteams aus unterschiedlichen Fachdisziplinen, wie sie zum Beispiel bei der Entwicklung eines verfahrenstechnischen Systems zusammenwirken, realisiert werden soll [KRUSE 1996, S. 9]. Unter Disziplin versteht er in diesem Zusammenhang jenen Wissensbereich, den üblicherweise ein qualifiziertes Team von Fachleuten für eine technische Systemerstellung bearbeitet [KRUSE 1996, S. 17]. Zwischen diesen Disziplinen ist die Verständigung aber oftmals schwierig, da sie nicht die gleiche Sprache sprechen [KRUSE 1996, S. 17 ff].

Der *Kunde* ist in den Augen Kruses der wichtigste Partner des Entwicklungsteams. Da er das zu entwickelnde System kaufen und einsetzen soll, sind seine Anforderungen die wichtigsten Entwicklungsparameter. Meist bereitet es aber große Schwierigkeiten, dem externen oder internen Kunden die fast immer vorhandenen Vorstellungen über das zu realisierende System

in Form von Anforderungen zu entlocken [KRUSE 1996, S. 71]. Zur methodischen Unterstützung greift auch Kruse für die Anforderungserhebung auf bereits bekannte Methoden (Suchmatrix, Checkliste, *QFD* ...) zurück, schlägt also keine neuen Methoden vor. Die von Kruse aufgestellten Thesen beziehungsweise Feststellungen zur *Anforderungsklä rung* lassen sich wie folgt zusammenfassen:

- Der größte Teil der Anforderungen ist in der Form, wie er in das Projekt hineingetragen wird, nur bedingt zu nutzen. Ein Projektteam muss mit methodischer Unterstützung die Aufbereitung durchführen [KRUSE 1996, S. 67].
- Anforderungen können im Laufe ihrer Definition verfeinert werden. Aus einer Anforderung werden mehrere Anforderungen abgeleitet, die eine Aufgabenstellung konkreter beschreiben [KRUSE 1996, S. 13].
- Eine Anforderung kann sich aus mehreren *Ausprägungen* ergeben („Schnittmenge“) [KRUSE 1996, S. 56].
- Die Tätigkeit der Gewinnung von Anforderungen ist während der Projektbearbeitungszeit niemals abgeschlossen, sondern muss kontinuierlich ihre Fortsetzung finden [KRUSE 1996, S. 67].

Auf diesen Aussagen baut Kruse seine weiteren Ausführungen auf und unterscheidet zwischen verschiedenen Fällen der Anforderungsklä rung, die sich in der Qualität der zu Beginn zur Verfügung stehenden Anforderungen stark unterscheiden. Diese Fälle ordnet er unterschiedlichen Entwicklungssituationen zu.

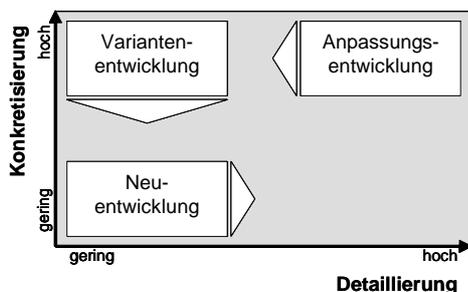


Abb. 3-13: Konkretisierungs-/Detaillierungsgrad von Kundenanforderungen in Abhängigkeit von der Entwicklungsart [KRUSE 1996, S. 74]

Kruse sieht vor allem in der Situation der Neuentwicklung die Kommunikation zwischen Kunde und Auftragnehmer besonderen Herausforderungen ausgesetzt, da die Umsetzung der Anforderungen in Systemmerkmale beiden Seiten verständlich sein muss, ohne dass der Aufwand für einen solchen Abgleich zu groß wird [KRUSE 1996, S. 76].

Kruse geht neue Wege bei der Ermittlung der Kundenanforderungen, indem er feststellt, dass diese nicht in den leeren Raum formuliert werden können und stattdessen direkt an eine abstrakte oder konkrete Eigenschaft des Gesamtsystems, einer Systemkomponente oder eines

Systemaspektes zu richten sind. Sollte dieses Anforderungsobjekt, also das Bauteil oder die Baugruppe noch nicht konzipiert sein, was bei Neuentwicklungen meist der Fall sein dürfte, so soll ein entsprechendes „Entwicklungsobjekt“ definiert werden, dem dann eine der Anforderung entsprechende Eigenschaft zugeordnet wird. Bei diesem Objekt muss es sich nicht zwingend um ein herstellbares Teil handeln, sondern kann auch einen logischen Raum beschreiben [KRUSE 1996, S. 82]. Damit ähnelt dieser Ansatz sehr dem von Humpert [HUMPERT 1995] beschriebenen Vorgehen.

Intensiv werden von Kruse auch die unterschiedlichen Beziehungsarten zwischen Anforderungen behandelt. Kruse unterteilt diese in strukturelle und semantische Beziehungen. Strukturelle Beziehungen sind Relationen zwischen Anforderungen, wobei in diesem Fall eine Anforderung einen Teil einer anderen Anforderung darstellt [KRUSE 1996, S. 78 ff]:

- **Konkretisierung:**
Übergang einer Anforderung von der Anforderungsdefinitionsphase in einen folgenden Punkt der Prozesskette (zum Beispiel von der Anforderungssicht über die Funktionssicht zur Gestaltsicht).
- **Spezialisierung:**
Zuwachs von Eigenschaften für eine Anforderung innerhalb eines Entwicklungsprozesses. Die Anforderung wird detaillierter beschrieben.
- **Dekomposition:**
Zerlegung einer Anforderung in ihre Teilforderungen.
- **Variation:**
Definition von alternativen Anforderungen, zum Beispiel zum Zwecke der Lösungsfindung.

Diese Unterteilung entspricht ebenfalls der von Humpert [HUMPERT 1995] beschriebenen Eigenschafts- und Ausprägungspräzisierung. Seitens der semantischen Beziehungen, die die Wirkung zwischen Anforderungen auf gleicher oder unterschiedlicher Strukturebene beschreiben, nennt Kruse folgende Möglichkeiten [KRUSE 1996, S. 81]:

- **Ausschluss:** Zwei Anforderungen sind miteinander nicht realisierbar.
- **Konkurrenz:** Zwei Anforderungen wirken gegenläufig auf die gleiche Eigenschaft.
- **Unterstützung:** Zwei Anforderungen wirken gleichsinnig auf eine Eigenschaft.

Im Rahmen der Anforderungsaufbereitung, die sich direkt an die Anforderungsermittlung anschließt und über den Entwicklungsprozess hinweg weiter verfolgt werden muss, ist eine der wichtigsten Tätigkeiten die Ermittlung der strukturellen und semantischen Beziehungen zwischen Anforderungen. Beziehungen, die bei der Anforderungserfassung schon bekannt beziehungsweise erkennbar sind, werden sofort festgehalten. Die anschließend dennoch vorhandenen Lücken in der Struktur müssen aufgefunden und gefüllt werden; einige werden sofort zu schließen sein, die Mehrheit wird aber, gerade bei Neuentwicklungen, erst im Rahmen der entwicklungsbegleitenden Anforderungsaufbereitung zu bearbeiten sein. Auch neue Anforderungen sollen sofort in die bis dahin vorhandene Struktur eingeordnet werden.

Eine Hilfe bei dieser Einordnung ist die vorherige Zuordnung einer Anforderung an ein Entwicklungsobjekt. Kruse empfiehlt die regelmäßige Überprüfung der Qualität der Anforderungsstruktur und damit der Anforderungen an sich anhand der semantischen Beziehungen [KRUSE 1996, S. 96 ff].

- Eine hohe Anzahl sich ausschließender Beziehungen zeigt eine mangelhaft abgestimmte Struktur, was die Entwicklung erschweren kann, da Lösungen immer wieder kollidieren können. Die Anforderungen müssen überprüft werden, um gegen Ende die Zahl dieser Beziehungen gegen null zu bringen.
- Viele konkurrierende Beziehungen lassen viele Iterationsschleifen in der Entwicklung erwarten, eine Entwicklung und damit Lösung des Problems ist aber prinzipiell möglich.
- Viele unterstützende Beziehungen zeugen von einem gut abgestimmten Anforderungsprofil. Gleichzeitig kann dies aber auf ein eng abgestimmtes Anforderungsnetz deuten, das auf Veränderungen sehr empfindlich reagieren kann.
- Existieren zwischen den Anforderungen nur wenige Beziehungen, kann dies zweierlei bedeuten: Entweder sind die Anforderungen sehr genau definiert, so dass keine Eigenschaftsüberschneidungen vorkommen oder die Beziehungsanalyse wurde nur unzureichend durchgeführt.

Die große Bedeutung der sorgfältigen Behandlung vor allem der sich ausschließenden und konkurrierenden Beziehungen zeigt sich auch darin, dass dazu eigene Forschungsaktivitäten existieren [zum Beispiel EILETZ 1999].

Kruse setzt sich auch mit der Wertigkeit von Anforderungen auseinander [KRUSE 1996, S. 84]. Hier baut er auf die von ihm entwickelte Anforderungsstruktur auf. Kruse schlägt eine Einteilung in Fest-, Ziel- und Wunschforderungen vor: *Festforderung* nutzt er um die nutzbaren Lösungen herauszufiltern, Ziel- und Wunschforderungen dienen dann der Wertung der nutzbaren Lösungen untereinander. Die Zuordnung der Wertigkeiten der Anforderung erfolgt in zwei Schritten und orientiert sich an der Gewichtung im Zielsystem einer Nutzwertanalyse [EHRENSPIEL 2003, S. 488 f]. Die anfänglich erhobenen Anforderungen werden zum Beispiel hinsichtlich ihrer Bedeutung für den Kunden gewichtet. Anschließend werden die ermittelten strukturellen Beziehungen genutzt, um den untergeordneten Anforderungen ihre Gewichtung zuzuordnen. Kruse erhält so eine normierte Anforderungsgewichtung [EHRENSPIEL 2003, S. 107].

Aus diesen grundsätzlichen Überlegungen leitet Kruse ein Tool zur informationstechnischen Unterstützung der Anforderungsklä rung ab, auf das aber an dieser Stelle nicht näher eingegangen wird.

3.3 Requirements Engineering

Im Zuge der immer rascher voranschreitenden Steigerung der Leistung von Computern und der dafür notwendigen Software entwickelte sich das Software Engineering, das als Gegenstück der Informationstechnologie zur methodischen Produktentwicklung gesehen werden kann. Bald erkannte man, dass eine schwierige Phase bei der Entwicklung einer neuen Software die Definition des Leistungsumfangs ist¹⁶, die einen entsprechend großen Einfluss auf den Erfolg eines Produkts hat¹⁷.

Aus dieser Erkenntnis heraus entwickelte sich eine eigene Strömung innerhalb des Software Engineerings, das so genannte Requirements Engineering. Requirements Engineering bezeichnet diejenige Teildisziplin des Software Engineerings, die sich mit der Ermittlung und Konstruktion der *Anforderungen* an Systementwicklungen beschäftigt [ANDELFINGER 1997, S. 63]. Nach Deifel [DEIFEL 2001, S. 2] beinhaltet das Requirements Engineering

- das systematische Erarbeiten von Anforderungen an eine Software,
- die Festlegung, welche Anforderung wo und wann realisiert werden soll,
- die Dokumentation zu realisierender Anforderungen in einer in sich konsistenten und validierten Spezifikation,

um zum einen Fehlentwicklungen aufgrund unverständener Problemstellungen zu vermeiden und zum anderen die Kommunikationsprobleme zwischen den an einer Systementwicklung Beteiligten zu bewältigen [ANDELFINGER 1997, S. 76].

In Anlehnung an das Prozessmodell von Pohl [POHL 1996], lässt sich das Requirements Engineering in vier Phasen untergliedern, die von Deifel [DEIFEL 2001, S. 7-23] wie unten stehend beschrieben werden. Innerhalb dieses Vorgehens ist keine sequentielle Abarbeitung vorgesehen, vielmehr sind ständige Rückkopplungsprozesse notwendig.

- **Herausarbeiten von Anforderungen:** Die an eine zu entwickelnde Software gestellten Anforderungen werden identifiziert. Dabei wird von einem direkten Kontakt mit dem *Kunden* ausgegangen. Dessen meist vage Äußerungen dienen der Identifizierung der damit verbundenen Anforderungen. Das Requirements Engineering unterscheidet in diesem Zusammenhang die Begriffe *Quelle* und *Kanal* einer *Anforderung* [DEIFEL 2001, S. 8]. Unter *Quelle* ist die Herkunft einer Anforderung zu verstehen. Mit *Kanal* wird hingegen der Anlass oder Weg, auf dem

¹⁶ „The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is as difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems.“ [BROOKS 1987, S. 17]

¹⁷ „No single Engineering activity can do more to ensure the success of a program than to properly manage requirements.“ [LINDSTROM 1993, S. 56]

eine Anforderung zu einem Hersteller beziehungsweise Entwickler gelangt, verstanden. Je individueller ein Produkt ist, desto weniger Quellen und Kanäle stehen zur Verfügung. Zur Unterstützung dieser Phase dienen Fragetechniken, Analyse der Einsatzumgebung, Anforderungen aus früheren Projekten, Szenarioanalyse zur Herausarbeitung von Abläufen in einem Softwaresystem und das so genannte Prototyping. Unter letzterem wird die möglichst frühzeitige Evaluation denkbarer Lösungen anhand ausführbarer Programme verstanden, um daraus Anforderungen abzuleiten.

- **Verhandlung von Anforderungen:** Diese Phase ist vergleichbar mit dem Element „Ziel strukturieren“ des *MVM* nach Lindemann [LINDEMANN 2005, S. 41]. Dabei sollen Konflikte aufgedeckt und gelöst, Anforderungen bewertet und priorisiert werden. Zusätzlich findet hier die der Softwareentwicklung eigene Versionsplanung statt. Hierzu existiert wenig methodische Unterstützung.
- **Spezifikation von Anforderungen:** In der Spezifikationsphase werden die ermittelten Anforderungen spezifiziert. Dies erfolgt meist in Form von beliebig gegliedertem Prosatext. Das ist der Ausgangspunkt für die Erstellung eines Systems und dient während der Entwicklung zur Kommunikation mit dem Kunden, um weitere Anforderungen herauszuarbeiten. Anforderungen sind demnach einem Erweiterungs- und Änderungsprozess unterlegen. Spätestens mit der Fertigstellung der Software muss eine in sich konsistente Spezifikation existieren, die der tatsächlich entwickelten Software entspricht. Nach Abschluss der Entwicklung werden die Anforderungen zur Validierung herangezogen. Zur Unterstützung werden neben der rein textuellen Beschreibung vor allem grafische Beschreibungsformen herangezogen, da diese eine übersichtlichere und intuitivere Darstellung mit sich bringen. Als Standard hat sich hierfür in den letzten Jahren zum Beispiel die objektorientierte Beschreibungssprache UML¹⁸ [BOOCH ET AL. 1999] herauskristallisiert.
- **Validierung/Verifikation von Anforderungen:** Unter Validierung ist die Überprüfung, ob das zu entwickelnde System den Wünschen des Kunden entspricht, zu verstehen. Verifikation hingegen meint die Kontrolle, ob das erstellte System der Spezifikation entspricht. Das bedeutet, dass eine entwickelte Software zweierlei grundsätzlichen Prüfungen unterzogen wird. Sollte sich dabei zum Beispiel herausstellen, dass die Software zwar die Verifikation (sie entspricht also der Spezifikation) nicht aber die Validierung (sie entspricht also nicht den Wünschen des Kunden¹⁹) besteht, würde dies bedeuten, dass die Spezifikation nicht den Wünschen des Kunden entspricht, beim Herausarbeiten der Anforderungen also nicht sorgfältig genug gearbeitet worden ist, oder die Vorstellungen des Kunden sich während der Entwicklung geändert haben, er dies aber nicht kommuniziert hat. Diese bewusste

¹⁸ UML: Unified Modeling Language

¹⁹ Dies wird meist erst bei der Anwendung der Software durch den Kunden aufgedeckt, oftmals sogar erst nachdem das Produkt schon einige Zeit im Einsatz ist und sich ganz spezifische Anwendungssituationen ergeben.

Unterscheidung von Validierung und Verifikation wurde in der oben beschriebenen Produktentwicklungsmethodik bisher nicht vorgenommen, findet aber im Zuge der zunehmenden Entwicklung mechatronischer Produkte zum Beispiel über die VDI-Richtlinie 2206 [VDI 2206 2004] Eingang in andere Disziplinen.

Ähnlich wie in der Produktentwicklung wird auch im Requirements Engineering gefordert, dass durch Anforderungen nur das „was“ aber nicht das „wie“ festgelegt wird. Andelfinger ist aber der Meinung, dass sich diese Theorie in der Praxis meist nicht durchführen lässt [ANDELFINGER 1997, S. 78]. Zum einen könne je nach Perspektive dieselbe Fragestellung einmal als „was“ und einmal als „wie“ verstanden werden, zum anderen müsse das „was“ oftmals mit dem „wie“ in Form der technischen Möglichkeiten abgestimmt werden, was ein Denken in Lösungen erfordert. Siddiqui [SIDDIQI 1994, S. 19] schreibt hierzu: „Dem klassischen Postulat des Requirements Engineering liegt die problematische (...) Annahme zugrunde, dass es eine objektive Welt gebe, zu der (...) ein neutraler Zugang möglich sei, der sich höchstens im Grad der Detaillierung unterscheide, jedoch von konkreten Methoden und den durchgeführten Personen im wesentlichen unabhängig sei.“ Deswegen wird das Requirements Engineering als dynamischer und perspektivischer Erkenntnisprozess verstanden, in dessen Verlauf Anforderungen sich dynamisch im Laufe der sozialen Interaktion zwischen den Beteiligten ergeben und immer wieder überprüft werden müssen [ANDELFINGER 1997, S. 79]. Nach Naur [NAUR 1985] muss jede Systementwicklung primär als eine Form von Theoriebildung über einen gegebenen Weltausschnitt angesehen werden, der zunächst von den Beteiligten inhaltlich verstanden sein muss, bevor eine programmiertechnische Modellierung und Realisierung erfolgen kann.

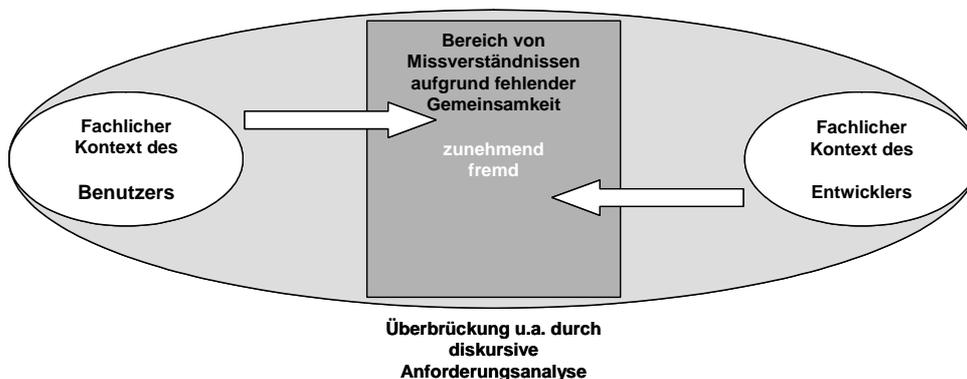


Abb. 3-14: Verständigungsproblematik bei der gemeinsamen Begriffsrekonstruktion
[ANDELFINGER 1997, S. 203]

Da aber gerade dieses inhaltliche Verstehen der Beteiligten aufgrund der vorherrschenden Interdisziplinarität Schwierigkeiten bereitet, entwickelte Andelfinger einen Ansatz zur diskursiven Anforderungsanalyse dessen Kernaufgabe die nachvollziehbare Identifizierung, Systematisierung und Modellierung von Anforderungen ist [ANDELFINGER 1997, S. 184]. Grundlegende Zielsetzung ist die Erarbeitung eines von allen an einer Entwicklung

Beteiligten gemeinsam geteilten Verständnisses über die für die Systementwicklung relevanten Aspekte, Bedingungen und Anforderungen. Da die Beteiligten einer Entwicklung mit dem Gegenstand der Betrachtung und den Ausdrucks- und Sichtweisen der übrigen Beteiligten meist sehr unterschiedlich vertraut sind, muss eine allen Beteiligten zugängliche Verständigungsbasis geschaffen werden, von der aus dann die gemeinsame Theoriebildung im Sinne der *Anforderungsklä rung* begonnen werden kann. Diesen gemeinsamen Verständigungsprozess bezeichnet Andelfinger als diskursive Systemanalyse [Andelfinger 1997, S. 201 ff]. Zur Schaffung einer gemeinsamen Verständigungsbasis schlägt Andelfinger heuristische Leitfragen zur Analyse des Situationsprofils vor, mit dessen Hilfe die Situation aus unterschiedlichen Sichten (Problemcharakter, Situationsdefinition, Interaktionskultur, Interessen, Werten, Erfolgskriterien, Akteuren, Rollen, Methoden, Repräsentationsformen, materielle Ressourcen und Geltungsanspruch) untersucht werden soll, um so eine von allen Beteiligten verstandene Ausgangsbasis zu schaffen.

3.4 Analyse der vorgestellten Ansätze

Nachfolgend wird zunächst ein Überblick über die oben vorgestellten Ansätze zur methodischen Unterstützung der *Anforderungsklä rung* gegeben. Dazu werden die Ansätze hinsichtlich unterschiedlicher Aspekte zusammengefasst, die aus den Ergebnissen aus Kapitel 2 in Verbindung mit Erkenntnissen aus der Analyse der Literatur herausgearbeitet werden konnten. Hierbei handelt es sich um Aspekte, die die Qualität einer Anforderungsklä rung maßgeblich beeinflussen [siehe auch HAVELKA 2002]. Wichtige, qualitätsbeeinflussende Aspekte der Anforderungsklä rung sind:

- **Kundeneinbindung:** Wird methodische Unterstützung zur Einbindung des *Kunden* in den Prozess der Anforderungsklä rung angeboten?
- **Weiterverarbeitung von Kundenanforderungen:** Wird methodische Unterstützung zur Weiterverarbeitung von Anforderungen angeboten?
- **Erkennen von Anforderungen:** Wird methodische Unterstützung zum *Erkennen* von *Anforderungen* angeboten?
- **Pflege und Erweiterung von Anforderungen:** Wird methodische Unterstützung zur Pflege und Erweiterung von Anforderungen angeboten?
- **Formulieren und Definieren von Anforderungen:** Wird methodische Unterstützung zum *Formulieren* und *Definieren* von *Anforderungen* angeboten? (siehe Kapitel 2.3.2)
- **Gewichtung von Anforderungen:** Wird methodische Unterstützung zur Gewichtung von Anforderungen angeboten?
- **Strukturierung von Anforderungen:** Wird methodische Unterstützung zum Ordnen, Clustern und Sortieren von Anforderungen angeboten?

- **Zielkonflikte und Vernetzung von Anforderungen:** Wird methodische Unterstützung für die Bearbeitung von Zielkonflikten und der Vernetzung von Anforderungen angeboten?
- **Vollständigkeit von Anforderungen:** Wird methodische Unterstützung zur Abschätzung der Vollständigkeit von Anforderungen angeboten?
- **Interdisziplinarität:** Wird methodische Unterstützung zur Handhabung der Schwierigkeiten der interdisziplinären Anforderungsermittlung angeboten?

Gleichzeitig werden diese Ansätze an den in Kapitel 2 erarbeiteten Problemen und Fehlern bei der Anforderungsklä rung gespiegelt, um deren Ursachen herzuleiten. Dazu wird versucht, die aufgedeckten Fehlerklassen soweit möglich den Aspekten der Anforderungsklä rung zuzuordnen. Diese Auswertung soll helfen Handlungsschwerpunkte für die Entwicklung einer verbesserten methodischen Unterstützung aufzudecken.

Zusätzlich wurden die in den vorgestellten Ansätzen aufgegriffenen Methoden zusammengefasst und hinsichtlich ihres Wirkens in der Anforderungsklä rung untersucht.

3.4.1 Ursachenanalyse

Kundeneinbindung

Eine *Anforderung* hat immer eine *Quelle* und einen *Kanal*. Unter *Quelle* ist die Herkunft einer Anforderung, unter *Kanal* der Anlass oder Weg, den eine Anforderung zum Entwickler nimmt, zu verstehen. Je individueller und neuer ein Produkt ist, desto weniger Quellen und Kanäle stehen zur Verfügung [DEIFEL 2001, S. 8]. Also genau in dem Entwicklungsumfeld, mit dem sich die vorliegende Arbeit befasst, gibt es nur wenige sich dem Entwickler direkt erschließende Quellen für Anforderungen. Eine der bedeutendsten ist der *Kunde*, der beste und effizienteste Kanal ist der direkte Kontakt mit dem Kunden als *Experten*.

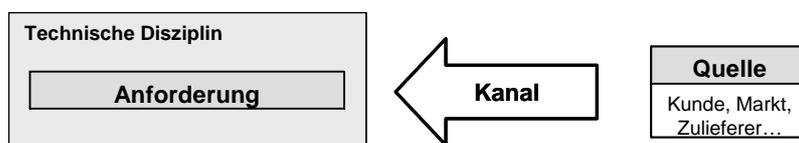


Abb. 3-15: Der Entwickler (technische Disziplin) klärt Anforderungen, die aus einer Quelle über einen Kanal gespeist wird.

In fast allen vorgestellten Ansätzen wird die Bedeutung des Kunden für eine erfolgreiche Produktentwicklung hervorgehoben. Vor allem die Ansätze in den umfassenden Modellen fördern und fordern die Einbindung des Kunden in die Anforderungsklä rung. Diese Einbindung beschränkt sich aber meist auf die frühen Phasen der Produktentwicklung. Eine Einbindung in den weiteren Produkterstellungsprozess ist nicht vorgesehen. Die Form der Einbindung erstreckt sich von einem reinen Kontakt zum Marketing zur Erfassung der

Kundenbedürfnisse bis hinzu intensiver Diskussion der Kundenbedürfnisse zwischen Kunden, Marketingfachleuten und Entwicklern. Letztere ist vor allem bei den marketingorientierten Ansätzen zu finden [zum Beispiel URBAN & HAUSER 1993]. Am deutlichsten wird dies am Ansatz von Ulrich & Eppinger, die die intensive Kommunikation von Kunde und Entwickler fordern, da in ihren Augen Innovation nur dann möglich ist und Kundenwünsche nur dann vollständig getroffen werden können, wenn Entwickler und Kunde miteinander interagieren und der Entwickler die Einsatzumgebung des Produkts erlebt [ULRICH & EPPINGER 2003, S. 55]. Letzteres bezieht sich vor allem auf Massenkonsumgüter und ist bei individuelleren Neuentwicklungen wohl nur schwer umzusetzen. Eine entwicklungsbegleitende Einbindung des Kunden wird nur in geringem Maße gefordert und gefördert.

Wie wichtig diese Form der Kundeneinbindung ist, zeigt eine Studie über die Einflüsse auf die Qualität der Anforderungsklä rung von Havelka [HAVELKA 2002, S. 223]: Von entscheidender Bedeutung sind das technische Wissen des Entwicklers im betrachteten Bereich, das domänenspezifische Wissen (auf der Anwenderseite), die Kommunikationsfähigkeit aller Beteiligten und das Systemverständnis des Anwenders. Von eher untergeordneter Bedeutung ist interessanterweise die Komplexität der Anwendung. Aus dieser Erkenntnis leitet Havelka ab, dass der Anwender beziehungsweise Kunde als Mitglied oder Berater des Entwicklungsteams eingebunden sein muss, besonders wenn er das domänenspezifische Wissen auf der Anwenderseite besitzt. Dass dieser Kontakt nicht nur bei Entwicklungsbeginn, sondern auch im weiteren Prozess aufrecht erhalten werden muss, bestätigt eine Studie von Hummel [HUMMEL ET AL. 2000], in der nachgewiesen wird, dass bei interdisziplinären Entwicklungen Anforderungen auch im weiteren Verlauf der Entwicklung nur mit allen Beteiligten geklä rt werden können.

Die Folgen einer zu geringen Einbindung des Kunden in den gesamten Entwicklungsprozess können an den eingangs beschriebenen Beispielen (Kapitel 2.1) abgelesen werden: Die falsche Materialwahl beziehungsweise die falschen elektrischen Anschlusswerte wären bei einer intensiveren Kundeneinbindung wahrscheinlich nicht aufgetreten.

Weiterverarbeitung von Kundenanforderungen

Zur Weiterverarbeitung von Kundenanforderungen in *technische Anforderungen* greifen die meisten Autoren auf das House of Quality aus dem *QFD* zurück. Dessen Anwendung wird im Entwicklungsprozess deterministisch zwischen die Erhebung der Kundenanforderungen und weiteren Konkretisierungsschritten der Anforderungen platziert. Die für die Gegenüberstellung im House of Quality notwendigen technischen Anforderungen beziehungsweise Produktmerkmale können aber gerade bei Neuentwicklungen meist nicht aus vorherigen Projekten übernommen werden. Wie nun aber die technischen Anforderungen aus den Kundenanforderungen abgeleitet werden sollen, wird in den wenigsten Ansätzen beleuchtet geschweige denn methodisch unterstützt. Vorschläge von Danner, wie zum Beispiel die *Methode* zur strukturierten Umformulierung von wörtlichen Kundenaussagen im Team nach King [KING 1994, S. 86], scheinen in einem hochgradig interdisziplinären Umfeld ungeeignet, besonders da auch diese Vorgehensweise suggeriert, dass aus einer Kundenanforderung jeweils eine technische Anforderung resultieren würde

[siehe CROSS 1996]. Dieser Ansatz berücksichtigt nicht ausreichend, dass dieses Ableiten der technischen Anforderungen aus Kundenanforderungen kein direkter Prozess ist, sondern einer Kundenanforderung meist zu mehreren technischen Beschreibungen weiterverarbeitet werden kann [ULRICH & EPPINGER 2003, S. 75]. Dies ist leicht an dem in Kapitel 2.1.1 angeführten Beispiel zu zeigen: Eine vom Kunden geäußerte Kernanforderung war „die Erzeugung eines pulsatilen Flusses“. Aus dieser Kundenanforderung konnte durch zahlreiche Gespräche und Recherchen eine Vielzahl von technischen Anforderungen abgeleitet werden. Erst dann konnten diese beiden Anforderungsarten über eine Matrix miteinander verknüpft werden.

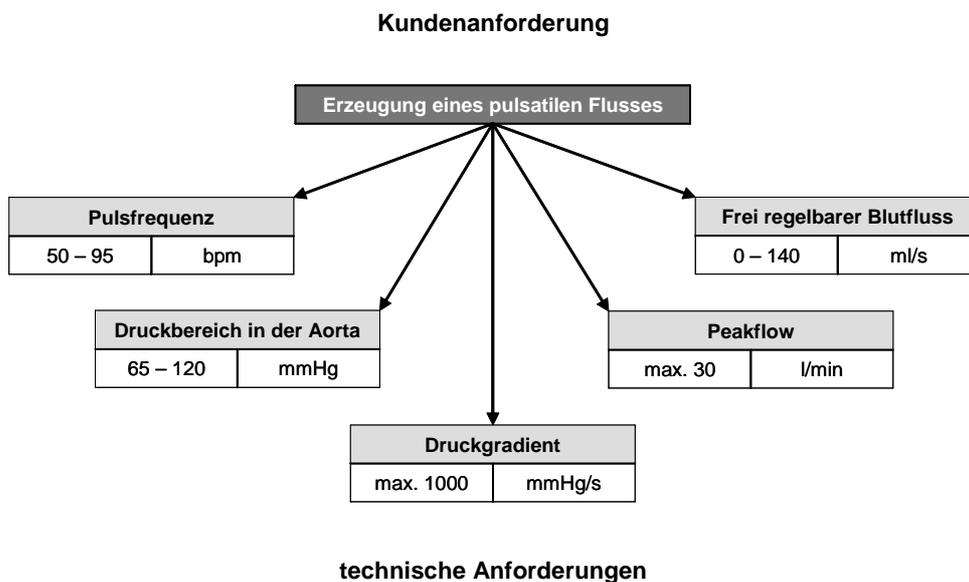


Abb. 3-16: Beispiel für die Weiterverarbeitung einer Kundenanforderung zu technischen Anforderungen

Durch die derzeit ungenügende Unterstützung beim Weiterverarbeiten von Kundenanforderungen werden Fehler in der Anforderungsklä rung begünstigt.

Erkennen von Anforderungen

Vor allem das frühzeitige *Erkennen* von *Anforderungen* ist ein in der vorgestellten Entwicklungsmethodik viel diskutierter zentraler Aspekt der Anforderungsklä rung. Um Anforderungen über die direkt aus den *expliziten* Kundenanforderungen (siehe Abb. 3-17) ableitbaren hinaus zu erkennen, greifen die meisten Ansätze auf unterschiedliche Formen der Checkliste [zum Beispiel EHRENSPIEL 2003; PAHL ET AL. 2003] zurück. Die Anordnung und Darstellung der Inhalte dieser Listen variiert, der Inhalt selbst und die Funktionsweise bleiben dieselbe. Der Einsatz von Checklisten wird in neueren Ansätzen durch Recherchen und den Einsatz von Konkurrenzanalysen (Benchmarking, Reverse Engineering etc.) ergänzt [zum Beispiel LINDEMANN 2005, S. 87 ff]. Einen umfassenderen Weg beschreitet Franke [FRANKE 1975] mit der Suchmatrix. Haben die Checklisten bisher vor allem einen produktorientierten Blick auf die Anforderungen ermöglicht, wird mit der Suchmatrix

zusätzlich der Lebenslauf des Produkts beleuchtet. Da Franke im Entwicklungsprozess ein mehrmaliges Durchlaufen der Suchmatrix vorsieht, um Anforderungen detaillieren und ergänzen zu können, gestaltet sich die Anwendung entsprechend aufwändig. Ein Teil der rechnerunterstützten Ansätze zeigt keine neuen Wege zur Erkennung von Anforderungen auf, sondern baut auf den bekannten Methoden auf [AHRENS 2000; BARRENSCHEN ET AL. 1989].

Die genannten Methoden zum Erkennen von Anforderungen (Checklisten, Suchmatrix, Funktionsdarstellungen etc.) helfen dem Entwickler besonders in den frühen Entwicklungsphasen durch Abstraktion, gezielte Fragen, Orientierung am Produktlebenszyklus, Assoziationen, Funktionsbetrachtung etc. ein Problem zu zerlegen, die Teilprobleme zu erkennen und in Form von meist globalen Anforderungen zu benennen. Solange der Entwickler sich in seinem eigenen fachlichen Umfeld bewegt, sind die genannten Methoden gut geeignet, um Anforderungen zu klären. Erfahrungen vor allem in medizintechnischen Projekten (siehe Kapitel 2.1.1) haben aber gezeigt, dass diesen Methoden enge Grenzen gesetzt sind. Sobald man sein eigenes thematisches Umfeld verlässt, können Assoziation und ähnliche Methoden allein nicht mehr zum Ziel führen [LINDEMANN ET AL. 2003]: Anforderungen werden gar nicht erst erkannt, da man als Entwickler zu wenig Einblick in die fremde Disziplin hat, beziehungsweise die Notwendigkeit einer bestimmten Anforderung zwar erkannt wird, das korrekte Formulieren ist aber aus ähnlichen Gründen nur bedingt möglich. Die Hauptursache ist wohl darin zu sehen, dass die vorhandenen Methoden vor allem den inneren Dialog des Ingenieurs unterstützen können. In den in Kapitel 2 beschriebenen interdisziplinären Projekten reichte dieser innere Dialog nicht aus, da das Wissen auf verschiedene Personen verteilt war, die auch noch unterschiedliche Terminologien benutzten. Verschärft wird diese Problematik, wenn die *Anforderungen* dem Kunden immer weniger bewusst also *implizit* sind. Solche Anforderungen können zum Beispiel durch die richtigen Assoziationen in Form von Begriffen angeregt werden. Die dafür notwendige Produktbezogenheit und Granularität für die späteren Entwicklungsphasen wird derzeit von keiner der vorgestellten allgemeingültigen Assoziationslisten (Checkliste etc.) geboten.

In der Entwicklungsmethodik werden als weitere Möglichkeiten zur Anforderungsklärun Recherchen, Interviews, Benchmarkings etc. [zum Beispiel LINDEMANN 2005, S. 84 ff] genannt. Aber auch diese können in der beschriebenen Situation nur bedingt zum Ziel führen. Denn für zielgerichtete Fragen beziehungsweise Recherchen muss man eine konkrete Frage an die *Quelle* wie zum Beispiel die Literatur oder den Kunden stellen können. Aber wie kommt man auf diese Fragen? Fragen ergeben sich, wenn man ein Problem erkannt hat. Wird man aber die Probleme ohne den fachlichen Hintergrund erkennen? Hinzu kommt, dass Quellen wie Literatur, Internet etc. meist nicht nach technischen Gesichtspunkten verfasst und geordnet sind und der Zufall bei der Suche nach Informationen eine dementsprechend große Rolle spielt.

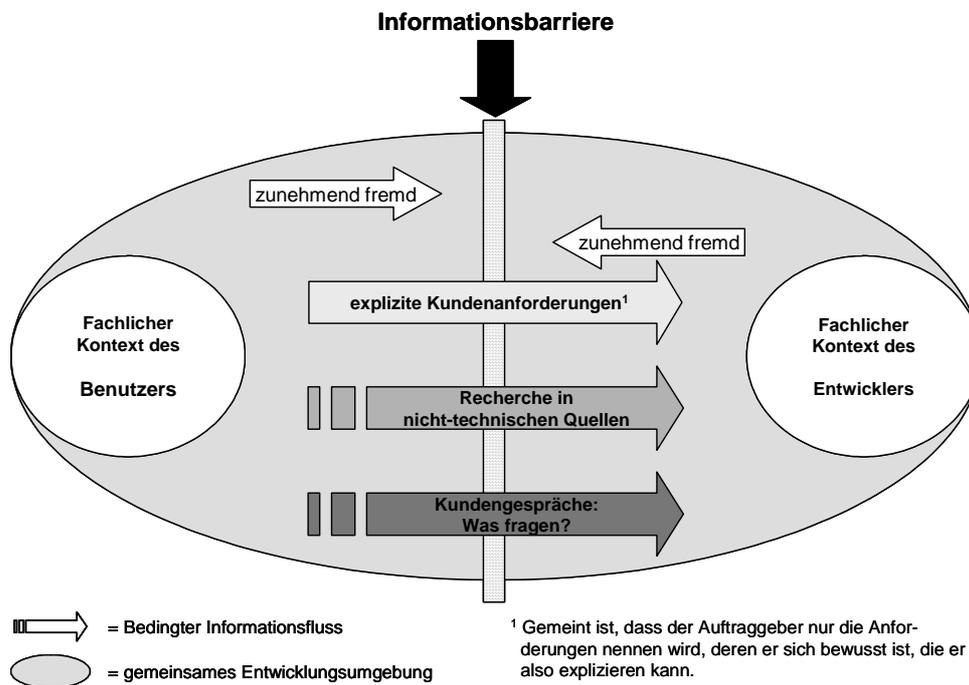


Abb. 3-17: Informationssammlung in interdisziplinären Projekten [in Anlehnung an ANDELFINGER 1997, S. 203]

Eine Studie von Shefelbine [SHEFELBINE 1998] stützt diese Ergebnisse: Jede der evaluierten Methoden hat ihre Stärken (wie zum Beispiel Strukturieren der Anforderungen, Abstimmen von Zielkonflikten etc.), sie unterstützen aber nur unzureichend das Erkennen von Anforderungen. Alexander [ALEXANDER 1999] hat im Rahmen einer Umfrage bei Entwicklern medizintechnischer Geräte herausgefunden, dass fast drei viertel der Befragten den Bedarf nach mehr Unterstützung beim Erkennen von Anforderungen äußerten.

Die geschilderten Ergebnisse erinnern stark an das von Lamnek [LAMNEK 1993, S.11] formulierte „Primat der Methode“, demzufolge das einmal ausgebildete methodische Instrumentarium immer wieder unverändert zur Verfolgung der unterschiedlichsten Fragestellungen herangezogen wird, ohne auf die Eigenarten der Aufgabenstellung genügend Rücksicht zu nehmen. Damit tritt in den Augen Lamneks die zu untersuchende Wirklichkeit unter die Maßgabe der vorhandenen Methode. Dieses verhindert die angemessene Erfassung der Sachverhalte. Zum Beispiel ist die Checkliste oder Hauptmerkmalsliste, soweit dies zu recherchieren war, erstmalig von Pahl & Beitz [PAHL & BEITZ 1972] vorgestellt worden und wird seitdem mehr oder weniger unverändert in verschiedenen Darstellungsformen zum Erkennen von Anforderungen empfohlen. Der steigenden Produkt- und Prozesskomplexität ist diese Methode aber nicht gefolgt, so dass daraus Fehler resultieren können. In diesem Zusammenhang ist auch die Kritik von Gröber zu sehen, dass diese Methoden keine problembeziehungswise bearbeiterspezifischen Aspekte berücksichtigen und derzeit zu sehr auf technische Belange (vor allem Checklisten u. ä.) [GRÖBER 1992, S. 19] ausgerichtet sind. Relevante Punkte anderer beteiligter Disziplinen fehlen, so dass die Methoden nicht alleine

zum Erfolg führen können. Er versucht dies mit seinem Ansatz der situationsabhängigen Methodenauswahl zu kompensieren.

Einen vollkommen neuen Ansatz zur Erkennung von Anforderungen stellen hingegen Humpert [HUMPERT 1995] und in ähnlicher Weise Kruse [KRUSE 1996] vor. Humpert ordnet die Anforderungen in einem hierarchischen Anforderungsmodell an. Dieses Anforderungsmodell verknüpft er mit weiteren Partialmodellen, zum Beispiel der Baustruktur. Durch die prozessbegleitende iterative Erweiterung der Baustruktur ergeben sich wiederum Rückschlüsse auf das Anforderungsmodell, so dass neue Anforderungen erkannt werden können. Das Anforderungsmodell hat seinerseits wieder Auswirkungen auf die Baustruktur. Von einigen Autoren wird im Zusammenhang der Erkennung von Anforderungen auch die Problematik der *impliziten Anforderungen* angesprochen [BREIING & FLEMMING 1993, S. 14, LINDEMANN 2005, S. 84, PAHL ET AL. 2003, S. 191]. Breiing & Flemming bezeichnen mit impliziten Anforderungen die aus den technischen und naturwissenschaftlichen Gesetzmäßigkeiten ableitbaren Anforderungen, die nicht quantifizierbar sind und nicht in eine Anforderungsliste gehören. Lindemann und Pahl et al. vertreten die Ansicht, dass implizite Anforderungen die unausgesprochen vom Kunden erwarteten Anforderungen sind, die für den Erfolg des Produkts von entscheidender Bedeutung sind. Dieser Definition schließt sich der Verfasser an.

