

TECHNISCHE UNIVERSITÄT MÜNCHEN  
Lehrstuhl für Produktentwicklung

## **Mustererkennung in komplexen Produktportfolios**

**Maximilian Philipp Kissel**

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. Markus Lienkamp  
Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. Udo Lindemann  
2. Univ.-Prof. Dr.-Ing. Dieter Krause,  
Technische Universität Hamburg-Harburg

Die Dissertation wurde am 22.01.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 08.09.2014 angenommen.



# VORWORT DES HERAUSGEBERS

## **Problemstellung**

Mit zunehmender Individualisierung von Produkten steigt die Komplexität des Produktportfolios eines Unternehmens stark an. Diese Komplexität zu beherrschen, ist eine Schlüsselkompetenz, die Wettbewerbsfähigkeit eines Unternehmens nachhaltig zu sichern. So entwickelt der Systemarchitekt Maßnahmen mit dem Ziel, eine unnötige Vielfalt im Produktportfolio zu verhindern und damit verbundene Kosten zu reduzieren. Zahlreiche Abhängigkeiten und Wechselwirkungen zwischen einzelnen Lösungsbausteinen und zwischen Produkten müssen dabei berücksichtigt werden. Der Aufwand ist groß, die dafür benötigten Informationen zusammenzutragen und fachlich aufzubereiten. In der Regel fehlen geeignete Werkzeuge, Transparenz über relevante Zusammenhänge zu schaffen und somit gute Entscheidungen über erforderliche Änderungen, Erweiterungen oder Reduzierungen im Produktportfolio zu treffen.

## **Zielsetzung**

Der Systemarchitekt benötigt für seine Arbeit methodische Unterstützung, den Herausforderungen der mühevollen Informationsbeschaffung und –Aufbereitung effektiver und effizienter zu begegnen. Hierzu wird eine Methode entwickelt, die den Systemarchitekten unterstützen soll, sich effektiv und effizient ein Bild des Gesamtsystems zu machen und kritische Handlungsfelder im Produktportfolio identifizieren und bewerten zu können. Forschungsziel der vorliegenden Arbeit ist es aufzuzeigen, wie mehr Transparenz geschaffen sowie effektiveres und effizienteres Arbeiten mit komplexen Produktportfolios ermöglicht werden kann. Ein rechnergestützter Demonstrator soll die Verwendung der Methode praxisnah veranschaulichen.

## **Ergebnisse**

Es wird ein Lösungsansatz vorgestellt, der eine graphenbasierte Repräsentation von komplexen Strukturen im Produktportfoliomanagement nutzt und eine regelbasierte Manipulation einsetzt, um die oben genannten Ziele erreichen zu können.

Große, verteilte Datenmengen werden in einem Graph als Datenbasis zusammengeführt. Der Graph besteht aus attribuierten und typisierten Knoten und Kanten. Dieser Schritt erfordert zunächst die Definition eines Metamodells, das die relevanten Aspekte und Relationen des Produktportfoliomanagements beschreibt. Somit erlaubt der hier gewählte Modellierungsansatz, die komplexen Zusammenhänge des Produktportfoliomanagements mit allen relevanten Aspekten und Zusammenhängen abzubilden. Im nächsten Schritt der Methode werden repetitive Aufgaben im Produktportfoliomanagement formalisiert. Zentrale Technologie ist hierbei die Mustererkennung. Mit Hilfe von Regeln werden Muster definiert. Nach jenen Mustern wird im Graph gesucht. Wenn ein Muster gefunden wurde, wird eine definierte Aktion durchgeführt. Mehrere Regeln können in Regelsequenzen verkettet werden. Jene Regelsequenzen können automatisiert auf den Graph angewendet werden. Diesen Mechanismus macht sich die Methode zunutze, im Rahmen des Produktportfoliomanagements für die Aufgabenfelder (1) Datenkonsolidierung, (2) Suchmuster und Diagnose von

Systemfehlern, (3) Bewertung und Überwachung, (4) Design und Strukturierung und (5) Planung und Prognose Muster zu suchen und zu verarbeiten. Im letzten Schritt werden die wesentlichen, entscheidungsrelevanten Ergebnisse der Analysen in aussagekräftige Visualisierungen überführt.

Die entwickelte Methode wurde in zwei Fallstudien in der Industrie evaluiert. Beim Hersteller von Elektrowerkzeugen Hilti war es das Ziel, effizienter mehr Transparenz im Produktportfolio der Antriebsentwicklung zu schaffen, um somit Entscheidungssituationen besser unterstützen zu können. Mit der Methode konnten erfolgreich die erforderlichen Arbeitsschritte formalisiert und somit eine deutliche Effizienzsteigerung bei der Schaffung von mehr Transparenz erreicht werden. In der Fallstudie bei der Elektronik GmbH mussten Abkündigungsentscheidungen innerhalb eines großen, über Jahre gewachsenen Produktportfolios vorbereitet werden. Angesichts der Komplexität konnte diese Aufgabe nicht mehr mit konventionellen Mitteln durchgeführt werden. Mit der hier vorgestellten Methode ließen sich die notwendigen Schritte formalisieren und erfolgreich umsetzen.

### **Folgerungen für die industrielle Praxis**

Die Vernetzung und effiziente Nutzung von Informationen in der Produktentwicklung entwickelt sich für Unternehmen immer mehr zum kritischen Erfolgsfaktor. Bei wachsender Komplexität müssen Eingriffe und Veränderungen im Produktportfolio unter Berücksichtigung zahlreicher Wechselwirkungen im Unternehmen und oft unter großem Zeit- und Kostendruck geplant werden. In den beiden Fallstudien in der Industrie wurde aufgezeigt, wie durch den Einsatz der Methode große Informationsumfänge effektiver und effizienter aufbereitet und verarbeitet werden konnte. Die Methode ist dabei nicht beschränkt auf die dargestellten Anwendungsfälle. Durch die Flexibilität der Metamodellierung und die Mächtigkeit der eingesetzten Regelsprache lässt sich die Methode auf neue Fragestellungen im industriellen Kontext über das Portfoliomanagement hinaus übertragen und anwenden.

### **Folgerungen für Forschung und Wissenschaft**

Der in dieser Arbeit entwickelte Ansatz zur Mustererkennung in komplexen Produktportfolios führt zwei Forschungsgebiete aus dem Systems Engineering und der Informatik zusammen. Es wird aufgezeigt, wie die Anwendung der in der Informatik erforschten Graphtransformation einen messbaren Mehrwert in der Systemarchitektur produzierender Unternehmen erzielen kann. Mit den hier beschriebenen Anwendungsbeispielen wird klar, dass Graphen als Datenbasis, Mustererkennung und Graphtransformationen für viele weitere Anwendungsfelder und Problemstellungen in der Produktentwicklung und Methodenforschung eine vielversprechende Technologie bieten.

Garching, Dezember 2014

Prof. Dr.-Ing. Udo Lindemann  
Lehrstuhl für Produktentwicklung  
Technische Universität München

## DANKSAGUNG

Diese Arbeit entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Produktentwicklung der Technischen Universität München von März 2010 bis Dezember 2013. Für das entgegengebrachte Vertrauen, die Betreuung der Arbeit und die stets konstruktive Kritik möchte ich meinem Doktorvater Herrn Prof. Dr.-Ing. Udo Lindemann danken. Er bot mir in einem positiven und kreativen Umfeld großzügige Freiheiten und Entfaltungsmöglichkeiten, meine Ideen zu diskutieren und wissenschaftlich weiterzuentwickeln.

Herrn Prof. Dr.-Ing. Dieter Krause möchte ich für die Übernahme der Zweitberichterstattung danken. Insbesondere bin ich ihm verbunden für die zahlreichen interessanten und inspirierenden Gespräche mit ihm und seinen Mitarbeitern des Instituts für Produktentwicklung und Konstruktionstechnik der Technischen Universität Hamburg-Harburg. Des Weiteren möchte ich Herrn Prof. Dr.-Ing. Markus Lienkamp für die reibungslose organisatorische Durchführung des Prüfungsvorsitzes danken.

Besonderer Dank gilt allen Kolleginnen und Kollegen des Lehrstuhls für Produktentwicklung. Durch sie habe ich meine Tätigkeiten am Lehrstuhl als große Bereicherung nicht nur in fachlicher, sondern auch in persönlicher Hinsicht erfahren. Die freundschaftliche und stets sehr gute Zusammenarbeit in Projekten, Forschung und Lehre habe ich sehr genossen und werde mich immer gerne daran zurückerinnern. Stellvertretend möchte ich besonders Dr. Bergen Helms, Arne Herberg, Wieland Biedermann und Phillip Schrieverhoff für ihre wertvollen Anregungen zu dieser Arbeit danken.

Ein großer Dank richtet sich an Dr. Josef Ponn und Alexander Hatzold. Ihre Bereitschaft, neue Ansätze offen zu diskutieren und in ihrem Verantwortungsbereich auszuprobieren, hat mir tiefe Einblicke in die Praxis ermöglicht und somit wesentlich zum Gelingen dieser Forschungsarbeit beigetragen. Herrn Dr. Matthias Kreimeyer möchte ich für sein Engagement, die vielen Ideen und Diskussionen in Projekten, Mentorengesprächen, bei der Betreuung von Studienarbeiten oder auch dem Verfassen von gemeinsamen Publikationen herzlich danken.

Ebenso möchte ich den zahlreichen Studentinnen und Studenten danken, die als wissenschaftliche Hilfskräfte oder beim Erstellen von Studienarbeiten wertvolle Ansatzpunkte erarbeitet und mich stets inspiriert haben.

Nicht zuletzt sei meiner Familie von ganzem Herzen gedankt, die mich bis zum heutigen Tag ausnahmslos unterstützt und zu meinem Weg ermuntert hat. Hier möchte ich vor allem meiner Freundin Isabel danken, die mir gerade in den intensiven Arbeitstagen an dieser Dissertation den Rücken gestärkt und mich uneingeschränkt unterstützt hat.

München, Dezember 2014

Maximilian Philipp Kissel



# INHALTSVERZEICHNIS

<b>1. Management komplexer Produktportfolios</b>	<b>1</b>
1.1 Herausforderungen bei der Arbeit mit komplexen Produktportfolios	1
1.1.1 Steigende Komplexität und deren Folgen	1
1.1.2 Motivationsbeispiel aus der Industrie	3
1.2 Zielstellung und Fokus der Arbeit	5
1.3 Thematische Einordnung und Forschungsmethodik	9
1.3.1 Thematische Einordnung der Arbeit	9
1.3.2 Forschungsmethodik	10
1.4 Struktur der Arbeit	11
<b>2. Aufbereitung der Problemstellung</b>	<b>15</b>
2.1 Abgrenzung relevanter Begrifflichkeiten	15
2.1.1 Systemverständnis	15
2.1.2 Komplexität von Systemen	16
2.1.3 Grundzüge des Systems Engineering	17
2.1.4 Architektur technischer Systeme	20
2.1.5 Mustererkennung	23
2.2 Systemarchitektur als Rolle in der Produktentwicklung	25
2.2.1 Aufgaben des Systemarchitekten im Systems Engineering	25
2.2.2 Unternehmensziele und Beitrag des Systemarchitekten	27
2.2.3 Instrumentalziele des Systemarchitekten	28
2.2.4 Voraussetzungen für eine erfolgreiche Systemarchitektur	30
<b>3. Strukturierung komplexer Produktportfolios</b>	<b>33</b>
3.1 Grundbegriffe des Produktportfolios	33
3.1.1 Produktgliederung	33
3.1.2 Strukturen in Produkten	36
3.1.3 Strukturen in Produktportfolios	38
3.2 Methoden zur Komplexitätsreduktion	44

---

3.2.1	Generelle Gestaltungsprinzipien komplexer Produktstrukturen	44
3.2.2	Handlungsfelder zur Reduzierung der Vielfalt	45
3.2.3	Entscheidungsunterstützung und Bewertung	51
3.2.4	Werkzeuge des Komplexitätsmanagements	54
3.3	Herausforderungen im Umgang mit komplexen Produktportfolios	58
3.3.1	Komplexität des Produktportfolios	58
3.3.2	Grundlegende Hürden für die Zielerreichung	68
3.4	Anforderungen an den Lösungsansatz und Beitrag der Arbeit	75
3.4.1	Anforderungen	75
3.4.2	Angestrebter Beitrag der Arbeit	76
<b>4.</b>	<b>Graphenbasierte Repräsentation und Manipulation von komplexen Strukturen</b>	<b>79</b>
4.1	Modellierung komplexer Produktstrukturen	79
4.1.1	Grundlagen der Modellbildung	79
4.1.2	Graphenbasierte Modellierung	80
4.2	Graphtransformation	84
4.2.1	Typen und Instanzen	86
4.2.2	Regeln und Transformationen	87
4.2.3	Softwaretechnische Umsetzung	89
<b>5.</b>	<b>Methodik zur Mustererkennung in komplexen Produktportfolios</b>	<b>93</b>
5.1	Rahmenwerk der Methodik	94
5.1.1	Aufbau	94
5.1.2	Situative Anpassung der Methodik	95
5.2	Datenaufbereitung	97
5.2.1	Zusammenführung aus verteilten Datenquellen	97
5.2.2	Aufbau des Metamodells für das Produktportfoliomanagement	98
5.3	Wissensformalisierung	103
5.3.1	Datenkonsolidierung	104
5.3.2	Suchmuster und Diagnose von Systemfehlern	107
5.3.3	Bewertung und Überwachung	113
5.3.4	Design und Strukturierung	116
5.3.5	Planung und Prognose	120



5.4	Ergebnisaufbereitung	122
5.4.1	Attribute zur Visualisierung	123
5.4.2	Layouter	125
5.4.3	Weitere Visualisierungsformen	126
5.4.4	Fazit zur Ergebnisaufbereitung	127
<b>6.</b>	<b>Evaluation des Lösungsansatzes</b>	<b>129</b>
6.1	Fallstudie 1	129
6.1.1	Ausgangssituation und Problemstellung	129
6.1.2	Zielstellung des Methodeneinsatzes und Vorgehen	135
6.1.3	Ergebnisse	135
6.1.4	Diskussion der Fallstudie 1	145
6.2	Fallstudie 2	146
6.2.1	Ausgangssituation und Problemstellung	146
6.2.2	Zielstellung des Methodeneinsatzes und Vorgehen	147
6.2.3	Ergebnisse	148
6.2.4	Diskussion der Fallstudie 2	160
<b>7.</b>	<b>Diskussion des Beitrags der Arbeit und Implikationen</b>	<b>162</b>
7.1	Schlussfolgerungen aus den Fallstudien	163
7.2	Beitrag der Methodik und Implikationen für die praktische Anwendung	165
7.2.1	Zusammenführung relevanter Daten in graphenbasierter Form	165
7.2.2	Formalisierung repetitiver Aufgaben	167
7.2.3	Automatische Muster- und Fehlererkennung	168
7.2.4	Intuitiv verständliche Visualisierung relevanter Zusammenhänge	169
7.3	Limitationen des Lösungsansatzes	170
<b>8.</b>	<b>Zusammenfassung und Ausblick</b>	<b>173</b>
8.1	Zusammenfassung	173
8.2	Ausblick	176
<b>9.</b>	<b>Literaturverzeichnis</b>	<b>179</b>
<b>10.</b>	<b>Anhang</b>	<b>199</b>

10.1	Metamodell	199
10.2	GrGen.NET	202
<b>11.</b>	<b>Dissertationsverzeichnis des Lehrstuhls für Produktentwicklung</b>	<b>203</b>

## ABKÜRZUNGSVERZEICHNIS

BPL	Blechpaketlänge
DC	Direct Current (Gleichstrom)
DMM	Domain Mapping Matrix
DoD	Department of Defense
DRM	Design Research Methodology
DSL	Domain Specific Language / domänenspezifische Sprache
DSM	Dependency Structure Matrix (auch Design Structure Matrix)
ERP	Enterprise-Resource-Planning
FMEA	Failure Mode and Effects Analysis
GrGen.NET	Graph Rewrite Generator auf .NET-Technologie
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
ISO	International Organization for Standardization
IT	Informationstechnologie
MDM	Multiple Domain Matrix
MySQL	relationales Datenbankverwaltungssystem
NASA	National Aeronautics and Space Administration
Neo4J	Open-Source-Graphdatenbank
NoSQL	Not only SQL
OEM	Original Equipment Manufacturer
OMG	Object Management Group
OWL	Web Ontology Language
PFCI	Product Family Commonality Index
PDM	Produktdatenmanagement Systeme
PLM	Product-Lifecycle Management Systeme
PSS	Product-Service-Systems
QFD	Quality Function Deployment
RFLP	Requirements-Functions-Logical-Physical
SAP	deutsches Software-Unternehmen
SE	Systems Engineering
SQL	Structured Query Language
SysML	Systems Modeling Language
UML	Unified Modeling Language
UNI	Universal
VDI	Verein Deutscher Ingenieure
VMEA	Variant Mode and Effects Analysis



# 1. Management komplexer Produktportfolios

Komplexität entsteht innerhalb eines Produktportfolios durch die Vielfalt und Unterschiedlichkeit von Elementen und Abhängigkeiten. Über Jahre heterogen gewachsene Strukturen, zahlreiche Einflussfaktoren innerhalb und außerhalb eines Unternehmens, mangelnde Transparenz über relevante Zusammenhänge und verteilte, teilweise unvollständige Informationsquellen erschweren das Management komplexer Produktportfolios. Es ist nahezu unmöglich, bei Bedarf den Status Quo abrufen, Entscheidungen auf einer fundierten Informationsbasis treffen oder Änderungen und deren Auswirkungen sinnvoll planen zu können. Der Systemarchitekt benötigt jedoch zur Vorbereitung rationaler Entscheidungen eine effektive wie effiziente Bereitstellung und Aufbereitung relevanter Daten.

Es fehlen Analyse- und Planungswerkzeuge, um die netzwerkartigen Strukturen der Architektur eines Produktportfolios schnell und kostengünstig analysieren und bewerten zu können. Somit lassen sich Entscheidungen in diesem Umfeld nur mit großem Aufwand rational und situationsgerecht treffen. Oft werden daher Bewertungen nicht im nötigen Umfang durchgeführt, da der Kosten- und Zeitaufwand zu hoch erscheinen oder schlichtweg die Ressourcen nicht ausreichen. Die Folgen von Fehlentscheidungen sind oft weitreichend und ein weiterhin unkontrollierter, kostspieliger Wildwuchs unvermeidbar. Dies verstärkt wiederum die Komplexität und eine Spirale setzt sich in Gang.

In dieser Arbeit wird eine Methodik vorgestellt, netzwerkbasierte Analysen werkzeuggestützt zu automatisieren, um somit die Arbeit und die Entscheidungen des Systemarchitekten effektiver und effizienter zu unterstützen. Zunächst wird die Motivation zu dieser Forschungsarbeit aus den Herausforderungen beim Arbeiten mit komplexen Produktportfolios abgeleitet. Es ist das Ziel, diese Herausforderungen mit einer geeigneten Methodik überwinden zu können. Anschließend werden die thematische Eingrenzung, die Forschungsmethodik und der Aufbau der Arbeit vorgestellt.

## 1.1 Herausforderungen bei der Arbeit mit komplexen Produktportfolios

### 1.1.1 Steigende Komplexität und deren Folgen

Hersteller technischer Produkte sehen sich zunehmend mit dem Problem konfrontiert, dass die wachsende Komplexität auf der Nachfrageseite zu einer **ausufernden Komplexität des eigenen Produktportfolios** führt (SCHUH 2005, S. 13). Mit der Erschließung neuer Kundensegmente, Expansion in neue Märkte, einer kontinuierlichen Produktpflege, Ausbau der Lieferantennetzwerke und zahlreichen weiteren Faktoren wachsen Teilstamm, Sachnummern, Varianten und Abhängigkeiten auf Produkt- und Prozessebene innerhalb und über die Unternehmensgrenzen hinaus. Das ist u.a. im Bedürfnis der Kunden begründet, Produkte individualisieren zu können (PINE, II. 1993). Aber auch die unterschiedlichen Märkte erfordern es, die Produkte hinsichtlich landespezifischer Anforderungen und Vorschriften zu lokalisieren.

Dies führt zu einer stark anwachsenden Variantenvielfalt und zu großen Herausforderungen in der Handhabung der internen Produktkomplexität (KRAUSE ET AL. 2013; RENNER 2007). Unternehmen können die unterschiedlichen Käufergruppen in unterschiedlichen Märkten nur dann erfolgreich bedienen, wenn es ihnen gelingt, die **Komplexität des Produktportfolios zu beherrschen** und flexibel und effizient die Diversität in den Produkthanforderungen zu erfüllen.

Das **Management von komplexen Produktportfolios** wird zusätzlich **erschwert** durch zahlreiche unternehmensinterne Interessensgruppen, die mit einbezogen werden müssen (WILDEMANN 2013). Teils konkurrierende Ziele unterschiedlicher Stakeholder entlang des Produktentstehungsprozesses gilt es im wirtschaftlichen Interesse des Unternehmens auszubalancieren. Außerdem laufen oft zahlreiche Entwicklungsprojekte parallel in unterschiedlichen Phasen, deren Arbeit untereinander synchronisiert werden muss. Nur eine entsprechende Transparenz über die parallel laufenden Aktivitäten und deren Wechselwirkungen untereinander verschafft den nötigen Überblick, Doppelarbeiten zu vermeiden und größtmögliche Synergien zu schaffen.

Zudem sind die Tätigkeiten im Produktportfoliomanagement mit **hohem Aufwand** verbunden, da die entscheidungsrelevanten **Informationen** meist erst **aufwendig beschafft und aufbereitet** werden müssen. In einer Studie des Fraunhofer Instituts wird der hohe Aufwand für das Suchen und Aufbereiten von Informationen im Ingenieursumfeld herausgestellt. Es wurden 1401 Ingenieurinnen und Ingenieure befragt, wie sie heute arbeiten, welche Aufgaben in ihrem täglichen Arbeiten eine Rolle spielen, wie viel Zeit sie für ihre Kernaufgaben haben und wie sie ihr Arbeitsumfeld bewerten (MÜLLER ET AL. 2013). Unter den Befragten waren Entwicklungsingenieure (31%), Projektleiter (27%) und Leiter aus der Geschäftsführung und Entwicklung (32%). Auf eine Streuung der Branchen, Fachdisziplinen und Positionierung in Wertschöpfungsketten wurde großen Wert gelegt. 71% der Befragten gaben an, die Datenbeschaffung als „besonders belastend“ zu sehen. Trotz Verwendung von PDM-, PLM- oder ERP-Systemen stehen die notwendigen Daten „nur etwa in der Hälfte der Fälle rechtzeitig (57%), bzw. in der richtigen Form (48%) zur Verfügung“ (MÜLLER ET AL. 2013, S. 23f). Aus Abbildung 1-1 geht hervor, dass gerade bei Projektleitern zwar naturgemäß der Zeitanteil für Entwicklungsaufgaben eher gering ist, allerdings erhebliche Zeitanteile für Kommunikation und Koordination, Informationssuche und –beschaffung sowie für Datenerfassung und Routineaufgaben benötigt werden.

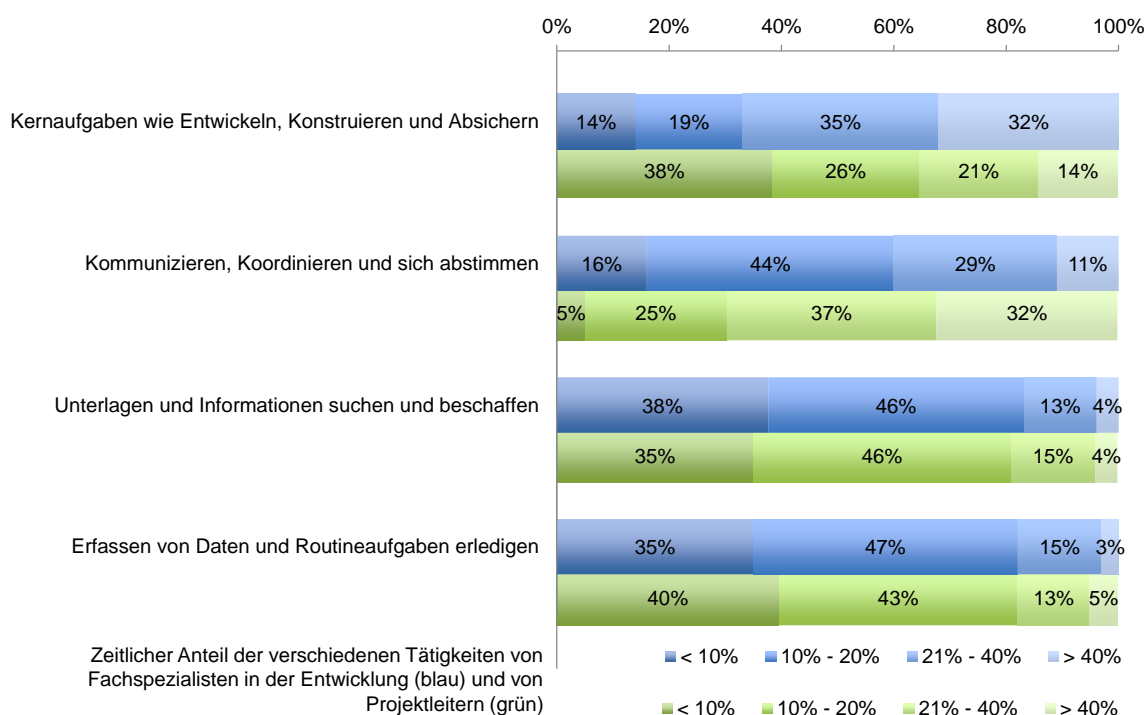


Abbildung 1-1: Ergebnisse aus einer Studie des Fraunhofer Instituts zu Zeitanteilen verschiedener Tätigkeiten im Ingenieursbereich (MÜLLER ET AL. 2013)

Das Zusammenwirken aus steigender Komplexität auf Grund marktseitiger und unternehmensinterner Faktoren und der hohe Aufwand, die wachsende Komplexität zu beherrschen, stellt Unternehmen vor große Herausforderungen. Der Unterstützungsbedarf ist groß. Im nachfolgenden Beispiel aus der Praxis wird dies anschaulich.

### 1.1.2 Motivationsbeispiel aus der Industrie

Das Produktportfolio der Elektronik GmbH<sup>1</sup> ist über Jahre um viele Produkte, Varianten und Sonderlösungen gewachsen. Individuelle Kundenwünsche führten immer wieder dazu, dass Produkte abgeändert wurden, somit zusätzliche Varianten entstanden sind und das Portfolio weiter anwuchs. Durch das breite Angebot werden viele Produkte in nur kleinen Stückzahlen nachgefragt, was sich negativ auf die Herstellungskosten auswirkt. Einige der Produkte machen seit geraumer Zeit keinen nennenswerten Umsatz mehr, verursachen aber weiterhin unnötige Kosten u.a. in Verwaltung, Lagerhaltung, Fertigung, Einkauf und Vertrieb. Daher sollen unwirtschaftliche Produkte abgekündigt werden.

Viele der abzukündigenden Produkte verwenden allerdings die gleichen Baugruppen oder Bauteile wie wirtschaftlich gesunde Produkte. Zudem können Fertigungsressourcen oder auch bestimmte Zulieferer von einer Abkündigung betroffen sein und werden nicht mehr benötigt. Diese Abhängigkeiten und Wechselwirkungen müssten für jedes Produkt individuell analysiert

<sup>1</sup> Das mittelständische Unternehmen möchte nicht namentlich genannt werden. Aus Gründen der Geheimhaltung werden daher Informationen verfremdet, die Rückschlüsse auf das Unternehmen zulassen. Die Fallstudie wird im Kapitel 6.2 ausführlich behandelt.

werden, um sie in einer rationalen Abkündigungsentscheidung berücksichtigen zu können. Wenn unter Zeitdruck Abkündigungskandidaten ermittelt und die Folgen abgeschätzt werden müssen, wird die Komplexität des Produktportfolios für den Verantwortlichen schnell zu einer großen Herausforderung. Das Problem sind der mangelnde Überblick über das Produktportfolio und der große Aufwand, der nötig wäre, die betroffenen Produkte und deren relevante Abhängigkeiten zu ermitteln.

Nachfolgende Abbildung 1-2 zeigt einen Ausschnitt aus dem Produktportfolio der Elektronik GmbH. Um den Graphen darstellen zu können, werden hier nur 250 von den insgesamt 6.784 Produkten dargestellt. Der Graph in der Abbildung umfasst 778 Knoten und 2.724 Kanten. Die Knoten beschreiben Produktfamilien (Grau), Produkte (Blau), Baugruppen und Teile (Rot). Die Kanten stehen für vertikale „ist Teil von“-Abhängigkeiten zwischen den Aggregationsstufen. Zusätzliche Wechselwirkungen mit Lieferanten und Produktionsressourcen konnten im Rahmen dieser Fallstudie nicht berücksichtigt werden.

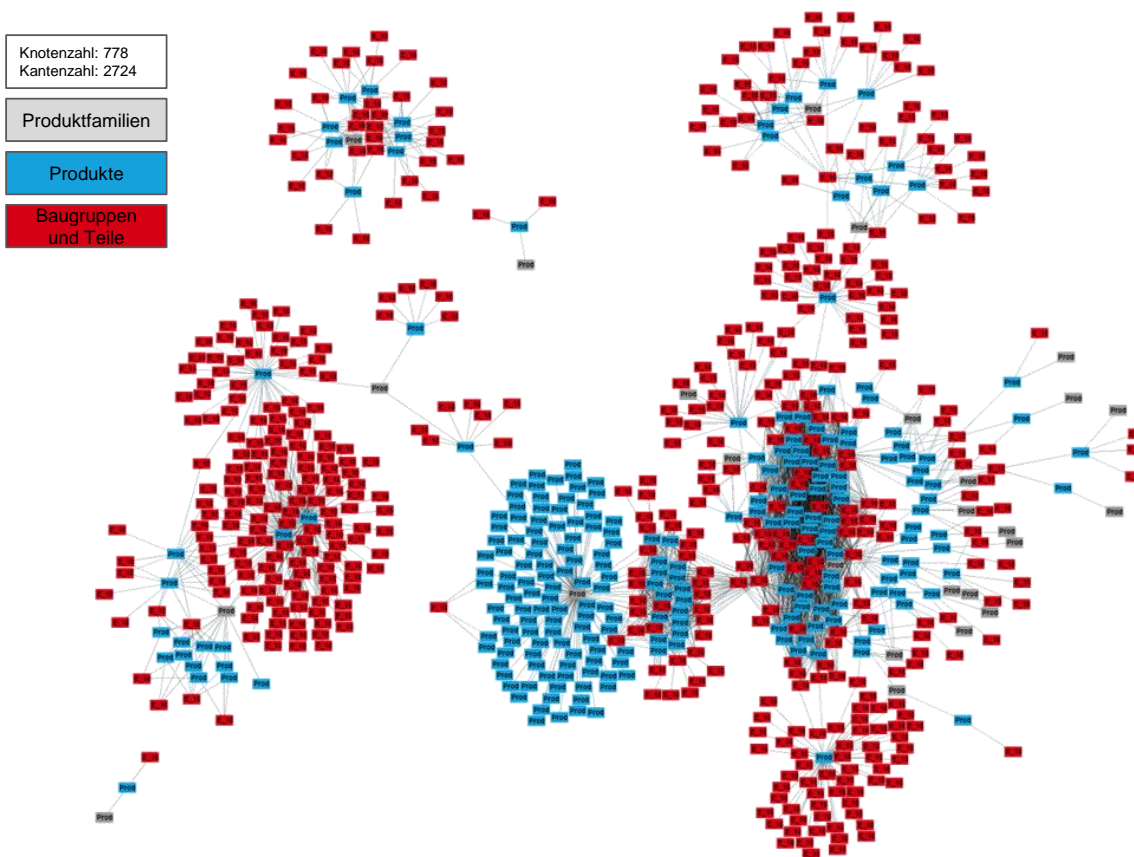


Abbildung 1-2: Komplexität des Produktportfolios der Elektronik GmbH

Die Komplexität der Aufgabe besteht darin, aus verteilten Datenquellen die nötigen Informationen zusammenzusuchen. Nicht alle wichtigen Informationen sind im Unternehmen einfach verfügbar. Zum Beispiel müssen Datenbankabfragen durch Experten getätigt, Zugriffsrechte beantragt, Workshops mit Fachleuten veranstaltet oder große Datensätze in Tabellenkalkulationsprogrammen verarbeitet werden. Die Wechselwirkungen von abzukündigenden Produkten und wirtschaftlich gesunden Produkten durch Wiederholteile



machen das Vorhaben der Portfoliobereinigung zu einer komplexen Aufgabe, die „manuell“ mit Standard-Rechner-Werkzeugen oft nicht mehr zu bewältigen ist.

Im Fallbeispiel der Elektronik GmbH müssen Verwendungsnachweise von Bauteilen erzeugt werden, um Wiederholteile in unterschiedlichen Produkten erkennen zu können. Dafür müssen komplexe Abfragen in verschiedenen, teils selbst entwickelten Datenbanksystemen durchgeführt werden. Produktinformationen wie Stücklisten können über das ERP<sup>2</sup>-System exportiert werden. Weiterhin müssen Umsatzzahlen, Deckungsbeiträge und Stückzahlen aller Produkte über die vergangenen Jahre gesammelt werden. Diese Zahlen müssen teilweise aggregiert auf Produktfamilien berechnet werden. Wenn nun noch Ausnahmen definiert werden, wie beispielsweise Produkte mit hohem Deckungsbeitrag auch dann abzukündigen, wenn sie unwirtschaftliche Produktionsressourcen oder Teile von abzukündigenden Lieferanten nutzen, so gestaltet sich die Suche nach solchen Mustern<sup>3</sup> in den verteilten Datenbanken wie die „Suche nach der Nadel im Heuhaufen“.

Das Fallbeispiel der Elektronik GmbH verdeutlicht, dass ein über die Jahre gewachsenes Produktprogramm die Wirtschaftlichkeit beeinträchtigen kann, wenn es nicht gelingt, umsatzschwache Produkte effektiv zu erkennen und abzukündigen. Das Unternehmen benötigt daher Werkzeuge zum effektiveren und effizienteren Umgang mit großen komplexen Produktstrukturen, die es erlauben, unterschiedliche Datenquellen zusammenzuziehen, aufwendige manuelle Tätigkeiten zu automatisieren und systematischer nach Suchmustern, Fehlern und Handlungsfeldern suchen zu können.

## 1.2 Zielstellung und Fokus der Arbeit

Die Entwicklung variantenreicher Produktfamilien wird seit Jahren intensiv in Wissenschaft und Praxis untersucht (SCHUH 1988; CAESAR 1991; MARTIN & ISHII 2002; SIMPSON 2004; JIAO ET AL. 2006; KRAUSE ET AL. 2013; HARLOU (2006); MEYER & LEHNERD 1997, uvm.). **Zahlreiche Methoden und Werkzeuge** sind entstanden, die Wettbewerbsfähigkeit von Unternehmen durch methodische Gestaltung ihrer variantenreichen Produktportfolios zu steigern. Mit dieser Arbeit soll in diesem Forschungsfeld ein Beitrag geleistet und ein Weg aufgezeigt werden, moderne Rechnerwerkzeuge einzusetzen, um den Umgang mit komplexen Produktportfolios effektiver und effizienter zu gestalten.

Im Mittelpunkt der hier entwickelten Methodik steht die **Unterstützung der Arbeit mit dem bestehenden Produktportfolio**. Die Unterstützung von Neuentwicklungen variantenreicher Produkte mit der Methodik mag ebenso möglich sein, wird hier aber nicht näher betrachtet.

Es gibt zahlreiche **Methoden in der Wissenschaft und Praxis**, die Strukturierung von komplexen Produktprogrammen zu planen und zu optimieren (SCHUH 2005). Bei den Entwurfsmethoden für Baukastensysteme (KÖHLHASE 1997; MARTIN & ISHII 2002; RENNER 2007), Plattformentwicklungen (MEYER & LEHNERD 1997), Modularisierungsmethoden (ULRICH 1995; BLEES 2011; ERIXON 1998), Entwicklung von Baureihen (GERHARD 1984),

---

<sup>2</sup> ERP – Enterprise Resource Planning

<sup>3</sup> Begriffsdefinition in Kapitel 2.1.5

variantengerechte Gestaltung (SCHUH 2005) und anderen stehen primär Neuentwicklungen im Fokus. Jene Methoden zielen auf die Reduktion und Beherrschung von Komplexität ab. Als Planungsgrundlage werden deskriptive Modelle des betrachteten Systems beispielsweise in Form von Matrizen (EPPINGER & BROWNING 2012), Baumstrukturen (SCHUH 1988), Netzwerkstrukturen (LINDEMANN ET AL. 2009) oder formalisierten Beschreibungssprachen wie UML<sup>4</sup>/SysML<sup>5</sup> (ISO 19505-1; ISO 19505-2) herangezogen.

In der Regel wird von einer klaren Datenlage, also von verfügbaren, vollständig dokumentierten und konsistent aufbereiteten Daten ausgegangen. Wechselwirkungen mit den bestehenden Bausteinen eines Produktportfolios und unternehmensspezifische Randbedingungen werden meist nicht berücksichtigt. Auch der nötige Initialaufwand, die entsprechenden Daten aufbereitet zur Verfügung zu stellen um somit die Methoden anwenden zu können, wird normalerweise vorausgesetzt. Hiermit besteht jedoch die erste große **Hürde**, die dazu führen kann, den Einsatz einer Methode angesichts des Initialaufwandes nicht weiter zu verfolgen.

Bei einer Problemstellung, wie sie beispielsweise in Kapitel 1.1.2 bei der Elektronik GmbH beschrieben wurde, **können Methoden** z. B. zur Baukastenentwicklung oder Definition von Baureihen **nicht ohne Weiteres eingesetzt werden**. Höchstens im Fall einer entkoppelten Neuentwicklung, die sich signifikant von Bestehendem absetzt, können diese Konzepte greifen. Jedoch ist nur bei einer geringen Anzahl von Entwicklungsprojekten der Freiheitsgrad so groß, dass ein Produkt mit all seinen Varianten von Beginn an geplant und entwickelt werden kann. Bei einem Großteil der Entwicklungstätigkeiten handelt es sich um Anpassungs- und Änderungstätigkeiten (JARRATT ET AL. 2011, S. 121).

Voraussetzung für die Anwendung bestehender Methoden ist **Klarheit zu haben über die Architektur<sup>6</sup> des vorhandenen Produktportfolios**. Der Ist-Zustand muss bewertet und kritische Handlungsfelder müssen erkannt werden können. Im Entwicklungsumfeld der Systemarchitektur besteht allerdings oft ein Mangel an Transparenz über die Strukturierung des Produktportfolios, Klassifikationen der Bausteine, standardisierte Schnittstellen, oder Ähnliches. Die Vielfalt an unterschiedlichen Bausteinen und Abhängigkeiten erlaubt es nicht, sich ohne Weiteres ein klares Bild über die aktuelle Lage zu verschaffen.

In dieser Arbeit soll daher der Fokus auf die **Reduzierung und Beherrschung der Komplexität** eines bestehenden Produktportfolios gelegt werden. Dies hat zum Ziel, die Grundlage für weitere Optimierungsmaßnahmen am Portfolio zu schaffen. Dafür ist es essentiell, dass sich der Systemarchitekt einerseits **effektiv und effizient** ein Bild des Gesamtsystems machen kann und andererseits **schnell kritische Handlungsfelder** im Detail des Portfolios identifizieren und diese bewerten kann. Die große Anzahl und Vielfalt an Elementen und Wechselwirkungen in einem Produktportfolio führt dazu, dass rasch der **Überblick verloren geht** und der **Aufwand, die relevanten Informationen zusammenzutragen und aufzubereiten, zu groß** wird. Angesichts der großen, ungefilterten

---

<sup>4</sup> Unified Modeling Language (UML)

<sup>5</sup> Systems Modeling Language (SysML)

<sup>6</sup> Begriffsdefinition in Kapitel 2.1.4

Informationsflut kommt es „auf die richtige Auswahl“ der Information an, um rationale Entscheidungen in komplexen Systemen treffen zu können (VESTER 2000, S. 20). Die Fragestellung, die in das Zentrum dieser Untersuchung gestellt wird, lautet demnach:

*Wie lässt sich im Entwicklungsumfeld der Systemarchitektur das Arbeiten mit dem Produktportfolio angesichts der hohen Komplexität effektiver und effizienter gestalten?*

Aus der Forschungsfrage geht als Hauptziel hervor, Komplexität in bestehenden technischen Produktportfolios zu reduzieren und zu beherrschen. Dieses Ziel lässt sich weiterhin operationalisieren. Die folgenden drei Teilziele gilt es mit der zu entwickelnden Methodik zu adressieren:

*Forschungsziel (1): Transparenz schaffen in einem komplexen Produktportfolio*

Mit der Erhöhung der Transparenz über die Elemente und Abhängigkeiten innerhalb eines Produktportfolios können die relevanten Zusammenhänge vollständiger in der Bewertung, Planung und Gestaltung berücksichtigt werden. Somit können Entscheidungen auf einer rationaleren Basis getroffen werden. In Anbetracht der unübersichtlichen Informationsflut wird die wahrgenommene Komplexität dadurch reduziert, dass unwesentliche Informationen gezielt ausgeblendet werden.

*Forschungsziel (2): Effektiveres Arbeiten mit komplexen Produktportfolios*

Durch den Einsatz der Graphtransformation soll eine Effektivitätssteigerung in der Arbeit des Systemarchitekten erreicht werden. Die Graphtransformation soll eine Formalisierung und Automatisierung von langwierigen Arbeitsgängen ermöglichen. Bei der Arbeit mit großen Strukturen soll das Reaktionsvermögen durch schnelles Erkennen von Suchmustern und somit die Identifikation von Handlungsfeldern erhöht werden. Zudem soll die Qualität der Analyse verbessert werden.

*Forschungsziel (3): Effizienteres Arbeiten mit komplexen Produktportfolios*

Der Aufwand für Analysen und Bewertungen soll gesenkt werden. Vor allem die Informationsbeschaffung und –aufbereitung sollen wirtschaftlicher gestaltet werden. Repetitive Aufgaben sollen nicht mehr „von Hand“ durchgeführt werden müssen, sondern automatisierbar sein.

Nachfolgende Abbildung fasst die Zielstellung der Arbeit noch einmal zusammen (vgl. Abbildung 1-3)

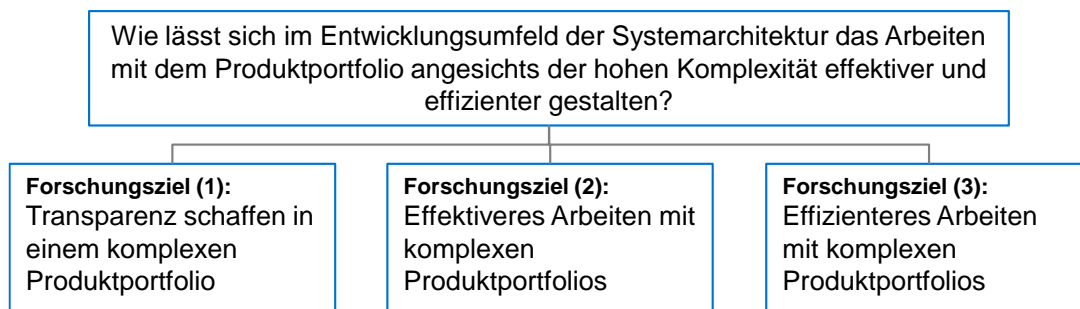


Abbildung 1-3: Zielstellung der Arbeit

In dieser Arbeit wird eine Methodik vorgestellt, die den Systemarchitekten in seiner täglichen Arbeit unterstützen soll. Die Pflege der Daten eines Produktportfolios, aber auch die Überwachung der Wirtschaftlichkeit und die Bedienung von Stakeholdern mit benötigten Informationen in den Unternehmensprozessen wird mit steigender Komplexität des Produktportfolios schwieriger. Daher soll ein deskriptives Modell eines Produktportfolios als Planungs- und Entwicklungsgrundlage dienen, in dem die wesentlichen Informationen und Abhängigkeiten abgelegt sind. Mit der hier vorgestellten Methodik soll dieses Portfoliomodell so auswertbar sein, dass folgende Aufgaben effektiver und effizienter durchgeführt werden können:

- **Bewertung des Ist-Zustandes erleichtern** durch die automatische Berechnung und Bereitstellung von relevanten Kennwerten über das Produktportfolio.
- **Kommunikation komplexer Sachverhalte** gegenüber unterschiedlichen Stakeholdern **unterstützen** durch fallspezifische Reduktion komplexer Zusammenhänge auf relevante Informationen.
- **Pflege der Produktstruktur ermöglichen**, wie z. B. bestehende Bausteine anhand ihrer Eigenschaften neu zu klassifizieren und zu strukturieren für eine klarere Architektur des Produktportfolios, effizientere Wiederverwendung von Bausteinen und sinnvollere Standardisierungen.
- **Entscheidungen über Anpassungen** für eine marktgerechtere Vielfalt des Produktportfolios **transparent machen** und den Planungsprozess durch effizientere Informationsbereitstellung und durch effektivere Muster- und Fehlersuche unterstützen und beschleunigen.

Die oben genannten Aufgaben sollen durch die Anwendung der Graphtransformation realisiert werden. Die *Methodik zur Mustererkennung in komplexen Produktportfolios* wird in den folgenden Kapiteln vorgestellt und diskutiert.

Anhand eines **rechnergestützten Demonstrators** soll die Methodik prototypisch umgesetzt werden. Mit dem Demonstrator soll ein „Proof of Concept“ erbracht werden. Damit sollen die Umsetzbarkeit der Methodik und die Generierung exemplarischer Ergebnisse für Fragestellungen in zwei Fallstudien in der Industrie gezeigt werden. Es wird im Rahmen dieser Arbeit davon abgesehen, eine fertige Softwarelösung bereitzustellen. Der Anspruch liegt vielmehr darin, den Mehrwert der erzielbaren Ergebnisse für das Produktportfoliomanagement aufzuzeigen.

## 1.3 Thematische Einordnung und Forschungsmethodik

In diesem Abschnitt erfolgt eine thematische Einordnung der Arbeit. Darüber hinaus wird die dieser Forschungsarbeit zugrundeliegende Forschungsmethodik erläutert.

### 1.3.1 Thematische Einordnung der Arbeit

Aus wissenschaftlicher Sicht sollen in dieser Forschungsarbeit drei Themenfelder zusammengeführt werden. Zwei davon – die variantenreiche Produktgestaltung und das Systems Engineering – überlappen sich stark und sind in vielen Aspekten kohärent. Das dritte Themenfeld – die Graphtransformation – wurde bislang noch nicht mit den beiden anderen intensiver in Verbindung gebracht. Abbildung 1-4 gibt einen Überblick über die thematische Einordnung der Arbeit.

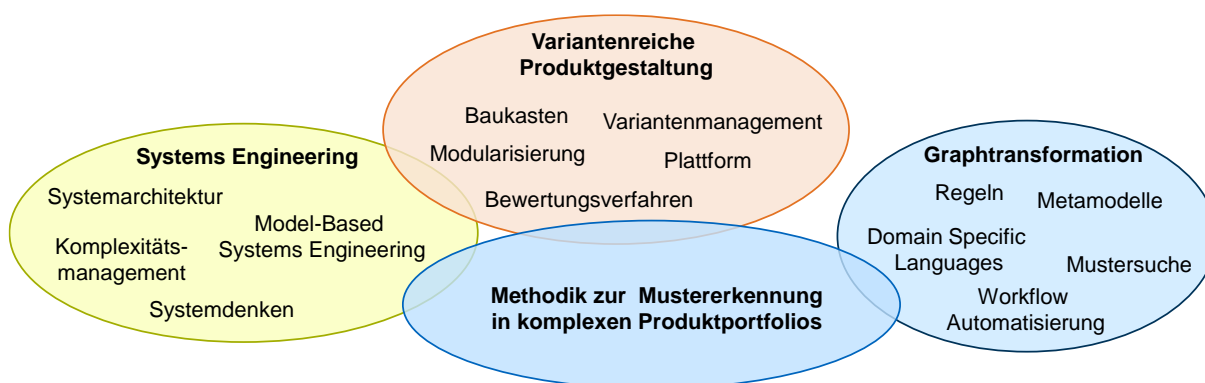


Abbildung 1-4: Thematische Einordnung der Arbeit

Die **Graphtransformation** ist ein Forschungsgebiet, das seit über 40 Jahren in der Informatik beheimatet ist. In dieser Disziplin werden Techniken für die Handhabung dynamischer graphenbasierter Strukturen entwickelt. Anwendungsfelder sind beispielsweise Rekonfigurationen von Softwarearchitekturen, Modelltransformationen, Refactoring etc.<sup>7</sup>. Allerdings finden sich weitere Anwendungsfelder beispielsweise in der Bioinformatik, Geschäftsprozess Modellierung oder der Ontologieforschung.

Im Rahmen dieser Forschungsarbeit sollen nun die Methoden der Graphtransformation im Kontext des Produktportfoliomanagements angewendet werden. Dafür wird das generative Programmiersystem GrGen.NET<sup>8</sup> verwendet (BLOMER ET AL. 2013). Dieses erlaubt die Definition einer eigenen domänenspezifischen Modellierungssprache. Die Idee ist es, mittels eines Metamodells die Struktur des Produktportfolios als Graph abzubilden. Anschließend soll Fachwissen der Systemarchitektur in Form von Transformationsregeln formalisiert und auf diesen Graph angewendet werden. Somit kann nach Mustern in einem komplexen Produktportfoliographen gesucht und entsprechende Ersetzungsmuster und Aktionen definiert werden.

<sup>7</sup> Aus dem Call for Papers zur 6<sup>th</sup> International Conference on Graph Transformation

<sup>8</sup> Graph Rewrite Generator auf .NET-Technologie (GrGen.NET)

Die Themenfelder Systems Engineering und variantenreiche Produktgestaltung sind hingegen klassische Themenfelder der Ingenieurwissenschaften. Das **Systems Engineering** liefert im Rahmen dieser Arbeit das grundlegende Systemverständnis und Systemdenken. Mit diesem Verständnis werden Anforderungen an die Methodik aus der Systemarchitektur abgeleitet. Aus dem Komplexitätsmanagement werden Anregungen eingeholt und fließen entsprechend in den Lösungsansatz mit ein.

Ebenso liefert der Stand der Technik der **variantenreichen Produktgestaltung** grundlegendes Verständnis über die Aufgaben und Methoden der Systemarchitektur und ist mit den darin enthaltenen Systematiken Inspiration für den Lösungsansatz. Methoden, Bewertungsverfahren und Gestaltungsprinzipien der Plattform- und Baukastenentwicklung, der Modularisierung und des Variantenmanagements sollen hinsichtlich ihres Unterstützungspotentials für die Ziele in der Systemarchitektur untersucht werden. Zudem soll geprüft werden, inwiefern bestehende Ansätze für die Wissensformalisierung mittels Transformationsregeln genutzt werden können.

### 1.3.2 Forschungsmethodik

Neben den inhaltlichen Zielen werden wissenschaftliche Anforderungen an diese Forschungsarbeit gestellt (BIEDERMANN ET AL. 2012, S. 15ff). So ist die **Relevanz** des zu lösenden Problems sowohl im akademischen wie auch im industriellen Bereich herauszustellen. Hierfür wurde einerseits eine umfangreiche Literaturrecherche durchgeführt, mit der die Relevanz des Problems aus wissenschaftlicher Sicht untermauert wird. Andererseits wurde anhand exemplarischer Problemstellungen in zwei Fallstudien die Relevanz des Problems aus industrieller Sicht aufgezeigt. Hierzu wurden mit Experten der jeweiligen Industrieunternehmen Workshops durchgeführt, um die Problemstellung herauszuarbeiten und Schwerpunkte der Fallstudien zu definieren.

Durch ein systematisches Vorgehen, Anwendung geeigneter Forschungsmethoden und eine fundierte Wissensbasis wurde die **Stichhaltigkeit** der Ergebnisse sichergestellt. Durch genaue Angaben zum Vorgehen, klare Schlussfolgerungen und kritische Hinterfragung des Vorgehens wurde für **Objektivität** und **Nachvollziehbarkeit** der Ergebnisse gesorgt (MILES & HUBERMAN 1994). Das Forschungsvorgehen wie auch der Aufbau der Arbeit orientieren sich weitgehend an der **Design Research Methodology** (DRM) nach BLESSING & CHAKRABARTI (2009).

Zunächst wurde ein grundlegendes Verständnis für die Problemstellung erarbeitet. Im ersten Schritt der **Klärung der Forschungsziele** (*research clarification*) wurde literaturbasiert der Fokus der eigenen Forschung erarbeitet und die Forschungsziele definiert. Grundlage für diesen Schritt bilden Erfahrungen und Beobachtungen schwerpunktmäßig aus den beiden Fallstudien, aber auch aus industrienahen Forschungsprojekten mit weiteren insgesamt 18 Industrie-, Beratungs- und IT-Unternehmen. Das Problemverständnis wurde in der **deskriptiven Studie I** vertieft und Erfolgsfaktoren abgeleitet. Daraus wurden Anforderungen an den Lösungsansatz abgeleitet und der Beitrag der Arbeit spezifiziert. In der **präskriptiven Studie** wurde darauf aufbauend der eigene Lösungsansatz erarbeitet. Dieser wurde in der **deskriptiven Studie II** anhand zweier Fallstudien evaluiert.

Für deskriptive Fallstudien führt YIN (2013, S. 45ff) unter anderen Verlässlichkeit (*reliability*) und externe Validität (*external validity*) als wichtige Kriterien auf. Für die Sicherstellung der

**Verlässlichkeit** wurde im Rahmen der Fallstudien auf eine klare Zielstellung, Konsolidierung mehrerer aussagekräftiger Quellen und die kontinuierliche Überprüfung der Datenqualität geachtet.

In der präskriptiven Studie wurde die Methodik zur Mustererkennung in komplexen Produktportfolios mit dem Anspruch erarbeitet, allgemein im Rahmen des Produktportfoliomanagements einsetzbar zu sein. Die **Übertragbarkeit der Methodik** wurde exemplarisch anhand der beiden Fallstudien nachgewiesen. Zudem sollte auch der **Nutzen** und die **Anwendbarkeit** der Methodik in den Fallstudien geprüft werden. Hierfür wurden Feedbackgespräche mit den involvierten Experten geführt, um Vor- und Nachteile des Einsatzes der Methodik kritisch zu diskutieren.

## 1.4 Struktur der Arbeit

Im ersten Schritt nach diesem einleitenden Kapitel wird in **Kapitel 2** die Problemstellung aufbereitet. Zunächst werden relevante Begrifflichkeiten geklärt. Anschließend werden die Rolle, die Aufgaben und die Ziele des Systemarchitekten näher erläutert, um somit die Problemstellung der Arbeit einzuordnen und die Voraussetzungen für eine erfolgreiche Architekturarbeit im Unternehmen zu klären.

Im Anschluss daran werden die Methoden und Konzepte zur Strukturierung komplexer Produkte im Stand der Technik diskutiert (**Kapitel 3**). Nach Klärung der Grundbegriffe werden Methoden zur Komplexitätsreduktion im Produktportfoliomanagement vorgestellt und eingeordnet. Daraufhin wird untersucht, welche Defizite heute im Umgang mit Komplexität vorhanden sind und welche Hürden sich dadurch für die Erreichung der Ziele der Systemarchitektur ergeben. Aus den Schlussfolgerungen werden die Anforderungen an einen Lösungsansatz abgeleitet und der angestrebte Beitrag der Arbeit formuliert.

Die zugrundeliegende Idee der Arbeit ist es, die Methoden des Produktportfoliomanagements mit der Technologie der Graphtransformation aus der Informatik zu verbinden, um somit Verbesserungen in der Systemarchitekturarbeit erreichen zu können. In die Graphtransformation führt **Kapitel 4** ein. Vorab wird in die Modellierung komplexer Produktstrukturen eingeführt, anschließend werden die Konzepte der Metamodellierung mit domänenspezifischen Modellierungssprachen, der Wissensformalisierung mit Regeln und der softwaretechnischen Umsetzung vorgestellt.

In **Kapitel 5** wird schließlich der Ansatz zur Mustererkennung in komplexen Produktstrukturen erläutert. Es werden das Rahmenwerk der Methodik und dessen Bestandteile dargestellt: die Datenaufbereitung, die Wissensformalisierung und die Ergebnisaufbereitung.

Der vorgestellte Ansatz wird in **Kapitel 6** anhand zweier Fallstudien in der Industrie evaluiert. Die Graphtransformation wird im Produktportfoliomanagement der Hilti Entwicklungsgesellschaft mbH eingesetzt, um mehr Transparenz in den Abhängigkeiten des Produktportfolios der Antriebsentwicklung zu ermöglichen. Bei der Elektronik GmbH unterstützt die Methode bei der Abkündigung von unwirtschaftlichen Produkten und Bausteinen.

Die Methodik, Schlussfolgerungen sowie Implikationen für die industrielle Anwendung werden daraufhin im **Kapitel 7** diskutiert. Mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf die weiteren Schritte (**Kapitel 8**) schließt diese Arbeit ab. Nachfolgende Abbildung 1-5 fasst den Aufbau der Arbeit in einer Übersicht zusammen.



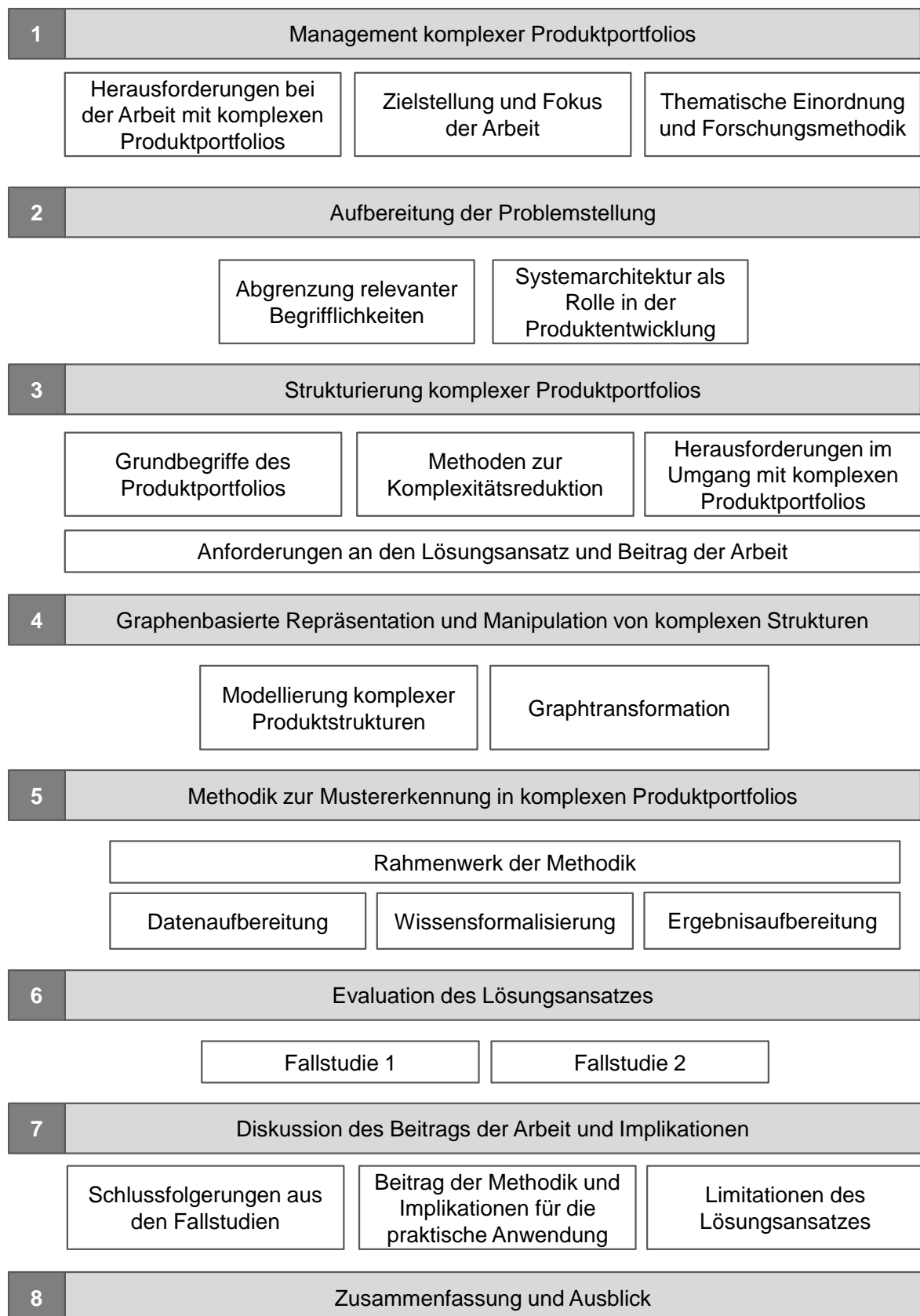


Abbildung 1-5: Struktur der Arbeit



## 2. Aufbereitung der Problemstellung

In diesem Kapitel soll die Problemstellung, wie im vorherigen Kapitel dargelegt, detaillierter analysiert werden. Zunächst werden die begrifflichen Grundlagen geklärt. Anschließend wird die Systemarchitektur als Rolle im Unternehmen analysiert. Es wird untersucht, welche Aufgaben ein Systemarchitekt erfüllen muss und welche Ziele er dabei verfolgt. Die Problemstellung wird abschließend dahingehend geschärft, welche Voraussetzung gegeben sein müssen, um eine erfolgreiche Architekturarbeit durchführen zu können.

### 2.1 Abgrenzung relevanter Begrifflichkeiten

Zunächst werden die zentralen Begrifflichkeiten definiert, um das Verständnis für die folgenden Ausführungen zu erhöhen. Daher wird das in dieser Arbeit zu Grunde liegende Systemverständnis erläutert und in diesem Zusammenhang der Begriff Komplexität geklärt. Anschließend werden die Grundzüge des Systems Engineering, der Architektur technischer Systeme sowie das Wesen der Mustererkennung dargelegt.

#### 2.1.1 Systemverständnis

Für den Begriff „System“ finden sich zahlreiche Definitionen in der Literatur. In der Antike wurde mit dem Begriff „System“ von der Ganzheit gesprochen, die aus Teilen besteht. LUHMANN ersetzt in der Sozialforschung diese Definition durch die „Differenz von System und Umwelt“ (LUHMANN 1991, S. 20ff). WEINBERG (1975, S. 52) beschreibt ein System als eine spezifische Art, auf die Welt zu blicken. Diese Definitionen sind für den Kontext dieser Arbeit zu abstrakt. Für technische und sozio-technische Systeme, wie sie im **Ingenieurbereich** auftreten, finden sich einander ähnliche Definitionen wie die von EHRENSPIEL & MEERKAMM (s.u.) bei DAENZER (1977, S. 14) oder auch PAHL ET AL. (2007, S. 17ff).

Hier soll der **Begriff „System“** nach der Definition von EHRENSPIEL gelten: *„Ein System besteht aus einer Menge an Elementen (Teilsystemen), die Eigenschaften besitzen und durch Beziehungen miteinander verknüpft sind. Ein System wird durch eine Systemgrenze von der Umgebung abgegrenzt und steht mit ihr durch Ein- und Ausgangsgrößen in Beziehung (offenes System). Die Funktion eines Systems kann durch den Unterschied der dem Zweck entsprechenden Ein- und Ausgangsgrößen beschrieben werden [...]. Nach ihrem Verhalten können statische und dynamische Systeme unterschieden werden“* (EHRENSPIEL & MEERKAMM 2013, S. 21). Auch immaterielle Elemente wie Dienstleistungen können in Kombination mit physischen Elementen Teil eines sog. „Product-Service-Systems“ (PSS) sein (ABRAMOVICI & SCHULTE 2005, S. 1). Das Denken in Systemen wird bei HABERFELLNER ET AL. beschrieben. Es ist Grundlage des Systems Engineering, das im nächsten Abschnitt beschrieben wird (HABERFELLNER ET AL. 2012, S. 39).

**Teilsysteme** oder auch Subsysteme sind nach EHRENSPIEL & MEERKAMM (2013, S. 23) Bestandteil eines **übergeordneten Systems** und können ihrerseits weiter untergliedert werden in Teilsysteme (vgl. Abbildung 2-1). Somit können unterschiedliche Detaillierungsebenen

erzeugt werden. Ein Produkt kann demnach als eigenes System oder auch als Teilsystem eines übergeordneten Systems definiert werden. Je nach gewählter Aggregationsstufe kann beispielsweise ein Produkt ein Teilsystem des übergeordneten Systems Produktfamilie sein. Das Produkt selbst kann wiederum in Teilsysteme zerlegt werden, also in seine Module, Submodule und Einzelteile.

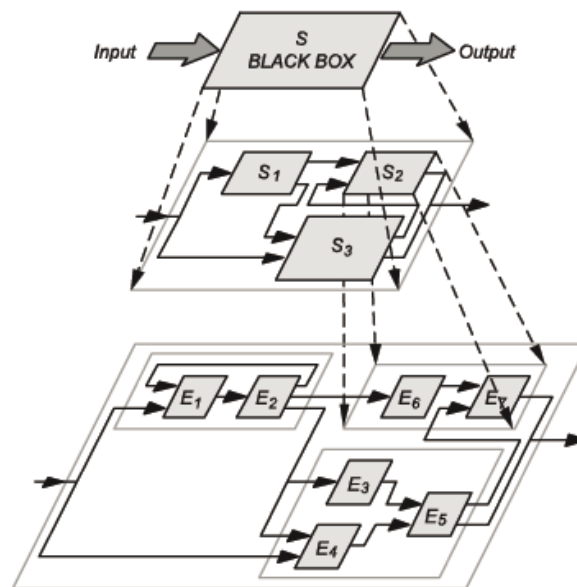


Abbildung 2-1: Die Struktur eines Systems in unterschiedlicher, hierarchischer Detaillierung nach EHRENSPIEL & MEERKAMM (2013, S. 23)

Je nach Modellierungszweck muss eine sinnvolle Systemgrenze des Betrachtungsumfangs und eine geeignete Aggregationsstufe gefunden werden. Mit dem Ziel, relevante Information im Produktportfoliomanagement in Modellen abbilden zu können, werden in den nächsten Kapiteln wesentliche Elemente und Beziehungen herausgearbeitet. Die Elemente und Beziehungen in einem System bestimmen auch wesentlich die Komplexität eines Systems.

### 2.1.2 Komplexität von Systemen

Die Komplexität eines Systems entsteht in der **Wahrnehmung des Betrachters**. Einerseits kann ein System als komplex empfunden werden, wenn es unüberschaubar ist und sich seine Struktur oder Ordnung nicht erschließen lässt. Andererseits kann auch durch den „Umfang, die Handlichkeit und die innere Vermaschung“ eine Überforderung der menschlichen Wahrnehmung entstehen (PATZAK 1982, S. 22). Auch Nicht-Linearität und nicht deterministisches Systemverhalten führen zu einem komplexen Eindruck. WEAVER (1948, S. 537ff) differenziert zwischen organisierter und disorganisierter Komplexität.

Nach PATZAK (1982, S. 22) bedeutet Komplexität in seiner etymologischen Bedeutung „sowohl ‚zusammenhängend‘ als auch ‚vielumfassend‘“. Strukturelle Komplexität definiert er mit den zwei Begriffen:

- Die **Konnektivität** wird durch Art (Beziehungsinhalt) und Anzahl (Verknüpfungsdichte) der Beziehungen im System bestimmt.

- Die **Varietät** wird durch Art (Unterschiedlichkeit der Elemente) und Anzahl (Elementenmenge) der Elemente im System bestimmt.

RENNER (2007, S. 27) führt neben Konnektivität und Varietät weitere Facetten der Komplexität an:

- **Dynamik** mit Art und Anzahl möglicher Zustände,
- **Wechselwirkungen** mit dem Umfeld,
- **Unschärfen** und **Unsicherheiten** im System.

Nach PILLER & WARINGER (1999, S. 20) entsteht Komplexität im industriellen Umfeld durch Kunden-, Varianten-, Teile-, und Lieferantenvielfalt sowie durch Produkt- und Prozessinnovationen und Fertigungssysteme.

Auf Grund der hohen Komplexität zahlreicher technischer Systeme und Entwicklungsprojekte mussten Vorgehensmodelle und Methoden entwickelt werden, um die Problemlösung in Teilprobleme untergliedern und jeweils von Spezialisten abarbeiten zu können. Unter Begriffen wie Systemtechnik, integrierte Produktentwicklung oder Systems Engineering wurden hierfür methodische Werkzeuge geschaffen.