Pflege und Erweiterung von Anforderungen

Die Pflege und Erweiterung von Anforderungen ist stark mit dem Erkennen von Anforderungen vernetzt, da das Erkennen ein Prozess ist, der zu Entwicklungsbeginn stattfindet und im Rahmen der Pflege und Erweiterung immer wieder stattfinden muss. Es wirken deshalb ähnliche Mechanismen wie beim Erkennen, so dass die vorangegangenen Ausführungen hier gültig sind und nur um einige Aspekte ergänzt werden sollen.

Die Notwendigkeit einer kontinuierlichen prozessbegleitenden Pflege und Erweiterung von Anforderungen wird von allen Autoren angesprochen: Einmal gefundenen Anforderungen müssen detailliert, verändert oder auf unterschiedliche Elemente (zum Beispiel Baugruppen) heruntergebrochen werden. Darüber hinaus ist eine konsequente Anforderungsverfolgung nötig, um Anforderungen konsequent umsetzen zu können und gleichzeitig Inkonsistenzen und redundante Anforderungsinformationen zu vermeiden [BENDER 2005, S. 118]. Dazu sind die angebotenen Methoden aufgrund fehlender Systematisierung aber nur bedingt geeignet. So betrachten Pahl et al. [PAHL ET AL. 2003], Ehrlenspiel [EHRENSPIEL 2003] etc. das Produkt in seiner Gesamtheit und wenden darauf zum Beispiel eine Checkliste an. Mit diesen Methoden kann aber nur das abgefragt werden, was auch in einer solchen Liste enthalten ist²⁰, und jeder Punkt der Liste wird zunächst nur auf das Produkt in seiner Gesamtheit angewendet. Untergliedert man ein Produkt in seine Elemente und Subelemente und wendet die Checklisten wiederum an, können sich dort weitere Anforderungen verbergen, die vorher nicht erkannt worden sind. Diese Betrachtung der einzelnen Elemente eines Produkts ist zwar mit den vorhandenen Methoden möglich, wird aber von der Entwicklungsmethodik weder

²⁰ Hiermit ist die fehlende Berücksichtigung problemspezifischer Aspekte vor allem bei Neuentwicklungen gemeint [siehe auch GRÖßER 1992, S. 19].

gefordert noch systematisiert. Auf diesen Umstand sind zum Beispiel die in Kapitel 2 geschilderten Probleme bei der Entwicklung eines Druckers zurückzuführen.

Eine der Ausnahmen bildet das Vorgehen nach Ulrich & Eppinger [ULRICH & EPPINGER 2003, S. 88 ff]: Nach einer ersten Konzeptphase ist eine Reflexion über die Anforderungen vorgesehen, um diese zu erweitern und endgültig festzulegen.

Die umfangreichsten Ansätze zur Pflege und Erweiterung von Anforderungen stellen aber die Ansätze von Größer, Humpert und Kruse dar. Jeder ermöglicht es auf seine Art, die Anforderungsklä rung prozessbegleitend zu unterstützen. Größer [GRÖßER 1992] kann durch die situations- und bearbeiterspezifische Methodenauswahl dem Entwickler zu jedem Zeitpunkt der Entwicklung die für ihn geeignete methodische Unterstützung anbieten, um die Anforderungsklä rung auch in den späteren Phasen der Entwicklung aktiv zu betreiben. Das grundsätzliche Vorgehen von Kruse und Humpert zur Anforderungsklä rung, also das Vernetzen der Anforderungen mit Elementen [KRUSE 1996] beziehungsweise mit anderen Partialmodellen [HUMPERT 1995], ist ebenfalls für eine kontinuierliche Anforderungsklä rung geeignet. Durch den iterativen Aufbau und die Vernetzung von Anforderungen mit Elementen/Partialmodellen gibt es immer wieder bidirektionale Rückschlüsse, die zu Veränderungen oder zu neuen Anforderungen führen können. Aber auch dieses iterative Weiterentwickeln ist nur bedingt geeignet, vollkommen neue, bisher unbekannte *Anforderungsfelder* zu eröffnen.

Die vorgestellten Methoden sind gut geeignet, um die wesentlichen Randbedingungen und globalen Anforderungen eines Produkts zu erfassen. Sie können aber nur schwer der zunehmenden Detaillierung des Produkts und dem damit wachsenden Kenntnisstand über das Produkt folgen, um die dann aufkommenden neuen, oftmals lösungsbezogenen Anforderungen systematisch zu erkennen beziehungsweise eine notwendige Detaillierung oder Änderung vorhandener Anforderungen [siehe GIAPOLIS ET AL. 1995, S. 478] zu erkennen und vorzunehmen. Darauf ist wohl auch die in Kapitel 2.3.1 herausgearbeitete abnehmende Intensität der Anforderungsklä rung im Projektverlauf zurückzuführen.

Formulieren und Definieren von Anforderungen

Die Qualität von Anforderungssammlungen hängt natürlich nicht nur davon ab, Anforderungen möglichst vollständig und richtig zu erkennen, sondern maßgeblich auch davon, die *Anforderungen* eindeutig und richtig zu *formulieren* und zu *definieren*. Mangelhaft formulierte/definierte Anforderungen können zu Missverständnissen und somit zu unterschiedlicher Interpretation der Anforderungen und der Entwicklungsaufgabe führen [siehe auch ALMEFELT 2003].

Um dies zu vermeiden, gibt die Entwicklungsmethodik „Regeln“ (Lösungsneutralität, Eindeutigkeit, positive Formulierung etc.) für das Formulieren von Anforderungen vor [LINDEMANN 2005, S. 97, CROSS 1996, S. 77 ff, ULRICH & EPPINGER 2003, S. 72 ff]. Auch findet man unterschiedliche Hinweise für die Darstellung der *Ausprägungen* von Anforderungen [EHRENSPIEL 2003, S. 348]. Trotzdem werden das Formulieren und Definieren von Anforderungen nicht ausreichend unterstützt [AHRENS 2000, S. 123], so dass auch hier immer wieder Probleme festgestellt werden. Als Beispiel seien hier die fehlerhafte

Energieversorgung oder die ungenügende Steuerung für den in Kapitel 2.1.1 beschriebenen Pulsator der Herz-Lungen-Maschine angeführt.

Eine Anforderung wird in der Entwicklungsmethodik als Eigenschaft verstanden, die möglichst zu quantifizieren ist. Viele Anforderungen sind aber gar nicht quantifizierbar, sondern können nur qualitativ näher beschrieben werden. Gerade diese qualitativen Beschreibungen werden oft mit in die eigentliche Beschreibung der Anforderung formuliert, da in Anforderungslisten hierfür kein entsprechendes Feld zu finden ist [siehe zum Beispiel LINDEMANN 2005, S. 96, PAHL ET AL. 2003, S. 197]. Dies kann wiederum zu Missverständnissen, Fehlinterpretationen etc. führen. Auch können viele Anforderungen, selbst wenn sie sehr früh erkannt werden, zu Beginn gar nicht oder nur qualitativ und erst zu einem späteren Zeitpunkt quantitativ beschrieben werden [SALUSTRI & PARMAR 2004]. Ebenso kommt es vor, dass deren qualitative Beschreibung von unterschiedlichen Quellen beeinflusst wird und erst später im Entwicklungsverlauf die endgültige Beschreibung festgelegt wird. Dies wird von den bekannten Methoden nicht ausreichend unterstützt.

Dies und die daraus resultierenden Probleme sind vor allem darauf zurückzuführen, dass nicht stringent zwischen dem *Merkmal* und der *Ausprägung* einer *Anforderung* getrennt wird (siehe hierzu Kapitel 2.3.2). Das hat zur Folge, dass natürlich auch nicht zwischen den zwei Schritten „Anforderung formulieren“ und „Anforderung definieren“ unterschieden werden kann, deren Durchführung im Prozess zeitlich weit voneinander getrennt sein kann.

Anforderung	
Merkmal	Ausprägung
wird formuliert	wird definiert

Abb. 3-18: Aufbau einer Anforderung

Einzig Ulrich & Eppinger [ULRICH & EPPINGER 2003, S. 79] differenzieren beim Klären einer Anforderung deutlich zwischen dem Merkmal und der Ausprägung einer Anforderung und bieten hierzu auch ein entsprechendes Vorgehen an. Auch sie betonen, dass es notwendig sei, möglichst alle Anforderungen zu quantifizieren. Dabei gehen sie aber zweistufig vor: Im ersten Schritt werden die Anforderungen mit einem Idealwert und einem gerade noch zulässigen Wert versehen. In einem späteren Schritt, wenn erste Lösungskonzepte vorliegen, wird die Ausprägung auf Basis der neuen Erkenntnisse eingegrenzt und festgelegt.

Erschwerend kommt hinzu, dass es keine Vorgaben oder Empfehlungen bezüglich der „Syntax“ eines Anforderungsmerkmals gibt. Dies wird in der Praxis sehr frei nach persönlichen Vorlieben gehandhabt und reicht von Substantiven, Kombinationen aus Verb und Substantiv bis hinzu ganzen Sätzen [siehe auch RESTREPO & CHRISTIANS 2003]. Diese unterschiedlichen Formulierungsarten bieten jedem, der die Anforderung liest und anwenden muss, teilweise sehr großen Interpretationsspielraum. Fehler sind dann vorprogrammiert.

Gewichtung von Anforderungen

Alle Autoren sind sich einig, dass eine Gewichtung notwendig ist, um Entwicklungsschwerpunkte herausarbeiten zu können. Die häufigste Form der Gewichtung ist wohl die Einteilung der Anforderungen in Forderungen und Wünsche. Doch auch bei dieser eher einfachen Form der Gewichtung kann man schon sehr deutliche Unterschiede in der Behandlung der Anforderungen finden. So fordern zum Beispiel Breiing & Flemming [BREIING & FLEMMING 1993, S. 15], dass am Ende einer Entwicklung in der Anforderungsliste keine Wünsche mehr erscheinen dürfen. Wünsche dürfen in ihren Augen nur in den frühen Phasen zur Diskussion gestellt werden, um den möglichen Leistungsumfang eines Produkts zu skizzieren. Lindemann und Roth [LINDEMANN 2005, S. 92 ff; ROTH 2000, S. 66] favorisieren eine quantifizierte und je nach Bedarf beliebig abstufbare Gewichtung, um so die Übererfüllung einzelner Anforderungen werten zu können, sofern diese nicht zwingend einen exakt vorgegebenen Wert erreichen müssen [siehe auch MONTAU 1996, S. 57]. Auf diese Weise lässt sich die Bedeutung von *Bereichsforderungen* gegeneinander werten. Kruse [KRUSE 1996, S. 106 ff] orientiert sich ebenfalls an einem solchen Ansatz. Da durch das von ihm vorgeschlagene Vorgehen die Anforderungen hierarchisch gegliedert werden, nutzt er diese Gliederung, um die Anforderungen im Sinne des Zielsystems der Nutzwertanalyse zu gewichten. Auf diese Weise erhält er auch bei komplexen Anforderungssystemen eine nachvollziehbare und normierte Anforderungsgewichtung. Kruse untergliedert seine Gewichtungsskala in Fest-, Ziel- und Wunschanforderung und empfiehlt, die Festanforderungen zum Aussortieren von Lösungen, die Ziel- und Wunschanforderungen zum Werten der übrigen Lösungen untereinander einzusetzen.

Strukturierung von Anforderungen

Die übliche Form der Strukturierung, also Gliederung von Anforderungen orientiert sich an den Hauptmerkmalen der oben bereits angesprochenen Checklisten (zum Beispiel Geometrie, Energie, Werkstoffe). Weitere Möglichkeiten der Strukturierung ergeben sich aus der Betrachtung der Anforderungen hinsichtlich der Produktlebensphasen, der Bedeutung der Anforderungen, der funktionalen Zugehörigkeit oder der Zugehörigkeit zu bestimmten Baugruppen beziehungsweise Modulen. Die Strukturierung nach funktionalen beziehungsweise modularen Gesichtspunkten wird von Humpert [HUMPERT 1995] und Kruse [KRUSE 1996] angewendet. Humpert ordnet seine Anforderungen anderen Partialmodellen, Kruse den Elementen des Systems zu, so dass darüber die Anforderungen gleichzeitig strukturiert werden können.

Sowohl die Gewichtung als auch Strukturierung von Anforderungen haben keinen direkten Einfluss auf die Qualität von Anforderungssammlungen, sind aber für ein systematisches Arbeiten von großer Bedeutung. Für einen ganzheitlichen Ansatz zur Anforderungsklärung werden beide Punkte deswegen später aufgegriffen.

Zielkonflikte und Vernetzung von Anforderungen

Hinsichtlich des Vorhandenseins, Erkennens und Verarbeitens von Zielkonflikten und der Vernetzung von Anforderungen untereinander weichen die vorgestellten Ansätze stark voneinander ab. Einige Autoren räumen erst gar nicht die Möglichkeit des Vorhandenseins einer Vernetzung ein [AHRENS 2000; GRÖGER 1990; FRANKE 1975; CROSS 1996; PAHL ET AL. 2003; URBAN & HAUSER 1993]. Andere hingegen behandeln diese Thematik sehr ausführlich. Nach Lindemann [LINDEMANN 2005, S. 89 ff] können Anforderungen doppelt genannt sein, sich neutral gegenüberstehen, sich positiv unterstützen oder sich widersprechen. Um diese Formen der Vernetzung herausarbeiten zu können, schlägt Lindemann den Einsatz von Konsistenzmatrizen, Funktionsmodellierungen oder Mind Maps beziehungsweise Wirkungsnetzen vor. Hat man die unterschiedlichen Vernetzungen aufgedeckt, ist vor allem den Widersprüchen besondere Aufmerksamkeit zu schenken: Diese Zielkonflikte müssen rechtzeitig im Prozess aufgelöst werden. Maßnahmen oder besondere Vorgehensweisen dazu werden aber nicht angeboten. Neues Potential für die Analyse der Vernetzungen von Anforderungen kann Humperts Ansatz [HUMPERT 1995] bieten. Durch den Aufbau eines hierarchischen Anforderungsmodells und dessen Vernetzung mit anderen Partialmodellen können sich ganz neue Wege für die Analyse und Bearbeitung vor allem von Zielkonflikten bieten. Diese Möglichkeiten wurden aber von Humpert nicht weiter beleuchtet. Sehr differenziert setzt sich Kruse [KRUSE 1996] mit dieser Problematik auseinander. Er unterscheidet zwischen zwei Vernetzungsarten: Strukturelle und semantische Beziehungen. Strukturelle Beziehungen sind Beziehungen zwischen Anforderungen, wobei in diesem Fall eine Anforderung einen Teil einer anderen Anforderung darstellt. Semantische Beziehungen beschreiben die Wechselwirkung zwischen Anforderungen auf gleicher oder unterschiedlicher Strukturebene. Sie können konkurrieren, sich unterstützen oder ausschließen. Durch diese Unterteilung wird der von Lindemann angesprochene Widerspruch beziehungsweise Zielkonflikt zwischen zwei Anforderungen unterteilt in die Konkurrenz und den Ausschluss von zwei Anforderungen. Mechanismen zum Umgang und Lösen von Zielkonflikten existieren auch hier nicht.

Zwar ist die Unterstützung zur Analyse der Vernetzung von Anforderungen nicht sehr umfangreich, auch fehlen in den beschriebenen Vorgehensweisen Mechanismen zum Umgang mit Zielkonflikten; die in den Beispielen (Kapitel 2.1) geschilderten Probleme bei der Anforderungsklä rung lassen sich aber darauf nicht zurückführen.

Vollständigkeit von Anforderungen

Ein grundsätzliches Dilemma der Anforderungsklä rung ist die Vollständigkeit von Anforderungen: Einerseits wäre eine absolut vollständige Anforderungsliste zu Entwicklungsbeginn zwar wünschenswert, da von vornherein genau bekannt wäre, was zu entwickeln ist. Andererseits könnte eine absolut vollständige Anforderungsliste zu Beginn einer Neuentwicklung aber auch hinderlich sein, da sie nicht lösungsneutral sein kann und somit Innovationen verhindern würde [siehe auch RESTREPO & CHRISTIANS 2003]. Real wird eine absolut vollständige Anforderungsliste wohl auch nicht möglich sein, eine Abschätzung

der Vollständigkeit am Projektende könnte aber sehr hilfreich sein, da der Entwickler dann eine höhere Sicherheit hätte alle relevanten Anforderungen erfasst zu haben.

Der Gedanke der Vollständigkeit von Anforderungen wird in der Literatur zwar verschiedentlich angesprochen, aber nicht intensiver beleuchtet beziehungsweise methodisch unterstützt. Die möglichst vollständige Klärung der Anforderungen wird zwar gefordert, die bekannten Methoden und Vorgehensweisen können dieser Forderung aber nicht nachkommen, da der Entwickler beim Einsatz zum Beispiel einer Checkliste lediglich sicher sein kann, alle Punkte der Checkliste angewendet zu haben, nicht aber, ob die Checkliste überhaupt alle relevanten Punkte enthält. Gerade bei Neuentwicklungen wird dies eher nicht der Fall sein. Hinzu kommt, dass aufgrund der oben bereits angesprochenen Betrachtung des Produkts in seiner Gesamtheit nicht sichergestellt ist, dass sich in einer untergeordneten Ebene des Produkts nicht doch eine Anforderung zu einem bereits „abgehakten“ Punkte der Checkliste befindet. Dies gilt natürlich genauso für andere Methoden wie die Suchmatrix, Frageliste oder ähnliches. Dass man mittels unterstützender Methoden wie zum Beispiel Recherchen die Vollständigkeit auch nicht überprüfen kann, liegt nahe.

Einige der vom Verfasser in der Praxis beobachteten Probleme lassen sich auf die nicht erkannte Unvollständigkeit von Anforderungslisten zurückführen. So wurde zum Beispiel bei der Entwicklung eines Manipulators für die minimal invasive Chirurgie die Fertigung eines Prototypen beauftragt, obwohl wichtige Anforderungen bezüglich der Seilzugsteuerung fehlten. Dies war aber nicht bemerkt worden [ENG 2005].

Zusammenfassend kann man sagen, dass grundsätzlich kein noch so umfassendes Verzeichnis (Checkliste etc.) ohne eine zusätzliche grundlegende Systematik ein Gefühl für die Vollständigkeit von Anforderungen liefern kann [MONTAU 1996, S. 69]. Hierfür müssen andere Mechanismen geschaffen werden.

Interdisziplinarität

Die besondere Problematik der Anforderungsklä rung in einer interdisziplinären Entwicklungsumgebung wird zwar verschiedentlich angesprochen, die methodische Unterstützung beschränkt sich aber meist auf allgemeine Empfehlungen wie das Klären der Anforderungen in einem interdisziplinär zusammengesetzten Team. Dies allein kann aber die auftretenden Probleme nicht lösen: Nicht-technischen Fachgebieten arbeiten mit anderer Begrifflichkeiten und Denkstrukturen als technische Fachgebiete und sind es meist nicht gewohnt technische Anforderungen zu formulieren. Die daraus resultierenden Kommunikationsschwierigkeiten machen eine effektive Anforderungsklä rung schwierig [siehe auch ALMEFELT ET AL. 2003]. Hinzu kommt, dass der Kunde (zum Beispiel ein Mediziner) das Wissen über Randbedingungen hat, welche die Funktionserfüllung beeinflussen, eine Vielzahl dieser dem Ingenieur meist aber nicht explizit nennen kann. Zudem sind dem Auftraggeber die Auswirkungen von fehlenden, falschen oder ungenügend geklä rten Anforderungen oftmals nicht bewusst. In dieser Situation befindet sich ein Ingenieur nun in einem Dilemma: In einer Zeit, in der es immer schwerer wird, in der eigenen Fachdisziplin auf dem Stand der Technik zu bleiben [GERHARD 1998, S. 43], muss er Anforderungen aus einem ihm fremden Fachgebiet möglichst vollständig erfassen, ohne aber über das dazu nötige (fremddisziplinäre) Wissen verfügen zu können, aus dem er die

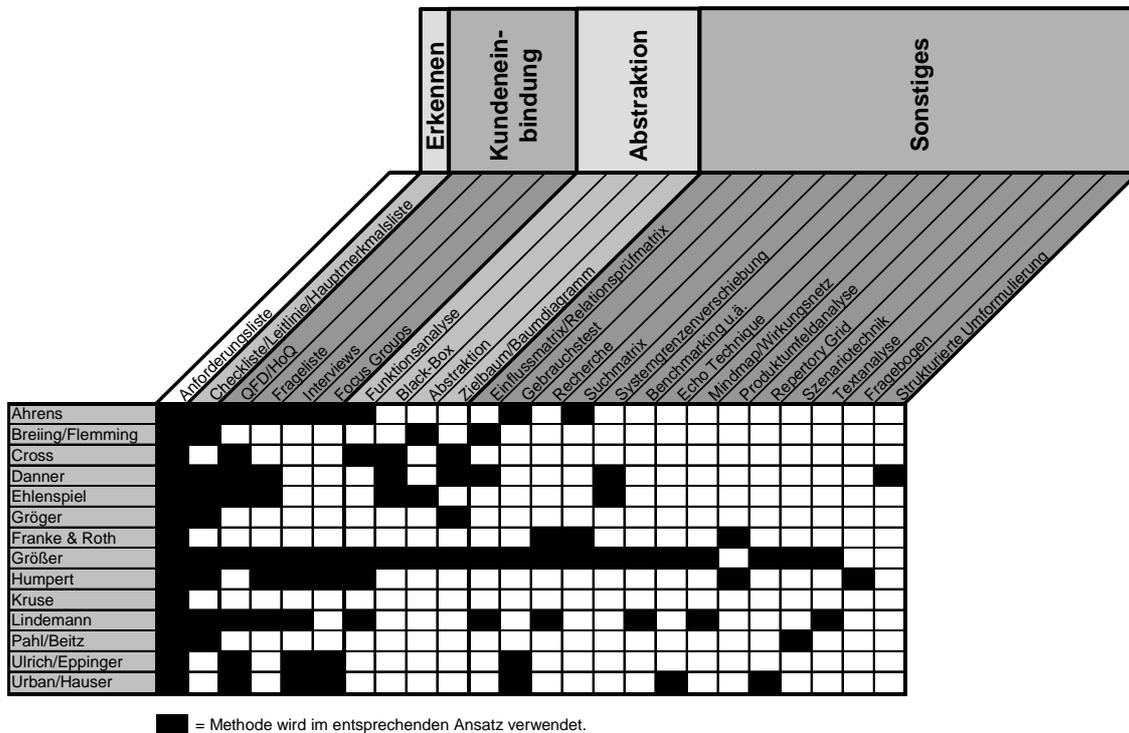
relevanten Informationen extrahieren könnte [siehe auch WARD ET AL. 2003]. Das Vorgehen von Kruse [KRUSE 1996, S. 10 ff] entsteht genau aus dieser Problematik heraus²¹. Kruse möchte mit seinem Ansatz, der entwicklungsbegleitend und von allen Domänen angewendet werden soll, ein gemeinsames, domänenübergreifendes Verständnis vom zu entwickelnden Produkt schaffen. Weitere Maßnahmen zur Verbesserung der interdisziplinären Kommunikation sind aber nicht vorhanden. In diesem Punkt geht der Ansatz von Andelfinger [ANDELFINGER 1997, S. 201 ff] wesentlich weiter, der als Grundvoraussetzung für ein gemeinsames Verständnis eine gemeinsame Verständigungsbasis sieht und hierfür heuristische Leitfragen zur Analyse des Situationsprofils entwarf, mit dessen Hilfe die Entwicklungssituation aus unterschiedlichen Perspektiven untersucht werden soll, um so eine von allen Beteiligten verstandene Ausgangsbasis zu schaffen.

Man wird verstehen, dass die beschriebenen Umstände nicht allein zu Fehlern bei der Anforderungsklä rung führen. Vielmehr hat der Grad der Interdisziplinarität als Randbedingung erheblichen Einfluss auf die unterschiedlichen Teilaspekte der Anforderungsklä rung und wird im weiteren Verlauf übergeordnet betrachtet.

3.4.2 Extraktion der eingesetzten Methoden

Die in den oben vorgestellten Ansätzen genannten Methoden wurden gesammelt und den einzelnen Ansätzen zugeordnet. Dabei wurden Checklisten, Hauptmerkmalslisten und Leitlinien, die Fragelisten, die *Methode* QFD und das daraus entnommene House of Quality (HoQ), das Mind Map und das Wirkungsnetz aufgrund ihrer Ähnlichkeit jeweils zu einem Punkt zusammengefasst. Die Methoden sind in der unten stehenden Tabelle nach der Häufigkeit ihrer Nennung sortiert.

²¹ *Kruses Ansatz entsteht aufgrund von Erfahrungen mit der Entwicklung verfahrenstechnischer Produkte.*



Tab. 3-3: Methoden zur Anforderungsklä rung

In dieser Darstellung fällt auf, dass von den Autoren bestimmte Methodenklassen bevorzugt genannt werden:

- Von allen Autoren wird die Anforderungsliste als Methode zur Anforderungsklä rung ausnahmslos genannt und auch die Empfehlungen zum Aufbau derselben entsprechen sich weitestgehend. Dies suggeriert aber einen falschen Eindruck. Eine Anforderungsliste alleine kann nicht wirklich das Klären von *Anforderungen* unterstützen, da sie keine Hinweise geben kann wo und wie man nach Anforderungen suchen kann. Die Anforderungsliste ist vielmehr ein Werkzeug zur Dokumentation der Anforderungen. Sie fordert dazu auf, eine geklä rte Anforderung in einer ganz bestimmten Art und Weise niederzuschreiben und die entsprechenden ergänzenden Informationen anzugeben.
- Beim Einsatz einer Checkliste (wird im Folgenden gleichbedeutend mit Leitlinie und Hauptmerkmalsliste verwendet) kann von einer echten Unterstützung gesprochen werden. Hier liegt ein deutlicher Schwerpunkt der vorgestellten Ansätze. Aber auch wenn die Checkliste von den meisten als gut und leicht einzusetzen beschrieben wird, wird doch mehrfach deren fehlende Allgemeingültigkeit hinsichtlich Produkt und Bearbeiter kritisiert [siehe auch GRÖBER 1992, S. 19]. Eine weitere Methodenklasse soll hier unter dem Begriff „kundeneinbindende Methoden“ (QFD, Frageliste, Interview, Focus Groups) zusammengefasst werden. Dies sind Methoden,

die allesamt geeignet sind, die Anforderungen und Bedürfnisse des Kunden zu ermitteln beziehungsweise die ermittelten Anforderungen mit technischen Anforderungen zu verknüpfen (QFD), um so Entwicklungsschwerpunkte aufdecken zu können. Durch den Einsatz dieser Methoden findet bei vielen Ansätzen eine meist intensive Einbindung des Kunden bei Entwicklungsbeginn statt. Eine entwicklungsbegleitende Einbindung des Kunden ist jedoch meist nicht vorgesehen.

- Die in der Reihung folgende Klasse „abstrakte Methoden“ fasst die Funktionsanalyse, die Blackbox, die Abstraktion und den Zielbaum beziehungsweise das Baumdiagramm zusammen. Diese Methoden sind geeignet mittels unterschiedlicher Formen der Abstraktion ein Problem zu gliedern und zu analysieren, um daraus Anforderungen ableiten zu können. Wie man aber der Übersicht entnehmen kann, wird diesen Methoden für die Anforderungsklä rung derzeit kein höherer Stellenwert eingeräumt.
- Die weiteren Methoden in der Übersicht spielen in der Gesamtheit der vorgestellten Literatur eine untergeordnete Rolle, was ihnen aber nicht deren Eignung für die Anforderungsklä rung absprechen soll. Zum Beispiel greifen viele Autoren QFD auf, um Kundenanforderungen und technische Anforderungen miteinander zu verknüpfen. Außer Danner [DANNER 1996] beschäftigt sich aber keiner ernsthaft mit der Frage, wie man denn gerade bei Neuentwicklungen die technischen Anforderungen aus den Kundenanforderungen ableitet bevor man diese einander gegenüberstellt. Danner schlägt hierzu die Methode „Strukturierte Umformulierung“ nach King [KING 1994] vor. Diese Methode wäre also auch für andere Ansätze geeignet, wird von diesen aber nicht angeführt. Des Weiteren sind einige Methoden zu finden, die in bestimmten Situationen zwar gut geeignet, aber entweder in ihren Möglichkeiten zu begrenzt (zum Beispiel der Fragebogen) oder zu aufwändig (zum Beispiel Suchmatrix nach Franke) sind.

Eine Ausnahme stellt der Ansatz von Kruse [KRUSE 1996] dar. Kruse greift praktisch auf keine vorhandenen Methoden zurück, sondern geht mit der Verknüpfung von Anforderungen mit Objekten des Produkts völlig neue Wege. In den Augen des Verfassers wäre eine Verbindung dieses Ansatzes mit weiteren Methoden durchaus denkbar (zum Beispiel der Einsatz einer Checkliste zur Assoziationsbildung bei der Konkretisierung oder Variation von Anforderungen).

Zusammenfassend kann man feststellen, dass die zur Unterstützung der Anforderungsklä rung angeführten Methoden meist die selben sind und in den letzten Jahren nur bedingt weiterentwickelt beziehungsweise ergänzt wurden. Viele Ansätze jüngerer Datums haben versucht die Anforderungsklä rung informationstechnisch zu unterstützen und dadurch zu verbessern. Wirklich neue Impulse wurden aber erst durch die Ansätze von Größer [GRÖßER 1992] und Lindemann [LINDEMANN 2005] beziehungsweise Humpert [HUMPERT 1995] und Kruse [KRUSE 1996] gegeben. Größer und Lindemann entwickeln eine vollkommen neue, freiere Sicht auf die gesamte Methodenlandschaft und schreiben nicht deterministisch vor, wann welche Methode einzusetzen ist, sondern empfehlen den situativen

und adaptiven Einsatz von Methoden. Humpert und Kruse entfernen sich von der Sicht der Anforderungen als „Tabelle“ und entwickeln eine Form von Anforderungsmodell²², das mit anderen Modellen vernetzt wird und im gesamten Entwicklungsprozess weiterentwickelt wird, um in Verbindung mit bekannten Methoden Anregungen für Anforderungen geben zu können.

3.4.3 Ableitung von Handlungsschwerpunkten

Fasst man die Ergebnisse der Ursachenanalyse und der Extraktion der eingesetzten Methoden zusammen, lassen sich daraus erste Handlungsschwerpunkte ableiten, die für die Formulierung des Forschungsbedarfs dienen können.

Die Aspekte „Gewichtung von Anforderungen“, „Definieren von Anforderungen“, „Zielkonflikte und Vernetzung von Anforderungen“, „Interdisziplinarität“ und „Vollständigkeit von Anforderungen“ werden in den betrachteten Ansätzen eher extensiv behandelt, wobei die Frage der Vollständigkeit von Anforderungen praktisch gar nicht diskutiert wird. Aspekte wie zum Beispiel die „Vernetzung und Gewichtung von Anforderungen“ werden zwar diskutiert, aber nur bedingt methodisch unterstützt.

Die Aspekte „Erkennen von Anforderungen“, „Strukturieren“, „Weiterverarbeitung von Kundenanforderungen“ und „Formulieren von Anforderungen“ werden intensiv behandelt, die methodische Unterstützung entspricht aber nicht immer dem Bedarf in der Praxis.

Auffallend sind die Aspekte „Kundeneinbindung“ und „Pflege und Erweiterung von Anforderungen“. Vor allem die Ansätze in vollständigen Modellen und hier besonders die eher marketingorientierten Ansätze von Urban & Hauser [URBAN & HAUSER 1993] und Ulrich & Eppinger [Ulrich & Eppinger 1993] befassen sich intensiv mit der Frage der Kundeneinbindung, diese Einbindung wird aber meist auf die frühen Phasen der Entwicklung und hier auf den Kontakt zum Marketing beschränkt. Die unabhängigen Ansätze hingegen kritisieren gerade diese begrenzte Form der Kundeneinbindung und fordern eine ganzheitlichere Betrachtung [zum Beispiel AHRENS 2000, S. 120 ff]. Intensiv diskutierte Lösungsansätze dazu finden sich aber nur zur Herstellung eines direkten Kontakts zwischen Kunde und Entwickler in den frühen Phasen, nicht aber für eine Aufrechterhaltung des Kontakts im weiteren Verlauf der Entwicklung. Auch in dem Forschungsprojekt „senta“ der TU Berlin wurde die Bedeutung der Kundeneinbindung erkannt; dort geht man aus ähnlichen Gründen sogar noch weiter. Man spricht von dem „Kunden als Produktdesigner“, der nicht nur Informationsquelle sondern auch Lieferant für Ideen sein soll [REINICKE 2004, S. 41]²³. Zur optimalen Nutzerintegration wird ein Modell zur Anpassung von Methoden an den

²² An dieser Stelle sei auch auf das Rechnerwerkzeug INKA (Integrierter Konstruktionsarbeitsplatz) hingewiesen, das prozessbegleitend eine interaktive Generierung von Datenobjekten ermöglicht. Diese können im Anschluss nach inhaltlichen und logischen Gesichtspunkten miteinander verknüpft werden. So können zum Beispiel technische Anforderungen mit Funktionen oder ähnlichem verknüpft werden [AMBROSY 1996, S. 158 ff].

²³ Es wird zwischen der aktiven und der passiven Einbindung unterschieden [REINICKE 2004, S. 110]. Im Sinne der vorliegenden Arbeit ist nur die aktive Kundeneinbindung gemeint.

Nutzer (des Produktes) entwickelt [REINICKE 2004, S. 120 ff], das aber für die hier geschilderte Problematik nicht geeignet scheint, da derzeit keine geeignete Methode zur Anforderungsklä rung zur Verfügung steht, die angepasst werden könnte.

Ebenso verhält es sich mit dem Aspekt „Pflege und Erweiterung von Anforderungen“. In eigentlich allen Ansätzen wird die Notwendigkeit einer intensiven Pflege und Erweiterung von Anforderungen angesprochen, ja sogar betont [zum Beispiel PAHL ET AL. 2003, S. 198, LINDEMANN 2005, S. 96 aber auch PUGH 1991, S. 44 ff]. Es gibt aber nur drei Ansätze, die diesen Aspekt sehr intensiv diskutieren [GRÖBER 1992, HUMPERT 1995, KRUSE 1996]. Außer in diesen drei Ansätzen werden Besonderheiten der prozessbegleitenden Anforderungsklä rung und Hinweise auf deren Durchführung nicht angeführt. Dass diese aber existieren und nicht ohne weiteres zu handhaben sind, zeigen die Bestrebungen von Gröber, Humpert und Kruse deutlich [siehe auch ANDERSSON 2003; RESTREPO & CHRISTIANS 2003].

Die größten Defizite in der methodischen Unterstützung mit gleichzeitig großen Auswirkungen sind also bezüglich der Aspekte „Interdisziplinarität“, „Kundeneinbindung“, „Erkennen von Anforderungen“, „Pflege und Erweiterung von Anforderungen“, und „Vollständigkeit von Anforderungen“ festzustellen. Diese Aspekte sollten dementsprechend die Schwerpunkte bezüglich der Entwicklung eines verbesserten Vorgehens bilden.

3.5 Zusammenfassung

Im Rahmen dieses Kapitels wurden unterschiedliche Ansätze zur Anforderungsklä rung vorgestellt und analysiert. Ausgehend von diesen Ansätzen und den Ergebnissen aus Kapitel 2 wurden Aspekte, die die Qualität einer Anforderungsklä rung beeinflussen und gleichzeitig funktionale Bestandteile einer Anforderungsklä rung sind, herausgearbeitet. Unter diesen Aspekten wurden die unterschiedlichen Ansätze zur Anforderungsklä rung zusammengefasst und in Bezug zu den in Kapitel 2 herausgearbeiteten Problemen bei der Anforderungsklä rung gesetzt, um die Ursachen dieser Probleme zu ergründen.

Einerseits sind bestimmte die Anforderungsklä rung beeinflussende Aspekte bisher nur am Rande beziehungsweise gar nicht von der Entwicklungsmethodik behandelt worden. Es konnte gezeigt werden, dass vor allem die Aspekte „Vollständigkeit“, „Interdisziplinarität“, „Zielkonflikte/Vernetzung“ und „Definieren von Anforderungen“ bisher eher vernachlässigt wurden. Kriterien wie zum Beispiel „Pflege und Erweiterung“ werden erst in jüngeren Ansätzen intensiver behandelt. Andererseits sind viele Aspekte bereits sehr intensiv behandelt worden und die entsprechenden Methoden konnten in der Praxis auch schon erfolgreich angewendet werden. Es konnte aber gezeigt werden, dass deren Anpassung und Anpassbarkeit an die im Rahmen dieser Arbeit betrachteten Entwicklungssituation nicht ausreichend ist. Diese Ergebnisse und die daraus abgeleiteten Handlungsschwerpunkte dienen als Grundlage für die Formulierung des Forschungsbedarfs im folgenden Kapitel.

4 Formulierung des Forschungsbedarfs

*Auf der Basis der Ergebnisse der Analyse des Standes der Forschung und der Ursachenanalyse werden Anforderungen an eine verbesserte, ganzheitlichere und interdisziplinär ausgerichtete methodische Unterstützung der Klärung von Anforderungen formuliert.*²⁴

Eine Methode zur *Anforderungskklärung*, die die in Kapitel 3.4 herausgearbeiteten Schwachstellen vermeiden soll, muss es dem Ingenieur ermöglichen, den Prozess der Anforderungskklärung zu leiten, um in Zusammenarbeit mit einem *Experten*²⁵ die *Anforderungen* möglichst vollständig, korrekt und widerspruchsfrei zu klären [LINDEMANN ET AL. 2003]. Abstrakt betrachtet ist die Entwicklung eines neuen Produkts nichts anderes, als die Lösung eines gestellten, bisher ungelösten Problems. Dies verdeutlicht zum Beispiel auch die Darstellung des Entwicklungsprozesses im Vorgehenszyklus von Ehrlenspiel [EHRENSPIEL 2003, S. 83]. Dieses Problem wird durch eine Vielzahl kleiner Teilprobleme repräsentiert, die in letzter Konsequenz bis auf die Anforderungsebene heruntergebrochen werden können: Jede Anforderung stellt also letztendlich ein kleines Teilproblem dar, das es zu lösen gilt. Man kann ein Problem aber nur lösen, entsprechend eine Anforderung klären, wenn man es erkannt und hinreichend beschrieben hat.

Deswegen ist es wichtig, dass der Ingenieur *Anforderungen* auch disziplinübergreifend erkennt, um sie im Anschluss *formulieren* und *definieren* zu können. Hierfür ist es unbedingt notwendig, sich in der fremden Disziplin entsprechend gut auszukennen oder *Experten* intensiv in die Entwicklung einzubinden [EHRENSPIEL 2003, S. 344; PAHL ET AL. 2003, S. 192, ULRICH & EPPINGER 2003, S. 55]. Von besonderer Bedeutung ist dies bei Neuentwicklungen, da hier der Informationsstand zu Beginn besonders klein und der Informationsbedarf dementsprechend groß ist [KRUSE 1996, S. 76]. Gleichzeitig sind die zur Verfügung stehenden Quellen und Kanäle meist sehr begrenzt [DEIFEL 2001, S. 8]. Im weiteren Projektverlauf muss es dem Entwickler ermöglicht werden, vorhandene Anforderungen pflegen, hinzukommende erkennen, formulieren und definieren und in die Anforderungsliste aufnehmen zu können.

²⁴ Teile dieser Darstellung sind vom Verfasser bereits veröffentlicht worden [LINDEMANN ET AL. 2003, LINDEMANN ET AL. 2005].

²⁵ Im Folgenden werden Kunden, Fachleute etc., also Personen, die als Quellen von Anforderungen dienen können unter dem Begriff „Experte“ zusammengefasst.

Wie bereits oben angemerkt, sind viele *technische Anforderungen* oder deren *Ausprägungen* lösungsabhängig²⁶ [ANDELFINGER 1997, S. 78], treten also erst im Verlauf des Entwicklungsprozesses zu Tage oder können erst spät endgültig festgelegt werden [KRUSE 1996, S. 96, PAHL ET AL. 2003, S. 198, ULRICH & EPPINGER 2003, S. 55 ff]. Um diesem Umstand gerecht zu werden, muss die Granularität der Anforderungsklä rung erhöht und eine prozessbegleitende methodische Unterstützung etabliert werden. Dafür ist es notwendig, wie bereits von Lindemann gefordert [LINDEMANN 2005], im Entwicklungsprozess „springen“ zu können und die Anforderungsklä rung iterativ wieder aufsuchen zu können. In der Realität könnte ein solches Vorgehen (siehe Abb. 4-1) wie folgt aussehen:

- Ausgehend von Kundenanforderungen werden globale technische Anforderungen an das Produkt geklärt und erweitert.
- Mit der zunehmenden Konkretisierung der Lösung(en) werden die Anforderungen ergänzt, konkretisiert und geändert.
- Jeder neuen Detaillierungsebene des Produktmodells entspricht eine neue Detaillierungsebene der Anforderungen.
- „Richtungsänderungen“ der Lösung(en) schlagen sich auch in den Inhalten der Anforderungsliste nieder, der eingeschlagene Lösungsweg beeinflusst also die Anforderungen.
- Das bedeutet, dass das Anforderungssystem mit der zunehmenden Lösungsdetaillierung wächst, detaillierter, konkreter und lösungsabhängiger wird.

Durch diese iterative Entwicklung von Produktmodell und Anforderungen²⁷ wird erreicht, dass die Konkretisierung der Anforderungen der zunehmenden Detaillierung des Produkts folgt und sich im Sinne von Humpert [HUMPERT 1995] Anforderungen und Lösung parallel entwickeln: Die Anforderungen werden immer granularer und weiten sich aus, während der Lösungsraum immer weiter eingengt wird. So kann erreicht werden, dass im Sinne von Roth [ROTH 2000, S. 71] die Anforderungen immer für den gerade folgenden Konstruktionsabschnitt vollständig sind.

²⁶ Zum Beispiel kann die Art einer Wellen-Nabenverbindung von der Wahl des Antriebs abhängen. Kann dieser erst sehr spät ausgewählt werden, kann auch die Art der Verbindung erst entsprechend spät festgelegt werden.

²⁷ Vergleiche auch die integrative Produktkonzipierung nach Gausemeier [GAUSEMEIER ET AL. 1996].