### 2.1.3 Grundzüge des Systems Engineering

Das Systems Engineering hat seine Ursprünge in den 1930er Jahren, als begonnen wurde, **technische Großprojekte** zu initiieren, die mit den üblichen Mitteln nicht mehr effektiv bearbeitet werden konnten. Ein Beispiel aus den 30er Jahren war die Entwicklung von Flugabwehrsystemen durch die Briten (HASKINS ET AL. 2010, S. 8). Weitere prominente Projekte waren die Mondlandung oder das Space Shuttle Programm der NASA (2007). Als nicht militärisches Beispiel kann die Entwicklung des Telefonnetzes durch die Bell Telephone Company<sup>9</sup> angeführt werden.

In Europa wurden weitgehend ähnliche Methoden wie die des Systems Engineering unter den Begriffen **Systemtechnik** (ROPOHL 1975; BRUNS 1991) und **Integrierte Produktentwicklung** entwickelt (vgl. ANDREASEN & HEIN 1987; EHRENSPIEL & MEERKAMM 2013). Allgemein beschreibt z. B. PATZAK die Systemtechnik als *„ein Methodengebäude zur Behandlung von Problemen mit hoher Komplexität. Sie befasst sich mit sämtlichen Lebensphasen von Systemen, d.h. mit der Planung und Realisierung einschließlich der Organisation der Nutzung und Außerdienststellung derselben, welche die ihnen zugedachten Funktionen voll erfüllen, sich optimal in ihre Umwelt einfügen sowie deren Subsysteme ‚reibungslos‘ zusammenwirken“* (PATZAK 1982, S. 15).

HELLENBRAND sieht trotz der unterschiedlichen Ursprünge **Überlappungen** zwischen Systems Engineering und Integrierter Produktentwicklung, beschreibt allerdings unterschiedliche Schwerpunkte. Beiden Ansätzen sei beispielsweise die Betonung der transdisziplinären, anforderungsgerechten Entwicklung technischer Systeme über den gesamten Lebenszyklus gemein (HELLENBRAND 2013, S. 57). EHRENSPIEL & MEERKAMM (2013, S. 19) sehen enge

---

<sup>9</sup> <http://www.corp.att.com/history/history1.html> (entnommen am 27.12.2013)

Beziehungen der Systemtechnik als Denkansatz zur Konstruktionswissenschaft nach HUBKA & EDER (1992) und PAHL ET AL. (2007).

Drei wesentliche Faktoren haben zur Entwicklung des Systems Engineering beigetragen (KOSSIAKOFF ET AL. 2011, S. 6ff):

- **Risiken in der Entwicklung durch fortschreitende Technologien**, die ein Risikomanagement zwingend erforderlich machen.
- **Steigender Wettbewerbsdruck** und der damit verbundene Zwang wirtschaftlicher und schneller mit besseren Produkten als die Konkurrenz auf den Markt zu kommen.
- **Spezialisierung in den Fachdisziplinen**, die eine Dekomposition komplexer Systeme in Teilprobleme und ein striktes Management der Schnittstellen und Wechselwirkungen erfordert.

Das Systems Engineering ist ein **Top-Down-Ansatz**. KOSSIAKOFF ET AL. (2011, S. 3) beschreiben das Systems Engineering als Lenkungsfunktion im Unternehmen für die Entwicklung komplexer Systeme. Der International Council on Systems Engineering (INCOSE) bietet folgende **Definition** an (HASKINS ET AL. 2010, S. 7):

*“Systems engineering (SE, A.d.V) is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing, and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.”*

Aus dieser Definition gehen die thematischen Schwerpunkte hervor. Zudem wird der Durchlauf durch das **V-Modell** (VDI 2206, S. 29ff) beschrieben (vgl. Abbildung 2-2). Von der Erfassung und Analyse der Kundenanforderung und Funktionen, über die Systemarchitektur und der Zerlegung der Entwicklungsaufgabe in Teilprobleme bis hin zur eigentlichen Realisierung läuft man den linken Ast des V-Modells von oben nach unten durch. Auf der rechten Seite werden die Einzellösungen erst zu Teil- und schließlich zum Gesamtsystem integriert, getestet und die Kundenanforderungen validiert.

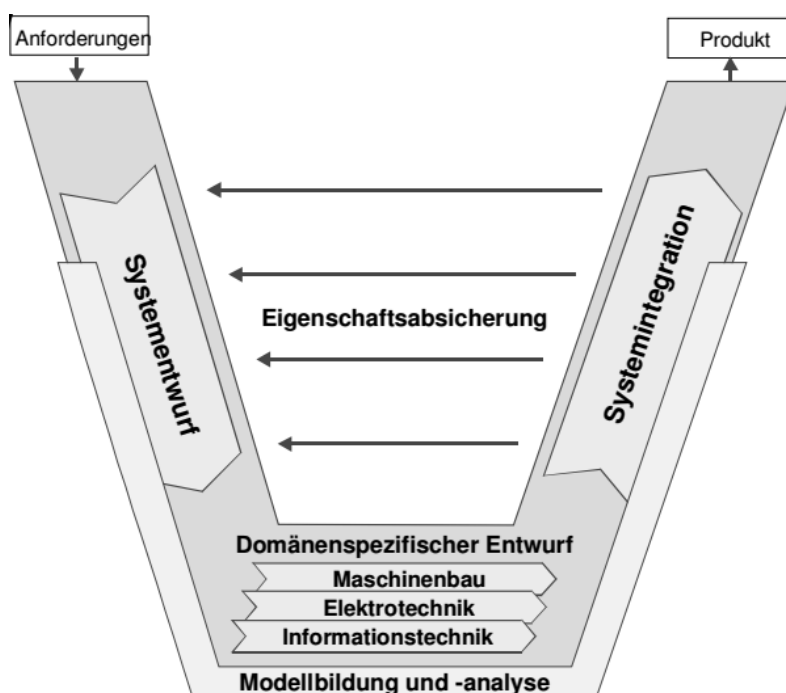


Abbildung 2-2: Das V-Modell als zentraler Ansatz im Systems Engineering (VDI 2206, S. 29ff)

Die NASA beschreibt die oben beschriebenen Tätigkeiten des hierfür verantwortlichen Systemingenieur folgendermaßen: „*The system engineer will usually play the key role in leading the development of the system architecture, defining and allocating requirements, evaluating design tradeoffs, balancing technical risk between systems, defining and assessing interfaces, providing oversight of verification and validation activities, as well as many other tasks*” (NASA 2007, S. 3).

Neben den im V-Modell dargestellten technischen Prozessen, werden im IEEE<sup>10</sup> Standard 15288-2004 (IEEE 2005) zusätzliche Aktivitäten des Systems Engineering definiert. So werden Aufgaben im **Projektmanagement, Unternehmens- und Abstimmungsprozessen** beschrieben (vgl. Abbildung 2-3). Es werden Sekundärprozesse beschrieben, die zur Unterstützung der eigentlichen technischen Prozesse dienen. Die Projektprozesse<sup>11</sup> umfassen Planung, Prüfung, Kontrolle, Entscheidungsfindung, Risikomanagement, Konfigurationsmanagement und Informationsmanagement (HASKINS ET AL. 2010, S. 5.1ff).

<sup>10</sup> Institute of Electrical and Electronics Engineers (IEEE)

<sup>11</sup> Im Handbuch der NASA „Technisches Querschnittsmanagement“ genannt (NASA 2007)

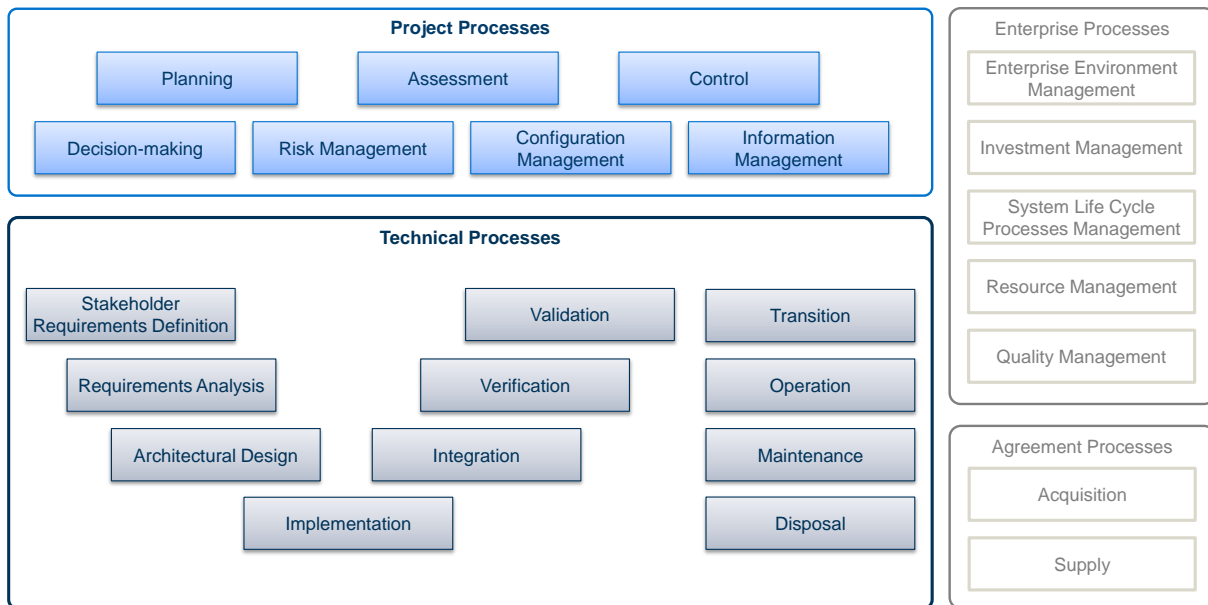


Abbildung 2-3: System Lebenszyklusprozesse, Überblick nach IEEE (2005, S. 49)

In der europäischen Industrie wird das Systems Engineering eine immer prominentere Disziplin. Neben Großprojekten zur Entwicklung großer technischer Einzelsystemen werden die Methoden des Systems Engineering auch für die Entwicklung variantenreicher, mechatronischer Produkte mit großen Stückzahlen eingesetzt, wie z. B. im Fahrzeug- oder Maschinenbau (GAUSEMEIER 2012). Hierfür müssen die Methoden und Vorgehensweisen des Systems Engineering entsprechend angepasst oder neu interpretiert werden. Für weitere Details zum Systems Engineering wird auf Department of Defense 2008; HABERFELLNER ET AL. 2012; HASKINS ET AL. 2010; NASA 2007; ROUSE & SAGE 2009; KOSSIAKOFF ET AL. 2011; DAENZER 1977 und andere verwiesen.

Ein zentraler Aspekt im Systems Engineering ist die Gestaltung der Systemarchitektur. Mit der Architektur technischer Systeme werden die Kundenwünsche und Anforderungen in Vorgaben für die technische Entwicklung übersetzt und maßgeblich die nachfolgenden Prozesse im Unternehmen bestimmt.

### 2.1.4 Architektur technischer Systeme

Die Architektur ist ein Merkmal eines beliebigen technischen Systems (MAURER 2007, S. 30f), sei es einer Baugruppe, eines Produktes, einer Produktfamilie oder eines Produktportfolios. Jedes technische System weist eine Architektur auf<sup>12</sup> – geplant oder ungeplant (MARTIN & ISHII 2002, S. 214). Die Wichtigkeit der Architektur spiegelt sich in drei Aspekten wider (CRAWLEY ET AL. 2004, S. 1):

<sup>12</sup> “Every system has an architecture. Architectures may arise in the process of deliberate de novo design of a system; by evolution from previous designs with strong legacy constraints; by obeying regulations, standards, and protocols; by accretion of smaller systems with their own architectures; or by exploration of form and behavioral requirements via dialogue between users and architects, to name a few known mechanisms.” (CRAWLEY ET AL. 2004, S. 2)



- als eine Zugangsform komplexe Systeme zu verstehen,
- zu entwickeln und
- diese auf längere Zeit wirtschaftlich durch Standardisierung zu pflegen.

Recht abstrakt definieren CRAWLEY ET AL. **Systemarchitektur** mit *“an abstract description of the entities of a system and the relationships between those entities”* (CRAWLEY ET AL. 2004, S. 2). Im DoD Architecture Framework wird eine **Architektur** definiert als *„structure of components, their relationships, and the principles and guidelines governing their design and evolution over time“* (Department of Defense 2008, S. 19). Im Rahmen des Engineering Design wird oft der Begriff **„Produktarchitektur“** verwendet. Eine vielzitierte Definition für Produktarchitektur bietet ULRICH (1995, S. 420) an. Er definiert die Produktarchitektur als

- die Anordnung von funktionalen Elementen,
- deren Zuordnung zu physischen Komponenten sowie
- die Spezifikation der Schnittstellen zwischen den physischen Komponenten.

Während ULRICH die Zuordnung von Komponenten zu funktionalen Elementen und die Spezifikation von Schnittstellen betont, werden im DoD Architecture Framework die Gestaltungsprinzipien und Leitlinien zur Weiterentwicklung als Teil der Architektur benannt. Gemeinsam ist beiden Definitionen die Strukturierung von Komponenten und Relationen.

RAPP (2010, S. 9) kritisiert an der Definition nach ULRICH die Vernachlässigung von Beziehungen zwischen Komponenten ohne gemeinsame physikalische Schnittstelle, wie z. B. Form- und Lagebeziehungen. Er setzt den Begriff Produktarchitektur mit der Produktstruktur gleich. Damit folgt er einer ähnlichen Definition nach EVERSHEIM (1989, S. 145). Seine Definition von Produktarchitektur beschränkt sich dabei auf die *„Zusammensetzung eines Erzeugnisses aus Teilelementen und deren Zuordnung untereinander“* (RAPP 2010, S. 9).

GÖPFERT (1998, S. 97) definiert die *„Architektur eines Produktes durch seine Funktions- und die Baustruktur sowie deren Transformationsbeziehungen“* (vgl. Abbildung 2-4). Dies erweitert die Definition von ULRICH (1995) um Aggregationsstufen auf Funktions- und Komponentenseite.

Systemarchitektur kann auch als Aufgabe oder als Rolle im Unternehmen verstanden werden. Der Systemarchitekt ist mit der Planung, der Aufbausteuerung und der Überwachung der Systemarchitektur betraut. Der Begriff *„Systems Architecting“* wurde erstmals von Eberhardt Rechtin (University of Southern California) verwendet (KOSSIAKOFF ET AL. 2011, S. 222). Im Kapitel 2.2.1 wird hierauf näher eingegangen.

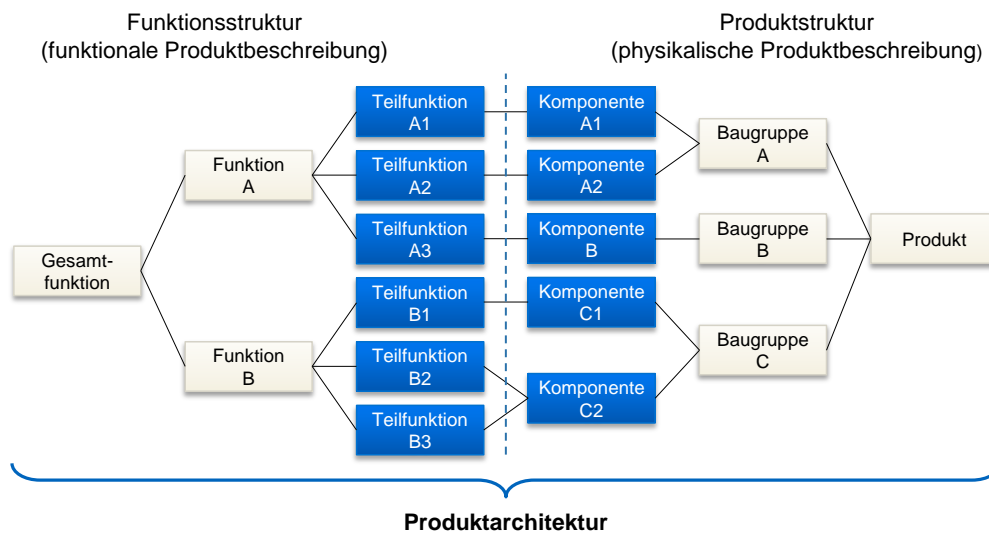


Abbildung 2-4: Architektur aus Funktions- und Baustruktur (in Anlehnung an GÖPFERT 1998, S. 97ff)

In dieser Arbeit bieten die Definitionen einer Produktarchitektur von ULRICH (1995) oder auch von GÖPFERT (1998) eine gute Arbeitsgrundlage. Mit der Betrachtung von komplexen Produktportfolios verschiebt sich allerdings die Systemgrenze des betrachteten Objektes von einem Produkt auf das gesamte Produktportfolio. Angelehnt an die abstraktere Definition von CRAWLEY ET AL. (2004) soll im Folgenden die „**Produktportfolioarchitektur**“ untersucht werden. Neben der Kopplung von Funktionen und Komponenten werden weitere planungsrelevante Typen von Elementen und Relationen berücksichtigt, wie die Einbindung der Komponenten in Produkte und Produktfamilien, die Wechselwirkungen mit Unternehmensprozessen oder auch die Verknüpfung mit verantwortlichen Personen. Elemente können hierbei auf unterschiedlichen Aggregationsstufen (z.B. Produktfamilie – Produkt – Komponente) betrachtet und weitere Elemente unterschiedlicher Domänen wie Funktionen, Verantwortlichkeiten, Projekte, Dokumente herangezogen werden.

Bei den Relationen wird zwischen horizontalen und vertikalen Abhängigkeiten unterschieden (GÖPFERT 1998, S. 18ff). In der objektorientierten Programmierung spricht man bei horizontalen Relationen von Assoziation, bei vertikalen von Komposition oder Aggregation. **Horizontale Assoziationen** können Flussbeziehungen wie Material-, Informations-, oder Energieflüsse sein oder auch logische Beziehungen wie geometrische Schnittstellen oder Lagebeziehungen (vgl. auch PIMMLER & EPPINGER 1994, S. 4). Auch komplexere Ausdrücke wie Konfigurationsregeln in Baukastensystemen können als Beziehung zwischen zwei oder mehreren Bausteinen ausgedrückt werden. **Vertikale Abhängigkeiten** können hierarchische Beziehungen ausdrücken. Im Systems Engineering wird meist die folgende top-down Klassifikation verwendet: System – Segment – Element – Subsystem – Baugruppe – Unterbaugruppe – Teil (NASA 2007, S. 3). Bei Komposition und Aggregation handelt es sich jeweils um eine „ist Teil von“-Beziehung. Bei der Aggregation (z. B. Produktfamilie – Produkt) kann das Teilobjekt ohne das Aggregationsobjekt existieren, während bei der Kompositionsbeziehung (z. B. Produkt – Modul) eine existentielle Abhängigkeit zwischen den Objekten besteht.

Ein weiterer Aspekt zur Definition einer Produktportfolioarchitektur sind die zu Grunde liegenden **Gestaltungsprinzipien**, also die konzeptionelle Ausgestaltung einer

Produktplattform, die Festlegung von Architekturstandards und Bestimmung von Bausteinklassifikationen über den Bausteininventar hinweg (vgl. Abschnitt 3.1). MARTIN & ISHII (2002, S. 214) verstehen unter einer Produktfamilienarchitektur: *“A family architecture implies that the different products have a common arrangement of elements, common mapping between function and structure, and common interactions among components”*.

JIAO & TSENG (2000a, S. 471ff) beschreiben die Produktplattform als gemeinsame Basis einer Produktfamilie (vgl. auch MEYER & LEHNERD 1997). In der Praxis zeigt sich allerdings, dass auch produktfamilienübergreifend Plattformelemente im Produktportfolio eingesetzt werden können. Die VOLKSWAGEN AG<sup>13</sup> liefert hierfür prominente Beispiele mit ihren modularen Querbaukasten, Längsbaukasten und Standardantrieb-Baukasten, die über die Modellreihen und Markenfamilien hinweg eingesetzt werden können.

Die Gestaltungsprinzipien, also Gemeinsamkeiten in Konzepten, Funktionen und Komponenten sowie die Schaffung von Architekturstandards tragen zur Spezifizierung der Produktportfolioarchitektur bei. Zusammenfassend soll in dieser Forschungsarbeit folgende Definition für die Produktportfolioarchitektur zu Grunde gelegt werden:

*Die **Produktportfolioarchitektur** beschreibt die Elemente der Produktstruktur, deren horizontalen wie vertikalen Relationen sowie die zugrundeliegenden Gestaltungsprinzipien.*

Die Architektur des Produktportfolios ist Grundlage für die Entwicklung einer Datenstruktur als Arbeitsgrundlage für das Produktportfoliomanagement. Das Metamodell, das diese Datenstruktur beschreibt, wird in Kapitel 5.2.2 vorgestellt. Die komplexe Struktur des Produktportfolios macht das Suchen von Elementen, die bestimmte Eigenschaften und Bedingungen erfüllen, schwierig. Diese Art der Suche wird hier als Mustererkennung bezeichnet und im nächsten Abschnitt näher erläutert.

### 2.1.5 Mustererkennung

Im Kontext dieser Arbeit werden Elemente und deren Relationen als Netzwerke bzw. **Graphen** aufgefasst. Ein umfangreiches Produktportfolio eines Unternehmens kann demnach als komplexer Graph beschrieben werden und somit als Modell und Arbeitsgrundlage für den Systemarchitekten dienen. Die Mustererkennung soll genutzt werden, um Tätigkeiten des Systemarchitekten zu vereinfachen und zu unterstützen. Innerhalb dieses Graphen können z. B. Substrukturen gesucht werden, die von besonderem Interesse sind oder Anlass dazu geben, Maßnahmen zu ergreifen und Änderungen durchzuführen.

**Muster** können Such- oder Fehlermuster sein. Sie beschreiben Eigenschaften von gesuchten Subgraphen und können neben dem Attributteil eine strukturelle Beschreibung beinhalten (vgl. Beispiele in ROBINSON ET AL. 2013). Sowohl attributbasierte als auch strukturelle Beschreibungen können als **Bedingungen** für die Definition eines Musters genutzt werden. Die Merkmale eines Elements können durch dessen **Attribute** beschrieben und als Auswertungsbedingungen genutzt werden. Somit können in einem Muster Attribute definiert werden, die über mehrere Elemente des Graphen hinweg

---

<sup>13</sup> <http://www.volkswagen.de/de/Volkswagen/InnovationTechnik/mqb.html> (entnommen am 15.11.13)

- gleich (z. B. alle Bauteile, die den Aluminiumwerkstoff AlMg3 verwenden),
- unterschiedlich (Unterscheidungsmerkmale zwischen varianten Baugruppen) oder
- ähnlich in bestimmten Wertebereichen (z. B. alle Produkte in einer Leistungsklasse von 120-150 W) sind.

Zudem kann ein Muster die **strukturelle Einbettung** eines gesuchten Subgraphen beschreiben. Hierbei werden die Relationen im Graph ausgewertet und nach Substrukturen gesucht, die die gesuchte strukturelle Einbettung aufweisen. Beispielsweise können somit bestimmte physische Verknüpfungen eines Bauteils in einem Produkt oder auch relevante Abhängigkeiten, wie Zuordnung eines Bauteils zu einem Bauteilverantwortlichen oder zu einer Produktionsressource gesucht werden. Beide Aspekte – die Attribut- und die Strukturauswertung – können in der Definition eines Musters beliebig kombiniert werden. Nachfolgende Abbildung zeigt ein Beispiel für ein Muster und entsprechende Treffer in einem Graphen. In dem Muster werden sowohl Attribute (hier die Knotenform) als auch strukturelle Einbettung (hier Dreiecke, die auf Vierecke zeigen, die wiederum auf Kreise zeigen) verwendet.

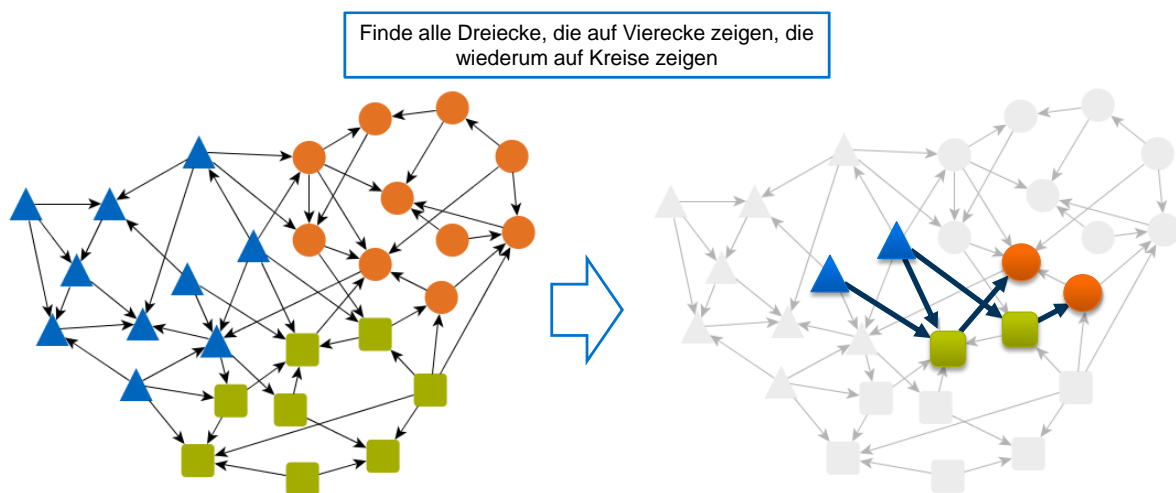


Abbildung 2-5: Beispiel einer Mustererkennung in einem Graphen

Im Englischen wird im Zusammenhang mit Data Mining der Begriff *Pattern Recognition* verwendet, was so viel wie ‚Muster entdecken‘ bedeutet. Hierfür werden statistische Methoden eingesetzt, um wiederkehrende Muster, Ähnlichkeiten und Gemeinsamkeiten in großen Datenmengen zu erkennen. Im Produktportfoliomanagement soll jedoch gezielt nach definierten Mustern gesucht werden. Im Kontext dieser Arbeit wäre daher der Begriff *Pattern Matching*, also ‚Muster finden oder erkennen‘ passender. In dieser Arbeit wird die Graphtransformation eingesetzt. Muster werden auf der Linken Seite einer Regel beschrieben. Auf der Rechten Seite wird eine Aktion definiert, die mit den gefundenen Mustern durchgeführt werden soll. Alle Instanzen des Musters im Graphen können somit aufwandsarm gefunden und bearbeitet werden. Eine ausführliche Darstellung dazu folgt in Kapitel 5.

Nachdem die Begriffe System, Komplexität, Systems Engineering Architektur und Mustererkennung geklärt wurden, werden im nächsten Kapitel die Aufgaben und Ziele des Systemarchitekten diskutiert.

## 2.2 Systemarchitektur als Rolle in der Produktentwicklung

Um zu dem Thema der *Mustererkennung in komplexen Produktportfolios* Zugang zu bekommen, werden im Folgenden potentielle Nutzer des hier entwickelten Ansatzes analysiert. Zudem soll erörtert werden, welche Rolle der Nutzer im Wertschöpfungsprozess einnimmt, welche Aufgaben er erfüllt und welche Ziele er im Unternehmen verfolgt.

### 2.2.1 Aufgaben des Systemarchitekten im Systems Engineering

Der hier vorgestellte Ansatz soll den **Systemarchitekten** in seinem Handeln und Entscheiden im Umgang mit komplexen Produktportfolios unterstützen. Trotz der Popularität des V-Modells und der steigenden Bedeutung des Systems Engineering in der Industrie ist die Rolle des Systemarchitekten noch nicht in der Breite produzierender Unternehmen etabliert. Die Aufgaben werden oft kommissarisch von Projektleitern, Konstrukteuren, Stücklistenverantwortlichen, dem Produktdatenmanagement oder anderen übernommen (HABERFELLNER ET AL. 2012, S. 192f). Im Rahmen dieser Arbeit soll allerdings nicht im Detail auf Spezialitäten in der Organisation einzelner Unternehmen eingegangen werden. Es wird angenommen, dass ein Systemarchitekt als Rolle im Produktentwicklungsprozess in Form einer handelnden Person oder Abteilung tätig ist. Daher wird im weiteren Verlauf der Arbeit der Einfachheit halber von dem „Systemarchitekten“ gesprochen, eingeschlossen seien alle in dieser Rolle tätigen weiblichen wie männlichen Personen.

Als Teil des Wertschöpfungsprozesses trägt der Systemarchitekt zur Erreichung der Unternehmensziele bei. Durch seine besondere, zentrale Stellung ist er prädestiniert dafür, den Überblick über die **marktseitigen Anforderungen** zu haben und mit den **technischen Lösungen** im Unternehmen **abzugleichen** (HABERFELLNER ET AL. 2012, S. 193). Dies bedingt eine weitsichtige Planung, Steuerung und Kontrolle der Lösungsbausteine des Produktportfolios.

Auch wenn in manchen Unternehmen die Rolle des **Produktportfolio Managers** eher marktorientiert ausgelegt ist, so sollte der Systemarchitekt doch entscheidenden Einfluss oder zumindest eine wichtige beratende Funktion für die Festlegung der technischen Gestaltungsprinzipien einnehmen. Mit seiner Rolle als Schnittstelle zwischen Marktanforderungen und technischer Realisierung kann er die richtige Balance aus interner und externer Komplexität herstellen.

Der Systemarchitekt trägt die Verantwortung dafür, das Produktportfolio im Sinne der Unternehmensziele effizienter zu gestalten. Seine **Rolle und Aufgaben im Produktentstehungsprozess** lassen sich am V-Modell aufzeigen (vgl. Abbildung 2-6). Die Aufgaben sind in den Systems Engineering Handbüchern der NASA (2007) und der INCOSE (HASKINS ET AL. 2010) abstrakt definiert.

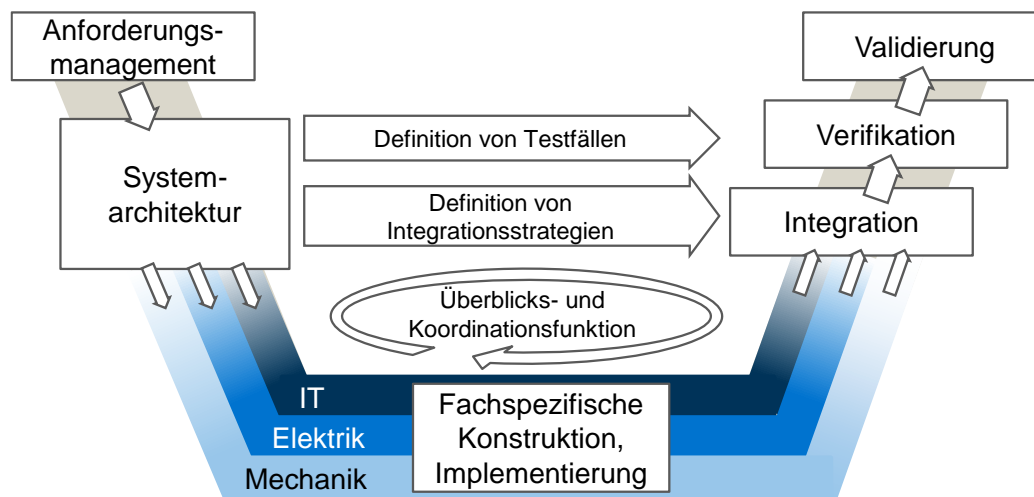


Abbildung 2-6: Rolle und die Aufgaben des Architekten im Produktentstehungsprozess

Kernaufgabe ist es, die Anforderungen an ein Produkt in Entwicklungsaufträge für die Fachabteilungen zu übersetzen. Die **Systemdekomposition** erfolgt durch die Unterteilung des Gesamtsystems in Subsysteme durch Modularisierung, Schnittstellendefinition und Strukturierung der Produkte eines Unternehmens. Hierbei können

- produktstrategische Aspekte (ERIXON 1998; BLEES 2011),
- technisch-funktionale Kriterien (GÖPFERT 1998; ULRICH 1995),
- oder auch organisationsspezifische Rahmenbedingungen (COLFER & BALDWIN 2010; SOSA ET AL. 2004; SANCHEZ & MAHONEY 1996)

Einfluss auf die Dekomposition haben. Im Handbuch der NASA (2007, S. 49) heißt es: “*System architecture activities drive the partitioning of system elements and requirements to lower level functions and requirements to the point that design work can be accomplished. Interfaces and relationships between partitioned subsystems and elements are defined as well*”. Neben der Systemdekomposition müssen auch **Integrationsstrategien und Testfälle** entwickelt werden, um die entwickelten Subsysteme zu einem funktionierenden Gesamtprodukt, das den Kundenanforderungen entspricht, zusammenzusetzen und zu testen.

Innerhalb der Produktentwicklung sollte der Systemarchitekt eine **Überblicks- und Koordinationsfunktion** einnehmen. Entwicklungsumfänge sollte er im Gesamtkontext des Produktportfolios und der übergeordneten Unternehmensziele bewerten.

Der Systemarchitekt muss die nötigen **Gestaltungsprinzipien und Architekturstandards** vorgeben und einfordern, um die Kompatibilität der verteilt entwickelten Teilsysteme sicherstellen zu können (BALDWIN & CLARK 2000). CRAWLEY ET AL. (2004, S. 9) fassen dies zusammen mit: “*Architecture as ‘rules to follow when creating a system’ conveys coordination rules, so that different people at different times and places can create systems that are compatible in various ways. This is efficient not only because of the advantages of coordination but also because elements and interconnection patterns with known behavior can be reused, increasing the speed with which such systems can be designed and put into operation.*”

## 2.2.2 Unternehmensziele und Beitrag des Systemarchitekten

Welche Ziele verfolgt der Systemarchitekt bei Erfüllung seiner Aufgaben und wie trägt er damit zu den Unternehmenszielen bei? An dieser Stelle soll zwischen **Fundamental-** und **Instrumentalzielen** unterschieden werden. „Ein Fundamentalziel ist ein Ziel, das um seiner selbst willen verfolgt wird und für den Entscheider keiner Begründung mehr bedarf“ (EISENFÜHR ET AL. 2010, S. 63). Ein Instrumentalziel wird verfolgt, um ein Fundamentalziel zu erreichen. Die Unterscheidung der beiden Zielarten ist relativ und abhängig vom Kontext (EISENFÜHR ET AL. 2010, S. 65). So können Fundamentalziele des Systemarchitekten Instrumentalziele für die Fundamentalziele des Gesamtunternehmens sein (EBD.).

Die **Fundamentalziele des Unternehmens** lassen sich am klassischen **Zieldreieck** Zeit-Kosten-Qualität erläutern. Mit begrenzten Ressourcen soll das Unternehmen effizient **kostengünstige Produkte** mit **nachfragegerechter Qualität** entwickeln und somit die Nachfrage der Märkte **zeitgerecht** bedienen. Dadurch entsteht ein klassischer Zielkonflikt aus Produktoptimierung, begrenzten Ressourcen und Zeitdruck (LINDEMANN 2007, S. 14).

Um an den Unternehmenszielen anzuknüpfen, ergeben sich Fundamentalziele im Kontext des Systemarchitekten:

- Schnelles Reaktionsvermögen bei Marktänderungen,
- Differenzierung vom Wettbewerb,
- Flexibilität in der Produktarchitektur,
- effektiver und effizienter Ressourceneinsatz und
- das Antizipieren von Lebenszyklusaspekten.

Nachfolgende Abbildung fasst die Zusammenhänge zwischen den Unternehmenszielen und den Fundamentalzielen des Systemarchitekten zusammen (vgl. Abbildung 2-7).

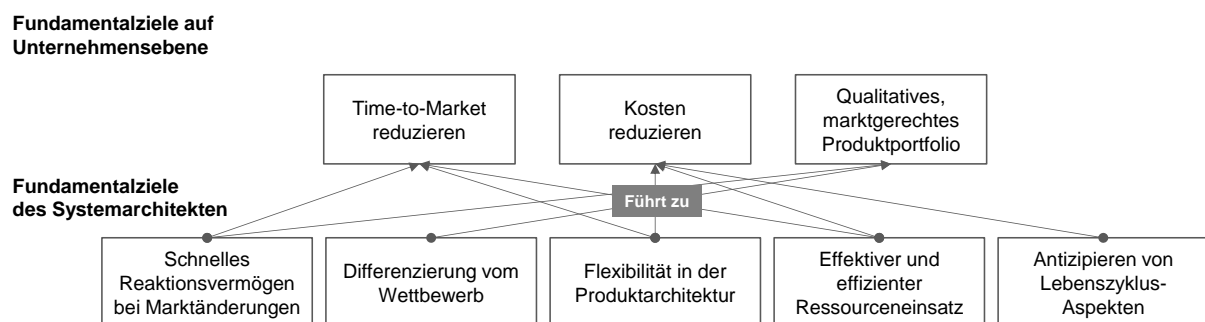


Abbildung 2-7: Fundamentalziele auf Unternehmensebene und Systemarchitekturebene

Die Kunst besteht darin, die Balance entgegenwirkender Ziele im Produktportfolio herzustellen (vgl. SIMPSON 2004, S. 8), um diese Fundamentalziele unterstützen zu können. Nach HABERFELLNER ET AL. (2012, S. 188ff) sind Anzeichen einer guten Architektur, dass sie:

- wettbewerbsfähige Produkte ermöglicht,
- strategische Unternehmensziele unterstützt,
- Gesetze und Regularien berücksichtigt,
- effizient betreibbar, wartbar und nachhaltig ist,
- mit wenig Aufwand erweiterbar und anpassbar ist,

- die Weiterentwicklung von Produkten im Rahmen möglicher Zeit und Ressourcen ermöglicht und
- „elegant“ ist.

Mit Innovationen, neuen Produkten und Varianten soll einerseits eine **Differenzierung vom Wettbewerb** erreicht werden, während andererseits **bei Marktveränderungen schnell reagiert** werden soll (FRICKE & SCHULZ 2005; ROSS ET AL. 2008). Letzteres Ziel spricht dafür, vorhandene Lösungen aufwandsarm zu neuen Produkten konfigurieren zu können. Es soll mit möglichst geringem Aufwand und kleiner interner Vielfalt ein möglichst breiter Markt mit großer externer Vielfalt flexibel bedient werden können (KRAUSE ET AL. 2013, S. 543ff). Beide Ziele müssen gegeneinander abgewogen werden, um ein qualitäts- und marktgerechtes Produktportfolio anbieten zu können.

Um zudem nachhaltig Kosten im Unternehmen reduzieren zu können, ist ein **effektiver wie effizienter Ressourceneinsatz** ein wichtiges Ziel. Der Einsatz von Unternehmensressourcen für die Entwicklung der Produkte sollte sich stets mit dem Kundennutzen und der Nachfrage am Markt rechtfertigen lassen (HABERFELLNER ET AL. 2012, S. 187f). Daher müssen die begrenzten Ressourcen auf die richtig priorisierten Projekte zugeteilt werden. Nur so kann unnötiger Zeitaufschub durch Fehlentwicklungen vermieden und die richtigen Produkte zeitgerecht am Markt platziert werden. Zudem sind Synergien im Unternehmen herzustellen mit dem Ziel, Skaleneffekte der Massenherstellung oder durch Erhöhung und Bündelung von Einkaufsvolumina zu ermöglichen (SCHUH 2005).

Maßgeblich zur Kostenreduktion und der Verkürzung der Zeit bis zum Markteintritt neuer Produkte trägt auch die **Flexibilität und Adaptierbarkeit in der Produktarchitektur** bei (SANCHEZ & MAHONEY 1996, S. 66). Sie erlaubt kürzere Produktentwicklungszeiten, vereinfacht Produktion und Logistik und lässt späte Anpassungen durch den Vertrieb oder den Kunden selbst zu (BROWNING & HONOUR 2008; ENGEL & BROWNING 2008; KISSEL ET AL. 2012).

Zunehmend gewinnen Themen der **Produkt-Lebenszyklus-Optimierung** bereits auf Architekturebene an Bedeutung (DE WECK ET AL. 2011, S. 65ff, FITCH & COOPER 2005b, 2005a). Unterschiedliche Design-for-X-Anforderungen – wobei das X für einen bestimmten Fokus wie z. B. Zusammenbau, Anpassbarkeit, Wartbarkeit oder Recycling steht – müssen in der Architekturgestaltung antizipiert und verschiedene, teils konkurrierende Stakeholder-Interessen ausgeglichen werden. Durch Antizipieren jener Lebenszyklus-Aspekte kann gleichermaßen ein Mehrwert für den Kunden erzeugt, wie auch Kosten reduziert werden.

### 2.2.3 Instrumentalziele des Systemarchitekten

Der Systemarchitekt sollte zur Erreichung der oben beschriebenen Fundamentalziele eine Reihe von **Instrumentalzielen** verfolgen:

- Hohe Flexibilität in der Produktarchitektur,
- ein konsistentes und klar strukturiertes Produktportfolio,
- die Wahl eines effizienten Grads der Wiederverwendung,
- sinnvolle Standardisierung und



- Pflege einer marktgerechten Vielfalt.

Abbildung 2-8 gibt einen Überblick über den Zusammenhang zwischen Fundamental- und Instrumentalzielen des Systemarchitekten.

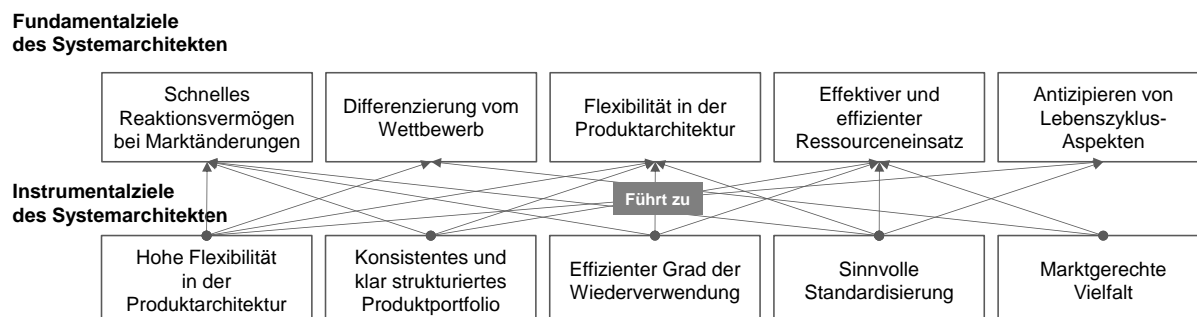


Abbildung 2-8: Zielestruktur in der Systemarchitektur

Der Wunsch, mit einer **hohen Flexibilität in der Produktarchitektur** schnell auf Änderungen am Markt reagieren zu können, erfordert es, bestehende Teillösungen in einem Produktportfolio aufwandsarm zu neuen Produkten konfigurieren zu können (KOHLHASE 1997, S. 54f). Ziel ist es, die Kosten einer Produktausleitung reduzieren zu können. Entscheidend hierfür ist wie funktional unabhängig die Bausteine in einem Produktportfolio entwickelt (GÖPFERT 1998) und deren Schnittstellen definiert werden, so dass sie kompatibel zueinander zu neuen Produkten konfiguriert werden können. Dies führt dazu, dass höhere Flexibilität in der Produktstruktur erreicht, aufwandsärmer Lebenszyklusaspekte berücksichtigt und eine Differenzierung vom Wettbewerb bewerkstelligt werden können.

Hierzu trägt auch ein **konsistentes und klar strukturiertes Produktportfolio** bei. Der Aufbau und die Gestaltungsprinzipien innerhalb eines Produktportfolios sollten einfach und klar im Unternehmen kommuniziert werden können (vgl. BALDWIN & CLARK 2000). Auch in der Absprache mit Lieferanten sollten klare Vorgaben gemacht werden können, um zugelieferte Subsysteme wirtschaftlich integrieren zu können. Es gibt zahlreiche Modelle, Produktportfolios zu strukturieren und handhabbarer zu machen (vgl. Kapitel 3). Baukastensystematiken, Plattformansätze, Bausteinklassifikationen, Festlegung von Schnittstellen, bestimmter Geometrie und weiterer Architekturstandards helfen das Produktportfolio klarer zu strukturieren. Strukturstücklisten, Merkmalsbäume und Ähnliches unterstützen zusätzlich das Verständnis über den Aufbau von Produkten, Varianten und des Produktportfolios. Sie sind Planungs- und Kommunikationswerkzeuge für den Systemarchitekten und vermitteln den Ist-Zustand des Produktportfolios.

Der Systemarchitekt muss sicherstellen, dass bestehende und neu entstehende Vielfalt im Portfolio durch die Nachfrage am Markt gerechtfertigt ist (HABERFELLNER ET AL. 2012, S. 188). Entsprechend gegensteuern muss er, wenn **unnötig oder aus Fehlern heraus Variantenvielfalt erzeugt** wird (vgl. Kapitel 3.3.1). Über die Zeit kann sich die Nachfrage von Varianten zurückbilden. Dies muss der Systemarchitekt erkennen und entsprechende Produkte abkündigen bzw. eine Abkündigung empfehlen.

Wirtschaftlich betrachtet steht einer **marktgerechten Variantenvielfalt** eine **effiziente Wiederverwendung** und **sinnvolle Standardisierung von Bausteinen** gegenüber. Hier ist für

ein ausgewogenes Nutzen-Aufwand-Verhältnis zwischen Individualisierung und Standardisierung zu sorgen (GRAESSLER, S. 90). Ressourcenschonender wirken sich die Verwendung bestehender Lösungen und Standards aus. Sie ermöglichen auch ein schnelleres Reaktionsvermögen und eine bessere Nutzung des Lösungsraumes im Rahmen der definierten Möglichkeiten. Dies sollte sich aber nicht innovationshemmend auswirken und somit nicht die Differenzierung im Wettbewerb langfristig beeinträchtigen. Durch Wiederverwendung und somit durch Erhöhung der Stückzahlen von Komponenten kann zudem die Erzielung von Skaleneffekten in Produktion und Einkauf erzielt werden (SCHUH 2005). Auf der anderen Seite können gegebenenfalls durch Individualisierung der Produkte die Marktanteile vergrößert und somit der Profit erhöht werden (KRAUSE ET AL. 2013, S. 543). Nach RENNER (2007, S. 47) bietet sich sinngemäß die Produktportfolioarchitektur als Bindeglied zwischen der nicht direkt zusammenhängenden internen und externen Variantenvielfalt oder Komplexität an.

Um die hier aufgeführten Instrumentalziele erreichen zu können, müssen einige Voraussetzungen erfüllt sein.

#### 2.2.4 Voraussetzungen für eine erfolgreiche Systemarchitektur

In den vorausgegangenen Abschnitten wurden die Aufgaben und Ziele des Systemarchitekten beschrieben. Der Erreichung der Ziele stehen eine Vielzahl von Schwierigkeiten entgegen, die es zu berücksichtigen gilt. In den Handbüchern und Richtlinien werden Annahmen und Vereinfachungen getroffen, die sich oft so nicht in der Realität widerspiegeln. Beispielsweise wird meist von der Entwicklung *eines* Produktes ausgegangen. In der Realität müssen in zahlreichen Entwicklungsprojekten Neuentwicklungen, Änderungen und Produktpflegemaßnahmen aufeinander abgestimmt werden, um bestmögliche Synergien für das Unternehmen erreichen zu können.

Außerdem werden im Rahmen von Entwicklungsprojekten Fehler gemacht, Rahmenbedingungen verändern sich und damit die Prämissen, die zu bestimmten Entscheidungen beim Aufbau des Produktportfolios geführt haben. In der Folge müssen kontinuierlich Fehler gesucht und behoben, Änderungen und deren Auswirkungen eingepflegt und somit ein hoher Aufwand erbracht werden, die Aktualität und Konsistenz des Portfolios zu gewährleisten.

Darüber hinaus ist es Aufgabe des Systemarchitekten, den verschiedenen Stakeholdern im Prozess von der Konstruktion, über Produktion bis hin zum Vertrieb Überblick und Transparenz über das Produktportfolio, dessen Strukturierung und Bausteine zu vermitteln. Wenn Anpassungen an den Bausteinen des Portfolios vorgenommen werden – sei es durch Neuentwicklung, Anpassungskonstruktionen oder auch Abkündigung durch einen Lieferanten – muss er Transparenz über die Reichweite von Änderungen schaffen können.

Für diese Aufgaben ist Voraussetzung, dass der Systemarchitekt:

- sich jederzeit Überblick über den Ist-Zustand des gesamten Produktportfolios machen,
- aufwandsarm analysieren und beurteilen und
- Fehler und Handlungsfelder erkennen kann.

Zudem muss er über die Mittel und Werkzeuge verfügen,

- Ordnung im Produktportfolio zu halten,
- klare und konsistente Strukturen zu pflegen und
- ungewollte Vielfalt zu vermeiden.

Nicht zuletzt werden effiziente Kommunikationswerkzeuge benötigt,

- die komplexen Zusammenhänge adressatengerecht gegenüber den Stakeholdern vermitteln und
- Managemententscheidungen transparent und rational vorbereiten zu können.

Ein konsistentes und klar strukturiertes Produktportfolio ist die Grundlage dafür, über einen effizienten Grad der Wiederverwendung oder über ein sinnvolles Ausmaß der Standardisierung urteilen zu können. Mit dem Überblick über unterschiedliche Bausteinarten, deren Beziehungen untereinander, zugrundeliegenden Gestaltungsregeln und wirtschaftlicher Eckdaten können geeignete Bewertungssysteme eingesetzt und auf die Unternehmensziele ausgerichtet werden. In gewissem Maße ist eine klare Architektur des Produktportfolios auch Voraussetzung dafür, geeignete Bausteine änderungsgerecht auszugestalten und somit höhere Flexibilität mit der Produktarchitektur zu erreichen.

Um allerdings zu einem konsistenten und klar strukturierten Produktportfolio zu gelangen, müssen einige Voraussetzungen im Unternehmen erfüllt sein:

- Etablierung der Rolle des Systemarchitekten und Ausstattung mit angemessenen Entscheidungskompetenzen.
- Entwicklung eines umfassenden Modells zur Beschreibung des Produktportfolios, das die Belange des Systemarchitekten und seiner Stakeholder berücksichtigt und als Planungsgrundlage dient.
- Einrichtung einer zentralen Datenbasis oder eines Werkzeuges, das die Zusammenführung verteilter Daten erlaubt, mit allen planungsrelevanten Informationen.
- Implementierung von Prozessen und Methoden für eine kontinuierliche Pflege und marktgerechte Weiterentwicklung des Produktportfolios.
- Einsatz geeigneter Rechnerwerkzeuge, die einen effektiven und effizienten Umgang mit den komplexen Strukturen des Produktportfolios erlauben.

Die Zielstellung dieser Arbeit ist darauf ausgerichtet, einen Beitrag dazu zu leisten, diese Voraussetzungen zu erfüllen und somit den Systemarchitekten in seiner Arbeit zu unterstützen. Dazu dient die Entwicklung eines Vorgehens zur Identifizierung von Schwachstellen in komplexen Produktstrukturen und Aufzeigen von kritischen Handlungsfeldern. Die Methodik soll die Reproduzierbarkeit, Objektivierung und Dokumentation der Entscheidungsunterstützung vereinfachen und somit die Effektivität erhöhen. Zudem soll die Effizienz verbessert werden, indem verteilte Datenquellen zusammengeführt werden und somit eine schnellere Zugänglichkeit der relevanten Informationen erreicht werden kann. Durch Automatisierung oft verwendeter Analyseverfahren soll auch unter Zeitdruck eine Ad-hoc-Verfügbarkeit dringender Analyseergebnisse möglich sein.

Den oben genannten Zielen stehen zahlreiche Herausforderungen im Wege, die es zu überwinden gilt. Angesichts des Zeit- und Kostendrucks und der wachsenden Komplexität ist es oft schwierig, die Grundvoraussetzungen zu erfüllen.

Im nächsten Kapitel soll daher untersucht werden, welche Hürden sich durch wachsende Komplexität bilden. Hierfür soll zunächst der prinzipielle Aufbau eines Produktportfolios geklärt werden. Anschließend wird untersucht, durch welche Faktoren die Komplexität zunimmt und welche Mittel und Methoden es gibt, um diese einzudämmen oder zu beherrschen.

## 3. Strukturierung komplexer Produktportfolios

Um den Aufbau von Produkten und Produktportfolios im Unternehmen planen, organisieren und kommunizieren zu können, gibt es zahlreiche Strukturierungs-, Klassifizierungs- und Standardisierungsmöglichkeiten. In diesem Kapitel wird ein Überblick über typische Strukturierungsformen gegeben. Anschließend werden Ansätze im Stand der Technik analysiert, inwiefern sie zur Zielstellung dieser Arbeit beitragen können. Daraus leiten sich die Herausforderungen und Hürden im Umgang mit komplexen Produktportfolios ab. Abschließend werden Anforderungen an den Lösungsansatz und der angestrebte Beitrag der Arbeit formuliert.

### 3.1 Grundbegriffe des Produktportfolios

#### 3.1.1 Produktgliederung

Die hierarchische Gliederung aller gebrauchts- und verkehrsfähigen Erzeugnisse eines Unternehmens wird hier als Produktportfolio aufgefasst. Synonym kann der Begriff Produktprogramm verwendet werden (RENNER 2007, S. 12). Je nach Unternehmen können unterschiedlich zahlreiche Gliederungsstufen Verwendung finden. Die Bestandteile der Produktgliederung definieren sich wie folgt:

#### **Einzelteil und Komponente**

Elementarteil eines Produktportfolios sind die **Einzelteile** DIN 199-1 (S. 8ff). Sie können nicht zerstörungsfrei zerlegt werden. Hier werden sie synonym mit dem Begriff Komponente verwendet. **Komponenten** können materielle Artefakte sein. Für den Aufbau mechatronischer Produkte oder Produkt-Service-Systeme muss es auch immaterielle Artefakte als Komponenten geben, also Software- und Servicekomponenten.

#### **Gruppe**

Zwei oder mehrere Teile oder auch Gruppen niederer Ordnung können eine **Gruppe** bilden. „Eine Gruppe kann in sich geschlossen sein (montiert) oder als ‚Gruppe loser Teile‘ bestehen“ DIN 199-1 (S. 12). Wenn der Zweck der Gruppierung die Bestimmung eines Montageumfanges ist, wird von einer **Baugruppe** gesprochen. Mit der Gruppierung können auch andere Zwecke erfüllt werden, wie z. B. der Lieferumfang eines Zulieferers in einer Gruppe zusammengefasst werden kann.

#### **Modul**

Modularisierung eines Systems meint die Unterteilung in **Module** nach z.B. technisch-funktionalen Kriterien (ULRICH 1995) oder nach Gesichtspunkten der Produktlebensphasen wie Entwicklung, Beschaffung, Herstellung, Vertrieb, Nutzung und Entsorgung (GU & SOSALE 1999; BLEES 2011; KRAUSE ET AL. 2014). Module sind abgrenzbare Einheiten, deren Relationen zwischen den Elementen innerhalb des Moduls stark ausgeprägt sind, aber nur schwach zu

anderen Elementen oder Modulen außerhalb der Modulgrenze (SIMON 1962, S. 473f, BALDWIN & CLARK 2000, S. 63). Eine modulare Architektur liegt nach ULRICH (1995, S. 422) dann vor, wenn eine Eins-zu-Eins-Verknüpfung zwischen Funktionselementen der Funktionsstruktur mit Komponenten der physikalischen Struktur gegeben ist und entkoppelte Schnittstellen zwischen den Modulen definiert sind.

PINE, II. (1993, S. 201) unterscheidet in Anlehnung an ABERNATHY & UTTERBACK (1978) sechs Arten der Modularität für die Entwicklung kundenindividueller Massenprodukte (vgl. Abbildung 3-1):

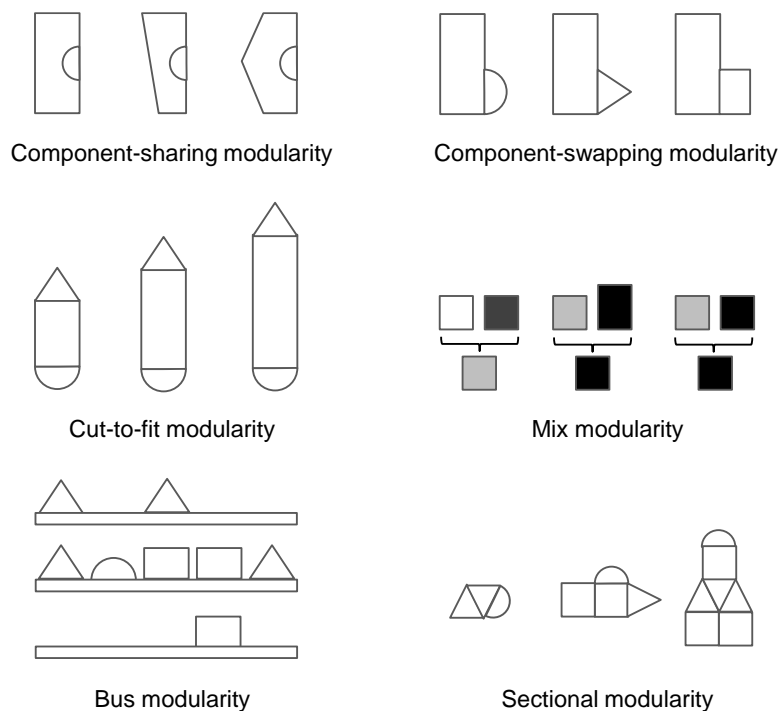


Abbildung 3-1: Sechs Arten der Modularität nach PINE, II. (1993, S. 201). Übersetzung nach SCHUH (2005, S. 131), v. l. n. r.: „Modularität durch Gemeinsamkeit von Bestandteilen, Modularität durch Austausch von Bestandteilen, Modularität durch passenden Zuschnitt, Misch-Modularität, Bus-Modularität und Teil-Modularität.“

Der Begriff Modularität ist eng mit dem Thema Produktarchitektur verknüpft (GERSHENSON ET AL. 2003, S. 301). Eine begriffliche Trennung von Baukastensystemen und modularen Systemen ist überwiegend im deutschsprachigen Raum vertreten. In der englischsprachigen Literatur werden „modular products“ bzw. „modular systems“ für beide Bedeutungen verwendet. Einen umfangreichen Literaturüberblick zu Definitionen und Methoden zur Modularität geben (GERSHENSON ET AL. 2003) und (GERSHENSON ET AL. 2004).

## Produkte und Erzeugnisse

Ein durch Produktion entstandener Gegenstand, der gebrauchsfähig und verkehrsfähig ist, wird als Erzeugnis bezeichnet DIN 199-1 (S. 11). GEMBRY (1998, S. 7) versteht synonym unter **Produkten** „materielle Angebotsbestandteile eines produzierenden Unternehmens“. Materielle Produkte können auch in Verknüpfung mit Serviceleistungen auf den Markt gebracht werden. Zunehmend ist auch ein Wandel am Markt erkennbar, in dem nicht mehr ein Kopierer, sondern

die Kopie, nicht mehr die Turbine, sondern „power by the hour“, oder nicht mehr ein Kraftfahrzeug, sondern Mobilität verkauft werden soll. Dies hat nachhaltigen Einfluss auf die Architektur eines Erzeugnisses sowie die Produktentwicklung selbst (VASANTHA ET AL. 2012).

## Varianten

„Varianten sind Gegenstände ähnlicher Form und/oder Funktion mit einem in der Regel hohen Anteil identischer Gruppen oder Teile“ (DIN 199-1, S. 15). Varianten können nach unterschiedlichen Merkmalen klassifiziert werden. Je nach Produktebene, Auftrittshäufigkeit, technische oder strukturelle Kriterien, Variantenfestlegung, Variantenstruktur, subjektive Wahrnehmung teilt HEINA (1999, S. 6) unterschiedliche Arten von Varianten ein (in Anlehnung an LINGNAU 1994, S. 24ff). Nachfolgende Tabelle 3-1 gibt einen Überblick:

Merkmale	Variantenausprägungen
Produktebene	<ul style="list-style-type: none"> <li>• Produktvariante</li> <li>• Baugruppenvariante</li> <li>• Teilevariante</li> </ul>
Auftrittshäufigkeit	<ul style="list-style-type: none"> <li>• Standardvariante</li> <li>• Sonderausführung</li> <li>• Kundenauftragsspezifische Variante</li> </ul>
Technische Kriterien	<ul style="list-style-type: none"> <li>• Geometrische Variante (Form- und Maßvariante)</li> <li>• Materialvariante</li> <li>• Technologievariante</li> <li>• Lagevariante</li> <li>• Produktions- oder Prozessvariante</li> </ul>
Variantenfestlegung	<ul style="list-style-type: none"> <li>• Herstellerspezifische Variante</li> <li>• Kundenspezifische Variante</li> </ul>
Variantenstruktur	<ul style="list-style-type: none"> <li>• Einfache Variante</li> <li>• Gemischt komplexe Variante</li> <li>• Komplexe Variante</li> </ul>
Strukturelle Kriterien	<ul style="list-style-type: none"> <li>• Kann-Variante (alternativ)</li> <li>• Muss-Variante (additiv)</li> </ul>
Subjektive Wahrnehmung	<ul style="list-style-type: none"> <li>• Periphere Variante</li> <li>• Fundamentale Variante</li> </ul>

Tabelle 3-1: Klassifizierung des Variantenbegriffs (HEINA 1999, S. 6)

Die Anzahl unterschiedlicher Varianten eines Produktes wird auch als Programmtiefe verstanden. Nach SCHUH (2005, S. 122) ist „Vielfalt niemals Selbstzweck, sondern die Reaktion

des Unternehmens auf die marktseitigen Anforderungen“. Eine Vielfaltsstrategie kann daher nicht für sich alleine ohne Bezugnahme auf die Unternehmensstrategie betrachtet werden.

## Produktfamilien

Eine Produktfamilie ist eine Auswahl von ähnlichen Produkten, die auf einer gemeinsamen Plattform entwickelt wurden und jeweils besondere Funktionalität aufweisen, um spezielle Kundenanforderungen zu erfüllen (MEYER & LEHNERD 1997; HARLOU 2006). Während eine Produktfamilie auf ein bestimmtes Marktsegment abzielt, adressieren die einzelnen Varianten der Produktfamilie nochmals einzelne Teilsegmente (JIAO ET AL. 2006, S. 2). In einem Produktportfolio kann mit der Aufteilung in Produktfamilien eine Hierarchiestufe gebildet werden und somit das Produktspektrum eines Unternehmens gegliedert werden.

Komponenten, Gruppen, Module, Produkte, Varianten und Produktfamilien können in unterschiedlichsten Formen Strukturen bilden. Dies kann geplant oder ungeplant erfolgen. Wenn sich durch Neuentwicklungen oder mangelnde Pflege ungünstige Abhängigkeiten auf horizontaler oder vertikaler Ebene unkontrolliert ausbilden, entsteht Wildwuchs. In den nachfolgenden Kapiteln wird dargelegt, wie Produkte (3.1.2) und komplexe Produktportfolios (3.1.3) typischerweise strukturiert werden können.

### 3.1.2 Strukturen in Produkten

#### Produktstruktur

Der Begriff „**Produktstruktur**“ wird in der Literatur uneinheitlich verwendet (RAPP 2010, S. 7). RAPP (2010) verwendet den Begriff synonym mit **Produktarchitektur**. JESCHKE (1997, S. 23) hingegen definiert die Produktstruktur als ein „*produktdarstellendes Modell, das die Gesamtheit der nach bestimmten Gesichtspunkten (z.B. Fertigung, Montage, Funktion, Disposition, Kalkulation) festgelegten Beziehungen zwischen Baugruppen und Einzelteilen eines Produktes beschreibt*“. Produktstrukturen können nach Zusammenbau-, Funktions-, oder Dispositionsgesichtspunkten erstellt werden.

Die Struktur eines Systems ist ein wichtiger Aspekt der **Systemarchitektur**. Die Architektur beschreibt allerdings neben der Struktur der verschiedenen Elemente und Relationsarten auch noch wie ein System aufgebaut werden soll und definiert die zugrundeliegenden Gestaltungsprinzipien und Architekturstandards.

PONN & LINDEMANN (2011, S. 65ff) verwenden u.a. die **funktionale Struktur**, die ein Produkt in seine Haupt- und Unterfunktionen hierarchisch gliedert. In der **Beziehungsstruktur** wird erfasst, welche Komponenten mit welchen verbaut sind oder welche Flussbeziehungen bestehen (PIMMLER & EPPINGER 1994). In der **Montagestruktur** wird das Produkt in seine Baugruppen hierarchisch untergliedert (CAESAR 1991, S. 48ff). Die Beziehungsstruktur sollte aus der Funktionsstruktur und die Montagestruktur wiederum aus der Beziehungsstruktur abgeleitet werden (DAHL 1990, S. 56ff).



Eine klare Produktstruktur hat positive Wirkung auf die Kosten im Unternehmen, da sie hilft Komplexität zu reduzieren. RAPP (2010, S. 105ff) beschreibt neben der Wirtschaftlichkeit weitere positive Effekte einer klaren Produktstrukturierung:

- **Wartbarkeit der Produkte:** Durch klare Definition von Schnittstellen und Funktionsbaugruppen, können Stillstandzeiten reduziert werden.
- **Rekonfigurierbarkeit:** Ähnlich wie bei der Wartbarkeit können durch Standardisierung und einfache Lösbarkeit von Schnittstellen Komponenten neuer Funktionalität in ein Produkt nachträglich eingebracht werden
- **Robustheit gegenüber Veränderungen:** Komponenten, die eine hohe Änderungswahrscheinlichkeit haben, können so in die Produktstruktur eingepflegt werden, so dass sie bei Änderungen kostengünstig ersetzt werden können.

Die Produktstruktur ist **zentrales Planungs- und Entwicklungsinstrument** und somit ein wichtiger Teil des Gesamtunternehmens (RAPP 2010, S. 93ff). Durch eine effektive Strukturierung der Produkte können die Voraussetzungen geschaffen werden, die Unternehmensstrategie umsetzen zu können. Darüber hinaus hat die Produktstruktur großen Einfluss auf die Kostenstruktur, die Organisation, die Innovationsfähigkeit und Flexibilität eines Unternehmens.

Gewissermaßen ist die Produktstruktur auch abhängig von den Rahmenbedingungen eines Unternehmens wie dem Produktionssystem (SCHUH 2005, S. 122). Ein Unternehmen mit einem hochflexiblen Produktionssystem, das in kleinen Stückzahlen individuelle Produkte herstellen kann, bietet dabei andere Randbedingungen als ein Massenhersteller, der höhere Stückzahlen zu günstigeren variablen Kosten bei niedrigerer Flexibilität darstellen kann.

## Stücklisten

Typisches Planungs- und Arbeitsmittel im Unternehmen sind **Stücklisten**. Sie geben die hierarchische Strukturierung von Produkten aus verschiedenen Sichtweisen wieder, wie z. B. der Entwicklungs-, Produktions- oder Vertriebsicht. Während die Entwicklung eine Funktionsstrukturierung dekomponiert nach Fachabteilungen heranzieht, wird der Vertrieb eine Sichtweise bevorzugen, aus der die Funktionen mit ihren Varianten und Optionen zur Konfiguration von Kundenwünschen hervorgeht. SCHUH (2005, S. 140) gibt einen Überblick über verschiedene Typen von Stücklisten als Hilfsmittel zur Produktstrukturierung in der Praxis (in Anlehnung an EVERSHEIM 1999, S. 7–47):

- Mehrstufige Produktstruktur
- Mengestückliste
- Strukturstückliste
- Baukastenstückliste
- Variantenstücklisten
  - Gleichteile-Stücklisten
  - Plus-Minus-Stücklisten
- Verwendungsnachweis

Die unterschiedlichen Hilfsmittel zur Produktstrukturierung werden bei SCHUH (2005, S. 140) näher erläutert. In der **Mengestückliste** werden alle Bauteile eines Erzeugnisses je einmal

unter Angabe der erforderlichen Stückzahl aufgelistet. Sie eignet sich z. B. zur Bedarfsermittlung von Zukaufteilen.

In der **Strukturstückliste** wird das Produkt in seine Baugruppen untergliedert bis hin zu den verwendeten Einzelteilen. Treten Baugruppen in einem Produkt öfters auf, werden diese in der Strukturstückliste wiederholt angezeigt. Einerseits ermöglicht diese Stücklistenart einen guten Überblick, verliert aber andererseits bei komplexeren Produkten schnell ihre Übersichtlichkeit.

Die **Baukastenstückliste** enthält nur jene „Elemente einer Baugruppe, die sich in der nächsttieferen Strukturebene befinden.“ Wiederverwendete Baugruppen werden mit einem Auflösungsverweis geführt. Für diese Baugruppen wird jeweils eine eigene Stückliste geführt. Auch wenn zahlreiche Einzelstücklisten mit Querverweisen entstehen, ist somit der Aufwand für die Erstellung, Speicherung und Änderung dieser Stücklistenart vermindert.