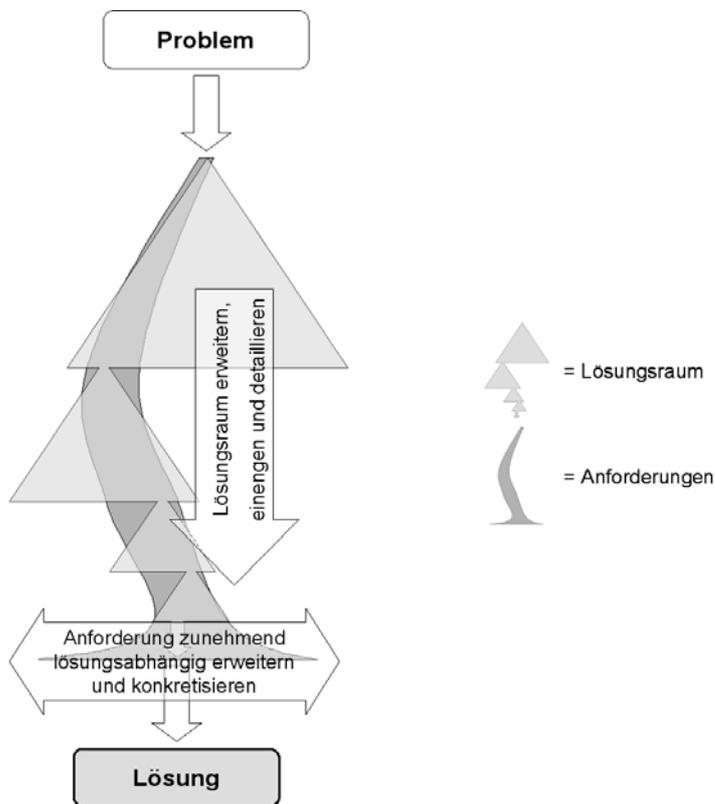


Abb. 4-1: Prozessbegleitende Anforderungsklä rung

Eingebettet in einen solchen Prozess muss das zu entwickelnde Vorgehen die folgenden Anforderungen beziehungsweise Funktionen erfüllen können:

- Systematisierung des Abfragens des zu entwickelnden Produkts nach Anforderungen unter Einbindung bewährter Methoden (zum Beispiel Checkliste, Hauptmerkmalsliste, etc.) und Durchführung des Abfragens bis zum Abschluss der Entwicklung, um so während des gesamten Entwicklungsprozesses (also parallel zum zunehmenden Detaillierungsgrad des Produkts) relevante Anforderungen erkennen, konkretisieren und präzisieren zu können.
- Für eine gute Anforderungsklä rung ist eine ausgezeichnete Kommunikation der Disziplinen von entscheidender Bedeutung [ULRICH & EPPINGER 2003, S. 54], gestaltet sich aber oftmals schwierig. Sie soll verbessert werden, indem das Vorgehen konkrete Fragen an den *Kunden* oder *Experten* der betrachteten Disziplin hervorbringt, um so wichtige oder implizite Wissensbereiche anstoßen zu können. Dadurch sollen auch *implizite Anforderungen* erfasst werden können.

- Um ein möglichst vollständiges Anforderungssystem entwickeln zu können, ist, besonders in interdisziplinären Projekten, eine intensive Kundeneinbindung über den gesamten Entwicklungsprozess notwendig [HUMMEL ET AL. 2000]. Nur so kann ein gemeinsames Verständnis von der Entwicklungsaufgabe aufgebaut werden [siehe auch KARCHER & WIRTZ 1999].
- Die in Kapitel 2.3.2 (Mikrobetrachtung) herausgearbeitete Mehrstufigkeit der Anforderungsklärung muss sich in dem Vorgehen wieder finden [siehe ULRICH & EPPINGER 2003, S. 73].
- Eine einmal aufgenommene Anforderung muss zu jedem Zeitpunkt während und nach der Entwicklung rückwärts zu ihrem Ursprung und vorwärts zu ihrer Anwendung beziehungsweise Ausführung verfolgbar sein [WARD ET AL. 2003].
- Dem Entwickler muss die Möglichkeit einer verbesserten Abschätzung des Anforderungssystems hinsichtlich seiner Vollständigkeit gegeben werden.
- Dem Entwickler müssen Hilfestellungen zum *Formulieren* und *Definieren* von *Anforderungen* gegeben werden [siehe AHRENS 2000, S. 123].
- Der Entwickler erhält Unterstützung beim Gewichten, Strukturieren und Vernetzen von Anforderungen.

5 Methode zur Unterstützung der entwicklungsbegleitenden Anforderungsklä rung

Um den Prozess des Erkennens von Anforderungen aktiv unterstützen zu können, wird zunächst nach den Ursachen von Anforderungen geforscht, um daraus einen allgemeinen Ansatz zum Erkennen von Anforderungen in interdisziplinären Entwicklungsprojekten abzuleiten. Dieser Ansatz wird dann zu einem eigenständigen, ganzheitlichen Vorgehen zur prozessbegleitenden Anforderungsklä rung ausgebaut. Um die Anwendung des Ansatzes in der Praxis zu testen und zu unterstützen wird ein Softwaretool entwickelt.²⁸

5.1 Über das Entstehen und Erkennen von Anforderungen

Kern einer jeden *Anforderungsklä rung*, das heißt erster notwendiger Schritt, ist das *Erkennen* von *Anforderungen*: Ohne dies können keine *Anforderungen* formuliert und *definiert*, gepflegt, strukturiert etc. werden. Wie aber oben gezeigt wurde, gibt es gerade hier Probleme. Es stellt sich also die Frage, wie man *technische Anforderungen* erkennen kann. Um dies zu ergründen, wird zunächst die Frage beleuchtet, woher denn Anforderungen eigentlich kommen. Damit ist nicht die Frage nach der *Quelle* (wie zum Beispiel der Markt oder der Kunde), sondern die Frage nach den direkten Ursachen von Anforderungen gemeint. Beides kann natürlich verschwimmen. Fragt man nach der Ursache für die Anforderung „Gehäusefarbe: Rot“ kann die Antwort der „Kunde“ sein, er nimmt also die Rolle der Quelle und der Ursache ein. Fragt man aber nach der Ursache für die Anforderung „Nabe der Eingangswelle: Keilwellenprofil DIN ISO 14“ kann die direkte Ursache der gewählte Antrieb sein, der ein Wellenende mit Keilwellenprofil besitzt. Der Hersteller des Antriebs kann die *Quelle* der *Anforderung* sein, ein Produktkatalog der *Kanal*. Wie dieses kleine Beispiel zeigt, haben Anforderungen Ursachen, die man ermitteln kann. Im Umkehrschluss kann dies bedeuten, dass man Anforderungen beziehungsweise *Anforderungsfelder* erkennen kann, wenn man die Ursachen von Anforderungen ermitteln kann. Um zu ergründen, ob und wie dies möglichst in Verbindung mit einem *Experten* der beteiligten Disziplin durchführbar ist, wird nachfolgend ein kleiner Exkurs in die Systemtechnik vorgenommen.

²⁸ Teile dieser Ausführungen sind vom Verfasser bereits veröffentlicht worden [LINDEMANN ET AL 2005].

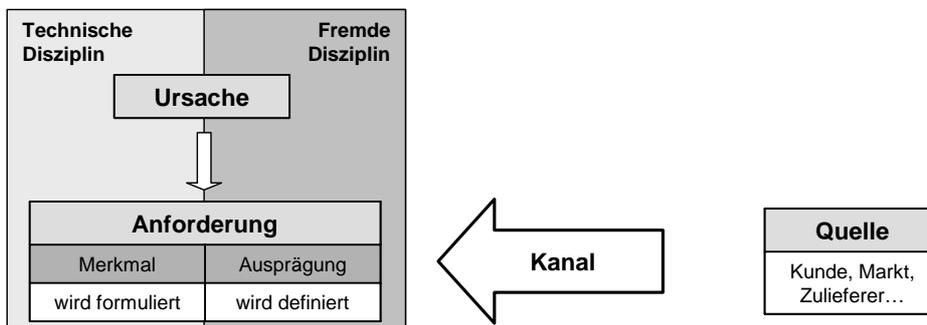


Abb. 5-1: Gemeinsame Suche der beteiligten Disziplinen nach Ursachen für Anforderung und Klärung derselben mittels Quelle und Kanal.

Im Systems Engineering [DAENZER 2002] wird ein *System* als das Zusammenspiel unterschiedlicher *Elemente*, die über *Relationen* miteinander in Verbindung stehen, verstanden. Dabei können die Relationen ganz unterschiedliche Formen annehmen, die von den jeweiligen Eigenschaften der Elemente abhängig sind: Ordnungs-, Fluss-, Funktions-, Lage-, Kräfterelationen etc. Ein System kann nie unabhängig sein, sondern steht immer mit seinem Umfeld [LINDEMANN 2005, S. 9] als übergeordnetem System in unterschiedlichster Wechselwirkung. Die Ein- und Ausgangsgrößen eines Systems beschreiben die Relation eines Systems zu seinem *Systemumfeld* und gehen über die Systemgrenzen hinweg. Mittels der Zustandsbeschreibung der Ein- und Ausgangsgrößen lässt sich die Kernfunktion eines Systems beschreiben. Alle Systembestandteile (Elemente, Relationen und das System selbst) besitzen Eigenschaften zum Beispiel räumlicher, funktionaler oder zeitlicher Natur. Mittels der Eigenschaften können Systeme und deren Zusammensetzung beschrieben und klassifiziert werden [EHRENSPIEL 2003, S. 17 f].

Wie man in Abb. 5-2 sehen kann, können die Systemelemente selbst wieder Systeme bilden, die aus Elementen und Relationen bestehen. Entscheidenden Einfluss auf die Gestalt eines Systems hat die Wahl der *Systemgrenze*. Die Systemgrenze dient der Abgrenzung gegenüber anderen Systemen, der Beschränkung und Erweiterung des betrachteten Bereichs, der Definition von Verantwortungsbereichen und der Klärung der Beziehungen zum Umfeld. Die Systemelemente können die unterschiedlichsten Relationen zueinander haben: Zum Beispiel Ordnungsbeziehungen, Flussbeziehungen (Stoff, Energie etc.), Funktions-, Lage-, Geometriebeziehungen.

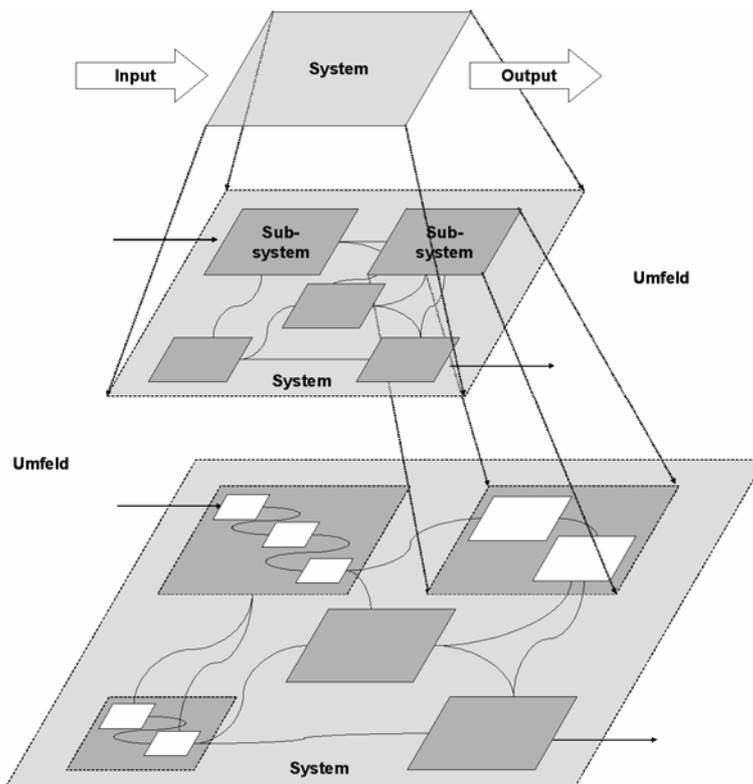


Abb. 5-2: Darstellung eines Systems in unterschiedlichen Detaillierungsstufen [in Anlehnung an EHRENSPIEL 2003, S. 17]

Grundsätzlich sind zwei unterschiedliche Vorgehensweisen zur Entwicklung einer Systemdarstellung denkbar [siehe auch EHRENSPIEL 2003, S. 16 f; STEINMEIER 1999, S. 34]:

- **Bottom up:** Ein bekanntes System wird analysiert, abstrahiert und in Form von Elementen und deren Relationen dargestellt. Die Elemente werden zu Subsystemen zusammengefasst, so dass das System bis zu einem beliebigen Grad abstrahiert werden kann. Dadurch wird das Systemverständnis erhöht und Lösungsneutralität erzeugt. Dies ist besonders zum Beispiel für Weiterentwicklungen geeignet.
- **Top Down:** Ein System wird ausgehend vom Abstrakten detailliert. Basierend auf einer Black-Box und deren In- und Output werden weitere Elemente definiert, hinzugefügt und vernetzt. So kann Schritt für Schritt ein System aufgebaut werden. Dieser Ansatz ist besonders für Neuentwicklungen geeignet, da hier das System parallel zum zunehmenden Kenntnisstand über das Produkt entwickelt werden kann.

Letzterer Ansatz bietet Potential für das in dieser Arbeit behandelte Problem. Bei einer Neuentwicklung wird ausgehend von einem Problem die Systemgrenze definiert, In- und Output ermittelt und ein erstes sehr abstraktes System aufgestellt. Mit Voranschreiten der Entwicklung müssen Systemelemente hinzugefügt und in die Zusammenhänge der bereits vorhandenen integriert werden, indem sie mit diesen in unterschiedlicher Weise in Beziehung treten.

Will man zum Beispiel einen Antrieb mit variabler Übersetzung entwickeln, hat man zunächst nur das System „Antrieb mit variabler Übersetzung“ mit dem Input „chemische Energie“ und dem Output „mechanische Energie“ und kann dies als Black-Box darstellen.

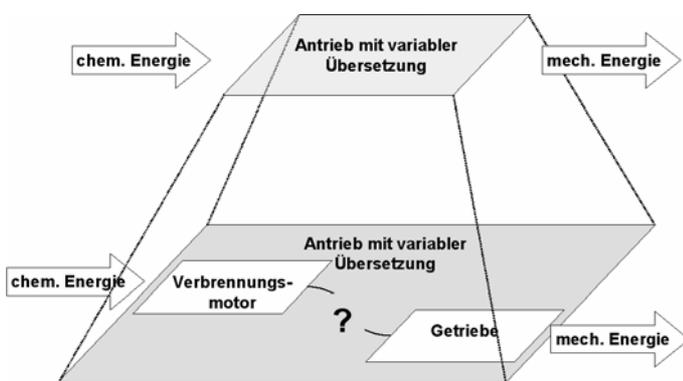


Abb. 5-3: Entwicklung des Systems „Antrieb mit variabler Übersetzung“

In der nächsten Konkretisierungsstufe wird man das System je nach gewählter Lösung in die beiden Elemente „Verbrennungsmotor“ und „Getriebe“ untergliedern. Die „chemische Energie“ ist der Input des „Verbrennungsmotors“, die „mechanische Energie“ der Output des „Getriebes“. Wie aber gestaltet sich die Relation der Elemente „Verbrennungsmotor“ und „Getriebe“ zueinander? Damit beide in Relation treten können, müssen sie „zusammenpassen“. Die Relation muss also eine Anforderung an beide Elemente stellen beziehungsweise lassen sich aus dieser Relation und ihrer Beschaffenheit (geometrische, energetisch, informationstechnisch etc.) Anforderungen an beide Elemente ableiten: zum Beispiel die Art des Flansches, der Wellen-Nabenverbindung oder der Festigkeit hinsichtlich der zu übertragenden Momente. Würde man noch ein weiteres Element „Motortemperatursensor“ hinzufügen, hätte dieses eine Beziehung zum „Verbrennungsmotor“ nicht aber zum Getriebe. Es würden also nur Anforderungen aus der Relation zum Motor folgen oder welche direkten Anforderungen könnte das Getriebe an den Sensor oder umgekehrt stellen?

Wie man an diesem kleinen Beispiel sehen kann, können Relationen aufgrund ihres Vorhandenseins und ihrer Beschaffenheit Hinweise auf Anforderungsfelder geben. Man kann also sagen, dass Anforderungen nur dort existieren können, wo mindestens zwei Elemente miteinander in Beziehung stehen. Auf diesem Grundgedanken aufbauend soll ein Vorgehen geschaffen werden, das den Entwickler anleitet, gemeinsam mit Experten

entwicklungsbegleitend ein System des Produkts aufzubauen und dessen Elemente auf das Vorhandensein von Relationen abzufragen. Kennt man die Relationen, muss man die Beschaffenheit der Relationen untersuchen und erhält daraus Hinweise auf Anforderungsfelder. Mit diesen Informationen („zwischen Element X und Element Y existiert eine Relation der Beschaffenheit Z“) können Entwickler und Kunde beziehungsweise sonstige Experten im Dialog die daraus resultierenden Anforderungen ableiten.

Ein solches Vorgehen kann die derzeit bestehende Lücke in den Methoden zur Anforderungsklä rung schließen beziehungsweise reduzieren und helfen Experten kontinuierlich einzubinden und so Anforderungen auch auf fremden Terrain zu erkennen und zu verarbeiten. Gleichzeitig käme man so der von Ward et al. [WARD ET AL. 2003] geforderte Verfolgbarkeit einer Anforderung zu ihrer Ausführung und zu ihrem Ursprung näher. Die Richtigkeit dieser Hypothese ist im Folgenden zu überprüfen.

5.2 Vorgehen zur relationalen iterativen Anforderungsklä rung

Um von dem grundlegenden Ansatz zu einem vollständig beschriebenen Vorgehen zu gelangen, sind vorab unterschiedliche Fragen zu klären:

- Wie muss ein *System* aufgebaut werden, damit es für die relationale *iterative Anforderungsklä rung* geeignet ist?
- Wie können die *Relationen* ermittelt und analysiert werden?
- Wie können aus den *Relationen Anforderungen* abgeleitet werden?

Der folgende Abschnitt setzt sich deswegen intensiv mit den notwendigen Grundlagen der relationalen iterativen Anforderungsklä rung auseinander. Da die sich aus den Fragen ableitenden Teilprozesse der relationalen iterativen Anforderungsklä rung „Entwicklung des Systems“, „Analyse der Relationen“ und „Ableiten von Anforderungen“ sehr eng miteinander verzahnt sind, werden sie zum besseren Verständnis im Folgenden zunächst unabhängig voneinander erläutert, um anschließend deren Zusammenwirken zu beschreiben.

Wo das bessere Verständnis es erfordert, werden die theoretischen Betrachtungen anhand eines kleinen Beispiels (Handstaubsauger) verdeutlicht. Dieses Beispiel diene gleichzeitig der ersten Verifikation des Ansatzes.

5.2.1 Iterativer Systemaufbau

Der iterative Systemaufbau lässt sich in mehrere Teilschritte untergliedern, die zunächst sequentiell und dann iterativ bearbeitet werden. Zum besseren Verständnis werden diese zunächst einzeln erläutert und abschließend in einem Diagramm (Abb. 5-6) zusammengefasst. Ausgangsbasis für das Vorgehen ist ein vorliegendes Problem, zum Beispiel der Auftrag zur Entwicklung eines Handstaubsaugers.

Systemanalyse

Für die relationale iterative Anforderungsklä rung ist die Wahl der *Systemgrenze* von entscheidender Bedeutung. Würde man diese genau um das zu entwickelnde Produkt legen, müsste man alle äußeren Einflüsse bereits genau kennen, um diese als In- beziehungsweise Output benennen zu können. Ansonsten könnten daraus resultierende Anforderungen verloren gehen. Da dies meist nicht der Fall sein dürfte, wird man das Gesamtsystem anfangs als Black-Box betrachten und sich auf die Systemgrenze konzentrieren [SIPHERIANU & TURNER 1998]. Im Fall der relationalen iterativen Anforderungsklä rung muss die Systemgrenze weiter gefasst werden als im klassischen Systems Engineering, da sie gleichzeitig die Anforderungsgrenze darstellt: Die Systemgrenze muss so gewählt werden, dass möglichst alles, was Ursache für Anforderungen sein kann, innerhalb dieser Grenze ist. Ansonsten müssen zumindest der exakte Einfluss beziehungsweise die Anforderung bereits bekannt sein. Dies kann zum Beispiel bedeuten, den *Kunden* als *Element* zu betrachten, da er Anforderungen an das Produkt stellt; die Kundenanforderungen wären damit Subelemente des Subsystems Kunde. Weitere mögliche Elemente können Baugruppen, -teile, Bediener, der Aufstellungsort, zu verarbeitende Medien etc. sein. Im Idealfall müsste der gesamte Produktlebenszyklus auf mögliche Elemente untersucht werden. Da aufgrund der zu erwartenden Komplexität wohl aber nicht alle Elemente berücksichtigt werden können, wird man eine sinnvolle Auswahl der für technische Anforderungen relevanten Elemente treffen müssen. Unterstützend können dabei Hilfsmittel wie die von Franke entwickelte Suchmatrix wirken [FRANKE 1975]. Ist zum Beispiel absehbar, dass die Entsorgung von großer Bedeutung ist und Einfluss auf technische Anforderungen haben könnte (zum Beispiel bei kontaminiertem Einwegmaterial im Bereich der Medizintechnik), sollte man diese in die Betrachtungen einbeziehen.

Initialisierung des Ursprungssystems²⁹

Die erste Sammlung von Elementen wird meist noch sehr abstrakt ausfallen, da der Informationsstand über das zukünftige Produkt und damit über das System noch niedrig ist. Es wäre in dieser Phase wenig sinnvoll, schon auf eine Detailebene zu gehen, die dann nicht mehr lösungsneutral wäre. Wie weit die Suche nach übergeordneten Systemelementen gefasst wird, hängt stark vom betrachteten Objekt und dessen Einsatzszenario ab.

Hat man erste Elemente gesammelt und will mit diesen Informationen das Ursprungssystem initialisieren, stellt sich die Frage, welche Art von Relationen dazu verwendet werden kann. Aufgrund ihrer besonders einfachen Handhabung bieten sich in dieser frühen Phase dafür besonders Ordnungsrelationen an, mit deren Hilfe Elemente in einer hierarchischen Struktur (in Bezug zum Beispiel auf das Produkt kann dies die Baustruktur sein) angeordnet werden können (siehe hierzu auch das Produktstrukturmodell nach Humpert [HUMPERT 1995, S. 63 ff]). Für das Ursprungssystem werden alle benötigten Elemente einem gedachten Gesamtsystem des Produkts hierarchisch untergeordnet. Das eigentliche Produkt stellt in diesem Gesamtsystem ein eigenes Element dar.

²⁹ Siehe hierzu auch die Initialisierung des Anforderungsmodells bei Humpert [HUMPERT 1995, S. 65].

Im unten stehenden Beispiel wurde anfangs zwar das Element „Garten“ identifiziert, bei der Entwicklung des Ursprungssystems aber ausgeblendet, da in Kundengesprächen ermittelt werden konnte, dass ein Einsatz außerhalb des Hauses nicht notwendig sei.

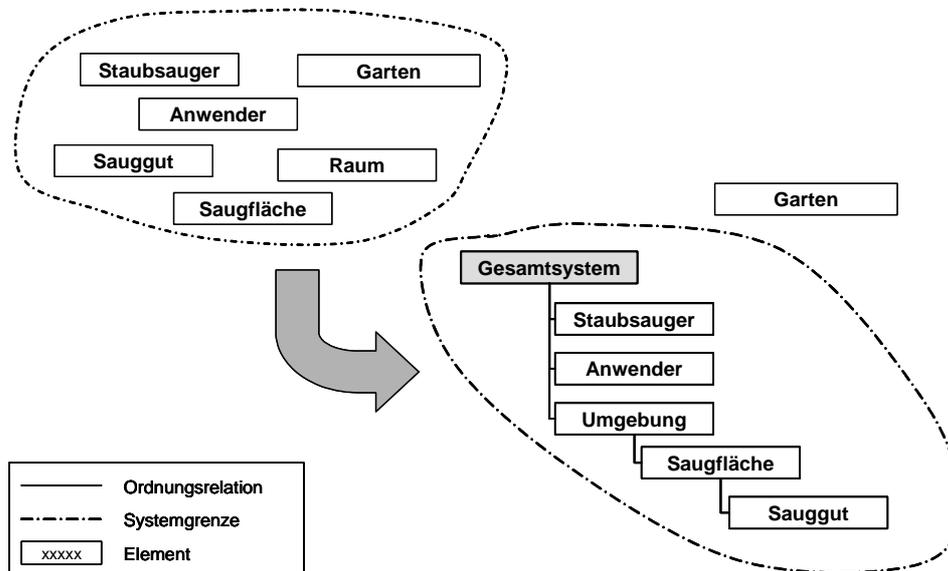


Abb. 5-4: Sammlung von Elementen und Generierung des Ursprungssystems für einen Tischstaubsauger

Systemdetaillierung und -modellierung

Im Rahmen der weiteren Entwicklungstätigkeit muss das primär initialisierte System Top Down und parallel zur zunehmenden Lösungsdetaillierung iterativ weiterentwickelt werden. Dazu ist notwendig, dass der Entwickler sich parallel mit der Suche nach Lösungen für das zukünftige Produkt beschäftigt. Die dabei gewonnenen Informationen können dann wieder in die Detaillierung des Systems einfließen.

Hat man zu Beginn einer Entwicklung nur eine sehr vage Vorstellung von dem späteren Produkt „Staubsauger“, kann man nach einer ersten lösungsorientierten Problembetrachtung eventuell schon festlegen, dass der Staubsauger zumindest aus einem „Gehäuse“, einer „Energieeinheit“ und einer „Saugeinheit“ bestehen wird. Die Einbeziehung eines *Experten* kann dabei sehr hilfreich sein. Im Beispiel konnten im Gespräch mit dem Anwender die Detaillierungen der Elemente „Anwender“, „Sauggut“ und „Saugfläche“ erarbeitet werden. Ein mögliches Ergebnis dieses Arbeitsschrittes zeigt Abb. 5-5.

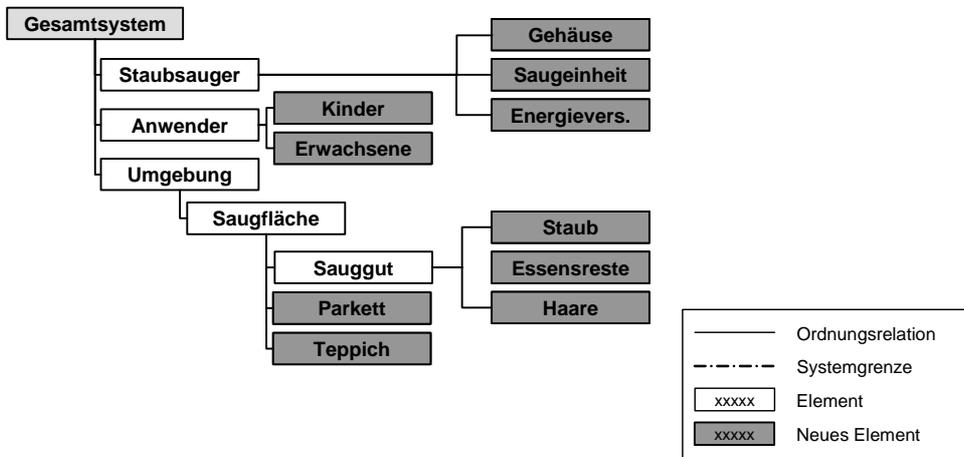


Abb. 5-5: Erweiterte Struktur des Gesamtsystems zum System „Handstaubsauger“

Dabei fällt auf, dass es zwei unterschiedliche Klassen von (Sub-)Elementen bezüglich ihres Auftretens gibt. Es existieren Elemente, die gleichzeitig auftreten: Die Subelemente „Gehäuse“, „Saugereinheit“ und „Energieversorgung“ sind für die Existenz des Elements Staubsauger gleichzeitig notwendig. Ebenso existieren Elemente, die nicht gleichzeitig auftreten können oder müssen: Die Subelemente „Kinder“ und „Erwachsene“ sind beide Elemente des Systems, treten aber nicht gleichzeitig als Anwender auf. Da für die Anforderungen letztendlich aber alle Elemente von Relevanz sind, wird auf eine Kennzeichnung dieser unterschiedlichen Zuordnung zugunsten der Übersichtlichkeit verzichtet.

Die beschriebene iterative Entwicklung des Gesamtsystems eines Produkts kann und muss während der gesamten Produktentwicklung parallel zur Lösungssuche, Lösungsauswahl und Lösungsdetaillierung fortgeführt werden, so dass bei Entwicklungsende eine vollständige systemtechnische Repräsentation des Produkts mit seinen beeinflussenden Nachbarsystemen vorliegt.

Nachfolgende Abbildung zeigt nochmals den prinzipiellen Ablauf des iterativen Systemaufbaus: Ausgehend von einer Systemanalyse und einem initial erstellten *Ursprungssystem* wird das Gesamtsystem durch die wechselweise Bearbeitung der Schritte Systemdetaillierung und Systemmodellierung fortentwickelt bis zum vollständig entwickelten System.

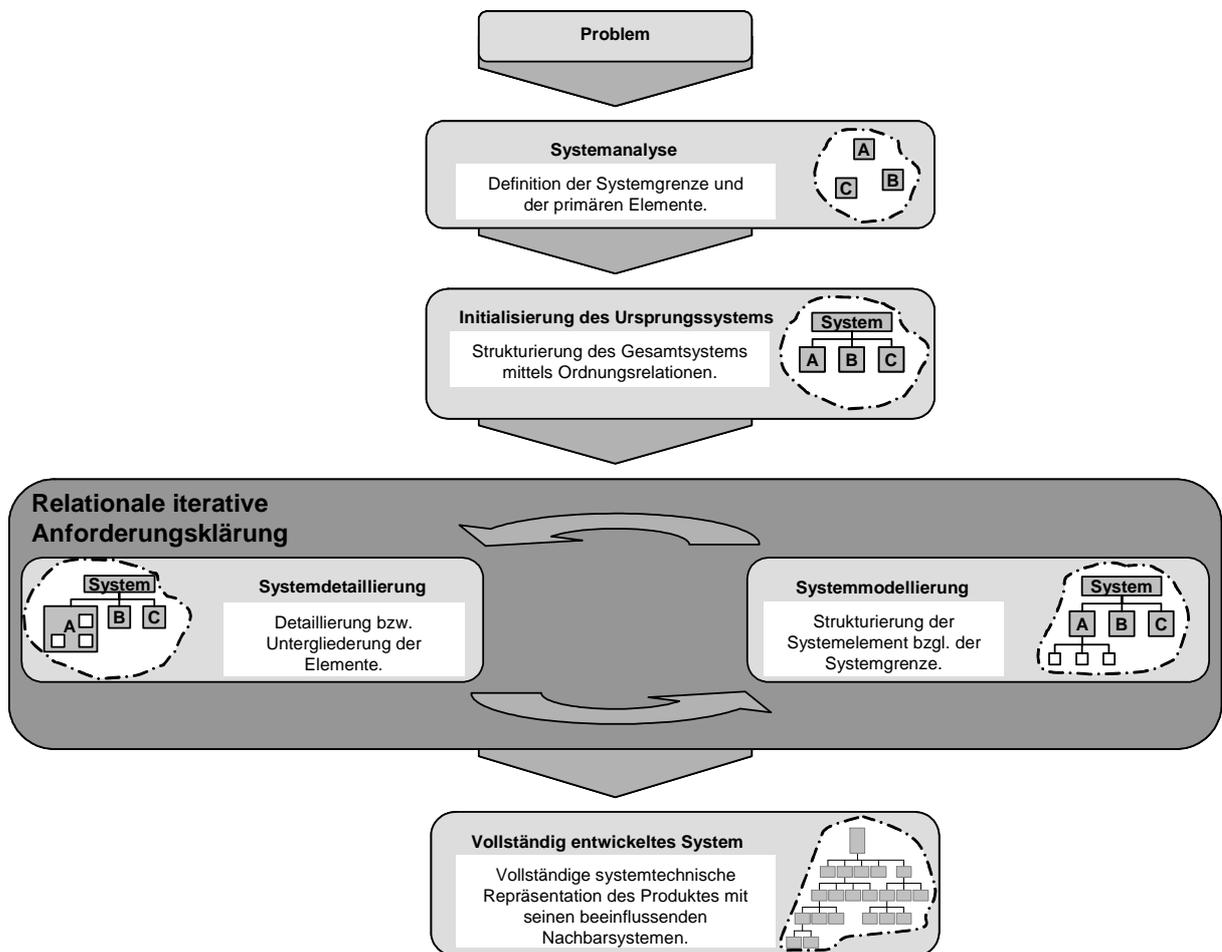


Abb. 5-6: Ablauf des iterativen Systemaufbaus

5.2.2 Relationenanalyse

Über die oben bereits angewendeten Ordnungsrelationen hinaus existieren in einem System weitere Relationen (Flussrelationen etc.). Das Analysieren dieser Relationen hat in dem hier beschriebenen Ansatz zweierlei Funktionen:

- Die Analyse der Relationen soll das Ermitteln der zunehmend lösungsabhängigen Elemente fördern.
- Die Analyse der Relationen in Verbindung mit dem Ermitteln der zunehmend lösungsabhängigen Elemente soll das Aufdecken von *Anforderungsfeldern* erleichtern.

Die Analyse der Relationen kann in mehrere einzelne Tätigkeiten untergliedert werden, die nachfolgend beschrieben werden.

Relationen identifizieren

Zunächst ist es wichtig, überhaupt Relationen zu identifizieren. Hierzu stellt man sich unter dem Gesichtspunkt möglicher Anforderungen die Frage, ob ein bestimmtes Element mit einem anderen Element eine *Relation* hat. Im Beispiel könnte man die Frage stellen, ob das Element „Anwender“ eine Relation zum Element „Staubsauger“ hat. In diesem Fall kann die Frage sehr einfach mit „ja“ beantwortet werden, da der Staubsauger vom Anwender bedient und eingesetzt wird. Es kann aber schon bei kleinen, übersichtlichen Systemen mit etwa 20 Elementen schwer werden, alle Relationen zu überblicken. Um trotzdem eine systematische Suche nach Relationen zu ermöglichen, ist der Einsatz einer Einflussmatrix zu empfehlen [LINDEMANN ET AL. 2004; LINDEMANN 2005, S. 232; ULRICH & EPPINGER 2003, S. 77]. In einer solchen Matrix können die Elemente eines Produkts auf sich selbst abgebildet werden. Es wird dann Schritt für Schritt ein Zeilenelement mit allen Spaltenelementen auf eine Relation überprüft. Dazu wird jeweils die Frage gestellt: „Wirkt dieses Zeilenelement auf dieses Spaltenelement?“ Durch diese Art der Frage wird nicht nur festgehalten, ob eine Relation besteht, sondern auch die Beeinflussungsrichtung. Nachfolgende Tabelle zeigt eine solche Einflussmatrix für eine frühe Entwicklungsstufe des Systems „Handstaubsauger“:

Systemebene	1	2	2	2	1	2	2	1	2	3	4	4	4	4	3	3
Systemebene	Staubsauger	Gehäuse	Saugeinheit	Energieversorgung	Anwender	Kinder	Erwachsene	Umgebung	Saugfläche	Sauggut	Staub	Essensreste	Haare	Parkett	Teppich	
1	Staubsauger															
2	Gehäuse				s											
2	Saugeinheit	st		st												
2	Energieversorgung	st	s													
1	Anwender	st														
2	Kinder	st	st	st												
2	Erwachsene	st	st	st												
1	Umgebung	st														
2	Saugfläche		m	m												
3	Sauggut		st													
4	Staub															
4	Essensreste															
4	Haare		st													
3	Parkett		st													
3	Teppich		st													

s = schwach
m = mittel
st = stark

Tab. 5-1: Entwicklung der Einflussmatrix des Systems Handstaubsaug

Was ist aber, wenn die Frage nach einer Relation nicht so einfach zu beantworten ist? In dieser Situation können ähnliche Mechanismen helfen, wie man sie bereits von der klassischen Anforderungsklä rung kennt. Dort werden Checklisten und ähnliches eingesetzt, um beim Anwender Assoziationen hinsichtlich neuer Anforderungen hervorzurufen [siehe auch EHRENSPIEL 2003, S. 350, LINDEMANN 2005, S. 228, PAHL ET AL. 2003, S. 194]. Hier können diese Mechanismen helfen durch Assoziation auf mögliche Relationen zu schließen. Für die weiteren Ausführungen wird eine Leitlinie mit Relationsattributen in Anlehnung an Pahl et al. [PAHL ET AL. 2003, S. 194] zugrunde gelegt (siehe Anhang 9.1). Dem Anwender wird die Möglichkeit gegeben, sich bei der Ermittlung der Relationen durch die Hauptmerkmale (zum Beispiel Geometrie, Energie, Information) anregen zu lassen, um so

die Frage nach einer Relation zuverlässiger beantworten zu können. Dieses Vorgehen hat des Weiteren den Vorteil, dass zwei Elemente in einem Schritt auf mehrfache Relationen überprüft werden können. Bezogen auf das Beispiel des Handstaubsaugers kann dann die Frage nach Relationen zwischen den Elementen „Staubsauger“ und „Anwender“ wie folgt beantwortet werden: „Ja, es existieren kraftbezogene, informationsbezogene, sicherheitsbezogene, ergonomische etc. Relationen.“

- Kraft: Das Gerät muss vom Anwender getragen werden können.
- Signal: der Anwender muss das Gerät schalten können.
- Sicherheit: Der Anwender darf vom Gerät nicht beeinflusst werden.
- Ergonomie: Der Anwender muss das Gerät benutzen.
- Kosten: Der Anwender muss das Gerät bezahlen.

Nachfolgende Abbildung verdeutlicht dieses Vorgehen nochmals.

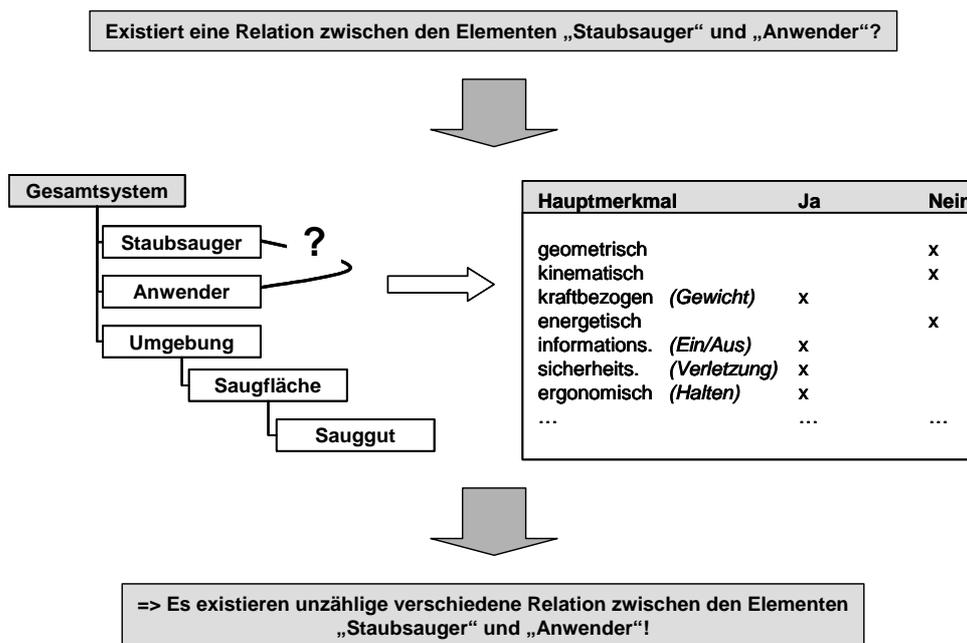


Abb. 5-7: Assoziativ gestützte Ermittlung von Relationen

Relationen klassifizieren

Die ermittelten Beziehungen zwischen den Elementen können durch unterschiedliche Attribute näher spezifiziert werden. Durch das Hinzuziehen einer Assoziationsliste zum Ermitteln von Relationen wurden diese bereits hinsichtlich ihrer Art spezifiziert und können mit dieser Spezifizierung als Attribut versehen werden. Weitere wichtige Attribute sind die Richtung einer Relation und deren Stärke [LINDEMANN 2005, S. 232]. Die Richtung einer

Relation gibt an, welches Element von welchem beeinflusst wird. Hierbei sind einseitige und bidirektionale Relationen zu unterscheiden. Bei einseitigen Relationen wird nur ein Element beeinflusst, das andere ist unbeeinflusst, bei einer bidirektionalen Relation liegt eine wechselseitige Beeinflussung vor. Feyrer hat in diesem Zusammenhang nachgewiesen, dass in Abhängigkeit von der Richtung einer Relation durchaus unterschiedliche Anforderungen abgeleitet werden können, man dies also durchaus berücksichtigen muss [FEYRER 2004, S. 62]. Mit der Stärke einer Relation kann die Stärke des Einflusses angegeben werden, den ein Element auf das andere nimmt. Dabei ist zu berücksichtigen, dass bei einer bidirektionalen Relation die Beeinflussungsstärke in beiden Richtungen nicht zwingend identisch sein muss. Es kann durchaus vorkommen, dass die Relation von Element A zu Element B sehr stark, umgekehrt aber nur schwach ist. Man kann sich vorstellen, dass es mannigfaltige Möglichkeiten der Angabe dieser Relationsstärke gibt (siehe hierzu auch die Gewichtung von Anforderungen in Kapitel 3.4.1), darauf wird aber an dieser Stelle nicht näher eingegangen. Auch kann man die Stärke einer Relation nicht absolut festlegen. Sie kann nur der qualitativen Einordnung einer Relation dienen, um so erste Hinweise auf die Bedeutung von Anforderungen, die aus diesen Relationen abgeleitet werden können, zu erhalten [LINDEMANN 2005, S. 242].

Man kann also festhalten, dass zwischen zwei Elementen Relationen existieren können, die durch die Attribute Art, Richtung und Stärke beschrieben werden können. Nachfolgende Abbildung stellt dies allgemein und anhand von Beispielen dar.