Varianten eines Produktes unterscheiden sich meist in nur wenigen Details und können in **Variantenstücklisten** zusammengefasst werden. Die **Gleichteilestückliste** unterteilt die Stückliste in Gleichteilegruppen und Variantengruppen. Sollte sich die Gleichteilegruppe ändern, so sind alle Stücklisten betroffen. Die **Plus-Minus-Stückliste** ist nach SCHÖNSLEBEN (2011, S. 383ff) eine Variantenstückliste mit hinzuzufügenden und wegzulassenden Positionen. Je kleiner der Gleichteilegrad eines variantenreichen Produktes ist, desto eher lohnt sich die kostenintensivere Plus-Minus-Stückliste.

„Der **Verwendungsnachweis** enthält alle Baugruppen der übergeordneten Strukturebenen, in denen ein Element enthalten ist, unter Angabe der jeweiligen Menge des betrachteten Elements“ (EVERSHEIM 1999, S. 7–49).

### 3.1.3 Strukturen in Produktportfolios

Ebenso wie Produkte lassen sich auch Produktportfolios strukturieren. Mehrere ähnliche Produkte werden in Produktfamilien zusammengefasst. RENNER (2007, S. 12) beschreibt ein Produktportfolio durch dessen Breite und Tiefe. Er bedient sich hier der Definition von LINGNAU (1994, S. 105ff), der die **Programmbreite** mit der Anzahl unterschiedlicher Produktarten und die **Programmtiefe** mit der Anzahl unterschiedlicher Varianten eines Produktes beschreibt.

In der Literatur finden sich einige Strukturierungsansätze. Als prominente Stellvertreter sollen hier

- Baureihen,
- Plattformen,
- Baukastensysteme und
- typische Klassifikationsmöglichkeiten für Bausteine

eines Produktportfolios aufgezeigt werden.

#### **Baureihe**

Nach GERHARD (1984) wachsen Baustufen einer Baugruppe oder eines Bauteils nach dem **Gesetz der Reihe**. Bei möglichst gleichen Fertigungsverfahren sollen unterschiedliche

Leistungsstufen ermöglicht werden (PAHL ET AL. 2007, S. 629ff). In der Entwicklung und der Qualitätssicherung können somit durch Anwendung von Ähnlichkeitsgesetzen Einsparungen realisiert werden. Nach EHRENSPIEL & MEERKAMM (2013, S. 725ff) sollten Vertreter einer Baureihe in folgenden Aspekten gleich sein:

- Funktion (qualitativ),
- konstruktive Lösung,
- Werkstoffe und
- Fertigung.

Hingegen unterschiedlich sind:

- Leistungsdaten (Funktion quantitativ),
- Abmessungen und davon abhängige Größen (Gewicht, Kosten usw.).

„Normzahlreihen nach DIN 323 sind dezimalgeometrisch gestufte Reihen, d. h. Zahlenreihen, bei denen sich innerhalb einer Dekade jedes Glied durch Multiplikation mit einem konstanten Faktor  $\phi$  aus dem vorherigen ergibt“ (EHRENSPIEL & MEERKAMM 2013, S. 727). Ähnliche Baugruppen oder Bauteile, die sich nicht nach dem Gesetz der Reihe ordnen lassen, können in **Typengruppen** zusammengefasst werden (RAPP 2010, S. 49f).

## Plattform

Mit einer Produktplattform wird versucht, die **Wiederholteilerate** und den Anteil an **Übernahmeteilen** über das Produktportfolio hinweg zu erhöhen (MEYER & LEHNERD 1997). Zudem wird versucht, dieselbe Architektur über ähnliche Produkte beizubehalten. Bausteine einer Plattform werden kosten-, änderungs- und lebenszyklusgerecht entwickelt. Somit können aufwendigere Baugruppen mit längeren **Entwicklungszyklen** (wie ein Motor im Kraftfahrzeug) über mehrere Produktgenerationen hinweg verwendet werden. Unabhängig davon können Umfänge mit kürzeren Innovationszyklen (wie das Infotainment-System im Fahrzeug) schneller in den abgeleiteten Produkten verfügbar gemacht werden. Der Entwicklungs- und Optimierungsfokus wechselt von einem Produkt hin zum produktfamilienübergreifenden Gesamtportfolio (RAPP 2010, S. 73).

Nach MEYER & LEHNERD (1997) ist für eine Produktplattform eine **modulare Produktstruktur** erforderlich. Eine Produktplattform besteht aus der Summe von Komponenten, Schnittstellen und Funktionen, die über die ganze Produktfamilie vereinheitlicht werden können und somit stabil sind.

Plattformteile sind in der Regel teuer und erfordern ein **hohes Investment**. Daher wird versucht, möglichst keine teuren Änderungen an der Plattform durchzuführen. Die Änderungsfortpflanzung würde sich im ungünstigen Fall auf gleich mehrere Produkte auswirken, in denen die Plattformteile verwendet werden. Für die Planung, Entwicklung und Optimierung von Plattformen wird auf ROBERTSON & ULRICH (1998), RAPP (2010) und MEYER & LEHNERD (1997) verwiesen.

**In der Praxis** werden unter einer Plattform auch technische Umfänge zusammengefasst, die beispielsweise in der Entwicklung nach denselben Prinzipien aufgebaut werden können (vgl. Fallstudie 1 bei Hilti: „Entwicklungsplattform Gleichstrommotoren“). In der

Automobilindustrie wurde lange Zeit auf das Plattform-Hut-Konzept gesetzt. So wurden bei zahlreichen Automobilkonzernen Produktplattformen entwickelt, die Gleichteileumfänge definierten und von unterschiedlichen Fahrzeugkonzepten genutzt wurden. Der Begriff Plattform hat sich in den letzten Jahren weiterentwickelt. Immer öfter ist von Baukastensystemen die Rede.

### Baukastensystematik

Ein **Baukasten** ist nach PAHL ET AL. (2007, S. 662ff) ein System zur Kombination von Bauteilen und Baugruppen, mit dem Ziel Produkte unterschiedlicher Gesamtfunktion abzuleiten. KOHLHASE (1997, S. 8) bietet folgende Definition an: „Ein Baukastensystem besteht aus einer Anzahl von Bausteinen, die anwendungsspezifisch ausgewählt und unter Beachtung von Verträglichkeit miteinander kombiniert werden, um in einem begrenzten Anwendungsbereich Baukastenprodukte zu konfigurieren.“ Baukästen können wiederum Baureihen, Module und Plattformen enthalten. RENNER (2007, S. 51) trägt ausführlich weitere Definitionen zum Thema Baukasten zusammen.


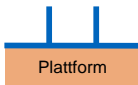

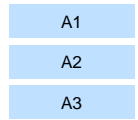
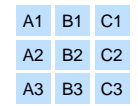
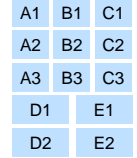

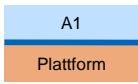
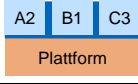
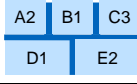
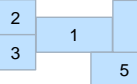
	Physikalische Plattform	Plattform mit Modulvarianten	Modulbaukasten mit fixer Struktur	Freie Modularisierung
Kommunaler Anteil über gesamte Produktstruktur				
Bausteininventar				
Beispielkonfiguration				
Kommunalitätsart	Physische Produktplattform		Kommunale Produktstruktur / Architektur	
	Kommunale Bausteine			
Einteilung nach KOHLHASE	Strukturgebundene Baukastensysteme			Modulare Baukastensysteme

Abbildung 3-2: Systematisierung von Baukästen (nach BRAUN 2013 und GRESHAKE 2011)

Abbildung 3-2 gibt einen Überblick über die Systematisierung von Baukastensystemen. Die Erzeugnisse, die aus einem Baukastensystem aufgebaut werden, können einen **kommunalen Anteil** aufweisen. Wie im vorherigen Abschnitt dargestellt, kann in der einfachen Form ein Basis-Baustein als **Plattform** dienen. Mit einem variablen Baustein, der als „**Hut**“ dient, wird das Erzeugnis auf die Anforderungen des Kunden angepasst.

Bietet der **Bausteininventar** mehrere wählbare Optionen, kann die Plattform mit **Modulvarianten** bestückt werden. Modulbaukästen mit fixer Struktur geben die grundsätzliche Anordnung der Bausteine vor. Ein explizit kommunaler Baustein als Plattform muss nicht

vorliegen. Bei der freien Modularisierung ist eine kommunale Architektur der Erzeugnisse und Anordnung der Bausteine nicht mehr gegeben. Es werden lediglich kommunale Bausteine verwendet.

Kompatibilität und somit Kombinierbarkeit von Bausteinen werden durch Gebote und Verbote ausgedrückt, die in einer **Verträglichkeitsmatrix** abgebildet werden (vgl. NURCAHYA 2009, S. 67ff; BONGULIELMI 2002, S. 57ff; ZAGEL 2010, S. 42f). Konfiguratoren gibt es für den externen Kunden zur individuellen Zusammenstellung gewünschter Produktoptionen, aber auch für den internen Produktentwickler zur Zusammenstellung von Lösungsbausteinen (KESPER 2012, S. 22ff).

In der Industrie werden Baukastensysteme unterschiedlicher Rangordnung eingesetzt. So gibt es nach Art der Herkunft beispielsweise Industriebaukästen, Unternehmensbaukästen oder Zuliefererbaukästen. Je nach Entscheidungsgremium innerhalb eines Unternehmens und Wichtigkeit eines Baukastensystems gibt es Haupt- und Teilbaukästen.

Ein typischer Vertreter der Baukastenbauweise ist der Volkswagenkonzern (Volkswagen 2014). Dort werden die Grundprinzipien verfolgt, die vom Kunden wahrnehmbaren Produktanteile zu individualisieren, wobei die übrigen Fahrzeugumfänge nach Möglichkeit standardisiert werden sollen. Die Module sollen marken- und segmentübergreifend eingesetzt werden. Dadurch sollen Kosteneinsparungen durch Standardisierung und Mehrfachverwendung von Modulen erreicht werden. Die Entwicklung neuer Fahrzeuge und Derivate soll dadurch schneller und kostengünstiger erfolgen. Die Standardisierung führt zu Qualitätssteigerung. Weitere Vorteile ergeben sich durch Reduzierung der Investitionskosten und eine Optimierung der Modullebenszyklen.

### **Klassifizierung von Bausteinen**

Durch Klassifizierung von Bausteinen kann die Ordnung innerhalb eines Produktportfolios erhöht werden. Bausteine mit gleichen Eigenschaften werden in Gruppen zusammengefasst und können somit einfacher überblickt und betreut werden. Klassifizierungen und Gruppierungen bieten die Möglichkeiten, die für den Systemarchitekten planungsrelevanten Informationen konsistent zu halten und Klarheit über das Produktportfolio zu vermitteln.

Nach PAHL ET AL. (2007, S. 668) und KOHLHASE (1997, S. 39) werden reine, geschlossene und offene Baukastensysteme unterschieden (vgl. Abbildung 3-3). JESCHKE (1997, S. 52) unterscheidet in Anlehnung an PAHL ET AL. (2007, S. 664) Grund-, Anpass-, Sonder-, Hilfs- und Nicht-Bausteine je nachdem, welche Funktionen der Baustein erfüllt (vgl. Abbildung 3-3).

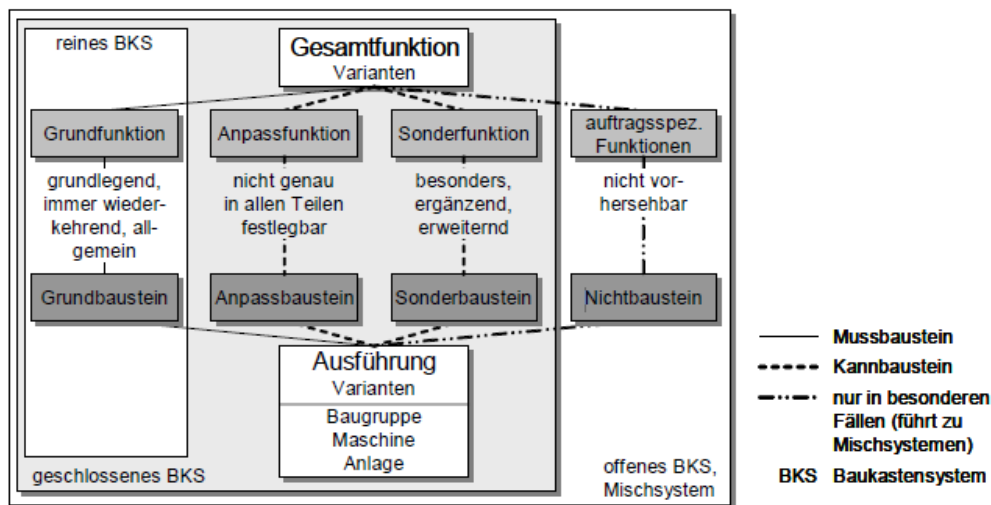


Abbildung 3-3: Unterscheidung von Baukastensystemen nach Bausteinarten (PAHL ET AL. 2007, S. 664)

**Muss-Bausteine** sind Funktionsträger, die in jedem Produkt vorhanden sein müssen, **Kann-Bausteine** werden nach Bedarf verbaut. Funktionale Abhängigkeiten zwischen den Bausteinklassen sind zu verhindern. Nachfolgende Tabelle 3-2 gibt einen Überblick über Klassifikationskriterien unterschiedlicher Literaturstellen.

Kriterium	Definition	Ausprägungen	Autoren
Bedeutung	Notwendigkeit eines Bausteins innerhalb der Baukastenproduktvarianten	Muss- und Kann-Bausteine	EHRENSPIEL & MEERKAMM (2013), PAHL ET AL. (2007)
Varianz	Existenz von Varianten zu den Bausteinen oder Vorkommen der Bausteine nur in einer einzelnen Ausführung	Substitutionale und nicht substitutionale Bausteine	BIEGERT (1971)
Konfiguration	Hierarchieebene eines Bausteines bei Verflechtung von Baukästen unterschiedlicher Ordnungen	Elementare und konfigurierte Bausteine	KOHLHASE (1997)
Funktion	Zu erfüllende Teilfunktion eines Bausteines zur Realisierung einer Gesamtfunktion	Grund-, Sonder-, Hilfs- und Anpassbausteine	EHRENSPIEL & MEERKAMM (2013), PAHL ET AL. (2007)
Kosten	Berücksichtigung der Kosten eines Bausteins	A-, B-, C-Bausteine	HANSEN ET AL. (2012)

Tabelle 3-2: Kriterien zur Klassifikation von Bausteinen in der Literatur (BRAUN ET AL. 2013, S. 26)

Eine Klassifizierung von Bausteinen kann auch durch deren **Verwendung** im Produktportfolio getätigt werden. **Gleichteile** werden von EHRENSPIEL ET AL. (2005, S. 311) als Teile definiert, die mehrmals in einem Produkt vorkommen. Durch Gleichteile können Skaleneffekte erzielt werden und sollten daher möglichst oft in einem Produkt eingesetzt werden (EHRENSPIEL &

MEERKAMM 2013, S. 675). **Wiederholteile** werden auch in anderen Produkten verwendet (EHRLENSPIEL ET AL. 2005, S. 310). Nach RENNER (2007, S. 67) werden die beiden Begriffe Gleich- und Wiederholteile in der Automobilindustrie oft synonym verwendet.

Des Weiteren dienen **Kommunalitäten** von Bausteinen als Klassifikationskriterium. RENNER (2007, S. 67ff) unterscheidet sieben Arten von Kommunalitäten von Produkten:

- **Konzeptkommunalität:** Ziel- bzw. Funktionserfüllung erfolgt auf dieselbe Art und Weise. Dadurch kann Entwicklungsaufwand eingespart werden.
- **Prozess- und Werkstoffkommunalität:** Es werden dieselben Prozesse durchlaufen. Häufig ergibt sich dadurch auch Kommunalität bei den Werkstoffen. Dadurch ist eine bessere Auslastung von Fertigungsanlagen möglich.
- **Lieferantenkommunalität:** Lieferant bietet Bauteile mit Kommunalität an, was auf ein dahinterstehendes Baukastensystem schließen lässt. Von den Kostenvorteilen können unter Umständen auch die OEMs profitieren.
- **Bauraum- und Schnittstellenkommunalität:** Durch eine feste Zuteilung von Bauräumen für Bauteile („Package“) können Synergien geschaffen werden.
- **Bauteilkommunalität:** Mehrfachverwendung oder Wiederverwendung von Bauteilen in Produkten und über die Produktfamilie hinweg. Dadurch können Skalen und Synergieeffekte bei Investitionen, Herstellkosten und Entwicklungsaufwand genutzt werden.

Eine weitere Möglichkeit ist es, Bausteine nach dem **Grad ihrer Standardisierung** zu klassifizieren. Genormte Bausteine können innerbetrieblich und überbetrieblich standardisiert sein. Neben Lösungsbausteinen können auch **Schnittstellen** standardisiert werden, um Austauschbarkeit von Bausteinen zu ermöglichen. PONN & LINDEMANN (2011, S. 167ff) definieren Schnittstellen als Verknüpfungen zwischen Elementen eines Systems, die im Falle einer Verbindung von physischen Bauteilen fest oder lösbar sein können. Ebenso wie PIMMLER & EPPINGER (1994) unterscheiden PONN & LINDEMANN (2011) zwischen geometrischen, stofflichen, energetischen oder signaltechnischen Schnittstellen.

Die Klassifikation von Bausteinen und Schnittstellen kann sich **über den Verlauf der Zeit verändern**. Wenn sich beispielsweise herausstellt, dass bestimmte Grundbausteine nicht mehr benötigt, oder ursprünglich als Sonderbaustein angelegte Module am Markt hoher Nachfrage erfreuen. Daher müssen in regelmäßigen Abständen die Klassifizierungen überprüft und angepasst werden. Ansonsten besteht die Gefahr, durch mangelnde Ordnung ungewollt die Komplexität zu steigern.

Die hier aufgeführten Strukturierungsmöglichkeiten können einerseits die Ordnung in komplexen Systemen erhöhen, andererseits kann sich eine nicht durchgängige und inkonsistente Verwendung negativ auf die Komplexität auswirken. Fehler, Ausnahmen und Sonderbestimmungen erschweren dann erheblich das Arbeiten mit diesen Strukturen. Im nächsten Kapitel werden Gestaltungsprinzipien, Handlungsfelder, Entscheidungshilfen und Werkzeuge zur Komplexitätsreduktion vorgestellt.

## 3.2 Methoden zur Komplexitätsreduktion

Die Gestaltung komplexer Strukturen ist selbst eine komplexe Aufgabe, wie RECHTIN es beschreibt: „*The essence of architecting is structuring. Structuring can mean bringing form to function, bringing order out of chaos, or converting the partially formed ideas of a client into a workable conceptual model. The key techniques are **balancing** the needs, **fitting** the interfaces, and **compromising** among the extremes*” (in KOSSIAKOFF ET AL. 2011, S. 223f).

In der Regel wird eine Produktportfolioarchitektur nicht neu entworfen, sondern **wächst über die Zeit** in Unternehmen entweder geplant oder ungeplant. Die Anzahl und Unterschiedlichkeit an Bausteinen, Regeln, Ausnahmen und Wechselwirkungen mit Stakeholdern im Unternehmen führen zu einer massiven Komplexitätssteigerung im Produktportfolio und damit auch für die Arbeit mit dieser Struktur. Somit stellt sich die Frage, wie die Architektur und Bestandteile des Produktportfolios bewertet, beeinflusst oder gesteuert werden können.

Die Kunst besteht im **Abwägen und dem Finden von Kompromissen** in einem Spannungsfeld von Zielen. Strategisches und operatives Denken sind erforderlich, um sinnvolle und nachhaltige Entscheidungen zu treffen. Eben jene Balance herzustellen, erweist sich als Aufgabe, für die Systemüberblick, klare Strukturen, Handlungsfähigkeit, leistungsfähige Bewertungs- und Analysewerkzeuge Voraussetzungen sind.

Der Stand der Technik weist eine **Vielzahl an Methoden** auf, Komplexität in einem Produktportfolio präventiv zu vermeiden, nachträglich zu reduzieren oder zu beherrschen. An dieser Stelle soll untersucht werden, inwieweit sich jene Methoden zur Steigerung der Transparenz sowie für das effektive und effiziente Arbeiten in komplexen Produktportfolios eignen. Hierfür werden generelle Gestaltungsprinzipien, Methoden zur Reduzierung der Vielfalt, Methoden zur Bewertung und Werkzeuge des Komplexitätsmanagement untersucht.

### 3.2.1 Generelle Gestaltungsprinzipien komplexer Produktstrukturen

Die Architektur eines Produktportfolios beschreibt neben den Elementen der Produktgliederung und deren horizontalen wie vertikalen Relationen die zugrundeliegenden Gestaltungsprinzipien. **Generelle Gestaltungsprinzipien** zum Aufbau komplexer Systeme sind (nach HABERFELLNER ET AL. 2012, S. 137ff):

- **Idealität:** Das Verhältnis von nutzbringenden zu ungewünschten Funktionen und Effekten soll möglichst groß sein. Im Sinne der Systemgestaltung ist dies die Forderung nach möglichst geringer Komplexität. Ein ideales System greift bereits bestehende Lösungsbausteine von „Um- und Übersystemen aus der Systemumgebung“ auf.
- **Unabhängigkeit:** Einflüsse sich verändernder Systemparameter sollen möglichst geringen Einfluss auf andere Systembestandteile haben (vgl. auch Axiomatic Design von SUH 2001). Systemparameter können physikalische Wirkprinzipien, technische Parameter oder auch Komponenten sein.
- **Modularer Aufbau:** Bausteine eines Systems sollen so gebildet werden, dass sie abgrenzbare, wiederverwendbare Funktionen erfüllen. Die Systemgrenzen der Bausteine sollen so gestaltet werden, dass sie nach außen hin möglichst wenige und



standardisierte Schnittstellen („loose coupling“) und starke Abhängigkeiten innerhalb der Systemgrenzen („strong cohesion“) haben.

- **Piecemeal Engineering:** Größere Veränderungen in unübersichtlichen, komplexen Systemen, sollten vermieden werden. Zwar kann eine umfassende Änderung geplant werden, diese sollte aber in kleineren Schritten durchgeführt werden, so dass mögliche Fehler wieder rückgängig gemacht werden können.
- **Minimale Präjudizierung:** Dieses Prinzip besagt, dass einer Lösungsalternative mit den größeren Freiheitsgraden und Gestaltungsfreiräumen im Zweifel der Vorzug gegeben werden sollte. Gerade bei Änderungen soll diese Flexibilität genutzt werden, eine gewisse Robustheit gegenüber Änderungen zu gewährleisten.

Im Zusammenhang mit der Architekturgestaltung fügen sie noch hinzu (HABERFELLNER ET AL. 2012, S. 188ff):

- **Integrierbarkeit** bedeutet, dass Kompatibilität und Interoperabilität von Elementen und Subsystemen durch Verwendung von offenen, standardisierten oder gemeinsamen Schnittstellen sichergestellt wird. Dies ermöglicht die Austauschbarkeit von Komponenten und somit die Flexibilität im Produktportfolio.
- **Skalierbarkeit** bedeutet, dass der Funktions- und Leistungsumfang eines Produktes aufwandsarm angepasst werden kann.
- **Dezentralisierung** bedeutet, dass bestimmte Funktionen, Steuerungen oder Informationsverarbeitung nicht an einem zentralen Ort eines Produktes zusammengefasst werden, sondern verteilt implementiert werden. Dies ermöglicht Sicherheit gegen Ausfall und leichtere Änderbarkeit.

Im Sinne der Gestaltungsprinzipien muss Übersicht und Klarheit im Produktportfolio geschaffen werden. Kontinuierliches „Aufräumen“ umfasst Strukturierung der eingesetzten und Aussortierung unnötiger Bausteine. Des Weiteren kann die Ordnung durch **Standardisieren und Vereinfachen** erreicht werden. EHRENSPIEL & MEERKAMM (2013, S. 723) beschreiben den Erfolg, der durch „Aufräumaktionen“ bei variantenreichen Bauteilen erzielt werden kann. In dem Beispiel wurden die Varianten eines Antriebsflansches von Hunderten auf nur mehr zwei Baugrößen reduziert. Hierfür wurden mit den zuständigen Konstrukteuren die 416 unterschiedlichen Ausprägungen für die sechs Hauptkonstruktionsmaße diskutiert und unnötige Ausprägungen gestrichen. Neben den Komplexitätskosten konnten die Herstellkosten auf 46% reduziert werden.

Angesichts der zu berücksichtigenden Menge von Produkten und Bausteinen müssen für die Einhaltung der Gestaltungsprinzipien geeignete Werkzeuge mit entsprechenden Analyse-, Strukturierungs-, Bewertungs- und Überwachungsfunktionen eingesetzt werden.

### 3.2.2 Handlungsfelder zur Reduzierung der Vielfalt

Die Vielfalt in Produktportfolios führt zu hoher Komplexität. In der Literatur finden sich **zahlreiche Ansätze** zur Reduzierung der vielfaltsinduzierten Komplexität. PULM (2004, S. 128ff) gibt einen umfangreichen Überblick über die Methoden des Varianten Managements. KRAUSE & RIPPERDA (2013) analysieren ausgewählte Methoden, die in den letzten zwanzig Jahren entwickelt wurden, inwiefern sie Aspekte unterstützen wie variante Produkte,

Modularisierungsmöglichkeiten, Visualisierungsformen, Einbindung interdisziplinärer Expertise, Nutzerleitfaden, situative Anpassbarkeit, Berücksichtigung von Prozessen und Kosten sowie Einsetzbarkeit in der Konzeptphase. Eine umfangreiche Zusammenfassung über die Gestaltungsmöglichkeiten von Produktfamilien und plattformbasierter Produktentwicklung bieten auch JIAO ET AL. (2006) an.

PINE, II. (1993) führte in die Problematik unter dem Oberbegriff **Mass Customization** ein. Im deutschsprachigen Raum findet sich das Thema unter Kundenindividuelle Massenproduktion wieder (PILLER 1998). Modularisierung und Plattformsätze werden allgemein im industriellen Umfeld angewendet, um Produktvarianten und deren interne Komplexität zu reduzieren (BALDWIN & CLARK 2000, S. 5ff).

Je nachdem welche **Strategie** ein **Unternehmen** verfolgt, werden unterschiedliche Schwerpunkte hinsichtlich der Ziele Variantenvielfalt und Komplexität gelegt. Das Produktprogramm eines Premiumherstellers in der Fahrzeugindustrie erfordert ggf. eine höhere Variantenvielfalt als das eines Massenherstellers (DANNER 1996, S. 146). Entsprechend verschieden sind auch die Strategien im Variantenmanagement. Ein Massenhersteller verfolgt ggf. das Ziel der Variantenreduzierung stärker als ein Premiumhersteller.

### Variantenmanagement

FRANKE (2002) fasst **Ansätze zum Variantenmanagement** in Theorie und Praxis zusammen. RENNER (2007, S. 66ff) grenzt Möglichkeiten einer variantengerechten Gestaltung voneinander ab. Er betrachtet in diesem Kontext Baukastenbauweise, Gleichteil- und Wiederholteilbauweise, Kommunalitäten, Modularisierung, Plattformen, Baureihen und Variantenbauweise. Ansätze zur Beherrschung der Variantenvielfalt werden in Ansätze mit oder ohne Beeinflussung der Produktstruktur eingeteilt (vgl. Tabelle 3-3). Diese Ansätze untergliedert RENNER (2007, S. 41) nach den Strategien zum Komplexitätsmanagement: Komplexität reduzieren, vermeiden, beherrschen (vgl. Tabelle 3-3).

Komplexität	Ansätze mit Beeinflussung der Produktstruktur	Ansätze ohne Beeinflussung der Produktstruktur
Reduzieren	<ul style="list-style-type: none"> <li>Gleich- und Wiederholteilbildung</li> </ul>	<ul style="list-style-type: none"> <li>Pflege und Bereinigung des Teilestamms (Sortimentbereinigung)</li> </ul>
Vermeiden	<ul style="list-style-type: none"> <li>Baureihenansatz</li> <li>Plattformsatz</li> <li>Modularisierung</li> </ul>	<ul style="list-style-type: none"> <li>Wiederholteilverwendung</li> <li>Verwendung überbetrieblicher und betriebsinterner Normen</li> <li>Paketbildung</li> </ul>
Beherrschen	<ul style="list-style-type: none"> <li>Baukastenansatz</li> </ul>	<ul style="list-style-type: none"> <li>Implementierung von verursachungsgerechten Bewertungsverfahren</li> </ul>

Tabelle 3-3: Ansätze zur Beherrschung der Variantenvielfalt (nach RENNER 2007, S. 41 und JESCHKE 1997, S. 61)

### Komplexität reduzieren

RENNER (2007, S. 41) schlägt zur Reduzierung der Komplexität Gleich- und Wiederholteilbildung sowie die Pflege und Bereinigung des Teilestamms vor. EILMUS ET AL. (2013) sehen großes Potential für eine produktfamilienübergreifende Gleich- und Wiederholteilstrategie und stellen eine Methode zur Entwicklung einer umfassenden Produktprogrammstrategie vor. Vielfaltsreduktion kann nach PILLER & WARINGER (1999, S. 29) auf Seiten der Produktplanung und –entwicklung und der Produktion und Auftragsabwicklung erfolgen. Hervorzuhebende **Maßnahmen der Produktentwicklung** sind Reduktion der Materialvielfalt, Erhöhung der Mehrfachverwendung und des Standardisierungsgrades und die Realisierung standardisierter Baugruppen und Komponentenschnittstellen. **Produktionsseitig** kann u.a. durch Verschiebung des Variantenbestimmungspunktes (vgl. auch SCHUH 2005) oder durch Fertigungssegmentierung die Vielfalt reduziert werden.

Für die Identifikation von **unnötigen Varianten** als Grundlage für die Bereinigung des Teilestamms verweisen EHRENSPIEL & MEERKAMM (2013, S. 720ff) auf die ABC-Analyse. Eine Gegenüberstellung aller Produkte der Produktpalette mit ihrem jeweiligen Beitrag zum Gesamtumsatz bzw. zum Deckungsbeitrag. Nach der 80-zu-20-Regel tragen 80 % der Produkte zu weniger als 20 % des Umsatzes bei. Daher „muss eine große Anzahl nicht gängiger Artikel als real unwirtschaftlich gewertet werden“ (EBD.). Weitere Hilfsmittel zur Suche nach unnötigen Varianten sind die Zeitreihenanalyse und die Portfoliotechnik (JESCHKE 1997, S. 18). Allerdings fehlen geeignete Rechnerwerkzeuge und Methoden systematisch Strukturfehler in großen Datenmengen zu finden. EHRENSPIEL & MEERKAMM (2013, S. 722) geben eine Übersicht über technische Maßnahmen zur Reduktion der Teilevielfalt:

Maßnahme	Maßnahmenkategorie
Umgestaltung mehrerer Teile zu einem Teil	Integralbauweise
Möglichst viele gleiche Teile in einem Produkt verwenden	Gleichteile
Teile in unterschiedlichen Produkten verwenden	Wiederholteile
Teile gleicher Funktion standardisieren	Teilefamilie
Mehrfachverwendung von Teilen und Baugruppen	Baukastensystem
Vermeidung von Sonderkonstruktionen bei Produkten gleicher Funktion	Baureihen
Normteile verwenden	Normteile
Kaufteile verwenden	Kaufteile

*Tabelle 3-4: Maßnahmen zur Reduktion der Teilevielfalt (EHRENSPIEL & MEERKAMM 2013, S. 722)*

Nach BEMBENNECK (2005) kann durch die Verwendung von Wiederholteilen die Konstruktionszeit im Maschinen- und Anlagenbau um 20-50%, die Herstellkosten um 5-20%, die Teilevielfalt um 20-40% und die Fehlerkosten um 10-30% reduziert werden. Allerdings

besteht bei Wiederverwendung auch die Gefahr innovationshemmend in der Produktentwicklung zu wirken.

### **Komplexität vermeiden**

Zur Vermeidung von Komplexität durch Erhöhung der Ordnung im Produktportfolio können Baureihensystematiken (vgl. Kapitel 3.1.3) eingesetzt werden (PAHL ET AL. 2007). GERHARD (1984, S. 13) erläutert Vorteile der Methodik **Baureihenentwicklungen**, die zu standardisierten, vereinheitlichten Konstruktionen und vereinfachter Planung führen. MAIER (1993) stellen hierfür ein Verfahren zur **Ähnlichkeitsermittlung** vor.

Auch durch **Plattformansätze, Modularisierung und Modularität von Baugruppen** kann Komplexität vermieden werden (vgl. Kapitel 3.1.1). Gute Modularisierung trägt nach HABERFELLNER ET AL. (2012, S. 138) dazu bei,

- die Abstimmung und Koordination im Unternehmen zu vereinfachen,
- die Übersichtlichkeit zu verbessern,
- Mehrfachverwendung von Bausteinen zu begünstigen,
- die Anpassbarkeit von Systemen zu steigern,
- die Wartung zu vereinfachen und
- Kosten in Logistik- und Produktionsprozessen zu reduzieren.

ULRICHS (1995) Definition von Produktarchitektur ist Grundlage für die technisch-funktionale Modularisierung. BLEES (2011) hat allerdings gezeigt, dass weitere Aspekte berücksichtigt werden sollten, wenngleich die technisch-funktionale Modularisierung als Schritt seiner Methodik beinhaltet ist. Er kombiniert die technisch-funktionale Modulbildung (vgl. GÖPFERT 1998; ULRICH 1995) mit produkt-strategischen Modultreibern entlang der Produktlebensphasen (in Anlehnung an BLACKENFELT 1999; STAKE 2000; ERIXON 1998). Die beiden Ansätze kombiniert er zu einem integrierten Vorgehen zur lebensphasengerechten Modulbildung.

### **Komplexität beherrschen**

Für die Komplexitätsbeherrschung schlägt RENNER (2007, S. 41) Baukastensystematiken vor (vgl. Kapitel 3.1.3). Des Weiteren wird die Implementierung von verursachungsgerechten Bewertungsverfahren empfohlen. Hierfür entwickelt JESCHKE (1997) eine Methode zur wirtschaftlichen Bewertung von Standardisierungsmaßnahmen.

RATHNOW (1993) adressiert die Beherrschung der Variantenvielfalt und die Bestimmung der optimalen Variantenvielfalt. LINDEMANN ET AL. (2006) schlagen ein Vorgehen zur Individualisierung von Produkten vor. Weitere Verfahren zur Beherrschung und Optimierung der Variantenvielfalt finden sich u.a. bei SCHUH (2005) vgl. VMEA, KRAUSE ET AL. (2014) (s.u.), PRASAD (1998) und WILDEMANN (2013, S. 83ff).

RENNER (2007, S. 89) illustriert die Wirkung von Maßnahmen, die Teilevielfalt im Produktportfolio einzudämmen (vgl. Abbildung 3-4). Die Ist-Situation sei dadurch gekennzeichnet, dass überproportional viele Teile bei steigender Variantenanzahl eingesetzt werden müssen. Durch die Einführung eines Ordnungssystems können Lösungen einfacher gefunden, Konfigurationsmöglichkeiten schneller verstanden und der Aufbau und Struktur besser verstanden werden. Produktbaukästen erhöhen weiterhin die Wiederverwendbarkeit von

Bausteinen. Nicht zuletzt kann in einem Produktkonfigurator Wissen über die Kombierbarkeitsregeln formalisiert und somit automatisiert werden.

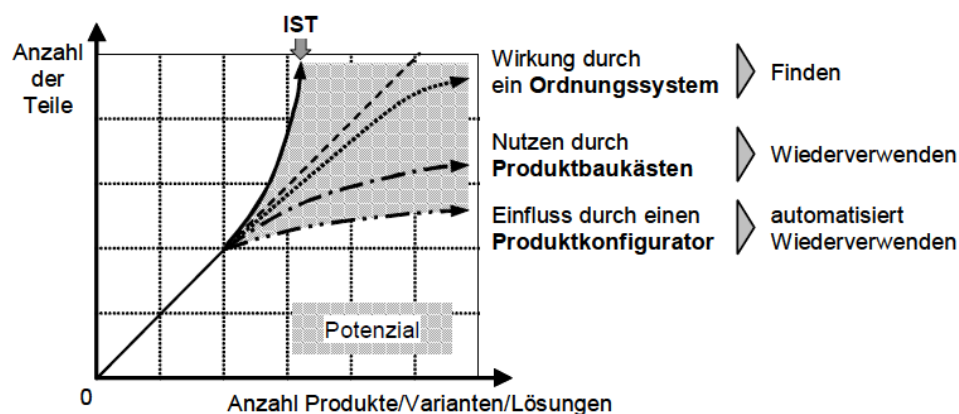


Abbildung 3-4: Wirkung von Maßnahmen des Varianten Managements auf die vielfaltsbedingte Komplexität (RENNER 2007, S. 89)

### Variant Mode and Effects Analysis

In der **Variant Mode and Effects Analysis (VMEA)**<sup>14</sup> wird eine Methode zur Analyse, Gestaltung und Bewertung eines Produktsortiments beschrieben. Im Rahmen der Produktgestaltung werden „u.a. die Festlegung der Kundenfunktionen, gesetzlichen Anforderungen, variantenerzeugenden Eigenschaften sowie die Ausgestaltung der anzubietenden Kombinatorik dieser Eigenschaften und Funktionen“ durchgeführt (SCHUH 2005, S. 145). Die variantenerzeugenden Eigenschaften und deren Ausprägungen können in einer Merkmals-Ausprägungsmatrix dargestellt werden.

Es ergibt sich ein eindimensionales Ordnungsschema, das technischen und marktseitigen Kombinationsgeboten und –verboten unterliegt. Die Kombinationsregeln werden in der Kombinationsmatrix festgehalten. Aus den erzeugten Informationen lässt sich dann eine **Variantebaumstruktur** ableiten (vgl. Abbildung 3-5). Basis der Variantenbaumstruktur ist ein „Grund- oder Trägerbauteil, an welches alle weiteren Anbauteile mit diesem durch Linien verbunden sind.“ Nach unten hin gliedert sich die Montagereihenfolge auf.

<sup>14</sup> CAESAR 1991; SCHUH 2005

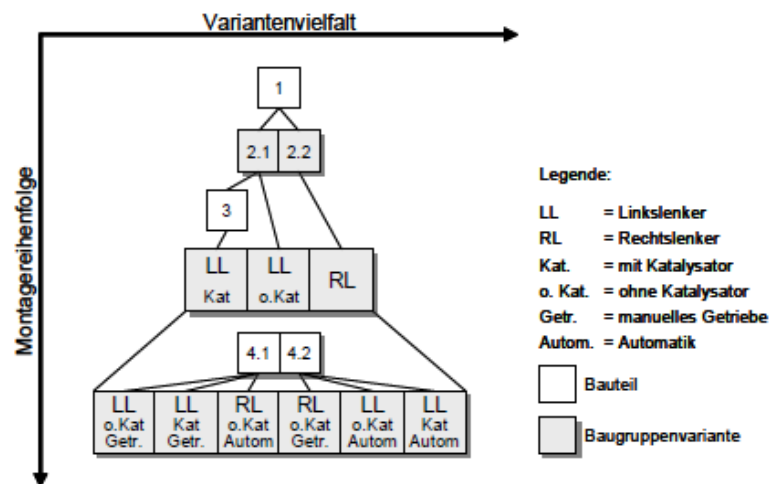


Abbildung 3-5: Merkmals-Ausprägungs- und Kombinationsmatrix (SCHUH 1988, S. 47)

SCHUH (2005, S. 166f) beschreibt **Ziele** einer variantenorientierten Produktgestaltung aus Produktionssicht:

- Anzahl der Varianten auf das am Markt geforderte Variantenspektrum reduzieren
- Funktions- und montagegerechte Konstruktion
- Erhöhung des Automatisierungsgrades
- Reduktion der Bearbeitungszeit

Daraus abgeleitet empfiehlt er Ziele und Maßnahmen für die Handlungsfelder Standardisierung, Individualisierung und Modularisierung.

Die VMEA Methode unterstützt dabei, Transparenz über planungsrelevante Zusammenhänge zu erhöhen und bietet Methodenbausteine für eine systematische Beherrschung der Variantenvielfalt. Zusätzlich wird die Veranschaulichung der Methoden mit dem Software-Werkzeug Complexity Manager angeboten.

### Integrierter PKT-Ansatz zur Entwicklung modularer Produktfamilien

Den **integrierten PKT-Ansatz zur Entwicklung modularer Produktfamilien** stellen KRAUSE ET AL. (2013) und KRAUSE ET AL. (2014) vor (vgl. Abbildung 3-6). Ziel des Ansatzes ist die geforderte externe Variantenvielfalt durch eine möglichst kleine interne Produktvielfalt darstellen zu können. Hierfür wird ein Methodenbaukasten erarbeitet, der die Reduzierung der Prozesskomplexität wie auch die Reduzierung der internen Vielfalt in Produktfamilien unterstützt. Methoden der Vielfaltsreduzierung in Produktfamilien umfassen Methoden

- zur Produktprogrammplanung (JONAS ET AL. 2012),
- zur variantengerechten Produktgestaltung (KIPP 2012; BLEES ET AL. 2010),
- zur Lebensphasenmodularisierung (KIPP ET AL. 2010; BLEES 2011),
- zur Entwicklung modularer Produktprogramme (EILMUS ET AL. 2013; EILMUS & KRAUSE 2012),
- sowie weitere Methodenbausteine (vgl. KRAUSE ET AL. 2014).

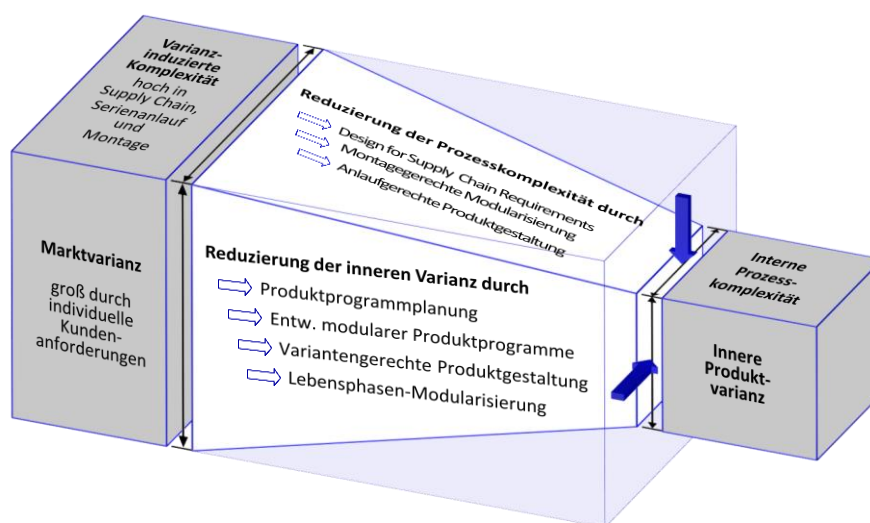


Abbildung 3-6: Integrierter PKT-Ansatz zur Entwicklung modularer Produktfamilien

Die Methoden beinhalten ein breites Spektrum an methodischen Werkzeugen und Visualisierungsmöglichkeiten. Beispiele finden sich in KRAUSE ET AL. 2013, S. 546ff:

- *Tree of Variety,*
- *Product Family Function Structure,*
- *Module Interface Graph,*
- *Variety Allocation Model,*
- *Module Process Chart,*
- *Program Structuring Model,* und
- *Carryover Assignment Plan.*

Der Ansatz unterstützt die Integration von technisch-funktionalen und produktstrategischen Modultreibern entlang der Produktlebensphasen, die Konstruktion variantengerechter Produkte, die Abstimmungstätigkeiten im Entwicklerteam durch geeignete Visualisierungen und die Reduzierung der Vielfalt unter Berücksichtigung unternehmensspezifischer Rahmenbedingungen (KRAUSE ET AL. 2013, S. 545). Ein Software-Werkzeug, in dem die Methoden und Visualisierungen rechnergestützt verwendet werden können, ist noch nicht verfügbar.

### 3.2.3 Entscheidungsunterstützung und Bewertung

Das Arbeiten mit komplexen Produktportfolios ist mit zahlreichen operativen wie strategischen Entscheidungen verbunden. Maßnahmen, wie sie aus den oben stehenden Methoden abgeleitet werden können, können weitreichende Folgen im Produktportfolio haben und unvorhergesehene Wechselwirkungen auslösen. Der Systemarchitekt muss Entscheidungen oft unter Zeitdruck und ohne die nötigen Informationen treffen.

#### Entscheidungstheoretische Grundlagen

LAUX ET AL. (2012, S. 4) unterscheiden zwischen **präskriptiver und deskriptiver Entscheidungstheorie**. Während die deskriptive Entscheidungstheorie beschreibt, wie aus

psychologischer Sicht Entscheidungen getroffen werden, behandelt die präskriptive (oder normative) Entscheidungstheorie die Logik der Entscheidungsfindung und wie rationale Entscheidungen getroffen werden können.

Entscheidungen im Alltagsleben unterscheiden sich nicht in der Art und Struktur der Schritte von komplexen Entscheidungen im industriellen Umfeld. In Abbildung 3-7 sind die Schritte der Entscheidungsfindung nach LAUX ET AL. (2012) dargestellt.

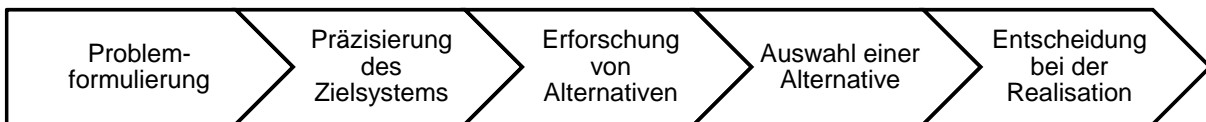


Abbildung 3-7: Schritte der Entscheidungsfindung (LAUX ET AL. 2012)

Die Einzelschritte der Entscheidungen werden im Alltag schneller und teilweise unbewusst durchlaufen. Komplexe Entscheidungssituationen können durch eine Vielzahl von Einflussfaktoren, eine multi-kriterielle Zielformulierung und zahlreiche, teilweise unvorhersehbare Auswirkungen beschrieben werden (LAUX ET AL. 2012).

Für „**rationale Entscheidung**“ müssen nach EISENFÜHR ET AL. (2010, S. 6) folgende Kriterien gelten:

- **Problemrelevanz:** Wird das richtige Problem gelöst?
- **Informationsaufwand:** Ist der Aufwand für die Beschaffung und Verarbeitung der Bedeutung der Entscheidung angemessen?
- **Objektivität der Datengrundlage:** Werden relevante und objektive Daten verwendet?
- **Nutzenorientierung:** Sind die Ziele klar?

Typische **Entscheidungen in der Produktentwicklung** fassen KRISHNAN & ULRICH (2001) zusammen. Sie differenzieren zwischen Entscheidungen im Rahmen laufender Projekte und um Entwicklungsprojekte zu initialisieren. LINDEMANN (2007) überträgt den Entscheidungsprozess in das Münchner Vorgehensmodell für Problemlösungen im Ingenieurbereich. Die Relevanz systematischer Entscheidungsprozesse bei Änderungen innerhalb der Produktportfolioarchitektur behandeln KISSEL & LINDEMANN (2013).

Alltägliche Beispiele zeigen, dass wir durchaus in der Lage sind, **Entscheidungen in komplexen Sachverhalten** intuitiv und aus Erfahrung „richtig“ zu treffen. Genauso gut können wir in Anbetracht der Fülle an Abhängigkeiten einer komplexen Entscheidungssituation scheitern und sogar kontraproduktive Folgen bewirken (vgl. DÖRNER 2003). Dennoch müssen komplexe Entscheidungen oft unter Zeitdruck nach dem Bauchgefühl getroffen werden, ohne zahlreiche Einflussfaktoren und deren Wechselwirkungen rational gegeneinander abzuwägen. In der Literatur werden daher zahlreiche Ansätze zur Verdichtung von Informationen und Bilden von Kennzahlen vorgestellt, um die Entscheidungsfindung in komplexen Entwicklungssituationen zu unterstützen.



## Bewertungsmethoden in der Architekturentwicklung

Bewertungen dienen dem Erkennen von kritischen Handlungsfeldern und der Entscheidungsunterstützung. Kennzahlen sollen hierbei wesentliche Informationen für den Entscheider in der Form verdichten, damit er Situationen umfassend bewerten kann. Für die Bewertung von Produktarchitektur und variantenreichen Produktportfolios können zahlreiche Ansätze in der Literatur gefunden werden.

JIAO ET AL. (2006, S. 11ff) geben einen breiten Überblick über die Architekturgestaltung und Möglichkeiten der Bewertung. Dabei stellen sie folgende Aspekte der Architekturgestaltung heraus:

- Mit **Modularität** soll der richtige Grad der Dekomposition im System gefunden werden. Hierfür werden unterschiedliche Kriterien herangezogen. HÖLTTÄ-OTTO ET AL. (2012) kategorisieren die Kriterien der Metriken in der Literatur nach Ähnlichkeit, Kombination und Kopplung. Dennoch gibt es keine Einigkeit darüber, wie Modularität bewertet werden soll. Ansätze finden sich unter anderen bei MARTIN & ISHII (1997), ERIXON (1998), NEWCOMB ET AL. (1998), SOSA ET AL. (2004) und HÖLTTÄ-OTTO & DE WECK (2007).
- **Kommunalität** soll den Grad der Gleichheit zwischen Produkten bewerten. MARTIN & ISHII (1997) führen den Kommunalitätsindex ein, der das Verhältnis von Gleichteilen zu singulären Teilen berechnet. THEVENOT & SIMPSON (2006) untersuchten hierzu sechs Kommunalitätsindizes anhand neun verschiedener Produktfamilien. Weitere Ansätze finden sich unter anderen bei KOTA ET AL. (2000) und JIAO & TSENG (2000b).
- Die **Varietät und Unterschiedlichkeit** sind die Gegenstücke zur Kommunalität. Ansätze hierzu finden sich unter anderen bei MARTIN & ISHII (1997), MARTIN & ISHII (2002), FUJITA & YOSHIDA (2004) und SIMPSON & D'SOUZA (2004).
- **Kostenbewertungen** greifen den Kompromiss aus Individualisierung und Standardisierung auf. Ansätze finden sich unter anderen bei KIM & CHHAJED (2001); FIXSON (2005); GONZALEZ-ZUGASTI ET AL. (2000).
- Die Ermittlung von **Profit** alleine gilt als unrealistisch (TARASEWICH & NAIR 2001). In der Regel werden multi-kriterielle Evaluationen zur Bewertung der Leistungsfähigkeit im Architekturmilieu herangezogen. Für Beispiele wird unter anderen auf DE WECK ET AL. (2003) und FUJITA & YOSHIDA (2004) verwiesen.
- Plattformbezogene Kennzahlen wie **Plattformeffizienz und Effektivität** finden sich bei MEYER & LEHNERD (1997). MARTIN & ISHII (2002) entwickeln als Kennzahlen den Generational Variety Index und den Coupling Index. Weitere Kennwerte werden unter anderen bei DE WECK ET AL. (2003) und JIAO & TSENG (2004) vorgestellt.

SIMPSON (2004, S. 9f) stellt heraus, dass vorrangig eine optimale Abwägung zwischen anforderungsgerechter Produktgestaltung und Erzielung von Kommunalitäten in der Gestaltung von Produktfamilien und Plattformen erreicht werden soll. Hierzu stellt er Optimierungsansätze aus der Literatur zusammen.

Des Weiteren entwickelt KOHLHASE (1997, S. 117ff) ein rechnergestütztes System zur Baukastenentwicklung und zieht hierfür Kennzahlen zur technisch-wirtschaftlichen Bewertung und Flexibilität heran. Kennzahlen zur Variantenvielfalt finden sich auch bei RAPP (2010, S. 42):

- **Mögliche Variantenzahl:** Anzahl der Produktkonfigurationen, die ohne Änderung der Produktstruktur erstellt werden können.
- **Reale Variantenzahl:** Anzahl der Produktkonfigurationen, die tatsächlich angeboten werden.
- **Teilevielfalt:** Entspricht der Anzahl an Stammdatensätzen, die verwaltet werden müssen.
- **Wiederverwendungsgrad:** Gibt an, in wie vielen Produktvarianten eine Elementvariante durchschnittlich vorkommt.
- **Wiederholhäufigkeit:** Gibt an, wie oft dasselbe Element durchschnittlich in einer Produktvariante auftaucht.

Zweifel an der praktischen Einsetzbarkeit einiger Kennzahlen äußern JUNGE (2005) und HEILEMANN ET AL. (2013). JUNGE (2005, S. 75) zieht als Fazit aus seiner Untersuchung zur Eignung von Kennzahlensystemen für die Bewertung von modularen Produktprogrammen folgende Schwachpunkte:

- **Aufwand der Datenbeschaffung:** Die erforderlichen Daten für die Kennzahlenermittlung werden in regulären Unternehmensprozessen häufig nicht erhoben. Der Aufwand der Datenerhebung ist groß.
- **Übersichtlichkeit der Information:** Unübersichtliche, unstrukturierte und überdimensionale Datenblätter erschweren die Informationsbeschaffung.
- **Transparenz der Ergebnisse:** Die Ermittlung der Kennzahlen muss nachvollziehbar und verständlich sein, um Akzeptanz zu erhalten.
- **Aussagekraft von Kennwerten:** Kennzahlen fassen Information in konzentrierter Form zusammen. Die Qualität der Aussagekraft hängt von der Visualisierung und Interpretierbarkeit ab.

HEILEMANN ET AL. (2013) untersuchten 20 ausgewählte Kennzahlen zur Bewertung der Modularität, Kommunalität und Varietät sowie strategisch und finanzorientierte Kennzahlen im industriellen Einsatz. Dabei kamen sie zu dem Schluss, dass viele der in akademischen Ansätzen geforderten Informationen in der Praxis nicht verfügbar oder nur mit mühevoller Datenbeschaffung und –aufbereitung zu erheben sind. Dies sehen die Autoren in Anbetracht mehrerer hundert Produktfamilien, Produkte und Komponenten im industriellen Kontext und der teils unklaren Übertragbarkeit der Ergebnisse als kritisch.

### 3.2.4 Werkzeuge des Komplexitätsmanagements

DÖRNER (2003, S. 58ff) beschreibt generelle Probleme im Umgang mit komplexen Systemen. Menschen, die mit komplexen Systemen interagieren müssen, scheitern oft daran, ihre Ziele zu erreichen. Die Gründe hierfür sind ungenügende Zielidentifikation, Beschränkung auf Ausschnitte eines Problems, einseitige Konzentration, unberücksichtigte Nebeneffekte, die Tendenz zum Übersteuern und zu autoritären Verhalten. Gerade im Umfeld des Systemarchitekten sind komplexe Aufgaben zu bewältigen. Daher wird methodische wie auch rechnergestützte Unterstützung benötigt.

## Strukturelles Komplexitätsmanagement

In der Wissenschaft weit verbreitete Ansätze, komplexe Sachverhalte zu modellieren, sind die Design Structure Matrix (DSM) und die Multi-Domain-Matrix (MDM) (BROWNING 2001; EPPINGER & BROWNING 2012; MAURER 2007; LINDEMANN ET AL. 2009). DSM wie MDM lassen sich problemlos in Graphform überführen (LINDEMANN ET AL. 2009). Die Elemente der Matrix bilden die Knoten des Graphen, die Verknüpfungen in der Matrix bilden die Kanten zwischen den jeweiligen Elementen der Zeilen und Spalten.

Die DSM wurde im ingenieurwissenschaftlichen Kontext erstmals bei STEWARD (1981) erwähnt. BROWNING (2001) und EPPINGER & BROWNING (2012) geben einen umfassenden Überblick über die Einsatzmöglichkeiten der DSM. In einer DSM können Elemente eines beliebigen Typs abgebildet werden, wie z. B. Komponenten, Prozessschritte, Dokumente oder Ähnliches.

LINDEMANN ET AL. (2009, S. 31ff) nutzen die matrixbasierte Modellierung als Werkzeug für das strukturelle Komplexitätsmanagement. Es umfasst die drei Strategien:

- Erfassen und Bewerten,
- Vermeiden und Reduzieren sowie
- Beherrschen und Kontrollieren von Komplexität.

Die Informationsbeschaffung ist jedoch sehr aufwendig (MAURER 2007, S. 94ff). Bei größeren Systemen mit mehreren hundert Elementen wird es schwierig, den Überblick zu behalten und die Handhabbarkeit der Matrizendarstellungen zu gewährleisten. Zudem erfordern die Modellierung und die Interpretation der Strukturkennzahlen im Kontext der Systemarchitektur eine hohe Expertise.

In der Regel kann nur eine Relationsart je DSM definiert werden, z. B. „... ist verknüpft mit...“. Ausnahmen finden sich z.B. bei PIMMLER & EPPINGER (1994), die vier Verknüpfungsarten (räumliche Verknüpfung, Material-, Informations- und Energiefluss) in Produkten mit einer DSM modellieren. Relationen werden durch ein Kreuz, eine „1“ oder einen beliebigen Zahlenwert in der Matrix abgebildet. Über einen Zahlenwert, kann die Stärke einer Relation zum Ausdruck gebracht werden; ein „X“ oder eine „1“ zeigen lediglich die Existenz einer Relation auf. Sind zwei Elemente durch eine gerichtete Kante verbunden, so kann das mit der gegebenen Lesart der Matrix nachvollzogen werden. Bei ungerichteten Kanten ist die Matrix symmetrisch. Abbildung 3-8 zeigt eine DSM am Beispiel einer Waschmaschine (HELLENBRAND 2013, S. 65).

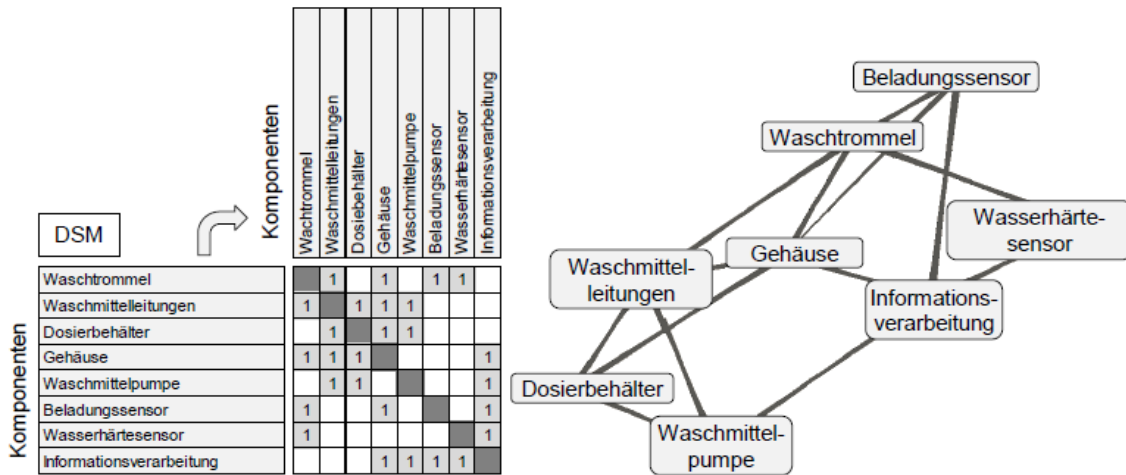


Abbildung 3-8: DSM am Beispiel einer Waschmaschine nach HELLENBRAND (2013, S. 65)

Die DSM wurde weiterentwickelt, so dass Elemente mehrerer Typen abgebildet werden können (MAURER 2007). In der Multi-Domain-Matrix werden Elemente jeweils eines Datentyps – in diesem Zusammenhang als *Domänen* bezeichnet – in Teilmatrizen zusammengefasst. Unterschiedliche Relationen innerhalb einer Domäne können in einzelnen DSMs abgebildet werden. Relationen zwischen Elementen unterschiedlicher Domänen werden in Domain Mapping Matrizen (DMM) angezeigt (DANILOVIC & BROWNING 2004). Den Aufbau einer MDM beschreibt MAURER (2007, S. 80) in Abbildung 3-9.

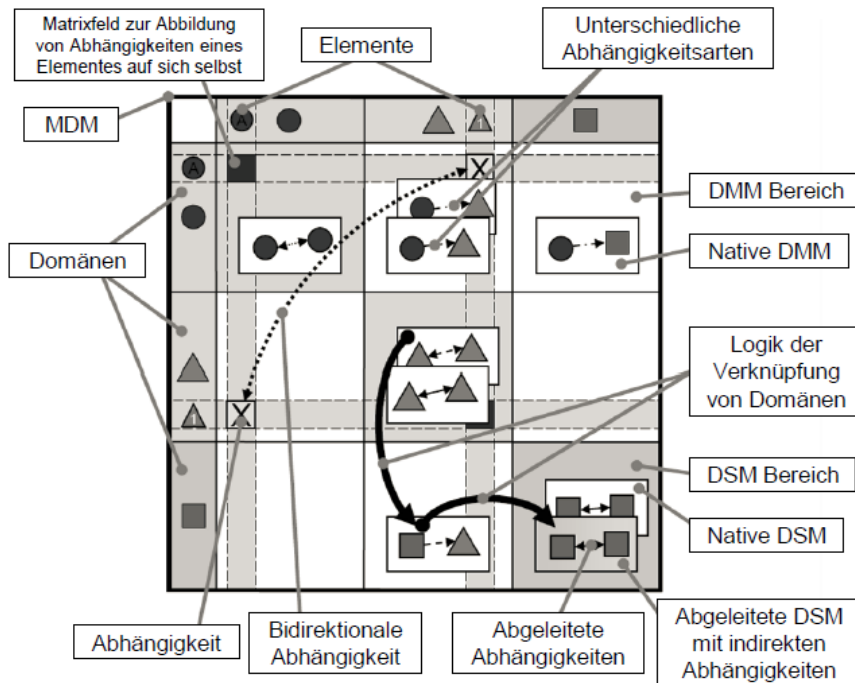


Abbildung 3-9: Aufbau einer MDM nach MAURER (2007, S.80)

Sowohl mit DSMs als auch mit MDMs können demnach typisierte Graphen beschrieben werden. Die Abbildung von komplexen Sachverhalten mittels MDM erlaubt die Nutzung graphentheoretischer Algorithmen. Somit kann auf ein großes Repertoire an Methoden der

Mathematik zurückgegriffen werden. Dies erlaubt Berechnungen von Strukturkennzahlen, Anwendung von Clusteralgorithmen, Ermittlung von Kreisschlüssen und andere Verfahren. Diese Verfahren können genutzt und auf unterschiedliche Problemstellungen im Ingenieurwesen übertragen werden.

Das Clustering kann beispielsweise für die Modularisierung von Produkten herangezogen werden (LINDEMANN ET AL. 2009, S. 171ff). Triangulierung einer DSM mit Prozessschritten kann ungewollte Kreisschlüsse aufzeigen (LINDEMANN ET AL. 2009, S. 138). Das Erkennen und Eliminieren von Kreisschlüssen kann auch zur Reduktion von Komplexität in Produktstrukturen genutzt werden (MAURER 2007). Weitere Anwendungen finden sich u.a. bei KREIMEYER & LINDEMANN (2011), die ein strukturbasiertes Messsystem für Produktentwicklungsprozesse mit Hilfe der MDM aufgebaut haben.

### Neuronale Netze

**Neuronale Netze** sind lernende Strukturen. „Neuronale Netze beziehen sich auf das Neuronennetz des menschlichen Gehirns. Dieses dient als Analogie und Inspiration für in Computern simulierte, künstliche neuronale Netze“<sup>15</sup> (vgl. REY & WENDER 2008). Sie können eingesetzt werden, wenn explizite Informationen zu einem Problem fehlen (MAURER 2007, S. 46). Neuronale Netze können für die evolutionäre Entwicklung eines Produktspektrums eingesetzt werden (PULM 2004, S. 177). KOHLHASE (1997) verwendet neuronale Netze für die Steuerung von Variantenvielfalt. Nach MAURER (2007, S. 46) eignen sie sich allerdings nicht dazu, ein besseres Verständnis über Abhängigkeiten in einem System und damit höhere Transparenz zu erlangen. Zudem erlauben sie keine direkte Nutzerinteraktion und eignen sich daher für die Unterstützung der Aufgaben in der Systemarchitektur nur bedingt. Für Einblicke in den Gebrauch neuronaler Netze im Ingenieurwesen wird auf KUSIAK (2000) verwiesen.

### Data Mining

Mustererkennung in großen Datenbeständen kann mit **Data Mining** Ansätzen erreicht werden (KUSIAK 2000). Es werden vorwiegend statistische Methoden eingesetzt zur Mustererkennung, Bildung von Clustern und Ähnlichkeitsanalysen. Das Ziel ist es nicht-triviale, implizite und vorher unbekannte Informationen und Muster in großen Datenbasen zu finden (FAYYAD ET AL. 1996). „Pattern Recognition“ im Sinne des Data Mining muss eher als „Muster entdecken“ aufgefasst werden. Daher eignen sich die Methoden im Kontext dieser Fragestellung nicht, da hier explizit bekannte Muster und Fehler gefunden werden sollen. Dennoch bergen generell die Methoden des Data Mining großes Potential, auch in der Anwendung innerhalb der Systemarchitektur. Die Anwendungsfelder erstrecken sich auf zahlreiche Disziplinen, wie auch im Ingenieurwesen. KÜSTNER ET AL. (2013) setzen beispielsweise Data Mining Methoden für die Optimierung von Windkraftanlagen ein. Weitere Informationen zum Data Mining finden sich bei FAYYAD ET AL. (1996), HAN ET AL. (2012) und WITTEN ET AL. (2011).

---

<sup>15</sup> <http://www.neuronalesnetz.de/einleitung.html> (entnommen am 02.12.13)

### 3.3 Herausforderungen im Umgang mit komplexen Produktportfolios

Bei einer klaren und durchgängigen Strukturierung eines Produktportfolios anhand der Systematiken, die im vorherigen Abschnitt vorgestellt wurden, kann ein hohes Maß an Transparenz erreicht werden. Die Zerlegung komplexer Systeme in Teilsysteme, die Ordnung von Bausteinen und Schnittstellen nach definierten Kriterien und die Vorgabe von sinnvollen Gestaltungsprinzipien führen zu einem besseren Überblick, einfacheren Verständlichkeit und schließlich zu einer Komplexitätsreduzierung in der Systemarchitektur. Der Einsatz der oben beschriebenen Methoden wirkt sich positiv auf die Zielerreichung des Systemarchitekten aus, wenn die nötigen Voraussetzungen für ihre Anwendung geschaffen werden können. In der Praxis sind jedoch gewachsene Strukturen, Ausnahmen, Sonderwege und Insellösungen weit verbreitet. Eine Restrukturierung oder ein Methodeneinsatz nach oben genannten Gesichtspunkten sind dann nicht mehr ohne Weiteres möglich.

PULM (2004, S. 136f) schließt aus seiner umfangreichen Recherche zu Methoden des Komplexitätsmanagements, dass es ein Überangebot an Methoden gibt. Dennoch seien viele Ansätze zum Teil unangemessen, da sie unter anderem:

- zu sehr auf eine reine Variantenreduzierung abzielen,
- keinen durchgängigen Prozess mit entsprechender Methodenunterstützung bieten,
- der realen, zahlenmäßigen Komplexität von Produktspektren nicht gerecht werden,
- zwar mathematisch korrekt, jedoch praktisch nicht anwendbar sind und
- die Produktstruktur und -architektur zu wenig berücksichtigen.

Um die Methoden zur Reduzierung der Variantenvielfalt einsetzen zu können, ist ein effektives und effizientes Arbeiten mit komplexen Strukturen und eine aufwandsarme Informationsbeschaffung und Aufbereitung erforderlich. Die Informationsbeschaffung und –aufbereitung werden in den Methoden oft vorausgesetzt. Da die benötigten Informationen in der geforderten Form meist nicht in Unternehmen verfügbar sind und der Aufwand der Erhebung als zu hoch empfunden wird, scheitert der Methodeneinsatz.

Ein Eingreifen in die Produktstruktur erfordert Überblick über die Wechselwirkungen mit Stakeholdern und Abhängigkeiten innerhalb des Produktportfolios. Bevor über Maßnahmen entschieden wird, müssen kritische Relationen ermittelt und geprüft werden. Ansonsten besteht die Gefahr, dass jene geplanten Veränderungen ungewollte Auswirkungen haben und somit höhere Kosten verursachen oder womöglich einen gegenteiligen Effekt erzielen.

Die Instrumentalziele des Systemarchitekten (vgl. Kapitel 2.2.3) stellen angesichts der steigenden Produktvielfalt eine große Herausforderung dar. In diesem Kapitel werden Komplexitätstreiber im Produktportfolio analysiert und gezeigt, inwiefern sich Komplexität als Hürde der hier aufgezählten Ziele darstellt.