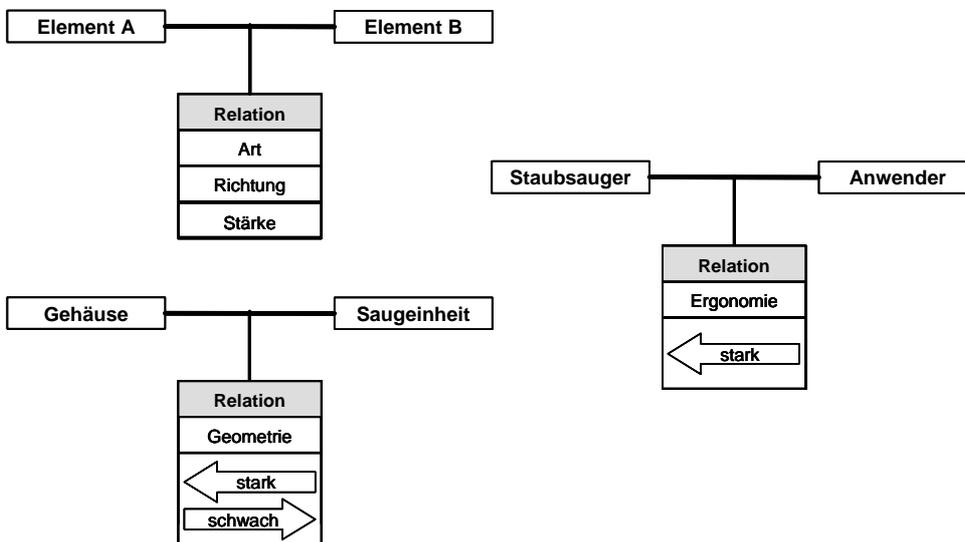


Abb. 5-8: Klassifizierung von Relationen

So wirkt zum Beispiel die ergonomische Relation zwischen den Elementen „Anwender“ und „Staubsauger“ stark in Richtung des Elementes „Staubsauger“, da die unveränderliche Beschaffenheit des Elementes „Anwender“ mit seinen ausführenden Extremitäten bestimmend ist für die spätere Formgebung des Staubsaugers. Betrachtet man die geometrische Relation zwischen dem Element „Gehäuse“ und dem Element „Saugereinheit“,

erkennt man, dass diese bidirektional wirkt. Da die Saugereinheit in einem Staubsauger eine zentrale Einheit ist und eventuell auch Zukaufteile enthalten kann, kann von einer starken Beeinflussung der geometrischen Gestaltung des Gehäuses ausgegangen werden, da dieses die Saugereinheit aufnehmen muss. Andersherum kann die Beeinflussung der Saugereinheit durch das Gehäuse eher schwach sein, da das Gehäuse eventuell nur hinsichtlich Befestigungspunkte etc. der Saugereinheit innerhalb des Gehäuses Einfluss auf diese nimmt.

Besonders auf abstrakten Betrachtungsebenen wird es vorkommen, dass man zwischen zwei Elementen eine Relation identifiziert, diese aber mit mehreren unterschiedlichen Attributen versehen werden kann. In diesem Fall kann die vollständige Information nicht mehr in einer wie oben dargestellten Matrix abgebildet werden. Hierzu müssen andere Hilfsmittel herangezogen werden, die weiter unten vorgestellt werden.

Relationen detaillieren und erweitern

Um eine prozessbegleitende Anforderungsklä rung realisieren zu können, ist es notwendig, im Vorfeld die Relationen prozessbegleitend iterativ und parallel zum System zu analysieren (siehe Kapitel 5.2.1). Hat man eine erste Repräsentation des Systems erstellt (zum Beispiel das Ursprungssystem), kann man die dort vorhandenen Relationen suchen und analysieren (siehe Abb. 5-9). Hat man Relationen auf dieser abstrakten Systemebene gefunden, können sie als Basis für die nächste Detaillierungsstufe herangezogen werden. Dabei ist immer zu berücksichtigen, dass ein Element zu seinem oder dessen übergeordneten Elementen keine Relation außer der Ordnungsrelation zu seinem direkten Vorgänger haben kann.

Eine Relation auf übergeordneter Ebene wird oft auf ein Element auf untergeordneter Ebene vererbt. Im Umkehrschluss bedingt eine Relation auf einer untergeordneten Ebene immer eine entsprechende Relation auf übergeordneter Ebene. Am Beispiel eines Fahrzeugs kann dies leicht nachvollzogen werden: Eine mechanische Relation zwischen Reifen und Fahrbahn setzt die mechanische Beziehung zwischen den übergeordneten Elementen Auto und Fahrbahn voraus. Am selben Beispiel kann aber auch leicht gezeigt werden, dass nicht jede Vererbung einer Relation sinnvoll ist: Die bereits angesprochene Relation zwischen Auto und Fahrbahn führt zum Beispiel nicht zwangsläufig zu einer mechanischen Beziehung zwischen der Innenbeleuchtung und der Fahrbahn, wird also nicht auf die Innenbeleuchtung vererbt.

Eine Relation kann aber nicht nur auf ein Element vererbt werden. Es kann genauso vorkommen, dass ein und dieselbe Relation übergeordneter Ebene auf mehrere Elemente untergeordneter Ebene vererbt werden kann aber nicht zwingend auf alle vererbt werden muss: Im Beispiel wird die sicherheitsbezogene Relation zwischen „Anwender“ und „Staubsauger“ auf die „Saugereinheit“ und die „Energieeinheit“, nicht aber auf das „Gehäuse“ vererbt.

Da Relationen manchmal erst in einer detaillierteren Systemstruktur zu erkennen sind, sind während der Detaillierung des Systems die Beziehungen auf übergeordneter Ebene stets zu hinterfragen und gegebenenfalls zu korrigieren beziehungsweise zu erweitern.

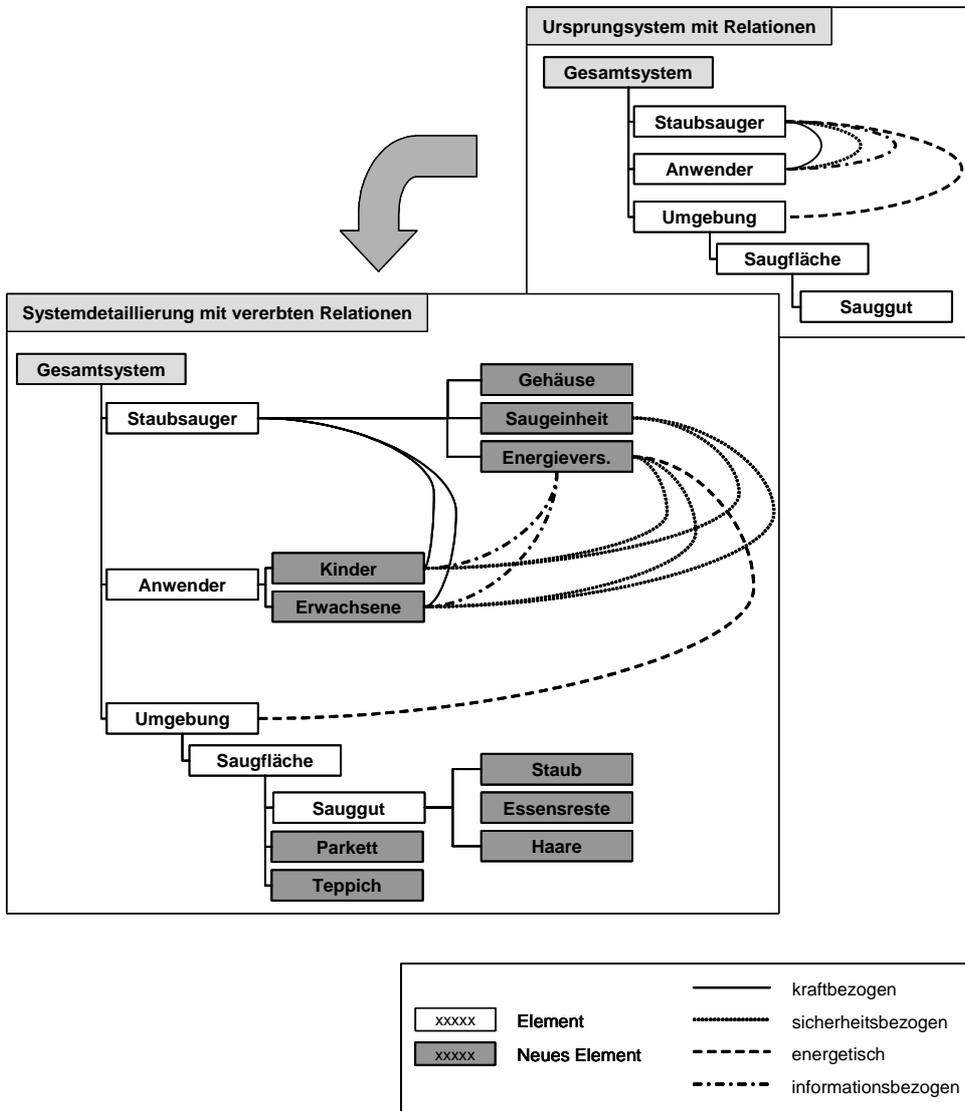


Abb. 5-9: Iterative Analyse und Vererbung von Relationen (der Übersichtlichkeit halber sind nicht alle Relationen eingetragen)

Nachfolgende Abbildung fasst den prinzipiellen Ablauf und das Ergebnis des iterativen Systemaufbaus inklusive der Relationenanalyse nochmals zusammen. Ausgehend von einem initial erstellten Ursprungssystem (die Systemanalyse und die Initialisierung des Ursprungssystems sind nicht dargestellt) werden das Gesamtsystem und dessen Relationen durch die wechselweise Bearbeitung der Schritte Systemdetaillierung, Relationenanalyse und Systemmodellierung fortentwickelt bis zum vollständig entwickelten System mit dessen Relationen.

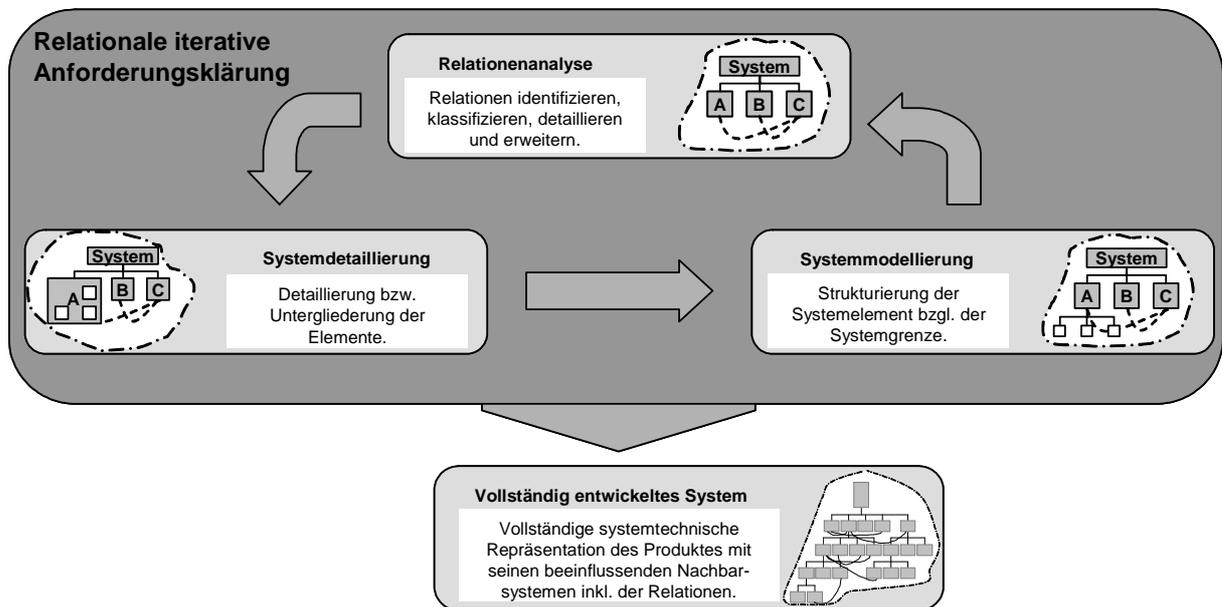


Abb. 5-10: Ablauf des iterativen Systemaufbaus, der Relationenanalyse und -bearbeitung

5.2.3 Ableitung von Anforderungen

Die eigentlichen Ziele des oben beschriebenen Prozesses sind das Identifizieren von Anforderungsfeldern und das Ableiten von Anforderungen. Die dazu notwendige Auseinandersetzung mit den Elementen eines Systems, den Relationen sowie deren Attribute stoßen einen Denkprozess an, um Abhängigkeiten zwischen den Systemelementen zu erkennen. Die dabei eingesetzte Leitlinie mit Relationsattributen stellt die notwendige Assoziationshilfe zur Verfügung, um Relationen und deren Art zu identifizieren. Für das *Erkennen von Anforderungen* ist gerade die Analyse von Relationen und deren Art von besonderer Bedeutung, denn durch die Art einer spezifischen Relation zwischen zwei Elementen wird ein Feld für mögliche Anforderungen eröffnet. Um aus diesem Anforderungsfeld eventuell vorhandene Anforderungen abzuleiten, stellt man sich folgende Fragen:

- Ergeben sich aus der Relation der Beschaffenheit X zwischen Element Y und Element Z eine oder mehrere Anforderungen und wie lautet/n deren Merkmal(e)?
- Wenn ja, welche *Ausprägung* nimmt (nehmen) diese Anforderung(en) bei dieser Relation an?

Dies soll an einem kleinen Beispiel verdeutlicht werden. Für den Staubsauger wurde eine „starke“ „kraftbezogene“ Relation zwischen den Elementen „Staubsauger“ und „Anwender-Erwachsener“ in Richtung des Staubsaugers identifiziert.

„Ergibt sich aus der starken kraftbezogenen Relation zwischen „Staubsauger“ und „Anwender-Erwachsener“ eine Anforderung?“

Mit den vorliegenden Informationen, dem Attribut und den Informationen über die zwei beteiligten Elemente, kann diese Frage unter zu Hilfenahme der Leitlinie mit Relationsattributen beziehungsweise im Gespräch mit einem *Experten* diskutiert werden. In der Leitlinie finden sich zum Attribut „kraftbezogen“ folgende Beispiele:

- Kraftgröße
- Kraftrichtung
- Krafthäufigkeit
- Gewicht
- Last
- ...

Daraus kann abgeleitet werden, dass der „Anwender-Erwachsener“ hinsichtlich des Gewichts des Staubsaugers Einfluss nimmt. Dies ist wiederum die Grundlage für die Formulierung des *Merkmals* „maximales Gewicht des Staubsaugers“.

„Welche Ausprägung nimmt die Anforderung für diese Relation an?“

Kann diese Frage nicht direkt oder von Experten beantwortet werden, kann es notwendig sein, entsprechende Recherchen oder aber auch einen orientierenden Versuch [LINDEMANN 2005, S. 255] durchzuführen, um die Ausprägung der *Anforderung* „maximal Gewicht des Staubsaugers“ *definieren* zu können.

Neben der Relation zwischen „Staubsauger“ und „Anwender-Erwachsener“ existiert eine weitere Relation mit denselben Attributen zwischen „Staubsauger“ und „Anwender-Kind“. Mit dieser Relation kann ebenfalls das bereits vorhandene Merkmal „maximales Gewicht des Staubsaugers“ verknüpft werden. Das bedeutet aber nicht, dass auch die Ausprägung der vorher genannten entspricht. Deswegen muss für diese Relation wiederum die Ausprägung ermittelt werden. Sollten die so ermittelten Ausprägungen unterschiedlich sein, muss dieser Zielkonflikt im weiteren Entwicklungsprozess ausgeglichen werden.

Ähnlich wie bei Humpert beschrieben wird [HUMPERT 1995, S. 65 ff] entstehen auf der Basis des oben gezeigten Vorgehens parallel zu dem eigentlichen Produkt zwei weitere Repräsentationen des Produkts:

- Eine iterativ entwickelte und mittels Relationen vernetzte Systemrepräsentation, die in ihrer letzten Vollendung die vollständige Produktstruktur in Verbindung mit deren Umfeld enthalten kann.
- Ein weitgehend vollständiges Anforderungssystem, das aus den Relationen heraus entwickelt wurde und mit diesen verbunden ist. So kann eine Anforderung stets, wie es von Ward gefordert wird [WARD ET AL. 2003], zu ihrem Ursprung, der Relation zwischen zwei Elementen, zurückverfolgt werden.

5.2.4 Vorgehensmodell der relationalen iterativen Anforderungsklärun

Nachfolgende Abbildung fasst das Vorgehen zur relationalen iterativen Entwicklung von System und Anforderungssystem zusammen: Ausgehend von einem initial erstellten Ursprungssystem werden das Gesamtsystem und dessen Relationen durch die iterative Bearbeitung der Schritte Systemdetaillierung, Relationenanalyse und Systemmodellierung fortentwickelt bis zum vollständig entwickelten System mit dessen Relationen. Parallel dazu werden aus den Relationen Anforderungen abgeleitet. Diese können ähnlich wie das System und dessen Relationen zu Beginn sehr abstrakt sein. Im Verlauf der Entwicklung und der damit zunehmenden Konkretisierung der Lösung für das Entwicklungsproblem werden das System selbst, aber vor allem das Anforderungssystem detailliert und erweitert.

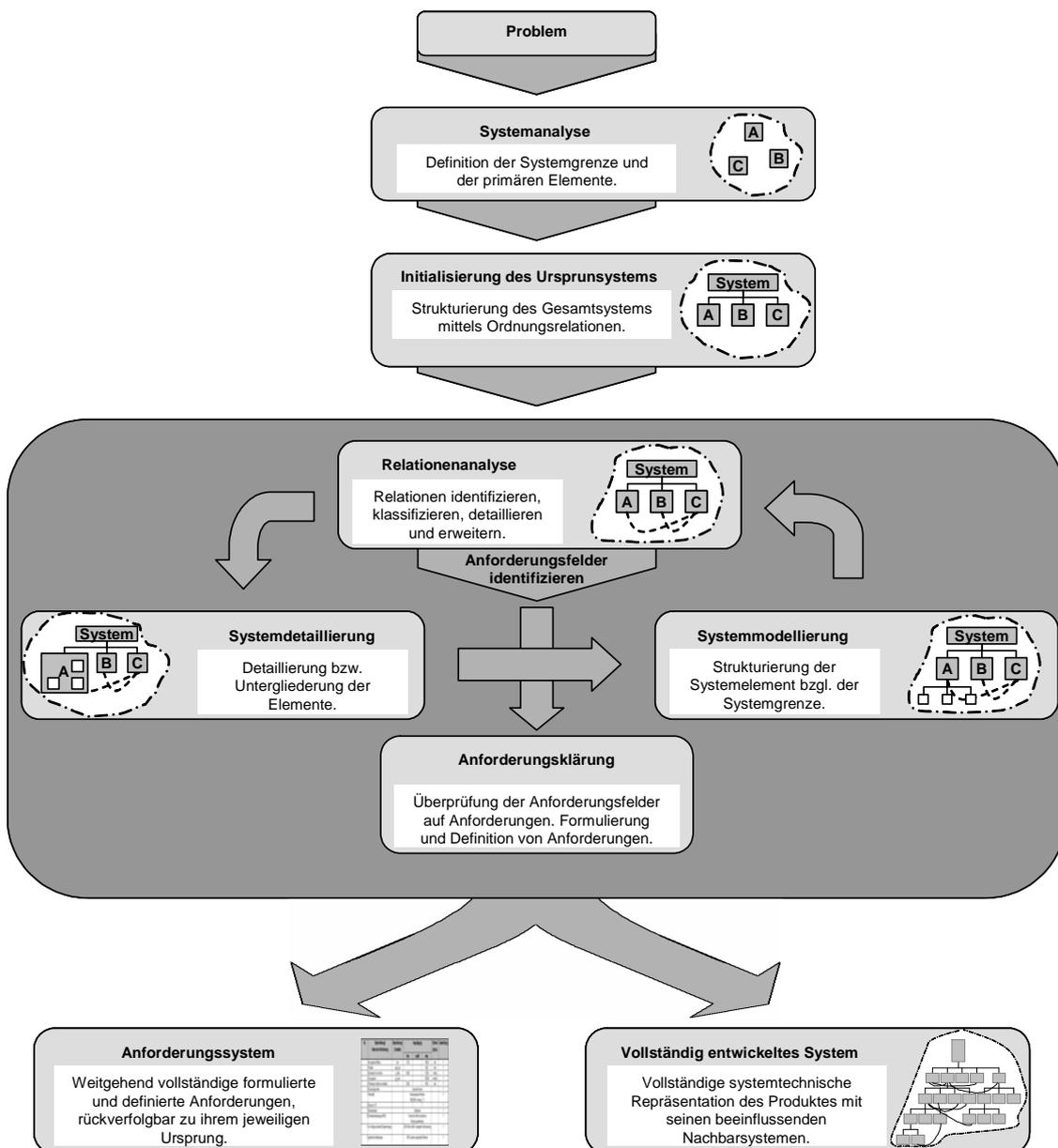


Abb. 5-11: Iterative Entwicklung von System und Anforderungssystem

5.3 Beschreibung eines ganzheitlichen Ansatzes zur Anforderungsklä rung

Jedes regelbasierte Vorgehen hat unterschiedliche Wirkungen. Um die volle Wirkung der relationalen iterativen *Anforderungsklä rung* als ganzheitlichem Ansatz beschreiben zu können, soll sie in das von Lindemann vorgestellte Modell zur Methodenbeschreibung eingeordnet werden. Mit einem bestimmten

- Zweck (hier die Klä rung von *Anforderungen*) wird für eine bestimmte
- Situation (hier eine interdisziplinä re Entwicklungsumgebung)

das Vorgehen beschrieben. Im Folgenden wird zunächst die Kernwirkung der *Methode* herausgearbeitet und abstrahiert. Im Anschluss werden weitere Wirkungen der Methode im Sinne der qualitätsbeeinflussenden Aspekte der Anforderungsklä rung (Kapitel 3.4) herausgearbeitet.

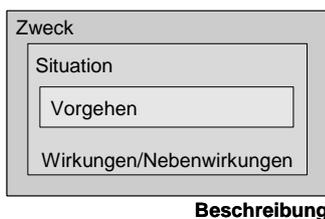


Abb. 5-12: Modell zur Methodenbeschreibung nach Lindemann [LINDEMANN 2005, S. 52]

5.3.1 Kernwirkung der relationalen iterativen Anforderungsklä rung

Um die Kernwirkungen des Vorgehens herausarbeiten zu können, soll das oben dargestellte Vorgehen abstrahiert werden. Dazu wird ein von Hutterer [HUTTERER 2005, S. 94 ff] vorgestellter Ansatz herangezogen. Hutterer bricht Methoden in so genannte Denksteine, das sind in unterschiedlichen Situationen und Methoden immer wieder aufgegriffene elementare geistige Tätigkeiten, auf. Angewendet auf die relationale iterative Anforderungsklä rung kann diese in die folgende iterativ anzuwendenden Denksteine gegliedert [HUTTERER 2005, S. 102 ff] werden, die gleichzeitig die Kernwirkung der relationalen iterativen Anforderungsklä rung beschreiben:

- **Definiere das Ziel:** Für ein gestelltes Problem werden die Anforderungen parallel zur Produktdetaillierung geklä rt.
- **Kontrolliere das Ziel:** Der Entwickler muss, bei Bedarf gemeinsam mit einem *Experten*, das Ziel ständig im Auge behalten und dafür das *System* und die Anforderungen mit dem gestellten Problem ständig abgleichen.

- **Betrachte die Abstraktionsgrade:** Der Abstraktionsgrad der Systementwicklung muss zum jeweiligen Entwicklungszeitpunkt dem Detaillierungsgrad der bis dahin entwickelten Lösung entsprechen. So können System und Lösung wechselseitig weiterentwickelt werden.
- **Betrachte die Teilsysteme:** Zerlegung der *Elemente* in ihre Teilelemente, um so ein strukturierteres und detaillierteres Bild des Gesamtsystems zu erhalten.
- **Betrachte die Relationen:** Die Bestandteile des Systems stehen in unterschiedlicher Beziehung zueinander. Zunächst werden die Ordnungsrelationen genutzt, um das System in Form einer Baumstruktur zu gliedern. Anschließend werden die Relationen der Elemente untereinander gesucht, um *Anforderungsfelder* zu identifizieren. So wird die innere Vernetzung des Systems iterativ analysiert und abgebildet.
- **Generiere Alternativen:** Bei der Generierung der Lösung für ein Problem kann die Suche nach mehreren Lösungen hilfreich sein. Gleiches gilt für die Suche nach Elementen und Relationen zur Systemdetaillierung. Besonders bei der Relationenanalyse ist es notwendig, in Alternativen zu denken, um so das Vergessen von Anforderungsfeldern zu vermeiden. Dies wird durch den Einsatz der Leitlinie mit Relationsattributen gefördert.
- **Hole Informationen ein:** Kann der Entwickler die Frage nach Elementen, Relationen oder Anforderungen nicht zweifelsfrei selbst beantworten, ist er gezwungen, sich diese Informationen von außen zu holen und dafür einen entsprechenden Experten einzubinden.
- **Ermittle Wichtigkeiten:** Um Anhaltspunkte für die spätere Gewichtung von Anforderungen zu erhalten, wird die Bedeutung beziehungsweise das Gewicht der einzelnen Relationen ermittelt.
- **Lagere Wissen aus:** Neue Informationen, die der Entwickler über sein System in Form von weiteren Elementen, Relationen, Anforderungen etc. erlangt hat, lagert er in geeigneter Form aus. Somit ist dieses Wissen für die weitere Verarbeitung gespeichert und abrufbar.

Diese neun Denksteine und deren iterative Anwendung helfen im Kern, dass parallel zur Detaillierung des Produkts dessen System schrittweise aufgebaut und verfeinert, strukturiert und vernetzt wird. Analog zu den von Cross vorgestellten Ansätzen zur Entwicklung von Baumstrukturen zur Anforderungsklä rung [CROSS 1996, S. 50 f] stellt die dabei entstehende Baumstruktur des Systems immer nur eine temporäre Darstellung dar, die mit dem Entwicklungsprozess verändert beziehungsweise erweitert wird. Ähnlich wie bei Cross [CROSS 1996, S. 53] liegt der Fokus der Systementwicklung innerhalb der relationalen Anforderungsklä rung nicht auf dem System selbst, sondern auf dem Prozess, der Arbeit mit der Methode am System, da erst durch die Diskussion der Elemente und Relationen das Systemverständnis erweitert wird, beziehungsweise Wissenslücken identifiziert werden. Diese aufgedeckten Wissenslücken führen dann zur Hinzuziehung von Experten. Die innerhalb dieses Prozesses neu gewonnenen Informationen fließen aus der

Systementwicklung in die Produktdetaillierung und umgekehrt. Durch dieses Vorgehen werden das Systemverständnis und der Detaillierungsgrad des Produkts sukzessive erhöht und gleichzeitig anschaulich dargestellt. Für das eigentliche Ziel der relationalen iterativen Anforderungsklä rung ist der Denkbaustein „Betrachte die Relationen“ von besonderer Bedeutung. Hier werden durch die Analyse der Relationen die für das *Erkennen* aber auch für das Pflegen und Erweitern von *Anforderungen* wichtigen Anforderungsfelder identifiziert. Aus diesen können im weiteren Verlauf Anforderungen abgeleitet und vorhandene detailliert werden: Mittels neuer Relationen werden neue Anforderungsfelder und in der Folge neue Anforderungen identifiziert. Genauso kann die Vererbung von Relationen oder die Detaillierung der an einer Relation beteiligten Elemente zu einer Detaillierung der betroffenen Anforderung (en) oder sogar zu einer Aufspaltung einer Anforderung auf mehrere führen.

Damit ist die in Kapitel 4 aufgestellte Kernanforderung „Systematisierung des Abfragens des zu entwickelnden Produkts nach Anforderungen unter Einbindung bewährter Methoden und Durchführung des Abfragens bis zum Abschluss der Entwicklung, um so während des gesamten Entwicklungsprozesses (also parallel zum zunehmenden Detaillierungsgrad des Produkts) relevante Anforderungen erkennen, konkretisieren und präzisieren zu können.“ erfüllt. Gleichzeitig werden damit die Aspekten der Anforderungsklä rung „Erkennen von Anforderungen“ und „Pflegen und Erweitern von Anforderungen“ erfüllt.

5.3.2 Nebenwirkung der relationalen iterativen Anforderungsklä rung

Experteneinbindung³⁰

Die relationale iterative Anforderungsklä rung sieht die permanente Einbindung von Experten vor. Durch die systematische Identifizierung von Wissenslücken wird dies gefördert: Der Entwickler ist jetzt in der Lage, seine Wissenslücken zu erkennen und kann in der Folge mit gezielten Fragen an den Experten herantreten, wie zum Beispiel mit der Frage nach einer *Relation* zwischen zwei ganz bestimmten Elementen oder der Frage nach einer Anforderung resultierend aus einer Relation definierter Art. Dadurch können beim Experten auch implizite Wissensbereiche angestoßen werden. Bezogen auf das Beispiel aus Kapitel 2.1.1 hätte eine „stoffliche“ Relation zwischen den Elementen „Blut“ und „Pulsatorkopf“ identifiziert werden können. Die Frage nach einer daraus resultierenden Anforderung hätte beim Experten den entscheidenden Denkprozess anstoßen können, um die entsprechende Anforderung bezüglich des Werkstoffs zu explizieren.

Weiterverarbeitung von Kundenanforderungen

Auch für die Weiterverarbeitung von Kundenanforderungen bietet die relationale iterative Anforderungsklä rung neues Potential. In der Literatur wird zwar vielfach beschrieben, wie Kundenanforderungen mit technischen Anforderungen vernetzt werden können, um deren Einflüsse auf die Entwicklung analysieren zu können [CROSS 1996; DANNER 1996;

³⁰ Analog zur Einführung des Begriffs „Experten“ soll auch die Kundeneinbindung auf die Einbindung von Experten ausgeweitet werden.

LINDEMANN 2005; ULRICH & EPPINGER 2003]. Dafür ist es aber notwendig, die *technische Anforderungen* zu kennen. Es fehlen aber Hilfsmittel um diese direkt aus Kundenanforderungen ableiten zu können. Bindet man bei der relationalen iterativen Anforderungsklä rung den *Kunden* als Element in das zu entwickelnde System ein (siehe das Beispiel des Handstaubsaugers in Kapitel 5.2) und ordnet diesem seine Kundenanforderungen als Subelemente zu, kann der Entwickler deren Relation mit anderen Systemelementen untersuchen. Es wird also nicht mehr wie beim Ansatz des QFD der Einfluss der Kundenanforderungen auf technische Anforderungen, sondern auf Systemelemente betrachtet. Hat man die daraus resultierenden Relationen gefunden und festgelegt, können technische Anforderungen abgeleitet werden. Dies entspricht dem von Lin et al. geforderten Vorgehen, globale Kundenanforderungen auf mehrere unterschiedliche Teilsysteme herunterzubrechen, um so Beziehungen zwischen Kundenanforderungen und Subsystemen ableiten und letztendlich Anforderungen an Teilsysteme erkennen zu können [LIN ET AL. 1996]. Dem Entwickler wird ein Weg gezeigt, Kundenanforderungen systematisch zu technischen Anforderungen weiterzuverarbeiten.

Erkennen von Anforderungen

Das Erkennen von Anforderungen wird durch die Kernwirkung der Methode erfüllt, und ist im vorangegangenen Kapitel 5.3.1 beschrieben worden.

Pflege und Erweiterung von Anforderungen

Über das primäre Erkennen hinaus (siehe Kapitel 5.3.1) müssen die einmal gefundenen Anforderungen angepasst, detailliert und ergänzt werden. Dieser Prozess kann unterschiedliche Auswirkungen haben. Zum einen kann dies einfach bedeuten, dass das Merkmal oder die Ausprägung einer Anforderung spezialisiert beziehungsweise endgültig festgelegt wird (siehe Kapitel 5.2.3). Dies kann aber auch bedeuten, dass aufgrund von neuen Relationen beziehungsweise einer erneuten Systemdetaillierung eine Zerlegung von Anforderungen in mehrere Teilanforderungen notwendig wird.

Kruse verwendet für diese Zusammenhänge den Begriff der strukturellen Beziehungen zwischen Anforderungen [KRUSE 1996, S. 78 ff]:

- **Spezialisierung:**
Zuwachs von Eigenschaften für eine Anforderung innerhalb eines Entwicklungsprozesses. Die Anforderung wird detaillierter beschrieben.
- **Dekomposition:**
Zerlegung einer Anforderung in ihre Teilforderungen.
- **Variation:**
Definition von alternativen Anforderungen, zum Beispiel zum Zwecke der Lösungsfindung.

Da der Entwickler nach jeder Änderung in seinem System (Systemdetaillierung, neue oder vererbte Relationen etc.) dasselbe zumindest in den betroffenen Bereichen iterativ analysiert, werden aufgrund der Verknüpfung von Element mit Relation mit Anforderung die notwendigen Informationen weitergegeben, um eine eventuell notwendige Erweiterung einer Anforderung in oben angesprochener Art anzuregen.

Formulieren und Definieren von Anforderungen

Für das *Formulieren* und *Definieren* von *Anforderungen* kann die relationale iterative Anforderungsklä rung bedingt Unterstützung geben. Unterstützt wird der Entwickler insofern, dass durch die Mehrstufigkeit der Anforderungsklä rung eine strikte inhaltliche und zeitliche Trennung der Klä rung von *Merkmal* und *Ausprägung* einer Anforderung vorgenommen wird: Anforderungen, deren Merkmal bereits die Ausprägung enthält oder ähnliches werden damit erschwert. Gleichzeitig bekommt der Entwickler durch die Art einer Relation und die daran beteiligten Elemente Hinweise auf das *Anforderungsfeld* und damit oftmals auf die Formulierung, vor allem aber auf die eventuell notwendige physikalische Einheit der Ausprägung.

Besonders bei Neuentwicklungen wird es immer wieder vorkommen, dass man die Ausprägung zu einer Anforderung nicht klä ren kann, ja sogar Anforderungen noch nicht vollständig aus Relationen ableiten kann, da der Kenntnisstand über das zukünftige Produkt nicht ausreichend ist. Wie auch von Restrepo oder Sigherianu gefordert wird [siehe auch RESTREPO & CHRISTIANS 2003; SIGHERIANU & TURNER 1998], sollte man in dieser Situation dennoch wenige, lösungsneutrale Anforderungen, wenn nötig auch ohne Ausprägung, formulieren, die Freiheit für eine spätere, spezifischere Interpretation lassen. Da die für diese Anforderungen relevanten Relationen innerhalb des iterativ durchlaufenen Prozesses der Systementwicklung vererbt und/oder verfeinert werden, können die unkonkreten Anforderungen während des Entwicklungs- und Konstruktionsprozesses parallel zur Generierung und Detaillierung der Lösung konkretisiert und dem gegenwärtigen Stand der Erkenntnisse angepasst werden. Analog zu Abb. 2-4 kann man diesen Prozess zur Klä rung einer einzelnen Anforderung wie in Abb. 5-13 darstellen:

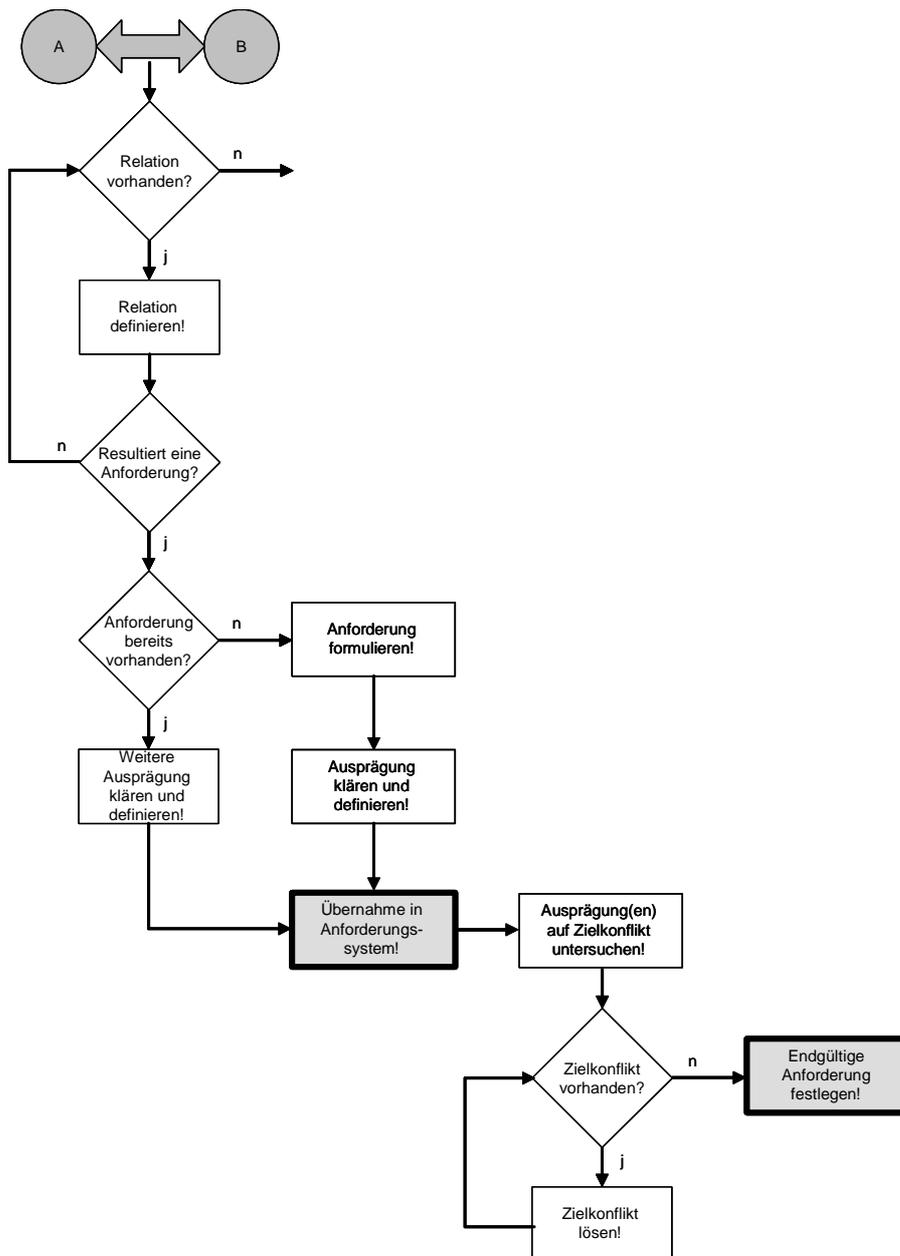


Abb. 5-13: Prozess zur Klärung einer einzelnen Anforderung mittels der relationalen Anforderungsklärung³¹

Zunächst wird geprüft ob zwischen zwei Elementen eine Relation vorliegt. Ist dies nicht der Fall, kann der Prozess verlassen werden. Liegt eine vor, wird diese hinsichtlich Art, Richtung und Bedeutung untersucht und diese festgelegt. Anschließend wird überprüft, ob aus dieser Relation eine Anforderung resultiert. Ist dies nicht der Fall beziehungsweise kann diese Frage noch nicht beantwortet werden, muss die Relation sofort (keine Anforderung resultiert) oder

³¹ Diese Darstellung fasst die Klärung einer einzelnen Anforderung zusammen, die in der Realität innerhalb der relationalen iterativen Anforderungsklärung über mehrere Iterationen hinweg erfolgen kann.

zu einem späteren Zeitpunkt (Frage kann nicht beantwortet werden) überprüft werden. Resultiert keine Anforderung, stellt sich die Frage, ob die Relation stimmt. Resultiert eine Anforderung, wird überprüft, ob diese Anforderung bereits im Anforderungssystem enthalten ist:

- Liegt diese Anforderung bereits aus einer anderen Relation vor, muss für die neue Relation die Ausprägung der Anforderung ermittelt, definiert und in das Anforderungssystem übernommen werden.
- Liegt diese Anforderung noch nicht vor, kann diese formuliert werden. Im Anschluss daran wird, soweit möglich, die Anforderung definiert und in das Anforderungssystem übernommen. An dieser Stelle ist die Darstellung gegenüber Abb. 2-4 vereinfacht dargestellt. Natürlich muss auch hier eine Analyse des bisher bekannten Systems auf die Ausprägung hin stattfinden, bis die Anforderung definiert werden kann.
- Liegen dann zu einem Merkmal mehrere Ausprägungen vor, müssen diese auf Zielkonflikte untersucht werden. Liegt ein Konflikt vor, muss dieser sofort oder zu einem späteren Zeitpunkt gelöst werden. Zuletzt kann die endgültige und bereinigte Anforderung in das Anforderungssystem übernommen werden. Liegt nur eine Ausprägung vor, kann diese mit ihrem Merkmal als endgültige Anforderung übernommen werden.

Der in der Darstellung angesprochene Zielkonflikt meint nur Zielkonflikte unterschiedlicher Ausprägungen innerhalb einer Anforderung, wenn also zu einem Merkmal mehrere widersprüchliche Ausprägungen existieren. Konflikte zwischen unterschiedlichen Merkmalen und deren jeweiligen Ausprägungen können auf diesem Weg nicht aufgedeckt und bearbeitet werden. Auf diese Zusammenhänge wird noch detaillierter eingegangen werden.

Für die sprachlichen Aspekte des Formulierens von Anforderungen, deren Bedeutung nicht zu vernachlässigen ist [LIN ET AL 1996], kann mit diesem Ansatz keine gesonderte Hilfestellung gegeben werden.

Gewichtung von Anforderungen

Da bereits bei der Ermittlung der Relationen deren Bedeutung beziehungsweise Stärke untersucht wird, wird dem Entwickler von den frühesten Entwicklungsphasen an Unterstützung für die Gewichtung von Anforderungen gegeben. Wird aus einer Relation eine Anforderung abgeleitet, kann die Bedeutung der Relation sogleich als Anhaltspunkt für die Gewichtung verwendet werden. Auch hier gilt wieder der Grundsatz: Vom Abstrakten zum Detail. Die Relationen wird man anfangs nur in ein eher einfaches Raster einordnen. Die Anforderungen können dann schon in ein differenzierteres Schema eingeordnet werden. So können sich schon in sehr frühen Phasen Entwicklungsschwerpunkte abzeichnen. Da auf die Gewichtungen auch bei der Bewertung und Auswahl von Lösungen zurückgegriffen wird, wird dem Entwickler somit die Möglichkeit einer durchgängigen Gewichtung und damit Identifizierung von Entwicklungsschwerpunkten gegeben.

In diesem Zusammenhang soll die Handhabung der Gewichtung von Kruse [KRUSE 1996, S. 84] als Empfehlung aufgegriffen werden. Unabhängig von der Wahl des Gewichtungsmaßstabs empfiehlt es sich in den Augen des Verfassers, *Festforderungen* gar nicht zu gewichten und *Bereichsforderungen* entsprechend ihrer Bedeutung in den Bewertungsmaßstab einzuordnen.

- Bei Festforderungen erübrigt sich eine Gewichtung insofern, dass sie in jedem Fall erfüllt sein müssen. Andernfalls wären sie mit einer falschen Ausprägung belegt. Diese Festforderungen können dann als absolutes Entscheidungskriterium (ja/nein) für die Lösungsauswahl herangezogen werden.
- Bereichsanforderungen dienen bezüglich der Einhaltung des in der Ausprägung angegebenen Bereichs ebenfalls als Kriterium für ja/nein Entscheidungen. Die Gewichtungen der Bereichsforderungen geben dann den Grad der Übererfüllung [MONTAU 1996, S. 57] einer Anforderung gegenüber anderen an und können zum Vergleich der verbleibenden Lösungen untereinander herangezogen werden. Dazu wäre es in den Augen des Verfassers hilfreich, für eine Bereichsforderung die bevorzugte Richtung der Übererfüllung anzugeben. Nicht immer ist eindeutig, ob das Optimum einer Bereichsforderung an der unteren oder oberen Grenze oder vielleicht sogar dazwischen liegt. Diese Information wäre aber für die Bearbeitung sehr hilfreich.