#### 3.3.1 Komplexität des Produktportfolios

Die Komplexität eines Produktportfolios ist zentraler Grund dafür, dass die oben genannten Ziele meist nur mit viel Aufwand und unter großer Ungewissheit verfolgt werden können. Sie

entsteht sowohl durch eine hohe **Varietät** und **Konnektivität** als auch durch starke **Wechselwirkungen** mit dem Umfeld durch dessen Einbettung in die Unternehmensprozesse. MAURER (2007, S. 3ff) beschreibt das Zusammenwirken der vier Komplexitätsfelder Markt, Produkt, Prozess und Organisation. Mit dem Produktportfolio als zentrales Betrachtungsobjekt dieser Arbeit, werden hier Einflussfaktoren aus Markt, Prozess und Organisation als **Umfeldfaktoren** bezeichnet. Nachfolgende Abbildung 3-10 gibt einen Überblick über die Faktoren der Komplexitätssteigerung in Produktportfolios, die in diesem Kapitel diskutiert werden.



Abbildung 3-10: Einflussfaktoren auf die Komplexität des Produktportfolios

## Auswirkungen von Komplexitätssteigerungen im Produktportfolio

Die Vielfalt in einem Produktportfolio führt zu einem Anstieg der Komplexität (SCHUH 2005). Dies hat einige **positive Wirkungen**, aber überwiegend **negative Auswirkungen** auf die Wettbewerbsfähigkeit eines Unternehmens (RENNER 2007, S. 28ff).

Komplexität muss nicht grundsätzlich negativ sein, sondern die erforderlichen Ansätze zum Umgang mit ihr müssen zielgerichtet eingesetzt werden (MAURER 2007, S. 8). Gerade die Differenzierung (HEINA 1999, S. 1) begünstigt die Erschließung von neuen Märkten und Kundensegmenten. Ein höherer Grad an beherrschter Komplexität kann sich als Wettbewerbsvorteil erweisen, wenn er als Markteintrittsbarriere für Konkurrenten fungiert (KAISER 1995, S. 102). Ein weiterer **Vorteil** liegt in der Flexibilisierung der Produktstruktur. Wenn ein bestimmtes Komplexitätsniveau vorliegt, kann damit schneller auf Änderungen am Markt reagiert werden (MAURER 2007, S. 40). (RENNER 2007, S. 28f) fasst die positiven Folgen der Variantenvielfalt zusammen: Die Steigerung des Kundennutzens, die Möglichkeit der Differenzierung im Markt und von Wettbewerbern sowie die Nutzung von Verbundeffekten sind positive Effekte einer richtig gewählten Vielfalt.

Auf der anderen Seite ergeben sich **zahlreiche Nachteile**: die Komplexitätskosten und der Aufwand für die Qualitätssicherung steigen überproportional an und Substitutionseffekte am Markt stellen sich ein (RENNER 2007, S. 29ff). Vielfaltsinduzierte Deckungsbeiträge steigen hingegen nur degressiv an (EBD.). PILLER & WARINGER (1999, S. 14) zählen Probleme verursacht durch Variantenvielfalt und deren Folgen für Unternehmen auf. Mehr Komplexität im Produktportfolio eines Unternehmens verursacht höhere direkte Kosten in der Entwicklung, Einkauf, Produktion, Vertrieb und Marketing sowie beim Kundendienst (BULLINGER ET AL. 2002, S. 390). Eine höhere Variantenzahl führt zu höheren Kostenfaktoren im Produktlebenszyklus durch Logistikkostenaufwand, Entwicklungsaufwand, Steuerungs- und Pflegeaufwand in der Serienproduktion, Skalenerlöse durch Varianten, Anlaufprobleme beim Serienstart und Werkzeugkosten (RENNER 2007, S. 31). Die Entwicklungsressourcen müssen sich bei steigender Variantenvielfalt auf ein breiteres Produktspektrum verteilen (EBD.). Dies kann zu höherem Aufwand für die Qualitätssicherung und bei gleichbleibenden Entwicklungskapazitäten zu einer höheren Wahrscheinlichkeit von Qualitätsproblemen führen.

Die Fertigungsverfahren werden aufwendiger, die Prozesse müssen leistungsfähiger ausgestaltet werden. In HOMBURG & DAUM (1997, S. 333) werden ergänzend Aufwendungen für Kundendienst, Schulungen, Qualitätssicherungsmaßnahmen, variantenspezifische Lagerbestände, Pflege und Verwaltung von Produkt- und Preisdokumentation aufgeführt. Als Opportunitätskosten werden zusätzlich Verluste durch Kannibalisierungseffekte unter den Produkten einer Produktfamilie genannt.

Darüber hinaus wird die Marktmacht beeinträchtigt durch geringere Effektivität des Vertriebssystems, abnehmende Aktions- und Reaktionsgeschwindigkeit, Überlastung der Entwicklungsressourcen, Timing der Produktlebensphasen und Produktkannibalismus im eigenen Produktprogramm (SCHUH 2005, S. 21).



JESCHKE (1997, S. 7) beschreibt einen *Circulus vitiosus*<sup>16</sup>: Stagnierende Absätze führen zu Diversifizierung und Bedienung von Nischenmärkten. Dies wiederum führt zu einer Veränderung der Kostenstruktur und zu höheren Gemeinkostenanteilen. Die Folge ist, dass die Wettbewerbsfähigkeit des Unternehmens sinkt und sich die Spirale mit wiederum stagnierenden Absätzen verschärft.

### **Komplexitätstreiber im Umfeld des Produktportfolios**

Eine Reihe von **Umweltfaktoren** beeinflusst das Produktportfolio, die zu einer Steigerung der Komplexität für die Systemarchitekten bei der Planung und Kommunikation in den Unternehmensprozessen führen. Nach ZICH (1996, S. 153) zeichnen sich erfolgreiche Unternehmen dadurch aus, dass Umweltfaktoren, Unternehmensstrategie und interne Konfiguration in Übereinstimmung gebracht werden.

Die Systemarchitektur als Teil des Entwicklungsprozesses ist in ein **komplexes Entwicklungsumfeld** eingebettet. Nach der Definition von PATZAK (1982) in Kapitel 2.1.2 ist in diesem Zusammenhang mit Vielfalt sowohl die Varietät und Konnektivität von Elementen wie auch von Relationen gemeint. Neben Produkten und deren Kompositionsstufen können beispielsweise auch weitere Domänen wie Anforderungen, Funktionen, Technologien, Prozesse, Produktionsressourcen oder auch Lieferanten in einem komplexen Netzwerk von Abhängigkeiten und Wechselwirkungen zusammengefasst werden (LINDEMANN ET AL. 2009). In solch einem Netzwerk können eine hohe Anzahl an Relationen und viele unterschiedliche Relationsarten auftreten, wie z.B. „beinhaltet“, „erfüllt“ oder „liefert“.

Systemarchitekten nehmen eine **zentrale Rolle im Produktentwicklungsprozess** ein. Sie betreuen oder unterstützen von der Übersetzung der Anforderungen in technische Spezifikationen bis hin zur Definition und Beauftragung einzelner Entwicklungsumfänge. Außerdem müssen sie durch Definition von Integrationsstrategien und Testverfahren die Erfüllung der Anforderungen sicherstellen (HASKINS ET AL. 2010, S. 94). Sie müssen sich daher mit Marketing und Vertrieb, Fachabteilungen der Entwicklung, Prototypenbau, Integration und Test sowie Qualitätssicherung eng abstimmen und deren Anforderungen berücksichtigen. Somit entstehen zusätzlich zahlreiche Wechselwirkungen mit Stakeholdern.

Komplexität entsteht auch in Unternehmensprozessen (KREIMEYER & LINDEMANN 2011). Die Abstimmung mit unterschiedlichen Fachdisziplinen im Unternehmen erfordert die Berücksichtigung unterschiedlicher Erfahrungshintergründe und damit unterschiedlicher **Kommunikationspfade und Organisationsabläufe** (KREIMEYER ET AL. 2006). Eine große Anzahl organisatorischer und technischer Abhängigkeiten ist auch durch die Integration mechatronischer Komponenten bedingt. Grund dafür ist, dass mechatronische Entwicklungen eine intensive Zusammenarbeit zwischen unterschiedlichen Fachdisziplinen und entsprechenden Koordinationsaufwand erfordern (HELLENBRAND 2013, S. 21f).

In einigen Branchen, wie der Automobilindustrie, ist die Wertschöpfungstiefe der OEMs stark reduziert worden. Immer größere Umfänge werden in einem **Netzwerk von Lieferanten** entwickelt und geliefert. Dies erfordert intensive Kooperationen mit externen Lieferanten

---

<sup>16</sup> (lat.) Teufelskreis, „Komplexitätsfalle“ bei RENNER 2007, S. 31

(KEIJZER & KREIMEYER 2006). Je nach Marktposition des Zulieferers können sich hierbei unterschiedlich große Freiheitsgrade bei der Produktgestaltung ergeben. Wenn beispielsweise nur kleine Stückzahlen abgerufen werden können, werden sich Sonderentwicklungen für den Lieferanten nur selten wirtschaftlich darstellen lassen. Auch die Integration günstiger Standardkomponenten („components-off-the-shelf“) kann dazu führen, dass dadurch entstandene Randbedingungen wie Schnittstellen, Geometrie oder Gewicht im Produktportfoliomanagement berücksichtigt werden müssen. Auch eine Abkündigung von Braugruppen durch einen Lieferanten kann erheblichen Aufwand für Substitutionsmaßnahmen bedeuten.

Ein weiterer Faktor, der die Komplexität erhöht, sind zahlreiche, zeitlich versetzte, **parallel laufende Entwicklungsprojekte**, zwischen denen möglichst Synergien erzielt werden sollen. DÖRNER<sup>17</sup> zeigte auf, dass viele Führungskräfte gar nicht beziffern könnten, wie viele Projekte im Unternehmen gleichzeitig laufen. Die Koordination, wann welche Bausteine mit welchem Reifegrad in welchem Projekt Verwendung finden, kann rasch zu einer dynamischen, komplexen Aufgabe werden. Um den Anforderungen realer Entwicklungssituationen in Unternehmen gerecht zu werden, müssen die Methoden des Systems Engineering an eine Multi-Projekt-Situation angepasst werden. Die Herausforderung besteht darin, die Unternehmensziele im Gesamtkontext zu verstehen und in den oft zahlreichen Entwicklungsprojekten zu vertreten, auch wenn dies zu scheinbaren Nachteilen in einzelnen Projekten führt. Hier kann z. B. ein Spannungsfeld zwischen Innovation und Differenzierung durch Neuentwicklungen auf der einen Seite und ressourcenschonendes Wiederverwenden von Bausteinen und gezieltes Variantenmanagement auf der anderen Seite entstehen. Die Bereitstellung der individuellen Sichtweisen auf das Produktportfolio hilft dabei, konkurrierende Interessen auszugleichen.

In der ISO-Norm 15288 (vgl. IEEE 2005) werden die Prozesse des Systems Engineering idealisiert mit Fokus auf der Entwicklung *eines* spezifischen Systems verstanden. Auch in den Richtlinien der INCOSE oder der NASA wird stets von einer Neuentwicklung eines Systems ausgegangen. Dies trifft allerdings nicht die Realität beispielsweise eines Automobilbauers oder eines Herstellers von Bohrmaschinen. In der Unternehmensrealität zielen Entwicklungsprojekte nicht ausschließlich auf grundlegende Neuentwicklungen ab, sondern dienen der **Weiterentwicklung, Produktpflege** oder **Anpassung**. Produktplanung, Innovations- und Technologiemanagement beeinflussen die Dynamik im Produktportfolio durch geplante Änderungen („initiated changes“), die mit genügend Vorlauf eingeplant werden müssen. Aber auch ungeplante Änderungen („emergent changes“) können auftreten, die entsprechend antizipiert werden müssen (JARRATT ET AL. 2011, S. 108).

Methoden in der Architekturentwicklung richten sich unter anderem auf die Ermöglichung einer **lebenszyklusgerechten Produktgestaltung**. Die Architektur wird derart angelegt, so dass die Produkte bestimmte „*lifecycle properties*“ aufweisen. Das Kunstwort „*Ilities*“, das ein Sammelbegriff für jene Eigenschaften eines Produktes ist, entstammt der englischen Wortendung *-ility*. Die „klassischen Ilities“ sind *safety, quality, usability* und *reliability*. (DE WECK ET AL. 2011, S. 68) Weitere Lebenszykluseigenschaften, die verstärkt Systeme aufweisen sollen und entsprechend vermehrt in der Literatur zu finden sind, beschreiben DE WECK ET AL.

---

<sup>17</sup> Vortrag auf den Klausurtagen des Lehrstuhls für Produktentwicklung 2013, Behringersmühle

(2012). Vor allem änderungsrelevante Eigenschaften wie *flexibility*, *robustness*, *scalability* und *adaptability* werden demnach prominenter. Die Herausforderung einer änderungsgerechten Produktgestaltung besteht darin, kosten- und wertgerechte Optionen bereits auf Architekturebene zu schaffen (SCHRIEVERHOFF ET AL. 2012). Entsprechende Entwicklungsmethoden zum Design for Changeability<sup>18</sup> finden sich bei GU ET AL. (2009); ENGEL & BROWNING (2008); HASHEMIAN (2005); KISSEL ET AL. (2012); ROSS ET AL. (2008); LI ET AL. (2008); FRICKE & SCHULZ (2005); LI ET AL. (2013) und anderen. Eine Modularisierungsstrategie zur Berücksichtigung von *Design for Adaptability*-Aspekten haben GREISEL ET AL. (2013) vorgestellt.

Neben den Umfeldfaktoren, die das Arbeiten mit einem Produktportfolio zu einer komplexen Aufgabe machen, ist die Varianten- und Teilevielfalt ein entscheidender Faktor, der zur Komplexität des Produktportfolios selbst beiträgt.

### Komplexität der Produktportfolios durch Variantenvielfalt

Als maßgeblicher Treiber für Komplexität innerhalb des Produktportfolios wird die **Varianten- und Teilevielfalt** gesehen. Das Variantenmanagement kann als eine Teilmenge des Komplexitätsmanagements angesehen werden (FIRCHAU 2003, S. 12). EHRENSPIEL & MEERKAMM (2013, S. 716) beschreiben die steigende Teilevielfalt mit einem selbstverstärkenden Kreis aus Produkt- und Sortiments-Vielfalt, Kundenvielfalt, Auftragsvielfalt und Lieferantenvielfalt. All jene Aspekte der Vielfalt führen gemeinsam zu hoher Komplexität, hohen Kosten und weniger Gewinn. Eine Steigerung der Teile- und Produktvielfalt erhöht die Komplexität der Produkte selbst wie auch die des gesamten Produktportfolios. Eine Reduzierung der Produktpalette um unwirtschaftliche Varianten sei zwingend erforderlich (EHRENSPIEL & MEERKAMM 2013, S. 714). Weiterhin zitieren sie WILDEMANN (1997), der nachgewiesen hat, dass eine Verdoppelung der Varianten eine Kostensteigerung von 20 - 30% nach sich zieht (vgl. auch WILDEMANN 2013, S. 8). Die Kostensteigerung könne auch darauf zurückgeführt werden, dass sich bei Variantenzuwachs und Losgrößenverringern die direkten wie indirekten Kosten erhöhen und sich die Durchlauf- und Lieferzeiten verlängern.

Zahlreiche Autoren<sup>19</sup> beschreiben **Ursachen für die Variantenvielfalt** in Produktportfolios. Auf der einen Seite tragen **unternehmensexterne Ursachen** wie Kunde, Markt, Wettbewerb oder externe Rahmenbedingungen zur vielfaltsinduzierten Komplexitätssteigerung bei. Auf der anderen Seite gibt es **interne Ursachen**, wie Fehler in der Unternehmensstrategie, im internen Controlling oder in der Organisation. Hinzu kommen übergeordnete Ursachen innerhalb der

---

<sup>18</sup> In dem Themenfeld finden sich Methoden zu den Schlagwörtern Design for Changeability, Flexibility, Agility, Scalability, Adaptability und Robustness. Als stellvertretender Oberbegriff wird hier daher Design for Changeability gewählt.

<sup>19</sup> BIEGERT 1971, S. 70ff; SCHUH 1988, S. 8ff; CAESAR 1991, S. 9ff; KOHLHASE 1997, S. 1f; JESCHKE 1997, S. 7ff; PILLER & WARINGER 1999, S. 5ff; FIRCHAU 2003, S. 4ff; KARTIKA 2010, S. 6; BRAUN ET AL. 2013, S. 29; EHRENSPIEL & MEERKAMM 2013, S. 715f u.v.m.

Unternehmen, wie methodische Defizite, Defizite hinsichtlich Transparenz und Informationsmanagement, fehlende Gestaltungsprinzipien und generelle Zeitproblematik.

Tabelle 3-5 gibt einen Überblick über **externe Ursachen** und Gründe für steigende Variantenvielfalt und damit verbundener steigender Komplexität. Die genannten externen Faktoren lassen sich durch die Systemarchitektur nicht aktiv beeinflussen, sondern müssen entsprechend als **Planungsvorgaben** berücksichtigt werden. Es ist auch nicht das Ziel, extern induzierte Vielfalt zu reduzieren. Vielmehr sollte versucht werden, die Marktposition durch Beherrschung der Komplexität zu verbessern. Vielfalt, die durch eine Veränderung der Nachfrage des Kunden entsteht, muss mit den Produkt- und Organisationsstrukturen des Unternehmens entsprechend effizient und flexibel dargestellt werden können.

	Cluster		Gründe für Variantenvielfalt	Quellen
Externe Ursachen	Kunden	Kunde	<ul style="list-style-type: none"> <li>▪ Zunehmende Diversifizierung bzw. Spreizung der Kundenwünsche</li> <li>▪ Zunehmend individueller werdende Kundenwünsche bzgl. Ausstattung und technischer Merkmale</li> <li>▪ Steigende Kundenanforderungen durch gut informierte und anspruchsvolle Kunden</li> </ul>	KVIST (2010, S. 10) PINE, II. (1993) PILLER (1998) FIRCHAU ET AL. (2002, S. 5)
		Gesellschaft	<ul style="list-style-type: none"> <li>▪ Änderung demographischer Strukturen (Alterung der Gesellschaft)</li> <li>▪ Verschiebung des Kaufkraftaufkommens</li> </ul>	FIRCHAU ET AL. (2002, S. 5)
	Markt und Wettbewerber	Internationalisierung der Märkte	<ul style="list-style-type: none"> <li>▪ Länderspezifische Auflagen (Normen, Vorschriften und Richtlinien)</li> <li>▪ Länderspezifische Anforderungen</li> <li>▪ Länderspezifische Wertauffassungen bzgl. Design, Handhabung, Ausstattung, Qualitätserwartung und Kosten</li> </ul>	BRAUN ET AL. (2013); EHRENSPIEL & MEERKAMM (2013, S. 49)
		Marktkomplexität und Wettbewerber	<ul style="list-style-type: none"> <li>▪ Erhöhte Wettbewerbsintensität und -dynamik durch Eintritt neuer Wettbewerber bzw. durch Produktpflegeprogramme existierender Wettbewerber</li> <li>▪ Sättigung traditioneller Märkte</li> <li>▪ Risikostreuung und Ausgleich von Marktschwankungen (Marktsegmentierung)</li> </ul>	WILDEMANN (2013) SCHMIDT (2009, S. 91ff)
	Externe Rahmenbedingungen	Gesetze und Normen	<ul style="list-style-type: none"> <li>▪ Zunehmend relevante internationale Gesetze und Normen</li> <li>▪ Abgas/Emission</li> <li>▪ Sicherheitsanforderungen</li> </ul>	EHRENSPIEL & MEERKAMM (2013, S. 49) FIRCHAU ET AL. (2002, S. 5)
		Technologischer Fortschritt	<ul style="list-style-type: none"> <li>▪ Verkürzung der Produktlebenszyklen (taktische Zyklen)</li> <li>▪ Schnelleres Entstehen neuer und verbesserter Technologien (strategische Zyklen)</li> <li>▪ Unterschiedlich geforderter Automatisierungsgrad durch Produktion in verschiedenen Ländern</li> </ul>	FIRCHAU ET AL. (2002, S. 5)

Tabelle 3-5: Externe Ursachen der Variantenvielfalt (in Anlehnung an BRAUN ET AL. 2013)

Auch eine Steigerung der Vielfalt, die ein Unternehmen durch Erschließung neuer Märkte oder zur Differenzierung vom Wettbewerb eingeht, gilt es effizient zu beherrschen. Gesetze, Normen sowie technologischer Fortschritt führen immer wieder zu Änderungen, die Anpassungen und Pflege des Produktbestandes erforderlich machen. Auch hier gilt es für den Systemarchitekten, die nötigen **Voraussetzungen in der Produktportfolioarchitektur zu schaffen**, um diese Änderungen kostengünstig und schnell umzusetzen.

Größeren Einfluss kann die Systemarchitektur auf **unternehmensinterne Faktoren**, die zur **Steigerung der Vielfalt** beitragen, nehmen. Tabelle 3-6 gibt einen Überblick über interne Faktoren und Gründe:

Wenn dem Thema Systemarchitektur und kontinuierliche Pflege des Produktportfolios zu wenig Beachtung in der **Unternehmensstrategie** geschenkt wird, kann ein gezieltes Variantenmanagement nicht durchgeführt werden (BRAUN ET AL. 2013). Die Folge ist, dass der Teilstamm über die Jahre wächst und unnötig Kosten erzeugt werden. Auch wenn kein aktives Variantenmanagement betrieben wird, können ohne den nötigen Überblick und entsprechender Bewertung auch defizitäre Varianten erzeugt werden.

Die **wirtschaftliche Situation** eines Unternehmens kann es dazu zwingen, sich immer mehr zu spezialisieren und kleine Nischenmärkte zu besetzen (JESCHKE 1997, S. 7). Dies führt dann zu einer größeren Diversifikation der Produktpalette und zur Verkleinerung der Stückzahlen pro Erzeugnis. Es können auch nicht immer ausschließlich Baukastenelemente für alle Varianten verwendet werden. Durch die Besetzung von Nischenmärkten entstehen eine Vielzahl an Anpass- und Sonderbausteinen (PAHL ET AL. 2007, S. 664) und somit auch eine höhere Komplexität im Portfolio.

Oft fehlt in Unternehmen die grundlegende Transparenz über die Erlösstruktur von Varianten, da nur selten **verursachungsgerechte Kostenrechnungsverfahren** eingesetzt werden (FIRCHAU ET AL. 2002, S. 5). Auch KVIST (2010, S. 10) sieht in dem Mangel an Kostenbewertungsmethoden einen Grund für steigende Komplexität. Das Bewusstsein für die Kosten einer zusätzlichen Variante fehlt dadurch. Auch das Einsparpotential durch die Verwendung von Übernahmeteilen und Synergieeffekte durch den Baukasten können somit nicht quantifiziert werden.

Hinzu kommen **organisatorische Gründe**, die zu einer Zunahme der Variantenvielfalt führen (FIRCHAU ET AL. 2002, S. 5). Oft wird die Rolle des Systemarchitekten im Unternehmen entweder gar nicht besetzt oder kommissarisch von Projektleitern oder Ähnlichen eingenommen. Dadurch gibt es oft keine klaren Verantwortlichen für Variantenmanagement im Unternehmen, die somit in den Entwicklungsprozessen fehlen, um entsprechende Entscheidungen zu treffen (vgl. FIORE 2005; KVIST 2010). Aus Ermangelung einer koordinierenden Übersichtsfunktion, werden „Fehler“ im Sinne des Gesamtsystems Unternehmen gemacht, die im Kontext einer einzelnen Produktentwicklung nicht als „Fehler“ gesehen werden. Mögliche Synergien mit anderen Projekten und Produkten bleiben unerkannt. Durch Projektfokussierung geht der Blick für das Gesamte verloren. Das führt dazu, dass Investitionen und Kosten im Kontext des Projektes und nicht im Kontext des Gesamtunternehmens bewertet werden.

	Cluster		Gründe für Variantenvielfalt	Quellen
Interne Ursachen	Unternehmensstrategie	Defizite in der Unternehmensstrategie	<ul style="list-style-type: none"> <li>▪ Historisch gewachsenes Produktspektrum ohne Reduktion des Teilestamms</li> <li>▪ Fehlende Planung und Bewertung von Varianten</li> <li>▪ Strategie der Produktindividualisierung</li> </ul>	Braun et al. (2013), EHRENSPIEL & MEERKAMM (2013, S. 715)
	Finanzen und Controlling	Wirtschaftliche Situation	<ul style="list-style-type: none"> <li>▪ Zwang der Bedienung von Nischenmärkten aufgrund von stagnierendem Absatz und Umsatz</li> </ul>	JESCHKE (1997, S. 7)
		Transparenz Erlösstruktur	<ul style="list-style-type: none"> <li>▪ Keine verursachungsgerechte Kostenrechnung</li> <li>▪ Keine Transparenz der Erlösstruktur von Varianten</li> </ul>	KVIST (2010, S. 10), FIRCHAU ET AL. (2002, S. 5)
	Organisation	Organisatorische Defizite	<ul style="list-style-type: none"> <li>▪ Mangel an Koordination, Synchronisation, Kommunikation und Zusammenarbeit innerhalb und zwischen den einzelnen Unternehmensbereichen</li> <li>▪ Ungeeignete Entscheidungsstruktur zur Variantenproblematik</li> <li>▪ Mangelhafte Klarheit von Verantwortlichen innerhalb der Prozessketten</li> <li>▪ Fehlende Sensibilisierung bei den Mitarbeitern hinsichtlich der Auswirkungen der Variantenvielfalt</li> </ul>	EHRENSPIEL & MEERKAMM (2013, S. 715), FIRCHAU ET AL. (2002, S. 5), FIORE (2005), KVIST (2010, S. 10)
	Übergeordnete Ursachen	Methodische Defizite	<ul style="list-style-type: none"> <li>▪ Nicht etabliertes, fehlendes bzw. unvollständiges Bewertungssystem zur Bewertung alternativer Möglichkeiten</li> <li>▪ Kein Werkzeug für eine markt- und kostengerechte Portfolioplanung</li> </ul>	FIRCHAU ET AL. (2002, S. 5), Renner (2007) PULM (2004, S. 177)
		Transparenz- und Informationsdefizite	<ul style="list-style-type: none"> <li>▪ Ungenügende oder nicht systematische Informationen zur Erkennung und Beschreibung der aktuellen Variantenvielfalt</li> <li>▪ Kein Suchsystem für vorhandene Lösungen und Abhängigkeiten</li> <li>▪ Unübersichtlichkeit der in vielen Inselsystemen vorhandenen Produktdaten aufgrund einer historisch gewachsenen IT-Infrastruktur</li> <li>▪ Mangelnde Datenhaltungsdisziplin</li> </ul>	KVIST (2010); RENNER (2007), EHRENSPIEL & MEERKAMM (2013, S. 715)
		Gestaltungsregeldefizite	<ul style="list-style-type: none"> <li>▪ Fehlende Gestaltungsregeln zur Variantenvermeidung und -reduzierung</li> <li>▪ Mangelnde Definition von Architekturstandards</li> </ul>	RENNER (2007), RAPP (2010, S. 155)
		Zeitproblematik	<ul style="list-style-type: none"> <li>▪ Zeitproblematik bei zeitkritischen Entwicklungsprojekten (keine Möglichkeit einer variantengerechten Planung aufgrund von hohem Zeitdruck)</li> </ul>	KÖNIGS (2013, S. 3),

Tabelle 3-6: Unternehmens-interne Ursachen für Variantenentstehung (in Anlehnung an BRAUN ET AL. 2013)

Ohne eine koordinierende Querschnittsfunktion wie die des Systemarchitekten kann die Kommunikation und Zusammenarbeit zwischen den Unternehmensbereichen durch „Silo-Bildung“ abnehmen (EHRENSPIEL & MEERKAMM 2013, S. 715). Dadurch bleiben Potentiale für Synergien ungenutzt. Wenn zudem Mitarbeiter nicht ausreichend sensibilisiert wurden für die Auswirkungen der Variantenvielfalt, kommt es unter Umständen zu unnötigen Neuentwicklungen. Beispielsweise werden (ähnliche) Lösungen neu entwickelt, weil die Suche nach vorhandenen Lösungen nicht in Betracht gezogen wird oder zu aufwendig erscheint. Zudem kann ein Mangel an Koordination und Kommunikation zu kostenintensiven Änderungen führen, wenn bei der Entwicklung von Teilumfängen Wechselwirkungen mit anderen Modulen, Fachbereichen, Funktionen außer Acht gelassen werden.

Trotz zahlreicher Ansätze in Wissenschaft und Praxis zur Beherrschung der Variantenvielfalt bestehen verbreitet noch immer **methodische Defizite**. Effiziente Bewertungswerkzeuge sind nicht etabliert, fehlen oder werden nicht richtig eingesetzt (PULM 2004, S. 177; FIRCHAU 2003, S. 5). RENNER beschreibt in diesem Zusammenhang, dass heute etablierte Ziel- und Steuerungssysteme in der Produkt- und Prozessentwicklung den adäquaten Umgang mit der Variantenvielfalt nicht beherrschen und bewertet die Methoden als unzureichend für eine Bewertung alternativer Handlungsoptionen zur Reduzierung vorhandener Komplexität. Durch konträre Zielgrößen der Konstruktion, des Designs oder des Controlling entstehen Zielkonflikte, die nur durch spezifische konstruktive Lösungen behoben werden. Dadurch kommt es häufig zu einer Komplexitätssteigerung (RENNER 2007, S. 47f). Ein Mangel an Gestaltungswerkzeugen und -hilfsmitteln macht sich negativ im Entwicklungsprozess bemerkbar (EHRENSPIEL & MEERKAMM 2013, S. 717f).

Ursächlich für die Entstehung von Variantenvielfalt sind auch **Transparenz- und Informationsdefizite** zu sehen. Information zur Erkennung und Beschreibung der aktuellen Variantenvielfalt sind entweder nicht vorhanden oder in ungeeigneter Form abgelegt (RENNER 2007, S. 47f). Effiziente Suchsysteme für vorhandene Lösungen und relevante Abhängigkeiten fehlen oft oder stehen dem betroffenen Mitarbeiter in der Produktplanung, Entwicklung, Konstruktion, Vertrieb und anderen Abteilungen nicht zur Verfügung (EHRENSPIEL & MEERKAMM 2013, S. 716). Wenn Zugriff besteht, liegen die Daten meist in Form von unübersichtlichen Datenbankexporten vor oder müssen aufwendig durch die Befragung von Experten oder Suche in lokalen Projektdokumentationen („Inselsysteme“) zusammengetragen werden. Der Umstand, dass unterschiedliche, aber teilweise inhaltlich sich überschneidende Informationen an unterschiedlichen Stellen gespeichert und gepflegt werden, führt zu erheblichen Pflegeaufwand und teilweise somit zu Inkonsistenz in der Datenhaltung. Große Datenmengen, die über die Zeit nur gewachsen und nie bereinigt wurden, erschweren zusätzlich die Suche und verursachen unnötige Kosten. KVIST (2010, S. 10) macht mangelnde Datenhaltungsdisziplin dafür verantwortlich. Diese wird angesichts höherer Prioritäten im Tagesgeschäft oft vernachlässigt (KRATOCHVIL & CARSON 2005; FIORE 2005).

RENNER (2007, S. 47f) führt an, dass **fehlende Gestaltungsregeln** zur Variantenvermeidung und -reduzierung als weitere Ursache für eine Komplexitätssteigerung angesehen werden kann. Wenn entsprechende Architekturregeln und -standards fehlen, werden beispielsweise Schnittstellen nicht angemessen hinsichtlich der Konfigurierbarkeit von Bausteinen entwickelt. Die Wiederverwendbarkeit einer Komponente sinkt dadurch. Oder die Chancen einer

marktgerechten Reihung unterschiedlicher Leistungsklassen (PAHL ET AL. 2007; GERHARD 1984) bleiben ungenutzt. Dies führt zu einer unnötigen, unwirtschaftlicheren Vergrößerung der Produktpalette. RAPP sieht in der Aufweichung der Gestaltungsrichtlinien und Schnittstellendefinitionen ein ernstzunehmendes Problem. Durch ungeeignetes oder fehlendes Produktstrukturmanagement entferne man sich immer weiter vom Idealzustand einer durchdachten Produktstrukturierung (RAPP 2010, S. 155).

Bei zeitkritischen Entwicklungsprojekten ergibt sich zudem eine **Zeitproblematik** (KÖNIGS 2013, S. 3). Angesichts mangelnder Werkzeuge und Methoden lässt dies in der Regel keine Möglichkeit einer variantengerechten Planung zu. Die Entscheidungen müssen dann entsprechend unter hohem Zeitdruck „aus dem Bauch heraus“ getroffen werden, ohne eine rationale Analyse und Bewertung vorab durchführen zu können. HABERFELLNER ET AL. (2012, S. 265) bezeichnen derartige Situationen als *improvisierte Entscheidungen*. Sie sind nur dann zulässig, wenn die Konsequenzen der Entscheidung unbedeutend sind, die Entscheidung nachträglich noch korrigiert werden kann, die Entscheidung per se auf Grund klarer Vorteile einer Option eindeutig oder auf Grund indifferenter Qualitätsunterschiede unbedeutend ist.

### 3.3.2 Grundlegende Hürden für die Zielerreichung

Die Komplexität des Produktportfolios kann sich erheblich negativ auf die Ziele des Systemarchitekten auswirken. Es müssen eine Menge an Informationen und relevanten Abhängigkeiten berücksichtigt werden, um die **flexible Konfiguration** von Bausteinen eines Produktportfolios gewährleisten zu können. Je komplexer ein Portfolio ist, also je mehr Bausteine und je mehr Abhängigkeiten zwischen den Bausteinen existieren, desto mehr Aufwand ist nötig, flexible Konfigurierbarkeit zu entwickeln, zu prüfen und anzuwenden.

In der Regel handelt es sich bei den Produktportfolios von Unternehmen um historisch gewachsene Strukturen. Es wurde nicht immer zielorientiert mit Blick auf eine **konsistente und klare Portfolioarchitektur** geplant. Nach SCHUH (2005, S. 120) sollte die Produktstrukturierung in enger Abstimmung zwischen Konstruktion, Produktion, Marketing und Vertrieb geschehen. „In der Realität findet man jedoch historisch gewachsene und unabgestimmte Produktstrukturen vor“. Oft ist auch „Wildwuchs“ im Sinne der Datenhaltung entstanden, wenn über die Jahre mit den Weiterentwicklungen in der Datenverarbeitung neue Datenbanksysteme eingeführt oder abgelöst oder wenn relevante Produktinformationen nicht prozessgerecht vollständig gespeichert wurden. Je mehr Ausnahmen und Abweichungen von Prozessen und Planung an der Systemarchitektur vorbei betrieben worden sind, desto vielfältiger und unüberschaubarer ist die Portfolioarchitektur. Komplexität wirkt sich darüber hinaus negativ auf die Kommunizierbarkeit einer konsistenten und klaren Portfolioarchitektur aus. Es gibt zahlreiche Stakeholder, die im Produktentwicklungsprozess auf die Informationen des Systemarchitekten zurückgreifen wollen. Je mehr Abhängigkeiten und Elemente im Entwicklungsprozess berücksichtigt werden müssen, desto schwieriger wird es, die Portfolioarchitektur übersichtlich und konsistent im Sinne der Daten und der Gestaltungsprinzipien zu halten und entsprechend Informationen dazu zu kommunizieren.

Zudem hat sich gezeigt, dass methodische Defizite und Defizite in der Definition von Gestaltungsprinzipien sowohl Komplexität verursachen können, aber auch durch Komplexität



verstärkt werden können. Somit wird auch das Erreichen eines **effizienten Grades an Wiederverwendung** und **sinnvollen Standardisierungen** durch die Komplexität des Portfolios erschwert. Je höher methodische Defizite sind, desto rascher steigt die Komplexität in einem Produktportfolio an, was wiederum den Methodeneinsatz umso schwieriger macht.

Ein unübersichtliches und komplexes Produktportfolio hat zur Folge, dass umsatzschwache Varianten und nicht mehr benötigte Bausteine unerkant bleiben. Die Komplexität hat somit negative Folgen auf eine **marktgerechte Vielfalt**. Wenn die Teile für diese Produkte noch weiter vorgehalten werden, weiterhin Kosten für die Dokumentation, Produktpflege und Ähnliches zu Buche schlagen, hat das auch Folgen für die wirtschaftlich guten Produkte. Die „gesunden“ müssen dann die umsatzschwachen Produkte quer-subventionieren. Je zahlreicher und unübersichtlicher die Wechselwirkungen in einem Produktportfolio werden, desto aufwendiger werden die Abkündigung von umsatzschwachen Produkten sowie die Pflege einer marktgerechten Vielfalt.

Einige der in Kapitel 3.3.1 genannten Faktoren können durch den Systemarchitekten nicht beeinflusst werden. Die **externen Faktoren** Kunden, Markt, Wettbewerb und gesetzliche Rahmenbedingungen müssen in der Planung berücksichtigt werden. Auch **Faktoren aus dem Umfeld**, also der hohen Anzahl an zu berücksichtigenden Stakeholdern, Lieferanten, parallel laufender Projekte sowie von außen eingebrachten Änderungen sind wenig beeinflussbare Randbedingungen für die Arbeit des Systemarchitekten. Einige **interne Faktoren** fallen dann ins Gewicht, wenn die Rolle des Systemarchitekten in einem Unternehmen nicht geschaffen, besetzt oder ausreichend definiert ist. Defizite in der Unternehmensstrategie, Finanz- und Controlling-Aspekte sowie Defizite in der Organisation werden hier nicht weiter untersucht, da diese Probleme durch die Unternehmensführung gelöst werden müssen.

Fokus der Arbeit liegt auf der Unterstützung von **ursächlichen Problemen**. Also jenen Problemen, die das Arbeiten mit einem komplexen Produktportfolio schwierig gestalten. Aus der oben dargestellten Analyse der Komplexitätstreiber werden vier wesentliche Hürden als Grundproblem identifiziert, die der Systemarchitekt in seiner Arbeit überwinden muss, um erfolgreich seine Ziele erreichen zu können.

Methodische Defizite und Defizite bei der Definition geeigneter Gestaltungsprinzipien können behandelt werden, wenn die Grundvoraussetzungen geschaffen wurden. Solange Zeitdruck, Intransparenz und Informationsdefizite vorliegen, können geeignete Methoden nicht sinnvoll eingesetzt werden. Im Folgenden werden die **Hürden** im technischen Produktportfoliomanagement eingehender analysiert, nämlich

- Mangel an Transparenz,
- hoher Aufwand für Informationsbeschaffung durch verteilte Daten,
- hoher Aufwand für Analysen und Bewertungen sowie
- unklare Handlungsfelder und ineffektive Fehlersuche.

Es wird untersucht, inwiefern sich diese Hürden negativ auf die Zielerreichung auswirken. Die Komplexität wächst unkontrolliert weiter an, solange Mangel an Transparenz über die planungsrelevanten Bausteine, deren vertikalen und horizontalen Abhängigkeiten und den Wechselwirkungen mit dem Umfeld besteht. Hinzu kommt ein großer Aufwand für die Informationsbeschaffung, da die benötigten Daten auf verschiedenen Datenbanken, in Dateien,

in den Köpfen der Entwickler oder Ähnliches verteilt sein können. Gesicherte Aussagen über den Zustand eines Portfolios zu treffen oder weiterführende Analysen und Bewertungen sind ebenfalls mit hohem Aufwand verbunden. Somit bleiben akute wie mittelfristige Handlungsfelder unklar. Einzelne Fehler im Portfolio bleiben angesichts der Komplexität unerkannt. Probleme werden in der Regel symptomatisch behandelt, wenn sie zufällig auftreten, ohne die Auswirkungen auf das Gesamtsystem abschätzen und berücksichtigen zu können. Ein wirksames und zeitnahes Eingreifen ist somit nicht möglich. Auf Grund mangelnder Bewertungsmöglichkeiten und hohem Zeitdruck, müssen Entscheidungen dann mehr oder weniger "aus dem Bauch heraus" getroffen werden.

Abbildung 3-11 gibt einen Überblick über die nachfolgende Analyse. Es werden die internen Komplexitätstreiber aus 3.3.1, die damit verbundene Hürden und deren Wirkung auf die Instrumentalziele in der Systemarchitektur untersucht.

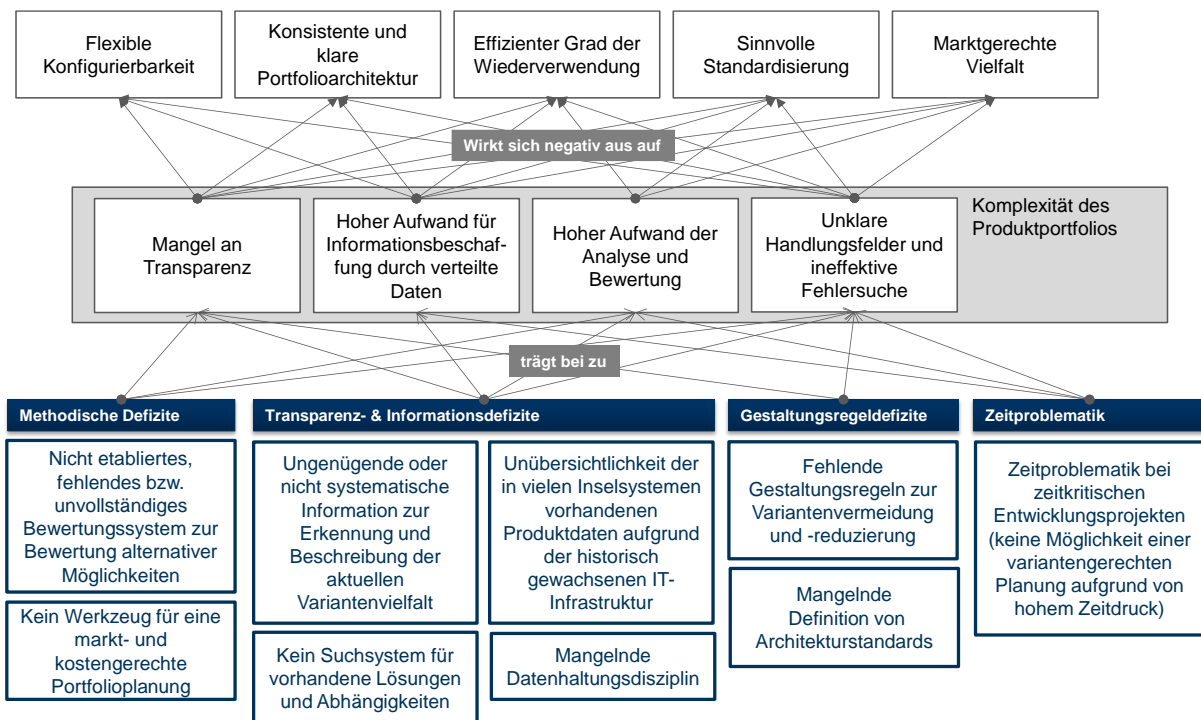


Abbildung 3-11: Problemfelder in der Systemarchitektur

## Mangel an Transparenz

Wie im vorherigen Abschnitt beschrieben, kann eine hohe Komplexität in der Portfolioarchitektur zu **Unklarheiten und zur Unübersichtlichkeit** führen. Der Aspekt der Intransparenz zielt aber nochmals verstärkt auf die Überfülle und Unübersichtlichkeit der Daten ab, aus der nur mühsam oder nahezu unmöglich die relevanten Informationen herausgefiltert werden können (KÖNIGS 2013, S. 3). Fehlende Rechnerwerkzeuge für die Portfolioplanung, ungenügende oder nicht systematisch angelegte und gepflegte Informationen und Insellösungen in der IT-Landschaft verstärken das Problem der Intransparenz. JESCHKE führt fehlende Transparenz, Gestaltungsregeln und Bewertungskriterien als Grundproblem der Systemarchitektur und der übrigen Beteiligten im Produktentstehungsprozess (JESCHKE 1997,

S. 13). Einen Mangel an Transparenz machen auch SCHUH (2005, S. 14) für einen unkontrollierten Anstieg der Produkt- und Prozesskomplexität verantwortlich. Unter anderem sind mangelnde Transparenz, eine zu kurzfristige Bauteilplanung sowie die isolierte Betrachtung einzelner Baugruppen typische Ursachen der Variantenvielfalt (RENNER 2007, S. 27).

Neben der Intransparenz über die relevanten Informationen und Zusammenhänge besteht auch oft Unklarheit darüber, welche Ziele auf Architekturebene verfolgt werden sollen und wie sie erreicht werden können. Oft herrscht Unklarheit darüber, was alles berücksichtigt werden muss, welche Wechselwirkungen wichtig sind und welche Auswirkungen Maßnahmen haben können. Die Systemarchitektur hat im Produktentstehungsprozess eine Koordinations- und Lenkungsaufgabe. Sie muss Zielstellungen unterschiedlicher Stakeholder ausgleichen und einen Kompromiss finden können (ROBERTSON & ULRICH 1998, S. 20f). Hierfür sollte der Systemarchitekt in der Lage sein, schnelle Entscheidungen auf Basis einer breiten, klar strukturierten und konsistenten Datenbasis treffen zu können. Je größer der Mangel an Transparenz, desto unklarer sind der tatsächliche Aufbau und die Gestaltungsprinzipien des Portfolios für den Systemarchitekten selbst, aber auch in der Kommunikation im Unternehmen.

Ein Mangel an Transparenz ist bei der Arbeit mit einem komplexen System problematisch. Intransparenz wirkt sich insbesondere negativ auf das Ziel aus, eine **konsistente und klare Portfolioarchitektur** als Planungsinstrument zu haben. Sowohl beim strategischen Planen wie auch beim operativen Arbeiten fehlen relevante Information. Die relevanten Informationen für die Entscheidungen entlang operativer und strategischer Schritte werden im Detail wie auch für den Überblick über das Gesamtsystem benötigt. Im Detail werden beispielsweise Informationen über einzelne Produktdaten benötigt oder Beziehungswissen einer Komponente, wo diese verbaut ist, welche Nachbarkomponenten angrenzen, welche Anforderungen mit ihr verknüpft werden. Mit Blick auf das Gesamtsystem interessiert wie ein System strukturiert ist, nach welchen Gestaltungsprinzipien Untersysteme gebildet werden. Wenn wesentliche Zusammenhänge wie z. B. Konfigurationsmöglichkeiten von Bausteinen unbekannt sind, da die Transparenz über diese Möglichkeiten fehlt, dann wirkt sich dies auch negativ auf das Ziel einer **Flexiblen Konfigurierbarkeit** im Portfolio aus.

Auch die Entscheidungen darüber, an welchen Stellen und in welchem Ausmaß **Standardisierungen sinnvoll** eingesetzt, **Wiederverwendung** von Baugruppen genutzt und **marktgerechte Vielfalt** zugelassen werden kann, bedürfen ein hohes Maß an Transparenz. Ein Mangel an Planungswerkzeugen und Informationsversorgung beeinträchtigt auch diese Ziele. Je weniger Transparenz in den Planungsinstrumenten der Systemarchitektur besteht, desto weniger zielgerichtet kann geplant werden und desto weniger kann auf fundierten Daten und klaren Gestaltungsregeln an einer Verbesserung der übrigen Ziele gearbeitet werden.

### **Hoher Aufwand für Informationsbeschaffung durch verteilte Daten**

Komplexe Portfolios umfassen eine **große Daten- und Informationsfülle** über technische Details von Bauteilen, Modulen und Produkten, Abhängigkeiten und Wechselwirkungen, Konfigurationslogik, Vertriebsinformationen wie Preise, Stückzahlen oder auch Prozessinformationen wie Verantwortlichkeiten oder Zuordnung zu Entwicklungsprojekten. Die für die Aufgaben des Systemarchitekten relevanten Informationen liegen in der Regel

verteilt auf diversen Wissensinseln. EHRENSPIEL & MEERKAMM (2013, S. 715f) führen mangelhafte Zugriffsmöglichkeit auf relevante Informationen, unzureichende Beschreibung der Produktstruktur und Fehlen von effektiven, schnellen Wiederholteil- und Ähnlichkeitssuchsystemen als Verursacher von Komplexität im Produktportfolio an.

Durch eine Vielzahl unterschiedlicher, teilweise zueinander inkompatibler IT-Systeme entstehen **Wissensinseln**. Verbreitet sind Enterprise-Resource-Planning- (ERP), Product-Data-Management- (PDM), Product-Lifecycle-Management-Systeme (PLM). Diese Systeme sind meist datenbankbasiert und gewährleisten eine konsistente Datenhaltung. Darüber hinaus werden viele Informationen in eigens entwickelten Software-Werkzeugen gepflegt. Tabellenkalkulationsprogramme wie Excel™, einfach nutzbare Datenbank-Werkzeuge oder Texteditoren schaffen Abhilfe, wenn sich individuell wichtige Informationen für ein Unternehmen nicht in den Datenfelder der großen oben genannten Werkzeuge ablegen lassen. Hinzu kommen Informationen, die nicht explizit dokumentiert wurden. Dazu gehören implizites Expertenwissen, Emails, einzelne Dateien auf lokalen Festplatten, Einträge in Abteilungs-Wikis und Ähnliches. Da die Wertschöpfungsquote beim OEM<sup>20</sup> teilweise niedrig ist, liegt sehr viel Information bei den Lieferanten. Hier ist es zusätzlich eine Frage der jeweiligen Absprachen, welche Informationen in welcher Form ausgetauscht werden können.

Die **Beschaffung der relevanten Informationen** ist für den Systemarchitekten unter Umständen **mit hohem Aufwand verbunden**. Mitarbeiter eines Industrieunternehmens scheitern bei der Informationsbeschaffung, da wichtige Informationen wie Baustein-Klassifikationen oder Umsatzdaten in den Datenbanksystemen nicht hinterlegt sind oder nicht gefunden werden können (BEMBENNECK 2005, S. 11). Oft wird ein Teil neu konstruiert, da der Aufwand der Suche nach ähnlichen oder sogar gleichen, bereits vorhandenen Lösungen im Unternehmen größer erscheint (MÜLLER ET AL. 2013). Der Datenexport aus den datenbankgestützten Systemen kann zwar normalerweise von einem Experten vorgenommen werden, allerdings erfordert dies meist einen definierten Prozess, der Managemententscheidungen voraussetzt, Ressourcen bindet und viel Zeit in Anspruch nimmt. Die Daten aus verteilten eigens entwickelten Werkzeugen zusammenzuführen, setzt für den Systemarchitekten einen guten Überblick über die Werkzeuge, eine gute Vernetzung mit den verantwortlichen Fachabteilungen und regelmäßige Kommunikation der eigenen Anforderungen an die Daten voraus. Implizite Informationen, die nicht standardmäßig dokumentiert sind, erfordern die Einberufung von Expertenkonferenzen. Der zeitliche wie finanzielle Aufwand für diese Form der Datenbeschaffung ist enorm und bringt meistens nur einen Bruchteil der benötigten Informationen zu Tage. Dennoch ist die Abfrage von Expertenwissen immer noch weitverbreitete Praxis.

Die verteilten umfangreichen Daten erfordern eine langwierige teils unmögliche Informationsbeschaffung und Auswertung. Eine **konsistente und klare Portfolioarchitektur** lässt sich somit nur mit größter Mühe bewerkstelligen. Auch für das Ziel der **flexiblen Konfigurierbarkeit** ist eine effiziente Datenbeschaffung Grundvoraussetzung. Damit ist ein manueller Aufwand verbunden, der angesichts eng gesetzter Meilensteine und hohem

---

<sup>20</sup> OEM steht für Original Equipment Manufacturer, Erstausrüster, der Produkte unter eigenem Namen in den Handel bringt (Wikipedia: <http://de.wikipedia.org/wiki/Erstausrüster>, entnommen am 28.10.2013)

Zeitdruck im Produktentwicklungsprozess nicht zu rechtfertigen ist. Geeignete Rechnerwerkzeuge fehlen meist oder lassen sich nicht flexibel genug einsetzen. Fehlentscheidungen über **sinnvolle Standardisierungen, marktgerechte Vielfalt** und **Wiederverwendung** können allerdings weitreichende Folgen haben. Die Rolle des Systemarchitekten erfordert es, in einer frühen Phase der Produktentwicklung unter Zeitdruck nachvollziehbare und nachhaltige Entscheidungen zu treffen. Die getroffenen Entscheidungen sind weichenstellend für den wirtschaftlichen Erfolg nachfolgender Entwicklungen. Aus Mangel an Unterstützungswerkzeugen werden Maßnahmen daher oft „aus dem Bauch heraus“ beschlossen ohne eine der Wichtigkeit angemessenen Bewertung durchzuführen. Je verteilter und unzugänglicher die benötigten Daten sind, desto höher ist der Aufwand der Datenbeschaffung und Aufbereitung und desto längere Zeit wird benötigt, Entscheidungen auf rationalen Grundlagen zu treffen.

### **Hoher Aufwand der Analyse und Bewertung**

**Analysen** im Umfeld der Systemarchitektur können **sehr aufwendig und umfangreich** werden. Zum einen muss eine große Datenbasis verarbeitet werden, zum anderen müssen zahlreiche Wechselwirkungen und unterschiedliche Stakeholder-Interessen berücksichtigt werden. Die aufgabenspezifischen Informationen müssen aus unterschiedlichen Datenquellen zusammengeführt und geprüft werden. Die Datensätze, die anschließend bearbeitet werden können, umfassen dann schnell mehrere tausend Datenpunkte. Mit Tabellenkalkulations-Programmen können hierbei noch eine Reihe von Analysen unter Einplanung entsprechender Ressourcen durchgeführt werden. Manuelle Operationen sind allerdings fehleranfällig und erreichen schnell die Grenzen ihrer Möglichkeiten. Außerdem erfordern sie Expertise, die nicht immer in der für die Aufgabe benötigten Zeit abgestellt werden kann. Sobald beispielsweise für eine Konfigurierbarkeits-Analyse alle Datenpunkte paarweise miteinander abgeglichen und jeweils mehrere Bedingungen berücksichtigt werden müssen, multiplizieren sich die Einzelschritte und Iterationen zu unüberschaubaren Arbeitspaketen. Rechnerwerkzeuge sind dann nötig. Das Angebot solcher Werkzeuge und deren Funktionsumfang sind allerdings begrenzt. ZAGEL (2010, S. 37ff) untersucht eine Auswahl von Varianten-Management-Systemen.

Ein weiterer Aspekt ist die **Bewertung komplexer Produktportfolios**. Die Ermittlung von Kennzahlen basiert oft auf inkonsistenten und unvollständigen Daten. JUNGE (2005) wie auch HEILEMANN ET AL. (2013) zeigen auf, dass viele der in der Literatur vorgeschlagenen Kennzahlen sehr aufwendig in der Berechnung sind, die erforderlichen Informationen im Unternehmen meist fehlen und die Aussagekraft meist stark begrenzt ist. Kennzahlen können ohnehin nur einen aggregierten Überblick über den Zustand des Produktportfolios geben. Um lokale Schwachpunkte erkennen und somit wirksame Handlungsfelder ableiten zu können, eignen sich Kennzahlen als Bewertungsinstrument nur bedingt.

Ohne ein entsprechendes **Mess- und Bewertungswerkzeug** werden Entscheidungen allerdings ohne Rückkoppelung getroffen. Die Auswirkungen von Entscheidungen lassen sich also nicht ex-ante abschätzen oder ex-post prüfen. Die Auswahl der richtigen Maßnahmen aus rationalen Gesichtspunkten gestaltet sich dann schwierig. Die manuelle Aufbereitung der Daten und die darauf aufbauende Bewertung und Ableitung von Handlungsfeldern können mit den begrenzten

Ressourcen oft nicht geleistet werden. Also können Entscheidungen über den richtigen Grad an Wiederverwendung, Standardisierung und Vielfalt nicht auf rationaler Entscheidungsgrundlage getroffen werden. Entscheidungen mit weitreichenden Folgen werden dann auf nicht rationalen Grundlagen getroffen. Je komplexer sich die Analyseaufgaben gestalten und je höher der Aufwand für Bewertungen ist, desto eher werden diese Tätigkeiten angesichts knapper Ressourcen vermieden und somit muss auf eine rationale Entscheidungsgrundlage verzichtet werden.

### **Unklare Handlungsfelder und ineffektive Fehlersuche**

Wie aus den vorherigen Kapiteln hervorgeht wird rationales Entscheiden erschwert durch den hohen Aufwand für die Informationsbeschaffung, Analyse und Bewertung komplexer Produktportfolios. Um rationale Entscheidungen in komplexen Produktportfolios treffen zu können, sind klar umrissene Ziele sowie messbare Kennzahlen erforderlich. Auch wenn Bewertungen durch Kennzahlensysteme ansatzweise genutzt werden können, so bietet diese Möglichkeit nur eine hochaggregierte Sicht auf die Portfolioarchitektur. Dadurch bleiben allerdings die **kritischen Handlungsfelder** und **wirksamen Stellhebel unklar**. Eine gezielte Fehlersuche oder Identifizierung von lokalen Verbesserungspotentialen ist damit nahezu unmöglich. Die **Fehlersuche** gestaltet sich zu einer Suche nach der sprichwörtlichen „Nadel im Heuhaufen“. Um die kritischen Elemente in einer komplexen Struktur mit mehreren Tausend Knoten zu finden, bedarf es aufwendigen, zeitintensiven Analysen. Wenn z. B. nach Mustern gesucht werden soll, die Elemente mit bestimmten Wechselwirkungen, benachbarten Elementen oder Ähnliches beschreiben, ist eine manuelle Suche aussichtslos.

Für eine effektivere Mustersuche müssen **effektivere Werkzeuge** bereitgestellt werden. Wenn Handlungsfelder unerkannt bleiben und Fehler nicht gefunden werden können, kann keine konsistente und klare Portfolioarchitektur sichergestellt werden. Fehlerhafte Bausteine können nicht herausgefiltert werden. Dies führt zu einer ungewollten zusätzlichen Vielfalt. Weiterhin könnten Bausteine, die eine flexible Konfigurierbarkeit, Wiederverwendung und sinnvolle Standardisierung beeinträchtigen, als Muster definiert und effektiv im Sinne der Zielerreichung verändert werden. Je unklarer allerdings die Handlungsfelder und je ineffektiver sich die Muster- und Fehlersuche gestaltet, desto weniger effektiv und effizient können die vorhandenen Ressourcen eingesetzt werden.

### **Zusammenfassende Fragen**

Die Herausforderungen, die durch die Hürden entstehen, können in folgenden Fragen zusammengefasst werden:

- (1) Wie kann die Komplexität in Produktportfolios reduziert bzw. besser beherrscht werden, um die Ziele der flexiblen Konfigurierbarkeit, einer konsistenten und klaren Portfolioarchitektur, einer wirtschaftlichen Wiederverwendung und Standardisierung sowie einer marktgerechten Vielfalt zu unterstützen?
- (2) Wie kann die Transparenz für den Systemarchitekt und seine Stakeholder, denen er Informationen bereitstellen muss, erhöht werden?

- (3) Wie können verteilte Daten effektiver und effizienter für die Aufgaben des Systemarchitekten zusammengeführt und aufwendige Tätigkeiten bei der Datenbeschaffung und Aufbereitung reduziert werden?
- (4) Wie können Analyse und Bewertung effektiver und effizienter gestaltet werden, um eine rationalere Entscheidungsgrundlage für zielgerichtetere Problemlösungen bereitstellen zu können?
- (5) Wie kann eine effektive Muster- und Fehlersuche bereitgestellt werden, um kritische Handlungsfelder im Produktportfoliomanagement schneller und ressourcenschonender zu identifizieren?

Aus diesen Fragen und den vorausgegangenen Untersuchungen sollen im nächsten Kapitel die Anforderungen an den Lösungsansatz abgeleitet werden.

## 3.4 Anforderungen an den Lösungsansatz und Beitrag der Arbeit

### 3.4.1 Anforderungen

In den vorausgegangenen Kapiteln wurden Probleme im Umgang mit komplexen Produktportfolios diskutiert. Als grundlegende Probleme wurden dabei mangelnde Transparenz, der hohe Aufwand für die Informationsbeschaffung und -aufbereitung sowie eine schwierige Identifikation von Fehlern und Handlungsfeldern identifiziert.

Die Methoden aus der Literatur bieten zahlreiche Ansätze, Vielfalt in Produktportfolios zu reduzieren und zu beherrschen. Auch zur Bewertung und Steuerung finden sich geeignete Methoden. Allerdings wird der Betrachtungsumfang immer stark auf einzelne Produkte oder Produktfamilien eingeschränkt. Die Methoden werden oft mit Formblättern, Checklisten, oder in Expertenworkshops durchgeführt. Dies ist mit hohem Aufwand verbunden und kann dementsprechend nur selten durchgeführt werden. Eine kontinuierliche Bewertung, Anpassung und Vorausplanung des Produktportfolios sind somit nicht möglich. In der Position des Systemarchitekten müssen die Systemgrenzen auf das gesamte Portfolio ausgeweitet werden und modellhaft abbildbar sein.

An den Lösungsansatz ergeben sich dadurch folgende Anforderungen:

- (A1) Die Methodik muss es ermöglichen, auch **größere Betrachtungsumfänge modellhaft abbilden** zu können. Zudem muss der Betrachtungsumfang **anpassbar** sein, also auf weitere relevante Aspekte aufwandsarm **erweitert** werden können. Hierbei sollten keine konzeptionellen Einschränkungen gelten, welche Datenarten aus welchen Datenquellen im Modell zusammengeführt werden können.

Eine konsistente Datenhaltung und eine rasche Verfügbarkeit von Informationen sind auf Grund mangelnder IT-Lösungen schwierig zu bewerkstelligen.

- (A2) Die Methodik soll ein **Rechnerwerkzeug** zur Verfügung stellen, das eine **konsistente Datenhaltung** ermöglicht.

In der Systemarchitektur ergeben sich zahlreiche repetitive Aufgaben, die mit sehr großem manuellen Aufwand verbunden sind.

(A3) Die Methodik soll eine Möglichkeit bieten, **Expertenwissen zu formalisieren** und somit für den Rechner interpretierbar und automatisch ausführbar zu machen. Somit können Wissensbibliotheken aufgebaut und die Arbeit der Systemarchitektur nachhaltig effizienter gestaltet werden.

Mit den heutigen Mitteln ist eine Mustersuche in komplexen Produktstrukturen nahezu unmöglich. Um Fehler in Produktportfolios erkennen und somit unnötige Komplexität reduzieren und langfristig vermeiden zu können, bedarf es einer effektiven Mustererkennung.

(A4) Die Methodik sollte Werkzeuge zur Verfügung stellen, **Mustererkennung in komplexen Produktstrukturen** für die Systemarchitektur verfügbar zu machen.

Um die Ergebnisse der Architekturarbeit im Unternehmen kommunizieren zu können, bestehen hohe Anforderungen an die Aufbereitung und Visualisierung. In zahlreichen Methoden wird das Ergebnis für eine bestimmte Situation in einer bestimmten Form visualisiert. Inhalt und Visualisierung als Einheit bilden das Modell ab. Hierfür werden die relevanten Informationen dem Verwendungszweck entsprechend reduziert.

Die Informationen müssen dann für zweckfremde Fragestellungen aufwendig und meist manuell aus den Modellen extrahiert und aus verteilten Modellen zusammengeführt werden. Um eine effiziente Nutzung der im Unternehmen verfügbaren Informationen zu bewerkstelligen, sollten Informationen unabhängig von ihrer Visualisierung und somit ohne zweckgebundene Reduzierung verfügbar gemacht werden.

(A5) Die Methodik sollte Ergebnisse in der Weise hervorbringen, dass diese ohne größeren Aufwand **situations- und stakeholder-gerecht aufbereitet und visualisiert** werden können. Die **Visualisierung** soll dabei sowohl **vom Inhalt** der Datenbasis als auch vom Inhalt der Ergebnisse einer Operation **unabhängig** bleiben.

Die Nachvollziehbarkeit von Entscheidungen ist in einem komplexen Entwicklungsumfeld wie dem der Systemarchitektur nicht immer gegeben. Oft können Entscheidungen nicht auf rationalen Fakten getroffen werden, sondern werden aus dem Bauch heraus entschieden.

(A6) Die Methodik sollte daher die **Reproduzierbarkeit, Objektivierung und Dokumentation der Entscheidungsunterstützung** vereinfachen und gewährleisten.

Im Rahmen dieser Arbeit soll anhand eines Demonstrators der „Proof of Concept“ dazu erbracht werden.

### 3.4.2 Angestrebter Beitrag der Arbeit

Die Methodik, die in dieser Arbeit vorgestellt wird, umfasst vier Einzelmethoden. Die prototypische Implementierung der Methoden erfolgt anhand eines rechnergestützten Demonstrators. Die Methoden leisten folgende Beiträge:

#### **Beitrag 1: Methode zur Zusammenführung relevanter Daten in graphenbasierter Form**

Der Anwendung der Methoden zu Komplexitätsreduzierung geht stets eine mehr oder weniger aufwendige Datenbeschaffung voraus. Die Daten werden aufbereitet, in der Methode



verarbeitet und aus dem Ergebnis im besten Fall Maßnahmen abgeleitet. Wenn die Maßnahmen erfolgreich durchgeführt wurden, müssen die ursprünglichen Datenquellen durchgängig und konsistent zueinander auf den aktuellen Stand gebracht werden. Dadurch entsteht wiederum Aufwand.

In der Methodik zur Mustererkennung in komplexen Produktportfolios soll aufgezeigt werden, wie sich die verteilte Datenbasis zusammenführen und konsistent pflegen lässt. Hierfür soll eine erweiterbare Datenbeschreibungssprache herangezogen werden. Sie soll gewährleisten, dass für die Aufgaben der Systemarchitekten alle relevanten Entitäten und Relationen abbildbar und verwertbar sind.

### **Beitrag 2: Methode zur Formalisierung repetitiver Aufgaben im Produktportfoliomanagement**

Einige Aufgaben der Systemarchitekten erfordern kontinuierlich eine hohe Ressourcenbindung. Repetitive Aufgaben wie z.B. die Bereitstellung von Kennzahlen über das Produktportfolio, Informationsbereitstellung für Stakeholder oder die Erstellung von Verwendungsnachweisen sind oft zeitkritisch. Sie dienen allerdings der Unterstützung meist wichtiger Entscheidungen.

In dieser Arbeit soll gezeigt werden, wie sich repetitive Aufgaben prinzipiell automatisieren lassen. Anhand von Beispielen soll veranschaulicht werden, wie mit Methoden der Graphtransformation solche repetitiven Aufgaben formal beschrieben und somit vom Rechner bearbeitet werden können.

### **Beitrag 3: Methode zur automatischen Muster- und Fehlererkennung**

In komplexen, verteilten Datenstrukturen ist es nahezu unmöglich, Fehler zu identifizieren. Schon ein einfacher Abgleich, ob beispielsweise die Klassifikationen von Bausteinen auch ihrer tatsächlichen Verwendung entsprechen, mag bei einem Einzelprodukt machbar erscheinen, kann bei großen verteilten Datenmengen eines Gesamtportfolios zu einer unerfüllbaren Aufgabe werden.

Mit der Methode zur automatischen Muster- und Fehlererkennung soll aufgezeigt werden, wie mit Hilfe der Graphtransformation Muster in Strukturen formal beschrieben und somit effizient durch den Rechner gefunden werden können. Somit lassen sich sowohl Ziel- wie auch Fehlermuster definieren und einfach in komplexen Strukturen finden.

### **Beitrag 4: Methode zur intuitiven Visualisierung relevanter Zusammenhänge**

Die Ergebnisse von Datenbankabfragen liegen in der Regel in tabellenförmigen Exportdateien vor, die in Tabellenkalkulationsprogrammen oder ähnlichen weiterverarbeitet werden können. Um zu Vorlagen zu gelangen, die in der Zusammenarbeit mit unterschiedlichen Stakeholdern als Entscheidungsunterstützung dienen sollen, ist großer Aufwand für die Aufbereitung nötig. Zudem ist es oft aufwendig entscheidungsrelevante Zusammenhänge so intuitiv aufzubereiten, dass kritische Handlungsfelder schnell ersichtlich werden.

Mit der Methode soll dargestellt werden, wie sich die Ergebnisse entscheidungsgerecht auf unterschiedliche Weise aufbereiten lassen. Mit der formalen Beschreibung des Ergebnisses in

einem Graph kann die Visualisierung unabhängig gewählt werden. Somit können unterschiedliche Analysemodelle konsistent erzeugt werden und die Entscheidungsfindung unterstützen.