Strukturierung von Anforderungen

Hinsichtlich der Strukturierung von Anforderungen bietet die relationale iterative Anforderungsklä rung unterschiedliche Möglichkeiten. In sehr frühen Phasen der Entwicklung können die Relationen hinsichtlich ihrer Attribute (Art, Richtung, Bedeutung) und ihrer Elementzugehörigkeit strukturiert werden. Analog können die aus den Relationen abgeleiteten Anforderungen hinsichtlich der Art, des Gewichts, der Relationen und damit der Elementzugehörigkeit strukturiert werden. Dies birgt zum Beispiel auch die für spätere Entwicklungsphasen sehr interessante und wichtige Möglichkeit in sich, Anforderungen nach ihrer Baugruppenzugehörigkeit zu strukturieren, so dass die jeweiligen Sachbearbeiter nur die für sie relevanten Anforderungen angezeigt bekommen. Gleichzeitig sind über die zu den Anforderungen gehörenden Relationen immer die entsprechenden system- oder produktbezogenen Schnittstellen einsehbar.

Zielkonflikte und Vernetzung von Anforderungen

Im vorangegangenen Abschnitt wurde die Behandlung von Zielkonflikten bereits angesprochen. Da mittels der relationalen Anforderungsklä rung Anforderungen nicht willkürlich gesammelt werden, sondern immer in Bezug zu einer Relation stehen und das Anforderungssystem bei jeder neu hinzukommenden Anforderung auf das Vorhandensein einer solchen Anforderung überprüft wird, werden Doppelnennungen vermieden.

Abgesehen davon, dass Anforderungen sich neutral gegenüberstehen können, existieren zwischen Anforderungen unterschiedliche Vernetzungen, die sich auch als Zielkonflikte

repräsentieren können. Unter anderem Kruse [KRUSE 1996, S. 81] schlägt für diese Form der Beziehungen, die er semantische Beziehungen nennt, folgende Einteilung vor³²:

- Ausschluss: Zwei Anforderungen sind miteinander nicht realisierbar.
- Konkurrenz: Zwei Anforderungen wirken gegenläufig auf die gleiche Eigenschaft.
- Unterstützung: Zwei Anforderungen wirken gleichsinnig auf eine Eigenschaft.

Dabei sind zwei Fälle zu unterscheiden:

- Fall 1: Eine Anforderung wird aus unterschiedlichen Quellen gespeist und/oder hat unterschiedliche Ursachen, die abweichende Ausprägungen oder auch Formulierungen liefern und somit zu Konflikten führen können (siehe das Beispiel Handstaubsauger in Kapitel 5.2.3).
- Fall 2: Zwei oder mehr unterschiedliche Anforderungen stehen zueinander in Beziehung (zum Beispiel hohe Geschwindigkeit und niedriger Kraftstoffverbrauch bei einem Pkw).

Mögliche Zielkonflikte, wie sie in Fall 1 beschrieben sind können durch die relationale iterative Anforderungsklä rung identifiziert werden, da unterschiedliche Ausprägungen einer Anforderung an dieselbe angehängt werden und nicht als eigene Anforderungen in das Anforderungssystem eingetragen werden. Für Zielkonflikte des Falls 2 kann die relative relationale Anforderungsklä rung keine eigene Hilfestellung bieten. Dazu wird auch auf den Ausblick (Kapitel 7.2.1) verwiesen, in dem dieser Punkt nochmals aufgegriffen wird.

Alle identifizierten Konflikte müssen rechtzeitig im Prozess gelöst werden. Dies bedeutet aber nicht, dass alle Konflikte möglichst früh im Prozess gelöst werden müssen. Zunächst muss nicht jeder Konflikt auch einen Widerspruch darstellen. In den Augen des Verfassers kann man grundsätzlich versuchen einen Konflikt durch Anpassung der Anforderungen zu eliminieren oder ihn als Herausforderung zu sehen und versuchen ihn durch Innovation technisch zu lösen. Man kann durch einen Konflikt quasi zur Innovation gezwungen werden. Erst wenn dieser Konflikt nicht technisch zu lösen ist muss er zum Beispiel durch einen Kompromiss gelöst werden. Natürlich sind noch weitere Randbedingungen zu beachten. So hat auf die Lösung von Zielkonflikten auch die Gewichtung der betroffenen Anforderungen großen Einfluss: Handelt es sich bei diesen um Festforderungen, ist die Konfliktlösung wesentlich sensibler, als wenn zumindest eine Anforderung eine (niedrig) gewichtete Bereichsforderung ist. Auch haben Kosten und Terminanforderungen oftmals einen großen Einfluss auf die Möglichkeiten der Konfliktlösung.

Wie man an diesen kurzen Ausführungen sehen kann, ist die Behandlung von Zielkonflikten ein sehr komplexes und umfassendes Thema, das den Rahmen dieser Arbeit mehr als sprengen würde, weshalb hier auf die verfügbare Literatur verwiesen sei [zum Beispiel EILETZ 1999]

³² Ähnliche Definitionen findet man auch bei Lindemann [LINDEMANN 2005, S. 90] oder Ehrlenspiel [EHRLENSPIEL 2003, S. 350].

Vollständigkeit von Anforderungen

Ein in der Analyse des Standes der Forschung festgestellter bedeutender Mangel ist die fehlende Überprüfbarkeit der Vollständigkeit eines Anforderungssystems. Bisher konnte man nur abschätzen, ob alle Punkte einer Checkliste bearbeitet wurden. Das lässt aber noch viel Spielraum für vergessene Anforderungen. Auch die relationale iterative Anforderungsklä rung kann hierfür kein absolutes Kriterium zur Verfügung stellen. Dies ist in den Augen des Verfassers auch nicht möglich. Der Ansatz spannt aber eine zweite Dimension auf, anhand deren der Entwickler die Möglichkeit hat, zumindest eine qualitative Abschätzung der Vollständigkeit vorzunehmen: Der Entwickler untersucht das System zum einen mittels einer Checkliste (erste und bereits bekannte Dimension) und muss diese Checkliste systematisch und iterativ auf das zunehmend detailliertere System anwenden (zweite und neue Dimension). In Abhängigkeit vom Detaillierungsgrad seines Systems kann der Entwickler ein „Gefühl“ für die Vollständigkeit des Anforderungssystems entwickeln, das der Realität näher kommen dürfte als die bisherige Abschätzung anhand der Checkliste. Unter der Annahme, dass am Ende einer Entwicklung das System vollständig abgebildet, alle Relationen aufgedeckt und die resultierenden Anforderungen abgeleitet wären, könnte man daher annehmen, dass das Anforderungssystem weitgehend vollständig sei. Natürlich ist dies eine sehr theoretische Annahme. In wieweit aber man dennoch von einer „Abschätzbarkeit“ der Vollständigkeit sprechen kann, bleibt zu untersuchen.

Interdisziplinarität

Die derzeit fehlende Unterstützung der interdisziplinären Zusammenarbeit wird durch die gemeinsame, schrittweise Entwicklung des Systems gefördert, da Entwickler und andere eingebundene Experten ein gemeinsames Systemverständnis und eine gemeinsame Terminologie entwickeln und etablieren können. Zwar kann es vorkommen, dass diese dann nur für die gerade betrachtete Entwicklung gültig ist, aber im Grundsatz ist es egal wie man etwas benennt, solange alle Beteiligten das selbe darunter verstehen. Dies kann durch das oben beschriebene Vorgehen erreicht werden. Man kommt somit zu einem ähnlichen Ergebnis wie Andelfinger [ANDELFINGER 1998, S. 201] mithilfe seiner diskursiven Systemanalyse.

Lin [LIN ET AL. 1996] hat festgestellt, dass die Autoren von Anforderungen unterschiedliche Terminologien verwenden, was immer wieder zu Fehlinterpretationen und Missverständnissen führt. Der Einsatz unterschiedlicher Terminologien kann durch die gemeinsame Systementwicklung und die damit einhergehende Entstehung eines gemeinsamen Systemverständnisses zwar vermindert, nicht aber gänzlich verhindert werden. Die Gefahr und der Grad der Fehlinterpretation kann aber reduziert werden, da eine Anforderung wie von Ward gefordert [WARD ET AL. 2003] über ihre Relation(en) immer zu ihrem Ursprung verfolgbar ist. Dadurch werden die Aussagekraft einer Anforderung stark erhöht und Fehlinterpretationen minimiert.

5.4 Werkzeug für die relationale iterative Anforderungsklä rung

Da sich bei der Entwicklung des methodischen Ansatzes bereits abzeichnete, dass eine rein gedankliche oder „papiergestützte“ Handhabung und Verarbeitung eines Systems, seiner Relationen und deren Attribute und der daraus abgeleiteten *Anforderungen* nur schwer, bei komplexeren Produkten wohl gar unmöglich sein würde, wurde ein Softwaretool zur Unterstützung des Ansatzes entwickelt [FEYRER 2004]. Das Tool sollte in erster Linie die Verifikation des Methodenansatzes ermöglichen und in der Folge als Grundlage für weitere Entwicklungen und Optimierungen dienen.

5.4.1 Anforderungen an ein unterstützendes Tool

Aus dem beschriebenen Vorgehen wurden die Kernfunktionen des zu entwickelnden Werkzeugs abgeleitet und als funktionale Anforderungen an das Werkzeug formuliert [FEYRER 2004, S. 77 ff] (siehe Anhang 9.2). Da das grundsätzliche Vorgehen der relationalen iterativen *Anforderungsklä rung* oben ausreichend beschrieben wurde, wird auf eine Beschreibung dieser Anforderungen verzichtet. Einzig auf das „Hinzufügen, editieren und löschen von unterschiedlichen *Ausprägungen* zu einer Anforderung bis zu deren Freigabe“ wird hier aufgrund seiner Besonderheit im Vergleich zu den bekannten Handlungsanweisungen zur Anforderungsklä rung kurz eingegangen:

Es wurde bereits auf das *Definieren* von *Anforderungen* eingegangen. Dabei wurde festgestellt, dass eine Anforderung aus mehreren Relationen abgeleitet werden kann. Das *Merkmal* der Anforderung ist dann dasselbe, dies muss aber nicht zwingend für die Ausprägung gelten. Aus den beteiligten Relationen können durchaus unterschiedliche Ausprägungen folgen³³. In solchen Fällen ist es sinnvoll, wenn für eine Anforderung zunächst mehrere Ausprägungen deren Merkmal zugeordnet werden können, um zu einem späteren Zeitpunkt, also spätestens für die Freigabe, einen Überblick über alle Ausprägungen zu haben, um dann eine Entscheidung über die endgültige Ausprägung treffen zu können. Dieses Vorgehen wurde in das Softwaretool implementiert.

5.4.2 Systemauswahl

Da auf dem Markt bereits unterschiedliche Software erhältlich ist, die die Darstellung von *Elementen* und deren Relationen ermöglicht, wurden verschiedene Systeme auf ihre Eignung für die beschriebene Aufgabe untersucht (zum Beispiel Touchgraph[®], PersonalBrain[®], Sensitivitätsmodell Prof. Vester[®]) [FEYRER 2004, S. 84 ff]. Alle betrachteten Systeme sind zwar auf ihrem Gebiet sehr weit fortgeschritten und grafisch sehr ansprechend aufbereitet.

³³ *Vergleiche hierzu auch das Beispiel des Handstaubsaugers in Kapitel 5.2.3: Hier folgt sowohl aus der Relation „Anwender: Erwachsener“ – „Staubsauger“ als auch aus der Relation „Anwender: Kind“ die Anforderung mit dem Merkmal „maximales Gewicht des Staubsaugers“. Die daraus ableitbare Ausprägung wird aber unterschiedlich ausfallen, da ein Kind wohl eine andere Vorstellung von maximalem Gewicht hat als ein Erwachsener.*

Die notwendigen Anpassungsarbeiten an die oben beschriebenen Funktionen wären aber im Vergleich zum angestrebten Ziel eines einfachen Tools zur Verifikation zu zeit- und kostenintensiv gewesen. Als günstigste Lösung kristallisierte sich die Umsetzung einer eigenen Lösung auf der Basis eines käuflichen frei programmierbaren Systems heraus. Die Wahl fiel auf das System Microsoft[®] Access.

5.4.3 Beschreibung des Werkzeugs RIA

In den folgenden Ausführungen sollen nicht die softwaretechnische Einzelheiten des entwickelten Werkzeugs RIA³⁴ beschrieben werden. Es werden nur die im Sinne der aufgestellten funktionalen Forderungen notwendigen Inhalte des Werkzeugs vorgestellt. Daher werden nur die wichtigen Benutzerschnittstellen der Anwendung abgebildet und kurz beschrieben. Mit diesen Ausführungen wird gleichzeitig die Anwendung des Werkzeugs beschrieben. Die Erklärungen erfolgen teilweise anhand des Beispiels „Handstaubsaugers“. Dabei stehen nicht die technischen Details des Beispiels sondern der Umgang mit und die Funktion des Werkzeugs RIA im Vordergrund.

Die Architektur des entwickelten Werkzeugs entspricht der untenstehenden schematischen Abbildung. Die Anwendung ist in Anlehnung an Raasch [RAASCH 1993] als Schichtenmodell aufgebaut und lässt spätere Änderungen und Erweiterungen aufwandsarm zu. Die unterste Schicht wird durch das Betriebssystem gebildet, das die Steuerung von Hardwarefunktionen übernimmt. Darauf setzt die Datenbanksystem-Schicht auf, welche die Eingangs- und Ausgangsgrößen für die Funktionen der Anwendungsschicht aufnimmt. Die oberste Schicht bildet die Benutzerschnittstelle³⁵. Diese ist für die Bedienung des Systems durch den Nutzer notwendig und als einzige für diesen sichtbar.

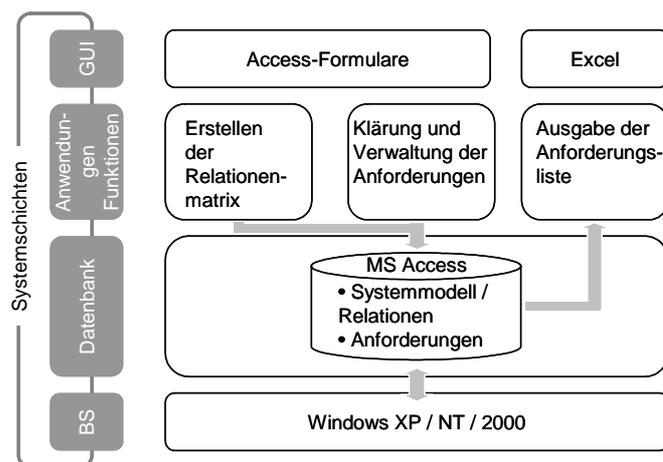


Abb. 5-14: Schematische Darstellung der Systemarchitektur [FEYRER 2004, S. 86]

³⁴ RIA – Relationale Iterative Anforderungsklä rung

Die Datenbank-Schicht enthält als Datenquelle Microsoft® Access, in der die Relationen und das Systemmodell abgebildet sowie die Anforderungen verwaltet werden. Die eigentliche Datenverarbeitung findet auf der Anwendungsschicht statt. Access bildet mit der Datenbank die Grundlage des Tools zur Unterstützung der Anforderungsklä rung. Für die graphische Benutzeroberfläche wurden die Formulare aus Access und Excel verwendet. Die Programmierung der Funktionen wurde mit Visual Basic for Application (VBA) durchgeführt [FEYRER 2004, S. 86].

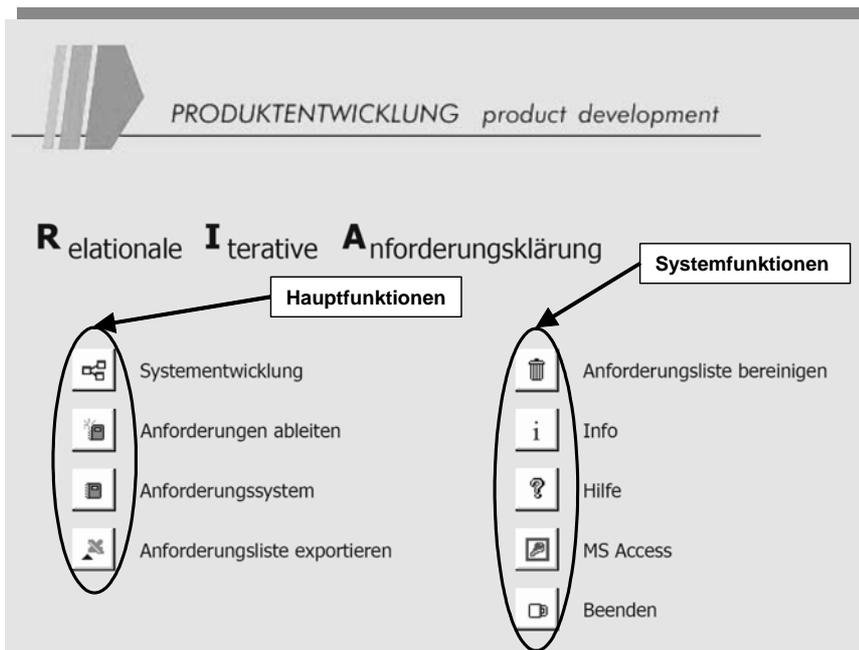


Abb. 5-15: Startmenü

Im Startmenü werden dem Anwender alle Funktionen der Anwendung angeboten. Dies sind zum einen die Kernfunktionen, die für die Durchführung der relationalen iterativen Anforderungsklä rung benötigt werden.

- **Systementwicklung:** Ruft die Routine auf, die den Anwender bei der Systemdetaillierung, Systemmodellierung und Relationenanalyse (siehe Abb. 5-11) unterstützt.
- **Anforderungen ableiten:** Ruft die Routine auf, die den Anwender beim Ableiten von Anforderungen aus den Relationen beziehungsweise Anforderungsfeldern unterstützt und leitet. Gleichzeitig kann hier eingesehen werden welche Anforderungen aus einer *Relation* folgen.

³⁵ GUI – Graphical User Interface

- **Anforderungssystem:** Ruft die Routine auf, die den direkten Zugriff auf alle Anforderungen ermöglicht. Hier kann eine Rückverfolgung einer Anforderung zu ihren ursächlichen Relationen vorgenommen werden. Gleichzeitig ist eine einfache Änderungshistorie von Anforderungen hinterlegt.
- **Anforderungsliste exportieren:** Ermöglicht den Export sowohl der Anforderungen als auch der Anforderungen in Kombination mit ihren ursächlichen Relationen in eine Microsoft Excel[®]-Datei.

Die weiteren Schaltflächen rufen verschiedene unterstützende Systemfunktionen auf beziehungsweise beenden die Anwendung. Auf diese wird hier nicht näher eingegangen, da sie für die Erfüllung der relationalen iterativen Anforderungsklä rung nicht zwingend notwendig sind.

Ruft der Anwender die Routine „Systementwicklung“ auf, zeigt sich ihm folgendes Bild. Um die Übersichtlichkeit zu erhöhen, werden die vier markierten Felder (A, B, C, D) dieses Formulars unten vergrößert dargestellt und einzeln erläutert.

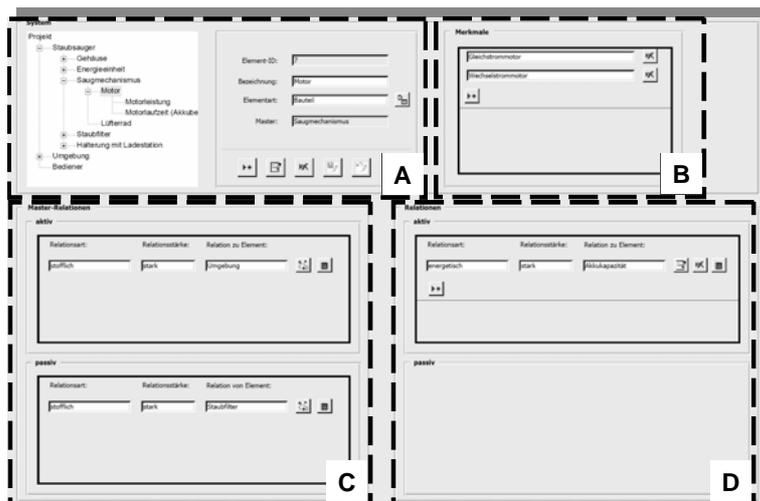


Abb. 5-16: Übersicht über die Systementwicklung

In dem „Feld A“ (siehe Abb. 5-17) wird dem Anwender eine grafische Übersicht über das von ihm erstellte System in Form einer Baumstruktur der Elemente gegeben. Wählt er innerhalb der Systemstruktur ein Element aus, erhält er in nebenstehenden Feldern Informationen zu diesem Element (Bezeichnung, Elementart und die Bezeichnung des übergeordneten Elements). In der Abbildung ist dies für das Element Motor zu sehen. Hat der Anwender ein Element angewählt, kann er diesem weitere Elemente unterordnen und sie mit den gerade angesprochenen Informationen versehen. Dabei hat er die Möglichkeit für die Elementart entweder aus einer Vorgabe auszuwählen oder beliebige eigene Elementarten neu zu definieren und hinzuzufügen. Auf diese kann er dann für das Anlegen weiterer neuer Elemente zugreifen.

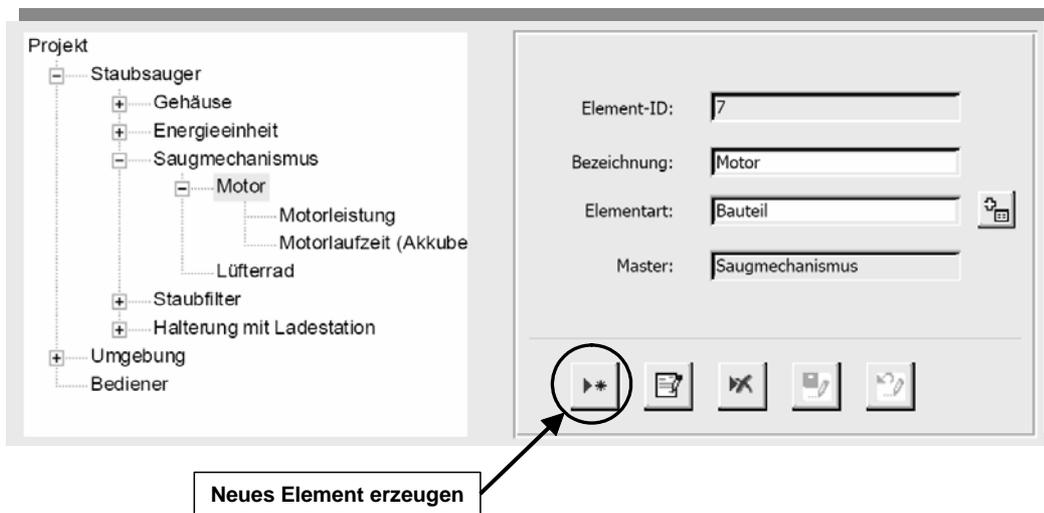


Abb. 5-17: Systementwicklung: „Feld A“ – Systemstruktur

Um einem Element noch weitere Informationen hinzufügen zu können, gibt es noch das nachstehend abgebildete Merkmalsfeld „Feld B“. Dieses bezieht sich im Beispiel auch auf den oben bereits angesprochenen Motor. In diesen Feldern hat der Anwender die Möglichkeit, jedem Element beliebige Merkmale hinzuzufügen. Diese können unterschiedlicher Natur sein. Sie werden dem Anwender bei unterschiedlichen Handlungen innerhalb der relationalen iterativen Anforderungsklä rung angezeigt und sollen ihn bei der Formulierung, vor allem aber beim Definieren von Anforderungen unterstützen.

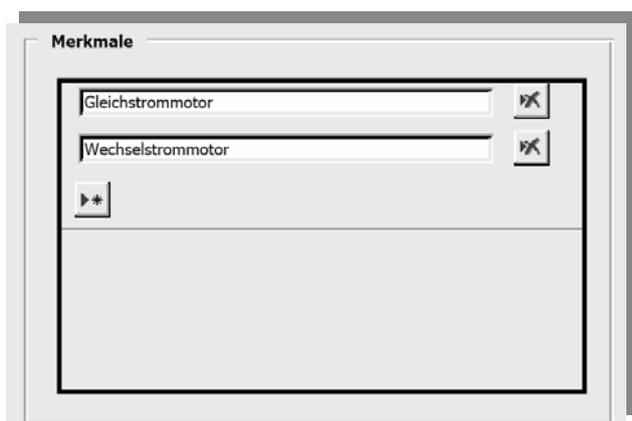


Abb. 5-18: Systementwicklung: „Feld B“ – Elementmerkmale

Die bisher erläuterten Felder dienen der Unterstützung der Systemdetaillierung und Systemmodellierung, die nachfolgend beschriebenen Felder (Abb. 5-19 und Abb. 5-20)

dienen in Verbindung mit dem „Feld A“ der Relationenanalyse. Die in diesen Feldern angezeigten Inhalte hängen von dem im Feld „System“ angewählten Element ab (im Beispiel das Element „Motor“). Beide Felder sind in die Bereiche „aktiv“ und „passiv“ unterteilt, die dementsprechend die aktiven (Element beeinflusst ein anderes) und passiven (Element wird von einem anderen beeinflusst) Relationen enthalten.

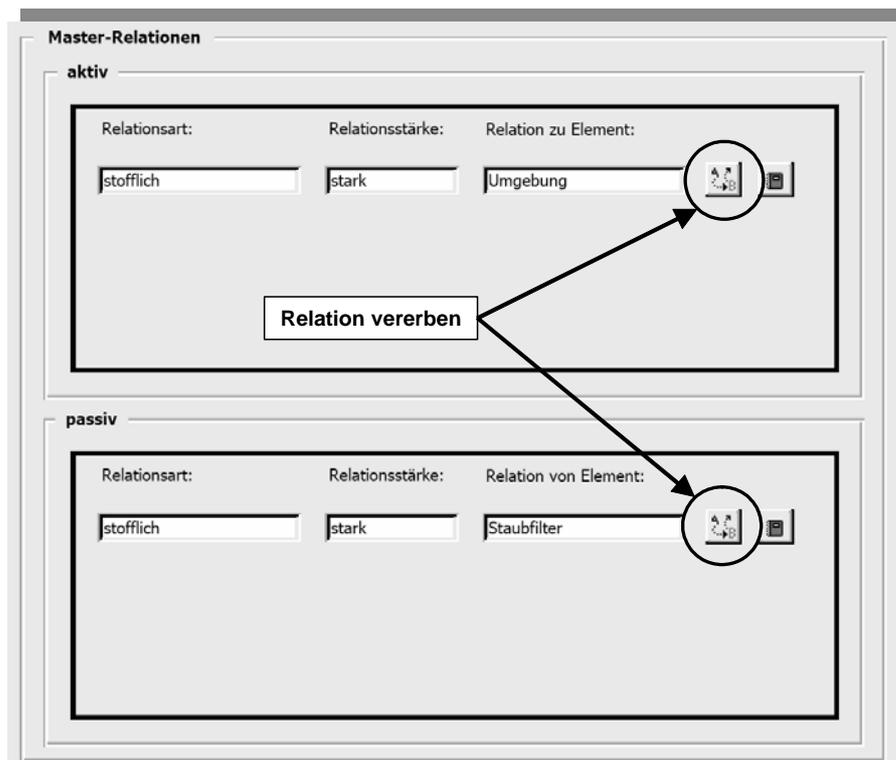


Abb. 5-19: Systementwicklung: „Feld C“ – übergeordnete Relationen (Master-Relationen)

So kann die Richtung einer Relation angegeben werden. Gleichzeitig wird zu jeder Relation deren Art und Stärke und natürlich das beteiligte Element angegeben. Der Unterschied zwischen „Feld C“ „Master-Relationen“ (Abb. 5-19) und „Feld D“ „Relationen“ (Abb. 5-20) liegt in der Betrachtungsebene: Während das „Feld C“ die Relationen des dem angewählten Element übergeordneten (im Beispiel das Element „Saugmechanismus“) anzeigt, gibt das „Feld D“ die Relationen des angewählten Elements an. Auf diese Weise hat der Anwender alle für ein Element relevanten Informationen auf einen Blick verfügbar. In Bezug auf das Beispiel bedeutet dies:

- Das dem Element „Motor“ übergeordnete Element „Saugenheit“ hat eine starke aktive stoffliche Relation zu dem Element „Umgebung“ (Abb. 5-19, aktiv).
- Das dem Element „Motor“ übergeordnete Element „Saugenheit“ hat eine starke passive stoffliche Relation von dem Element „Staubfilter“ (Abb. 5-19, passiv).

- Das Element „Motor“ hat eine starke energetische Relation auf das Element „Akkukapazität“ (Abb. 5-20, aktiv).
- Eine passive Relation auf das Element „Motor“ ist in diesem Beispiel noch nicht identifiziert worden, würde dann aber auch nicht ausgehend vom Element „Motor“ erzeugt werden, sondern kann nur von dem jeweiligen beeinflussenden Element aus eingetragen werden. Diese Relation würde bei dem beeinflussenden Element dann als aktive, beim Motor als passive Relation erscheinen.

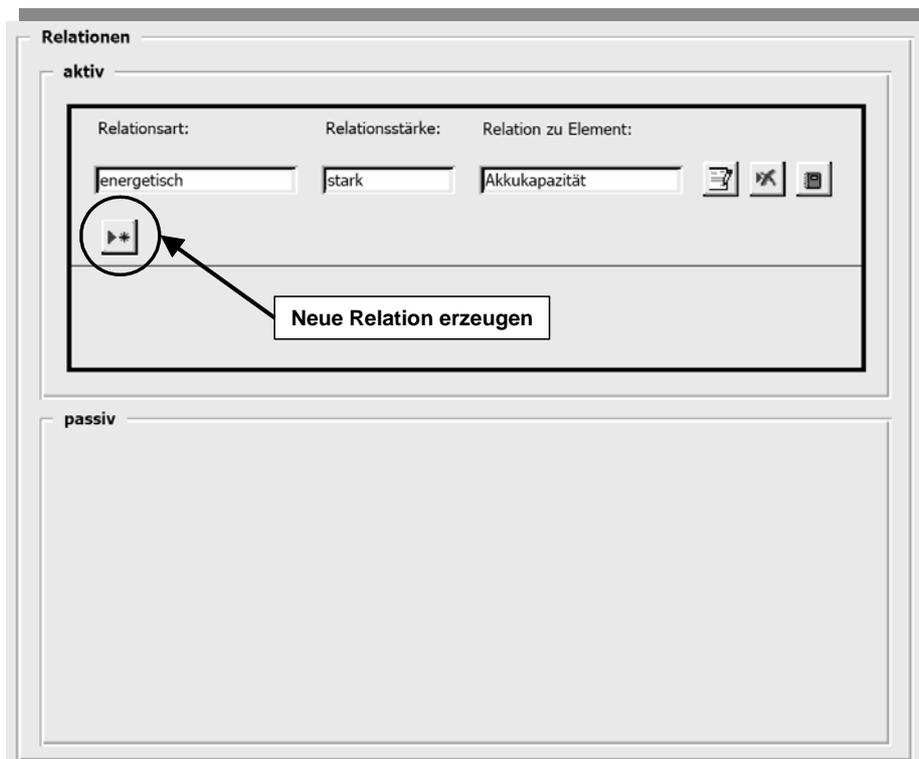


Abb. 5-20: Systementwicklung: Ausschnitt D – Relationen

Natürlich dienen diese Felder nicht ausschließlich der Anzeige von Relationen, sondern vor allem der Analyse der Relationen. Zur Analyse der Relationen wird sich der Anwender zunächst auf übergeordneter Ebene mit der Frage beschäftigen, ob das betrachtete Element ein anderes Element beeinflusst. Anhand dieser Frage muss er dann das System Schritt für Schritt bearbeiten. Sollte er dabei eine Relation identifizieren, muss er im Feld „Relationen“ eine neue (aktive) Relation von dem aktuell angewählten Element auf ein anderes Element eintragen (Abb. 5-20). Dazu erzeugt er eine neue Relation und wählt in dem sich öffnenden Dialogfenster Art und Stärke der Relation und das beeinflusste Element aus. Für die Art der Relation kann er wieder aus einer Vorgabe, die auf der Leitlinie mit Relationsattributen aufbaut, auswählen oder weitere individuelle Attribute eintragen. Auf diese kann später wieder zugegriffen werden. Die Art der Relation eröffnet gleichzeitig ein Anforderungsfeld. Erkennt der Anwender, dass eine Relation des übergeordneten Elements für das betrachtete

gültig ist, muss er diese Relation nicht neu erzeugen, sondern kann die bereits vorhandene vom übergeordneten Element auf das betrachtete vererben (Abb. 5-19). Die Relation wird bei dieser Maßnahme nur auf der Seite des aktuell betrachteten Elements vererbt. Das andere „Ende“ der Relation bleibt zunächst unverändert. Bei Bedarf kann er die Relationsattribute an die veränderten Gegebenheiten anpassen.

Will der Anwender aus den analysierten Relationen Anforderungen ableiten, wählt er im Startmenü die Funktion „Anforderung ableiten“. Ihm zeigt sich dann das in Abb. 5-21 zu sehende Dialogfeld. In diesem kann man alle bereits analysierten Relationen sequentiell oder gefiltert einsehen. Zunächst ist jede Relation als „unbearbeitet“ klassifiziert. Im oberen Feld werden die gerade betrachtete Relation und deren Attribute zu einer Frage nach einer möglichen Anforderung formuliert.

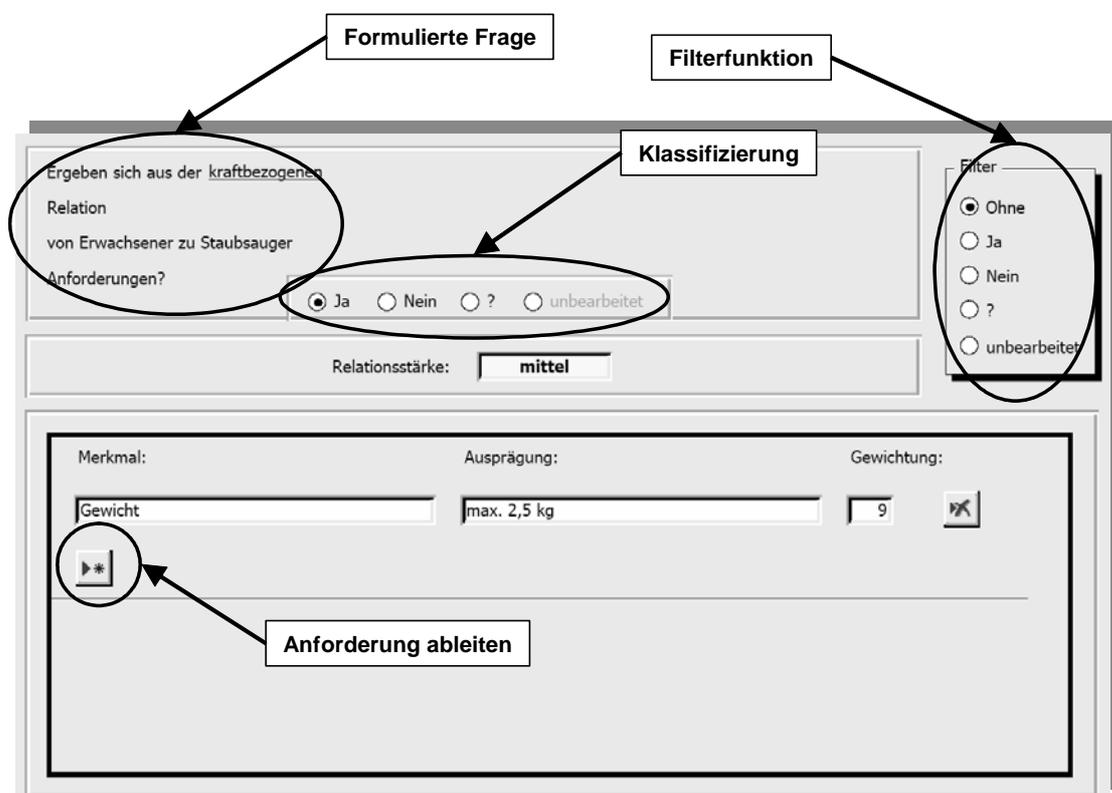


Abb. 5-21: Identifizierung und Ableitung von Anforderungen

Der Anwender hat nun die Möglichkeit die Relation ausgehend von dieser Frage zu klassifizieren. Kann die Frage eindeutig beantwortet werden, kann die Klassifizierung „Ja“ oder „Nein“ zugewiesen werden, ansonsten muss ein „?“ zugewiesen werden, womit diese

Frage für eine spätere Bearbeitung zurückgestellt wird³⁶. Mittels dieser Klassen können die Relationen auch jederzeit gefiltert betrachtet werden. Sollten Relationen identifiziert worden sein, aus denen auch zu späteren Zeitpunkten, wenn bereits detaillierte Informationen über das zukünftige Produkt vorliegen, keine Anforderungen abgeleitet werden können (Klassifizierung „Nein“), sollte man überprüfen, ob diese Relationen wirklich existent sind, und sie bei Bedarf wieder aus dem System löschen, da so die Übersichtlichkeit gewahrt bleibt.

Kann aus einer Relation eine Anforderung abgeleitet werden, gelangt man (siehe Abb. 5-21) in den untenstehenden Dialog, der der eigentlichen Anforderungsklä rung dient. Dieses Dialogfeld lässt sich in zwei Bereiche untergliedern: In der oberen Hälfte befinden sich die eine Anforderung vollständig beschreibenden Attribute, die sich an der von Lindemann [LINDEMANN 2005, S. 96] vorgestellten Anforderungsliste orientieren.

- Anforderungs-ID: Laufende, von der Anwendung vergebene Nummer
- Merkmal: Beschreibendes Merkmal einer Anforderung
- Kategorie: Kategorie einer Anforderung, die aus der Relationsart abgeleitet wird. Die Kategorie dient später der Strukturierung der Anforderungsliste.
- Kurzvariable: Hier können Variablen für Anforderungen hinterlegt werden, die den Bezug zu Skizzen, technischen Zeichnungen etc. erleichtern sollen (zum Beispiel „l“ für „Länge“).
- Gewichtung: Gewichtung der Anforderung, die aus der Relationsstärke abgeleitet werden kann
- Datum: Datum des Ersteintrags oder der letzten Änderung der Anforderung
- Ersteller: Name des Anforderungserstellers
- Ursprung: Herkunft einer Anforderung (Normen, Gesetze ...)
- Anlage: Näher definierende Anlagen zu einer Anforderung (Norm, Zeichnung, Katalog ...)
- Ausprägung: Endgültige, freigegebene Ausprägung einer Anforderung. Diese sollte möglichst numerisch, kann aber auch textuell angegeben werden. Der Anwender hat unterschiedliche Möglichkeiten eine Ausprägung anzugeben: Untere Schranke, obere Schranke oder einen festen Wert. Zusätzlich kann die Einheit der Ausprägung angegeben werden.

³⁶ Dies entspricht der in Abb. 5-13 dargestellten rekursiven Bearbeitung einer nicht eindeutig zu beurteilenden Relation.

Die untere Hälfte des Dialogfelds stellt die Verbindung zum System und den Relationen des zu entwickelnden Produkts dar und dient der Anforderungsableitung. Auf der linken Seite ist für den Anwender noch einmal die Relation mit ihren beteiligten Elementen und Attributen formuliert. Gleichzeitig werden dem Anwender die Merkmale der an der Relation beteiligten Elemente angegeben. Die rechte Seite bietet dem Anwender die Möglichkeit die für die gerade betrachtete Relation gültige Ausprägung der betrachteten Anforderung anzugeben. Die Angabe von mehreren vorläufigen Ausprägungen ist möglich.

Anforderung

Anforderungs-ID: Gewichtung:

Merkmal: Datum:

Kategorie: Ersteller:

Kurzvariable:

Ursprung: Anlage:

Ausprägung: min: Wert oder Text: Einheit:

max:

Anforderungsableitung

Relation

Anforderungsklä rung aus der kraftbezogenen Relation (mittel)
von Erwachsener zu Staubsauger

Merkmale von: Erwachsener

Erwachsener	Percentil 95
-------------	--------------

vorläufige Ausprägung

Ausprägung: Gewichtung:

Datensatz:

Abb. 5-22: Anforderungssystem

Die Klärung von Anforderungen in dem beschriebenen Dialogfeld läuft wie folgt ab:

- Nachdem der Anwender eine Relation identifiziert hat, aus der eine Anforderung resultiert, gelangt er in das beschriebene Dialogfeld. Zunächst kann er die bereits dort eingetragenen Anforderungen durchsehen, um zu überprüfen, ob eine entsprechende Anforderung bereits formuliert ist. Ist dies der Fall, kann der nächste Schritt übersprungen werden.
- Aus der Formulierung der Relation und dem assoziierten Anforderungsfeld in Verbindung mit den eventuell bekannten Merkmalen der beteiligten Elemente wird, bei Bedarf mit einem *Experten*, das Anforderungsfeld und daraus die Formulierung einer Anforderung abgeleitet. Des Weiteren werden die notwendigen beziehungsweise bekannten Attribute der Anforderung angegeben (Ersteller, Anlage etc.). Die Gewichtung kann zunächst weggelassen werden. Ebenso wird in den Feldern der Ausprägung zunächst kein Eintrag vorgenommen.
- Soweit es möglich ist, wird eine vorläufige, oftmals noch grobe Ausprägung eingetragen und mit der Relation verknüpft. Wenn diese Anforderung bereits im System vorhanden war, kann hier einfach eine weitere vorläufige Ausprägung hinzugefügt werden und mit der aktuellen Relation verknüpft werden. Auch kann eine bereits mit einer anderen Relation verknüpfte vorläufige Ausprägung mit der betrachteten Relation verknüpft werden, wenn diese sich entsprechen. Wichtig ist, dass für jede vorläufige Ausprägung eine aus der Relationsstärke abgeleitete Gewichtung angegeben wird.

Die so festgehaltenen vorläufigen Ausprägungen können jederzeit verändert, erweitert oder gelöscht werden. Dabei ist zu berücksichtigen, dass nach dem Löschen der letzten vorläufigen Ausprägung einer Anforderung diese mit keiner Relation mehr verknüpft ist. Ist der Kenntnisstand zu einer Anforderung ausreichend hoch, kann eine endgültige Ausprägung getroffen werden. Existiert nur eine vorläufige Ausprägung, die auch mehrfach verknüpft sein kann, so kann diese als endgültige Ausprägung übernommen werden. Liegen dagegen mehrere unterschiedliche vorläufige Ausprägungen vor, müssen diese für eine endgültige Ausprägung zusammengefasst werden. Ist eine endgültige Ausprägung eingetragen worden, unterliegt die Anforderung der Änderungshistorie: Jede Änderung an der Anforderung wird protokolliert und kann nur unter Angabe eines Grundes und eines Bearbeiters durchgeführt werden.

Sowohl für die Diskussion von Anforderungen und deren Ausprägungen, als auch für die Verarbeitung von Anforderungen in der Entwicklung ist es oftmals hilfreich, diese in listenmäßiger Darstellung vorliegen zu haben. Dafür kann das komplette Anforderungssystem in eine Anforderungsliste in Microsoft® Excel exportiert werden. Um der unterschiedlichen Behandlung der vorläufigen und endgültigen Ausprägungen gerecht zu werden, gibt es zwei unterschiedliche Listen. Erstere enthält jede mit einer Anforderung verknüpfte Relation mit der (den) dazugehörigen vorläufigen Ausprägung(en). Sie gibt einen Überblick über die Relationen und deren verknüpften Anforderungen und hilft so die vorläufigen Ausprägungen zu endgültigen zusammenzufassen.

Nr.	Merkmal	vorläufige Ausprägung	Gewichtung	Relation von	Relationsart	Relationsstärke	Relation zu
1	Energie						
1	Akkukapazität	2,1	10	Motor	energetisch	stark	Akkukapazität
2	Betriebsdauer Akku						
2	Ergonomie						
1	Lautstärke	max. 60 db	5				
3	Kraft						
1	Gewicht	max. 1,5 kg	9	Kind	kraftbezogen	mittel	Staubsauger
1	Gewicht	max. 2,5 kg	9	Erwachsener	kraftbezogen	mittel	Staubsauger
4	Mechanik						
1	Motorleistung						

Tab. 5-2: Anforderungsliste mit vorläufigen Ausprägungen und dazugehörigen Relationen

Die zweite Liste gibt jede Anforderung mit ihrer endgültigen Ausprägung und ihrer Änderungshistorie wieder³⁷.

Nr.	Merkmal	Kurzvariable	Ausprägung			Einheit (phys.)	Gewichtung	Ursprung	A.
			min.	Wert oder Text	max.				
1	Energie								
1	Akkukapazität				1,4	Ah	10,00		
2	Betriebsdauer Akku	t	5			Minuten		Kundenanforderung (HoQ)	
2	Ergonomie								
1	Lautstärke				60 db	db			
3	Kraft								
1	Gewicht	m			2	kg		Kudentest	
4	Mechanik								
1	Motorleistung	P	120			W		Kundenanforderung	
1	Motorleistung	P	120			W	7,00	Kundenanforderung	

Anlage	Ersteller	Datum	Änderungsverfolgung		
			Was	Wer	Datum
			Was	Wer	Datum
	Feyrer	16.06.2004			
	Feyrer	16.06.2004	Ersteintrag Zahlenwert	Feyrer	16.06.2004
	Feyrer	16.06.2004			
	Jung	29.03.2005			
	Feyrer	16.06.2004	Ersteintrag Zahlenwert	Feyrer	16.06.2004
	Feyrer	16.06.2004	Ersteintrag Gewichtung	Feyrer	16.06.2004

Tab. 5-3: Anforderungsliste mit endgültigen Ausprägungen

In den Abbildungen kann man zum Beispiel anhand der Anforderung Gewicht die Funktion der beiden Listen erkennen. In der ersten Abbildung ist zu erkennen, dass die Anforderung Gewicht mit zwei Relationen verknüpft ist, die unterschiedliche Ausprägungen mit sich bringen. In der unteren Abbildung wird ersichtlich, dass man sich hier auf einen Kompromiss geeinigt hat. In beiden Darstellungen wird in dieser Ausbaustufe auf die Kategorie (also das Anforderungsfeld) einer Anforderung als ordnendes Kriterium zurückgegriffen.

Durch die konsequente iterative und rekursive Anwendung der oben beschriebenen Programmfunktionen kann ein Anforderungssystem Schritt für Schritt entwickelt und konkretisiert und der Entwicklung in Form von Listen zugänglich gemacht werden.

5.5 Zusammenfassung

Um einen möglichst allgemeingültigen und einfachen Ansatz zur Unterstützung der Anforderungsklä rung entwickeln zu können, wurde nach den Ursachen von Anforderungen geforscht. In diesem Zuge konnte nachgewiesen werden, dass sich Anforderungen aus den Relationen zwischen Elementen eines Systems ableiten lassen und von diesen bestimmt werden.

Ausgehend von dieser Erkenntnis wurde ein iteratives Vorgehen zur relationalen iterativen Anforderungsklä rung entwickelt, das den Entwickler einerseits unterstützt das zu einem Produkt gehörende System aufzubauen, dessen Relationen zu analysieren und daraus Anforderungen abzuleiten. Gleichzeitig hilft dieses Vorgehen dem Entwickler Wissenslücken zu identifizieren. Er erkennt, wann er einen Experten hinzuziehen muss und wird in der Formulierung von Fragen an den Experten unterstützt. Im Sinne der an eine Methode zur Anforderungsklä rung gestellten Anforderungen wurde das Vorgehen hinsichtlich seines Wirkens analysiert und den Anforderungen gegenübergestellt. Es konnte gezeigt werden, dass das Vorgehen die gestellten Anforderungen erfüllen kann.

Für den praktischen Einsatz und die Verifikation des Ansatzes wurde ein Softwaretool entwickelt, das das beschriebene Vorgehen abbildet, den Anwender unterstützt und durch den Prozess leitet.

³⁷ Wurde eine Anforderung verändert, wird diese in der listenmäßigen Repräsentation des Anforderungssystems durchgestrichen und unter Angabe der Begründung die veränderte eingetragen.

6 Fallbeispiele

Um die relationale iterative Anforderungsklä rung und das dazugehörige Werkzeug RIA in der Praxis testen zu können, wurden sie in Entwicklungsprojekten eingesetzt. Für die erste praktische Verifizierung des Ansatzes wurde ein einfaches Produktbeispiel herangezogen, das aus wenigen Teilen besteht, aber gleichzeitig eine interdisziplinäre Dimension besitzt. Hierfür erschien eine medizinische Applikationsspritze geeignet. Für die Anwendung in einem deutlich komplexeren Projekt wurde die Entwicklung eines orthopädischen Instruments ausgewählt. Beide Projekte und das Vorgehen im Projekt werden beschreiben und anschließend hinsichtlich des Methodeneinsatzes analysiert. Daraus werden die Stärken, Schwächen und Optimierungspotentiale der Methode und des Tools herausgearbeitet.

6.1 Anforderungsklä rung für eine Applikationsspritze

Das Produkt erschien aufgrund seiner Einfachheit und der leicht zugänglichen Informationen für eine erste Verifikation gut geeignet. Das Entwicklerteam wurde durch zwei Studenten und den Verfasser dieser Arbeit als Beobachter repräsentiert. Ein Mediziner, Mitarbeiter eines Universitätsklinikums, stand als Experte zur Verfügung. Das Entwicklungsteam hatte keine medizinischen Vorkenntnisse. Dieses Beispiel ist aufgrund seiner geringen Komplexität ausführlicher beschrieben, um nicht nur das Ergebnis, sondern auch die Arbeit mit der Methode und deren Wirkungsweise darstellen zu können.

6.1.1 Projektbeschreibung

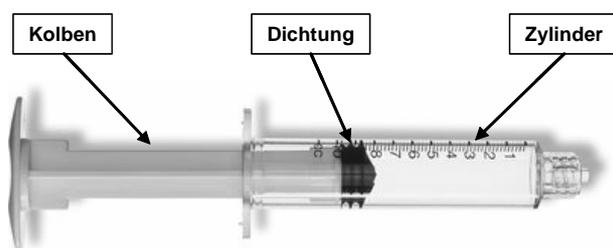


Abb. 6-1: Applikationsspritze

Eine Applikationsspritze besteht im Wesentlichen aus einem Zylinder, einem Kolben und einer Dichtung. Diese Spritzen kommen nicht nur am Patienten, sondern zum Beispiel auch im Labor zum Einsatz. Für die Anforderungsklä rung waren folgende Kundenanforderungen zu berücksichtigen, die in Interviews mit dem Mediziner (im Folgenden Experte genannt) ermittelt worden waren:

- Füllvolumen: 10ccm
- Einweganwendung
- Einsatz im Krankenhaus- und Laborbetrieb
- leichte Bedienung
- Ablesbarkeit
- Griffsicherheit
- sehr genaue Dosierbarkeit
- Spritze geeignet für verschiedenartige Fluide

Die erste Kundenanforderung wurde sogleich als quantifizierte Anforderung in das Anforderungssystem übernommen. Für die Relationen zur Klärung weiterer technischer Anforderungen wurde anschließend das System „Spritze“ aufgebaut. Als übergeordnete Systemelemente wurden die „Kundenanforderungen“, die „Applikationsspritze“ selbst, das zu spritzende „Fluid“, der „Anwender“ und das „Versuchsfeld“ identifiziert. „Anwender“, „Fluid“ und „Versuchsfeld“ wurden anhand von Merkmalen näher spezifiziert (Abb. 6-2). Im linken Teil des Bildes sind die oberste Ebene der Systemstruktur, im rechten Teil die Merkmale (Arzt, Pflegepersonal, Patient) für das angewählte Element „Anwender“ abgebildet. Das Element „Applikationsspritze“ wurde bereits in die Subelemente „Kolben“ und „Zylinder“ untergliedert.

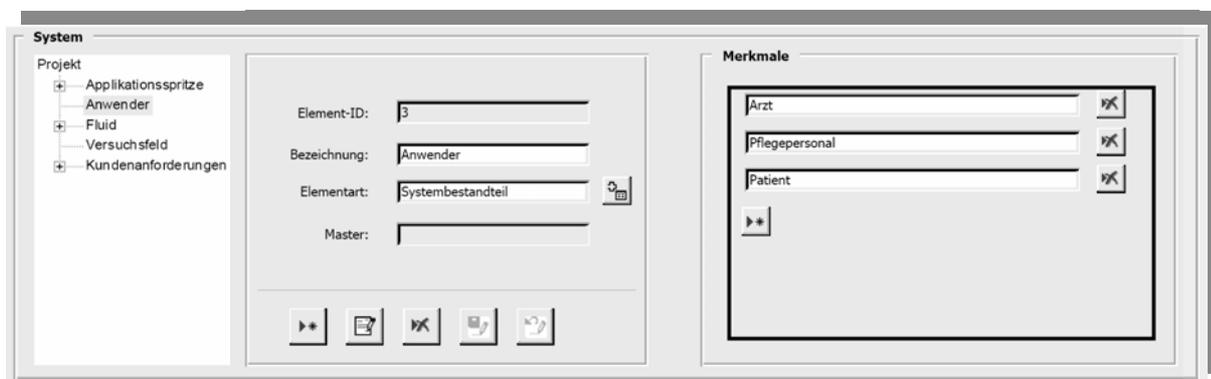


Abb. 6-2: Systemkomponenten und deren Ausprägungen (am Beispiel des Anwenders)

Nach dem Aufbau dieser ersten Systemstruktur wurden die Relationen zwischen den bisher bekannten Elementen ermittelt. Am Beispiel der Beziehungen zwischen Spritze und Anwender soll dies gezeigt werden. Zuerst stellte man sich die Frage „Hat die Spritze eine Relation zum Anwender?“. Da diese Frage positiv beantwortet werden konnte (der Anwender benutzt die Spritze), stellte man sich als nächstes die Frage, welcher Art die Relation sei. Hier kam die Leitlinie mit Relationsattributen zum Einsatz. Es wurde erkannt, dass die Spritze und der Anwender ergonomische (die Spritze muss gehalten und bedient werden können) und

kraftbezogene Relationen (die Kraft, die der Anwender auf die Spritze ausüben muss, hängt von der Gestaltung des Produkts ab) haben (Abb. 6-3). Diese Relation wurde entsprechend ihrer Richtung, Stärke und Art eingetragen.

The screenshot shows a window titled "Relationen" with two sections: "aktiv" and "passiv".

aktiv section:

Relationsart:	Relationsstärke:	Relation zu Element:	
ergonomisch	mittel	Zylinder	[Icon]
ergonomisch	mittel	Kolben	[Icon]

Below the active relations is a button with a double arrow icon.

passiv section:

Relationsart:	Relationsstärke:	Relation von Element:	
kraftbezogen	stark	Kolben	[Icon]
kraftbezogen	stark	Zylinder	[Icon]

Abb. 6-3: aktive und passive Relationen des Elements „Anwender“

Die zu Anfang ermittelte erste Struktur des Systems wurde während der voranschreitenden Entwicklung immer weiter untergliedert. Dazu wurden zum Beispiel einzelnen Kundenanforderungen als Elemente des Systems aufgenommen und das Element „Fluid“ durch seine physikalischen und chemischen Eigenschaften spezifiziert. Die auf übergeordneter Ebene bereits identifizierten Relationen wurden, falls zutreffend, auf die untergeordneten Elemente übertragen und weiterverarbeitet (Abb. 6-4).

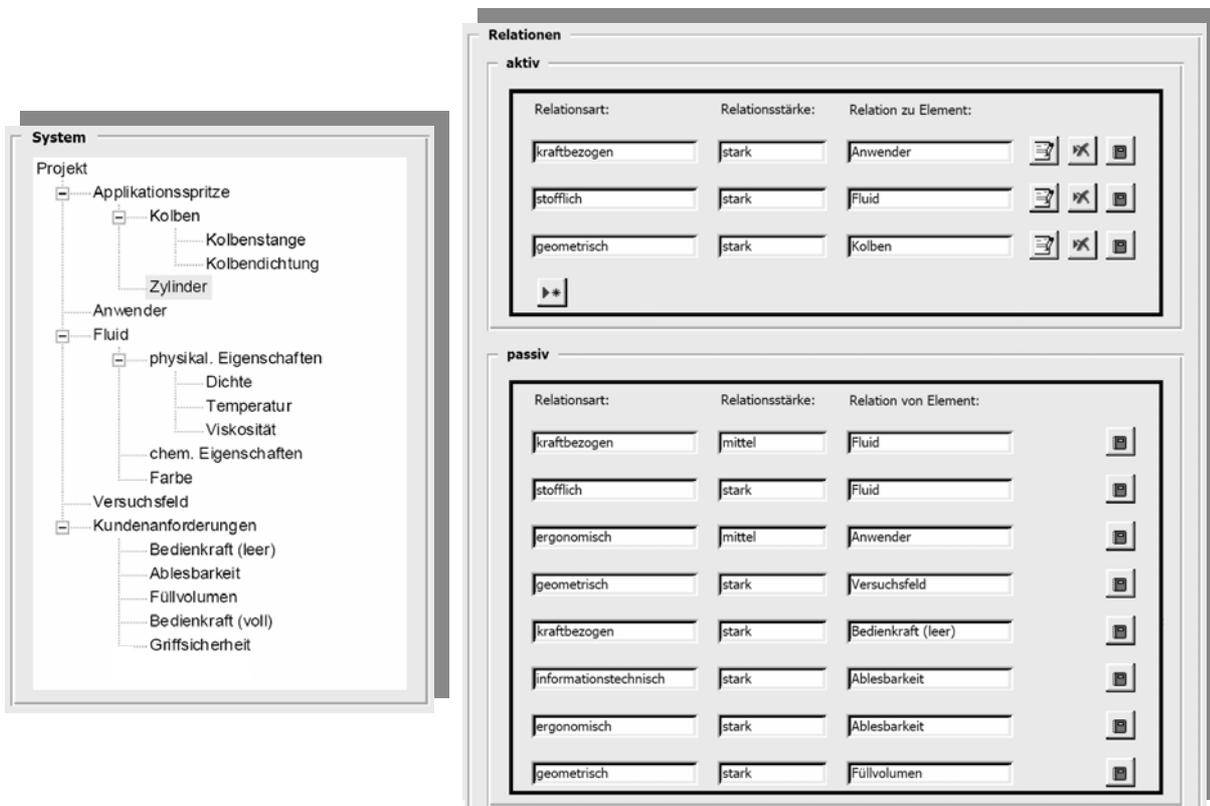


Abb. 6-4: Detaillierte Struktur des Systems "Spritze" und erweiterte Relationen des Elements „Zylinder“

Bis zu diesem Zeitpunkt waren gelegentliche telefonische oder persönliche Gespräche mit dem Experten ausreichend, um aufkommende Fragen zu klären. Parallel zur Systementwicklung sollten aber auch die aus den Relationen resultierenden Anforderungen ermittelt werden. Hierzu stellte sich das Entwicklungsteam zu jeder Relation die Frage, ob daraus eine Anforderung resultiere oder nicht. Aufgrund dieser Fragen wurde das Team sich sehr bald seiner Wissensdefizite im medizinischen Bereich bewusst. Um Abhilfe zu schaffen, wurde ein Arbeitstreffen mit dem Experten durchgeführt, in dem gemeinsam die anstehenden Fragen beantwortet und soweit als möglich Anforderungen abgeleitet werden sollten. Zu Beginn des Treffens wurde das bis dahin entwickelte System nochmals besprochen, um eine gemeinsame Ausgangsbasis zu schaffen. Anschließend wurden die bis dahin identifizierten Relationen gemeinsam hinsichtlich ihrer Ursächlichkeit für Anforderungen analysiert.

Merkmal:	Ausprägung:	Gewichtung:
Ausdrückgeschwindigkeit	Geschwindigkeit beeinflusst Ausdrückkraft	5
Kolbendurchmesser	Kolbendurchmesser beeinflusst Ausdrückkraft	10
Spritzendruck	Maximum hängt ab von Visko, Geschw., Düse	5
Spritzendruck	Maximum hängt von den Fluiden ab (Blut)	10

Abb. 6-5: Klären der Anforderungen am Beispiel der energetischen Relation zwischen Fluid und Kolben

Anhand der energetischen Relation zwischen Fluid und Kolben soll dieses Vorgehen erläutert werden (Abb. 6-5). Zunächst wurde die Leitlinie mit Relationsattributen zur Hand genommen und nach Assoziationen gesucht. Die stärkste Assoziation wurde bei allen Teilnehmern bei dem Begriff „Druck“ ausgelöst.

Zunächst wurde erkannt, dass durch den Kolben eine Kraft auf das Fluid ausgeübt wird, die sich in Form von Druck fortpflanzt. Deshalb wurde die Anforderung „Spitzendruck“ formuliert. Als man die dazugehörige Ausprägung definieren wollte, erkannte der Experte, dass die Ausprägung nicht so einfach festzulegen sei, da sie vom betrachteten Fluid abhängen kann (für die Spritze war eine Eignung für unterschiedliche Fluide gefordert). So sollte zum Beispiel Blut nicht plötzlich zu hohem oder niedrigem Druck ausgesetzt werden, da dies zur Schädigung der Blutkörperchen führen kann. Ob dies auch für andere Fluide gelte, konnte keiner der Beteiligten sagen, so dass eine vorläufige Ausprägung „Maximum hängt von den Fluiden (Blut) ab“ formuliert und mit der Relation verknüpft wurde. Durch diese Relation und die entsprechende Assoziation konnte also beim Experten eine Information angestoßen und abgerufen werden. Gleichzeitig wurde bei allen Beteiligten eine Wissenslücke aufgedeckt. Die fehlenden Informationen sollten im weiteren Verlauf eingeholt werden.

Im Rahmen der angesprochenen Diskussionen wurde ein weiterer wichtiger Zusammenhang aufgedeckt, der den Ingenieuren des Teams bewusst war, dem fremddisziplinären Experten aber erst durch die Diskussion des Systems und seiner Relationen verdeutlicht werden konnte: der in der Spritze beim Ausdrücken entstehende Druck hängt maßgeblich von dem Kolbendurchmesser, der Ausdrückgeschwindigkeit, der Viskosität der Fluide und dem Durchmesser der Austrittsöffnung (Anforderung „Schnittstelle Spritze Innendurchmesser“) ab. Die Folge dieser Erkenntnis war sehr interessant: der Experte bestätigte, dass er sich noch nie Gedanken über die Geschwindigkeit mit der der Kolben bewegt wird (Ausdrückgeschwindigkeit) gemacht hat, sondern dies intuitiv machen würde, räumte aber ein, dass man dies nicht mit beliebig hoher Geschwindigkeit machen würde, „man hätte das halt im Gefühl“. Die Bedeutung dieser Information war ihm aber durchaus bewusst. Daher wurden orientierende Versuche unternommen, um ein Gefühl für die Ausdrückgeschwindigkeit zu entwickeln. Bei diesen Versuchen wurde neben der Ausdrückgeschwindigkeit gleichzeitig die dabei aufgewendete Kraft gemessen. Die daraus ableitbaren Anforderungen „Ausdrückkraft“ und „Ausdrückgeschwindigkeit“ wurden in das Anforderungssystem übernommen.

In ähnlicher Form wie beschrieben wurden auch alle weiteren Relationen bearbeitet. Dabei entstand quasi als Zwischenergebnis die unten abgebildete Anforderungsliste (Tab. 6-1) mit vorläufigen Ausprägungen und den verknüpften Relationen. Die darin enthaltenen unterschiedlichen, teils widersprüchlichen Ausprägungen der Anforderungen wurden im weiteren Verlauf zusammengefasst, gegeneinander abgewogen und abgeglichen, um die endgültigen Ausprägungen festzulegen. Der in Tab. 6-2 dargestellte Stand der Anforderungen an eine Applikationsspritze ist natürlich nicht der endgültige Stand. Zum Beispiel ist die Frage des Werkstoffs noch nicht endgültig geklärt (hierzu ist noch keine Ausprägung eingetragen) genauso gilt es noch weitere Anforderungen bezüglich der Ablesbarkeit zu klären (zum Beispiel hängt von der Farbe der zu behandelnden Fluide die Farbe der Skaleneinteilung ab).

Nr.	Merkmal	vorläufige Ausprägung	Gewichtung	Relation von	Relationsart	Relationsstärke	Relation zu
1	Energie						
1	Ausdrückgeschwindigkeit	Geschwindigkeit beeinflusst Ausdrückkraft	5	Fluid	energetisch	mittel	Kolben
1	Ausdrückgeschwindigkeit	0.4 ml/s	5	Fluid	energetisch	mittel	Kolben
2	Spritzendruck	Maximum hängt ab von Visko. Geschw., Düse	5	Bedienkraft (voll)	kraftbezogen	stark	Zylinder
2	Spritzendruck	Maximum hängt ab von Visko. Geschw., Düse	5	Bedienkraft (voll)	kraftbezogen	stark	Kolben
2	Spritzendruck	Maximum hängt ab von Visko. Geschw., Düse	5	Fluid	energetisch	mittel	Zylinder
2	Spritzendruck	Maximum hängt ab von Visko. Geschw., Düse	5	Fluid	energetisch	mittel	Kolben
2	Spritzendruck	Maximum hängt von den Fluiden ab (Blut)	10	Fluid	energetisch	mittel	Zylinder
2	Spritzendruck	Maximum hängt von den Fluiden ab (Blut)	10	Fluid	energetisch	mittel	Kolben
2	Ergonomie						
1	Ablesewinkel	+/- 45°	7	Ablesbarkeit	ergonomisch	stark	Zylinder
2	Kolbenschnittstelle Mensch	Anpassung an Daumen (abnutschicher)	5	Anwender	ergonomisch	mittel	Kolben
3	Zylinderschnittstelle (Mensch)	Anpassung an Zeige- und Mittelfinger	5	Anwender	ergonomisch	mittel	Zylinder
3	Geometrie						
1	Füllvolumen	10ml	10	Füllvolumen	geometrisch	stark	Zylinder
2	Kolbendurchmesser	entspricht Zylinderdurchmesser	10	Zylinder	geometrisch	stark	Kolben
2	Kolbendurchmesser	Kolbendurchmesser beeinflusst Ausdrückkraft	10	Fluid	energetisch	mittel	Kolben
3	Schnittstelle Spritze Innendurchmesser	2mm	10	Versuchsfeld	geometrisch	stark	Zylinder
4	Schnittstelle Spritzenausritt	Luer Lock	10	Versuchsfeld	geometrisch	stark	Zylinder
5	Zylinderinnendurchmesser	entspricht Kolbendurchmesser	10				
6	Zylinderlänge (Nutzlänge)	$=V/(P \cdot Zdr^2)$	10	Füllvolumen	geometrisch	stark	Zylinder
4	Information						
1	Ablesgenauigkeit	0.2 +/- 0.2 ml	10	Ablesbarkeit	informationstechnisch	stark	Zylinder
5	Kraft						
1	Ausdrückkraft	max 5.5N	5	Bedienkraft (voll)	kraftbezogen	stark	Kolben
1	Ausdrückkraft	max 5.5N	5	Zylinder	kraftbezogen	stark	Anwender
1	Ausdrückkraft	max 5.5N	5	Kolben	kraftbezogen	stark	Anwender
1	Ausdrückkraft	max 5.5N	5	Bedienkraft (voll)	kraftbezogen	stark	Zylinder
2	Ausdrückkraft (leer)	3N	8	Bedienkraft (leer)	kraftbezogen	stark	Zylinder
2	Ausdrückkraft (leer)	3N	8	Bedienkraft (leer)	kraftbezogen	stark	Kolben
2	Ausdrückkraft (leer)	3N	8	Zylinder	kraftbezogen	stark	Anwender
2	Ausdrückkraft (leer)	3N	8	Kolben	kraftbezogen	stark	Anwender
3	Reibungskraft	entspricht Ausdrückkraft leer	8	Bedienkraft (leer)	kraftbezogen	stark	Zylinder
6	Stoff						
1	Werkstoff	Fluid darf durch Werkstoff nicht verändert werden	10	Kolben	stofflich	stark	Fluid
1	Werkstoff	Fluid darf durch Werkstoff nicht verändert werden	10	Zylinder	stofflich	stark	Fluid
1	Werkstoff	Werkstoff darf von Fluid nicht angegriffen werden	10	Fluid	stofflich	stark	Zylinder
1	Werkstoff	Werkstoff darf von Fluid nicht angegriffen werden	10	Fluid	stofflich	stark	Kolben

Tab. 6-1: Anforderungsliste der Spritze mit vorläufigen Ausprägungen und dazugehörigen Relationen

Nr.	Merkmal	Kurzvariable	Ausprägung		Einheit (phys.)	Gewichtung	Ursprung/ Erläuterung	Anlage	Ersteller	Datum
			min.	max.						
1	Energie									
	1/Ausdrückgeschwindigkeit	V_voll	0,4		ml/s	3,00	Messung		Ju	11.11.2004
	2/Spritzendruck	p_i		500	mmHg	10,00			Ju	11.11.2004
	Ergonomie									
2	1/Ablesewinkel				°	9,00			Ju	11.11.2004
	2/Kolbenschnittstelle Mensch			männl. Daumen 95. Perzentil		3,00			Ju	11.11.2004
	3/Spritzenweite			130,00	mm	9,00			Ju	11.11.2004
	4/Zylinderschnittstelle (Mensch)			männl. Zeige- und Mittelfinger 95. Perzentil		9,00			Ju	11.11.2004
3	Geometrie									
	1/Füllvolumen	V	10		ml	10,00			Ju	11.11.2004
	2/Kolbendurchmesser	Kd	5,5		mm	9,00	Messung		Ju	11.11.2004
	2/Kolbendurchmesser	Kd		5,5	mm	9,00	Messung		Ju	11.11.2004
	3/Schnittstelle Spritze Innendurchmesser	d_i				10,00	ISO 594-1 (1986)		Ju	11.11.2004
	4/Schnittstelle Spritzenaustritt	Luer Lock				10,00	ISO 594-1 (1986)		Ju	11.11.2004
4	5/Zylinderinnendurchmesser	Zd	5,5		mm				Ju	11.11.2004
	6/Zylinderlänge (Nutzlänge)	Zl	106		mm				Ju	11.11.2004
	Information									
	1/Ablesegenauigkeit		0,2 +/- 0,1		ml	9,00			Ju	11.11.2004
5	Kraft									
	1/Ausdrückkraft	F_voll		5,5	N	9,00	Zusätzlich zur Ausdrückkraft (leer)		Ju	11.11.2004
	2/Ausdrückkraft (leer)	F_leer		3	N	3,00			Ju	11.11.2004
6	3/Reibungskraft	F_reib			N	3,00	Entspricht der Ausdrückkraft (leer)		Ju	11.11.2004
	Stoff									
1	Werkstoff								Ju	11.11.2004

Tab. 6-2: Anforderungsliste mit den endgültigen Ausprägungen (dargestellt ohne die Angaben zur Änderungshistorie)

6.1.2 Erfahrungen

Die Erfahrungen in diesem Projekt zeigten, dass gerade durch die erzwungene Auseinandersetzung mit dem System und dessen inneren Zusammenhängen sehr schnell eine große Systemkompetenz der Beteiligten erzielt werden konnte. Da diese Form der Beschäftigung mit der Entwicklungsaufgabe stets im Team und zusammen mit dem Experten stattfand, wurden Missverständnisse aufgrund fachlicher Unterschiede schnell ausgeräumt, so dass man zu einem gemeinsamen Systemverständnis und Sprache gelangte.

Durch die Suche und Diskussion der Relationen zwischen den Elementen und der Frage nach daraus resultierenden Anforderungen konnten die notwendigen Wissensbereiche angestoßen werden, um Anforderungen zu formulieren. Vor allem wurden auch Anforderungsbereiche aufgedeckt, die wohl bei einer konventionellen Anforderungsklä rung unentdeckt geblieben wären. Es ist zum Beispiel fraglich, ob das Entwicklungsteam ohne die dazugehörigen Relationen die Anforderung „Spitzendruck“ erkannt hätte beziehungsweise ob der Experte diese Anforderung ohne die Relationen angesprochen hätte. Zumindest kann festgehalten werden, dass eine solche Anforderung im Rahmen der ersten Gespräche, in denen auch die oben angeführten Kundenanforderungen ermittelt wurden, von Expertenseite nicht zur Sprache kam und von Entwicklerseite implizit für nicht notwendig gehalten wurde.

Weiterhin wurde vom Entwicklungsteam als äußerst positiv die gleichzeitig entstehende Dokumentation des Systems und der Anforderungen aufgenommen. Vor allem die feste Verknüpfung jeder Anforderung mit ihren Relationen bewährte sich, da bei Diskussionen einzelner Anforderungen immer auf die Ursache(n) derselben zurück geschlossen werden konnte. Dies ermöglichte wesentlich zielgerichtete Diskussionen, als sie aus bisherigen Projekten bekannt waren. Besonders konnten die Auswirkungen von Änderungen am System beziehungsweise an Anforderungen sehr schnell und mit einer hohen Zuverlässigkeit überblickt werden. Zum Beispiel konnte das Entwicklungsteam die Folgen der Änderung des zulässigen Spitzendrucks für die Geometrie der Spritze (Länge und Durchmesser) und damit auch für den Spritzenweg und die notwendige Spreizung der Hand zum Bedienen der Spritze schnell überblicken. Mittels der an den Spitzendruck assoziierten Relationen konnten die von der Änderung betroffenen Elemente und über deren Relationen wiederum weitere Elemente ermittelt werden: Eine beliebige Reduzierung des zulässigen Spitzendrucks in der Spritze wäre nicht zulässig gewesen, da aufgrund der damit verbundenen Querschnittsreduzierung des Kolbens der Spritzenweg und damit die notwendige Spreizung der Hand die zulässigen Werte überschritten hätte.

Der zeitliche Aufwand für die Erstellung dieses Systems und die Diskussion der Relationen und Anforderungen nahm in Summe etwa 5 Stunden in Anspruch und war in den Augen des Verfassers vom Nutzen-Aufwand Verhältnis mehr als gerechtfertigt.

6.2 Entwicklung einer Templatevorrichtung für chirurgische Gelenkkorrekturen

Im Gegensatz zum vorangegangenen Beispiel liegt der Schwerpunkt der Schilderung weniger auf dem Vorgehen und der Arbeit mit Methode und Werkzeug sondern vielmehr auf dem Ziel und dem Nutzen des Methodeneinsatzes. Anhand ausgewählter Anforderungen wird der Gewinn der Methode herausgestrichen.

6.2.1 Projektbeschreibung

Bei der intertrochantären Umstellungsosteotomie wird zum Beispiel bei Fehlstellungen der Hüfte der Hüftkopf (Caput femoris) vom Oberschenkelknochen (Femur) getrennt und nach Entfernen eines Knochenkeils auf diesem neu orientiert fixiert (siehe Abb. 6-6). Auf diese Weise kann das Caput femoris im Gelenk neu eingestellt werden. Im Idealfall lässt sich so die Implantation einer Hüftendoprothese vermeiden. Durch die gewünschte Endposition des Caput femoris auf dem Femur wird die Lage und Ausrichtung der Schnittflächen des zu entfernenden Keils bestimmt. Der Operationserfolg wird maßgeblich von der Genauigkeit dieses Vorgangs bestimmt. Derzeit hängt diese Genauigkeit überwiegend von dem Augenmaß, dem räumlichen Vorstellungsvermögen und dem handwerklichen Geschick des Chirurgen ab. Er muss sich die gewünschte Position des Caputis Femoris gedanklich vorstellen können, daraus die Geometrie des zu entfernenden Keils ableiten und mit Winkelschablonen sowie Augenmaß und freihändig diese Informationen auf den Knochen übertragen.

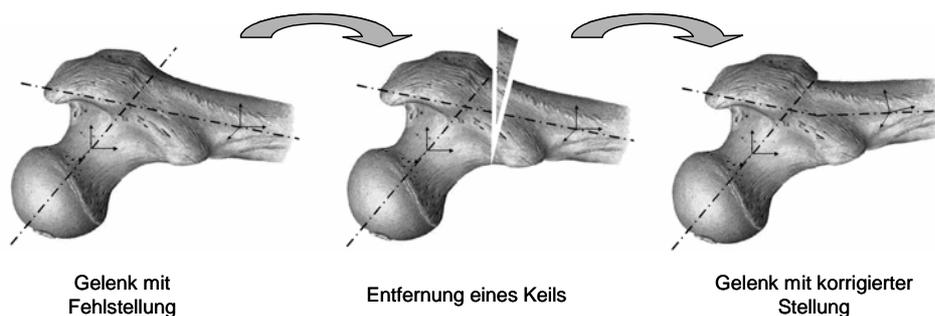


Abb. 6-6: Prozessschritte der intertrochantären Umstellungsosteotomie

Um diesen Prozess zu verbessern (siehe Abb. 6-7), ist eine Software entwickelt worden, die mittels zwei navigierter Fluoroskopie-Aufnahmen des betroffenen Gelenkes ein vereinfachtes Geometriemodell des proximalen Femurs einliest und daraus ein dreidimensionales Bild errechnet. Anschließend wird dem Chirurgen ermöglicht, die gewünschte neue Position des Gelenkes in der Software zu definieren. Auf dieser Basis kann die Software die Schnittkoordinaten für den zu entfernenden Keil berechnen.

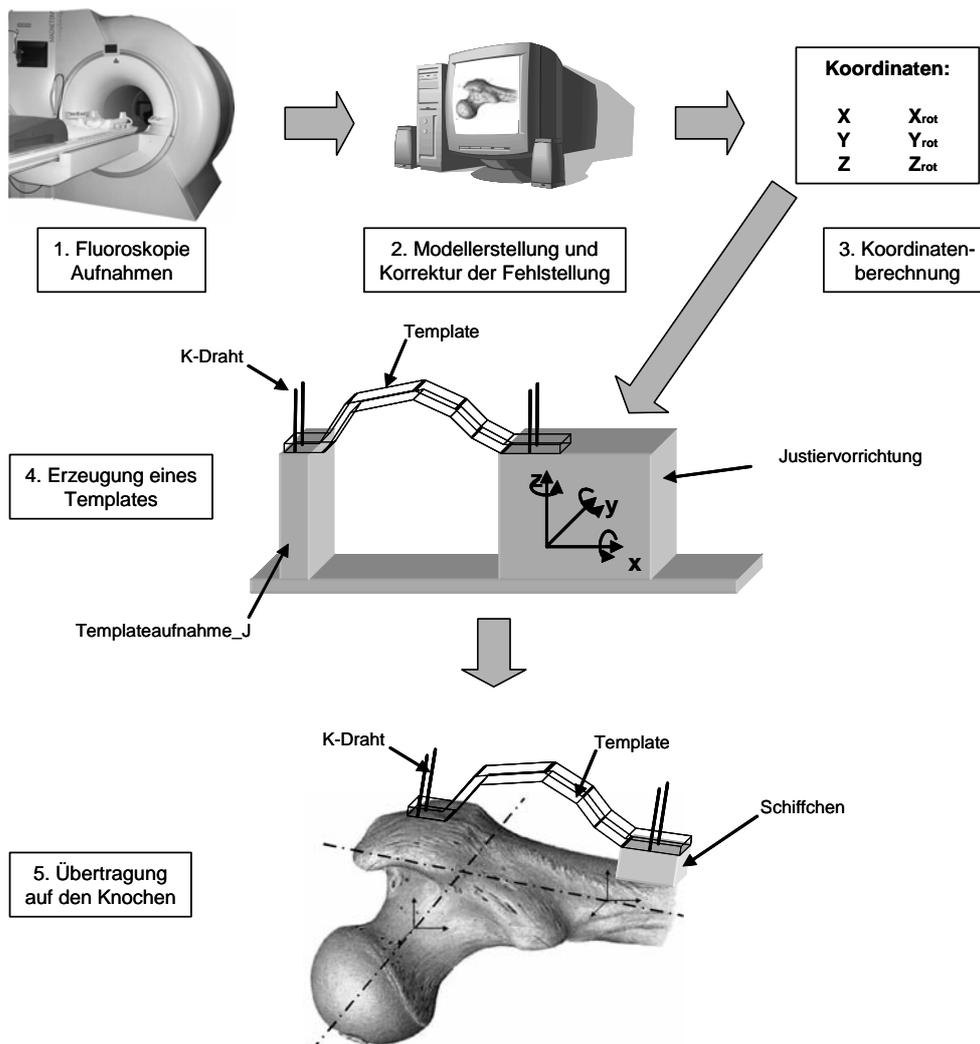


Abb. 6-7: Übertragung der Schnittkoordinaten auf den Patienten

Die Schwierigkeit besteht aber darin, die virtuellen Schnittflächen auf den realen Knochen zu übertragen, da zur eindeutigen Festlegung von Lage und Position 6 Freiheitsgrade übertragen werden müssen (drei Rotationen, drei Translationen). Um die Übertragung der Koordinaten auf den Knochen zu ermöglichen, ist vom Projektpartner der Prototyp einer 6 DoF Template-

Vorrichtung³⁸ für die intertrochantäre Umstellungsosteotomie entwickelt und aufgebaut worden. Die Template-Vorrichtung besteht aus den Elementen Justiervorrichtung, Template und Templateaufnahme. Letzteres ist im Gerät fixiert und kann weder seine Position noch Lage verändern. An dem Gegenstück wird das Template mittels zweier Knochendrähte (K-Drähte) befestigt. Diese Verbindung entspricht einer Stiftung. Das Template selber kann in seiner Geometrie in sechs Achsen verändert und fixiert werden. An der Justiervorrichtung wird das Template ebenfalls mittels K-Drähten befestigt. Der Befestigungspunkt an der Justiervorrichtung kann mittels sechs manuell verstellbaren Achsen in Position und Lage verändert werden. Hierfür werden die Koordinaten aus der Software benötigt. Sind die Koordinaten eingestellt, können die Achsen des Templates fixiert werden und dasselbe aus der Vorrichtung entnommen werden. Anschließend wird das Template mittels K-Drähten an einem am Femur befestigten und vermessenen Bauteil (Schiffchen) befestigt und dient mit seinem anderen Ende als Schablone für weitere Arbeitsschritte. Keines der angesprochenen Bauteile verbleibt im Patienten.

Der vorhandene Prototyp der Template-Vorrichtung erfüllte aber noch nicht die Anforderungen für einen operativen OP-Einsatz (mangelnde Stabilität, Genauigkeit, Kalibrierbarkeit, Sterilisierbarkeit etc.). Im Rahmen dieses Entwicklungsprojektes wurde der existierende Prototyp optimiert und weiterentwickelt, um durch Vermeidung der Nachteile des Prototyps einen operativen Einsatz zu Versuchszwecken zu ermöglichen.

Bei diesem Projekt handelte es sich zwar um eine Neuentwicklung, da vergleichbare Produkte am Markt noch nicht verfügbar waren, es lagen aber bereits sehr viele detaillierte Informationen und erste Erfahrungen aus Versuchen vor. Der Prototyp und das darin verwirklichte Funktionsprinzip sind von Seiten der medizinischen Kooperationspartner, die ein umfangreiches technisches Wissen vorwiesen, entwickelt, aufgebaut und in Versuchen verifiziert worden. Das Entwicklungsteam bestand aus einem Diplomanden, einem Mediziner und zwei betreuenden Ingenieuren, einer davon der Verfasser. Der mit der Optimierung des Prototyps betraute Diplomand (im Folgenden Entwickler bezeichnet) konnte nur geringe medizinische Vorkenntnisse aufweisen. Dieses Projekt weist im Vergleich zur Applikationsspritze insgesamt stark veränderte Randbedingungen auf. Durch den Einsatz der relationalen iterativen Anforderungsklä rung und des unterstützenden Werkzeugs RIA in diesem Vorhaben sollte gezeigt werden, dass auch mit einem bei Entwicklungsbeginn hohen Informationsstand ein gewinnbringender Einsatz der Methode möglich ist. Folgenden Nutzen versprach man sich von dem Einsatz:

- Schnelle Einarbeitung des Entwicklers in die unbekannte Problematik und das medizinische Fachgebiet
- Rasche Entwicklung einer „gemeinsamen Sprache“ im Entwicklungsteam
- Überprüfung der vorhandenen Anforderungen auf Richtigkeit und Vollständigkeit
- Aufdecken von zusätzlichen, bisher nicht bedachten Anforderungen

³⁸ Mit der Templatevorrichtung wird eine Vorlage (Template) zur Übertragung der Informationen auf den Knochen erstellt. Das Gerät selbst muss nicht auf oder an dem Patienten gebracht werden.

- Fortsetzung der Entwicklung mit der Unterstützung von RIA

Aufgrund der bereits vorliegenden Informationen gestaltete sich das Vorgehen anders als im vorangegangenen Beispiel. Einerseits wurden die bereits explizit vorhanden Anforderungen zusammengetragen und in das Anforderungssystem von RIA eingespeist. Zum anderen wurden die grundsätzliche Funktionsweise und der Aufbau des bestehenden Prototyps analysiert und abstrahiert, um ausgehend von diesen Informationen das Ursprungssystem in RIA aufzubauen. Da das Funktionsprinzip beibehalten werden sollte, stand somit zu Beginn des Projektes bereits ein recht detailliertes Ursprungssystem zur Verfügung. Für diese Tätigkeiten wurden zu Beginn Interviews beziehungsweise Diskussionen mit den Kooperationspartnern am Prototyp geführt, um die grundlegende Problematik der gestellten Aufgabe, die Funktionsweise und den Aufbau des Prototyps zu verstehen. Als wichtige Elemente, die nicht zum Produkt selbst gehören, wurden der Patient, der Bediener, die Sterilisation und die oben angesprochen Software identifiziert. Die gewonnenen Informationen wurden soweit nötig abstrahiert und als Systemstruktur (siehe Abb. 6-8) in RIA eingepflegt. Gemeinsam wurden die Relationen des Systems analysiert und mit Anforderungen verknüpft.