In Abbildung 3-12 wird der Zusammenhang zwischen dem Beitrag und den Zielen der Arbeit aufgezeigt und wie dadurch das Problem zu großer Komplexität und damit verbundenen Hürden abgemildert werden soll. Durch den Einsatz der vier oben genannten Methoden sollen die drei Forschungsziele (1) Transparenz schaffen, (2) effizienteres Arbeiten und (3) effektiveres Arbeiten mit komplexen Produktportfolios erreicht werden (vgl. Kapitel 1.2).

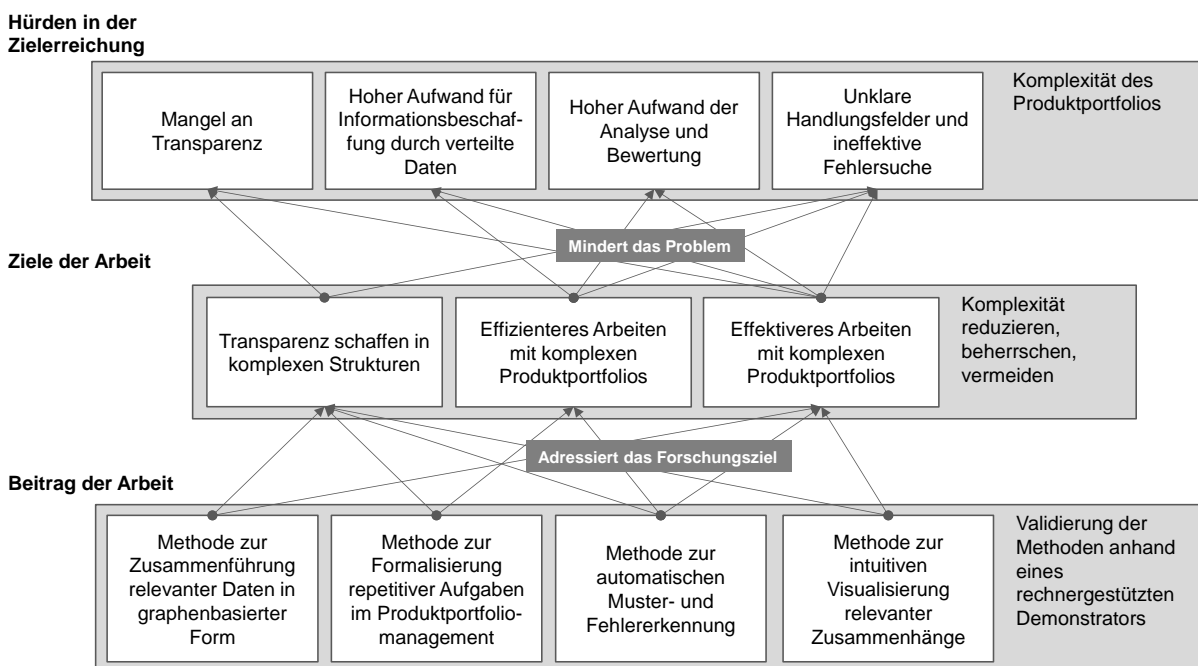


Abbildung 3-12: Verknüpfung von Beitrag mit den Zielen der Arbeit

In den folgenden Kapiteln wird zunächst in die technologischen Grundlagen der Graphtransformation eingeführt (Kapitel 4). Daraufhin wird die Methodik in Kapitel 5 vorgestellt und in Kapitel 6 anhand von zwei Anwendungsfällen in der Industrie evaluiert.

## 4. Graphenbasierte Repräsentation und Manipulation von komplexen Strukturen

Grundlagen für das Arbeiten mit dem Produktportfolio sind eine oder mehrere modellbasierte Repräsentationen dieses komplexen Systems. Über das Produktportfolio liegen Informationen über Elemente unterschiedlicher Elementtypen vor, ebenso wie Informationen über Verknüpfungen zwischen den Elementen. Im Folgenden wird erläutert, mit welcher Methode das Produktportfolio den oben aufgeführten Anforderungen entsprechend modelliert wird. Dafür wird zunächst in die Modellierung auf Basis von typisierten, attribuierten Graphen eingeführt und anschließend das Konzept der Graphtransformation erläutert.

### 4.1 Modellierung komplexer Produktstrukturen

#### 4.1.1 Grundlagen der Modellbildung

Modellhafte Abbildungen dienen der Veranschaulichung komplexer Systeme und sind somit ein wesentliches Prinzip des Systemdenkens (HABERFELLNER ET AL. 2012, S. 41). Nach KOSSIAKOFF ET AL. (2011, S. 317ff) sind **Modelle** ein unverzichtbares Werkzeug des Systems Engineering, um komplexe Strukturen und Relationen für den Analysten und für den Konstrukteur verständlich zu machen. HABERFELLNER ET AL. (2012, S. 132ff) stellen die strukturorientierte Betrachtung von Systemen als geeignet heraus, Elemente unterschiedlicher Domänen mit verschiedenen Beziehungsarten abzubilden. Modelle sind Vereinfachungen der Realität und zeigen daher immer nur Teilaspekte auf. Welche Elemente und Beziehungen betrachtet werden, „hängt von der Art des zu untersuchenden Problems und von Zweckmäßigkeitüberlegungen ab“ (HABERFELLNER ET AL. 2012, S. 132).

In der Literatur finden sich **Ansätze und Methoden** zur Erstellung, Modellierung und Analyse von Produktstrukturen (PONN & LINDEMANN 2011). Modelle lassen sich ihrem Verwendungszweck entsprechend nach Produkt-, Prozess-, Berechnungs-, Simulations- und Prognosemodell oder über die Phasen des Entwicklungsprozesses nach Anforderungs-, Funktions-, Prinzip- und Gestaltmodell klassifizieren (VAJNA ET AL. 2009, S. 146–149).

In Modellen können unterschiedliche **Aspekte** berücksichtigt werden. ULRICH (1995) sowie GÖPFERT (1998) stellen eine Verknüpfung von Funktions- und Komponentenstruktur vor. UMEDA ET AL. (1996) entwickelten ein Modellierungs-Schema, -Prozess und -Werkzeug („Function-Behavior-State-Modeler“), die eine Verknüpfung von Funktions- mit Verhaltens- und Strukturmodellierung ermöglichen. Im „Axiomatic Design“ nach SUH (2001) werden Kunden-, Funktions-, Bauteil- und Prozessdomäne miteinander verbunden. Ansätze zur rechnergestützten Modellierung mechatronischer Systeme mittels formalisierter Partialmodelle entwickelten GAUSEMEIER ET AL. (vgl. GAUSEMEIER ET AL. 2008; GAUSEMEIER ET AL. 2012; FRANK 2006). MALMQVIST (2002, S. 209) gibt einen Überblick über matrixbasierte Modellierungsmethoden in der Produktentwicklung und stellt als wesentliche Elemente von Produktmodellen Eigenschaften, Anforderungen, Funktionen, Gestaltungsparameter, Komponenten, Module, Prozesse entlang des Lebenszyklus und der Varianten dar. Aspekte wie Auftretenswahrscheinlichkeit und Auswirkung von Änderungen in Produkten und

Produktarchitekturen werden unter anderem bei CLARKSON ET AL. (2004); GIFFIN ET AL. (2009); KISSEL ET AL. (2011) und KOH ET AL. (2012) berücksichtigt.

### 4.1.2 Graphenbasierte Modellierung

Die **Anforderungen** an ein umfassendes Modell des Produktportfolios im Rahmen dieser Arbeit sind, Transparenz zu schaffen und ein effektiveres und effizienteres Arbeiten mit dem Modell zuzulassen. Daher sollen entscheidungsrelevante Informationen im Modell abgebildet, effizient ausgewertet und intuitiv dargestellt werden können. Alle erforderlichen Informationen und Zusammenhänge lassen sich als Netzwerk darstellen. Ein solches Netzwerk ist mathematisch gesehen ein Graph.

Graphen bestehen aus **Knoten** und **Kanten**. Die Elemente eines Systems werden als Knoten dargestellt, die Beziehungen zwischen den Elementen als Kanten. Die Kanten können dabei **gerichtete** oder **ungerichtete** Beziehungen abbilden (vgl. auch HABERFELLNER ET AL. 2012; LINDEMANN ET AL. 2009). Ein gerichteter Graph mit einem Pfeil als Relation zwischen jeweils zwei Knoten wird als *Digraph* bezeichnet. Bei einem *Multigraph* können jeweils zwei Knoten auch durch mehrere Kanten miteinander verbunden sein.

Eine andere Form der Darstellung graphenbasierter Informationen ist die **Matrix**. Die Elemente des Systems werden in der ersten Spalte bzw. Zeile einer Tabelle eingetragen, die Relationen werden mit einem Eintrag in der Matrix gekennzeichnet (LINDEMANN ET AL. 2009). Mathematisch wird der Graph durch seine **Adjazenzmatrix** ausgedrückt. Ein ungerichteter Graph  $G = (V, E)$  mit den Kanten  $E$  und den Knoten  $V$  habe  $n$  Knoten. Die Adjazenzmatrix  $A(G)$  ist dann eine  $n \times n$  Matrix mit den Eintragungen  $a_{ij}$  = Anzahl der Kanten zwischen  $i$  und  $j$ , mit  $i, j$  von  $1 \dots n$  (TITTMANN 2003). Nachfolgende Abbildung zeigt die drei Darstellungsarten.

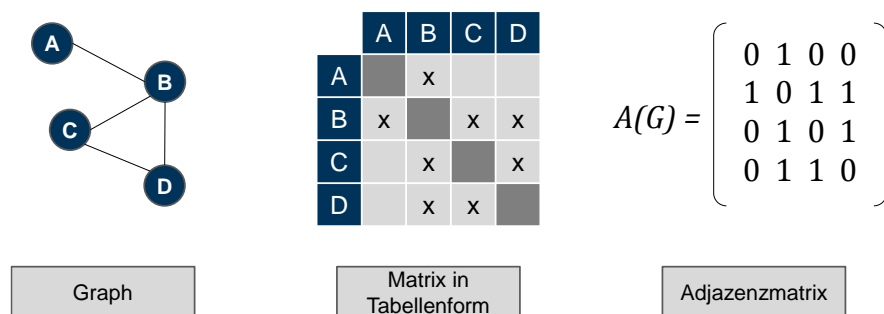


Abbildung 4-1: Gleichwertige Darstellungsformen von netzwerkartigen Strukturen

Graphenbasierte Modelle können **umfeld-, wirkungs- und strukturorientierte Betrachtungen** unterstützen (HABERFELLNER ET AL. 2012, S. 41ff). Bei der umfeldorientierten Betrachtung konzentriert man sich auf die Wechselwirkungen des Systems mit seiner Umgebung. Die wirkungsorientierte Betrachtung stellt Input-/Output-Beziehungen in den Vordergrund und beobachtet das Systemverhalten. Bei der strukturorientierten Betrachtung, die hier schwerpunktmäßig Verwendung findet, interessieren die strukturelle Vernetzung der Elemente, deren hierarchischer Aufbau und Flussbeziehungen untereinander.

Graphen können eine Reihe von **Strukturmerkmalen und Metriken** aufweisen. MAURER (2007) stellt die Bedeutung von Kreisschlüssen im Kontext von technischen Produkten als Komplexitätstreiber heraus. KREIMEYER & LINDEMANN (2011) interpretieren Strukturmerkmale im Kontext des Prozessmanagements. BIEDERMANN & LINDEMANN (2012) untersuchen Strukturcharakteristika, lokale Knotenkennzahlen und deren Signifikanz für die Bewertung von technischen Systemen. Die Berechnung und Auswertung von Strukturcharakteristika unterstützen Rechnerwerkzeuge wie Loomeo™ der Firma Teseon GmbH<sup>21</sup>. Die Interpretation im jeweiligen Kontext der Untersuchung erfordert in der Regel Experten. Für eine Vertiefung der Thematik wird auf LINDEMANN ET AL. (2009) verwiesen.

### Typisierte Graphen

In einem **einfachen Graph** werden ausschließlich Elemente und Relationen jeweils eines Typs (einer Domäne) verwendet, wie z. B. Bauteile oder Prozessschritte. Die Typisierung kann, muss aber nicht explizit angegeben sein. Werden Elemente mehrerer Domänen in einem Graph abgebildet, handelt es sich um einen **typisierten Graph** (vgl. BARDOHL ET AL. 2003; CORRADINI ET AL. 1996). Ein solcher Graph beinhaltet beispielsweise Anforderungen, Funktionen und Komponenten. Die Knoten, die die Elemente der Anforderungen repräsentieren, müssen entsprechend als Anforderungsknoten typisiert werden, ebenso wie Funktionen und Komponenten. Unterschiedliche Kantenarten sind in einem typisierten Graph ebenfalls zulässig. Somit können unterschiedliche Relationsarten abgebildet werden. Die Definition der Typisierung erfolgt im Metamodell.

Vertreter der typisierten Graphen wurden bereits in Kapitel 3.2.4 vorgestellt. Die Multi-Domain-Matrix (MAURER 2007) ist eine typisierte Strukturbeschreibung, die sich in Graphform überführen lässt. Durch die vordefinierten Domänen und Relationsarten ist eine MDM quasi Metamodell und Modell in einem.

### Attribuierte typisierte Graphen

Werden den Knoten und Kanten eines typisierten Graphen weitere Attribute zugeordnet, liegt ein **attribuierter typisierter Graph** vor. Die Attribute können innerhalb des Metamodells vererbt werden. Führt ein Eltern-Elementtyp beispielsweise das Attribut *name*, so können die Kinder-Elementtypen dieses und weitere Attribute erben. Dies ermöglicht einerseits die Entwicklung eines schlanken Metamodells, indem Mehrfachdefinitionen eingespart werden können. Andererseits können komplexe Abfragen vereinfacht werden, da generische Konzepte oben in der Vererbungshierarchie angesprochen werden können.

Attribuierte typisierte Graphen werden beispielsweise in standardisierten Modellierungssprachen UML und SysML verwendet, die im nächsten Abschnitt beschrieben sind. Auch für Wissensrepräsentationen mit Ontologien werden semantische Netzwerke entwickelt, die attribuiert werden können. Hierfür wird die Sprache **Web Ontology Language (OWL)** verwendet.

---

<sup>21</sup> [www.teseon.de](http://www.teseon.de) (entnommen am 15.11.13)

## Standardisierte und Domänenspezifische Modellierungssprachen

Die **Unified Modeling Language (UML)** wurde von der Object Management Group (OMG) als standardisierte Modellierungssprache für die Entwicklung von Software entwickelt und ist von der ISO standardisiert (ISO 19505-1; ISO 19505-2). In Zusammenarbeit mit INCOSE wurde UML eine Standardsprache im Systems Engineering. Die UML legt fest, welche Objekte und Beziehungen für die Systemmodellierung verwendet werden dürfen und definiert die Notationsvorschriften. Durch den integrierten Erweiterungsmechanismus können neue sogenannte Stereotypen definiert werden (WEILKIENS 2008, S. 23). Das macht die Modellierungssprache anpassbar. Um den Anforderungen des Systems Engineering gerecht zu werden, wurde die UML angepasst und der „Dialekt“ **Systems Modeling Language (SysML)** entwickelt.

Die **SysML** ergänzt und verändert ein paar Konzepte der UML. Neu eingeführt wurden ein Anforderungs- und ein Zusicherungsdiagramm, abgeändert wurden das Interne Blockdiagramm, das Blockdefinitionsdiagramm und das Aktivitätsdiagramm. In der SysML werden Klassen der UML weggelassen, die schwerpunktmäßig für die Softwareentwicklung und weniger für die Entwicklung physisch-technischer Systeme benötigt werden (WEILKIENS 2008, S. 299ff). Nachfolgende Abbildung 4-2 zeigt die Diagramme der SysML und die Veränderungen gegenüber der UML auf. Die Sprachen unterscheiden grundsätzlich zwischen der Struktur- und der Verhaltensmodellierung. Weiterführende Informationen finden sich bei u.a. bei FRIEDENTHAL ET AL. (2011) und WEILKIENS (2008).

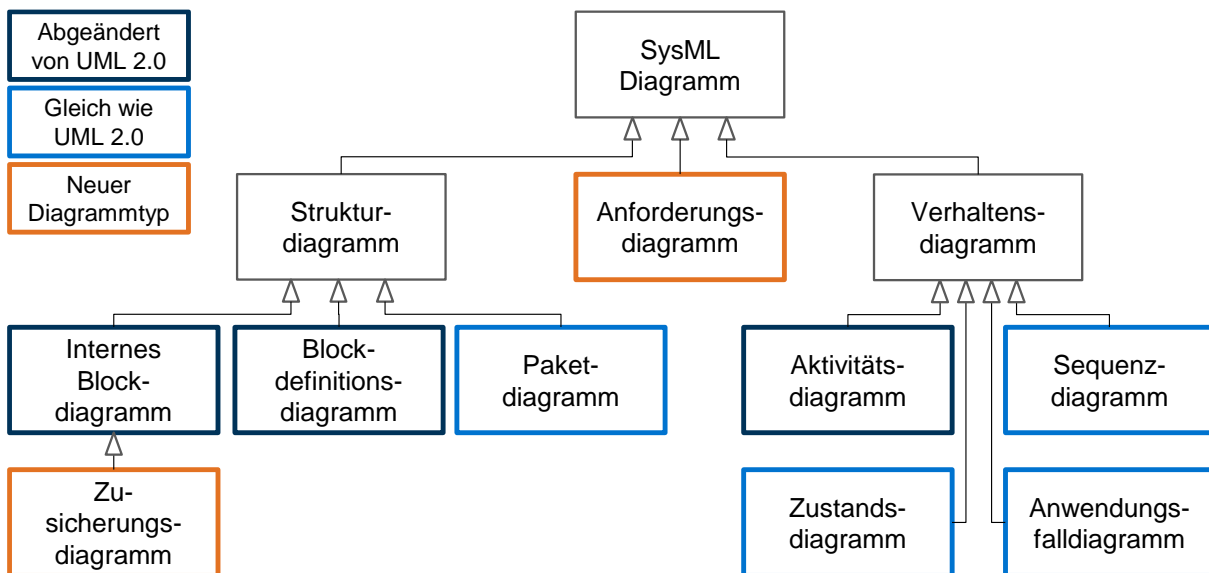


Abbildung 4-2: SysML Diagramm Taxonomie (OMG SysML v1.3, S.167)

Eine **domänenspezifische Sprache (DSL)** ist eine formale aber nicht unbedingt standardisierte Modellierungssprache, die für ein bestimmtes Anwendungsproblem entwickelt wird.

Der **Vorteil** in der Verwendung einer DSL liegt darin, dass Benutzer keine Modellierungssprache neu erlernen müssen, wenn beim Aufbau der Sprache vertraute Begrifflichkeiten und Konzepte verwendet werden. Somit werden die Modelle intuitiv verständlich und sind entsprechend eingängig darstellbar. Das Erlernen von UML oder SysML

ist mit verhältnismäßig großem Aufwand verbunden, da die abstrakten Sprachkonzepte erst erlernt und dann auf die entsprechende Problemdomäne übertragen werden müssen. Weitere Vorteile sind, dass alle benötigten Objekte und Relationsarten festgelegt werden können ohne „Behelfskonstrukte“ einer standardisierten Sprache nutzen zu müssen (z. B. bei der Modellierung von Varianten in SysML). Hinzu kommt, dass die Darstellung der Objekte frei wählbar und somit den Anforderungen unterschiedlicher Stakeholder der Modelle berücksichtigt werden können.

Die **Nachteile** einer DSL liegen generell darin, dass durch den Mangel einer Standardisierung die Konsistenz und Fehlerfreiheit in der formalen Modellbeschreibung geringer sein können und die Verbreitung von Metamodellen und Modellen nur eingeschränkt möglich ist. Weitere Nachteile einer DSL sind, dass es keine Standardtools gibt, keine spezifische Literatur vorhanden ist und Best Practices ggf. nur eingeschränkt verfügbar sind. Durch die Erweiterbarkeit von UML und SysML können auch diese beiden Sprachen zu einer DSL weiterentwickelt werden. KERZHNER & PAREDIS (2009) erweiterten beispielsweise die Konstrukte der SysML um eine Komponentenhierarchie, die die Struktur, das dynamische Verhalten, Kosten und Kompatibilität abbildet.

### Auswahl der Modellierungssprache

Alle Wissensrepräsentationsformalisten werden nach PUPPE (1988, S. 18) als **Kalkül** aufgefasst. Es beschreibt, mit welchen syntaktischen Mitteln der semantische Problembereich beschrieben werden kann. „*Korrektheit* des Kalküls ist gegeben, wenn alle syntaktisch herleitbaren Schlussfolgerungen auch semantisch folgen“. Die *Mächtigkeit* drückt aus, welche Aussagen über die Realität ausdrückbar sind. Mit der *Adäquatheit* und *Effizienz* eines Kalküls wird ausgedrückt, wie einfach und elegant („natürlich“) Problembereiche beschrieben und Lösungen hergeleitet werden können.

Wie UML und SysML liegt einer DSL ein Metamodell zu Grunde, das die Objekte und Relationen formal beschreibt. Jedes Modellelement, das mit der domänenspezifischen Modellierungssprache erstellt wird, instanziiert dann den jeweiligen, definierten Elementtyp im Metamodell.

Die hier verwendete Modellierungssprache ist eine domänenspezifische Sprache. Auf Grund ihrer Eigenschaften erfüllt sie am besten die oben genannten Eigenschaften des Kalküls. Bei der Entwicklung wurden die Konstrukte der **UML** und **SysML** berücksichtigt. Es wurde jedoch darauf verzichtet, die Methodik auf Basis der SysML komplett zu entwickeln, da einerseits der Einarbeitungsaufwand in die Modellierungssprache und deren Aufbau hoch ist, nötige Freiheitsgrade eingeschränkt sind und die Anpassung käuflicher SysML-Rechnerwerkzeuge nicht zielführend erscheint.

Mit den **matrixbasierten Modellierungsansätzen** wie DSM und MDM ist es möglich, aufwandsarm eine DSL zu spezifizieren. Die Matrixdarstellung ist intuitiv und auch für den Laien einfach zu verstehen. Die Matrixdarstellung wird oft zur Informationsakquise und zur Erzeugung von Graphen verwendet (LINDEMANN ET AL. 2009). Diese Ansätze können hier allerdings auch nicht weiter verfolgt werden. Sie stoßen rasch an ihre Grenzen, wenn die Graphen komplexer werden, da beispielsweise keine Hierarchien von Elementtypen im Metamodell abbildbar sind. Sobald ein Graph mehr als hundert Knoten aufweist, werden

Matrixdarstellungen unübersichtlich und unhandlich. Graphen eines Produktportfolios können aber schnell mehrere tausend Knoten aufweisen.

Hinzu kommen **konzeptionelle Grenzen der Ausdrucksfähigkeit** der Matrixdarstellung. Eine MDM wird dazu genutzt, das Metamodell des Graphen formal zu beschreiben (vgl. Abbildung 3-9 in Kapitel 3.2.4, S. 56). Die Vielzahl an unterschiedlichen Elementtypen und Relationsarten zur Modellierung eines Produktportfolios erfordern das Anlegen zahlreicher Domänen und teilweise mehrfach belegter Relationsfelder, was zusätzlich die **Übersichtlichkeit** beeinträchtigt. Bestehen **Vererbungsbeziehungen** zwischen Knoten- oder Kantentypen, wie es als Konzept der objektorientierten Programmierung bekannt ist, kann die Matrix-Darstellung nicht mehr zur Definition des Metamodells verwendet werden (HELMS 2013, S. 44). Auch eine **Attribuierung** der Knoten und Kanten ist mit matrixbasierten Modellierungsansätzen nur bedingt möglich. So ist es möglich Namen, IDs und Kommentare zu vergeben. Eine große Anzahl von Attributen, die für die Analyse herangezogen werden können, kann nicht angelegt werden.

Durch die Ausdrucksfähigkeit der **Web Ontology Language** und der Verfügbarkeit praktischer Rechnerwerkzeuge wie Protégé (GENNARI ET AL. 2003), haben Ontologien auch in der Produktentwicklung weite Verbreitung gefunden. Mit der Abfragesprache SPARQL<sup>22</sup> können graphenbasierte Abfragen generiert werden. Die Ergebnisse der Abfragen lassen sich zwar modifizieren, ein regelbasierter Eingriff in die Daten und Datenstruktur ist allerdings nur sehr eingeschränkt möglich (FEIGENBAUM 2014). Die Formulierung von Abfragen erfordert zudem ein tiefes Verständnis der jeweiligen Ontologie und umfangreiches Expertenwissen (HELMS 2013, S. 36). Des Weiteren müssen im Produktportfoliomanagement zum Teil große Graphen mit großen Datenmengen verarbeitet werden. Die verfügbaren Rechnerwerkzeuge sind dafür allerdings nicht entwickelt worden. Von einer weiteren Verwendung wurde daher abgesehen.

Die Definition einer **eigenen domänenspezifischen Modellierungssprache** bietet die nötigen Freiheiten und Leistungsfähigkeit, die Methodik zur Mustererkennung in komplexen Produktstrukturen zu entwickeln. Dafür bietet sich eine eigene formale Beschreibung der Portfoliostruktur an. Es müssen keine Hilfskonstrukte bemüht werden. Es bestehen keinerlei Einschränkungen in der Anzahl und Verschiedenartigkeit der Elemente, Verknüpfungen und Attribute und es können leistungsfähige Modellierungs-, Analyse- und Visualisierungswerkzeuge eingesetzt werden.

## 4.2 Graphtransformation

Der Graph eines komplexen Produktportfolios kann schnell unübersichtlich werden, wenn mehrere tausend Bausteine, Produkte, Funktionen, Anforderungen, Verantwortlichkeiten etc. darin abgebildet sind. Schon ab einer Anzahl von mehr als hundert Knoten werden Visualisierungen am Bildschirm unlesbar. Der Graph im Ganzen wird daher nur als

---

<sup>22</sup> SPARQL Protocol And RDF Query Language (SPARQL)



Datenbasis<sup>23</sup> herangezogen. Bevor jedoch die Visualisierung erfolgt, wird die Graphtransformation vorgeschaltet. Mit ihr kann eine problemgerechte Sicht auf die Zusammenhänge erzeugt werden, indem unnötige Information ausgeblendet und wesentliche Zusammenhänge aufbereitet und hervorgehoben werden.

Die Graphtransformation ist ein Forschungsgebiet in der Informatik. In dieser Disziplin werden Techniken für die Handhabung graphenbasierter Strukturen entwickelt. Der **Begriff** wird im Englischen auch mit „Graph Rewriting“ übersetzt. Im Deutschen wird außerdem noch der Begriff Graphersetzungssystem verwendet. Entstanden ist die Graphtransformation aus dem Bedürfnis, die klassischen Ansätze wie Chomsky Grammatiken (CHOMSKY 1956) und seine Methode zur Begriffstransformation („Term Rewriting“) aus der Sprachenforschung auf nicht-lineare Strukturen auszuweiten (HECKEL 2006, S. 188).

Eine Graphtransformation ist eine formale, generative Methode, die aus einem Vokabular (also das Metamodell) und einem Regelwerk besteht. Das Vokabular umfasst alle validen Elemente, also Knoten und Kanten. Das Regelwerk definiert alle Transformationen, wie diese Elemente generiert, gelöscht oder verändert werden können (HELMS 2013, S. 55ff).

In der Literatur finden sich zahlreiche **Ansätze im Ingenieursumfeld**, in denen die Graphtransformation mit unterschiedlichen Fragestellungen erfolgreich eingesetzt wurde. HELMS (2013) und HELMS & SHEA (2012) nutzen die Graphtransformation im Rahmen von Computational Design Synthesis. Durch Entwicklung von Graph Grammatiken können Gestaltungsregeln formalisiert werden. Diese Regeln werden dann angewendet, um synthetische Lösungen anforderungsgerecht vom Rechner generieren zu lassen. Dadurch kann ein sehr großer Lösungsraum von mehreren tausend Lösungen erzeugt, untersucht, bewertet und somit die optimalste Lösung darunter gefunden werden. Dieses Verfahren setzen sie u.a. zur Erzeugung von Flugzeugkabinenlayouts und Getriebekonfigurationen ein.

Im **Kontext des Produktportfoliomanagements** finden sich die Arbeiten von DU ET AL. (2002b); DU ET AL. (2002a) und SIDDIQUE & BODDU (2005). Den Arbeiten ist gemein, dass sie Graphersetzungssysteme für die Gestaltung und Generierung von neuen variantenreichen Produkten einsetzen. Die Ansätze werden allerdings nicht dafür eingesetzt, den Umgang mit bestehenden, komplexen Strukturen zu unterstützen.

Das **Ziel dieser Arbeit** ist die Unterstützung des Systemarchitekten. Er soll sich durch höhere Transparenz, effektiveres und effizienteres Arbeiten in der Komplexität der bestehenden teils inkonsistenten Datenrepräsentation des Produktportfolios zurechtzufinden. Die Graphtransformation wird dazu genutzt, verteilte Daten in einem Graph zusammenzuführen. Wissen über repetitive Aufgaben wird in Regeln formalisiert und somit automatisch auf den Graphen angewendet. Des Weiteren wird die Graphtransformation eingesetzt, Such- und Fehlermuster zu definieren und automatisch in der Struktur des Graphen zu finden. Zur Unterstützung der Kommunikation komplexer Zusammenhänge wird die Graphtransformation verwendet, planungsrelevante Zusammenhänge intuitiv aufzubereiten und zu visualisieren.

---

<sup>23</sup> Vorteile einer graphenbasierten Datenbasis wie hohe Leistungsfähigkeit, Flexibilität und Agilität beschreiben ROBINSON ET AL. 2013

### 4.2.1 Typen und Instanzen

Modelle repräsentieren ein Abbild der Realität. *Konkrete Objekte* werden in *Konzepten* generalisiert (HECKEL 2006, S. 188). Diese generalisierten Konzepte bilden die *Typen*, die in einem *Metamodell* definiert werden. Konkrete Objekte in der Realität bilden dann *Instanzen* der Typen. Somit repräsentiert ein unveränderlicher *Typengraph* die Konzeptebene, also das Metamodell. Eine Momentaufnahme eines individuellen Graphs bildet dann einen möglichen *Instanzengraph*. Eine solche Unterscheidung findet sich als wiederkehrendes Muster in der objektorientierten Programmierung mit Klasse und Objekt, in der Datenbanktechnik mit Datenbankschema und Zustand sowie bei einem XML-Schema und einem Dokument, das in diesem Schema erstellt wird (HECKEL 2006, S. 190). Weitere Informationen finden sich u.a. bei EHRIG ET AL. (1999) und ROZENBERG & EHRIG (1997).

Nach ISO 11179-1 ist ein **Metamodell** ein Datenmodell, das ein oder mehrere Datenmodelle spezifiziert. Ein Metamodell definiert demzufolge die Bausteinarten, die für den Aufbau eines Produktportfolios benötigt werden. Somit ist das Metamodell mit der Spezifizierung von Knoten und Kantentypen die Basis für die Formulierung von Regeln für die Graphtransformation (HELMS 2013, S. 57).

Ein Knoten  $o:T$  repräsentiert ein Objekt  $o$  vom Typ  $T$ . Die Beziehung zwischen Typen- und Instanzen-Ebene wird durch folgende **Kompatibilitätsbedingungen** beschrieben (HECKEL 2006, S. 190):

- Für jeden Knoten  $o:T$  im Instanzengraph muss es einen Typenknoten  $T$  im Typengraph  $TG$  geben.
- Für jede Kante zwischen Objekten  $o_1:T_1$  und  $o_2:T_2$  muss es einen korrespondierenden Kantentyp im Typengraph zwischen den Knotentypen  $T_1$  und  $T_2$  geben.
- Für jedes Attribut  $a = v$  eines Objektes  $o:T$  in einem Instanzengraph, muss es eine korrespondierende Deklaration  $a:DT$  im Knotentyp  $C$  geben, so dass der Wert  $v$  vom Datentyp  $DT$  ist.

Im Aufbau eines Metamodells kann das **Vererbungsprinzip** angewendet werden. Dadurch lassen sich Metamodelle ineinander verschachteln und „dienen zur ökonomischen Datenhaltung“ (PUPPE 1988, S. 29). Es bildet die Basis dafür, eine strukturierte Definition der Bausteine zu realisieren (HELMS 2013, S. 57). KOMOTO & TOMIYAMA (2010) haben dieses Prinzip erfolgreich eingesetzt und gezeigt, dass sich kaskadierte Metamodelle für die Abbildung sehr großer Wissensbasen im Ingenieursumfeld eignen.

Das Vererbungsprinzip lässt sich auch für eine strukturierte **Attribuierung** von Knoten- und Kantentypen nutzen (HELMS 2013, S. 31). In übergeordneten abstrakten Klassen können grundlegende Attribute und Default-Werte definiert werden, die dann in den konkreten Kinderknotenklassen vererbt werden. Dadurch sinkt der Aufwand bei der Erstellung wie auch bei der Pflege des Metamodells. Mit generischen Regeln können generische Konzepte in der Vererbungshierarchie verarbeitet werden. Dies erleichtert die Beschreibung von Mustern.

### 4.2.2 Regeln und Transformationen

Die Prinzipien, mit denen **Regeln** aufgebaut werden, folgen denen von Bedingungssätzen der natürlichen Sprache (BEIERLE & KERN-ISBERNER 2008, S. 73). Expertenwissen wird daher auch meistens in Regelform formuliert (PUPPE 1988, S. 21). Regeln eignen sich daher, heuristisches Wissen abzubilden (DYM & LEVITT 1991, S. 151), wie es in vielen ingenieurwissenschaftlichen Disziplinen vorliegt (HELMS 2013, S. 29). Die Logik ist einfach, flexibel und robust (RUDE 1998, S. 72). Durch die Aufbereitung von Expertenwissen „in möglichst kleine eigenständige ‚Wissensstücke‘ macht eine Wissensbasis modular und damit leicht veränderbar“ (PUPPE 1988, S. 21). Weitere Vorteile durch die Formalisierung von Wissen mit Regeln sieht PUPPE (1988, S. 21) beim Umgang mit unsicherem und unvollständigem Wissen.

Dennoch kann die Formalisierung von Expertenwissen durch Formulierung von Regeln aufwendig sein (HELMS 2013, S. 29). Die Argumentationslogik in verketteten Regeln zur Lösung komplexer Probleme kann schwierig nachzuvollziehen sein (RUDE 1998, S. 72). Die Pflege eines sehr großen Regelwerks kann sich sehr mühevoll erweisen, vor allem wenn Regeln ineinander verstrickt sind („Regelspaghetti“) (PUPPE 1988, S. 26). Mittels generischer Regeln und logischer Konstrukte zur Ablaufsteuerung kann dies aber weitgehend vermieden werden.

Trotz möglicher Nachteile eignen sich Regeln gut dazu, logische Abläufe, Wissen und Methoden der Systemarchitekten zu formalisieren.

Das Kernstück der **Graphtransformation** ist die Spezifikation von Transformationen instanzierter Graphen anhand von Regeln (HECKEL 2006, S. 191). Es können also Operationen (Regeln) auf alle Graphen angewendet werden, die dem im Typengraphen spezifizierten Metamodell entsprechen.

Formal besteht nach HECKEL (2006, S. 191f) eine Graphtransformations-Regel  $p: L \rightarrow R$  aus dem Namen  $p$  und einem Paar von Instanzgraphen über  $TG$ , deren Struktur kompatibel ist. Kompatibel heißt, Knoten mit derselben Identität in  $L$  und  $R$  haben denselben Typ und dieselben Attribute; Kanten mit derselben Identität sind vom gleichen Typ und haben dieselben Ursprungs- und Zielknoten. Die Linke Seite  $L$  repräsentiert das gesuchte Muster (*pre-condition*) der Regel, die Rechte Seite  $R$  beschreibt die Aktion (*post-condition*).

Regeln sind die Generalisierungen von Transformationen. Sie generieren Transformationen, indem sie Muster, die in der Linken Seite definiert und im Instanzgraph gefunden wurden, mit einer Kopie des Musters, das in der Rechten Seite definiert wurde, ersetzen. Wird das Muster der Linken Seite im Graph nicht gefunden, wird keine Transformation ausgeführt (HELMS 2013, S. 59). Eine Graphtransformation von einem Vor-Zustand  $G$  zu einem Nach-Zustand  $H$  wird formal notiert mit  $G \Rightarrow H$  (HECKEL 2006, S. 192). Nachfolgende Abbildung zeigt den Aufbau einer Regel anhand eines einfachen Beispiels. In einem Graph wird eine Knotenkette aus einem blauen Dreieck, einem grünen Quadrat und einem orangenen Kreis gesucht. Wenn das Muster gefunden wird, soll das blaue Dreieck durch ein rotes Achteck ersetzt werden.

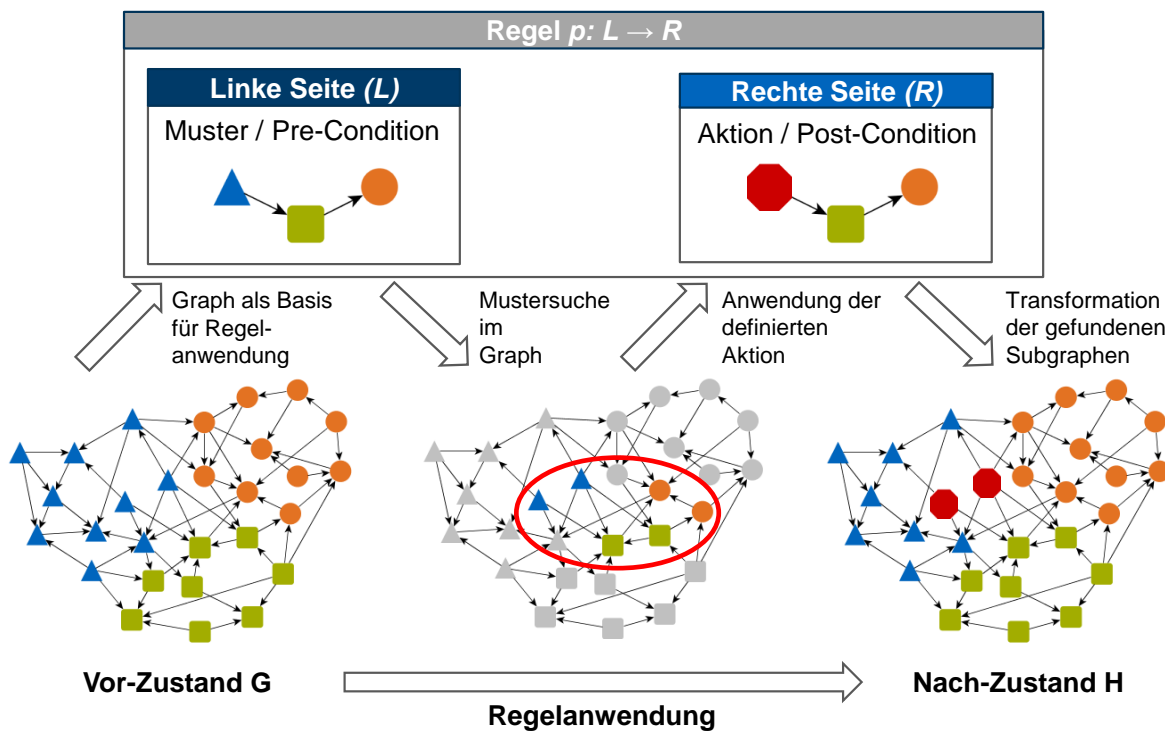


Abbildung 4-3: Graphischer Aufbau einer Regel

Die Schnittmenge zwischen L und R ist der **Kontext**  $K (L \cap R)$ . Der Kontext bildet den Ort der Transformation und bleibt selbst unverändert. Er beinhaltet alle Elemente, die durch die Regelanwendung unberührt bleiben. Somit werden Elemente der Linken Seite, die nicht Teil von K sind, gelöscht; Elemente der Rechten Seite, die nicht in K sind, werden hinzugefügt. Diese mengenbasierte Definition der Graphtransformation wird auch Algebraischer Ansatz zur Graphtransformation genannt (CORRADINI ET AL. 1996).

Bei der Definition von Regeln kann man sich einiger Vorteile der Programmierung bedienen, die zur Flexibilität und Ausdrucksstärke beitragen (HELMS 2013, S. 59):

- Regeln können mit Eingangsvariablen aufgerufen werden.
- Regeln können Rückgabewerte als Ergebnis ausgeben, die für weitere Operationen genutzt werden können. Standardmäßig wird das Ergebnis der Regelanwendung zurückgegeben. Wenn also das Muster der Linken Seite nicht gefunden und die Regel nicht angewendet wurde, wird *False* ausgegeben, ansonsten *True*.
- In Regeln können Bedingungen für die Mustersuche oder für die Evaluierung von Attributen gesetzt werden. Die Definition der Bedingungen in Regeln bedient sich dabei einer programmierähnlichen Sprache (vgl. GEIß ET AL. 2006).

Nachfolgend wird beispielhaft die Suche von Elementen nach bestimmten Kriterien und Veränderung von Attributen formuliert:

1. Suche  $\langle \text{ein Set von Knoten} \rangle$ , das die  $\langle \text{Bedingungen } 1-n \rangle$  erfüllt und lösche die übrigen Knoten und Kanten.
2. Ersetze den Wert  $v$  des  $\langle \text{Attribut } a \rangle$  jedes Knotens durch das Ergebnis der  $\langle \text{Operation } f(v) \rangle$
3. Gebe die Knotennamen mit den Werten in einer Liste aus.

In einer Regel formuliert, die für die Graphtransformation genutzt werden kann, lautet dann die Regel (Code vereinfacht):

```
rule BedingteSuche {  
  //LINKE SEITE  
  Knoten  
  IF (Bedingung 1, ..., Bedingung n);  
  //RECHTE SEITE  
  Knoten.a = f(v);  
  EMIT (Knoten.name + Knoten.a);}
```

Auf der Linken Seite werden Knoten gesucht, die die Bedingungen der *IF*-Klammer erfüllen. Wenn dieses Muster gefunden wurde, wird in der Rechten Seite die Transformation definiert. Zum einen soll dem Attribut *a* ein Neuberechneter Wert zugewiesen werden. Zum anderen soll anschließend der Wert des Namensattributs mit dem neuen Wert für *a* ausgegeben werden. Diese Regel kann nun sofort aufgerufen werden bis das Muster nicht mehr gefunden wird.

Regeln können verkettet werden. Mit logischen Operatoren können Regeln nacheinander ausgeführt werden. Je nach Rückgabewert, ob eine Regel angewendet werden konnte oder nicht, können dann weitere Regeln nachgeschaltet werden. Somit können Konstrukte zur Ablaufsteuerung aufgebaut werden. Die Ausführung der **Regelsequenzen** erfolgt durch einen Interpreter (HELMS 2013, S. 63), der die Anweisungen abarbeitet und die Transformationen durchführt.

### 4.2.3 Softwaretechnische Umsetzung

Um die Graphtransformation für das Produktportfoliomanagement nutzbar zu machen, wird hier exemplarisch das Software-Werkzeug **Soley® Studio**<sup>24</sup> eingesetzt. Soley Studio ist eine Weiterentwicklung der Software **Booggie**<sup>25</sup>, die von HELMS (2013) im Rahmen seiner Forschungsarbeit für die Design Synthese entwickelt wurde. Soley Studio erweitert Booggie unter anderem um Funktionalitäten wie Graphvisualisierung, leistungsfähigere Modellierungs- und Analysebausteine.

Die Grundbausteine zur Graphtransformation sind in Booggie enthalten. Booggie nutzt GrGen.NET<sup>26</sup> als Kernbibliothek zur Graphtransformation. Genauere Details zu Anpassungen von GrGen.NET, Definition und Aufbau der Softwarearchitektur von Booggie finden sich bei

---

<sup>24</sup> Soley Studio ist ein Softwarewerkzeug der Firma Soley, einer Ausgründung des Lehrstuhls für Produktentwicklung der Technischen Universität München (KISSEL & HELMS 2014); [www.soley-technology.com](http://www.soley-technology.com) (entnommen am 15.11.13)

<sup>25</sup> Brings object-oriented graph grammars into engineering (Booggie), [www.booggie.org](http://www.booggie.org) (entnommen am 15.11.13)

<sup>26</sup> Graph Rewrite Generator auf .NET-Technologie (GrGen.NET). Weitere Informationen finden sich unter <http://www.info.uni-karlsruhe.de/software/grgen/> (entnommen am 15.11.13) und bei BLOMER ET AL. 2013

HELMS (2013, S. 95ff). Die hier verwendete Entwicklungsumgebung ermöglicht es, eigene Metamodelle zu definieren, Graphen zu generieren, Regeln zu editieren und Transformationen auf Graphen anzuwenden. Die Funktionalitäten von Soley Studio werden dafür genutzt, Analysen großer Graphen leistungsfähiger durchzuführen und die Ergebnisse in der gewünschten Form zu visualisieren. Der konzeptionelle Aufbau der Entwicklungsumgebung dieser Forschungsarbeit ist in Abbildung 4-4 zusammengefasst.

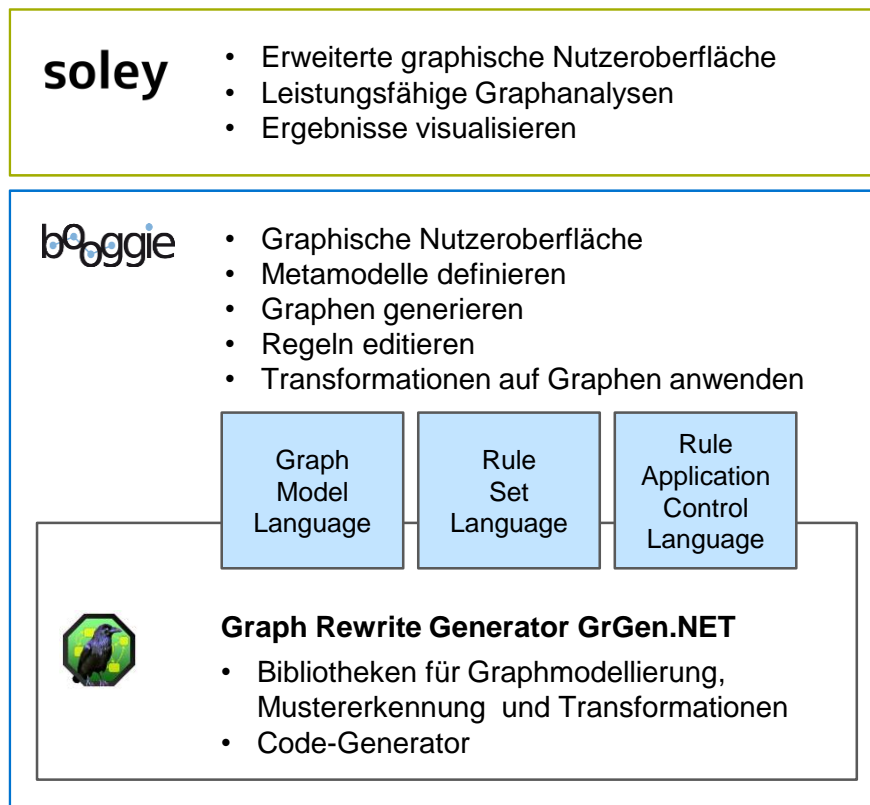


Abbildung 4-4: Konzeptioneller Aufbau der Entwicklungsumgebung dieser Forschungsarbeit

In Booggie werden essentielle Elemente von GrGen.NET nutzbar gemacht. GrGen.NET bieten die softwaretechnische Infrastruktur, um die entwickelten Modelle umzusetzen. Drei Modellierungssprachen wurden zur Verwendung von GrGen.NET definiert (HELMS 2013, S. 55/107):

- *Graph Model Language*: die Sprache ermöglicht die Spezifizierung von Metamodellen typisierter attribuerter Multigraphen mit multipler Vererbung über Knoten und Kanten und die Berücksichtigung abstrakter Typen.
- *Rule Set Language*: die Sprache kann zur Definition von Regeln verwendet werden. Sie deckt die Musterspezifikation (Linke Seite) und die Ersetzungsspezifikation (Rechte Seite) ab. Außerdem erlaubt sie Attribute-Operationen in der Regelnanwendung.
- *Rule Application Control Language*. Die Sprache wird verwendet, um komplexe Graphtransformationen in einzelnen Graphtransformationsregeln zusammenzufassen. Dadurch müssen Regeln nicht einzeln ausgeführt werden, sondern können in komplexen Kontrollstrukturen – sog. Regelsequenzen – zusammengefasst werden.

GrGen.NET ist ein generatives Programmiersystem (BLOMER ET AL. 2013), d.h. aus den Eingaben des Nutzers wird mittels eines Generators künstlich Code erzeugt. Der Nutzer kann die erforderliche Transformation also auf einer höheren Abstraktionsstufe mit Knoten und Kanten modellieren, Transformationsregeln definieren und Modifikationen durchführen. GrGen.NET erzeugt und optimiert dann die Algorithmen und generiert ein Programm in C#. Sowohl Graph als auch Regelwerk liegen dann als ausführbare Datei vor. Dies erlaubt eine performante Abarbeitung und Anwendung der Regeln auch auf sehr große Graphen. Einen Benchmark, der die Schnelligkeit von GrGen.NET aufzeigt, findet sich bei GEIß & KROLL (2007).

Die Vorteile der objektorientierten Programmierungskonzepte in GrGen.NET, die auch der Methodik zur Mustererkennung im Produktportfoliomanagement zu Gute kommen, beschreibt HELMS (2013, S. 55):

- Das Metamodell und alle darin definierten Knoten- und Kantentypen und Attribute sind ebenso wie das Regelwerk erweiterbar und anpassbar für neue Anwendungen. Dadurch ist eine Weiterentwicklung der Methodik stets realisierbar.
- Teile des Metamodells oder als Ganzes sind wiederverwendbar. Dies gilt auch für die Regeln.
- Typendefinitionen und Regeln anderer Domänen können verwendet werden, wenn sie den Kompatibilitätsbedingungen im Metamodell gerecht werden.
- Die Formalisierung von Wissen wirkt sich effizienzsteigernd auf IT- und Humanressourcen aus.
- Der Aufbau von Metamodell und Regelwerken sind intuitiv verständlich und somit einfach zu ergänzen.

Die eingesetzte Entwicklungsumgebung erlaubt eine unkomplizierte Entwicklung einer eigenen domänenspezifischen Modellierungssprache für das Produktportfoliomanagement.





## 5. Methodik zur Mustererkennung in komplexen Produktportfolios

In diesem Kapitel wird die Methodik zur Mustererkennung in komplexen Produktstrukturen beschrieben. Ziel der Methodik ist die Überwindung der in Kapitel 3.3.2 ausführlich behandelten Hürden der Zielerreichung. Dem Systemarchitekten soll es somit möglich sein, durch die Verwendung der vorgestellten Methoden die **Transparenz** über die Struktur im Produktportfolio zu **erhöhen**, Arbeitsschritte wesentlich schneller (**Effizienzsteigerung**) durchführen zu können und mit Hilfe der Mustersuche eine Verbreiterung der einsetzbaren Methoden zu erreichen (**Effektivitätssteigerung**).

In Kapitel 2.2.4 wurden die Voraussetzungen für eine erfolgreiche Systemarchitektur beschrieben. Mit begrenzten Ressourcen und hohem Zeitdruck muss der Systemarchitekt in einem dynamischen Umfeld Änderungen einpflegen, Fehler erkennen, Entscheidungen treffen und unterstützen. Hierfür muss er den **Überblick über das Produktportfolio** behalten, eine **klare und konsistente Architektur** pflegen, um somit seine Ergebnisse **stakeholder-gerecht kommunizieren** zu können. Hierfür benötigt er performante Werkzeuge, Methoden und eine konsistente Datenbasis. Methoden des Komplexitätsmanagements und zur Reduzierung der Variantenvielfalt nehmen zum Teil starke Vereinfachungen an. Es werden oft lediglich begrenzte Produktumfänge betrachtet, man geht von einer gut gepflegten und breiten Datenbasis aus, vernachlässigt den **Initialaufwand der Datenbeschaffung** oder lässt **Wechselwirkungen mit anderen Prozessen oder Stakeholdern unberücksichtigt** (vgl. Kapitel 3.2).

Daher wird mit der hier beschriebenen Methodik versucht, Verbesserungen für den Systemarchitekten herbeizuführen. Die Methodik gliedert sich in **vier Methoden** auf. Nachfolgende Tabelle 5-1 zeigt auf, welche Methode welche Anforderungen (vgl. Kapitel 3.4.1) maßgeblich erfüllen soll:

Methoden	Anforderungen
Zusammenführung relevanter Daten in graphenbasierter Form	(A1), (A2)
Formalisierung repetitiver Aufgaben im Produktportfoliomanagement	(A3), (A6)
Automatische Muster- und Fehlererkennung	(A4)
Intuitiv verständliche Visualisierung relevanter Zusammenhänge	(A5)

Tabelle 5-1: Verknüpfung von Methoden des Ansatzes mit den Anforderungen

## 5.1 Rahmenwerk der Methodik

### 5.1.1 Aufbau

Die einzelnen Bausteine der Methodik zur Mustererkennung in komplexen Produktstrukturen zeigt Abbildung 5-1. Sie sind auf den zwei Ebenen Datenverarbeitung und Wissensformalisierung organisiert.

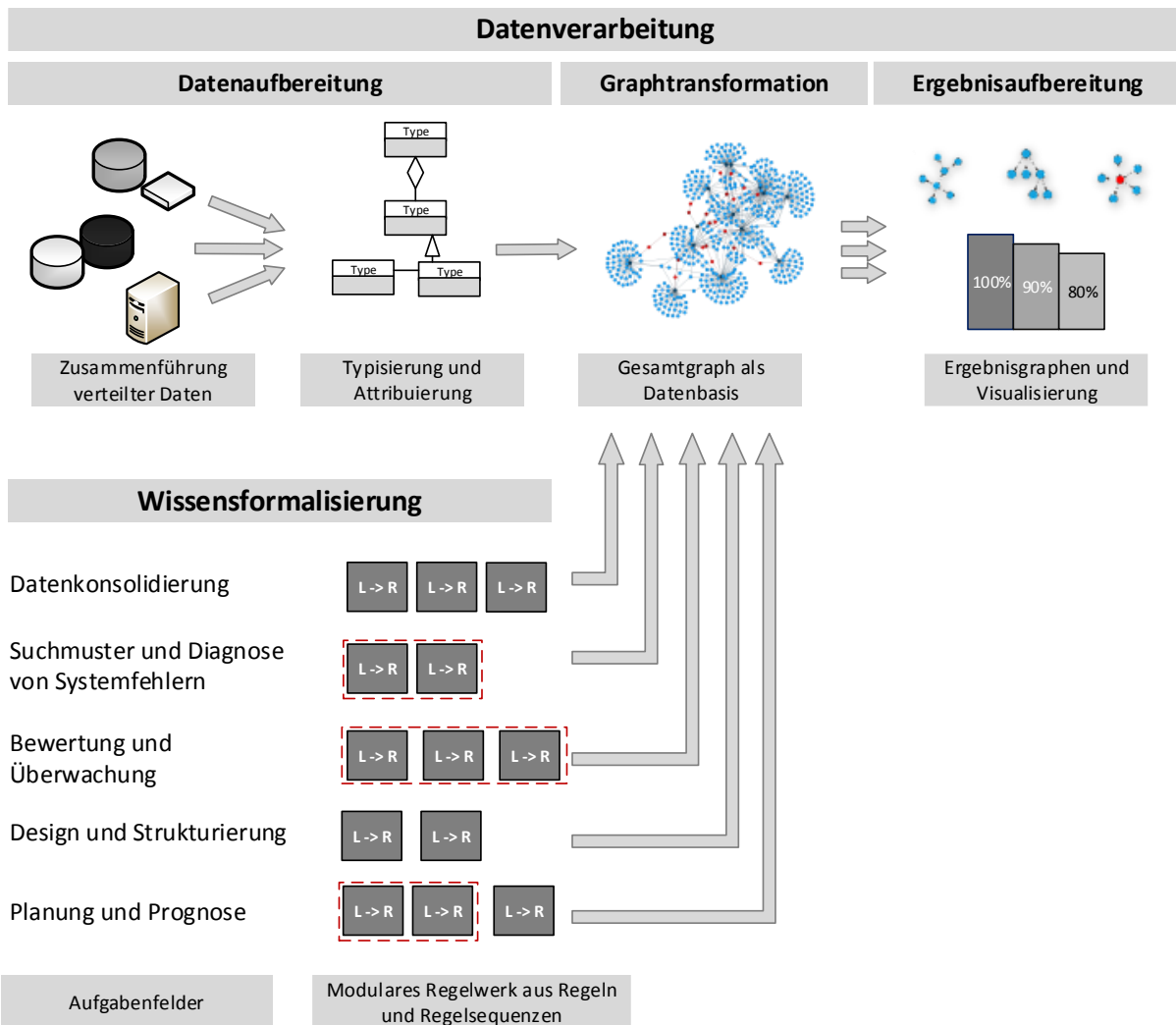


Abbildung 5-1: Rahmenwerk der Methode zur Mustererkennung in komplexen Produktstrukturen

Prämisse des Lösungsansatzes ist die **Entkoppelung von Datenhaltung, Wissensformalisierung und Visualisierung der Ergebnisse**. Dies erhöht die Anpassbarkeit und Konsistenz der Datenhaltung, die Erweiterbarkeit der Wissensformalisierung und die Flexibilität der Ergebnisaufbereitung.

Auf der **Ebene der Datenverarbeitung** erfolgt zunächst die Datenaufbereitung. In der Methode zur Zusammenführung planungsrelevanter Daten in graphenbasierter Form werden im Pre-Processing verteilte Daten, die für die Entscheidungsfindung relevant sind,

zusammengefasst und über ein vorgegebenes Importformat eingebunden. Im Metamodell für das Produktportfoliomanagement werden die Modellierungskonstrukte definiert, d. h. welche Elementtypen wie attribuiert und wie verknüpft werden können. Sobald das Modell des Produktportfolios instanziiert wurde, liegt ein Graph vor, auf den Regeln und Regelsequenzen angewendet werden können.

Auf der **Ebene der Wissensformalisierung** können nun ausgehend vom individuellen Zielesystem eines Unternehmens Regelwerke aufgebaut werden. In den Methoden zur Formalisierung repetitiver Aufgaben und zur automatischen Muster- und Fehlererkennung kann Expertenwissen in Regelform beschrieben werden. PUPPE (1988, S. 9f) beschreibt typische Problemlösungstypen in Expertensystemen (vgl. auch STEFIK ET AL. 1982, S. 136ff):

- Interpretation: Ableitung von Situationsbeschreibungen aus Kennwerten
- Diagnostik: Ableitung von Systemfehlern aus Beobachtungen
- Überwachung: Vergleich von Beobachtungen mit Sollwerten
- Design: Konfigurierung von Objekten unter Berücksichtigung besonderer Anforderungen
- Planung: Entwurf einer Folge von Aktionen zum Erreichen eines Ziels
- Vorhersage: Ableitung von möglichen Konsequenzen gegebener Situationen

Eine ähnliche Einteilung findet sich auch bei CLANCEY (1985). Der hier entwickelte Ansatz orientiert sich an diesen Problemlösungstypen. Zusätzlich wird die Datenkonsolidierung eingefügt, die weniger ein Problemlösungstyp ist, aber in der Auflistung als wichtiger Wissensbaustein zu besonders repetitiven Aufgaben mitaufgeführt werden soll. Zur Unterstützung des Systemarchitekten werden somit folgende fünf Regelkategorien definiert:

- Datenkonsolidierung
- Suchmuster und Diagnose von Systemfehlern
- Bewertung und Überwachung
- Design und Strukturierung
- Planung und Prognose

In folgenden Kapiteln werden die Kategorien näher erläutert und Beispiele gegeben. Die jeweiligen Regelwerke können je nach Bedarf auf den Graph angewendet und Ergebnisgraphen erzeugt werden. Je nach Zweck der Entscheidungsunterstützung werden die Ergebnisse derart dargestellt, dass Handlungsfelder erkennbar sind und Maßnahmen abgeleitet werden können. Die Methode zur intuitiven Visualisierung relevanter Zusammenhänge wird in Kapitel 5.4 beschrieben.

### 5.1.2 Situative Anpassung der Methodik

Bevor die Methodik sinnvoll eingesetzt werden kann, ist es essentiell, die Problemstellung im Unternehmen zu verstehen und Lösungsstrategien entsprechend anzupassen. Wenn Such- und Fehlermuster definiert, ein Produktportfolio bewertet oder überwacht, Eingriffe in die Struktur durchgeführt oder Planungen erstellt werden sollen, müssen vorab die Randbedingungen geklärt sein. Die Unternehmensstrategie und die angebotenen Produktarten eines Unternehmens haben großen Einfluss darauf, wie das individuelle Vorgehen auszugestalten ist

und welche Schwerpunkte zu setzen sind. Somit müssen die verwendeten Regeln situativ entsprechend angepasst werden. Vor dem Einsatz der Methodik bietet es sich daher zur Vorbereitung an, mit den entsprechenden Verantwortlichen, Experten und Schlüsselpersonen im Unternehmen in Workshops grundsätzliche Fragen zu klären. Beispielhaft werden hier Fragen als Leitfaden zur Klärung der Problemstellung aufgeführt:

- Welchen Einfluss hat die **Art der Produkte** auf die Entscheidungen? Es können unterschiedliche Strategien verfolgt werden, wenn Produkte bei jedem Auftrag neu konfiguriert oder wenn sie auf Vorrat produziert werden (vgl. „engineered-to-order“, „assembled-to-order“, built-to-stock“-Konzepte und deren Derivate). Einfluss auf mögliche erzielbare Skaleneffekte haben auch Stückzahlen, einsetzbare Fertigungstechnologien, Qualitätsanforderungen und viele weitere.
- Spielt die **Lebenszyklusphase** der betrachteten Produkte eine Rolle? Muss zwischen Neuentwicklungen, vermarktbareren Produkten oder Produkten, die am Ende ihres Lebenszyklus stehen, unterschieden werden?
- Welche **Stakeholder** müssen eingebunden werden? Welche Entscheidungen müssen diese treffen und auf Basis welcher Informationen?
- Wer stellt die **benötigten Informationen** bereit und in welcher Form? Welche Daten werden im Unternehmen gespeichert und gepflegt? Welche Informationen müssen zusätzlich erhoben werden? Welchen unmittelbaren Nutzen haben die Personen im Unternehmen davon, dass sie die Daten speichern, pflegen und bereitstellen?
- Inwieweit müssen **bestehende Produkte und Strukturen** berücksichtigt werden? Kann ein Teilsystem des Produktportfolios isoliert vom Rest betrachtet werden? Müssen Wechselwirkungen zum Umfeld berücksichtigt werden?
- Muss die **Veränderung über die Zeit** und Anpassungen an sich **verändernde Rahmenbedingungen** berücksichtigt werden? So können Bausteine, die ursprünglich mal als Sonderbausteine konzipiert und klassifiziert waren, mit der Zeit ihre Bedeutung in den Produkten verändern. Ebenso wie sich Produkteigenschaften, wie bei KANO (1995) beschrieben, über die Zeit von Begeisterungsmerkmalen zu Standardmerkmalen ändern können, können auch Sonderbausteine zu Varianten oder zu Grundbausteinen werden.

Die Antworten auf diese und weitere Fragen können im jeweiligen Unternehmenskontext sehr unterschiedlich ausfallen. Ein **universal gültiges „Kochrezept“** kann die hier vorgestellte Methodik daher nicht sein, sondern erfordert eine **situative Anpassung** bei Verwendung.

Mit diesen Eingangsinformationen wird auf Ebene der Wissensformalisierung ein individuelles Untersuchungsziel festgelegt und daraus geeignete Such-, Fehlermuster, Transformationsbedingungen und Zielkennwerte festgelegt. Beispielsweise spielt eine hohe Wiederholteilerate bei variantenreichen Produkten mit tendenziell niedrigeren Stückzahlen je Variante eine bedeutendere Rolle für die Bewertung des Produktportfolios als bei Unternehmen, die große Stückzahlen mit kleiner Varianz eines Produktes produzieren. Je nachdem welche Ziele mit dem Einsatz der Methodik erreicht werden sollen, kann entweder auf der bestehenden Datenbasis mit der Datenaufbereitung und Analyse aufgesetzt werden oder es müssen zusätzliche Informationen eingeholt werden.

## 5.2 Datenaufbereitung

### 5.2.1 Zusammenführung aus verteilten Datenquellen

Mit dem Metamodell für das Produktportfoliomanagement kann eine breite Spanne von heterogenen Informationen in den Graph überführt werden. Natürlich kann es sein, dass bestimmte Elementtypen von Unternehmen nicht dokumentiert vorliegen. An dieser Stelle muss dann entschieden werden, mit welchem **Aufwand** die Daten erhoben werden können und wie groß der **Nutzen der Datenerhebung** auch im Sinne der Zielstellung der Untersuchung ist.

Außerdem können die Daten in einer **anderen Form und Detaillierung** vorliegen, als es das Metamodell erfordert. Beispielsweise wird in einem Stücklistenexport nicht zwangsläufig explizit zwischen Produkten, Baugruppen oder Teilen unterschieden. Es wird lediglich eine Liste mit Namen und Sachnummern ausgegeben. Durch die Definition eines **Import-Knotentyps** im Metamodell können diese Daten zunächst „neutral“ in den Graph importiert werden. Durch Retypisierungsregeln wie sie in Kapitel 5.3.4 erläutert werden, können diese Daten dann nachträglich korrekt typisiert werden.

Als **Informationsquellen** eignen sich eine Vielzahl von Datenbanken und weiterer Datenquellen im Unternehmen. MALMQVIST (2002, S. 208f) gibt einen Überblick über Produktmodellierungsmethoden, die ebenfalls als Datenquelle genutzt werden können, wenn sie im Unternehmen eingesetzt werden (z. B. Axiomatic Design Matrizen, Produktbewertungen, Komponenten-DSMs etc.). Folgende Aufzählung zeigt Ansatzpunkte auf, hat aber keinen Anspruch auf Vollständigkeit.

- Datenbank-gestützte Systeme
  - Produktdatenmanagement Systeme (PDM)
  - Product-Lifecycle Management Systeme (PLM)
  - Enterprise-Resource-Planning Systeme (ERP)
  - Anforderungsmanagement-Systeme
  - Individuelle datenbankgestützte Systeme
- Projektdokumentationen
  - Projekt-Wikis
  - Wissensdatenbanken
  - Ausgefüllte Methodenformblätter, wie z. B. QFD, FMEA oder Einflussmatrizen
  - Berichte
  - Präsentationen
- Individuelle Abfragen
  - Interviews
  - Expertenworkshops

Viele entscheidungsrelevante Daten liegen im Unternehmen verteilt auf lokalen Festplatten oder Projektlaufwerken. Oft werden wichtige Informationen in Office-Anwendungen dokumentiert und nicht immer zentral abgelegt. Moderne Produktdaten- und Product-Lifecycle Management Systeme (PDM/PLM) bilden mittlerweile eine breite Datenbasis ab. Umfangreiche Produktdaten werden zentral in Datenbanksystemen abgelegt. Sie integrieren

z. B. eine Anforderungs-, Funktions-, Logische und Physikalische Ebene („RFLP“-Ansatz bei Dassault und Siemens). Zudem bilden die Systeme auch Tracing-Beziehungen zwischen unterschiedlichen Elementtypen ab, wie zwischen Anforderungen und Komponenten. Wenn derartige Datenstrukturen konsistent gepflegt sind, eignen sie sich hervorragend für die Instanziierung des Produktportfolio-Graphen. Allerdings verbreiten sich diese Ansätze in der Industrie noch langsam.

### 5.2.2 Aufbau des Metamodells für das Produktportfoliomanagement

Die Definition einer **eigenen domänenspezifischen Sprache** (DSL) mit der *Graph Model Language* ermöglicht die Spezifizierung von Metamodellen typisierter attributierter Multigraphen mit multipler Vererbung über Knoten und Kanten und die Berücksichtigung abstrakter Typen (vgl. Kapitel 4.1.2). Somit können relevante Informationen aus unterschiedlichen, verteilten Datenquellen in einen Graphen überführt und somit für Graphtransformationen genutzt werden.

Mit diesen Eigenschaften bietet eine eigene DSL die Möglichkeit, alle wesentlichen Aspekte der Produktportfolioarchitektur abzubilden. Zudem bietet sie die nötige Flexibilität, auf firmenindividuelle Besonderheiten zu reagieren. So kann das **Metamodell** an spezifische Belange **angepasst** werden. Zum einen kann es sein, dass bestimmte Daten, die im Metamodell als Elementtyp vorgesehen sind, nicht vorhanden sind oder nur mit Mühe bereitgestellt werden können. Zum anderen kann es im unternehmensspezifischen Analysekontext auch sinnvoll sein, zusätzliche Daten<sup>27</sup> heranzuziehen. In diesem Fall kann das Metamodell aufwandsarm erweitert werden. Unter Beachtung der Vererbungslogik können in Elementtypen und Relationen weitere Attribute aufgenommen bzw. neue Elementtypen und Relationen definiert werden.

Es ist nicht Ziel der Arbeit, ein universell vollständiges Metamodell für das Produktportfoliomanagement zu definieren. Der Anspruch ist es, im Rahmen der Methodik ein **Basis-Metamodell** anhand der Erkenntnisse aus dem Stand der Technik abzuleiten, das auch im weiteren Verlauf den Anforderungen der anzuwendenden Regel und Visualisierung gerecht wird. Nach PULM (2004, S. 147) ist dies im Sinne einer „individuellen Kontingenzschließung, d. h. nicht *das* richtige Produktmodell ist entscheidend, sondern *ein* Produktmodell, das handlungsfähig macht.“ Durch die Erweiterbarkeit und Anpassbarkeit kann das Metamodell um Spezialitäten individueller Problemstellungen entsprechend ergänzt werden, wenn die Typendefinitionen des Basis-Metamodells nicht ausreichen.

Sobald der Graph zum ersten Mal erfolgreich mit einem ggf. angepassten Metamodell instanziiert worden ist, können **Aktualisierungen der Datenbasis** aufwandsarm eingepflegt werden. Durch die Wiederverwendbarkeit und Übertragbarkeit können Regelwerke auf alle Graphen, die mit dem zugrundeliegenden Typengraph instanziiert wurden, angewendet werden. Mit dem Aufbau der Datenstruktur werden somit die Anforderungen (A1) und (A2) adressiert.

---

<sup>27</sup> In der Fallstudie der Elektronik GmbH kommt dies zum Einsatz.

## Grundelemente des Metamodells

An oberster Stelle in der Vererbungshierarchie dieses Metamodells stehen **abstrakte Elementtypen**. In abstrakten Elementtypen können nach dem Vererbungsprinzip Attribute gesammelt werden, die alle in der Hierarchie darunterliegenden Knoten- oder Kantentypen im Graph erben. Hier verfügen somit alle Knoten über Attribute, die zur Visualisierung benötigt werden und oder zur generellen Beschreibung des Knotens dienen. Das Gleiche gilt für die Kanten. Nachfolgende Liste fasst die abstrakten Knoten mit ihren Attributen zusammen:

- *Visueller Knoten* ( $x$ ,  $y$ ,  $label$ ,  $color$ ,  $height$ ,  $width$ ,  $diameter$ ,  $shape$ )
- *Portfolio Element* mit den Attributen ( $id$ ,  $name$ ,  $description$ ,  $time\_stamp$ ,  $type\_name$ ), der die Attribute von *Visueller Knoten* erbt
- *Portfolio Kante* ( $name$ ,  $weight$ ,  $color$ ,  $thickness$ )

In der nächsten Stufe erben die drei Elementtypen „Portfolio Knoten“, „Stakeholder“ und „logische Gruppen“ die Attribute der abstrakten Knoten. Diese drei Elementtypen sind wiederum übergeordnete Typen, die dann im Folgenden weiter untergliedert werden. Auf dieser Abstraktionsstufe können aufwandsarm die Kantentypen „is responsible“ („ist verantwortlich“) und „depend on“ („ist abhängig von“) zwischen Portfolio Knoten und Stakeholder definiert werden. Siehe Abbildung 5-2.

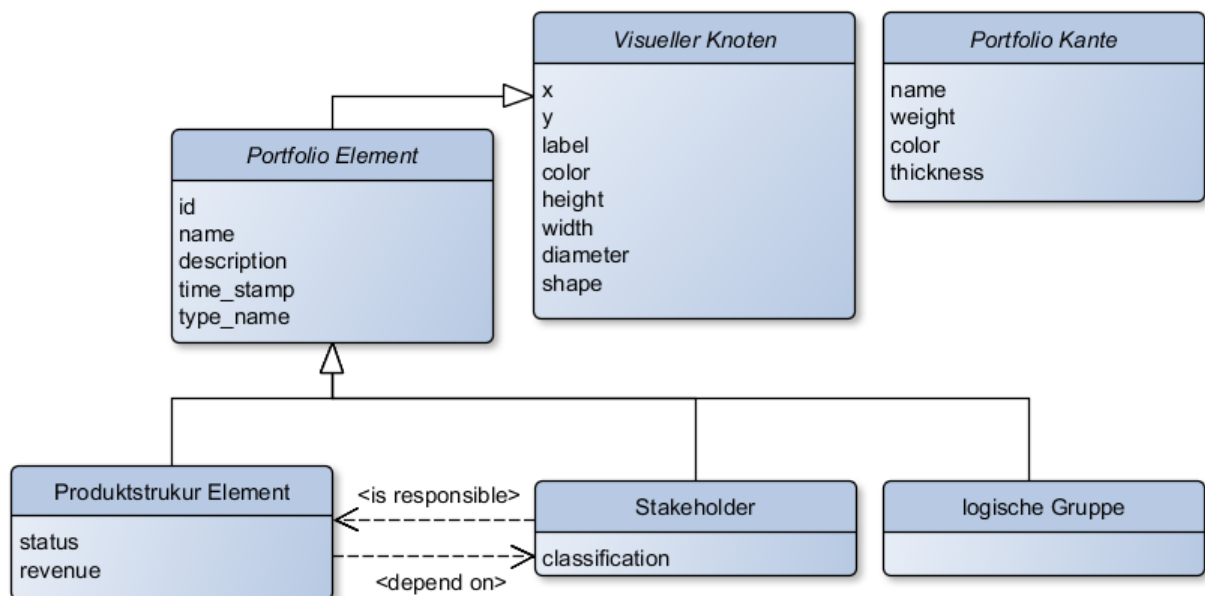


Abbildung 5-2: Abstrakte Elementtypen

Die **konkreten Knoten** werden im nächsten Teilabschnitt „Aufbau des Metamodells“ näher erläutert.

Zwischen den Systemelementen können unterschiedliche **Arten von Beziehungen** bestehen. Die UML definiert folgende Beziehungsarten, die hier übernommen werden (WEILKIENS 2008; ISO 19505-1; ISO 19505-2):

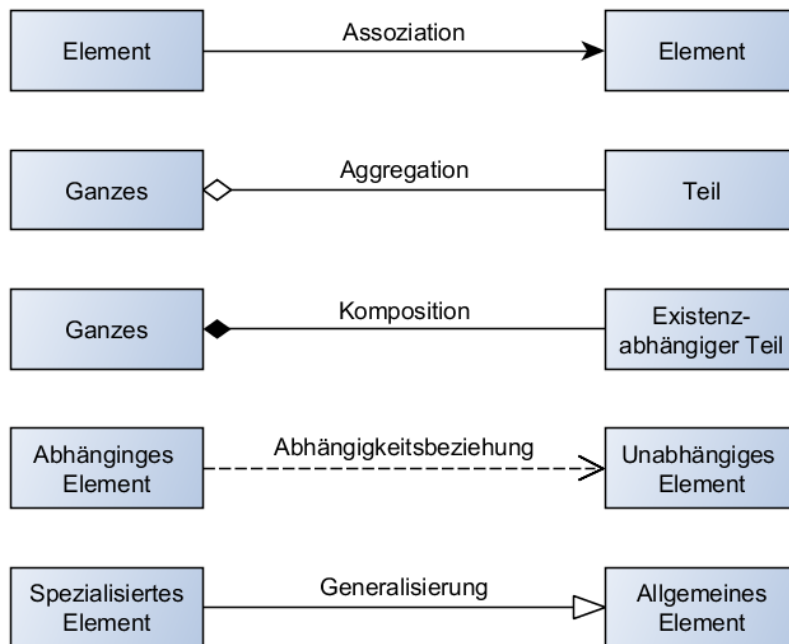


Abbildung 5-3: Kantenarten im Metamodell

Die Kanten werden wie folgt verwendet:

- **Assoziationen** für z. B. physische Verbindungen zwischen Bauteilen oder für Flussbeziehungen (Material, Energie, Informationen) mit Richtungspfeil
- **Aggregationen** drücken „ist Teil von“/“is part of“-Beziehungen aus (Teil einer Gruppe; die Gruppe würde aber auch ohne Teil bestehen)
- **Kompositionen** drücken „beinhaltet“/“contains“-Beziehungen aus (z. B. Produkt beinhaltet Baugruppe)
- **Abhängigkeitsbeziehungen** für z. B. Funktions- oder Anforderungszuordnungen
- **Generalisierungen** drücken Verallgemeinerungen aus (z. B. Kunde und Stakeholder)

Relationen können zwischen Elementen desselben Typs (intra-domäne Beziehungen) und unterschiedlichen Typs (inter-domänen Beziehungen) auftreten (LINDEMANN ET AL. 2009). Eine weitere typische Form der Verknüpfung zwischen Komponenten und Baugruppen sind **Konfigurationsregeln**. Sie drücken durch Verkettung von Booleschen Operatoren aus, welche Bausteine unter welchen Bedingungen kombiniert werden können (vgl. ZAGEL 2010, S. 80ff). Diese logischen Ausdrücke können bei Instanziierung der Kante als Kantenbedingung formuliert in die Kantenerzeugungsregel eingebracht werden.

### Kern des Metamodells

Aus dem Stand der Technik ergeben sich einige wesentliche Aspekte der Produktportfolioarchitektur, die im Metamodell abgebildet werden. Kern der Methode ist ein **produktzentriertes Metamodell des Produktportfolios** (vgl. Abbildung 5-4). Die Elemente der **Produktstruktur** erben die Attribute von *Produktstruktur\_Element* (vgl. Abbildung 5-2). Sie sind in DIN 199-1 definiert. Ein Produktportfolio besteht aus Produktfamilien, diese wiederum aus Produkten. Eine Variante ist eine Spezialisierung eines Produktes, einer Baugruppe oder eines Teils. Sie besteht wiederum aus Varianten und kann weiter in Untervarianten spezialisiert werden (PULM 2004, S. 148).



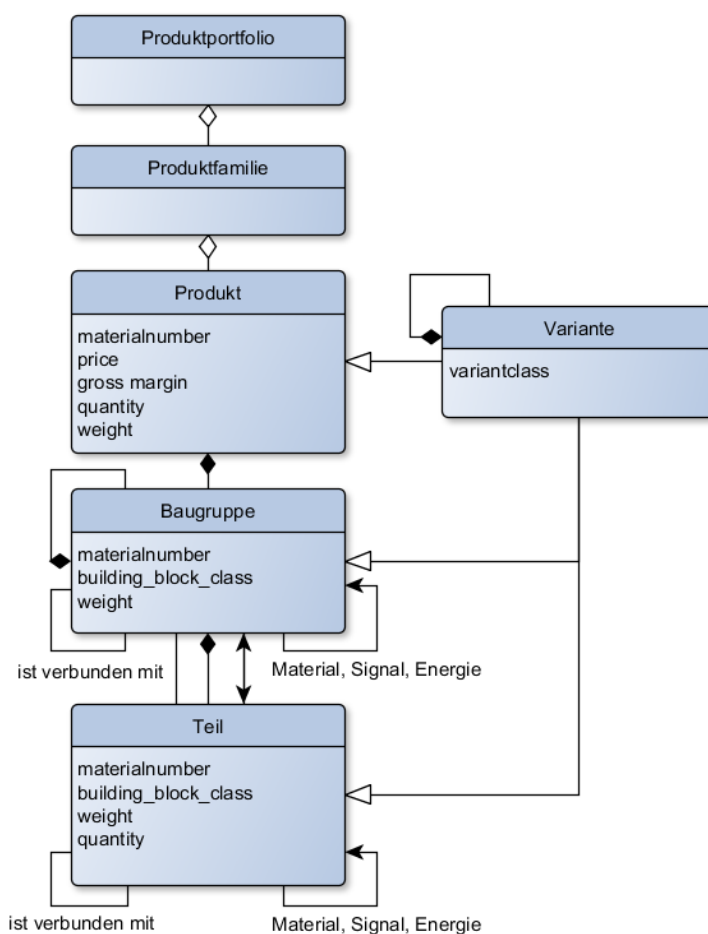


Abbildung 5-4: Produktzentriertes Metamodell des Produktportfolios

Ein Produkt besteht aus Baugruppen, die wiederum aus Teilen bestehen. Baugruppen können wiederum Unterbaugruppen enthalten. Baugruppen wie Teile können untereinander physisch verbunden sein. Zudem können Flussbeziehungen zwischen den beiden Typen bestehen. Produkt, Baugruppe, Teil und Variante werden durch weitere Attribute ergänzt (vgl. Abbildung 5-4). Für den zugehörigen Code des gesamten Metamodells wird auf den Anhang 10.1 verwiesen.

## Erweiterungen des Metamodells

Um das zentrale Metamodell des Produktportfolios gruppieren sich drei Erweiterungen:

- Knotentypen der Produktkonkretisierung.
- Container-Knotentypen, deren Instanzen logische Gruppen bilden.
- Stakeholder-Knotentypen, deren Instanzen Elemente mit relevanten Wechselwirkungen zum Produktportfolio sind.

Die Knotentypen der **Produktkonkretisierung** (vgl. Abbildung 5-5) sind Anforderung, Funktion und Testfall. Diese Knotentypen erben die Eigenschaften von dem abstrakten Typ *Portfolio Element*. Ein Produkt, eine Baugruppe oder auch ein Teil erfüllen Anforderungen und erbringen Funktionen. Funktionen wiederum erfüllen Anforderungen. Anforderungen werden von Stakeholdern definiert. Es wurde bewusst der generalisierte Knotentyp Stakeholder

verwendet, womit alle Kinderknoten von Stakeholdern ebenfalls Anforderungen definieren können. Zuletzt verifiziert ein Testfall eine Anforderung.

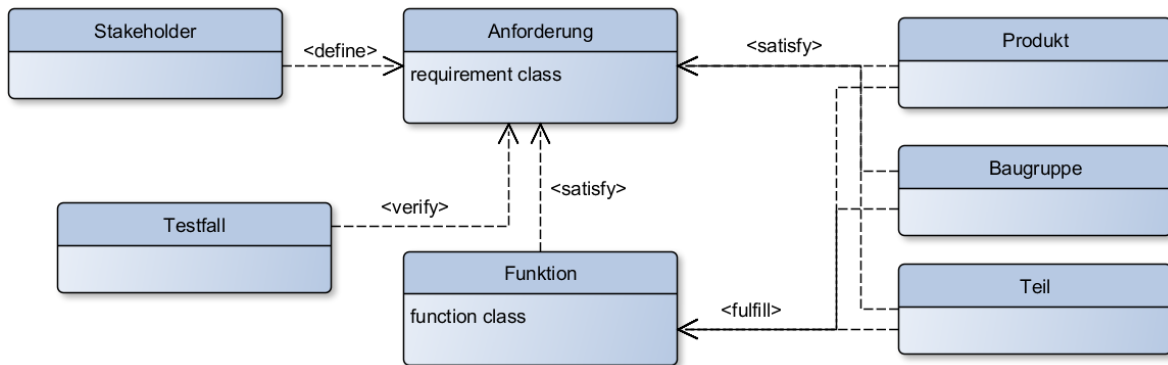


Abbildung 5-5: Knotentypen der Produktkonkretisierung

**Container-Knotentypen** (vgl. Abbildung 5-6) stellen **logische Gruppen** dar, die zur Strukturierung, zur Erzeugung von Sichten und zur Vereinfachung der Planung dienen. Man kann sich einen Container wie einen Angelpunkt in einem komplexen Gewirr an Knoten und Zusammenhängen vorstellen. Wenn man den Angelpunkt herauszieht, sind alle damit verknüpften Objekte verbunden. Somit können aufwandsarm Sichten erzeugt werden, die zur Entscheidungsunterstützung herangezogen werden können.

Eine **Technologie** kann alle Produkte, Baugruppen, Teile, Funktionen und Stakeholder zusammenfassen, die ihr zuordenbar sind. **Bausteinklassen** ordnen Baugruppen und Teile z. B. nach Grund-, Anpass- und Sonderbausteinen. Auch eine **Schnittstelle** kann als logische Gruppe verstanden werden. Ihr werden die jeweiligen Baugruppen und Teile zugeordnet, die sie verbindet. Zusätzlich können einer Schnittstelle noch Anforderungen zugeordnet werden, wenn sie z. B. bestimmte Standards erfüllt. Ebenso können **Änderungen** als logische Gruppe fungieren. Einer Änderung können sowohl Produkte wie auch Baugruppen und Teile zugeordnet werden. Zudem können Anforderungen, Funktionen und Stakeholder von der Änderungsgruppe betroffen oder deren Auslöser sein. **Logische Module** fassen Module nach unterschiedlichen Gesichtspunkten zusammen. Neben der technisch-funktionalen Modularisierung kann eine Modulbildung nach verschiedenen Lebenszyklusperspektiven erfolgen.

Die dritte Perspektive im Metamodell bilden die **Stakeholder-Knotentypen**. In den oben genannten Perspektiven wurde bereits darauf eingegangen, wie diese Knoten verknüpft werden können. In nachfolgender Abbildung 5-7 sind die Spezialisierungen der Stakeholder-Typen dargestellt. Neben den klassischen natürlichen Personen als Stakeholder wie Kunde, Lieferant und Interner Stakeholder, werden hier auch noch Rollen als Stakeholder aufgeführt, nämlich Märkte, Gesetzgeber, Produktion, Prozessverantwortliche und Projektleiter. Durch die Verknüpfung einzelner Stakeholder mit Produktumfängen, Änderungen, Anforderungen und dergleichen lassen sich wiederum individuelle Sichtweisen generieren.

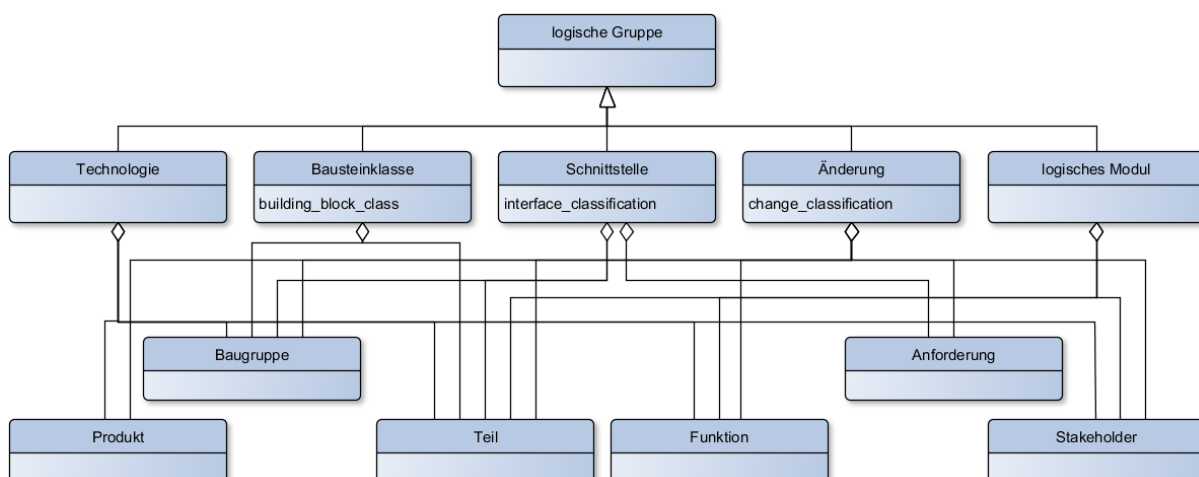


Abbildung 5-6: Container-Knotentypen

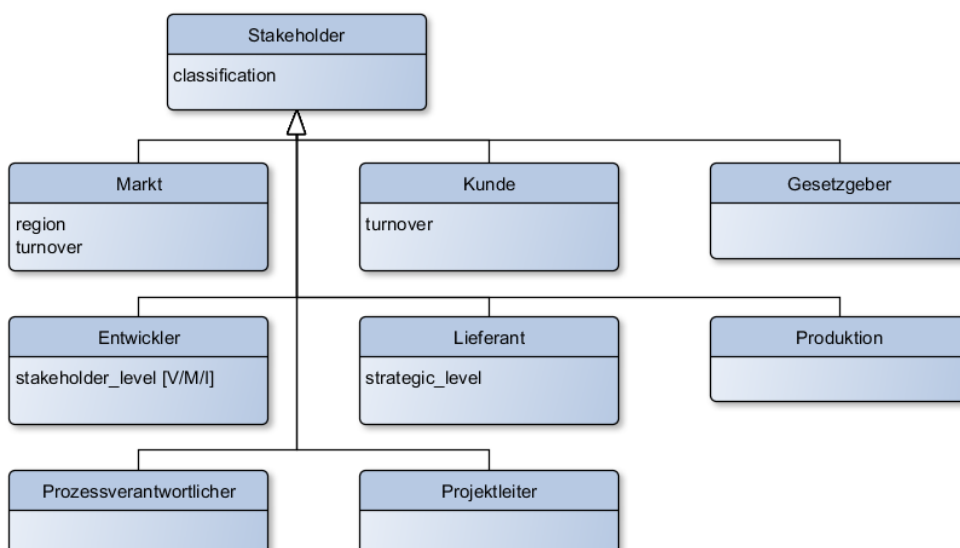


Abbildung 5-7: Stakeholder-Knotentypen

### 5.3 Wissensformalisierung

Nach PUPPE (1988, S. 11ff) sind Regeln und objektorientierte Darstellung besonders geeignete Grundtechniken zur Wissensrepräsentation und –verarbeitung. Weitere Grundtechniken sind probalistisches Schließen, nicht-monotones Schließen, temporales Schließen und Constraints (EBD.). Grundsätzlich erlaubt die verwendete Methode eine unbegrenzte Anzahl und Vielfalt unterschiedlicher Regeln. In diesem Abschnitt werden exemplarisch unterschiedliche Formen von Regelwerken dargestellt und wie diese im Umfeld des Produktportfoliomanagements eingesetzt werden können. Dafür werden Stereotypen von Problemlösungen entwickelt und deren Funktionsweise dargestellt. Diese müssen dann ggf. an die Randbedingungen im Einsatz angepasst werden. Die fünf Regelkategorien mit den jeweiligen Regel-Stereotypen sind in der nachfolgenden Übersicht dargestellt Abbildung 5-8.

## Regelwerk für das Produktportfolio Management

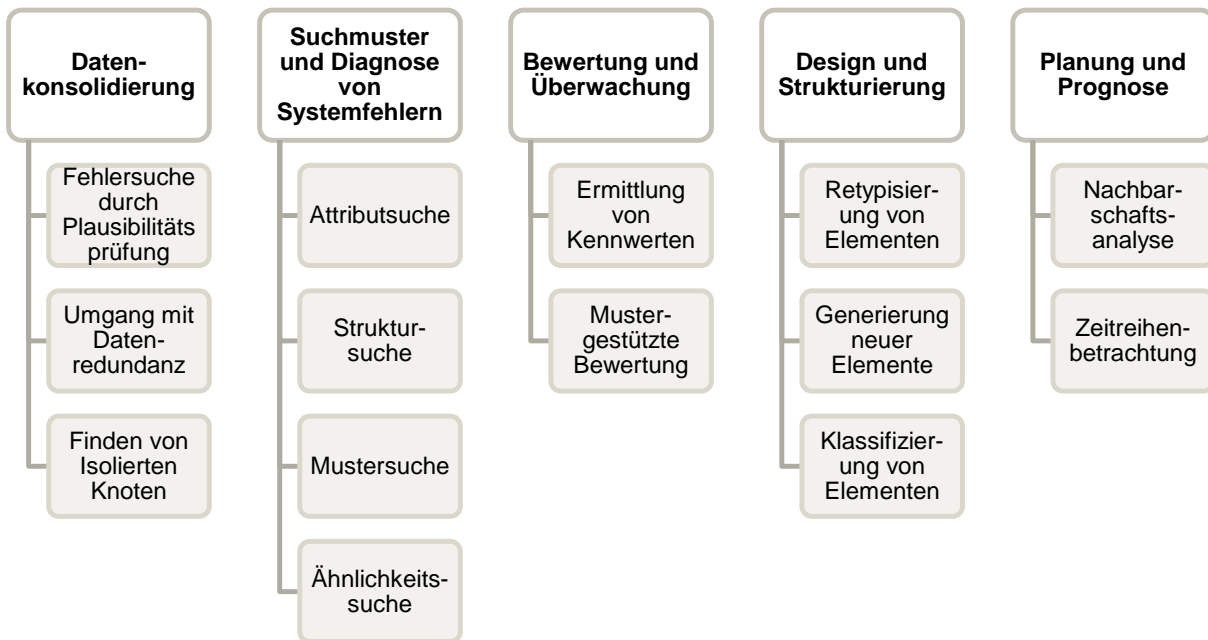


Abbildung 5-8: Übersicht über die Aufgabenfelder und die Regelbausteine des Regelwerks für das Produktportfoliomanagement

Eine einzelne Regel wird hier als **modularer Regelbaustein** bezeichnet. Durch den Einsatz der *Rule Application Control Language* werden diese Regelbausteine zu **Sequenzen** zusammengeführt. Die Regelbausteine können mit logischen Operatoren verkettet werden und somit **Ablaufstrukturen** definiert werden. Mit Operatoren kann die Art der Regelausführung gesteuert werden. Beispielsweise kann eine Regel solange ausgeführt werden bis sie alle Muster im Graph gefunden hat oder auch nur einmal bei einem zufällig gefundenen Muster. Ein Überblick über die Operatoren der Sequenzsprache befindet sich im Anhang 10.2.

Somit ist es möglich, **umfangreiche Workflows** zu generieren, die Methodendurchläufe komplett oder in Schritten abbilden können. Iterative Arbeitsschritte des Systemarchitekten werden mit dieser Technologie automatisierbar. Interaktions- und Entscheidungspunkte des Experten mit den Regelsequenzen können idealerweise so eingebettet werden, dass aufwendige monotone Arbeiten reduziert und sich größere Zeitanteile für kreativ-strategische Aufgaben ergeben. Ebenso können Aufgaben vom Rechner durchgeführt werden, die angesichts unüberschaubarer Datenmengen manuell mit Tabellenkalkulationsprogrammen oder Datenbankabfragen nicht oder nicht wirtschaftlich bearbeitet werden können. Damit werden die Anforderungen (A3), (A4) und (A6) adressiert.

### 5.3.1 Datenkonsolidierung

Beim Import von Daten aus unterschiedlichen Quellen entstehen teilweise **Redundanz** und **Inkonsistenz** von Informationen. Zudem können Daten bereits vor dem Import fehlerhaft gewesen sein. Zur Datenkonsolidierung ist es daher sinnvoll, den vollständig generierten Graph in seiner Rohform vor der Verwendung mit Hilfe eines vorgeschalteten Regelwerkes vorzubereiten.

## Fehlersuche durch Plausibilitätsprüfung

Der Graph in seiner Rohfassung wird instanziiert mit den Daten, wie sie aus Datenbanken exportiert oder von Mitarbeitern abgefragt wurden. Wenn **fehlerhafte oder leere Datenfelder** einer Datenbank Attributen des Graphs zugewiesen wurden, können die entsprechenden Knoten mit **Plausibilitätsabfragen** identifiziert und ausgegeben werden.

**Beispiel (1):** Aus dem ERP-System wurden die Teilelisten mit den Attributen *Sachnummer*, *Name* und *Gewicht* exportiert und als Knoten im Graph instanziiert. Auf der Linken Seite der Regel können nun Fehlermuster und nicht plausible Werte definiert werden:

- Teileknoten, die ein leeres Attribut bei *Name* oder *Sachnummer* haben.
- Teileknoten, deren Attribut *Sachnummer* einen Wert aufweist, der nicht der Sachnummer-Systematik des Unternehmens entspricht
- Teileknoten, deren Attribut *Gewicht* einen Wert kleiner als 5 g oder größer als 1200 g aufweist

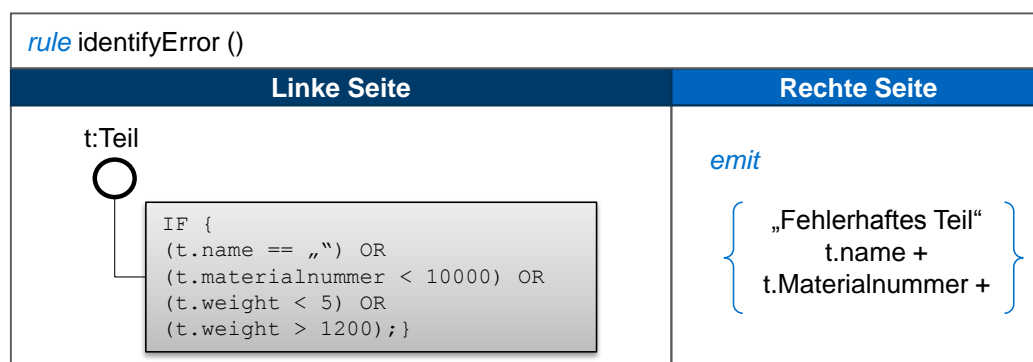


Abbildung 5-9: Beispiel zur Datenkonsolidierung (Plausibilitätsprüfung)

Auf der Rechten Seite kann dann die Aktion festgelegt werden, die mit den gefundenen Fehlerknoten durchgeführt werden soll. Beispielsweise könnte mit dem Befehl *emit* ein Report generiert werden, in dem die fehlerhaften Knoten aufgelistet sind, damit sie manuell in der Datenbank berichtigt werden können.

## Umgang mit Datenredundanz

Es kann vorkommen, dass einem Attribut *a* eines Knotens aus mehreren Datenquellen Werte zugewiesen werden und somit **Redundanz** entsteht. Bei der Erzeugung des Graphen können alle Quellen ausgelesen und deren Werte in die Attribute  $a_1 \dots a_n$  abgelegt und verglichen werden. Treten Konflikte – also Unterschiede – auf, können Strategien vorgegeben werden, welches Attribut gelten soll.

Beispielsweise kann ein Attribut Werte aus zwei Quellen *Q1* und *Q2* erhalten. Die Quelle *Q1* ist aber unvollständig und weist leere Felder auf. Dennoch sind die Daten, die dort gespeichert sind, die aktuelleren. Daher wird bei der Instanziierung eines Knoten mit diesem Attribut der Quelle *Q1* der Vorzug gegeben und im Falle eines leeren Feldes die zweite Quelle *Q2* gewählt.

## Finden von isolierten Knoten

Isolierte Knoten sind mit anderen Knoten nicht verbunden. Dies kann daran liegen, weil sie mit dem betrachteten System keine Relationen haben, also fälschlicherweise im Graph gespeichert sind oder weil Fehler in den Daten waren, und somit keine Kanten erzeugt werden konnten.

**Beispiel (2):** Eine simple Regel könnte demnach lauten, auf der Linken Seite nach **isolierten Knoten** ohne Relationen zu suchen und diese auf der Rechten Seite zu eliminieren oder in einem Report wieder zur manuellen Überprüfung zusammenzustellen.

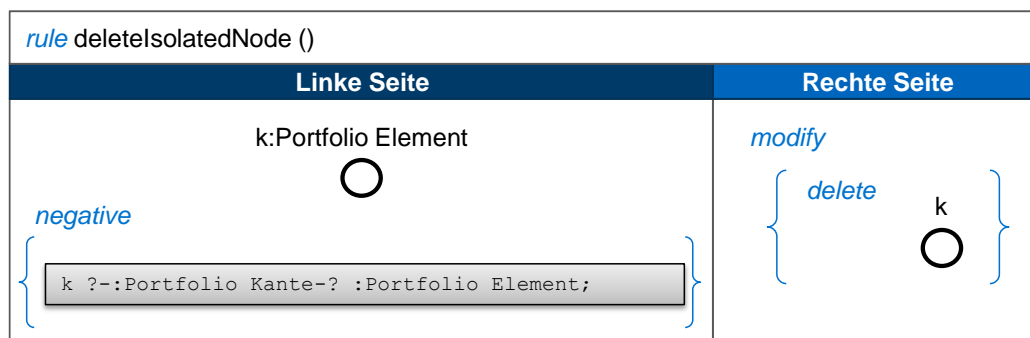


Abbildung 5-10 Beispiel zur Datenkonsolidierung (Löschen isolierter Knoten)

Auf der Linken Seite wird nach einem Knoten  $k$  vom abstrakten Typ *Portfolio Element* gesucht, was sämtliche Knoten des Graphen einschließt. Im negative-Ausdruck wird die Bedingung definiert, die der gesuchte Knoten *nicht* erfüllen darf. Hier wird als Bedingung gesetzt, dass der Knoten  $k$  Kanten jeglicher Art und Richtung zu Elementen jeglichen Typs haben darf. Das bedeutet im Umkehrschluss, dass isolierte Knoten identifiziert werden, die auf der Rechten Seite gelöscht werden.

## Fazit aus der Datenkonsolidierung

Je nach Datenqualität kann die Konsolidierung viel Zeit in Anspruch nehmen, wenn sich Fehlerreports mit sehr unterschiedlichen Fehlerarten ergeben. Dann ist der Aufwand für jede Fehlerart eine eigene Regel zu definieren ggf. höher als die Fehler manuell zu berichtigen. Die Erstellung und Verwendung eines Regelwerks zur Konsolidierung hilft allerdings dabei, den **Aufwand insgesamt zu reduzieren**. Bei der Ersterstellung können die Regeln auch bei großen Datenmengen effizient angewendet werden. Die Überprüfung komplexerer Abfragen kann solange vom Rechner durchgeführt werden, bis kein entsprechendes Muster mehr im Graph gefunden wird. Durch die Wiederverwendbarkeit von Regeln kann der spätere Aufwand bei einer möglichen Einpflege von Updates signifikant reduziert werden.

Die *Methode zur Zusammenführung relevanter Daten in graphenbasierter Form* ermöglicht somit bereits im initialen Graph Fehler und Unstimmigkeiten effektiver und effizienter zu finden. Durch Berichtigung der Fehler kann ungewollte Komplexität im Portfolio reduziert und die Übersichtlichkeit insgesamt gesteigert werden. Hiermit wird Anforderung (A2) adressiert.

### 5.3.2 Suchmuster und Diagnose von Systemfehlern

Wenn die Arbeitsfähigkeit mit einem konsolidierten Graph hergestellt ist, können beliebige Regeln und Regelsequenzen angewendet werden. Eine Grundanwendung stellt die Suche dar.

#### Attributsuche

Für eine **Attributsuche** bieten sich parametrisierte Regeln an, denen ein oder mehrere Parameter können übergeben werden. Auf der Linken Seite werden dann die Parameter in den Bedingungen eingebunden. Somit können Knoten eines bestimmten Typs mit den gesuchten Attributen in einem Graph identifiziert werden. Die Suchbedingungen können so spezifiziert werden, dass **Filtern** von Wertebereichen, verkettete Bedingungen und Ähnliches möglich ist. Auf der Rechten Seite der Regel kann wiederum eine beliebige Aktion gewählt werden, wie z. B. die Ausgabe in einem Report, Einfärbung in einer Signalfarbe oder Löschen der Knoten.

**Beispiel (3):** In einem Portfoliograph werden alle Produkte gesucht, die den Status „Abkündigung“ haben und die Stückzahl kleiner als 10 liegt. Diese sollen aus dem Graphen gelöscht werden.

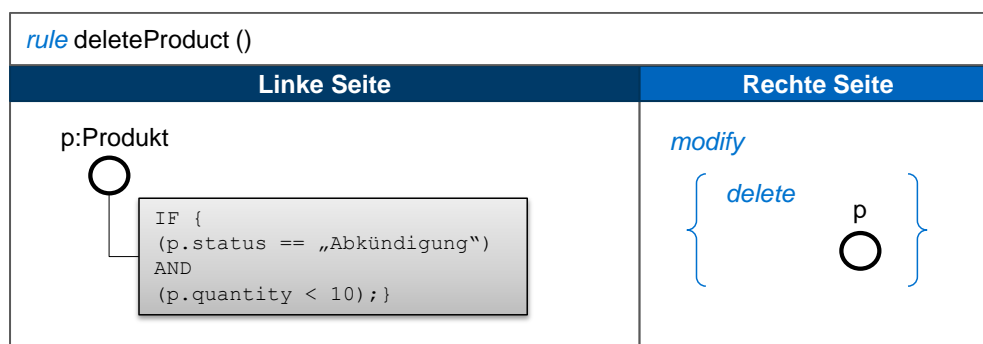


Abbildung 5-11: Beispiel zur Attributsuche

Alle Knoten  $p$  vom Typ *Produkt* im Graph, die beide Bedingungen auf der Linken Seite erfüllen, werden demnach auf der Rechten Seite gelöscht.

#### Struktursuche

In der Struktursuche wird gezielt nach Knoten-Kanten-Kombinationen gesucht. Damit kann das Problem eines aufwendigen **Verwendungsnachweises** elegant gelöst werden. Der Suchmechanismus ist für einen Verwendungsnachweis eines Bauteils der gleiche wie für Abhängigkeitsbeziehungen insgesamt. Es können also auch verknüpfte Anforderungen, Funktionen, Bauteilverantwortliche etc. so gesucht werden.

**Beispiel (4):** Gesucht werden alle Baugruppen, in denen ein Bauteil mit der Materialnummer „1234“ verbaut ist. Die entsprechenden Baugruppen sollen rot eingefärbt werden.

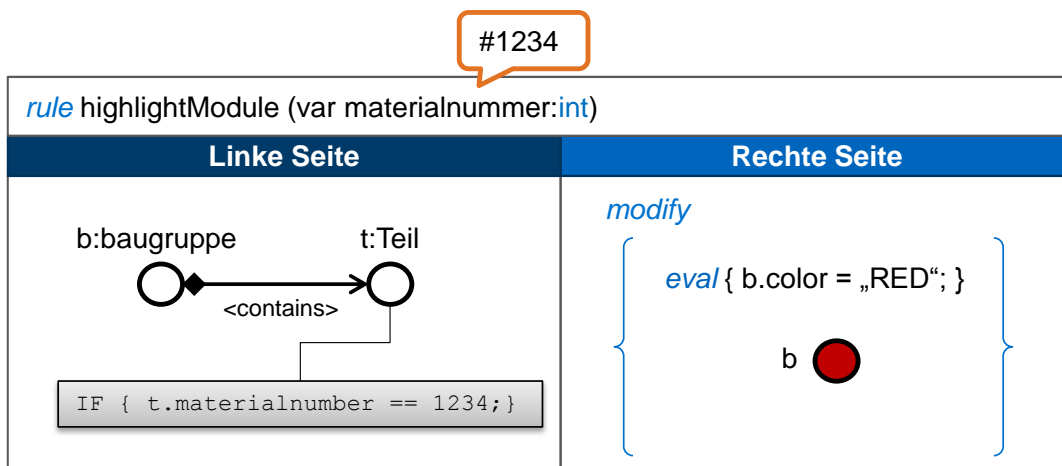


Abbildung 5-12: Beispiel zum Verwendungsnachweis

Zunächst wird der Regel eine Materialnummer als Variable übergeben. Auf der Linken Seite wird nach der Struktur gesucht, in der ein *Teil t* mit dieser Materialnummer in einer *Baugruppe b* enthalten ist. Auf der Rechten Seite wird der gefundene Baugruppenknoten rot eingefärbt.

Die **Suche nach strukturellen Fehlern** illustriert nachfolgendes Beispiel.

**Beispiel (5):** Es wird eine Struktur gesucht, die so nicht vorkommen darf: Teile, die wiederum Teile enthalten, sind per Definition inkorrekt typisiert. Sie müssen als Baugruppe typisiert sein. Daher sollen alle Knotennamen, die eine solche Struktur aufweisen, zur Überprüfung in einen Report ausgegeben werden.

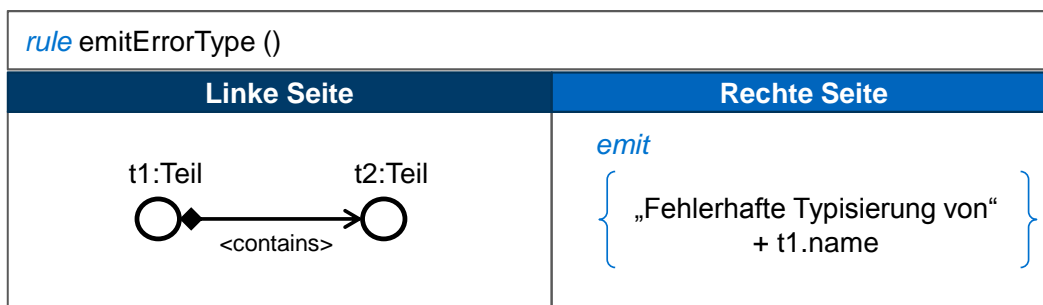


Abbildung 5-13: Beispiel zur Struktursuche

Auf der Linken Seite wird nach einem Teil *t1* gesucht, das ein weiteres Teil *t2* enthält. Wenn ein solches Muster gefunden wird, soll es auf der Rechten Seite mit dem Namen des Teils ausgegeben werden. Auf der Rechten Seite wäre auch direkt eine Retypisierung der fehlerhaften Teile in Baugruppen möglich (vgl. hierzu die Fallstudien in Kapitel 6).

## Mustersuche

Werden Attribut- und die Struktursuche kombiniert, wird nach Mustern gesucht (vgl. Kapitel 2.1.5). Somit können **Such- und Fehlermuster** in einer Regel definiert werden. Zu einem Fehlermuster gehört auch immer ein Soll-Muster, von dem es abweicht. Das Soll-Muster ergibt sich aus der individuellen Zielsetzung des Systemarchitekten. In der Mustersuche kann man sich zudem die Vererbung von Typen zunutze machen.



**Beispiel (6):** Gesucht werden alle Produktstrukturelemente, die Anforderungen erfüllen, die explizit mit dem chinesischen Markt verknüpft sind. Diese Knoten sollen mit ihrem Typ, Name und Umsatz ausgegeben werden.

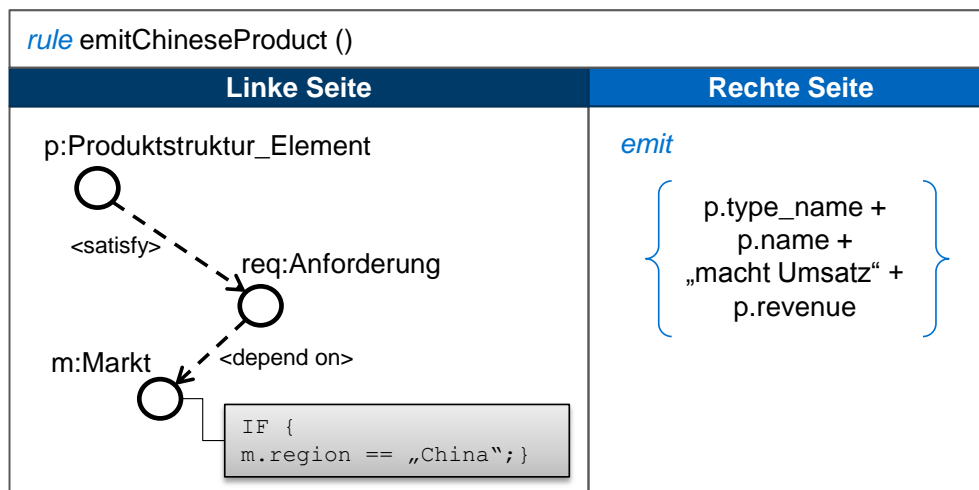


Abbildung 5-14: Beispiel zur Mustersuche

Sobald ein *Produktstruktur\_Element* *p* gefunden wurde, das mit einer *Anforderung* *req* über eine *satisfy*-Beziehung verknüpft ist, muss geprüft werden, ob diese Anforderung mit einem *Markt*-Knoten verbunden ist, der im Attribut *region* einen String „China“ stehen hat. Alle gefundenen Knoten werden mit ihrem Typ, Name- und Umsatz-Attribut als Text ausgegeben. Gefunden werden alle Knoten, die entweder als *Produktstruktur\_Element* typisiert sind, aber auch alle Knoten, deren Subtyp von diesem Typ die Eigenschaften erbt. In diesem Beispiel werden also alle Knoten von den Typen *Produktportfolio*, *Produktfamilie*, *Produkt*, *Baugruppe*, *Teil* und *Variante* in die Suche miteinbezogen. Die Suche kann weiter verallgemeinert werden, wenn die Regel beispielsweise mit beliebigen Regionen („China“, „Europa“, „Nordamerika“ etc.) als Parameter aufgerufen wird.

Ein weiteres Beispiel soll die **Suche nach Fehlermustern** illustrieren:

**Beispiel (7):** In einem umfangreichen Bausteininventar eines Baukastensystems sollen fehlerhafte Klassifikationen festgestellt werden. Wird eine Bausteinklassifizierung nach PAHL ET AL. (2007, S. 664) angewendet, dann sollen Grundbausteine Träger von Grundfunktionen sein. Analog gilt dies für Anpassbausteine und -funktionen, Sonderbausteine und -funktionen wie auch für Nicht-Bausteine und auftragsspezifische Funktionen. Wenn nun also ein Grundbaustein Anpassfunktionen erfüllt, so liegt ein konzeptioneller Architekturfehler vor. Mit der Methodik können diese Muster erkannt und dargestellt werden. Es ist dann Aufgabe des Systemarchitekten zu entscheiden, ob die Klassifizierung des Bausteins oder die der Funktion sinnvoll ist. Oder es wäre zu prüfen, ob sich die funktionale Koppelung des Bausteins und der Funktion konstruktiv ändern ließe.

Daher soll als Fehlermuster auf der Linken Seite der Regel diejenige Komponenten-Funktions-Paare definiert werden, die nicht dem Soll-Muster entsprechen (vgl. Kapitel 3.1.3). Auf der Rechten Seite werden die gefundenen Fehler ausgegeben und dienen als Grundlage für

die Definition weiterer Maßnahmen. Nachfolgende Abbildung zeigt das Fehlermuster für Grundbausteine. Anpass-, Sonder- und Nicht-Bausteine werden entsprechend analog definiert.

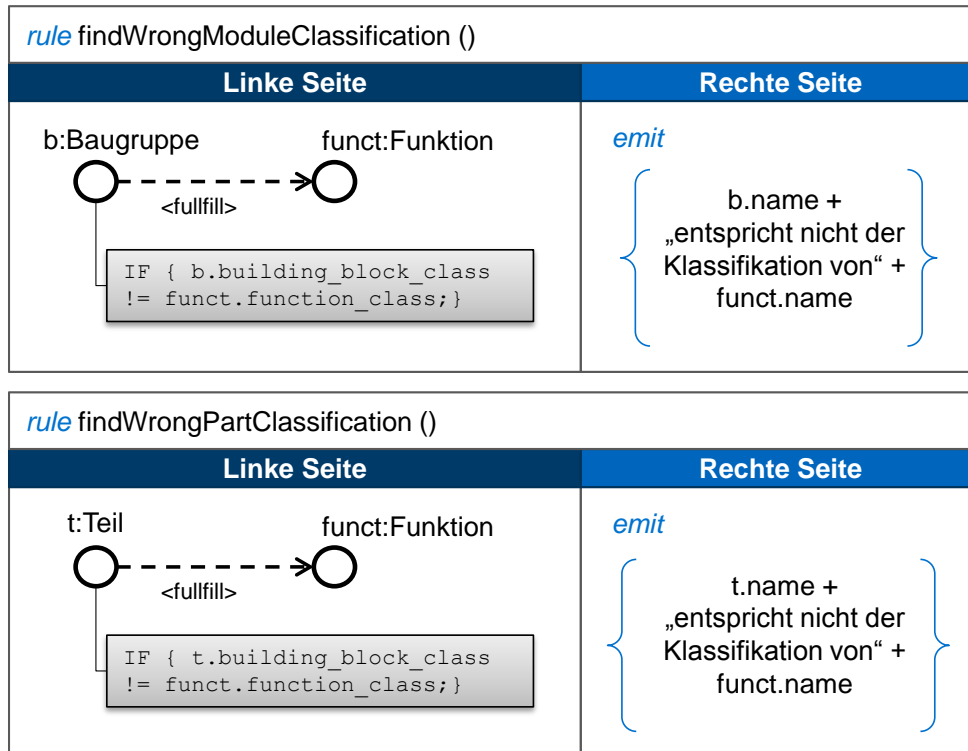


Abbildung 5-15: Beispiel zur Fehlermustererkennung

Bausteine können Baugruppen wie auch Teile sein und eine oder mehrere Funktionen erfüllen. Daher gibt es zwei Regeln. Es wird nach einer Baugruppe *b* oder einem Teil *t* gesucht, die jeweils mit einer *fulfill*-Kante mit einer Funktion verknüpft sind. Entspricht die Bausteinklassifizierung nicht der Klassifizierung der Funktion, wird ein Fehler erkannt und das Knotenpaar als Text auf der Rechten Seite ausgegeben.

## Ähnlichkeitssuche

Die vierte Form der Suche ist die Ähnlichkeitssuche. Es können **ähnliche Knoten** oder auch **ähnliche Strukturen** gesucht werden. Im Prinzip handelt es sich um eine erweiterte Mustersuche mit Filtern. Knoten sind ähnlich, wenn beispielsweise bestimmte Attribute in einem gemeinsamen Wertebereich liegen. Die Bedingungen auf der Linken Seite müssen so definiert werden, dass Ähnlichkeit zwischen Elementen festgestellt werden kann. Der Nachteil des Verfahrens liegt darin, dass die Attribute, die für eine Ähnlichkeitsanalyse herangezogen werden, vorher ausgewählt und bekannt sein müssen.

Eine weitere Möglichkeit der Ähnlichkeitssuche ist beim **Attributvergleich** festzulegen, dass beispielsweise 8 von 10 Attributen bei einem paarweisen Vergleich gleich sein müssen, damit zwei Knoten als ähnlich bezeichnet werden. Ebenso kann nach ähnlichen Relationen von Knoten gesucht werden. Also die Suche nach Knoten mit **ähnlicher struktureller Einbettung** in einem Graph, die eine ähnliche Anzahl von Relationen haben, mit den gleichen Elementen verknüpft sind oder Relationen zu gleichen Knotentypen aufweisen.

Die Ähnlichkeitssuche kann in vielerlei Hinsicht die Arbeit des Systemarchitekten verbessern und vereinfachen. Bei der Definition von **neuen** Knoten können **vorhandene Knoten mit ähnlichen Attributen** gesucht und angezeigt werden. Zum einen kann geprüft werden, ob es beispielsweise bereits ähnliche Bauteile gibt, die den Anforderungen gerecht werden. Somit kann der Aufwand einer Neuentwicklung reduziert oder vermieden werden.

Zum anderen kann man sich **an vorhandenen ähnlichen Knoten orientieren**, welche weiteren Attribute diese typischerweise führen, welche Nachbarelemente mit ihnen verknüpft sind und welche Kennwerte sie charakterisieren – also Prinzipien, wie sie beispielsweise im eCommerce eingesetzt werden, wenn beim Kauf eines Produktes weitere Produkte angeboten werden, die andere Benutzer zusammen mit dem gewünschten Produkt zuvor gekauft hatten. Dies kann helfen, die Datenqualität zu erhöhen, unnötige Vielfalt zu reduzieren und die Klassifizierung im Produktportfolio zu unterstützen. Für weitere Vorteile der Gleichteileanalyse und Ähnlichkeitsermittlung in Produktportfolios wird auf MAIER (1993) verwiesen.

**Beispiel (8):** Für die Bildung einer Bausteinklasse „Stahlteile A“ sollen ähnliche Teile in einer logischen Gruppe zusammengefasst werden. Als Attribute für den Ähnlichkeitsabgleich werden *weight*, *material* und *quantity* herangezogen. Alle Teile, die zwischen 800 g und 1000 g schwer sind, als Material Stahl verwenden und eine Stückzahl größer als 100 aufweisen, sollen dieser Bausteinklasse hinzugefügt werden.

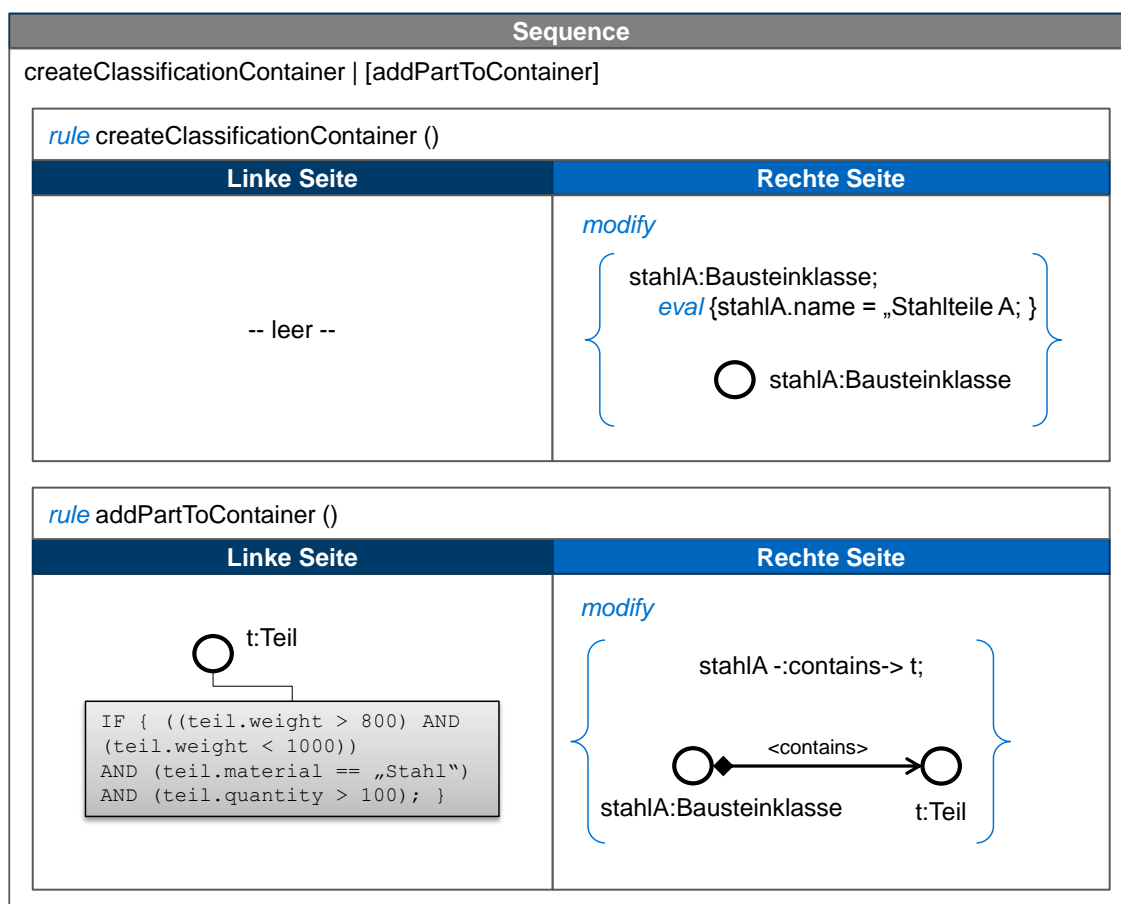


Abbildung 5-16: Beispiel zur Ähnlichkeitssuche

Es wird eine Regelsequenz mit zwei Regeln aufgebaut. Zunächst wird ohne Vorbedingung ein Knoten *stahlA* des Typs *Bausteinklasse* erzeugt und benannt. In der zweiten Regel wird nun nach Knoten *t* des Typs *Teil* gesucht, die die Bedingungen der Bauteilklassse erfüllen. Wenn alle Bedingungen erfüllt sind, wird der gefundene Knoten der Bausteinklasse zugeordnet und nach dem nächsten Teil gesucht bis alle geprüft wurden.

Auch **Substrukturen in einem Graphen** können **verglichen** werden, in welchen Bereichen sie ähnlich oder gleich sind. Beispielsweise unterscheiden sich Produktvarianten oft nur in wenigen Teilen. Bei einem Vergleich von Varianten können Unterschiede ebenso wie Gemeinsamkeiten hervorgehoben werden.

**Beispiel (9):** Von einem Produkt gibt es ein Basisprodukt und mehrere Varianten. Jede Variante soll im paarweisen Vergleich mit dem Basisprodukt verglichen werden. Gleiche Baugruppen sollen grün eingefärbt werden.

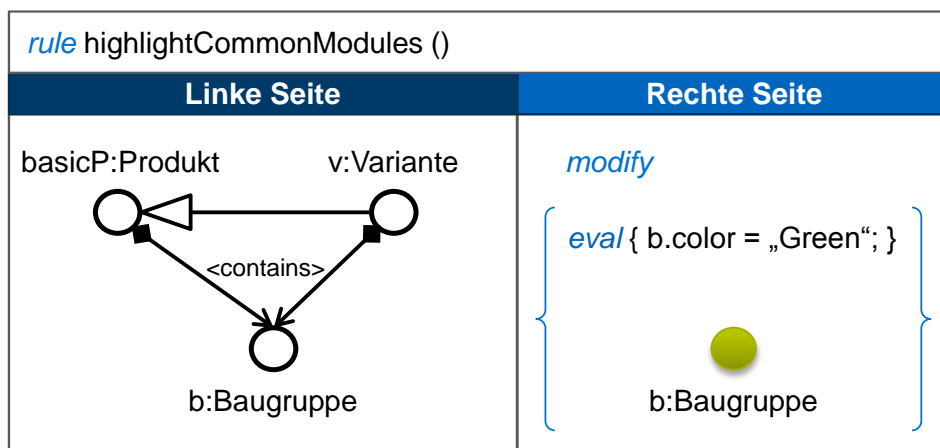


Abbildung 5-17: Beispiel zu ähnlichen Substrukturen

Auf der Linken Seite werden zunächst Baugruppen gesucht, die eine *contains*-Kante zum Basisprodukt *basicP* und zur Variante *v* haben. Die Variante *v* muss eine Spezialisierung des Produktes *basicP* sein. Wird dieses Muster gefunden, wird der Bauteilknoten *b* grün eingefärbt.

### Fazit zu Suchmustern und Diagnostik von Systemfehlern

Eine graphenbasierte Suche unter Verwendung von Regeln ermöglicht es, auch komplexe Abfragen zu generieren. Diese Abfragen können durch einen Rechner leistungsfähig abgearbeitet werden und somit effektiv wie effizient zu Suchergebnissen führen. Neben der Suche nach Attributen kann auch eine Suche nach Substrukturen oder mit der strukturellen Einbettung eines Elementes als Suchbedingung durchgeführt werden. Dies erweitert die Suchmöglichkeiten im Produktportfoliomanagement und vereinfacht bestehende Abfragemöglichkeiten. Mit der Methode zur effektiven Mustererkennung wird somit Anforderung (A4) adressiert.

### 5.3.3 Bewertung und Überwachung

Eine weitere Kategorie ist die **Bewertung von Elementen und Strukturen** sowie die **Überwachung** eines Produktportfolios. Die Methodik soll dabei unterstützen, vor allem in diesem Problemfeld zu höherer Effektivität und Effizienz zu gelangen. Da die Ermittlung von Kennzahlen meist mit sehr hohem Aufwand verbunden ist, können entscheidungskritische Informationen oft nicht zeitgerecht ermittelt werden (JUNGE 2005, S. 75ff). Mit einer Formalisierung von Bewertungswissen können die Kennzahlen kontinuierlich berechnet und beobachtet werden.

Die Kennzahlen müssen so definiert werden, dass sie zur Steuerung des Portfolios im Sinne der Unternehmensziele dienen. Für eine **sinnvolle Operationalisierung der Unternehmensziele**, Erhebung von geeigneten Kennzahlen und Ableitung von Maßnahmen wird auf EISENFÜHR ET AL. (2010) verwiesen. Es ist nicht Fokus der Arbeit, die passenden Kennzahlen für das Produktportfoliomanagement vorzuschlagen, sondern aufzuzeigen, wie Kennzahlen effizient ermittelt werden können. Für eine Auswahl von Methoden zur Kennzahlenermittlung wird auf Kapitel 3.2.3 verwiesen. In Folgenden wird vorgestellt, wie Kennwerte ermittelt und mustergestützte Bewertungen durchgeführt werden können.

#### Ermittlung von Kennwerten

Die **Ermittlung von Kennwerten** ist maßgeblich davon abhängig, mit welchen Daten der Graph instanziiert wurde. Die Anzahl der möglichen Attribute je Knoten- und Kantentyp ist im Prinzip unbegrenzt, sollte aber im Sinne der Rechnerleistung und Datenpflege auf ein rationales Maß begrenzt sein. Mit den Attributen der Knoten und Kanten können Berechnungen beliebiger Art durchgeführt werden. Wenn in den Regeln zusätzlich Kriterien aus der strukturellen Einbettung eines Knotens herangezogen werden, handelt es sich um eine **mustergestützte Bewertung**. Diese wird im nächsten Abschnitt behandelt. Die Auswertung von Attributen wird zunächst in einem einfachen Beispiel illustriert.

**Beispiel (10):** Für das nächste Steuerkreistreffen eines Unternehmens sollen die Jahresumsätze aller Produkte aus den vorliegenden Quartalsumsätzen berechnet und als Liste ausgegeben werden.


<i>rule</i> calculateAnnualRevenue ()	
Linke Seite	Rechte Seite
p:Produkt 	<pre> <i>modify</i> {   <i>eval</i> { p.jahresumsatz = p.umsatz_q1 +           p.umsatz_q2 + p.umsatz_q3 + p.umsatz_q4; }   <i>emit</i> ( „Jahresumsatz von“ + p.name + „ist“ +           p.jahresumsatz; ) }           </pre>

Abbildung 5-18: Beispiel zur Berechnung von Kennwerten

Auf der Linken Seite wird nach einem Produkt  $p$  gesucht. Auf der Rechten Seite wird zunächst der Jahresumsatz aus den Quartalsumsätzen  $umsatz_{q1} - umsatz_{q4}$  berechnet und das Ergebnis dem Attribut *jahresumsatz* zugewiesen. Anschließend wird der Knoten mit *name* und *jahresumsatz* ausgegeben.

### **Mustergestützte Bewertung**

Die Mächtigkeit der Kennwertermittlung und der Bewertung zeigt sich, wenn **strukturgebundene Kriterien** mitberücksichtigt werden. Weiter oben im Beispiel (9) wurde die Ähnlichkeit von Strukturen eines Basisprodukts und seiner Varianten untersucht. Dieses Beispiel soll mit der Ermittlung einer Kennzahl erweitert werden.

**Beispiel (11):** Mit dem Gleichteilegrad soll das Verhältnis von Gleichteilen zu Teilen insgesamt ausgedrückt werden. Die Kennzahl hat einen Wertebereich von 0 bis 1. Hohe Werte drücken einen hohen Gleichteileanteil aus. Von einem Produkt gibt es ein Basisprodukt und mehrere Varianten. Jede Variante soll im paarweisen Vergleich mit dem Basisprodukt verglichen werden. Dabei soll festgestellt werden, wie viele Teile gleich und wie viele unterschiedlich sind. Das Verhältnis aus Gleich/Ungleich der jeweiligen Variante soll als Kennzahl dem Attribut *commonparts\_ratio* zugeordnet und als Report ausgegeben werden.

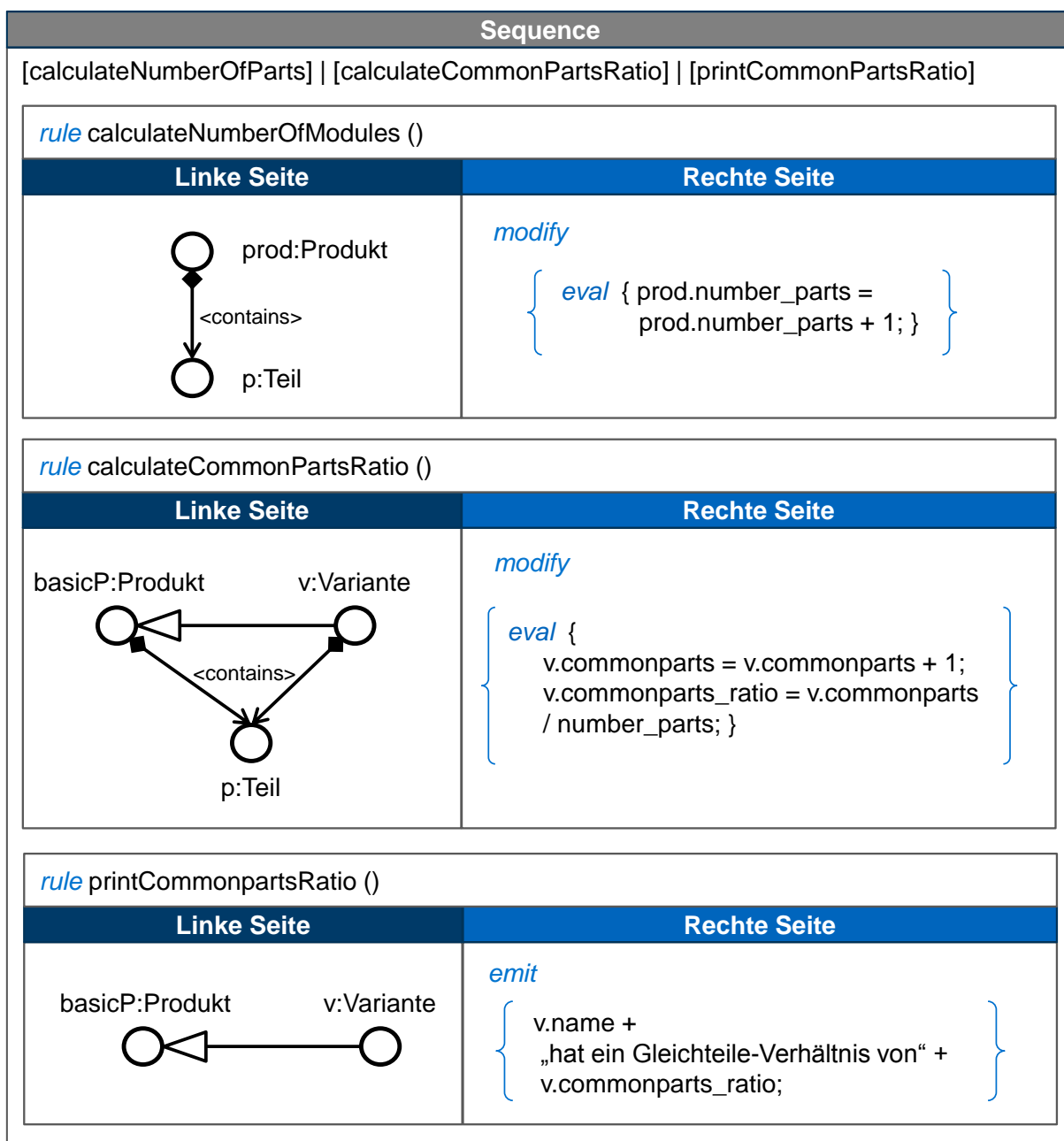


Abbildung 5-19: Beispiel zur mustergestützten Bewertung

Drei Regeln sind in einer Sequenz zusammengefasst. In der ersten Regel wird zunächst die Anzahl der Teile eines jeden Produktes berechnet und dem Attribut *number\_parts* zugeordnet. Da der Typ *Variante* die Eigenschaften von *Produkt* erbt, sind die Varianten in dieser Berechnung mit eingeschlossen. In der zweiten Regel werden auf der Linken Seite Teile gesucht, die eine *contains*-Kante zum Basisprodukt *basicP* und zur Variante *v* haben. Die Variante *v* muss hierbei eine Spezialisierung des Produktes *basicP* sein. Wird dieses Muster gefunden, zählt der Gleichteilezähler der Variante *v* im Attribut *commonparts* um 1 hoch. Dem Attribut *commonparts\_ratio* wird der Wert des Verhältnis von *commonparts* zu *number\_parts* zugewiesen. In der dritten Regel werden die Kennzahlen *commonpart\_ratio* der Varianten des Produktes *basicP* ausgegeben.

## Fazit aus der Bewertung und Überwachung

Das Prinzip der mustergestützten Bewertung lässt sich **auf zahlreiche Sachverhalte übertragen**. Beispielsweise können mit diesem Verfahren Umsatzzahlen, Kosten, Stückzahlen oder Ähnliches über die Kompositionsstufen in der Produktstruktur hochaggregiert werden. Zudem können aufwandsarm Statistiken über die Anzahl von Elementen verschiedener Datentypen erstellt und Kennzahlen daraus ermittelt werden. Somit können Summen wie z. B. Anzahl von Teilen, Modulen und Produkten in einer Produktfamilie errechnet werden. Des Weiteren können Verhältniskennzahlen gebildet werden, wie z. B. das Verhältnis von Grund- zu Anpassbausteinen innerhalb einer Produktfamilie oder auch das Verhältnis von externer Produktvielfalt zu interner Bausteinvielfalt.