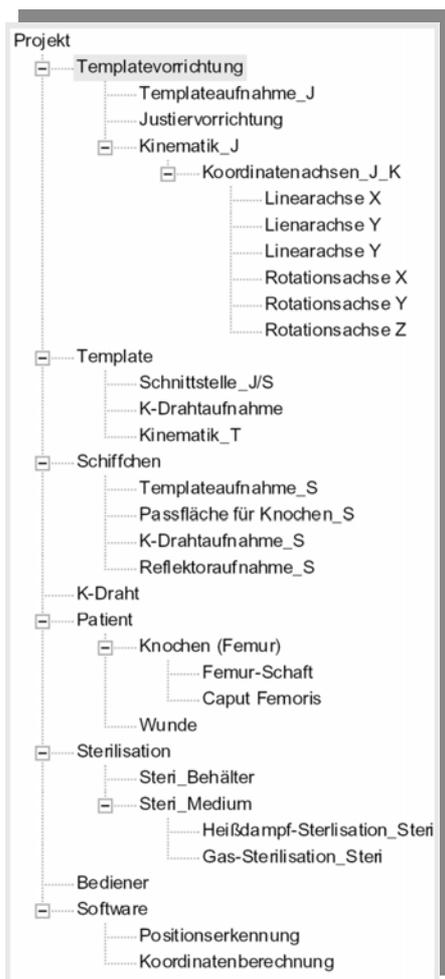


Abb. 6-8: Systemstruktur der Templatevorrichtung

Auf diese Weise wurden zum abgebildeten System 41 Relationen und etwa 20 vorhandene Anforderungen ermittelt. Im nächsten Schritt wurde versucht aus den Relationen Anforderungen abzuleiten. Dazu wurde auf zweierlei Art vorgegangen: Zum einen wurden die Relationen mit den vorhandenen Anforderungen verknüpft. Gleichzeitig wurden diese Anforderungen auf ihre formale und inhaltliche Richtigkeit untersucht. Besonderes Augenmerk wurde dabei auf die Ausprägungen der Anforderungen gelegt. Zu einigen Anforderungen konnten weitere Ausprägungen abgeleitet werden, die später mit den bestehenden abgeglichen werden mussten. Parallel wurde mittels der Relationen nach fehlenden Anforderungen gesucht. In diesem Schritt wurden 15 bisher unberücksichtigte Anforderungen erkannt. Diese betrafen vor allem die Sterilisation und Bedienung/Wartung. Insgesamt wurden durch dieses Vorgehen 45³⁹ endgültige Anforderungen identifiziert. An einigen ausgewählten Beispielen sollen weitere Mechanismen der Methode erläutert werden.

Zum Beispiel existiert eine Anforderung „Verstellkraft des Templates“, die im geöffneten Zustand des Templates möglichst gering sein soll (1N). Für den fixierten Zustand war von dem Kooperationspartner gefordert worden, dass das Template „vollständig blockiert“ sein solle. Erst durch die Relationen des Templates mit dem Knochen des Patienten und dem Bediener wurde aufgedeckt, dass eine solche Forderung zur zusätzlichen Schädigung des Patienten führen könnte, sollte auf das am Patienten befestigte Template zu viel Kraft ausgeübt werden (zum Beispiel beim Bohren der K-Drähte). Deshalb wurde eine zusätzliche Ausprägung „100N im blockierten Zustand“ eingeführt, die später in eine eigene Anforderung überführt wurde.

Vor allem die Sterilisation war von den Kooperationspartnern bisher nicht berücksichtigt worden. Allen Beteiligten war klar, dass die Elemente, die direkten mit dem Patienten in Berührung kommen würden voll sterilisierbar sein sollten. Dass dies auch für die Templatevorrichtung selbst gelten muss und welche Folgen dies haben würde wurde erste in Folge der Relationenanalyse und Anforderungsklä rung klar. Man war bisher davon ausgegangen, dass eine Desinfektion in Verbindung mit einer sterilen Folieneinhüllung ausreichend und geeignet wäre. In Folge der Relationen wurde nämlich die Frage nach in Frage kommenden Sterilisationsverfahren und deren Eigenschaften (Temperaturen, Drücke, Dauer, verwendete Stoffe etc.) gestellt. Es wurde sehr schnell klar, dass diese sowohl auf die Wahl der Werkstoffe als auch auf die Wahl der Zukaufteile massiven Einfluss haben würden. Beispielhaft sind nachfolgen die unterschiedlichen Ausprägungen zur Anforderung „Werkstoff Template“ aufgeführt.

³⁹ Die Summe der vorläufigen Anforderungen, also alle Merkmale mit ihren unterschiedlichen mit Relationen verknüpften Ausprägungen betrug 75.

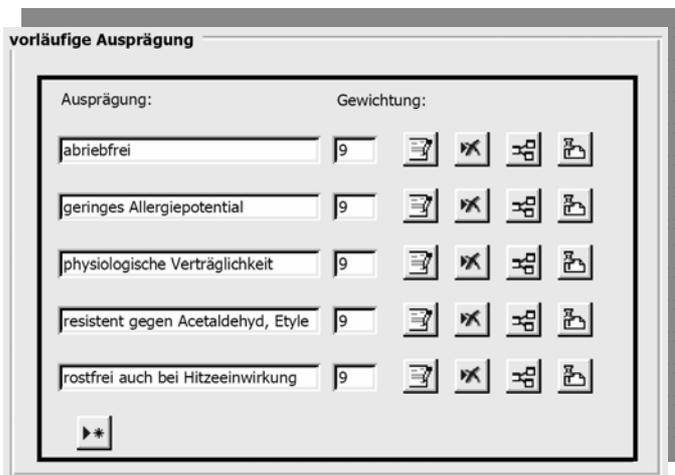


Abb. 6-9: Vorläufige Ausprägungen der Anforderung „Werkstoff Template“

Mehrfach ist angemerkt worden, dass Anforderungen lösungsabhängig aufkommen beziehungsweise sich verändern und beeinflusst werden. Dies kann anhand der Sterilisation sehr gut nachvollzogen werden. Aufgrund der sehr unterschiedlichen Randbedingungen der Dampf- und Gassterilisation musste eine Entscheidung für eines der beiden Verfahren getroffen werden. Diese Entscheidung hat aber wiederum Einfluss auf die Anforderungen an das Produkt. So sind die Temperaturen und Drücke bei der Dampfsterilisation wesentlich höher als bei der Gassterilisation. Dafür kommen bei der Gassterilisation giftige Gase zum Einsatz, die die Eigenschaft haben, sich zum Beispiel in Kunststoffe und vor allem Schmierstoffe einzulagern und dann nur sehr langsam ausgasen. Deren Restkonzentration darf aber nur noch sehr gering sein, bevor das sterilisierte Gut wieder der Verwendung zugeführt werden darf. Eine Gassterilisation erfordert deshalb den Einsatz flüchtiger Schmierstoffe, eine Dampfsterilisation den Einsatz nicht flüchtiger, um ein Eindringen von Wasser in geschmierte Bereiche zu verhindern. Diese Zusammenhänge sind aufgrund der Anlage der Methode und des Tools RIA sehr früh zu Tage getreten, so dass das Team sich bald aufgrund der einfacheren Handhabbarkeit und der geringeren Kosten für die Dampfsterilisation entschied. In der Folge wurden die Ausprägungen an die neue Lösungssituation angepasst.

In ähnlicher Weise wie beschrieben wurden sämtliche Relationen und Anforderungen bearbeitet, bereinigt und erweitert, so dass bei Freigabe des Gesamtkonzeptes, mit Beginn der Detailkonstruktion und der Auswahl von Zukaufteilen die Anforderungsliste aus etwa 80 Anforderungen zu den unterschiedlichen Bereichen bestand. Im Rahmen der weiteren Entwicklung wurde RIA weiterhin eingesetzt und diente sowohl der Erfassung von weiteren, zunehmend lösungsfixierten Anforderungen als auch der Dokumentation derselben und des Systems.

6.2.2 Erfahrungen

Auch wenn die relationale iterative Anforderungsklä rung erst zu einem späten Zeitpunkt in das Projekt einbezogen worden ist, haben sich der Einsatz und die damit zwischenzeitliche zusätzliche Arbeit dennoch rentiert.

Da das Entwicklungsteam bei dem Entwurf des Prototyps noch nicht eingebunden war, bot der erzwungene Systemaufbau in RIA die Möglichkeit, das zugrunde liegende Problem schnell zu durchdringen und gleichzeitig das vorliegende Konzept und dessen Anforderungen kritisch zu hinterfragen. Das Entwicklungsteam konnte sich so rasch das notwendige fremddisziplinäre Wissen und das bereits vorhandene Wissen über die Lösung der Entwicklungsaufgabe aneignen.

Gleichzeitig konnten die Projektpartner eine gemeinsame Sprache und ein gemeinsames Systemverständnis entwickeln. Beispielsweise wurden von den Kooperationspartnern, die den Prototypen entwickelt hatten, in Vorgesprächen immer wieder unterschiedliche Bezeichnungen für Bauteile oder sonstige Elemente des Systems verwendet, was natürlich zu entsprechenden Verwirrungen führte. Im Zuge der gemeinsamen Systemabstraktion mussten gemeinsame Begriffe gefunden beziehungsweise Definitionen getroffen werden, die mit dem Ursprungssystem in RIA dann auch dokumentiert waren und als Basis für alle weiteren Diskussionen dienen konnten.

Die gemeinsame Diskussion der Relationen des Systems hatte zur Folge, dass den Kooperationspartnern die inneren Zusammenhänge nochmals vor Augen geführt wurden und das Entwicklungsteam sehr schnell auf denselben Stand gelangte. Durch die sich anschließende Anforderungsklä rung wurden die vorhandenen Anforderungen nochmals überprüft und neue hinzugenommen. Hier kam der Kern der relationalen iterativen Anforderungsklä rung zum tragen: Erst jetzt wurde zum Beispiel erkannt, welcher großen und eben widersprüchlichen Einfluss die unterschiedlichen Sterilisationsverfahren auf die Entwicklung haben. Dies ist vorher keinem der Beteiligten in dieser Form bewusst gewesen und wäre ohne diese Systematik wohl erst später bekannt geworden. Ebenso konnte eine Vielzahl von Anforderungen bereinigt beziehungsweise korrigiert werden, da sie vorher falsch oder missverständlich waren.

Zusammenfassend kann man sagen, dass der Einsatz der relationalen iterativen Anforderungsklä rung auch in Entwicklungssituationen mit zu Beginn bereits hohem Informationsstand zielführend ist, da zum einen eine schnelle Systemdurchdringung erleichtert wird, zum anderen auch hier sich zeigte, dass meist Anforderungen vergessen beziehungsweise übersehen worden sind und diese durch den Methodeneinsatz schnell identifiziert werden konnten.

6.3 Auswertung des Methodeneinsatzes

Der Einsatz der relationalen iterativen Anforderungsklä rung soll im Folgenden nochmals zusammenfassend betrachtet werden, um daraus sowohl Vor- und Nachteile, als auch Optimierungspotential ableiten zu können.

Zunächst sei angemerkt, dass keine Methode und kein Werkzeug, auch nicht die relationale iterative Anforderungsklä rung und RIA, eine Anforderungsklä rung „durchführen“ kann, sie kann immer nur eine Unterstützung darstellen. Weder die relationale iterative Anforderungsklä rung noch RIA kann den Entwickler von seiner sorgfältigen Arbeit und einem intensiven Kontakt mit dem Kunden oder Experten entbinden, vielmehr unterstützen und leiten sie den Entwickler dabei. Gerade das ist die Stärke der relationalen iterativen Anforderungsklä rung. Zum einen gibt sie dem Entwickler einen Weg und ein Instrumentarium vor, mit Experten einen Kontakt aufzubauen und eine gemeinsame Sprache und ein gemeinsames Systemverständnis zu entwickeln, andererseits zwingt sie ihn auch diesen Weg einzuschlagen. Gerade durch dieses vorgeschriebene strukturierte Denken und die schrittweise Analyse des Systems werden Problempunkte und Wissenslücken, auch implizite, identifiziert und beschrieben. Fragen können abgeleitet werden und Wissenslücken damit geschlossen werden. So ist zum Beispiel der zulässige Spitzendruck in einer Applikationsspritze nicht berücksichtigt worden, die Notwendigkeit dieser Anforderung ist erst bei der Analyse des Systems aufgekommen und auch da erst dem medizinischen Kooperationspartner bewusst geworden, konnte also rechtzeitig expliziert werden. Noch deutlicher trat dieser Vorteil im Falle der Templatevorrichtung zutage: Von Seiten der Kooperationspartner war die Sterilisation bisher nicht gesondert berücksichtigt worden. Das Entwicklungsteam besaß zu wenig Erfahrung, um einerseits die deutlich unterschiedlichen Randbedingungen der Sterilisationsverfahren zu kennen, zum anderen deren Auswirkungen beurteilen zu können. Erst durch die Relationenanalyse wurden die widersprüchlichen Anforderungen aus den beiden Verfahren offensichtlich, so dass man sich für die Dampfsterilisation entschied und damit Zielkonflikte zu einem frühen Zeitpunkt ausschließen konnte.

Von großer Bedeutung sind die Art und der Umfang der Ausführung (Begrifflichkeiten, Aufbau, Abstraktionsgrad etc.) des betrachteten Systems, die stets eine sehr individuelle Sicht des Entwicklungsteams auf das Produkt darstellen. Die eine „richtige“ Darstellung des Systems kann und wird es nicht geben. Gerade deshalb ist es sehr wichtig, dass das Entwicklungsteam kontinuierlich zusammenarbeitet. Ansonsten kann kein wirkliches gemeinsames Systemverständnis entwickelt werden und es besteht wiederum viel Raum für Interpretation, was dem hier geschilderten Vorgehen und den verfolgten Zielen hinderlich wäre. Auch wenn die Systementwicklung ein sehr individuell geprägter Prozess ist, hat sich in den Fallstudien folgende Hilfestellung bewährt, um die Komplexität eines Systems handhabbarer zu gestalten: Eine flache, breite Systemhierarchie ist stets einer schmalen vorzuziehen, da dadurch die Zahl der Vererbungen von Relationen reduziert werden kann. Gleichzeitig wird die Übersichtlichkeit erhöht.

Aufgrund der Individualität eines Systems kann auch keine Empfehlung über den für die Anforderungsklä rung notwendigen Detaillierungsgrad des Systems gemacht werden. Bricht man die Systementwicklung auf einem zu abstrakten Niveau ab, besteht die Gefahr Anforderungen zu vergessen. Führt man die Systemdetaillierung jedoch zu weit (zum Beispiel bei einem Pkw bis auf einzelne Schrauben oder sogar elektronische Bauelemente wie Widerstände), besteht die Gefahr, sich in der Komplexität des Systems zu verlieren. In der Praxis ist es so, dass nicht jede Untergliederung eines Elements notwendig ist, wenn es sich zum Beispiel um Zukaufteile handelt oder absehbar ist, dass eine Untergliederung keine neuen Informationen bringen wird. Deswegen ist es notwendig, stets maßvoll zu prüfen, welche Untergliederungen zielführend sind und welche nicht. Dafür ist eine entsprechende Entwicklungserfahrung notwendig.

Darüber hinaus wurden folgende Erfahrungen gemacht:

- Es wurden verschiedene Mechanismen entdeckt, die man nutzen kann, um die Systemkomplexität zu reduzieren, ohne die Qualität des Ergebnisses zu reduzieren. Auch wenn verschiedene Elemente in einem System für die Anforderungsklä rung notwendig sind, sind dennoch nicht alle Relationen zwischen diesen Elementen notwendig. Zum Beispiel sind für die Entwicklung der Templatevorrichtung die Elemente „OP“ und „Patient“, notwendig, die Relationen zwischen diesen beiden Elementen sind aber für die Anforderungen an das zu entwickelnde Produkt nicht von Belang.
- Es hat sich gezeigt, dass eine Unterscheidung zwischen aktiven und passiven Relationen nicht zwingend notwendig, teilweise sogar überflüssig ist. Oftmals ist es so, dass das betrachtete Produkt mit Elementen in Wechselwirkung steht, die bereits existent sind und nicht verändert werden können (zum Beispiel der Sterilisator). In diesem Fall wird das Produkt den Sterilisator nicht beeinflussen, wird aber in einigen Anforderungen stark vom Sterilisator beeinflusst.
- Bei der Arbeit mit der Methode hat sich gezeigt, dass aus unterschiedlichen Ausprägungen zu einer Anforderung nicht immer eine endgültige Ausprägung definiert werden kann, sondern sich daraus oft mehrere untergeordnete Anforderungen ableiten lassen. Derzeit geht in RIA aber die Verknüpfung zur ursprünglichen, übergeordneten Anforderung verloren.
- Ähnliches gilt für die Verarbeitung von Zielkonflikten. Derzeit können mittels der unterschiedlichen Ausprägungen einer Anforderung Zielkonflikte innerhalb einer Anforderung aufgedeckt und bearbeitet werden. Es fehlen aber Mechanismen, mit deren Hilfe man Konflikte zwischen unterschiedlichen Anforderungen aufdecken und bearbeiten kann.

Man kann also festhalten, dass die an die Methode gestellten Kernanforderungen erfüllt werden konnten und das grundsätzliche Vorgehen geeignet ist, die Anforderungsklä rung unter den hier gestellten Rahmenbedingungen deutlich zu verbessern: Es wurde ein Vorgehen entwickelt, das die oftmals schwierige Kommunikation unterschiedlicher Disziplinen verbessert und eine intensive Experteneinbindung über den gesamten Entwicklungsprozess

fordert und fördert. Sie systematisiert das Abfragen des zu entwickelnden Produktes nach Anforderungen, auch impliziter Natur, und hilft Anforderungen zu erkennen und zu konkretisieren. Es sei angemerkt, dass die Grundidee, nämlich Anforderungen aus Relationen abzuleiten, auch „auf dem Papier“ funktionieren würde, ein operativer Einsatz aber ohne die Unterstützung durch eine Software aufgrund der Komplexität nicht möglich wäre.

7 Zusammenfassung und Ausblick

Zunächst werden das im Rahmen dieser Arbeit identifizierte Problem, das Vorgehen und das entwickelte Ergebnis nochmals zusammengefasst. Im Anschluss werden ausgehend von den in den Verifikationsbeispielen gemachten Erfahrungen einerseits Möglichkeiten der Optimierung der relationalen iterativen Anforderungsklä rung und des Werkzeugs RIA skizziert. Zum anderen werden Vorgehensweisen zu erweiterten Praxistests und zur Implementierung der Methode in das vorhandenen Instrumentarium vorgeschlagen.

7.1 Zusammenfassung

Die Anforderungsklä rung ist von nicht zu unterschätzender Bedeutung für den Erfolg eines Produkts. Aber gerade in hochgradig interdisziplinären Projekten gestaltet sich diese oftmals sehr schwierig. Anhand von Praxisbeispielen und einer umfangreichen Analyse des Standes der Forschung konnte gezeigt werden, dass dies zum einen an den großen fachlichen und damit terminologischen Unterschieden der einzelnen Disziplinen liegt. Zum anderen liegt dies an der teilweise nicht ausreichenden Ausrichtung des vorhandenen Methodeninstrumentariums auf diese speziellen Probleme der Anforderungsklä rung. Dem interessierten Leser wurde ein umfassender Überblick über derzeit verfügbare Methoden, deren Einsatzgebiete und Vor- und Nachteile gegeben. Daraus konnten zudem Kriterien abgeleitet werden, die maßgeblich für die Qualität einer Anforderungsklä rung verantwortlich sind.

Ausgehend von diesen Erkenntnissen wurde sich mit der Frage der Ursache von Anforderungen auseinandergesetzt. Es konnte gezeigt werden, dass Anforderungen vor allem aus der Relation zwischen Elementen eines Systems resultieren und der Inhalt der Anforderung sich aus diesen Relationen ableiten lässt. Um diese grundlegende Erkenntnis wurde ein neuer Ansatz zur Anforderungsklä rung entwickelt, der das Produkt nicht nur in seiner Gesamtheit betrachtet, sondern unter Einbindung bekannter Methoden den Entwickler auffordert, zusammen mit Experten das produktdefinierende System iterativ und systematisch in seine einzelnen Elemente zu zerlegen und zu strukturieren. Innerhalb dieses Systems sind die Relationen zwischen den Systembestandteilen zu analysieren, um daraus Anforderungsfelder abzuleiten und die dazugehörigen Anforderungen zu klären.

Zu Verifikationszwecken wurde ein Softwaretool entwickelt, das die Grundideen dieses Vorgehens wiedergibt. Gleichzeitig dient diese Software der Unterstützung des Entwicklers hinsichtlich weiterer Kriterien, die die Qualität der Anforderungsklä rung beeinflussen.

In ersten Projekten wurde das Vorgehen verifiziert, dessen Vor-, Nachteile und Optimierungspotential herausgearbeitet. Es konnte gezeigt werden, dass mit diesem Vorgehen dem Entwickler eine Möglichkeit an die Hand gegeben wird, in ihm fremden Fachgebieten Anforderungen unter kontinuierlicher Einbindung eines Experten zu klären. Durch die systematische Abfrage der Relationen nach Anforderungen bei jeder weiteren Detaillierung des Systems erreicht das Anforderungssystem iterativ eine immer höhere Detaillierungsstufe.

So wird erreicht, dass die daraus abgeleitete Anforderungsliste immer für den gerade folgenden Konstruktionsabschnitt vollständig ist.

Dörner bezeichnet die „geistige Ausstattung“ des Menschen als kognitive Struktur, die er in epistemische und heuristische Strukturen untergliedert [DÖRNER 1987, S. 27 ff]. Die epistemische Struktur enthält das Wissen über den Realitätsbereich, die heuristische wird für das Lösen von Problemen benötigt, für die es keine Löseverfahren gibt. Gröber [GRÖBER 1992, S. 63 f] untergliedert das anforderungsbezogene epistemische Wissen in Allgemeinwissen⁴⁰, Berufswissen⁴¹ und Erfahrungswissen⁴². Die heuristische Struktur wird benötigt, wenn das in der epistemischen Struktur gespeicherte Wissen für die Lösung eines Problems, hier die Klärung von Anforderungen, nicht ausreicht. Sie enthält Heurismen (Findeverfahren), um diesen Mangel durch Reorganisation und Neukombination von Handlungsprogrammen zu kompensieren.

In interdisziplinären Entwicklungsprojekten fehlt dem Entwickler meistens das epistemische Wissen der beteiligten Disziplin, dem Experten aus der beteiligten Disziplin dagegen das epistemische Wissen des Entwicklers. Der Entwickler ist also in hohem Maße auf die heuristische Struktur seines Wissens und das epistemische Wissen seines Gegenüber angewiesen. Letztendlich sind also beide vom epistemischen Wissen des anderen abhängig, um eine gute Anforderungsklä rung durchführen und ein erfolgreiches Produkt entwickeln zu können. Dies kann nur in enger Zusammenarbeit erfolgen. Das dafür notwendige bindende Glied muss die heuristische Struktur beider sein, durch die das epistemische Wissen der Beteiligten „zusammengefasst“ werden kann. In der Praxis fehlt den beteiligten Disziplinen meist eine gemeinsame Basis, ein Verbindungselement, um auf dieser Grundlage Fragen diskutieren zu können, so dass jeder seinen notwendigen Anteil des epistemischen Wissens zur Identifikation der Anforderung beitragen könnte. Vielleicht könnte das hier vorgestellte methodische Vorgehen dieses fehlende Verbindungselement sein, um in Form der Systemstruktur und eines gemeinsamen Systemverständnisses eine gemeinsame heuristische Struktur zu schaffen und so das epistemische Wissen der Beteiligten zusammenzuführen und nutzbar zu machen.

⁴⁰ Wissen, das man im Rahmen der Allgemeinbildung über einen Anforderungsbereich hat.

⁴¹ Wissen, das im Rahmen der Berufsausbildung über einen Anforderungsbereich vermittelt wird.

⁴² Wissen, das man sich über zahlreiche Anforderungsklä rungsprojekte hinweg zu einem Anforderungsbereich angeeignet hat.

7.2 Ausblick

Um die relationale iterative Anforderungsklä rung und das dazugehörige Softwaretool RIA zu einem operativen Einsatz zu bringen, sind unterschiedliche vorbereitende Tätigkeiten und Maßnahmen notwendig, die nachfolgend erläutert werden. Hierfür wird es zunächst notwendig sein, die Methode und das Tool auf Basis der in den Entwicklungsprojekten gemachten Erfahrungen zu optimieren und weiter zu testen, um die Methode dann in die bestehende Methodenlandschaft zu implementieren.

7.2.1 Methodenoptimierung

Im Folgenden werden die aus den in Kapitel 6.3 geschilderten Erfahrungen abgeleiteten Optimierungsvorschläge den drei Phasen des Vorgehens Systemaufbau, Relationenanalyse und Ableiten von Anforderungen zugeordnet.

Systemaufbau

Die oben bereits angesprochen Komplexität der entstehenden Systeme kann durch eine entsprechende grafische Darstellung vermindert werden. Die derzeit in RIA gewählte Darstellung als Baumstruktur ist schon eine große Hilfe, wird dem Anwender aber nur in dem Modul „Systementwicklung“ angezeigt. Eine interaktive Anzeige dieser Systemdarstellung in anderen Modulen der Software wäre hilfreich, da dort derzeit ein grafischer Überblick über das System fehlt und die Arbeitsaufträge (Relationenanalyse, Anforderungsklä rung etc.) entsprechend schwer zu lösen sind.

Relationenanalyse

Wie bereits angesprochen, ist für die Beherrschung der Systemkomplexität eine grafische Darstellung von großer Hilfe. Dies gilt auch für die Darstellung der bereits ermittelten Relationen. Derzeit werden in RIA nur die Struktur des Systems, nicht aber die bereits identifizierten Relationen grafisch angezeigt. Würde zum Beispiel immer die aktuell betrachtete Relation in der Systemdarstellung interaktiv abgebildet werden, würde dies die Anforderungsklä rung stark vereinfachen.

Wie bereits erwähnt, kann die Systemkomplexität reduziert werden ohne die Qualität des Ergebnisses zu beeinflussen, da Elementpaarungen existieren können, deren sämtlichen möglichen Relationen für die Anforderungsklä rung ohne Belang sein können. Erkennt man diese Paarungen und schließt deren Relationen von vornherein aus, wird die maximal mögliche Komplexität des Systems stark reduziert. Ein entsprechender Mechanismus müsste dafür in RIA hinterlegt werden.

Darüber hinaus könnte man sich vorstellen, dass RIA registriert, welche der in einem System ohne Berücksichtigung ihrer Art theoretisch möglichen Relationen bereits untersucht worden sind. Dies würde auch der Forderung nach einer Überprüfbarkeit der Vollständigkeit von

Anforderungen nachkommen, da so überprüft werden kann, ob zu einem bestimmten Zeitpunkt alle bis dahin möglichen Relationen auf ihre Existenz überprüft worden sind.

Durch eine differenzierte Betrachtung der aktiven und passiven Relationen kann die Gesamtkomplexität ebenfalls reduziert werden. Es sollte individuell geprüft werden, ob wechselseitige Relationen notwendig sind. Ist dies nicht der Fall, können Relationen zugunsten einer geringeren Relationszahl weggelassen werden. Auch hierfür könnte in RIA eine entsprechende Funktion vorgesehen werden, die nur die relevanten Relationen einblenden lässt.

Ableiten von Anforderungen

Zunächst soll festgehalten werden, dass die an die Methode gestellten Erwartungen weitestgehend erfüllt werden konnten. Dennoch kann die Arbeit mit den Anforderungen noch verbessert werden. Dies betrifft vor allem die Punkte Strukturierung, Pflege und Erweiterung und Vernetzung von Anforderungen und in diesem Kontext auch die Bearbeitung von Zielkonflikten.

Zunächst wäre es sehr hilfreich, wenn RIA bei der Anforderungsklä rung verschiedene Filter für Anforderungen anbieten würde. Denkbar wäre eine Filterung nach Relationsmerkmalen, aber auch nach Relationszugehörigkeit und deren Elementzugehörigkeit. Dies würde zum einen die Auswahl der vorhandenen Anforderung beim Ableiten von Anforderungen aus Relationen erleichtern, zum anderen wäre damit eine Ausgabe von Anforderungen nach ihrer Zugehörigkeit zu bestimmten Elementen möglich. So könnte einem Konstrukteur genau der Teil an Anforderungen ausgegeben werden, die für seinen Verantwortungsbereich relevant sind. Gleichzeitig hat er aber über die Verknüpfung der Anforderungen mit Relationen immer einen Überblick über die für ihn wichtigen Schnittstellen im Produkt.

Wird derzeit eine Anforderung in RIA auf mehrere Anforderungen aufgespaltet, geht die Verknüpfung zur ursprünglichen, übergeordneten Anforderung verloren. Für die Nachvollziehbarkeit eines Anforderungssystems wäre es von großer Bedeutung, diese Verknüpfungen von Anforderungen abbilden zu können. Dies würde der Abbildung der strukturellen Beziehungen von Anforderungen im Sinne von Kruse [KRUSE 1996, S. 78] entsprechen.

Ähnliches gilt für die Verarbeitung von Zielkonflikten. Derzeit können nur Zielkonflikte innerhalb einer Anforderung aufgedeckt und bearbeitet werden. Darüber hinaus wäre es aber sinnvoll auch übergreifende Zielkonflikte darstellen und unter Zuhilfenahme bekannter Methoden [EILETZ 1999] bearbeiten zu können. Dies würde der Abbildung der semantischen Beziehungen von Anforderungen im Sinne von Kruse [KRUSE 1996, S. 81] entsprechen.

Darüber hinaus wäre es vorstellbar, die gewonnenen Anforderungen mit der Produktstruktur selbst, also den Elementen zu verknüpfen, die die Anforderungen erfüllen. Dies entspräche dem Ansatz von Humpert [HUMPERT 1995, S. 75] und böte den Vorteil, dass zu jeder Anforderung ein Informationsnetz von ihren sie verursachenden Elementen über die Anforderung bis zu den sie erfüllenden Elementen existiert.

7.2.2 Erweiterte Verifikation und Methodenimplementierung

Nach Durchführung der genannten Optimierungsmaßnahmen wird es notwendig sein, die relationale iterative Anforderungsklä rung nochmals verstärkt in der Praxis zu testen. Hierfür wären zwei unterschiedliche Vorgehensweisen denkbar, die auch kombiniert werden könnten.

Zunächst müssten die Methode und RIA in komplexeren Projekten getestet werden, um nachzuweisen, dass sie auch oder gerade für komplexe, interdisziplinäre Projekte geeignet sind und eine Verbesserung gegenüber dem bisherigen Vorgehen darstellen. Dies könnte im Rahmen von interdisziplinären Entwicklungsprojekten geschehen, die den in Kapitel 1.3 formulierten Randbedingungen weitestgehend entsprechen. Wichtig ist dabei, dass der Aufwand zur Erfassung der Anforderungen aus Sicht des *Kunden* dem Produkt entsprechen muss [KRUSE 1996, S. 85] und dies dem Kunden auch vermittelt werden kann.

Darüber hinaus wäre eine Untersuchung der Methode und RIA´s unter experimentellen Bedingungen von großem Nutzen. Hierfür müsste eine in der Komplexität überschaubare Entwicklungsaufgabe formuliert werden (wie zum Beispiel das Beispiel der Spritze), die dann von mehreren interdisziplinär zusammengesetzten Entwicklungsteams mit und ohne Einsatz von RIA unter kontrollierten Bedingungen (definierte Hilfsmittel, begrenzte Zeit etc.) ausgeführt werden müsste. Dabei wären für die Auswertung zumindest Ton-, besser Videoprotokolle und Protokolle der Eingaben in RIA über der Zeit notwendig. Neben dem reinen Vergleich der Anforderungslisten und der Entwicklungsergebnisse aus den einzelnen Experimenten wären noch tief greifendere Untersuchungen denkbar. Auf diese Weise könnte zum Beispiel ermittelt werden, welche Anforderungen, zu welchem Zeitpunkt, auf welchem Weg und aus welchem Grund ihren Weg in das Anforderungssystem gefunden haben, um so den Erfolg und die Notwendigkeit einer prozessbegleitenden Anforderungsklä rung nachzuweisen. Im Rahmen dieser Arbeit musste auf solche Experimente aufgrund deren Zeit- und Kosten- und Personalintensität und fehlender finanzieller Mittel verzichtet werden.

Sollten auch diese Untersuchungen positiv verlaufen, wäre eine Integration dieses Ansatzes in die bestehende Methodenlandschaft anzustreben.

Hierfür erscheint es sinnvoll, nochmals auf die Ansätze von Humpert [HUMPERT 1995] und Kruse [KRUSE 1996] einzugehen. Humpert baut in seinem Ansatz ein hierarchisches Anforderungssystem auf und verknüpft dieses mit der Baustruktur des Produktes. Kruse befasst sich vor allem mit den Beziehungen von Anforderungen untereinander und verknüpft diese mit den sie erfüllenden Elementen. Letztendlich gleichen sich diese beiden Ansätze sehr stark. Die relationale iterative Anforderungsklä rung befasst sich vor allem mit der Ursache von Anforderungen und nutzt diese Erkenntnis um Anforderungen erkennen, formulieren und definieren zu können. Damit stellt die relationale iterative Anforderungsklä rung das fehlende Glied für eine ganzheitliche Anforderungsklä rung und Pflege dar. Fasst man die drei Ansätze zusammen, würde man eine umfassende Methodik erhalten, die die hier angesprochenen Aspekte der Anforderungsklä rung abdecken könnte und den Vorteil böte, dass zu jeder Anforderung ein Informationsnetz von ihren sie verursachenden Elementen über die Anforderung bis zu den sie erfüllenden Elementen existiert. Hinzu kommen die abgebildeten Vernetzungen mit anderen Anforderungen. Man könnte so also bei der Änderung einer Anforderung sehr detailliert die Auswirkungen oder zumindest die betroffenen „Nachbarn“ überblicken. Damit einhergehen würde eine ganzheitliche Dokumentation sowohl des

Systems, also der Produktstruktur, als auch des Anforderungssystems mit all seinen Vernetzungen. Auch eine Einbeziehung des Produktlebenszyklus im Sinne von Franke [FRANKE 1975] wäre denkbar.

Natürlich bleibt kritisch zu prüfen, ob und bis auf welchem Detaillierungsgrad die aufgrund der zahlreichen Relationen zu erwartende Komplexität handhabbar und überblickbar ist. Dennoch ist die relationale iterative Anforderungsklä rung und der ihr zugrunde liegende Ansatz der Analyse der Anforderungsursachen in den Augen des Verfassers ein wichtiger Schritt, um die bisher beschrittenen Wege zur Klärung von Anforderungen zu erweitern und neue Impulse für die Anforderungsklä rung in der zunehmend interdisziplinären Welt des Entwicklers zu setzen.

8 Literaturverzeichnis

AHRENS, G.:

Das Erfassen und Handhaben von Produkthanforderungen.
Berlin: Dissertation der TU Berlin 2000.

AKAO, Y.:

QFD - Quality Function Deployment.
Landsberg a. L.: moderne industrie 1992.

ALEXANDER, K.:

Design for validation of medical devices and equipment.
PhD thesis, University of Cambridge: Engineering Department 1999.

ALMEFELT, L.; ANDERSSON, F.; NILSSON, P.; MALMQVIST, J.:

Exploring Requirements Management in the Automotive Industry. In: Folkesson, A.; Gralén, K.; Norell, M.; Sellgren, U. (Hrsg.): International Conference on Engineering Design - ICED 2003, Stockholm.
Stockholm: Design Society 2003.

ALTSCHULLER, G.:

Erfinden - Wege zur Lösung technischer Probleme.
Berlin: VEB Verlag Technik 1984.

AMBROSY, S.:

Methoden und Werkzeuge für die integrierte Produktentwicklung.
Aachen: Shaker 1997. Zugl.: München: Diss. der TU München 1996.

ANDELFINGER, U.:

Diskursive Anforderungsanalyse.
Frankfurt a. M.: Peter Lang 1997.

BARRENSCHEEN J., DREBING U., SIEVERDING H.:

Rechnerunterstützte Erstellung von Anforderungen.
VDI-Z 131 (1989) 4, 84-89.

BARTHOL, R. P.; BRIDGE, R. G.:

Echo Multi-Response Method for Surveying Value and Influence Pattern in Groups.
Psychological Reports 22 (1968) 3.

BAUMANN, H. G.:

Systematisches Projektieren und Konstruieren.
Berlin, Heidelberg, New York: Springer 1982.

BECK, R. M.:

Untersuchung von Oberflächenbeschichtungen bei Gefäßstützen zur Reduktion der
Restenoserate.
Tübingen: Diss. der Eberhard-Karls-Universität 2001.

BENDER, K.:

Embedded Systems – qualitätsorientierte Entwicklung.
Berlin, Heidelberg, New York: Springer 2005.

BERNHART, M.:

Entwicklung einer Regelung der Bluttemperatur während der Extrakorporalen Zirkulation.
TU München: unveröffentlichte Semesterarbeit Nr. 1925; Lehrstuhl für Produktentwicklung
2002.

BICHLMAIER, C.:

Methoden zur flexiblen Gestaltung von integrierten Entwicklungsprozessen.
München: Herbert Utz 2000. Zugl.: München: Diss. der TU München 2000.

BIRNBAUM, D.; BETZ, P.:

Extracorporeal circulation as a risk factor in heart surgery.
Z Kardiol 79 (1990) 0, 87-93.

BOKODI, A.:

Aufbau eines Konzepts für die Regelung der Bluttemperatur während der Extrakorporalen
Zirkulation.
TU München: unveröffentlichte Semesterarbeit Nr. 1992; Lehrstuhl für Produktentwicklung
2003.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.:

UML: The Unified Modeling Language user guide.
Reading, Mass.: Addison Wesley 1999.

BREIING, A.; FLEMMING, M.:

Theorie und Methoden des Konstruierens.
Berlin, Heidelberg: Springer 1993.

BROOKS, F. P.:

No Silver Bullit - Essence and Accidents of Software Engineering.
IEEE Computer. 1987, 10-19.

CONRAD, P.; SCHIEMANN, H.; VÖMEL, P. G.:

Erfolg durch methodisches Konstruieren.
Grafenau: Lexika. Verlag 1977.

CROSS, N.:

Engineering Design Methods.
Chicester: Wiley 1996.

DAENZER, W. F.:

Systems Engineering.
Zürich: Verlag Industrielle Organisation 2002.

DANNER, S.:

Ganzheitliches Anforderungsmanagement.
Aachen: Shaker 1996. Zugl.: München: Diss. der TU München 1996.

DEIFEL, B.:

Requirements Engineering komplexer Standardsoftware.
München: Diss. der TU München 2001.

DGQ - DEUTSCHE GESELLSCHAFT FÜR QUALITÄT (HRSG.):

Qualitätsmanagement in der Entwicklung.
Berlin, Wien, Zürich: Beuth Verlag GmbH 1995.

DIEHL, H.:

Konzeptionelle Entwicklungen für ein Minilab auf der Basis der Inkjet-Technologie.
München: unveröffentlichte Semesterarbeit Nr. 1930; TU München, Lehrstuhl für
Produktentwicklung 2001.

DOHMEN, W.:

Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer
Systeme.
München: Dissertation der TU München 2002.

DÖRNER, D.:

Problemlösen als Informationsverarbeitung
Stuttgart: Kohlhammer 1987.

DRAKE, J. M.; XIE, W. W.; TSAI, W. T.; ZUALKERNAN, I. A.:

Approach and case study of requirement analysis where end users take an active role.
15th international conference on Software Engineering, Baltimore, 1993, 177-186.

EHRENSPIEL, K.:

Integrierte Produktentwicklung.
München: Hanser 2003.

EHRENSPIEL, K.; KIEWERT, A.; LINDEMANN, U.:

Kostengünstig Entwickeln und konstruieren.
Berlin, Heidelberg, New York: Springer 2000.

EILETZ, R. :

Zielkonfliktmanagement bei der Entwicklung komplexer Produkte – am Bsp. PKW-
Entwicklung.
Aachen: Shaker 1999. Zugl. München: Diss. der TU München 1999.

ENG, Z.-H.:

Entwicklung eines Instrumentensystems für minimalinvasive Chirurgieroboter.
München: unveröffentlichte Semesterarbeit Nr. 2152; TU München, Lehrstuhl für
Produktentwicklung 2005.

ENGSTLER, F.:

Optimierung eines Aggregats zur pulsatilen Perfusion während herzchirurgischer Eingriffe.
München: unveröffentlichte Semesterarbeit Nr. 2037; TU München, Lehrstuhl für
Produktentwicklung 2004.

FELDHUSEN, J.:

Systemkonzepte für die durchgängige und flexible Rechnerunterstützung des
Konstruktionsprozesses.
Berlin: Dissertation der TU Berlin 1989.

FEYRER, A.:

Entwicklung eines Tools zur Unterstützung der Anforderungsklä rung in interdisziplinären
Projekten..
München: unveröffentlichte Diplomarbeit Nr.963; TU München, Lehrstuhl für
Produktentwicklung 2004.

FRANK, G.; DAHM, M.; HAVERICH, A.; LOWES, D; OELERT, H.:

Herzchirurgie.
Balingen: PERIMED-spitta 1995.

FRANKE, H.-J.:

Methodische Schritte beim Klären konstruktiver Aufgaben.
Konstruktion 27 (1975), 395-402.

GAUSEMEIER, J.; BREXEL, D.; HUMPERT, A.:

Anforderungsbearbeitung in integrierten Ingenieursystemen.
Konstruktion 48 (1996), 119-127.

GERHARD, E.:

Entwickeln und konstruieren mit System.
Renningen-Malmsheim: Expert Verlag 1998.

GERL, T.:

Konzeptionelle Entwicklungen für ein Minilab auf der Basis der Inkjet-Technologie.
München: unveröffentlichte Semesterarbeit Nr. 1926; TU München, Lehrstuhl für
Produktentwicklung 2001.