Sobald eine Bewertungsmethode formalisiert wurde, kann sie beliebig oft auf den Graphen angewendet werden. Somit kann ein **kontinuierlicher Überwachungsprozess** im Produktportfoliomanagement aufgebaut und z. B. kritische Kennwerte oder die Einhaltung von Gestaltungsprinzipien stets überwacht werden. Die effiziente Ermittlung von Kennwerten und die mustergestützten Bewertungen sollen unterstützen, Entscheidungen besser vorzubereiten, nachvollziehbar zu machen, zu objektivieren und die Dokumentation zu erleichtern. Somit wird Anforderung (A6) adressiert.

### 5.3.4 Design und Strukturierung

Regeln aus der Kategorie Design und Strukturierung zeichnen sich dadurch aus, dass der **Schwerpunkt der Regelanwendung auf der Rechten Seite** in der Veränderung des Graphs liegt. Diese Kategorie von Regeln wird auch in der Design Synthese dazu verwendet, einen sehr großen Lösungsraum mit Graphgrammatiken zu generieren. Die Generierung von Netzwerken spielt jedoch in der Systemarchitektur eine untergeordnete Rolle. Hier werden Design- und Strukturierungsregeln eingesetzt, Anpassungen an der Datenbasis effizient durchzuführen.

Auf Grund von **Fehlern, veränderten Randbedingungen** oder auch **aktiven Änderungsbestrebungen** kann es erforderlich sein, die Struktur und die Zusammensetzung der Elemente eines Graphen verändern zu müssen. Manuelle Eingriffe in eine komplexe Struktur können fehleranfällig, monoton und aufwendig sein. Hier werden ein paar ausgewählte Beispiele dargestellt, um das Prinzip zu veranschaulichen. Beispielsweise können Knoten und Kanten neu typisiert, Elemente neu erzeugt oder die Klassifizierung von Elementen verändert werden.

### Retypisierung von Elementen

Eine **Retypisierung** kann schon bei der Datenkonsolidierung vorgenommen werden, wenn das Format der zu importierenden Daten nicht konform zur Definition im Metamodell ist. Sie kann aber auch später in einem bestehenden Graph eingesetzt werden, wenn mit der Zeit bestimmte Datentypen nicht mehr passend erscheinen. Beispielsweise kann es sinnvoll sein, den Datentyp *Anforderung* weiter zu detaillieren in Subtypen wie *Kundenanforderung*, *Interne\_Anforderung*, *Gesetzgebung*, *Normen* etc. Die bestehenden Anforderungen sollen dann ihren Eigenschaften entsprechend retypisiert werden.



**Beispiel (13):** Die Problematik in Beispiel (5) wird hier noch mal aufgegriffen. Teile die Teile enthalten, sind per Definition falsch typisiert und sollen in Baugruppen retypisiert werden.

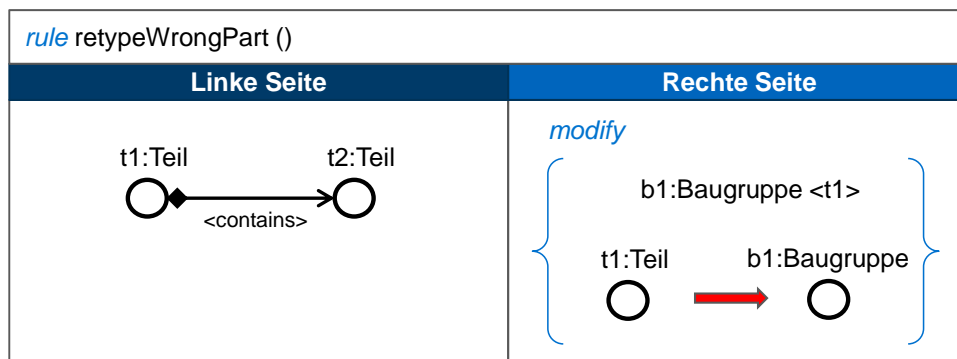


Abbildung 5-20: Beispiel zur Retypisierung von Elementen

Sobald im Graph das Suchmuster auf der Linken Seite gefunden wurde, wird auf der Rechten Seite der fehlerhafte Knoten *t1* vom Typ *Teil* in den korrekten Knoten *b1* vom Typ *Baugruppe* geändert.

### Generierung neuer Elemente

Die Generierung neuer Elemente kann sowohl neue Knoten wie auch neue Kanten umfassen. Neu erzeugte **Knoten** sollten in der Struktur vernetzt werden, wenn sie nicht im Graphen isoliert sein sollen. Dies kann durch direkte Anweisung oder durch Definition von Bedingungen auf Grund logischer Zusammenhänge gemacht werden. Die Erzeugung einer **Kante** in einem Graph kann unterschiedliche Anlässe haben, wie z. B. das Fehlen einer Kante auf Grund von Datenfehler, Visualisierung indirekter Abhängigkeiten, die noch nicht explizit dargestellt waren oder auch Änderungen im Graph, die zu neuen logischen Abhängigkeiten führen.

**Beispiel (14):** In modularen Produkten werden entkoppelte Module angestrebt. Die Abhängigkeiten zwischen verschiedenen Modulen sollen möglichst gering sein im Vergleich zu den Abhängigkeiten innerhalb eines Moduls. Baugruppen erbringen Funktionen. Auch wenn eine 1-1-Verknüpfung zwischen Baugruppe und Funktionen angestrebt wird (vgl. SUH 2001; GÖPFERT 1998; ULRICH 1995), bestehen in der Realität meist n-m-Beziehungen. Somit können indirekte Abhängigkeiten zwischen Baugruppen bestehen, die gemeinsam Funktionen erfüllen. Diese Abhängigkeiten sind allerdings nicht im Graphen abgebildet, sollen aber in diesem Beispiel für weitere Planungen aufgezeigt werden. Die indirekten Abhängigkeiten zwischen Baugruppen sollen daher explizit als Kanten erzeugt und hervorgehoben werden. Falls dieser Kantentyp im Metamodell noch nicht vorgesehen war, muss er vorab definiert werden.

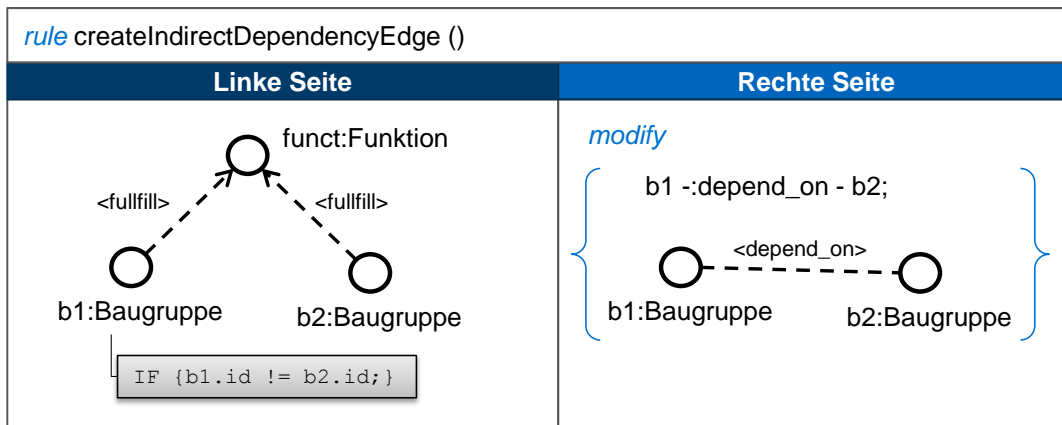


Abbildung 5-21: Beispiel für die Generierung von Kanten

Wenn auf der Linken Seite ein Funktionsknoten *funct* gefunden wird, an dem zwei unterschiedliche Baugruppenknoten *b1* und *b2* hängen, dann wird auf der Rechten Seite eine Kante vom Typ *depend\_on* zwischen *b1* und *b2* erzeugt. In einem weiteren Schritt könnten nun die Kanten indirekter Abhängigkeiten jedes einzelnen Baugruppenknotens gezählt und ihm als Kennzahl zugeordnet werden.

In einem weiteren Beispiel soll die **Generierung von Knoten** gezeigt werden.

**Beispiel (15):** Durch eine Änderung in der Gesetzgebung wird die Anforderung mit der ID 1234 geändert. Im Zuge des Änderungsmanagements sollen alle Produkte, die mit der Änderung verknüpft sind, in einer logischen Gruppe mit der ID 5678 zusammengefasst werden (vgl. Abbildung 5-22).

In diesem Beispiel wird wieder eine Regelsequenz erstellt. Zunächst wird ein Knoten *change* vom Typ *Änderung* erzeugt. Auf der Linken Seite ist keine Vorbedingung nötig, somit bleibt sie leer. Dem erzeugten Knoten wird die *id* 5678 zugewiesen. Die Regel konnte erfolgreich angewendet werden und gibt ein *True* zurück. Daraufhin wird die nächste Regel aufgerufen. Auf der Linken Seite werden Knoten *p* vom Typ *Produktstruktur\_Knoten* gesucht, die eine *satisfy*-Beziehung zu der Anforderung *reg* mit der ID 1234 haben. Es werden hier bewusst Knoten vom übergeordneten Typ *Produktstruktur\_Knoten* gesucht, um neben Produkten ebenso mögliche *satisfy*-Beziehungen von Teilen und Baugruppen zu Anforderungen zu erkennen. Zudem muss der zuvor erzeugte Knoten *change* wieder gesucht werden. Auf der Rechten Seite wird dann eine *contains*-Kante zwischen der Änderung *change* und dem Knoten *p* gesetzt.

Somit sind alle Produkte, Baugruppen und Teile, die mit der geänderten Anforderung verknüpft sind, in einer logischen Gruppe gesammelt. In einer weiteren Regel können beispielsweise die Baugruppenverantwortlichen der logischen Gruppe zugeordnet und somit Abstimmungstreffen entsprechend geplant werden.

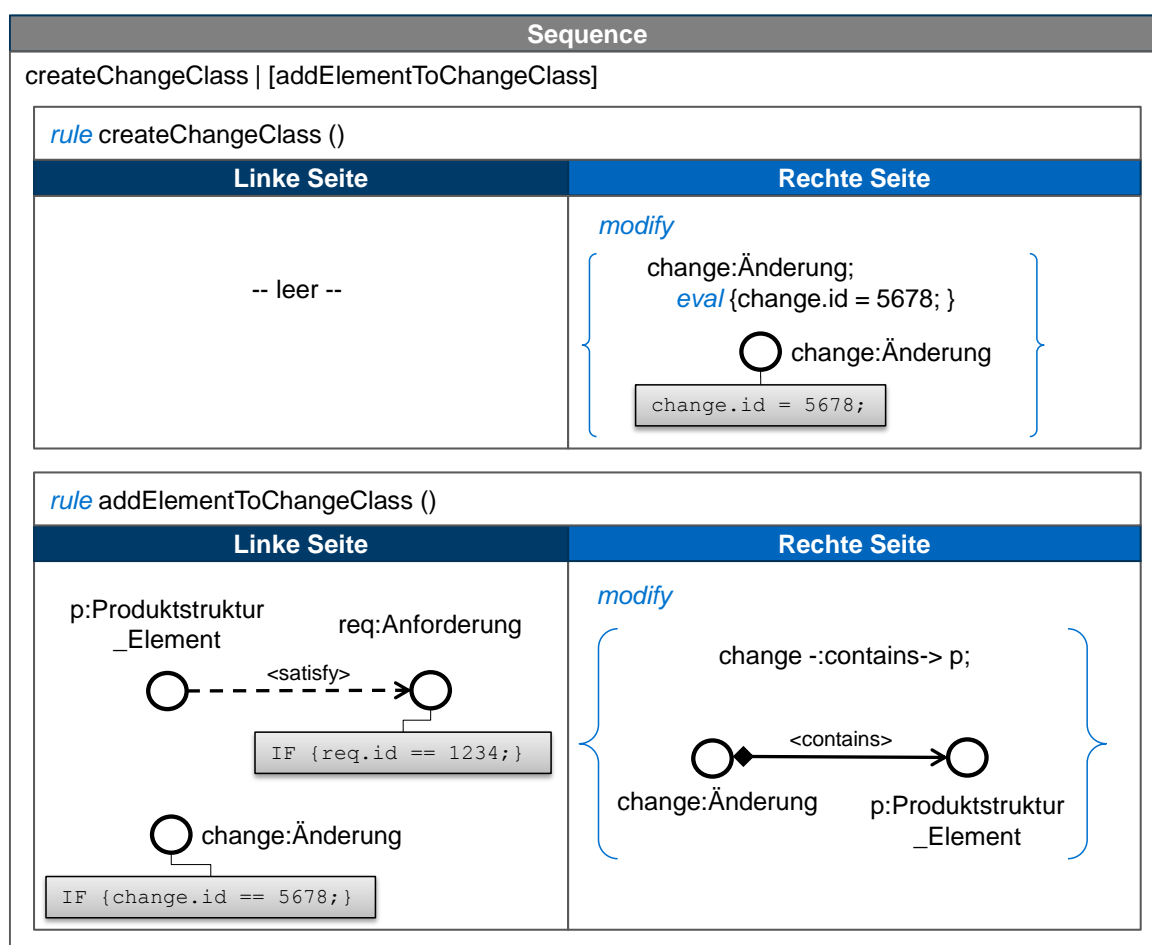


Abbildung 5-22: Beispiel für die Generierung von Knoten und Kanten

## Klassifizierung von Elementen

Bei der Klassifizierung werden Bausteine anhand bestimmter **gemeinsamer Eigenschaften gruppiert**. Diese Eigenschaften können teilweise aus den Attributen erschlossen oder aus der strukturellen Einbindung eines Knoten abgeleitet werden. Weiter oben (vgl. Kapitel 3.1.3) wurden Arten der Klassifizierung in Produktportfolios vorgestellt.

Eine Klassifikation erleichtert das **Management eines Produktportfolios**. Sie ist vergleichbar mit dem Aufräumen einer Werkzeugkiste oder eines Kleiderschranks. Durch Sortieren, Ausmisten und Ordnen wird Überblick geschaffen und unnötige Vielfalt erkannt. Durch das Bilden von Bausteinklassen können differenziertere Strategien angewendet, gezieltere Verbesserungen des Bausteininventars durchgeführt und einfache Gestaltungsprinzipien postuliert werden.

Der hohe Aufwand einer manuellen Klassifikation sowie die Anpassung und Einarbeitung in die Datenbanken wird teilweise gescheut. Jedoch **erhöht** die Klassifikation von Bausteinen die **Ordnung im System** und reduziert somit die Komplexität. Aber auch wenn eine Klassifizierung bereits besteht, können sich Rahmenbedingungen verändern. Die Klassifizierung von Elementen kann dann überholt sein und muss angepasst werden.

Als Beispiel sei auf **Beispiel (8)** verwiesen. Hier wurde zunächst eine Bausteinklasse erzeugt und in einem zweiten Schritt Teilen, die bestimmte Kriterien erfüllen, zugeordnet.

### **Fazit zu Design und Strukturierung**

Die Anpassung des Produktportfolios an veränderte Rahmenbedingungen, die Klassifizierung von Bausteinen, die Behebung von Fehlern und dergleichen sind Aufgaben, die viele Ressourcen in Anspruch nehmen, wenn keine rechnergestützten Verfahren eingesetzt werden können. Mit der Möglichkeit den Graphen und somit die Datenbasis gezielt in seiner Struktur verändern zu können, unterscheidet sich die Graphtransformation von klassischen Datenbankabfragen. Muster können auch in Ontologien durch Einsatz von Abfragesprachen wie SPARQL (vgl. Kapitel 4.1.2) gesucht und gefunden werden. Eingriffe in die Struktur oder Erzeugung und Einbettung neuer Elemente sind nicht oder nur bedingt möglich. Somit wird die Arbeit des Systemarchitekten mit den Methoden zur regelbasierten Generierung von Knoten und Kanten (Design) und Strukturierung effektiv und effizient unterstützt und erleichtert. Damit wird Anforderung (A3) adressiert.

### **5.3.5 Planung und Prognose**

Die fünfte Kategorie umfasst Regelbausteine, die den **Planungsprozess und die Prognose** unterstützen sollen. Hierbei wird angenommen, dass ein graphenbasiertes Produktportfoliomanagement angewendet wird und über einen nennenswerten Zeitraum bereits Kennzahlen erhoben und regelbasierte Veränderungen durchgeführt wurden.

#### **Nachbarschaftsanalyse**

Die Nachbarschaftsanalyse eines Knoten kann für Planungszwecke herangezogen werden, wenn **Auswirkungen von Maßnahmen oder Änderungen** bewertet werden müssen. Der bestimmende Gedanke ist, alle Abhängigkeiten und Wechselwirkungen von betroffenen Knoten mit deren Nachbarelementen aufgezeigt zu bekommen. Somit können alle planungsrelevanten Zusammenhänge systematisch berücksichtigt oder Irrelevantes bewusst vernachlässigt werden. Durch diese Form der Strukturierung und systematischer Analyse der Relationen können Abstimmungsrunden, Terminpläne und Vorgehen sorgfältiger geplant werden ohne Wichtiges zu vergessen.

**Beispiel (16):** Ein Lieferant kündigt ein zugeliefertes Bauteil ab. Die benachbarten Elemente des Bauteils mit der Materialnummer „4567“ sollen angezeigt werden.

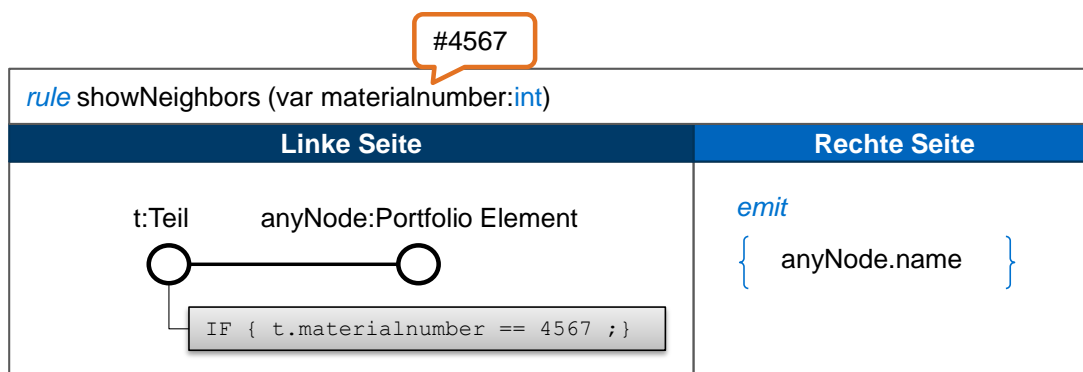


Abbildung 5-23: Beispiel zur Nachbarschaftsanalyse

Der Regel wird zunächst eine Materialnummer als Variable übergeben. Auf der Linken Seite werden beliebige Knoten aller Typen gesucht, die auf das Teil  $t$  mit der Materialnummer „4567“ zeigen. Auf der Rechten Seite werden die gefundenen Elemente ausgegeben. In einem weiteren Schritt kann diese Abfrage noch um die indirekten Abhängigkeiten erweitert werden, wie sie im Beispiel (14) beschrieben wurden.

### Zeitreihenbetrachtungen

Mit der **kontinuierlichen Berechnung von Kennzahlen** können Zeitreihenbetrachtungen durchgeführt werden. Angenommen, es werden kontinuierlich Umsatz- und Deckungsbeitragszahlen aus Sicht eines Marktes, einer Produktfamilie, eines Produktes, einer Baugruppe und eines Teils berechnet und gespeichert. Über die Zeit können Datenreihen erstellt und Entwicklungen analysiert werden. Somit kann ein differenzierteres Bild über die Entwicklungen der Artefakte im Portfolio erzeugt werden. Dies führt zu besserer Planung und Prognosen auf unterschiedlichen Detaillierungsstufen des Portfolios.

Zudem kann die **Strukturveränderung über die Zeit** betrachtet werden. Mit einer regelmäßigen Speicherung von Momentaufnahmen der Struktur („Snapshots“) und der Ähnlichkeitssuche können Veränderungen von Attributen und in der Struktur über die Zeit hervorgehoben werden. Wie bei der Ähnlichkeitssuche müssen die Attribute und Kanten vorab bestimmt werden, die untersucht werden sollen.

**Beispiel (17):** Zwei Momentaufnahmen eines Graphen sollen verglichen werden. Im Ergebnisgraph sollen nur Knoten und Kanten angezeigt werden, deren Attribut *status* sich verändert hat.

Für diese Abfrage müssen zwei Snapshots des Portfoliographen in einen Graphen zusammengeführt werden. Damit die Zusammenführung gelingen kann, wird statt dem eindeutigen Attribut *id* das *timestamp*-Attribut zur eindeutigen Identifikation eines Knotens verwendet. Das Attribut *id* dient in diesem Fall zur Bestimmung von Knotenpaaren. Damit gibt es alle *gleichen* Knoten genau zweimal, die in allen Attributen gleich sind außer dem *timestamp*-Attribut. Die Knotenpaare, die zusätzlich noch im Attribut *status* unterschiedlich sind, sollen im Graph bleiben, Knotenpaare ohne *status*-Änderung werden gelöscht.

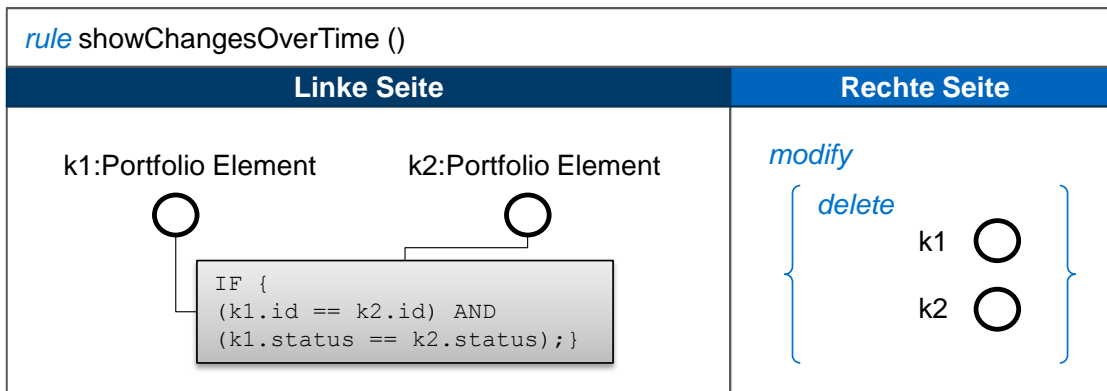


Abbildung 5-24: Beispiel zur Zeitreihenanalyse

Auf der Linken Seite wird ein Knotenpaar *k1* und *k2* gesucht, das die gleiche *id* und den gleichen *status* hat. Diese werden auf der Rechten Seite gelöscht. Knotenpaare mit *status*-Änderung bleiben im Graph enthalten.

### Fazit zu Planung und Prognose

Die Nachbarschaftsanalyse eignet sich zur Orientierung in einem komplexen Geflecht aus Abhängigkeiten und Wechselwirkungen. Sie schafft Transparenz und unterstützt dabei, wesentliche Wechselwirkungen mit benachbarten Elementen umfassend zu berücksichtigen und Maßnahmen weitblickender zu planen. Unter der Annahme, dass bereits über einen längeren Zeitraum Snapshots von Produktportfoliographen mit entsprechenden Zeitstempeln gespeichert wurden, kann eine Zeitreihenanalyse durchgeführt werden und Veränderungen von Attributen und Struktur über die Zeit analysiert werden. Diese ex-post-Analyse kann helfen ex-ante-Prognosen darüber abzugeben, wie sich Elemente, Teilstrukturen und das Gesamtsystem in Zukunft weiterentwickeln. Damit wird Anforderung (A3) adressiert.

## 5.4 Ergebnisaufbereitung

Die Ergebnisaufbereitung kann sehr unterschiedlich gestaltet werden. Die **Ergebnisse der Analyse** sind zunächst einmal **unabhängig von der Visualisierungsform**. Daher kann die Aufbereitung davon abhängig gemacht werden,

- für welchen Zweck die Ergebnisse verwendet werden,
- welche Entscheidungen damit unterstützt werden,
- wem die Ergebnisse zugänglich gemacht werden und welchen Erfahrungshintergrund diese Personengruppe hat,
- welche softwaretechnischen Mittel für die Visualisierung genutzt werden.

Für die Kommunikation komplexer Sachverhalte werden **stakeholder-gerechte Aufbereitungen** der Informationen benötigt, um eine rationale Entscheidungsfindung zu unterstützen. Einige Beispiele hierfür:

- Den Einkäufer interessiert eine Sicht auf das Produktportfolio sortiert nach Eigenerzeugnissen und Zukaufteilen. Er möchte Bündelungsmaßnahmen ableiten und Standardisierungen im Sinne von Industriebaukästen prüfen.

- Ein Mitarbeiter des Vertriebs möchte ggf. die Produkte nach Regionen aufteilen oder eine funktionsorientierte Konfiguration der Produkte erarbeiten.
- Der Entwickler möchte bei der Produktkonzipierung Übernahmeteile aus der Vorgängergeneration festlegen, die Anzahl der Wiederholteile erhöhen, die Konfigurierbarkeit durch konstruktive Maßnahmen an Schnittstellen verbessern, Re-Modularisieren oder auch Änderungsauswirkungen bewerten.

Dementsprechend müssen bereits in der Analyse die Schwerpunkte entsprechend gelegt werden. Die **planungs- und entscheidungsrelevanten Ergebnisse** müssen so in den Attributen der Knoten und Kanten abgelegt werden, dass sie für die Visualisierung gezielt angesprochen werden können. Je nach Erfahrungshintergrund des Betrachters können unterschiedliche Darstellungsformen bevorzugt werden.

Unterschiedliche Ausgabeformate der Analyse z. B. in Form einer Graphml-Datei – ein standardisiertes Datenformat zur Speicherung von Graphdateien – erlauben es, verschiedene **Visualisierungswerkzeuge** einzusetzen. Hier wird das Werkzeug Soley Studio<sup>28</sup> für die Visualisierung verwendet, da damit die Visualisierung am freiesten gestaltet werden kann.

#### 5.4.1 Attribute zur Visualisierung

Ein Graph kann zunächst einmal als Netzwerk aus willkürlich verteilten Knoten und Kanten dargestellt werden. Aus diesem „Knäuel“ wird man aber keine weiteren Erkenntnisse ziehen können. Für die Darstellung des Graphen kann die Visualisierung der Elemente entweder gezielt über **Attributzuweisung** beeinflusst oder deren Anordnung über **Layouter** berechnet werden.

Das Metamodell dieser Methodik sieht die abstrakten Typen **Visueller Knoten** und **Portfolio Kante** an oberster Stelle der Vererbungshierarchie vor. Somit erben alle Knoten und Kanten deren Attribute. Diese Attribute können gezielt für die Visualisierung genutzt werden (vgl. Abbildung 5-25).

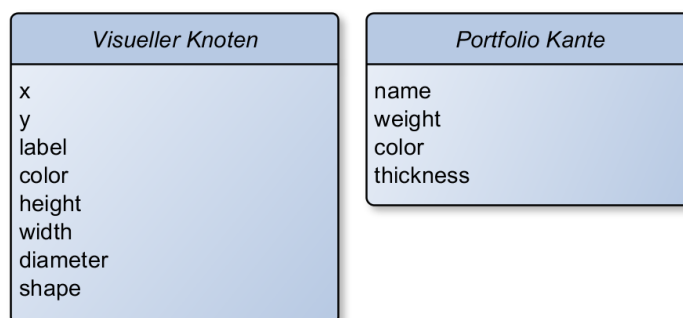


Abbildung 5-25: Abstrakte Knoten- und Kantentypen für die Visualisierung

Der **Visuelle Knoten** stellt die Attribute *x*, *y*, *label*, *color*, *height*, *width*, *diameter* und *shape* zur Verfügung. Somit können die Position eines Knotens in einem x-y-Koordinatensystem definiert

<sup>28</sup> [www.soley-technology.com](http://www.soley-technology.com) (entnommen am 01.12.13)

sowie die Beschriftung, Farbgebung, Größe und Form zugewiesen werden. Mit den Attributen der **Portfolio Kante** können Beschriftung, Farbe und Kantendicke angesprochen werden.

In den Beispielen (4) und (9) wurde bereits innerhalb einer Regel das Farb-Attribut eines Knotens angesprochen, um das **Ergebnis einer Suche hervorzuheben**. Ebenso können **Fehlermuster** in der Struktur hervorgehoben werden.

Durch die Formulierung entsprechender Regeln können den **Visualisierungsattributen Werte zugewiesen** werden. Damit kann beispielsweise die Variation der Knotengröße nach Umsatz und Knotenfarbe nach Stückzahl erfolgen. Nachfolgende Abbildung 5-26 zeigt ein Beispiel für diese attributgesteuerte Veränderung der Knotengröße und –farbgebung.

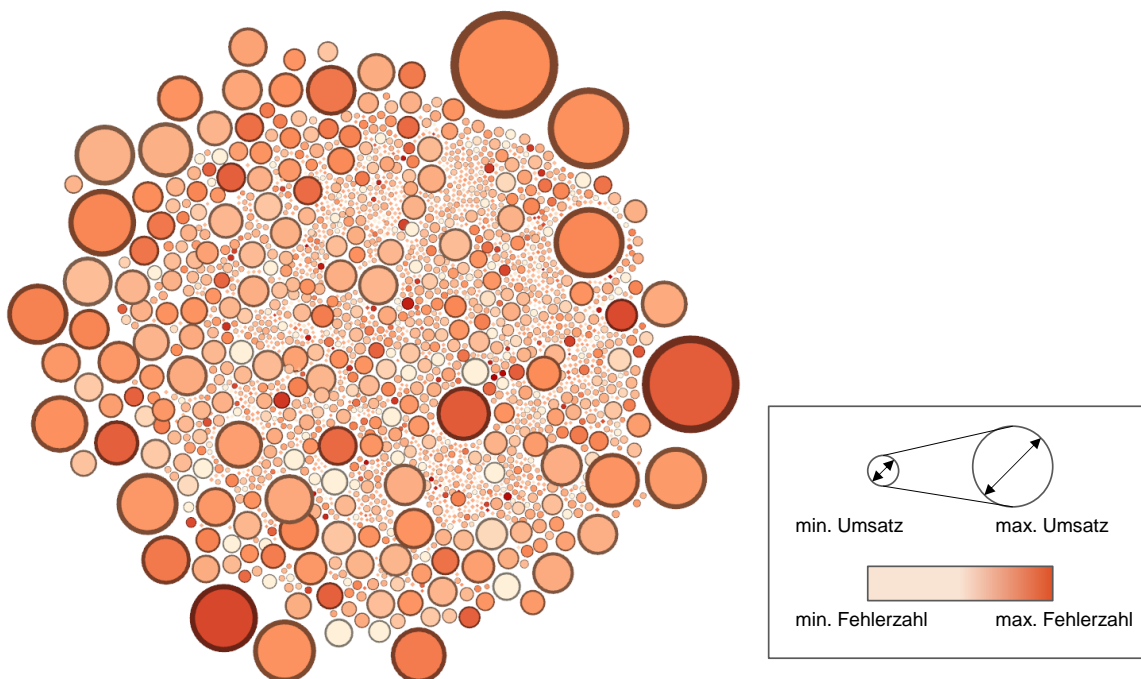


Abbildung 5-26: Beispiel für eine attributgesteuerte Knotengröße und –farbgebung (erstellt mit Gephi<sup>29</sup>)

Zwei **Zuweisungsmethoden der attributgesteuerten Darstellung** von Knoten und Kanten können angewendet werden. Die eine Zuweisung erfolgt durch Vorgabe eines **Verlaufs** („Range“), die andere durch **Kategorisierung** („Partition“).

Ist der **Wertebereich** des zuzuweisenden Attributs **stetig**, dann können ein Minimum und ein Maximum eines Verlaufs bestimmt werden. Beispielsweise haben Attribute wie Gewicht, Umsatz, Stückzahl oder viele berechnete Kennzahlen stetige Wertebereiche.

Der Verlauf kann nun auf die **Farbe, Größe** oder die **x/y/z-Position der Knoten** angewendet werden. Ein Beispiel findet sich in Abbildung 5-27. Mit der Wahl unterschiedlicher **Funktionsverläufe** (z. B. linear, exponentiell, logarithmisch) kann die Verteilung der zugewiesenen Attribut-Werte zwischen dem Minimum und Maximum verändert werden. Ebenso lässt sich der Verlauf für die Veränderung von **Kanten-Farbe** und **-Stärke** einsetzen.

<sup>29</sup> [www.gephi.org](http://www.gephi.org) (entnommen am 01.10.13)



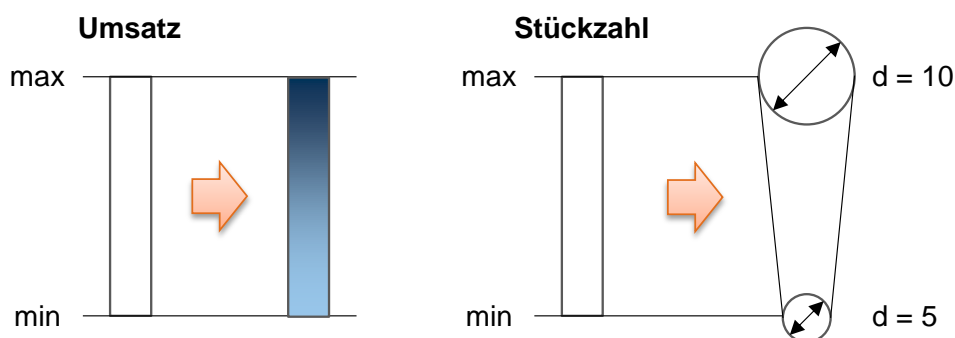


Abbildung 5-27: Attributzuweisung mittels Verlauf

Ist der **Wertebereich** des zuzuweisenden Attributs **diskret**, dann müssen Kategorien bestimmt werden, denen die diskreten Werte des Attributs zugewiesen werden. Nachfolgende Abbildung 5-28 gibt hierzu ein Beispiel.

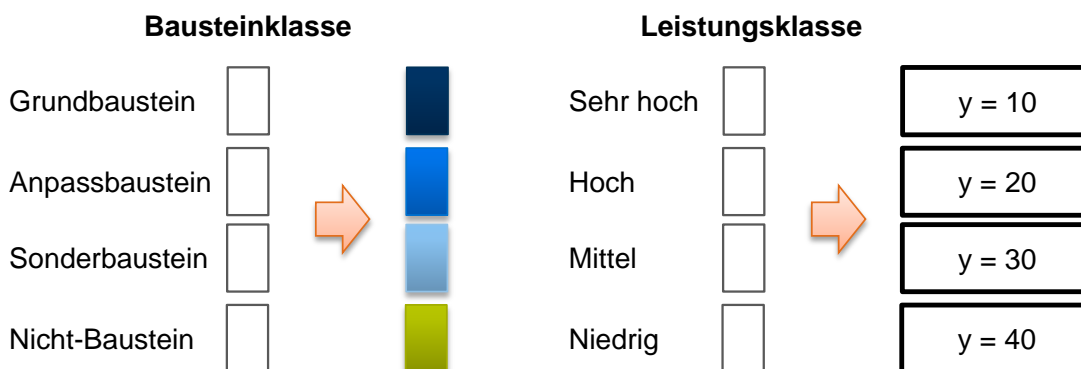


Abbildung 5-28: Attributzuweisung mittels Kategorisierung

In einer Darstellung können beide Zuweisungsmethoden zum Einsatz kommen.

## 5.4.2 Layouter

Layouter sind in zahlreichen Graph-Visualisierungs-Werkzeugen integriert. Sie geben Algorithmen vor, wie Graphen dargestellt werden können. Beispiele sind stärkebasierter Graph, hierarchische oder zirkuläre Anordnung (vgl. Abbildung 5-29).

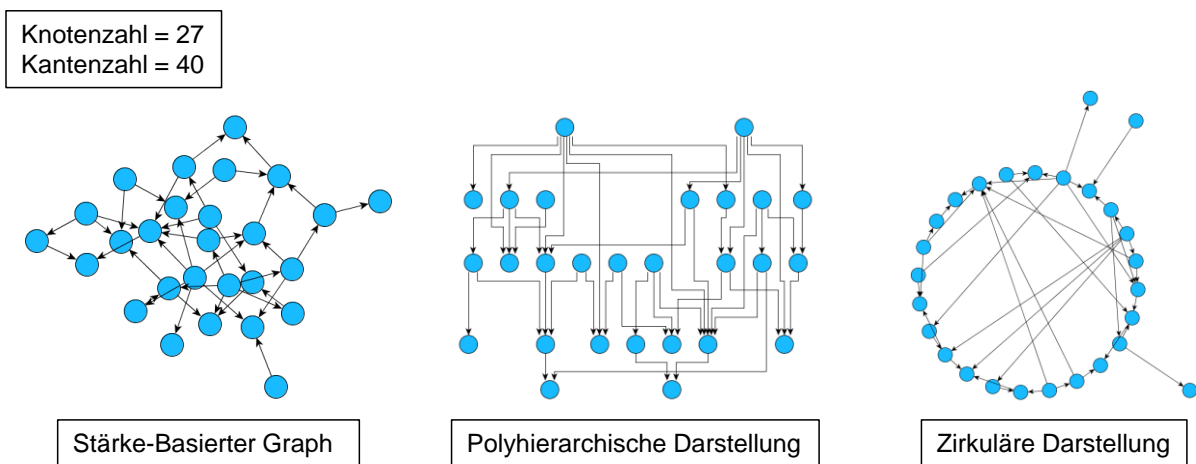


Abbildung 5-29: Layouter im Vergleich (erstellt mit yEd<sup>30</sup>)

Baumdarstellungen sind nur bei streng-hierarchischen Systemen möglich. Im Zusammenhang mit einem Produktportfolio liegen aber meist **polyhierarchische** oder **netzwerkartige** Strukturen vor. Bei Verwendung eines Layouters muss beachtet werden, dass die x- und y-Attribute im Graph nicht durch Attributzuweisung gesetzt werden können.

### 5.4.3 Weitere Visualisierungsformen

Alternativen zur graphenbasierten Visualisierung können ebenfalls in Betracht gezogen werden:

- **Textform:** Die einfachste Form das Ergebnis einer Graphtransformation auszugeben, ist ein Text.
- **Tabellenform:** Die Ergebnisse können auch in Tabellenform, nach bestimmten Attributen sortiert, ausgegeben werden.
- **Matrixform:** Wie im Stand der Technik dargestellt wurde, können Graphen- und Matrizenform einfach ineinander übersetzt werden. Bei größeren Datenmengen werden diese Formen der Darstellung allerdings rasch unübersichtlich. Dennoch eignen sich Textausgabe, Tabellen und auch Matrizenform besonders dazu, rasch Zwischenergebnisse z. B. während einer Regelentwicklung zu visualisieren oder Fehler manuell zu korrigieren.
- **Venn-Diagramme:** Aus der Mengenlehre sind die Venn-Diagramme bekannt. Diese können beispielsweise zur Visualisierung von Gleichteilebetrachtungen oder Ähnlichkeitsanalysen herangezogen werden.
- **Portfolio-Darstellung:** In den Wirtschaftswissenschaften ist diese Darstellung geläufig, aber auch zur Darstellung von Einflussportfolios in der Produktentwicklung werden Portfolio-Darstellungen eingesetzt. Durch Attributzuweisung der x- und y-Werte können diese aufwandsarm erzeugt werden.

<sup>30</sup> [http://www.yworks.com/de/products\\_yed\\_about.html](http://www.yworks.com/de/products_yed_about.html) (entnommen am 01.10.13)

#### 5.4.4 Fazit zur Ergebnisaufbereitung

Die Methode zur intuitiven Visualisierung relevanter Zusammenhänge weist ein hohes Maß an Flexibilität auf, so dass die Ergebnisse der Graphtransformation situations- und stakeholdergerecht aufbereitet und visualisiert werden können. Die Ergebnisse liegen in einem neutralen Datenformat vor, das unabhängig von der Visualisierungsform ist. In der Visualisierung können Attribute in dem Ergebnisgraph zielgerichtet angesprochen werden, um entscheidungsrelevante Schwerpunkte hervorzuheben. Allerdings können auch Layouter in Graphbetrachtungswerkzeugen und andere Visualisierungsformen genutzt werden, die Ergebnisse auf unterschiedliche Art und Weise effizient zu visualisieren. Somit wird Anforderung (A5) adressiert.



## 6. Evaluation des Lösungsansatzes

Der Mehrwert des vorgestellten Lösungsansatzes soll in zwei Fallstudien im industriellen Umfeld veranschaulicht werden. In der Fallstudie bei der Hilti Entwicklungsgesellschaft mbH ist es zunächst das Ziel, Transparenz im bestehenden Produktportfolio herzustellen (vgl. Kapitel 6.1). Durch eine Automatisierung zeitraubender wiederkehrender Tätigkeiten sollen dann im nächsten Schritt die Grundlagen für typische Entscheidungssituationen im Produktportfoliomanagement effizienter vorbereitet werden können. Hier liegen die Schwerpunkte auf der Evaluierung der Methoden zur Zusammenführung von verteilten Daten, Automatisierung repetitiver und aufwendiger Aufgaben und die Visualisierung der Ergebnisse zur Entscheidungsunterstützung.

Die zweite Fallstudie wurde bei der Elektronik GmbH durchgeführt (vgl. Kapitel 6.2). Hier war es das Ziel, eine Entscheidungsgrundlage für Maßnahmen hinsichtlich eines marktgerechteren Produktportfolios zu schaffen. Hierzu sollen mit Hilfe der Mustererkennung unwirtschaftliche Produkte identifiziert und abgekündigt werden. Durch die Verästelung der Abhängigkeiten zwischen den Produkten ist diese Aufgabe mit konventionellen Mitteln nicht darstellbar. In der zweiten Fallstudie soll der Schwerpunkt der Evaluierung auf der Zusammenführung relevanter Daten, der automatischen Mustererkennung und der intuitiven Visualisierung der Ergebnisse liegen.

### 6.1 Fallstudie 1

Die Hilti Gruppe ist ein in Schaan, Liechtenstein, ansässiges Familienunternehmen. Hilti beschäftigt weltweit 21.000 Mitarbeiter und ist in etwa 120 Ländern vertreten. Unter anderem unterhält das Unternehmen einen Entwicklungs- und Produktionsstandort in Kaufering<sup>31</sup>. Das Unternehmen ist Hersteller von Produkten für professionelle Anwender in der Bauindustrie und der Gebäudeinstandsetzung. Das Produktportfolio beinhaltet unter anderen das Geschäftsfeld Elektrogeräte mit den Produktlinien Bohr- und Abbautechnik, Säge- und Schleiftechnik, Diamanttechnik, Schraubtechnik und Akkutechnik. Weitere Produktlinien sind die Messtechnik, die Direktmontage, Dübeltechnik, Installationstechnik und die Brandschutztechnik. Im Rahmen der Fallstudie wurde ein Ausschnitt aus dem Geschäftsfeld Elektrogeräte näher untersucht.

#### 6.1.1 Ausgangssituation und Problemstellung

Die Fallstudie wurde mit dem Verantwortlichen für die mechanische Antriebskonstruktion und dem Projektmanager für Plattformentwicklung durchgeführt. Die Entwicklungsabteilung für elektrische Antriebe bedient die für die Geräteentwicklung verantwortlichen Business Units mit

---

<sup>31</sup> Informationen der Unternehmenswebseite unter <https://www.hilti.de/hilti-im-ueberblick> (entnommen am 01.11.2013)

Elektromotoren, den zugehörigen Steuerungselektroniken und Akkutechnologie. Über die Zeit ist das Antriebsportfolio stark angewachsen. Seit 2004 sind Elektronik- und Antriebsentwicklung, Systemtest, Simulation und der Prototypenbau als Querschnittsfunktion in der Entwicklung integriert (DEIMLING 2013, S. 30f). Bei einer Geräteentwicklung bilden Entwickler und Versuchsingenieure aus den Querschnittsfunktionen mit Mitarbeitern der Business Units das Kernteam in einem Entwicklungsprojekt. Die Aufgaben des Portfoliomanagements im Bereich Antriebe werden durch den Entwicklungsleiter, die Gruppenleiter und den Projektleiter Plattformen wahrgenommen. Nachfolgende Abbildung 6-1 zeigt die Organisation der Hilti Geräteentwicklung.

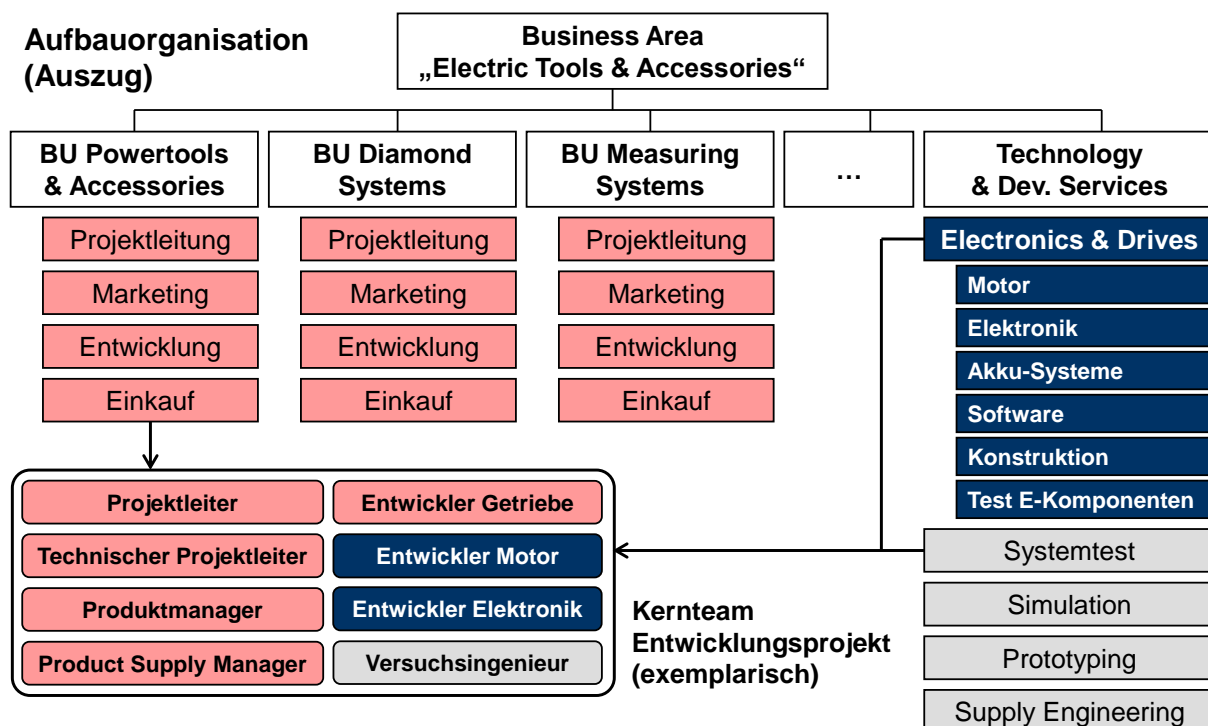


Abbildung 6-1: Auszug aus der Organisation der Hilti Geräteentwicklung (Darstellung Fa. HILTI)

Die Geräte sind klassische „built-to-stock“ Produkte. Bei den sogenannten „Flagship-Products“ handelt es sich um markendefinierende Geräte, bei denen mit hohen Investments eine Differenzierung von Wettbewerbern angestrebt wird. Bei „Portfolio“-Geräten liegt hingegen höheres Augenmerk auf den Kosten (LOCK 2013, S. 33). Der Endkunde kann in der Regel aus länderspezifischen Varianten und einer begrenzten Auswahl an Zusatzfunktionen optional auswählen. Eine kundenspezifische Konfiguration von Geräten wird in der Regel nicht durchgeführt. Die Geräte werden demnach gemäß den Anforderungen der Zielmärkte entwickelt. Hierbei herrscht eine hohe Dynamik hinsichtlich der Anforderungen und in der Planung der Roadmaps. Dieser Umstand erfordert ein flexibles Reaktionsvermögen seitens der Antriebsentwicklung.

Die Stakeholder der Antriebsentwicklung sind die Geschäftseinheiten (Business Units genannt: „BU“), die jeweils die unterschiedlichen Produktlinien entwickeln. Die Auftragsituation lässt sich in dieser Konstellation eher als „engineered-to-order“ beschreiben. Die Motoren, Steuerungselektronik und Schalter weisen breite Varianz auf. Mit klar strukturierten

Plattformansätzen und Baukastensystematik soll eine effiziente Verwendung der Einzelbausteine erreicht werden (vgl. Abbildung 6-2).

Je nach Anforderungsprofil, das mit einem neuen Gerät an den Antrieb gestellt wird, müssen mit vorhandenen und neu zu entwickelnden Bausteinen die gewünschten Antriebe frist- und kostengerecht entwickelt werden. Auf Grund typischer Zielkonflikte hinsichtlich Kosten, Gewicht, Leistung und angestrebter Marktreife, ist eine intensive Abstimmung mit den verantwortlichen BUs, der Produktion und dem Einkauf erforderlich. Entsprechend wird eine transparente und stakeholder-gerechte Gesprächsgrundlage benötigt, die wesentliche Zusammenhänge aufzeigen kann.













<b>Produktlinie</b>	C-Combi / Rotary	Sawing	Screw Fastening	Dia Drilling
<b>Produktfamilie</b>	Akku-Bohrhammer	Akku-Säge	Netzschrauber	Dia Kernbohrgerät
<b>Produkt (Variante)</b>	Akku-Bohrhammer 1 	Akku-Säbelsäge 1 	Trockenbauschrauber 1 	Dia Kernbohrgerät 1 
<b>Motor (Variante)</b>	Motor Akku-Bohrhammer 1	Motor Akku-Säbelsäge 1	Motor Trockenbauschrauber 1	Motor Dia Kernbohrgerät 1
<b>Baugruppen / Bauteile (Varianten)</b>	Lüfter	Lüfter	Lüfter	Lüfter
	Stator komplett	Stator komplett	Stator komplett	Stator komplett
	Rotor komplett 	Rotor komplett 	Rotor komplett 	Rotor komplett 
	Rotorwelle	Rotorwelle	Rotorwelle	Rotorwelle
	Kommutator	Kommutator	Kommutator	Kommutator
	Blechpaket	Blechpaket	Blechpaket	Blechpaket
	Rotorblech 	Rotorblech 	Rotorblech 	Rotorblech 
<b>Motorplattform</b>	DC 1	DC 1	UNI 1	UNI 2

Abbildung 6-2: Produkte der Business Units und Komponenten der Antriebsentwicklung (Darstellung Fa. HILTI)

## Ziele der Antriebsentwicklung

In der Antriebsentwicklung wird das Ziel verfolgt, auf Anfragen der Business Units mit **hohem Reaktionsvermögen** schnell und flexibel reagieren zu können. Wenn also ein neues Gerät oder eine Gerätefamilie entwickelt werden, wird die Abteilung angefragt, in welchem Zeitraum zu welchen Kosten sie einen neuen Antrieb mit den geforderten Eigenschaften entwickeln kann.

Zum einen sollen **verlässliche Aussagen** zum geschätzten Entwicklungsaufwand getroffen werden können, zum anderen soll auch schnell geprüft werden können, welche Umfänge aus dem **bestehenden Vorrat an Bausteinen wiederverwendet** werden können und wie hoch der **Anpassungsaufwand** geschätzt wird. Man setzt sich zum Ziel, eine **flexiblere Konfigurierbarkeit** innerhalb des Antriebs-Programms sowie eine wirtschaftlichere Nutzung

von **Kommunalitäten** im Produktprogramm sowohl **auf der Komponenten-** wie auch auf der **Konzeptebene** zu erreichen.

Mit der richtigen **Baukastensystematik, Modularisierung** von Baugruppen und Nutzung von **Wiederhol- und Gleichteilen** soll mit möglichst wenigen Bausteinen im Portfolio ein möglichst breites und marktgerechtes Antriebe-Spektrum ausleitbar sein. So sind heute gleiche oder ähnliche Motoren in unterschiedlichen Geräten eingesetzt. Es ist dabei jedoch nicht das primäre Ziel, Variantenvielfalt zu vermeiden. Hilti möchte sich mit einer hohen **Innovationsfähigkeit** von seinen Wettbewerbern **am Markt** unter Beachtung der Wirtschaftlichkeit **differenzieren**. Die **Erzielung von Skaleneffekten** in Entwicklung, Produktion und Einkauf wurde speziell im Antriebsbereich in den letzten Jahren immer wichtiger (DEIMLING 2013, S. 33).

Um diese Ziele erreichen zu können, muss **Transparenz** über den **Ist-Stand des Antriebsportfolios** effizient hergestellt werden können. Somit können Handlungsfelder einfacher eingegrenzt werden, wo unnötige Vielfalt eingeschränkt und Komplexität reduziert werden kann.

### Identifizierte Herausforderungen

In Rahmen von zwei Studienarbeitsprojekten, drei Workshops mit Führungskräften und Plattformexperten in der Antriebsentwicklung und fünf Expertengesprächen mit dem Leiter Konstruktion Antriebsentwicklung und dem Projektleiter Plattformentwicklungen wurden über einen Zeitraum von zwölf Monaten die Herausforderungen der Antriebsentwicklung bei Hilti aufgenommen. Zudem wurden Zwischenergebnisse und mögliche Lösungsansätze gemeinsam diskutiert und weiterentwickelt. Dabei wurde erarbeitet, dass folgende Hürden in den Themenfeldern Produkte, Organisation und Prozesse unter anderem die Erreichung der oben genannten Ziele erschweren (in Anlehnung an LOCK 2013, S. 41 und DEIMLING 2013, S. 38):

#### Produkte

- **Hohe über die Jahre gewachsene interne Variantenvielfalt**, insbesondere auf Komponentenebene. Dadurch entstand auch ungewollte Komplexität durch Anlegen ähnlicher Bausteine im PDM-System.
- Hoher Aufwand, um **Transparenz** über die zahlreichen Zusammenhänge zwischen Produktlinien, Entwicklungsplattformen, Geräten, Baugruppen, und insbesondere über Gleich- und Wiederholteile zu schaffen.
- Teilweise **unklare Strukturierung** des Antriebsportfolios und zulässige Konfigurationsmöglichkeiten.
- Teilweise **fehlende Klassifikationen** und Organisation von Bausteinen und Schnittstellen als Grundlage für einen Plattform- oder Baukastenansatz.

#### Organisation

- Zahlreiche Stakeholder mit unterschiedlichen, teils **entgegengesetzten Interessenslagen** müssen auf Grund der Matrixorganisation berücksichtigt werden.
- Stakeholder haben **unterschiedliches Verständnis und Sichtweisen** sowohl auf das Geräte- als auch auf das Antriebsportfolio.



- Es fehlt ein Werkzeug dafür, die benötigten Informationen aufwandsarm für die **Kommunikation mit Stakeholdern** zusammenzustellen und zu visualisieren.

### Prozesse

- Hoher **Aufwand für die Daten-Beschaffung, -Aufbereitung und -Visualisierung** als Kommunikationsgrundlage mit den Stakeholdern und zur Unterstützung von Entscheidungssituationen im Produktportfoliomanagement.
- **Aufwendige Verwendungsnachweise und Auffinden** bereits vorhandener Bausteine anhand von Kriterien erschweren eine effiziente Wiederverwendung von Bausteinen.
- Informationen zu Produkten werden **unterschiedlich genau dokumentiert und gepflegt**. Die Datenbasis für eine umfassende Analyse ist meist nicht vollständig oder nicht explizit digital verfügbar.
- Für die Planung und Steuerung relevante **Informationen sind verteilt** in Datenbanken, lokalen Dateien und in den Köpfen der Entwickler.
- Hohe, teilweise **marktbedingte Dynamik** in der Planung der Roadmap führt zu Planungsunsicherheiten und erfordert schnelle Reaktionsgeschwindigkeiten.
- Zahlreiche **parallel laufende Projekte** mit Wechselwirkungen zu Plattform-Entwicklungs-Umfängen.

In der Fallstudie spiegeln sich die Hürden wider, wie sie in Kapitel 3.3.2 geschildert wurden. Durch gewachsene Strukturen stieg die Komplexität des Produktportfolios an, während die Transparenz sank. Nur mit hohem Aufwand können Daten beschafft werden, um wiederum aufwendige Analysen und Bewertungen durchführen zu können. Dadurch wird die **Kommunikation mit den Stakeholdern** der Antriebsentwicklung, aber auch die **Planung, Steuerung und Abstimmung innerhalb der Abteilung** erschwert.

Um mehr Transparenz über das Portfolio und seine inneren Zusammenhänge zu schaffen, wurden bereits Methoden entwickelt, die Kommunikation und Entscheidungsfindung im Unternehmen zu unterstützen. Mit der sogenannten „**Tapete**“ werden die Zusammenhänge zwischen Motoren- und Geräteprogramm übersichtlich visualisiert. Aus der Tapete wird zudem ersichtlich, welche Statorblechschnitte und Blechpaketlängen in den jeweiligen Motoren verbaut sind. Das Dokument ist ein geeignetes Planungs- und Kommunikationsinstrument. Im Dialog mit den Business Units wird die Tapete eingesetzt, um schnell Zusammenhänge klar machen zu können.

Der exemplarische Ausschnitt in Abbildung 6-3 zeigt den Zusammenhang zwischen beispielhaften Motoren der Gleichstrom (DC)- und Universal (UNI)-Technologie, möglichen Blechschnitten und Blechpaketlängen und in welchen Geräten die jeweiligen Motoren verbaut sind. Daneben wurden mittlerweile weitere Tapeten erarbeitet, die ausschnittsweise Zusammenhänge zwischen dem Geräte- und Antriebs-Portfolio visualisieren (z. B. für Motorelektroniken, Akkus, Schalter etc.).

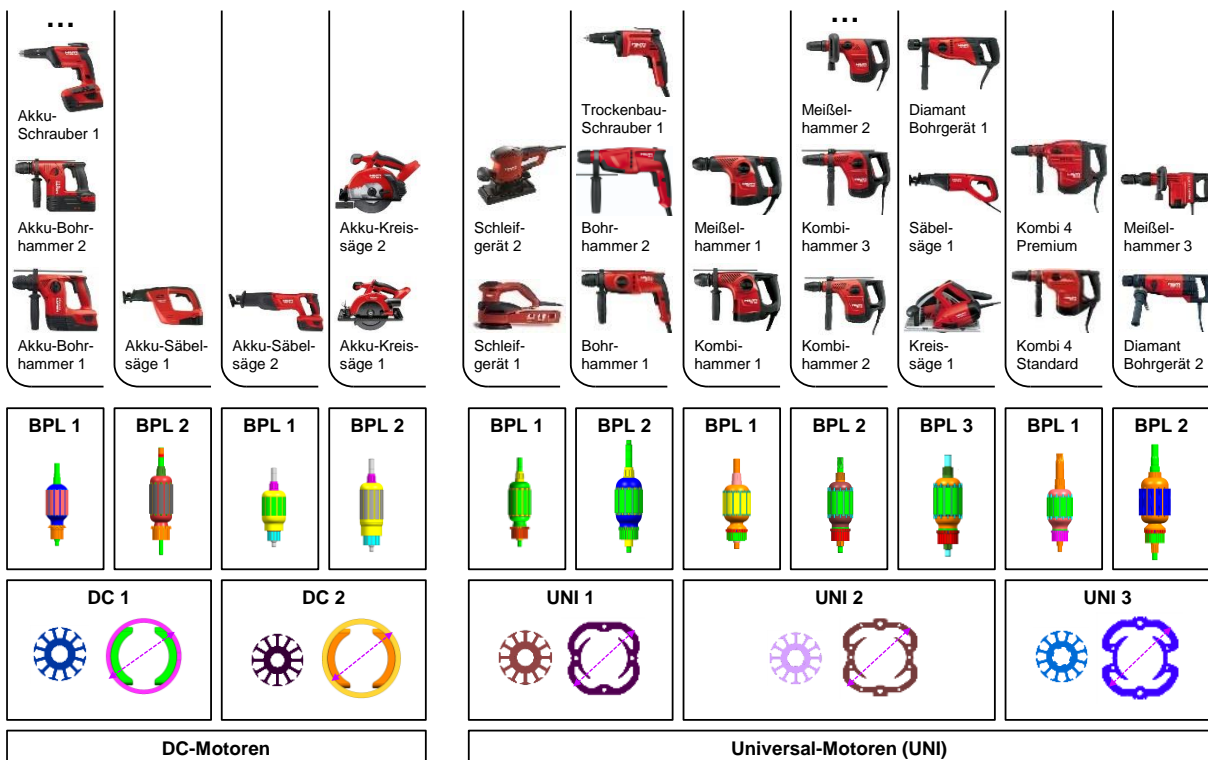


Abbildung 6-3: Kompakte Darstellung des Motoren- und Geräteprogramms in Form einer „Tapete“ (BPL = Blechpaketlänge, Darstellung Fa. HILTI)

Allerdings ist es mit **hohem Aufwand** verbunden, das Dokument aktuell und konsistent zu halten. Derzeit wird es in MS PowerPoint™ gepflegt. Der **Umfang der Daten** ist jedoch so groß, dass immer nur Ausschnitte abgebildet werden können. Änderungen im Geräte- oder Antriebsportfolio ziehen eine meist aufwendige, aber wenig anspruchsvolle Umsortierung von Bild- und Graphikelementen nach sich.

Um die Aktualität gewährleisten zu können, müssen regelmäßig **verteilte Informationen** zusammengetragen werden. Dafür müssen Datenexporte aus datenbankgestützten PDM-System (hier: SAP) und dem Product Data Publishing Portal erzeugt werden, mit Informationen aus lokal gespeicherten Dokumentationen ergänzt und durch Abfrage der Mitarbeiter zusätzlich angereichert werden. Das Fachwissen ist zumeist implizit in den Köpfen der Experten vorhanden, jedoch somit gerade in Situationen der Entscheidungsfindung in der Architekturgestaltung nicht immer verfügbar (DEIMLING 2013, S. 36).

LOCK (2013, S. 80ff) hat den **Beschaffungsaufwand** der benötigten Datenbasis untersucht. Der Aufwand entsteht dabei durch die Datenmenge, die Aktualität und Detailtiefe der Daten. Anforderungen, Funktionen oder Produkteigenschaften müssen aus den Projektunterlagen zusammengetragen werden. Angaben über Stückzahlen, Umsätze oder Ähnliches sind nur mit den entsprechenden Zugriffsrechten abrufbar, liegen dann aber meist nicht in der passenden Aggregationsstufe vor. Beispielsweise sind Daten zu den jeweiligen Geräten einfach zu bekommen, auf Komponentenebene muss eine aufwendige Datenerhebung erst beauftragt werden. Daten wie Materialnummern, Stücklisten von Komponenten, Baugruppen und Geräten sind zwar gut gepflegt, die große Datenmenge erfordert allerdings langwierige manuelle Aufbereitung in Tabellenkalkulationsprogrammen oder ähnlichen Werkzeugen.

Aus Nutzersicht wäre eine häufigere Überarbeitung der Tapete zwar wünschenswert, aber angesichts des Aufwandes nicht praktikabel. Mit dieser Darstellungsform wurde ein erster guter Anfang gemacht. Die Informationsdichte, Aktualisierbarkeit und Skalierbarkeit des Dokuments können jedoch verbessert werden. In typischen Situationen wie der Beauftragung eines neuen Motors oder der Abkündigung einer Elektronikkomponente durch einen Zulieferer ist die Aktualität der „Tapete“ essentiell, aber mit den derzeitigen Mitteln nicht wirtschaftlich zu bewerkstelligen. Im nächsten Abschnitt wird skizziert, wie der aktuelle Stand der Planungsinstrumente mit Hilfe der hier entwickelten Methodik weiterentwickelt werden soll.

### 6.1.2 Zielstellung des Methodeneinsatzes und Vorgehen

Durch die Anwendung der Methodik soll eine **Effizienzsteigerung der Informationsaufbereitung und -Analyse** erreicht werden. Zudem sollen die Bemühungen zur Weiterentwicklung der Plattformstrategien im Antriebsbereich bei Hilti durch geeignete **Klassifizierung und Visualisierung** unterstützt und somit die **Struktur des Ist-Portfolios klarer** werden.

Das Konzept der „**Tapete**“ wird aufgegriffen und mit den Möglichkeiten der Graphtransformation verbessert. Somit kann die Tapete aufwandsarm in unterschiedliche Visualisierungsformen gebracht werden, was durch manuelle Bearbeitung zu aufwendig wäre und unverhältnismäßig viele Ressourcen binden würde. Dadurch soll erreicht werden, dass die **Kommunikation im Unternehmen und Entscheidungen in der Planung und Steuerung der Portfolioentwicklung**, die auf diesen Zusammenhängen basieren, **effektiver und effizienter vorbereitet** werden können. Statt die „Tapete“ manuell in Microsoft PowerPoint™ aktuell und konsistent zu halten, soll eine „Tapete auf Knopfdruck“ erzeugt werden können.

Mit der Fallstudie sollen die Prinzipien der Methodik und deren Übertragbarkeit auf einen industriellen Kontext evaluiert werden. Es soll aufgezeigt werden, dass durch Einsatz der Methodik, die Transparenz über die Zusammenhänge des Portfolios verbessert werden kann, sich effizientere Abläufe darstellen lassen und effektiver Absprachen und Entscheidungen unterstützt werden können.

### 6.1.3 Ergebnisse

#### **Aufbau der Datenstruktur**

Für die Fallstudie wurde von den Verantwortlichen bei Hilti ein reduzierter, repräsentativer Datensatz definiert und zur Verfügung gestellt. Er umfasst 13 Geräte und Motoren. Die Geräte sind in acht Produktlinien organisiert; die Motoren lassen sich drei Technologien und sieben Plattformen zuordnen. Zusätzlich sind für jeden Motor Strukturstücklisten verfügbar. Anspruch bei der Auswahl der Geräte war es, dass relevante Zusammenhänge bestehen und die Prinzipien der Methodik angewendet und somit evaluiert werden können. Die Informationen wurden zusammengeführt aus den Unternehmensdatenbanken PDM-System, Product Data Publishing Portal und entsprechender Dokumentationen, die in Tabellenkalkulationsprogrammen auf lokalen Laufwerken vorlagen. Informationen, die nicht explizit vorlagen, wurden von Experten

in der Abteilung abgefragt und in die Datenbasis aufgenommen (LOCK 2013, S. 78ff). Für die Studie und eine spätere Veröffentlichung wurden die Daten verfremdet. Die Ergebnisse lassen sich jedoch über einen Zuordnungsschlüssel wieder auf die Originaldaten zurückführen und somit durch die Experten validieren.

Der neu eingefügte Typ *Motor* ist eine Spezialisierung des Typs *Baugruppe*. Im Metamodell aus Kapitel 5.2 wurde der Elementtyp *Produkt* um die folgenden Hilti-spezifischen Attribute ergänzt:

- Produkt
  - energieverorgung [Netz- oder Akkubetrieb]
  - merker [-]
- Motor
  - motortyp [Technologie, wie UNI, SR, DC etc.]
  - statorpaketlaenge [in mm]
  - rotorpaketlaenge [in mm]
  - minleistung [in W]
  - maxleistung [in W]
  - merker [-]

Mit dem erweiterten Basis-Metamodell aus Kapitel 5.2.2 wurde das Produktportfoliomodell mit allen zur Verfügung gestellten Daten instanziiert und ein Graph erzeugt.

### Konsolidierung der Daten

Bevor der Graph für Analysen verwendet werden kann, müssen bestimmte Aspekte der Daten aufbereitet werden. Durch den Import der Beispieldaten ergibt sich ein Typisierungsproblem. Die Elemente der 13 Strukturstücklisten der Motoren wurden zunächst alle mit dem Typ *Teil* importiert, da in der Importvorlage lediglich die Auflösungsstufen, nicht aber die Knotentypen explizit angegeben sind (vgl. Abbildung 6-4). Problematisch dabei ist, dass Baugruppen **fehlerhaft** als *Part* **typisiert** sind. Dies kann allerdings durch eine einfache Transformation behoben werden. Daher wurde die Regel im Beispiel (13) aus Kapitel 5 angewendet, um fehlerhaft typisierte Knoten zu retypisieren. 186 Baugruppen wurden somit retypisiert, 519 Teile blieben unverändert. Als Ausgabe erfolgt der Report in folgender Form (Ausschnitt):

[...]

Wrong part found: Rotor -contains-> Rotorwelle. Retyping Rotor to Module

Wrong part found: Stator -contains-> wicklung. Retyping Stator to Module

Wrong part found: Rotorpaket -contains-> Rotorblech. Retyping Rotorpaket to Module

Correct HiltiPart: Schutzblech

Correct HiltiPart: Kontaktfeder

Correct HiltiPart: Magnetkern

[...]