- GIAPOULIS, A.; SCHLÜTER, A.; EHRENSPIEL, K.; GÜNTHER, J.:
Effizientes Konstruieren durch generierendes und korrigierendes Vorgehen. In: Hubka, V. et al. (Hrsg.): ICED 95, Praha.
Zürich: Edition Heurista 1995, 477-483.
- GRIFFIN, A.; HAUSER, J. R.:
The Voice of the customer.
Marketing Science 12 (1993) 1.
- GRÖGER, B.:
Ein System zur rechnerunterstützten und wissensbasierten Bearbeitung des Konstruktionsprozesses.
Konstruktion 42 (1990), 91-96.
- GRÖBER, H.:
Systematische rechnerunterstützte Ermittlung von Produkthanforderungen.
Darmstadt: Diss. der TU Darmstadt 1992.
- HARASHIMA, F.; TOMIZUKA, M.; FUKUDA, T.:
Mechatronics - "What is it, Why and How?" An Editorial.
IEEE/ASME Transactions on Mechatronics 1 (1996), 1-4.
- HAVELKA, D.:
Requirements Determination: An Information System Specialist Perspective of Prozess Quality.
Requirements Engineering (2002) 6, 220-236.
- HEINZLER, M.; KILGUS, R.; NÄHER, F.; PAETZOLD, H.; RÖHRER, W.; SCHILLING, K.:
Tabellenbuch Metall.
Haan-Grutten: Verlag Europa-Lehrmittel 2002.
- HERBIG, B.:
Vergleichende Untersuchung von Struktur und Inhalt expliziten und impliziten Wissens im Arbeitskontext.
Aachen: Shaker 2001. Zugl. München: Diss. der TU München 2001

HERBST, T.:

Entwicklung eines Rollstuhls aus faserverstärktem Kunststoff.

München: unveröffentlichte Semesterarbeit Nr. 1901; TU München, Lehrstuhl für Produktentwicklung 2001.

HEYNEN, N.:

Entwicklung einer Fernauslösung für Lawinenairbags.

München: unveröffentlichte Semesterarbeit Nr. 2016; TU München, Lehrstuhl für Produktentwicklung 2003.

HUMMEL, J.M.; ROSSUM, W. VAN, VERKERKE, G.J.; RAKHORST, G.:

Medical Technology Assessment: The Use of the Analytic Hierarchy Process as a Tool for Multidisciplinary Evaluation of Medical Devices.

International Journal of Artificial Organs 23 (2000) 11, 782-787.

HUMPERT, A.:

Methodische Anforderungsverarbeitung auf Basis eines objektorientierten Anforderungsmodells.

Paderborn: HNI-Verlagsschriftenreihe 1995.

HÜNECKE, W.:

Entwicklung einer Fernauslösung für Lawinenairbags.

München: unveröffentlichte Semesterarbeit Nr. 2017; TU München, Lehrstuhl für Produktentwicklung 2003.

HUTTERER, P.:

Reflexive Dialoge und Denkbausteine für die methodische Produktentwicklung.

München: Diss. der TU München 2005.

ISO 594-1

Conical fittings with a 6% (Luer) taper for syringes, needles and certain other medical equipment. Part 1: General Requirements. 1986.

JUNG, C.; GRAMANN, J.:

Optimisation of the pressure and flow ratio under extracorporeal circulation.

IDEA TV Ges. für kommunikative Unternehmensbetreuung mbH. Entnommen am 03.12.2004.

URL: http://www.innovations-report.de/html/berichte/medizin_gesundheit/bericht-35135.html.

JUNG, C.; GRAMANN, J.; FELGEN, L.:

Optimierung der Kopplung zwischen EK-Pumpe und Organismus: Die HLM als autonomer Roboter. Unveröffentlichter Forschungsbericht des Projektes des Deutschen Herzzentrums München Nr.: 62-00.

München: TU München, Lehrstuhl für Produktentwicklung 2003.

KANO, N.; SERAKU, N.; TAKAHASHI, F.; TSUJI, S.:

Attractive Quality and Must Be Quality.

Quality 14 (1984) 2, 39-48.

KARCHER, A.; WIRTZ, J.

Requirement Engineering for PDM based collaboration in Virtual Enterprises. Engineering Data Management Newsletter, Volume 8, Number 5 & 6.

Genf/Schweiz: 1999.

KARCHER, A.; GLANDER, M.; WIRTZ, J.

Ein integriertes Anforderungsmodell für die praxisnahe Einführung von Integrationssystemen. Proc. 7. Internationaler CSC Ploenzke Kongreß '98 'Life Cycle Management - Produkte und Prozesse gestalten für die Märkte von morgen'; CSC Ploenzke AG.

Wiesbaden: 1998, 644-654.

KELLY, G. A.:

The Psychologie of Personal Constructs.

New York: W. W. Norton 1955.

KING, B.:

Doppelt so schnell wie die Konkurrenz - Quality Function Deployment.

St. Gallen: gfmt 1994.

KÖCK, T.:

Entwicklung eines Moduls zur Strahlenverschiebung bei der Belichtung in Minilabs.

München: unveröffentlichte Semesterarbeit Nr.1957; TU München, Lehrstuhl für Produktentwicklung 2002.

KOLLER, R.:

Konstruktionslehre für den Maschinenbau.

Berlin, Heidelberg, New York, Tokyo: Springer 1985.

KRUSE, P. J.:

Anforderungen in der Systementwicklung.
Düsseldorf: VDI-Verlag 1996.

KRUSE, P.J.; DIETZ, P.; LESCHONSKI, K.:

Behandeln von Anforderungen in der Entwicklung verfahrenstechnischer Prozesse, Maschinen und Anlagen.
Konstruktion 49 (1997) 0, 41-48.

LAMNEK, S.:

Qualitative Sozialforschung. Band 1: Methodologie..
Weinheim: Psychologie Verlags Union 1993.

LAUER, W.:

Entwicklung eines Instrumententrägers für die Minimalinvasive Roboterchirurgie.
München: unveröffentlichte Semesterarbeit Nr. 2007; TU München, Lehrstuhl für Produktentwicklung 2003.

LIN, J.; FOX, M. S.; BILGIE, T.:

A Requirement Ontology for Engineering Design.
Concurrent Engineering: Research and Applications 4 (1996) 3, 279-302.

LINDEMANN, U. ; JUNG, C. ; FEYRER, A. :

Methode zur Unterstützung der Anforderungsklä rung in fremddisziplinär bestimmten Entwicklungsprojekten.
ZWF-Zeitschrift für wirtschaftlichen Fabrikbetrieb 100 (2005) 1-2, S.53-58.

LINDEMANN, U.:

Methodische Entwicklung technischer Produkte.
Berlin, Heidelberg, New York: Springer 2005.

LINDEMANN, U.; PULM, U.; MAUERER, M.:

Early evaluation of product properties for individualized products. 2nd Interdisciplinary World Congress on Mass Customization and Personalization (MCPC03), München. CD-ROM, 2004.

LINDEMANN, U; JUNG, C.; GRAMANN, J.:

Fremddisziplinär bestimmte Entwicklungsprojekte.
ZWF - Zeitschrift für wirtschaftlichen Fabrikbetrieb 98 (2003) 3, 102-107.

LINDSROM, D. R.:

Five Ways to Destroy a Software Projekt. IEEE 68. 1980, 55-58.

MEITINGER, K.-H.:

Konzeptionelle Entwicklung für ein Minilab auf der Basis der Inkjet-Technologie.
München: unveröffentlichte Semesterarbeit Nr. 1927; TU München, Lehrstuhl für
Produktentwicklung 2001.

MONTAU, R.:

Föderatives Produktdatenmanagement anhand semantischer Informationsmodellierung.
Düsseldorf: VDI-Verlag 1996.

MPG:

Medizinproduktegesetz, 2003.

MPR:

Medizinprodukterichtlinie 93/42/EWG, 1993.

NAUR, P.:

Programming as Theory Building.
Microprocessing and Microprogramming 15 (1985), 253-261.

OTTO, K.N.; AHRENS, G.:

Eine Methode zur Definition technischer Produkthanforderungen.
Konstruktion 49 (1997), 19-25.

PAHL, G., BEITZ, W.:

Klären der Aufgabenstellung und Erarbeitung der Anforderungsliste.
Konstruktion 24 (1972), 195-199.

PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K. H.:

Konstruktionslehre.
Berlin, Heidelberg, New York: Springer 2003.

PECQUET, N.:

Instrumentenentwicklung für die roboterassistierte minimalinvasive Herzchirurgie.
München: unveröffentlichte Semesterarbeit Nr. 1922; TU München, Lehrstuhl für
Produktentwicklung 2002.

PERBANDT, T.:

Konzeptionelle Entwicklung für ein Minilab auf der Basis der Inkjet-Technologie.
TU München: unveröffentlichte Semesterarbeit Nr. 1928; Lehrstuhl für Produktentwicklung
2001.

PETERMANN, M.:

Optimierung eines Aggregats zur pulsatischen Perfusion während herzchirurgischer Eingriffe.
München: unveröffentlichte Semesterarbeit Nr. 2038; TU München, Lehrstuhl für
Produktentwicklung 2003.

POHL, K.:

Requirements Engineering: An Overview.
1996: RWTH Aachen 1996.

PUGH, S.:

Total Design. Integrated Methods for Successful Product Engineering.
Addison-Wesley Publishing Company: Wokingham 1991.

RAASCH, J.:

Systementwicklung mit strukturierten Modellen: Leitfaden für Praxis und Studium..
München: Hanser 1993.

REINICKE, T.

Möglichkeiten und Grenzen der Nutzerintegration in der Produktentwicklung – Eine
Systematik zur Anpassung von Methoden zur Nutzerintegration.
Berlin: Diss. der TU Berlin 2004

RESTREPO, J.; CHRISTIANS, H.:

Design Requirements: Conditioners or Conditioned. In: Folkesson, A.; Gralén, K.; Norell, M.;
Sellgren, U. (Hrsg.): International Conference on Engineering Design - ICED 2003,
Stockholm.
Stockholm: Design Society 2003.

ROTH, K.:

Konstruieren mit Konstruktionskatalog. Band 1: Konstruktionslehre.
Berlin, Heidelberg, New York: Springer 2000.

RUNGE, T.M.; COHEN, D. J.; HANTLER, C. B.; BOHLS, F. O.; OTTMERS S. E.; BRICENO J. C.:
Achievement of physiologic pulsatile flow on cardiopulmonary bypass with a 24 French
cannula.

ASAIO-J 38 (1992) 0, 726-729.

SALUSTRI, F. A.; PARMAR, J.:

Product Design Schematics: Structured Diagramming for Requirements Engineering. In:
Marjanovic, D. (Hrsg.): International Design Conference - Design 2004, Dubrovnik.
Zagreb: Sveucilicisna tiskara 2004.

SCHWANKL, L.:

Analyse und Dokumentation in den frühen Phasen der Produktentwicklung.
München: Dr. Hut 2002. Zugl.: München: Diss. der TU München 2002.

SHEFELBINE, S.:

Requirements Capture for Medical Device Design.
MPhil thesis, University of Cambridge: Engineering Department 1998.

SIDDIQI, J.:

Challenging Universal Truths of Requirements Engineering. IEEE Software. 1994, 18-19.

SIGHIREANU, M.; TURNER, K.:

Requirement Capture, Formal Description and Verification of an Invoicing System.
Montbonnot St Martin: INRIA - Institut National de Recherche en Informatique et en
Automatique 1998.

STEINMEIER, E.:

Realisierung eines systemtechnischen Produktmodells - Einsatz in der PKW-Entwicklung.
Aachen: Shaker 1999. Zugl.: München: Diss. der TU München 1999.

STUTTGARTER ZEITUNG:

Peinliche Panne beim Brückenbau. Entnommen am 14.01.2004. URL <http://www.stuttgarter-zeitung.de/stz/page/detail.php/586632>.

TERNINKO, J.:

Systematic Innovation: an Introduction to TRIZ.
Boca Ration: St. Luca Press 1998.

TILKE, C.:

Konzeptionelle Entwicklungen für ein Minilab auf der Basis der Inkjet-Technologie.
München: unveröffentlichte Semesterarbeit Nr. 1929; TU München, Lehrstuhl für
Produktentwicklung 2001.

TROITZSCH, D.; TENDRICH, G.; KÖRFER, R.:

Praktische Durchführung der extrakorporale Zirkulation.
In: Tschaut, R. J. Extrakorporale Zirkulation in Theorie und Praxis. Lengerich: Pabst 1999,
181-216.

ULRICH, K. T.; EPPINGER, S. D.:

Product Design and Development.
New York: McGraw-Hill 2003.

URBAN, G. L.; HAUSER, J. R.:

Design and Marketing of New Products.
New Jersey: Prentice-Hall 1993.

VDI-RICHTLINIE 2206:

Entwicklungsmethodik mechatronischer Systeme.
Düsseldorf: VDI-Verlag 2004.

VDI-RICHTLINIE 2221:

Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte.
Düsseldorf: VDI-Verlag 1997.

VDI-RICHTLINIE 2222:

Konstruktionsmethodik - Methodisches Entwickeln von Lösungsprinzipien.
Düsseldorf: VDI-Verlag 1997.

VOSS, B.; KRANE, M.; SCHAD, H.; HEIMISCH, W.; JUNG, CHR.; LANGE, R.;
BAUERNSCHMITT, R.:

Effect of "physiological" pulsatile perfusion on renal blood flow during cardiopulmonary
bypass.
34.Jahrestagung der Deutschen Gesellschaft für Thorax-, Herz- und Gefäßchirurgie, Hamburg
2005.

WAGNER, J. J.:

Konzeptionelle Entwicklungen für ein Minilab auf der Basis der Inkjet-Technologie.
München: unveröffentlichte Semesterarbeit Nr. 1931; TU München, Lehrstuhl für
Produktentwicklung 2001.

WAHRIG, G.; WAHRIG-BURFEIND, R.:

Wahrig - Deutsches Wörterbuch.
Gütersloh: Wissen Media Verlag 2002.

WARD, J.; SHEFELBINE, S.; CLARKSON, P. J.:

Requirement Capture for Medical Device Design. In: Folkesson, A.; Gralén, K.; Norell, M.;
Sellgren, U. (Hrsg.): International Conference on Engineering Design - ICED 2003,
Stockholm.
Stockholm: Design Society 2003.

WEITKEMPER, H.-H.; TROITZSCH, D.; KÖRFER, R.:

Praktische Durchführung der extrakorporalen Zirkulation. In: Tschaut, R. J. Extrakorporale
Zirkulation in Theorie und Praxis.
Lengerich: Pabst 1999, 358-374.

9 Anhang

9.1 Leitlinie mit Relationsattributen

Relationsattribut	Beispiele
geometrisch	Größe, Höhe, Breite, Länge, Durchmesser, Raumbedarf, Anzahl, Anordnung, Anschluss, Ausbau und Erweiterung
kinematisch	Bewegungsart, Bewegungsrichtung, Geschwindigkeit, Beschleunigung
kraftbezogen	Kraftgröße, Kraftrichtung, Krafthäufigkeit, Gewicht, Last, Verformung, Steifigkeit, Federeigenschaften, Stabilität, Resonanz
energetisch	Leistung, Wirkungsgrad, Verlust, Reibung, Ventilation, Zustandsgrößen wie Druck, Temperatur, Feuchtigkeit, Erwärmung, Kühlung, Anschlussenergie, Speicherung, Arbeitsaufnahme, Energieumformung
stofflich	Physikalische und chemische Eigenschaften des Eingangs- und Ausgangsprodukts, Hilfsstoffe, vorgeschriebene Werkstoffe (Nahrungsmittelgesetz u.ä.), Materialfluss und -transport
informations-bezogen	Eingangs- und Ausgangssignale, Anzeigart, Betriebs- und Überwachungsgeräte, Signalform
sicherheits-bezogen	Unmittelbare Sicherheitstechnik, Schutzsysteme, Betriebs-, Arbeits- und Umweltsicherheit
ergonomische	Mensch-Maschine-Beziehung: Bedienung, Bedienungsart
fertigungs-bezogen	Einschränkung durch Produktionsstätte, größte herstellbare Abmessung, bevorzugtes Fertigungsverfahren, Fertigungsmittel, mögliche Qualität und Toleranzen
kontroll-bezogen	Mess- und Prüfmöglichkeit, besondere Vorschriften (TÜV, ASME, DIN, ISO, AD Merkblätter)
montage-bezogen	Besondere Montagevorschriften, Zusammenbau, Einbau, Baustellenmontage, Fundamentierung
gebrauchs-bezogen	Geräuscharmheit, Verschleißrate, Anwendung und Absatzgebiet, Einsatzort (z.B. schwefelige Atmosphäre, Tropen, ...)
wartungs-bezogen	Wartungsfreiheit bzw. Anzahl und Zeitbedarf der Wartung, Inspektion, Austausch und Instandsetzung, Anstrich, Säuberung
recycling-bezogen	Wiederverwendung, Wiederverwertung, Entsorgung, Endlagerung, Beseitigung

Tab. 9-1: modifizierte Leitlinie mit Hauptmerkmalsliste [PAHL ET AL. 2003, S. 194]

9.2 Funktionale Anforderungen an RIA

Bei den hier angegebenen Funktionen handelt es sich nicht um eine vollständige Wiedergabe aller notwendigen und im entwickelten Werkzeug hinterlegten Funktionen. Im Zuge der besseren Lesbarkeit sind untenstehend nur die für das Verständnis notwendigen Funktionen aufgeführt.

Systementwicklung und Relationenanalyse:

- Abbilden einer Systemstruktur
- Hinzufügen, editieren und löschen von Elementen innerhalb einer Systemstruktur
- Klassifizieren von Elementen (Bauteil, Kunde, physikalische Größe etc.)
- Nutzerabhängige, flexible Erweiter- beziehungsweise Reduzierbarkeit der Elementarten
- Anzeigen, hinzufügen und löschen von spezifizierenden, individuellen Merkmalen an die Elemente
- Hinzufügen, editieren und löschen von Relationen an Elemente
- Eingabe der Relationsattribute Art, Bedeutung, Richtung
- Nutzerabhängige, flexible Erweiter- beziehungsweise Reduzierbarkeit der Attribute
- Ausgabe aller Relationen eines Systems
- Ausgabe aller Relationen eines Systems sortiert nach Relationsattributen
- Anzeigen der aktiven (Element beeinflusst ein anderes) und passiven (Element wird beeinflusst) Elements
- Anzeigen der aktiven und passiven Relationen des übergeordneten Elements
- Vererbung von Relationen übergeordneter Elemente auf untergeordnete Elemente

Anforderungsklärung:

- Zuordnung der Relationen zu den Klassen „es resultiert eine Anforderung“, „es resultiert keine Anforderung“, „es ist noch unbekannt, ob eine Anforderung folgt“ und „Relation wurde noch nicht bearbeitet“
- Filterung von Relationen nach diesen Klassen
- Anzeige der mit einer Relation verknüpften Anforderungen
- Verknüpfbarkeit einer Relation mit mehreren Anforderungen
- Verknüpfbarkeit einer Anforderung mit mehreren Relationen
- Darstellbarkeit von unterschiedlichen Ausprägungen zu einer Anforderung
- Lösbarkeit der Verknüpfungen von Relation mit Anforderung

Handhabung von Anforderungen

- Hinzufügen, editieren und löschen von Anforderungen
- Detaillierung von Anforderungen mit Attributen gemäß der Anforderungsliste nach Lindemann [LINDEMANN 2005, 220 f]
- Anzeige einzelner Anforderungen mit allen Attributen
- Nutzerabhängige, flexible Erweiter- beziehungsweise Reduzierbarkeit der Attribute
- Strukturierung von Anforderungen nach den Attributen
- Hinzufügen, editieren und löschen von unterschiedlichen Ausprägungen zu einer Anforderung bis zu deren Freigabe (sog. vorläufige Ausprägungen) Anzeige des Anforderungssystems
- Listenmäßige Anzeige des Anforderungssystems (Anforderungsliste)
- Erfassung der Änderungshistorie von Anforderungen (Wer, Wann, Warum)
- Angabe der mit einer Ausprägung einer Anforderung verknüpften Relationen

10 **Dissertationsverzeichnis des Lehrstuhls für Produktentwicklung**

Lehrstuhl für Produktentwicklung
Technische Universität München
Boltzmannstraße 15
85748 Garching

Dissertationen betreut von

- Prof. Dr.-Ing. W. Rodenacker,
- Prof. Dr.-Ing. K. Ehrlenspiel und
- Prof. Dr. Ing. U. Lindemann

- D1 COLLIN, H.:
Entwicklung eines Einwalzenkalenders nach einer systematischen Konstruktionsmethode.
München: TU, Diss. 1969.
- D2 OTT, J.:
Untersuchungen und Vorrichtungen zum Offen-End-Spinnen.
München: TU, Diss. 1971.
- D3 STEINWACHS, H.:
Informationsgewinnung an bandförmigen Produkten für die Konstruktion der Produktmaschine.
München: TU, Diss. 1971.
- D4 SCHMETTOW, D.:
Entwicklung eines Rehabilitationsgerätes für Schwerstkörperbehinderte.
München: TU, Diss. 1972.
- D5 LUBITZSCH, W.:
Die Entwicklung eines Maschinensystems zur Verarbeitung von chemischen Endlosfasern.
München: TU, Diss. 1974.
- D6 SCHEITENBERGER, H.:
Entwurf und Optimierung eines Getriebesystems für einen Rotationsquerschneider mit
allgemeingültigen Methoden.
München: TU, Diss. 1974.
- D7 BAUMGARTH, R.:
Die Vereinfachung von Geräten zur Konstanthaltung physikalischer Größen.
München: TU, Diss. 1976.
- D8 MAUDERER, E.:
Beitrag zum konstruktionsmethodischen Vorgehen durchgeführt am Beispiel eines
Hochleistungsschalter-Antriebs.
München: TU, Diss. 1976.

- D9 SCHÄFER, J.:
Die Anwendung des methodischen Konstruierens auf verfahrenstechnische Aufgabenstellungen.
München: TU, Diss. 1977.
- D10 WEBER, J.:
Extruder mit Feststoffpumpe – Ein Beitrag zum Methodischen Konstruieren.
München: TU, Diss. 1978.
- D11 HEISIG, R.:
Längencodierer mit Hilfsbewegung.
München: TU, Diss. 1979.
- D12 KIEWERT, A.:
Systematische Erarbeitung von Hilfsmitteln zum kostenarmen Konstruieren.
München: TU, Diss. 1979.
- D13 LINDEMANN, U.:
Systemtechnische Betrachtung des Konstruktionsprozesses unter besonderer Berücksichtigung der Herstellkostenbeeinflussung beim Festlegen der Gestalt.
Düsseldorf: VDI-Verlag 1980. (Fortschritt-Berichte der VDI-Zeitschriften Reihe 1, Nr. 60).
Zugl. München: TU, Diss. 1980.
- D14 NJOYA, G.:
Untersuchungen zur Kinematik im Wälzlager bei synchron umlaufenden Innen- und Außenringen.
Hannover: Universität, Diss. 1980.
- D15 HENKEL, G.:
Theoretische und experimentelle Untersuchungen ebener konzentrisch gewellter Kreisringmembranen.
Hannover: Universität, Diss. 1980.
- D16 BALKEN, J.:
Systematische Entwicklung von Gleichlaufgelenken.
München: TU, Diss. 1981.
- D17 PETRA, H.:
Systematik, Erweiterung und Einschränkung von Lastausgleichslösungen für Standgetriebe mit zwei Leistungswegen – Ein Beitrag zum methodischen Konstruieren.
München: TU, Diss. 1981.
- D18 BAUMANN, G.:
Ein Kosteninformationssystem für die Gestaltungsphase im Betriebsmittelbau.
München: TU, Diss. 1982.
- D19 FISCHER, D.:
Kostenanalyse von Stirnzahnradern. Erarbeitung und Vergleich von Hilfsmitteln zur Kostenfrüherkennung.
München: TU, Diss. 1983.
- D20 AUGUSTIN, W.:
Sicherheitstechnik und Konstruktionsmethodiken – Sicherheitsgerechtes Konstruieren.
Dortmund: Bundesanstalt für Arbeitsschutz 1985.
Zugl. München: TU, Diss. 1984.
- D21 RUTZ, A.:
Konstruieren als gedanklicher Prozess.
München: TU, Diss. 1985.

- D22 SAUERMAN, H. J.:
Eine Produktkostenplanung für Unternehmen des Maschinenbaues.
München: TU, Diss. 1986.
- D23 HAFNER, J.:
Entscheidungshilfen für das kostengünstige Konstruieren von Schweiß- und Gussgehäusen.
München: TU, Diss. 1987.
- D24 JOHN, T.:
Systematische Entwicklung von homokinetischen Wellenkupplungen.
München: TU, Diss. 1987.
- D25 FIGEL, K.:
Optimieren beim Konstruieren.
München: Hanser 1988.
Zugl. München: TU, Diss. 1988 u. d. T.: Figel, K.: Integration automatisierter
Optimierungsverfahren in den rechnerunterstützten Konstruktionsprozess.
- Reihe Konstruktionstechnik München**
- D26 TROPSCHUH, P. F.:
Rechnerunterstützung für das Projektieren mit Hilfe eines wissensbasierten Systems.
München: Hanser 1989. (Konstruktionstechnik München, Band 1).
Zugl. München: TU, Diss. 1988 u. d. T.: Tropschuh, P. F.: Rechnerunterstützung für das
Projektieren am Beispiel Schiffsgetriebe.
- D27 PICKEL, H.:
Kostenmodelle als Hilfsmittel zum Kostengünstigen Konstruieren.
München: Hanser 1989. (Konstruktionstechnik München, Band 2).
Zugl. München: TU, Diss. 1988.
- D28 KITTSTEINER, H.-J.:
Die Auswahl und Gestaltung von kostengünstigen Welle-Nabe-Verbindungen.
München: Hanser 1990. (Konstruktionstechnik München, Band 3).
Zugl. München: TU, Diss. 1989.
- D29 HILLEBRAND, A.:
Ein Kosteninformationssystem für die Neukonstruktion mit der Möglichkeit zum Anschluss an
ein CAD-System.
München: Hanser 1991. (Konstruktionstechnik München, Band 4).
Zugl. München: TU, Diss. 1990.
- D30 DYLLA, N.:
Denk- und Handlungsabläufe beim Konstruieren.
München: Hanser 1991. (Konstruktionstechnik München, Band 5).
Zugl. München: TU, Diss. 1990.
- D31 MÜLLER, R.
Datenbankgestützte Teileverwaltung und Wiederholteilsuche.
München: Hanser 1991. (Konstruktionstechnik München, Band 6).
Zugl. München: TU, Diss. 1990.
- D32 NEESE, J.:
Methodik einer wissensbasierten Schadenanalyse am Beispiel Wälzlagerungen.
München: Hanser 1991. (Konstruktionstechnik München, Band 7).
Zugl. München: TU, Diss. 1991.

- D33 SCHAAL, S.:
Integrierte Wissensverarbeitung mit CAD – Am Beispiel der konstruktionsbegleitenden Kalkulation.
München: Hanser 1992. (Konstruktionstechnik München, Band 8).
Zugl. München: TU, Diss. 1991.
- D34 BRAUNSPERGER, M.:
Qualitätssicherung im Entwicklungsablauf – Konzept einer präventiven Qualitätssicherung für die Automobilindustrie.
München: Hanser 1993. (Konstruktionstechnik München, Band 9).
Zugl. München: TU, Diss. 1992.
- D35 FEICHTER, E.:
Systematischer Entwicklungsprozess am Beispiel von elastischen Radialversatzkupplungen.
München: Hanser 1994. (Konstruktionstechnik München, Band 10).
Zugl. München: TU, Diss. 1992.
- D36 WEINBRENNER, V.:
Produktlogik als Hilfsmittel zum Automatisieren von Varianten- und Anpassungskonstruktionen.
München: Hanser 1994. (Konstruktionstechnik München, Band 11).
Zugl. München: TU, Diss. 1993.
- D37 WACH, J. J.:
Problemspezifische Hilfsmittel für die Integrierte Produktentwicklung.
München: Hanser 1994. (Konstruktionstechnik München, Band 12).
Zugl. München: TU, Diss. 1993.
- D38 LENK, E.:
Zur Problematik der technischen Bewertung.
München: Hanser 1994. (Konstruktionstechnik München, Band 13).
Zugl. München: TU, Diss. 1993.
- D39 STUFFER, R.:
Planung und Steuerung der Integrierten Produktentwicklung.
München: Hanser 1994. (Konstruktionstechnik München, Band 14).
Zugl. München: TU, Diss. 1993.
- D40 SCHIEBELER, R.:
Kostengünstig Konstruieren mit einer rechnergestützten Konstruktionsberatung.
München: Hanser 1994. (Konstruktionstechnik München, Band 15).
Zugl. München: TU, Diss. 1993.
- D41 BRUCKNER, J.:
Kostengünstige Wärmebehandlung durch Entscheidungsunterstützung in Konstruktion und Härterei.
München: Hanser 1994. (Konstruktionstechnik München, Band 16).
Zugl. München: TU, Diss. 1993.
- D42 WELLNIAK, R.:
Das Produktmodell im rechnerintegrierten Konstruktionsarbeitsplatz.
München: Hanser 1994. (Konstruktionstechnik München, Band 17).
Zugl. München: TU, Diss. 1994.

- D43 SCHLÜTER, A.:
Gestaltung von Schnappverbindungen für montagegerechte Produkte.
München: Hanser 1994. (Konstruktionstechnik München, Band 18).
Zugl. München: TU, Diss. 1994.
- D44 WOLFRAM, M.:
Feature-basiertes Konstruieren und Kalkulieren.
München: Hanser 1994. (Konstruktionstechnik München, Band 19).
Zugl. München: TU, Diss. 1994.
- D45 STOLZ, P.:
Aufbau technischer Informationssysteme in Konstruktion und Entwicklung am Beispiel eines elektronischen Zeichnungsarchives.
München: Hanser 1994. (Konstruktionstechnik München, Band 20).
Zugl. München: TU, Diss. 1994.
- D46 STOLL, G.:
Montagegerechte Produkte mit feature-basiertem CAD.
München: Hanser 1994. (Konstruktionstechnik München, Band 21).
Zugl. München: TU, Diss. 1994.
- D47 STEINER, J. M.:
Rechnergestütztes Kostensenken im praktischen Einsatz.
Aachen: Shaker 1996. (Konstruktionstechnik München, Band 22).
Zugl. München: TU, Diss. 1995.
- D48 HUBER, T.:
Senken von Montagezeiten und -kosten im Getriebebau.
München: Hanser 1995. (Konstruktionstechnik München, Band 23).
Zugl. München: TU, Diss. 1995.
- D49 DANNER, S.:
Ganzheitliches Anforderungsmanagement für marktorientierte Entwicklungsprozesse.
Aachen: Shaker 1996. (Konstruktionstechnik München, Band 24).
Zugl. München: TU, Diss. 1996.
- D50 MERAT, P.:
Rechnergestützte Auftragsabwicklung an einem Praxisbeispiel.
Aachen: Shaker 1996. (Konstruktionstechnik München, Band 25).
Zugl. München: TU, Diss. 1996 u. d. T.: Merat, P.: Rechnergestütztes Produktleitsystem
- D51 AMBROSY, S.:
Methoden und Werkzeuge für die integrierte Produktentwicklung.
Aachen: Shaker 1997. (Konstruktionstechnik München, Band 26).
Zugl. München: TU, Diss. 1996.
- D52 GIAPOULIS, A.:
Modelle für effektive Konstruktionsprozesse.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 27).
Zugl. München: TU, Diss. 1996.
- D53 STEINMEIER, E.:
Realisierung eines systemtechnischen Produktmodells – Einsatz in der Pkw-Entwicklung
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 28).
Zugl. München: TU, Diss. 1998.

- D54 KLEEDÖRFER, R.:
Prozess- und Änderungsmanagement der Integrierten Produktentwicklung.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 29).
Zugl. München: TU, Diss. 1998.
- D55 GÜNTHER, J.:
Individuelle Einflüsse auf den Konstruktionsprozess.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 30).
Zugl. München: TU, Diss. 1998.
- D56 BIRSACK, H.:
Methode für Kraftleinleitungsstellenkonstruktion in Blechstrukturen.
München: TU, Diss. 1998.
- D57 IRLINGER, R.:
Methoden und Werkzeuge zur nachvollziehbaren Dokumentation in der Produktentwicklung.
Aachen: Shaker 1998. (Konstruktionstechnik München, Band 31).
Zugl. München: TU, Diss. 1999.
- D58 EILETZ, R.:
Zielkonfliktmanagement bei der Entwicklung komplexer Produkte – am Bsp. PKW-
Entwicklung.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 32).
Zugl. München: TU, Diss. 1999.
- D59 STÖSSER, R.:
Zielkostenmanagement in integrierten Produkterstellungsprozessen.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 33).
Zugl. München: TU, Diss. 1999.
- D60 PHLEPS, U.:
Recyclinggerechte Produktdefinition – Methodische Unterstützung für Upgrading und
Verwertung.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 34).
Zugl. München: TU, Diss. 1999.
- D61 BERNARD, R.:
Early Evaluation of Product Properties within the Integrated Product Development.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 35).
Zugl. München: TU, Diss. 1999.
- D62 ZANKER, W.:
Situative Anpassung und Neukombination von Entwicklungsmethoden.
Aachen: Shaker 1999. (Konstruktionstechnik München, Band 36).
Zugl. München: TU, Diss. 1999.

Reihe Produktentwicklung München

- D63 ALLMANSBERGER, G.:
Erweiterung der Konstruktionsmethodik zur Unterstützung von Änderungsprozessen in der
Produktentwicklung.
München: Dr. Hut 2001. (Produktentwicklung München, Band 37).
Zugl. München: TU, Diss. 2000.

- D64 ASSMANN, G.:
Gestaltung von Änderungsprozessen in der Produktentwicklung.
München: Utz 2000. (Produktentwicklung München, Band 38).
Zugl. München: TU, Diss. 2000.
- D65 BICHLMAIER, C.:
Methoden zur flexiblen Gestaltung von integrierten Entwicklungsprozessen.
München: Utz 2000. (Produktentwicklung München, Band 39).
Zugl. München: TU, Diss. 2000.
- D66 DEMERS, M. T.
Methoden zur dynamischen Planung und Steuerung von Produktentwicklungsprozessen.
München: Dr. Hut 2000. (Produktentwicklung München, Band 40).
Zugl. München: TU, Diss. 2000.
- D67 STETTER, R.:
Method Implementation in Integrated Product Development.
München: Dr. Hut 2000. (Produktentwicklung München, Band 41).
Zugl. München: TU, Diss. 2000.
- D68 VIERTLBÖCK, M.:
Modell der Methoden- und Hilfsmiteleinführung im Bereich der Produktentwicklung.
München: Dr. Hut 2000. (Produktentwicklung München, Band 42).
Zugl. München: TU, Diss. 2000.
- D69 COLLIN, H.:
Management von Produkt-Informationen in kleinen und mittelständischen Unternehmen.
München: Dr. Hut 2001. (Produktentwicklung München, Band 43).
Zugl. München: TU, Diss. 2001.
- D70 REISCHL, C.:
Simulation von Produktkosten in der Entwicklungsphase.
München: Dr. Hut 2001. (Produktentwicklung München, Band 44).
Zugl. München: TU, Diss. 2001.
- D71 GAUL, H.-D.:
Verteilte Produktentwicklung - Perspektiven und Modell zur Optimierung.
München: Dr. Hut 2001. (Produktentwicklung München, Band 45).
Zugl. München: TU, Diss. 2001.
- D72 GIERHARDT, H.:
Global verteilte Produktentwicklungsprojekte – Ein Vorgehensmodell auf der operativen Ebene.
München: Dr. Hut 2002. (Produktentwicklung München, Band 46).
Zugl. München: TU, Diss. 2001.
- D73 SCHOEN, S.:
Gestaltung und Unterstützung von Community of Practice.
München: Utz 2000. (Produktentwicklung München, Band 47).
Zugl. München: TU, Diss. 2000.
- D74 BENDER, B.:
Zielorientiertes Kooperationsmanagement.
München: Dr. Hut 2001. (Produktentwicklung München, Band 48).
Zugl. München: TU, Diss. 2001.

- D75 SCHWANKL, L.:
Analyse und Dokumentation in den frühen Phasen der Produktentwicklung.
München: Dr. Hut 2002. (Produktentwicklung München, Band 49).
Zugl. München: TU, Diss. 2002.
- D76 WULF, J.:
Elementarmethoden zur Lösungssuche.
München: Dr. Hut 2002. (Produktentwicklung München, Band 50).
Zugl. München: TU, Diss. 2002.
- D77 MÖRTL, M.:
Entwicklungsmanagement für langlebige, upgradinggerechte Produkte.
München: Dr. Hut 2002. (Produktentwicklung München, Band 51).
Zugl. München: TU, Diss. 2002.
- D78 GERST, M.:
Strategische Produktentscheidungen in der integrierten Produktentwicklung.
München: Dr. Hut 2002. (Produktentwicklung München, Band 52).
Zugl. München: TU, Diss. 2002.
- D79 AMFT, M.:
Phasenübergreifende bidirektionale Integration von Gestaltung und Berechnung.
München: Dr. Hut 2003. (Produktentwicklung München, Band 53).
Zugl. München: TU, Diss. 2002.
- D80 FÖRSTER, M.:
Variantenmanagement nach Fusionen in Unternehmen des Anlagen- und Maschinenbaus.
München: TU, Diss. 2003.
- D81 GRAMANN, J.:
Problemmodelle und Bionik als Methode.
München: Dr. Hut 2004. (Produktentwicklung München, Band 55).
Zugl. München: TU, Diss. 2004.
- D82 PULM, U.:
Eine systemtheoretische Betrachtung der Produktentwicklung.
München: Dr. Hut 2004. (Produktentwicklung München, Band 56).
Zugl. München: TU, Diss. 2004.
- D83 HUTTERER, P.:
Reflexive Dialoge und Denkbausteine für die methodische Produktentwicklung.
München: Dr. Hut 2005. (Produktentwicklung München, Band 57).
Zugl. München: TU, Diss. 2005.
- D84 FUCHS, D.:
Konstruktionsprinzipien für die Problemanalyse in der Produktentwicklung.
München: TU, Diss. 2005.
- D85 PACHE, M.:
Sketching for Conceptual Design.
München: Dr. Hut 2005. (Produktentwicklung München, Band 59).
Zugl. München: TU, Diss. 2005.

D86 BRAUN, T.:

Methodische Unterstützung der strategischen Produktplanung in einem mittelständisch geprägten Umfeld.

München: Dr. Hut 2005. (Produktentwicklung München, Band 60). Zugl. München: TU, Diss. 2005.

11 Glossar

Anforderung	Eine Anforderung ist eine Forderung bestimmter Eigenschaften oder Funktionen, die ein Produkt erfüllen soll. Die Anforderung kann von einer beliebigen Person bewusst oder unbewusst gestellt sein, die in irgendeiner Weise Interesse an der Erfüllung der Anforderung hat. Eine Anforderung besteht wie eine Eigenschaft immer aus einem beschreibenden Merkmal und einer definierenden Ausprägung (Quantifizierung).
Anforderung definieren	Festlegen der quantitativen oder qualitativen Ausprägung einer Anforderung (zum Beispiel 20 mm).
Anforderung formulieren	Formulieren des beschreibenden Merkmals einer Anforderung, das lösungsneutral und frei von Quantifizierungen o. ä. sein muss.
Anforderungsfeld	Inhaltlicher Bereich, der die Klassifizierung von Anforderungen zulässt (zum Beispiel Geometrie oder Energie). Geeignet zum Aufdecken und Strukturieren von Anforderungen.
Anforderungsklä rung	Die Anforderungsklä rung umfasst alle Maßnahmen und Tätigkeiten, die notwendig sind, um eine gestellte Aufgabe vollständig zu erfassen und die notwendigen Anforderungen zu erkennen, abzuleiten, zu formulieren und zu definieren.
Anforderungssystem	Summe aller Anforderungen in Verbindung mit ihren inneren Zusammenhängen.
Ausprägung (einer Anforderung)	Quantitativer oder Qualitativer Anteil einer Anforderung (bei Bedarf inkl. physikalischer Einheit)
Bereichsforderung	Anforderung, deren Ausprägung durch eine Unter- und Oberschranke definiert ist.
Element	Bestandteil eines Systems, das nicht zwingend gegenständlicher Natur sein muss.
Erkennen einer Anforderung	Feststellen des Bedarfs einer Anforderung in einem bestimmten Kontext.
Experte	Darunter werden hier alle die verstanden, die wichtige Informationen über das zu entwickelnde Produkt

	liefern können.
explizite Anforderung	Anforderung, die bewusst geäußert werden kann und wird.
Festforderung	Anforderung, deren Ausprägung einen exakten Wert annimmt.
fremddisziplinär bestimmt	Eine Disziplin nicht-technischer Art hat erheblichen Einfluss auf eine Entwicklung.
implizite Anforderung	Anforderung, die zwar notwendig ist, aber nicht bewusst geäußert wird oder werden kann.
Iteration	Wiederholte Durchführung einer Handlung bezogen auf das gleiche Problem bei veränderter Eingangsbedingung.
Kanal einer Anforderung	Weg, auf dem eine Anforderung von der Quelle zum Entwickler kommt.
Kunde	Organisation oder Person, die Produkte oder Dienstleistungen bezieht (kauft), nutzt, produziert, installiert, wartet, vermarktet etc. Es kann also auch der Kunde eines Kunden für die Produktentwicklung interessant sein (z.B. für einen Druckmaschinenhersteller nicht nur die Druckerei sondern auch der Leser der Zeitung o. ä. [LINDEMANN 2005, S. 87]).
Merkmal einer Anforderung	Beschreibender Anteil einer Anforderung im Gegensatz zur quantitativen beziehungsweise qualitativen Ausprägung
Methode	Planmäßiges, regelbasiertes Vorgehen zum Erreichen eines bestimmten Ziels [LINDEMANN 2005, S. 288]
MVM	Münchener Vorgehensmodell
QFD	Quality Function Deployment
Quelle einer Anforderung	Herkunft einer Anforderung (Kunde, Fachmann, Literatur Prospekt etc.)
Relation	Beziehung
System	Elemente und zwischen ihnen vorhandene Relationen, durch eine Systemgrenze vom Umfeld abgegrenzt und durch Input-/Outputgrößen mit diesem Umfeld verbunden [LINDEMANN 2005, S. 291].
Systemgrenze	Abgrenzung der im zu betrachtenden System enthaltenen Elemente von den außerhalb des Systems liegenden Elementen und Systemen.

	[LINDEMANN 2005, S. 291]
Systemumfeld	Außerhalb des betrachteten Systems liegende Elemente und Systeme. Tritt mit dem betrachteten System durch summierte In- und Outputs über die Systemgrenze hinweg in Wechselwirkung.
technische Anforderungen	Anforderungen, die ein Produkt hinsichtlich Gestalt, Aufbau, Zusammensetzung und Funktion etc. beschreiben.
Ursprungssystem	Sehr einfaches abstraktes System, das als Ausgangsbasis für die weitere Systementwicklung dient. Kann im einfachsten Fall eine Blackbox mit einem In- und einem Output sein.