Komponenten des Motor Akku-Bohrhammer 2			
Auflösungsstufe	Positionsnr	Komponentennummer	Objektkurztext
.1	0001	368367	Rotor komplett
.2	0001	362815	Rotor Blechpaket
...3	0001	362809	Rotorwelle
...3	0002	201758	Rotorblech
.2	0002	201763	Kommutator
.2	0003	368368	Wicklung
.2	0004	201643	Nutisolation
.1	0003	360763	Stator komplett
.2	0001	360727	Ring
.2	0002	360728	Magnet
.1	0004	415330	Schleuderscheibe
.1	0005	201637	Rillenkugellager
.1	0006	203849	Lüfter
.1	0007	203848	Luftführung
.1	0008	361565	Ritzel
.1	0010	348287	
	0013	407041	

Abbildung 6-4: Auszug aus einer Stückliste zu importierender Knoten

Ein weiterer Aspekt, der eine Nachbereitung des Graphen erfordert, ergibt sich aus dem Ziel der **Klassifizierung von Produkten** nach dem klassifizierungsbestimmenden Attribut *statordurchmesser*. Der Außendurchmesser der Statorbleche wurde bei Hilti im Zuge der Vereinheitlichung der Bezeichnung der Motoren als klassifizierungsbestimmendes Merkmal definiert. Zum Beispiel Gleichstrommotoren mit einem 52 mm großen Durchmesser des Statorblechs werden sinngemäß der „DC 52“ Entwicklungsplattform zugeordnet. Das Attribut *statordurchmesser* wird daher im Metamodell im übergeordneten abstrakten Typ Portfolio Element zugeordnet. Das in einem Motor (mehrfach) enthaltene Teil Statorblech soll nun den Wert seines Attributs *statordurchmesser* an das Produkt weitergeben. Nachfolgende Abbildung 6-5 beschreibt die vorgenommene Operation graphisch.

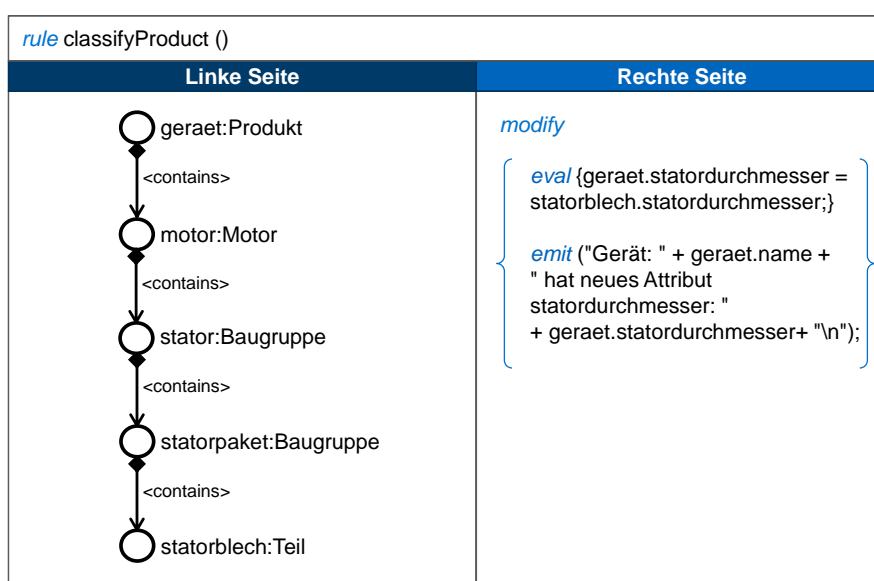


Abbildung 6-5: Zuordnung des klassifizierungsbestimmenden Attributs *statordurchmesser* zu den beinhaltenden Geräten

Auf der Linken Seite der Regel wird nach *contains*-Ketten von Geräten zu Statorblechen gesucht. Wenn ein solches Muster gefunden wurde, wird auf der Rechten Seite der Wert des Attributs *statordurchmesser* dem Produkt zugewiesen und ein kurzer Report ausgegeben. Folgender Report wurde dabei generiert (Ausschnitt):

Gerät: Akku-Bohrhammer 1 hat neues Attribut statordurchmesser: 52

Gerät: Akku-Bohrhammer 2 hat neues Attribut statordurchmesser: 52

Gerät: Bohrhammer 1 hat neues Attribut statordurchmesser: 60

[...]

Nachfolgende Abbildung 6-6 zeigt den vollständigen Graph nach der Konsolidierung, der nun in dieser Form als Datenbasis für weitere Regelanwendungen genutzt wird. Tabelle 6-1 gibt einen Überblick über die Eckdaten des Graphs und die Farblegende nachfolgender Abbildungen.

Elementtyp	Anzahl	Farbe
Knoten insgesamt	756	
Kanten insgesamt	1262	
Portfolio Knoten	1	Schwarz
Produktlinien	8	Hellrot
Produktfamilien	9	Mittelrot
Produkte	13	Rot
Motorplattformen	7	Grau
Motoren	13	Dunkelblau
Module	186	Mittelblau
Teile	519	Hellblau

*Tabelle 6-1: Überblick über die Eckdaten des Hilti-Graphs und Farblegende nachfolgender Abbildungen*

Aus der „Vogelperspektive“ ist in Abbildung 6-6 zu erkennen, dass sich zwei Cluster ausbilden, die nur durch wenige Elemente verbunden sind. Verbindende Elemente sind hier der Portfolio-Knoten (schwarz) und Standardteile wie O-Ringe oder Passscheiben. Der Cluster oben rechts im Bild umfasst Geräte, die Motoren der Motorenplattformen UNI 1, DC 1 und DC 2 einsetzen (vgl. hervorgehobenen Ausschnitt). Der Cluster unten links beinhaltet die Motorenplattformen SR 1, SR 2, UNI 2 und UNI 3.

Dennoch ist eine solche Darstellung auf Grund der vielen Elemente und Verknüpfungen noch zu unübersichtlich, um die nötige Transparenz herzustellen. Im Folgenden wird der Graph als Datenbasis genutzt und für die unterschiedlichen, benötigten Ansichten auf wesentliche Elemente und Zusammenhänge reduziert.



„C-Combi / Rotary“ sollen Profile (Leistung, Qualität, Kosten etc.) für Nachfolgergeräte innerhalb der Produktlinie erarbeitet werden. Hierfür ist eine Prüfung bzw. Definition der Technologien und konkreten Motortypen für die neuen Gerätevarianten (z. B. Standard- und Premium-Variante) nötig.

(3) In welchen Geräten sind bestimmte Motorkomponenten verbaut?

Exemplarischer Anwendungsfall in der Praxis: Bei Abkündigung einer Komponente oder auch bei Qualitätsproblemen an einer Komponente wird eine Prüfung erforderlich, welche Geräte betroffen sind.

Für die Beantwortung der **ersten zwei Fragen** werden bereits ausschnittsweise Tapeten manuell gepflegt (vgl. Abbildung 6-7).

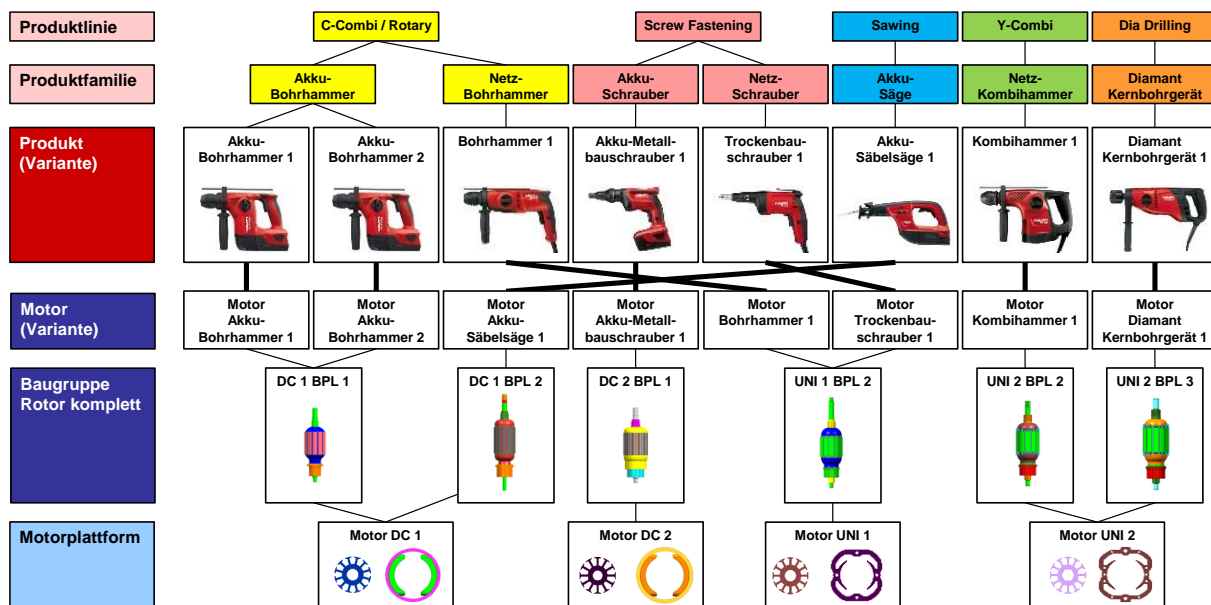


Abbildung 6-7: Beispiel einer "Tapete" (Ausschnitt) auf Basis des Beispieldatensatzes dieser Fallstudie (Darstellung Fa. HILTI)

Um den konsolidierten Graph automatisch in eine entsprechende Form zu bringen, müssen drei Arbeitsschritte formalisiert werden. Zunächst müssen die Baugruppen „Rotor komplett“ jedes Motors (vgl. dazu auch das Beispiel in Abbildung 6-4, erste Zeile der dargestellten Tabelle) der jeweiligen Motorenplattform direkt zugeordnet werden. Außerdem wird das Attribut *rotorpaketlaenge*, das im Motor-Knoten gespeichert ist, dem Rotor-Knoten zugeordnet. Hierfür wurde die Regel in Abbildung 6-8 angewendet.

Auf der Linken Seite wird ein Muster gesucht, in dem eine Motorplattform *mp* auf einen Motor *motor* zeigt. Diese Kante bekommt hier einen Namen, damit sie anschließend wieder identifiziert und gelöscht werden kann. Der Motor enthält wiederum eine Baugruppe *rotor* mit dem Namen „Rotor komplett“ (dies trifft im Beispiel für jeden Motor zu). Auf der Rechten Seite der Regel wird die Motorplattform mit der Baugruppe Rotor verknüpft. Um die geforderten Informationen der Baugruppe Rotor zu ergänzen, wird dem Namens-Attribut noch der Name der jeweiligen Motorplattform vorangestellt. In der Beschreibung des Knotens werden „BPL“ (Blechpaketlänge) und der entsprechende Wert, der im Motorknoten gespeichert ist, abgelegt. Die Kante zwischen Motor und Motorplattform wird abschließend gelöscht.



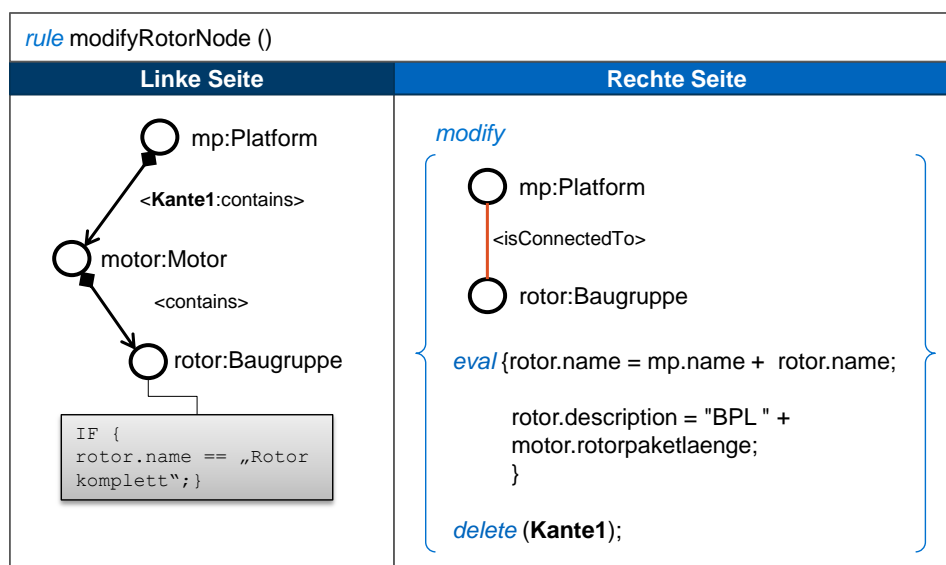


Abbildung 6-8: Regelbasierte Veränderungen am Knoten "Rotor komplett"

Im zweiten Schritt werden der Portfolio-Knoten sowie alle Teile- und Modul-Knoten aus dem Graphen gelöscht. Davon ausgenommen sind die Rotor-Knoten, die im Schritt zuvor bearbeitet wurden.

Auch wenn manche Motoren gleiche oder ähnliche Rotoren verwenden (z. B. UNI 1 in Bohrhammer 1 und 2 sowie im Trockenbauschrauber 1), haben diese unterschiedliche Materialnummern und sind daher jeweils eigene Knoten im Graph. Im dritten Schritt müssen daher diese Knoten zu einem Knoten zusammengefasst werden. Dies wird in folgender Regel formalisiert.

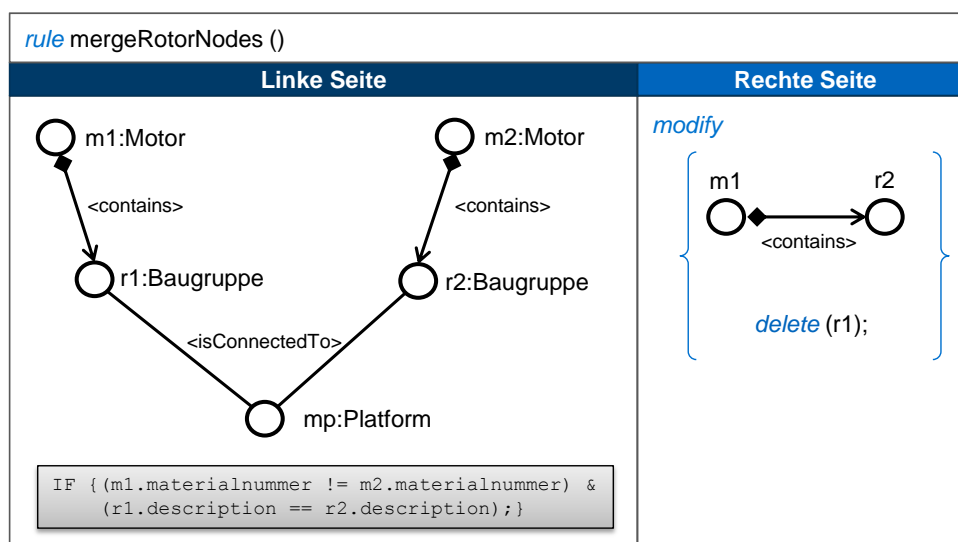


Abbildung 6-9: Regel zur Zusammenführung von Rotor-Knoten mit gleichen Rotorpaketlängen

Auf der Linken Seite wird nach einem Ähnlichkeitsmuster gesucht, in dem zwei unterschiedliche Motoren  $m1$  und  $m2$  jeweils zwei Rotoren  $r1$  und  $r2$  verwenden. Die Rotoren haben dieselbe Beschreibung und sind mit derselben Motorplattform verbunden. Auf der Rechten Seite wird der Motor  $m1$  mit dem Rotor  $r2$  verbunden und Rotor  $r1$  gelöscht.

Unter Verwendung eines hierarchischen Layouters wurde das Ergebnis, wie in Abbildung 6-10 und Abbildung 6-11 dargestellt, erzeugt.

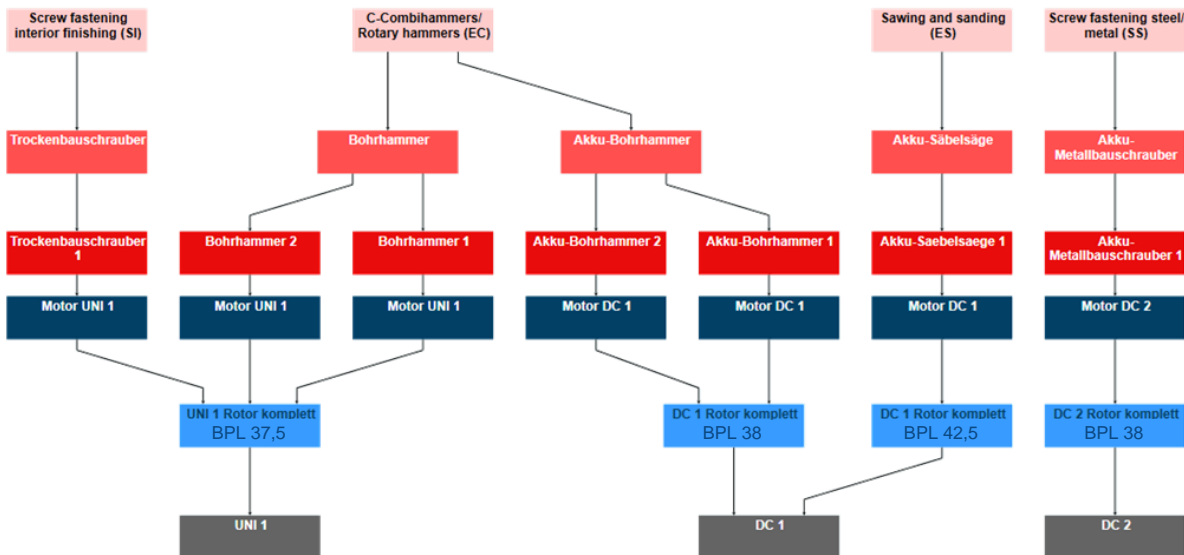


Abbildung 6-10: Automatisierte Generierung der Hilti-Tapete (Teil 1, Legende in Tabelle 6-2)

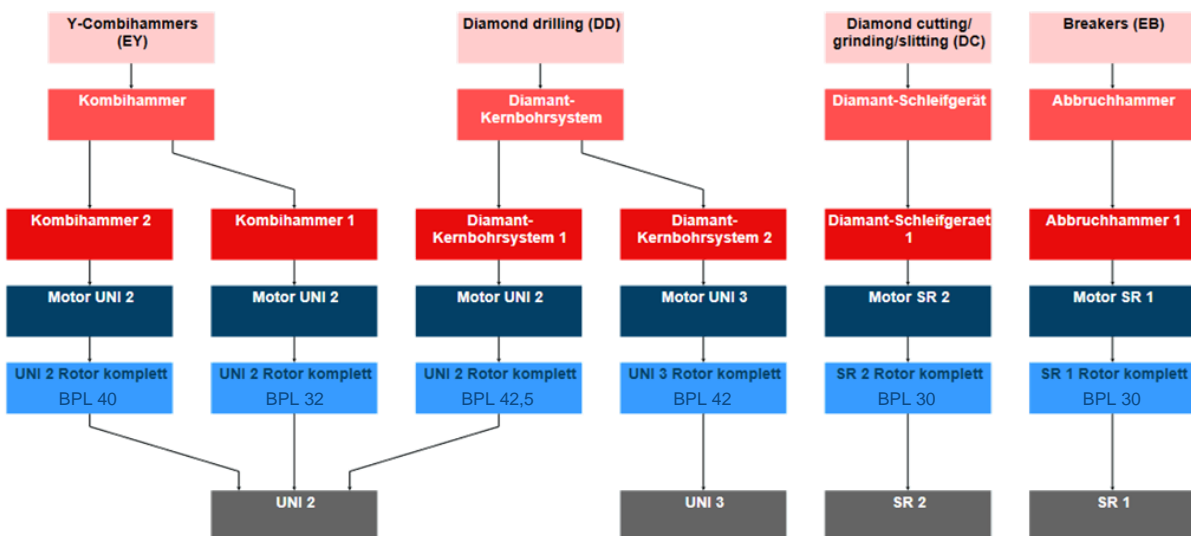


Abbildung 6-11: Automatisierte Generierung der Hilti-Tapete (Teil 2, Legende in Tabelle 6-2)

Die beiden oben dargestellten Ergebnisbilder wurden in einem Ergebnis generiert und im Sinne der Darstellbarkeit geteilt. Bei der Bildteilung wurde die weiter oben erwähnte Clusterbildung in Abbildung 6-6 herangezogen. Wie in der manuell erstellten Tapete können die benötigten Zusammenhänge mit einer „Tapete auf Knopfdruck“ dargestellt werden. Auf die Darstellung der Geräte- und Motorenbilder wurde an dieser Stelle verzichtet. In einem weiteren Schritt könnten die Bilder aber den entsprechenden Knoten zugeordnet und dargestellt werden.

Somit können die ersten beiden Fragen analog wie mit der manuell erstellten Tapete beantwortet werden.

- Im Anwendungsfall in Frage (1) wären demnach Wechselwirkungen durch eine Änderung des Motors DC 1 in der Akku-Säbelsäge 1 auf die Akku-Bohrhämmer 1 und 2 zu prüfen.
- Für die Planung der Überarbeitung der Produktlinie „C-Combi / Rotary“ müssen die Motorplattformen und jeweilig enthaltenen Motoren UNI 1 und DC 1 berücksichtigt werden.

Die automatisch erzeugte Tapete ist allerdings nicht auf diesen Teilausschnitt des Portfolios von Hilti beschränkt. Das gesamte Portfolio kann ohne Erhöhung des Aufwandes mit den hier vorgestellten Regeln in Tapeten-Form überführt und verarbeitet werden.

Die **dritte** der oben aufgeführten **Fragen** war, in welchen Geräten bestimmte Motorkomponenten verbaut sind. Als exemplarische Anwendungsfälle wurden die Abkündigung einer Komponente oder auch die Prüfung einer Komponente bei Qualitätsproblemen angeführt und die Frage, welche Geräte davon betroffen sind. Mit der manuellen Version der Tapete lässt sich diese Frage nur beantworten, wenn eine entsprechende Detaillierung dargestellt ist. Ansonsten müssen die benötigten Informationen zusammengetragen und in entsprechender Form aufbereitet werden.

Anhand von drei ausgesuchten Komponenten soll das regelbasierte Suchprinzip erläutert werden. Als Annahme seien beispielhaft die folgenden drei Komponenten von einer Änderung betroffen:

- Statorpaket DM 34 mit der Materialnummer 201227,
- Luftführung mit der Materialnummer 203848 und
- Rillenkugellager mit der Materialnummer 206210.

Anhand folgender Regelsequenz (vgl. Abbildung 6-12) sollen nun die betroffenen Geräte ermittelt werden. Im ersten Schritt wird die Regel *findSpecificItem* mit der Materialnummer des gesuchten Teils bzw. der gesuchten Baugruppe aufgerufen und in der lokalen Variablen *mnr* gespeichert. In der Regelsequenz erfolgt dieser Regelaufruf nacheinander jeweils für die Materialnummern der gesuchten Komponenten.

Auf der Linken Seite werden Knotenkettensuchen durchgeführt, die Teile oder Baugruppen mit dieser Materialnummer enthalten. Um beide Knotentypen finden zu können, wurde die Vererbungshierarchie im Basis-Metamodell (vgl. Kapitel 5.2.2) genutzt und als gesuchten Knotentyp *PortfolioElement* verwendet. Mögliche Zwischenbaustufen werden in Abbildung 6-12 vereinfacht grau dargestellt. Die Baustufen werden bei einer Tiefensuche mit entsprechenden Kettenmustern in der Regelprogrammierung berücksichtigt. Auf der Rechten Seite werden die Attribute *merker* aller Knoten der gefundenen Ketten auf „betroffen“ gesetzt. Im anschließenden Schritt werden alle nicht „betroffenen“ Knoten gelöscht.

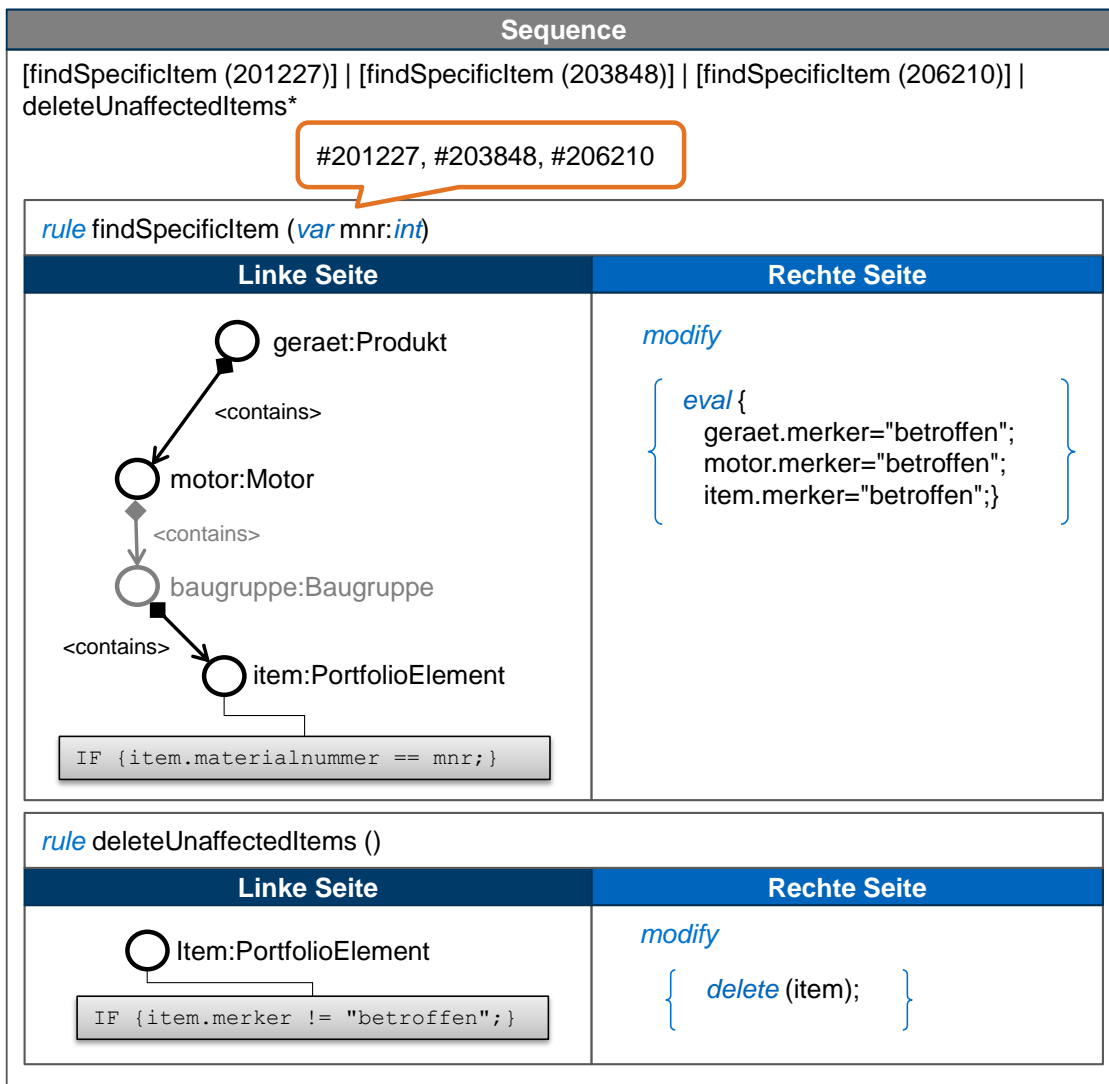


Abbildung 6-12: Regelsequenz zur Ermittlung betroffener Elemente bei Änderung einer Komponente

Bei Anwendung der Regelsequenz auf den Basis-Graph wird folgendes Ergebnis ausgegeben (vgl. Abbildung 6-13):

- Das Statorpaket DM 34 mit der Materialnummer 201227 wird auf der zweiten Baustufe verbaut in den Statoren 327690, 354655 und 326764 gefunden. Von einer Änderung sind die Geräte Bohrhammer 1 und 2 sowie der Trockenbauschrauber 1 betroffen.
- Die Änderung der Luftführung mit der Materialnummer 203848 betrifft die Geräte Akku-Bohrhammer 1 und 2.
- Von der Änderung des Rillenkugellagers mit der Materialnummer 206210 sind die Geräte Diamant-Kernbohrsystem 1 und 2 sowie der Kombihammer 1 betroffen.

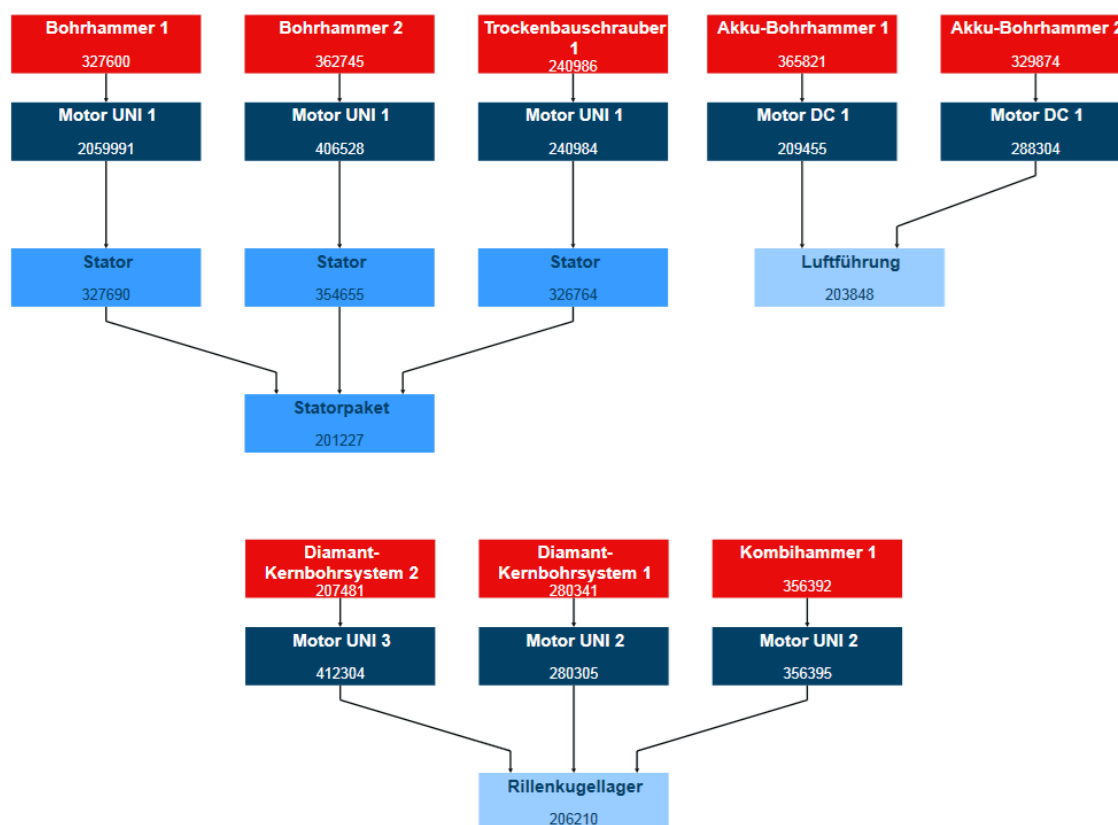


Abbildung 6-13: Ergebnis der Suche nach betroffenen Produkten (Legende in Tabelle 6-2)

Die verwendeten Regeln ermöglichen eine flexible Anpassung der Suche. Wenn nicht nach der Materialnummer, sondern beispielsweise nach bestimmten Namen, Geometriemaßen oder Wertebereichen von Attributen gesucht werden soll, kann die Regelsequenz aufwandsarm abgeändert werden.

#### 6.1.4 Diskussion der Fallstudie 1

In der Fallstudie bei Hilti konnten wesentliche Aspekte der Methodik zur Mustererkennung in komplexen Produktportfolios evaluiert werden. Neben der Datenkonsolidierung konnten die Mustersuche und die regelbasierte Strukturierung erfolgreich eingesetzt werden, um die Ziele der Fallstudie zu erreichen.

Es konnte gezeigt werden, dass verteilte Produktinformationen in einem Graphen zusammengeführt werden können. Fehlerhaft typisierte Elemente konnten durch logische Mustersuche korrekt retypisiert werden. Zudem konnte die Klassifizierung von Elementen des Graphs anhand des Beispiels Statordurchmesser demonstriert werden. Somit wurde ein konsolidierter Basis-Graph erzeugt, der für weitere Analysen herangezogen wurde.

Ziel der Fallstudie war es, eine Effizienzsteigerung der Informationsaufbereitung und -Analyse zu erreichen. Zentrales Arbeitsmittel ist bei Hilti die „Tapete“. Die Erstellung der Tapeten kann durch Wiederverwendung der hier erzeugten Regeln auf Knopfdruck geschehen. Dadurch ergibt sich eine drastische Effizienzsteigerung gegenüber der manuellen Erstellung in einem Graphikprogramm. Im Rahmen des hier verwendeten Beispiels von 13 Produkten ist die

Reduzierung des Arbeitsaufwandes für den Systemarchitekten noch nicht signifikant groß. Bei Anwendung derselben Regeln auf das Gesamtportfolio bei Hilti ergeben sich allerdings umfangreiche Zeiteinsparungen gegenüber einer manuellen Datenpflege und -analyse. Eine Effektivitätssteigerung konnte in der Fallstudie durch die gezielte Suche von betroffenen Geräten bei Änderung einer Komponente erzielt werden. Hier konnte durch Verwendung der Methodik gezeigt werden, dass sich planungsrelevante Informationen aufwandsarm verdichten und übersichtlich darstellen lassen. Zudem können die entsprechenden Regeln flexibel auf die Randbedingungen der Suche angepasst werden.

Dennoch müssen Einschränkungen bei dieser prototypischen Umsetzung angeführt werden. Der tatsächliche Mehrwert der Methodik kann erst bei Betrachtung des gesamten Geräteportfolios festgestellt werden. Zudem kann unter Einbeziehung weiterer Attribute wie Umsatz, Kosten, Stückzahlen und Ähnlichen ein noch differenzierteres Bild in den Analysen erstellt werden. Hierfür müssen die Aufbereitung und Visualisierung der Tapete noch weiter optimiert und an die Unternehmensprozesse angeglichen werden.

Dennoch konnten durch Einsatz der Methodik eine Effizienzsteigerung der Informationsaufbereitung und -Analyse gezeigt, Klassifizierung und Visualisierung unterstützt und die Struktur des Ist-Portfolios klarer dargestellt werden. Mit dem Ziel, typische Abläufe in der Arbeit der Antriebsentwicklung automatisiert unterstützen zu können, könnte das Regelwerk weiterentwickelt werden. Durch einen breiteren Einsatz muss dann in einem weiteren Schritt gezeigt werden, dass die Methodik die Kommunikation im Unternehmen verbessert und die Entscheidungen in der Planung und Steuerung der Portfolioentwicklung effektiver und effizienter vorbereitet.

## 6.2 Fallstudie 2

### 6.2.1 Ausgangssituation und Problemstellung

Die Elektronik GmbH<sup>32</sup> ist ein **mittelständischer Hersteller von Elektromotoren und Ansteuerungsgeräten**. Das Unternehmen besteht seit über 60 Jahren und vertreibt international seine Produkte. Das Produktportfolio umfasst Steuerungssoftware, Leistungsverstärker, Steuerungen, Motoren und mechanische Komponenten<sup>33</sup>. Die Produkte werden vorwiegend auf Kundenwunsch angefertigt oder angepasst. Entsprechend muss das Unternehmen mit einem **hohen Grad an Individualisierung** auf die Wünsche seiner Kunden eingehen.

Dies führte zu einem historisch **breit gewachsenen Produktportfolio** mit Einzelprodukten, die teilweise nur geringe Stückzahlen aufweisen. Mit der Anzahl der Produkte ist auch die Menge an unterschiedlichen Bauteilen gewachsen, die Kosten verursachen. Die Komponenten werden im Lagersystem gepflegt und bevorratet; sie verursachen Kosten in den Einkaufs- und

---

<sup>32</sup> Aus Gründen der Geheimhaltung wird der Name verfremdet.

<sup>33</sup> Unternehmenswebsite (entnommen am 31.10.13)

Vertriebsprozessen; und, solange sie bestellbar sind, müssen auch die benötigten Werkzeuge und Fertigungstechnologien vorgehalten werden (ELEKTRONIK GMBH 2013<sup>34</sup>).

Auf Grund der stark angewachsenen, teils unnötigen Komplexität des Produktportfolios ist es nötig, **Vielfalt im Produktportfolio systematisch abzubauen** und unwirtschaftliche Produkte abzukündigen. Das Unternehmen hat daher eine Liste mit möglichen Abkündigungskandidaten zusammengestellt. Jedoch fehlt Transparenz darüber, ob diese Liste erweitert werden muss und welche Abhängigkeiten zwischen den abzukündigenden und den verbleibenden Produkten berücksichtigt werden müssen. In zahlreichen Produkten sind Wiederholteile verbaut. Die nötigen Informationen darüber sind in umfangreichen Datenbankexporten enthalten, die produktweise ausgewertet werden müssen.

Die Analyse des Produktportfolios hinsichtlich abzukündigender Produkte ist mit einem **erheblichen Aufwand** verbunden. Allein mit konventionellen Mitteln wie Datenbankabfragen und Datenaufbereitung in Tabellenkalkulationsprogrammen ist diese Aufgabe nicht lösbar. Das Produktportfolio umfasst 6.784 Einzelprodukte, die mit 26.246 Komponenten aufgebaut sind. Relationen zwischen Komponenten sind in Datentupeln gespeichert, in denen jeweils zwei Materialnummern miteinander verknüpft sind. Daraus können Stücklisten generiert und abgeglichen werden. Angesichts der großen Anzahl von Produkten ist dies aber nicht praktikabel, um Wechselwirkungen von Abkündigungskandidaten mit verbleibenden Produkten zu ermitteln.

### 6.2.2 Zielstellung des Methodeneinsatzes und Vorgehen

Es soll eine **schrittweise Aufbereitung der Problemstellung** „vom Groben zum Detail“ – also von einer Gesamtportfolioebene hin zu Einzelprodukten – durchgeführt werden. Auf Portfolioebene soll festgestellt werden, ob Produktfamilien unwirtschaftlich sind und somit effizient als Ganzes abgekündigt werden können. Auf Einzelproduktebene sollen dann Diskussionen im Detail unterstützt werden, welche Produkte besonders kritisch sind, welche Reihenfolge der Abkündigungen sinnvoll ist, welche Wechselwirkungen mit anderen Produkten beachtet werden müssen und dergleichen. Um eine Abkündigungsentscheidung vorzubereiten, soll im Rahmen dieser Fallstudie<sup>35</sup> festgestellt werden,

- in welcher Produktlebensphase sich entsprechende Produkte befinden,
- welche Produkte keinen oder zu niedrige Umsätze bzw. Deckungsbeiträge in den letzten drei Jahren erwirtschaftet haben und
- welche Baugruppen und Teile von Abkündigungskandidaten ebenfalls in anderen Produkten verwendet werden (Wiederholteile).

Für die Diskussion mit der Geschäftsführung müssen die **Ergebnisse der Analyse** anschaulich **visualisiert** werden, um eine rationale Entscheidungsfindung unterstützen zu können. Es wird daher nach einer Methodik gesucht, effektiver und effizienter zu einer Entscheidungsgrundlage für eine Abkündigung von Produkten zu kommen. Daher soll die Methodik zur

---

<sup>34</sup> Interview mit dem Leiter Produktstrategie am 30.10.13

<sup>35</sup> Unter Berücksichtigung der zur Verfügung gestellten Daten

Mustererkennung in komplexen Produktportfolios eingesetzt werden, um den Abkündigungsprozess mit effektiven und effizienten Analysen zu unterstützen.

In dieser Fallstudie wird das Ziel verfolgt, die Abkündigungsentscheidungen mit graphenbasierten Analysen zu unterstützen. In enger Abstimmung mit dem Leiter für Produktstrategie und dem technischen Leiter des Unternehmens wurde diese Fallstudie durchgeführt. In fünf Workshops wurden die Ziele, das Vorgehen, die Datenbereitstellung und anschließend die Ergebnisse diskutiert. Bei der Analyse wurde ein stufenweises **Vorgehen** verfolgt:

1. Zusammenführung und Aufbereitung relevanter Daten
2. Festlegung und Formalisierung der Abkündigungskriterien in einem Regelwerk
3. Aufbereitung und Visualisierung der Ergebnisse

Die Schritte des Vorgehens und die jeweiligen Ergebnisse werden im nächsten Abschnitt dargestellt.

### 6.2.3 Ergebnisse

#### **Zusammenführung und Aufbereitung relevanter Daten**

Als erster Schritt müssen die relevanten **Daten zusammengeführt und aufbereitet** werden. In der Fallstudie steht eine umfangreiche Datenbasis über das gesamte Produktportfolio der Elektronik GmbH zur Verfügung. Aus Gründen der Geheimhaltung wurden Bezeichnungen und Materialnummern von Produkten und Teilen verfremdet. Über eine Zuordnungstabelle können die Ergebnisse später wieder den Originaldaten zugeordnet und geprüft werden.

Das Unternehmen pflegt in allen Unternehmensbereichen **umfangreiche Datenbanken**, die teilweise Eigenentwicklungen sind. Die nötigen Informationen für die Bearbeitung der Abkündigungsproblematik mussten aus verteilten Datenquellen beschafft werden. Der hier zusammengeführte Datensatz wurde durch die Industriepartner aus den folgenden Datenquellen zusammengestellt:

- ERP-System PSI-Penta (Stammdaten zu Produkten)
- Rückmeldetool für die Fertigung (Ausschuss, Gut-Teile etc.)
- Produktinformationsmanagement-Werkzeug (Verwaltung von Produktkategorien und Zusatzdaten)

Die Daten wurden mittels des angepassten Basis-Metamodells aus Kapitel 5.2.2 in einen Graphen überführt. Dabei wurden die Originalnamen und Materialnummern verfremdet. Als zusätzliche Attribute wurden neben denen des Basis-Metamodells die Folgenden im Metamodell definiert:

- status [Kennzahl zum Produktlebenszyklus zwischen 400 und 2700; Erläuterungen folgen weiter unten]
- gueltigkeit [„Gültig“ / „Ungültig“]
- beschaffungskenner [„Einkauf“ / „Eigenfertigung“]
- warengruppe [z. B. „Motoren“, „Handelsware“, „Elektronik“ etc.]
- verrechnungspreis [in €]



- durchschnittspreis [in €]
- listenpreis [in €]
- v\_Gesamt: Verbrauch Gesamt (Werte für die vergangenen drei Jahre) [in €]
- umsatz\_Gesamt: Umsatz Gesamt (Werte für die vergangenen drei Jahre) [in €]
- u\_Menge\_Gesamt: Stückzahl Gesamt (Werte für die vergangenen drei Jahre) [Anzahl]
- dp\_Gesamt: Deckungsbeitrag Gesamt (Werte für die vergangenen drei Jahre) [in €]
- counter: Hilfsattribut für Berechnungen [-]

Mit den Knoten, Attributen und Relationen wurde ein Basis-Graph erzeugt, der für weitere Schritte genutzt werden kann. Die besondere Herausforderung in dieser Fallstudie liegt im Datenumfang des Produktportfolios, der zu bewältigen ist. Der Graph umfasst:

- 295 Produktfamilien,
- 6.784 Produkte und
- 26.246 Teile.

Insgesamt sind 156.553 Relationen enthalten, von denen 6.784 Aggregationsbeziehungen zwischen Produktfamilien und Produkten und 149.769 Kompositionsbeziehungen zwischen Produkten und Komponenten sind.

Um die Abkündigungsentscheidungen sinnvoll zu unterstützen, wird der Graph in seiner Ursprungsform zunächst **konsolidiert**. Insgesamt wurden bei diesen Konsolidierungsschritten 3.240 Knoten aus dem Graph entfernt:

- Da sie für die Problemstellung keine Bedeutung haben, werden Produkte gelöscht, die **Handelsware** sind (sog. „*Singlets*“). Mit einer einfachen Regel werden daher auf der Linken Seite der entsprechenden Regel Knoten gesucht, die im Attribut *warengruppe* den Wert „Handelsware“ haben. Auf der Rechten Seite der Regel werden diese Knoten gelöscht. Im Zuge dieser Datenbereinigung wurden somit 3.017 Knoten aus dem Graph entfernt.
- Des Weiteren werden **isolierte Knoten** identifiziert, in einem Report ausgegeben und anschließend gelöscht. Es wurden 223 isolierte Knoten gefunden und gelöscht. Hierbei handelt es sich beispielsweise um Teile, die mit Handelsware verknüpft waren oder um falsche Datensätze.

Anschließend müssen die Komponenten im Graph noch **nachträglich typisiert** werden. Wie in der Hilti-Fallstudie werden Teile, die weitere Teile enthalten, als Baugruppen typisiert. Für diesen Schritt konnte die Regel aus der Hilti-Fallstudie und aus Beispiel (13) in Kapitel 5 wiederverwendet werden. Es wurden 9.345 Muster im Graphen gefunden, in denen Teile weitere Teile beinhalten. Die gefundenen Elemente wurden als Baugruppen retypisiert. Somit hat der Graph 9.345 Baugruppen und 16.901 Teile. Der so erzeugte, konsolidierte Graph kann nun für weitergehende Analysen verwendet werden. Für eine Visualisierung ist er allerdings noch zu groß.

### Festlegung und Formalisierung der Abkündigungskriterien

Mit dem Ziel unwirtschaftliche Produktfamilien und Produkte abzukündigen, werden **Abkündigungskriterien festgelegt**. Für die Abkündigung spielen aus Sicht der Elektronik GmbH drei Kriterien eine wichtige Rolle:

1. Produkte, die kurz- oder mittelfristig auf jeden Fall im Produktportfolio verbleiben sollen, wurden im Attribut *status* manuell gekennzeichnet (z. B. Produkte der Produktfamilie\_127). Ebenso wurden Produkte gekennzeichnet, die in jedem Fall abgekündigt werden sollen (z. B. Produkte der Produktfamilie\_289).
2. Produkte, die in den letzten drei Jahren einen Deckungsbeitrag<sup>36</sup> von weniger als 50.000 € und einen Umsatz von weniger als 100.000 € erwirtschaftet haben, sollen detaillierter analysiert werden.
3. Produkte, deren Produktstatus größer als „2000“ sind – also ein aktives Marketing bereits eingestellt worden ist (vgl. Tabelle 6-2) – sollen ebenfalls regelmäßig überprüft werden. Diese Produkte sind in ihrem Produktlebenszyklus so weit vorangeschritten, dass eine abklingende Nachfrage am Markt aus Erfahrung zu erwarten ist.

Eine Liste mit möglichen Produktstatus findet sich in nachfolgender Tabelle. Im Folgenden werden Produkte ohne Produktstatus im Graphen blau dargestellt:

Status	Produktstatus	Kommentar	Knotenfarbe
400	Lastenheft freigegeben	aktiv	grün
1200	NULL-Serie freigegeben von Planung	aktiv	grün
1800	Serie freigegeben für Vertrieb	aktiv	grün
2000	Marketing eingestellt	aktiv	grün
2100	Abkündigung einleiten?	zu prüfen	gelb
2200	Abkündigung beschlossen	abkündigen	rot
2300	Abkündigung begonnen	abkündigen	rot
2500	Abkündigung beendet	erledigt	schwarz
2700	Produktlebenszyklus beendet	erledigt	Schwarz

*Tabelle 6-2: Mögliche Status im Produktlebenszyklus bei der Elektronik GmbH*

Um Produkte möglichst effizient abzukündigen, möchte die Elektronik GmbH vorzugweise Produktfamilien im Ganzen abkündigen, wenn es sinnvoll ist. Sowohl für die Abkündigung von Produktfamilien, als auch von einzelnen Produkten sind Gleichteile mit Bestandsprodukten zu identifizieren. Es ist abzuklären, inwiefern sich Abkündigungen durch diese gegebenen Abhängigkeiten auf die Bestandsprodukte auswirken.

---

<sup>36</sup> Der Deckungsbeitrag eines Produktes errechnet sich aus dem Umsatz bzw. dem Erlös gemindert um die variablen Kosten. Somit handelt es sich um den Betrag, der zur Deckung der fixen Kosten eingesetzt werden kann. (Quelle: <http://wirtschaftslexikon.gabler.de/Definition/deckungsbeitragsrechnung.html>; entnommen am 27.12.2013)

### **Analysen und Visualisierung zur Unterstützung der Abkündigungsentscheidung**

Um sich einen ersten Überblick verschaffen zu können, muss der Graph stark reduziert werden, um graphisch dargestellt werden zu können. Es sollen daher nur alle Produktfamilien und zugeordneten Produkte dargestellt werden. Baugruppen und Teile werden für diese Ansicht gelöscht. Produkte werden gemäß dem Status ihres Produktlebenszyklus eingefärbt (Produkte ohne expliziten Status werden blau dargestellt). Um einen Eindruck über die Umsatzverteilung über das Produktportfolio hinweg zu bekommen, wird die Größe der Produktknoten proportional zu den aufsummierten Umsatzzahlen der letzten drei Jahre skaliert. Die sternförmige Darstellung wurde mit einem Layouter erzeugt. Die folgende Regelsequenz (vgl. Abbildung 6-14) wurde auf den konsolidierten Graphen angewendet.

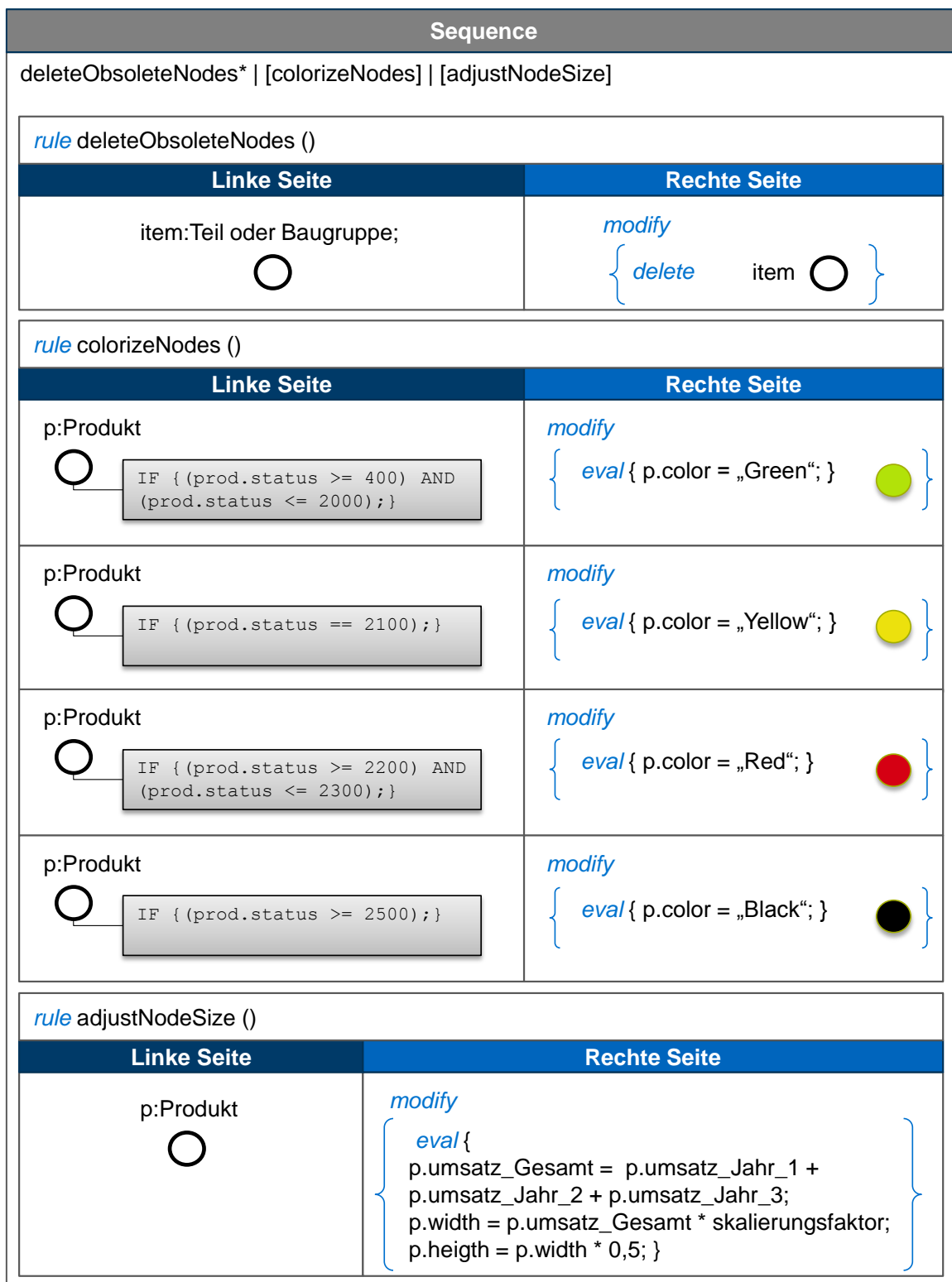


Abbildung 6-14: Regelsequenz zur Generierung einer Übersicht über das Produktportfolio der Elektronik GmbH

In Abbildung 6-15 ist das Ergebnis dargestellt. Auffällig bei einer ersten Übersicht ist, dass mit nur wenigen Produkten große Umsätze erzielt wurden. Eine sehr große Menge an Produkten machen augenscheinlich nur sehr geringe Umsätze.

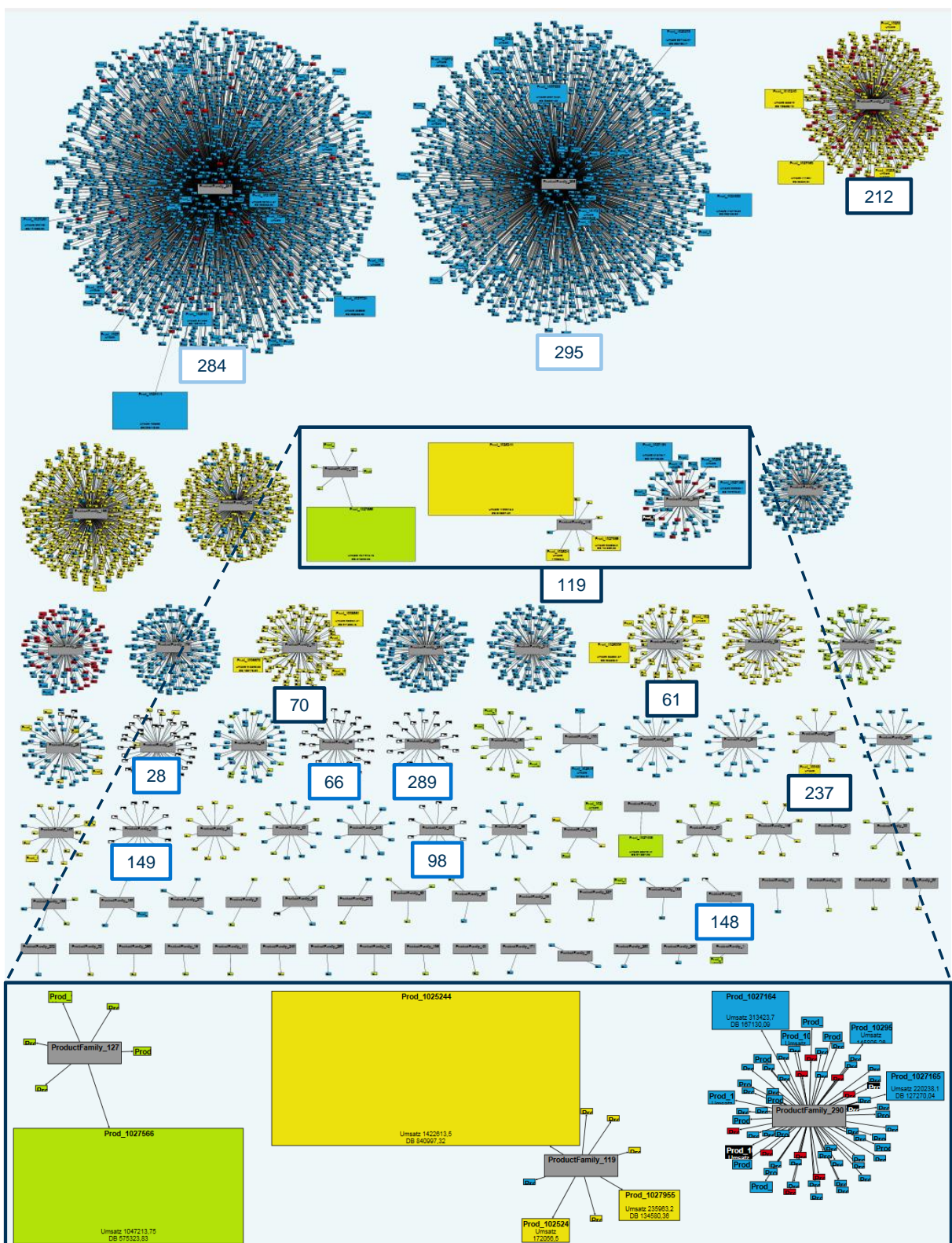


Abbildung 6-15: Übersicht über das Produktportfolio der Elektronik GmbH (Erläuterungen zur Farbgebung in Tabelle 6-2, die Größe der Knoten ist proportional zum Umsatz der letzten drei Jahre)

Die Darstellung zeigt zahlreiche interessante Zusammenhänge auf. Anhand dieser Übersicht können erste Handlungsfelder priorisiert und Strategien entwickelt werden. Beispielhaft sollen hier einige Punkte hervorgehoben werden:

- Die zwei Produktfamilien mit den meisten Produkten (295 und 284) beinhalten überwiegend blaue Produktknoten. Viele sehr kleine Knoten zeugen von kleinen bzw. keinen Umsätzen, während nur wenige Produkte größere Umsätze machen. Innerhalb dieser Produktfamilien wäre zu prüfen, ob es sich beispielsweise bei den kleinen Knoten um unnötige Varianten der größeren Knoten handelt.
- Die Produktfamilien 212, 119, 70, 61 und 237 (von links nach rechts) beinhalten sehr große oder zumindest größere Knoten, die für Produkte mit großen Umsätzen stehen. Die gelbe Einfärbung bedeutet, dass eine Abkündigung dieser Produkte zu prüfen ist. Angesichts der hohen Umsätze in den letzten drei Jahren wären Produkte mit kleineren Umsätzen hier in der Abkündigungsfolge vorzuziehen.
- Einige Produktfamilien (v. l. n. r: 28, 66, 289, 149, 98 und 148) beinhalten überwiegend oder ausschließlich schwarze Produktknoten. In diesen Produktfamilien ist die Abkündigung abgeschlossen und die Produkte können aus dem Produktportfolio entfernt werden.

Die Abkündigungskandidaten sollen in einer weiteren Analyse priorisiert werden. Hierzu sollen folgende Kriterien berücksichtigt werden:

- Der Status im Produktlebenszyklus: Produkte, die den Status kleiner, gleich 2000 haben und grün gekennzeichnet sind, sollen von der Analyse ausgenommen werden. Unter diesen Produkten wurden einige erst vor Kurzem eingeführt und konnten noch keine signifikanten Umsätze erzielen.
- Zusätzlich sollen Produkte ausgenommen werden, die einen Umsatz (aufsummiert über die letzten drei Jahre) von mehr als 100.000 € erzielten.

Zunächst wurden die grünen Produktknoten gelöscht. Anschließend wurden die umsatzstarken Produkte gelöscht (vgl. Abbildung 6-16). Gerade bei den gelb eingefärbten Knoten wäre an dieser Stelle zu prüfen, inwieweit eine Abkündigung zum jetzigen Zeitpunkt sinnvoll ist oder aufgeschoben werden sollte. Für eine Entscheidungsunterstützung der Geschäftsleitung können diese zehn Produkte fokussiert diskutiert werden

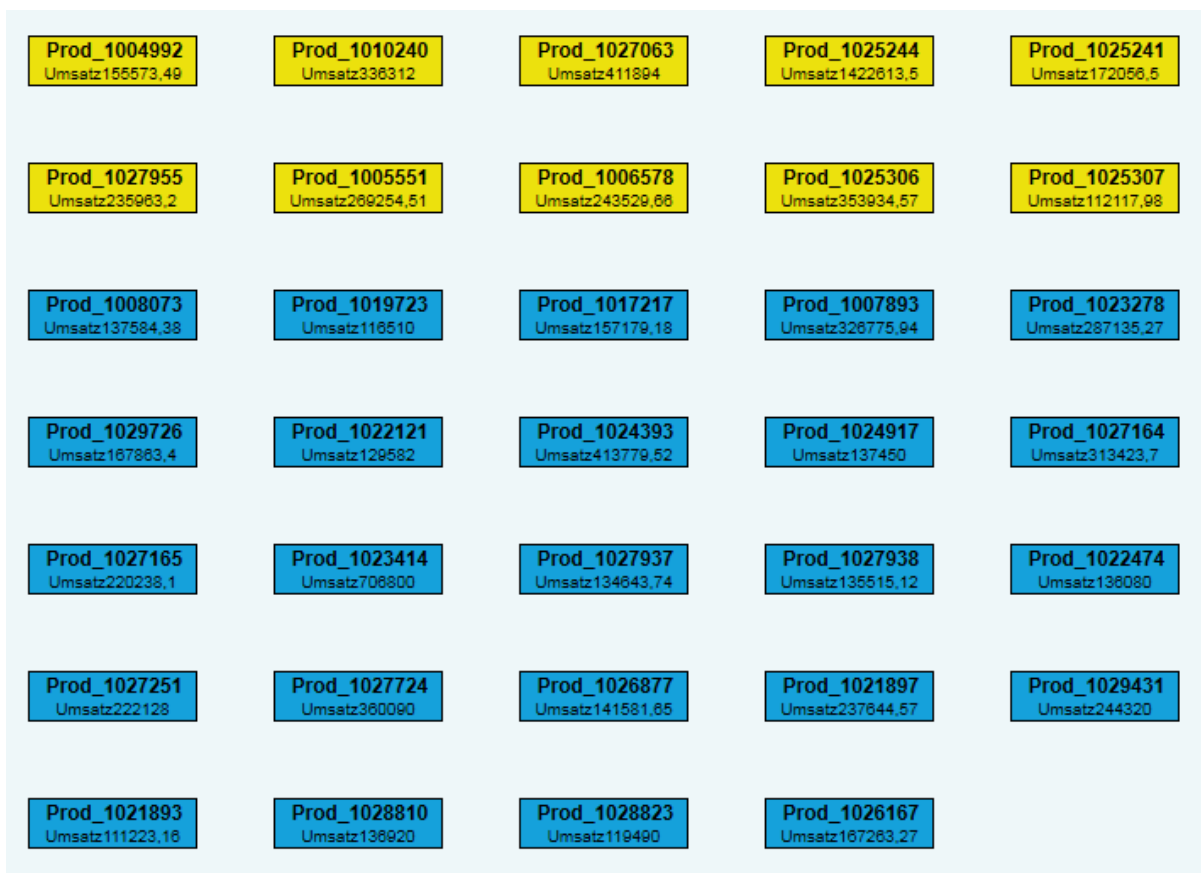


Abbildung 6-16: Umsatzstarke Produkte, die von der weiteren Analyse ausgenommen wurden

Die übrigen Produktknoten mit mäßigen bis schwachen Umsätzen werden gemäß ihrer Umsätze (der letzten drei Jahre) auf der x-Achse und gemäß ihrer Deckungsbeiträge (der letzten drei Jahre) auf der y-Achse eines Koordinatensystems aufgetragen. Hierfür wurden die jeweiligen aufsummierten Umsatz- und Deckungsbeitragswerte den x- und y-Koordinaten der Produktknoten regelbasiert zugeordnet. Nachfolgende Abbildung zeigt das Ergebnis.

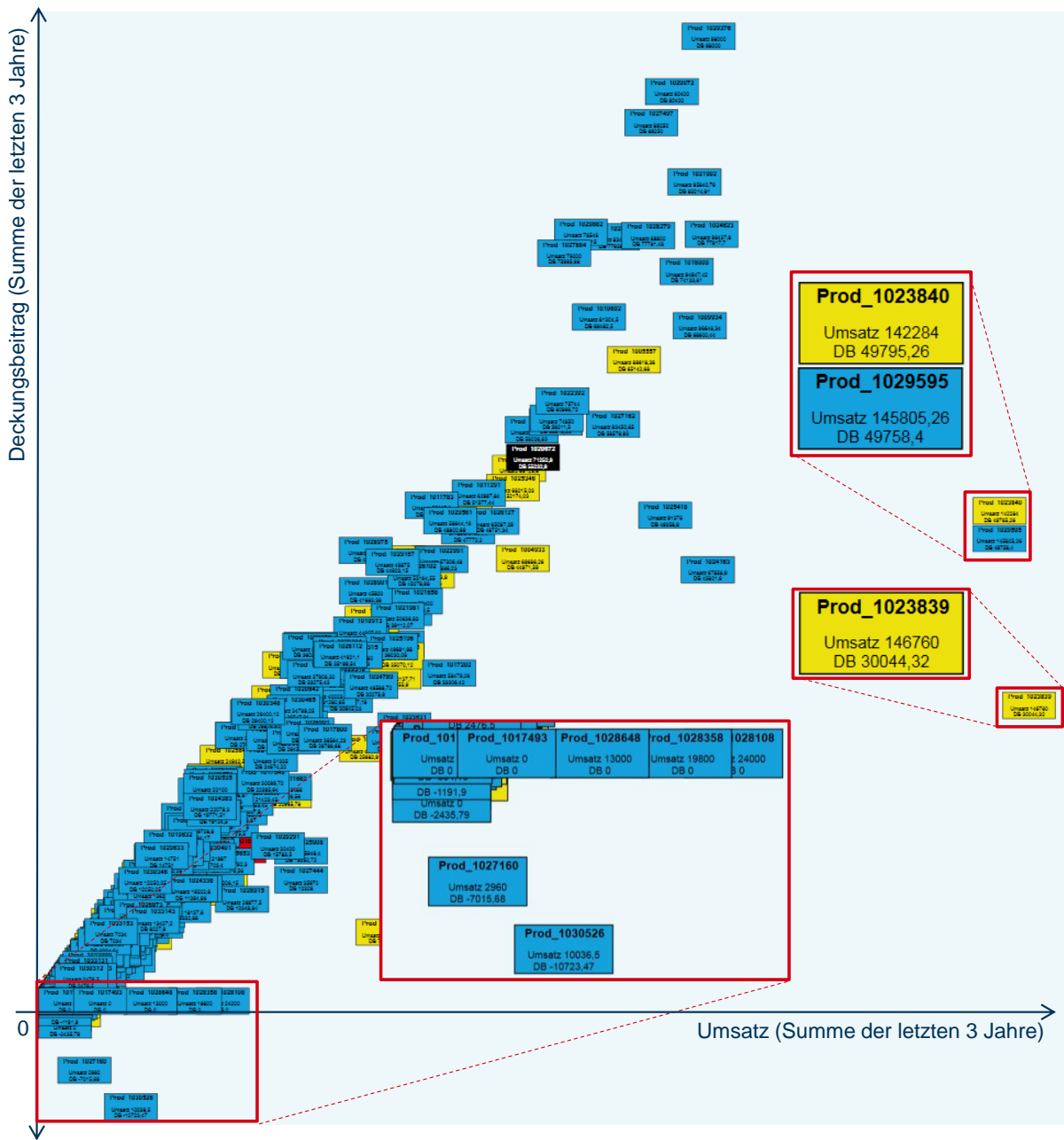


Abbildung 6-17: Produkte der Elektronik GmbH sortiert nach Umsatz und Deckungsbeitrag der letzten drei Jahre

Es ist auffällig, dass der überwiegende Anteil der Produkte keinen oder nur einen sehr kleinen Umsatz bzw. Deckungsbeitrag erwirtschaftet (vgl. untere linke Ecke in Abbildung 6-17). Manche Produkte erzielen sogar einen negativen Deckungsbeitrag. Es handelt sich dabei auch um zahlreiche blau gefärbte Produkte, also Produkte, deren Produktstatus noch nicht gepflegt wurde. Für diese Produkte sollte priorisiert ein Abkündigungsprozess eingeleitet werden.

Zudem lässt sich erkennen, dass sich die meisten Produkte in dieser Darstellung an einer Geraden ausrichten. Bei diesen Produkten sind Umsatz und Deckungsbeitrag proportional zueinander. Drei Produkte – Prod\_1023840, Prod\_1029595 und Prod\_1023239 – weichen



davon deutlich ab. Sie weisen einen verhältnismäßig unterdurchschnittlichen Deckungsbeitrag auf. Hier wäre zu klären, ob die Kostenstruktur dieser Produkte verbessert werden kann.

Ein weiterer Aspekt, der berücksichtigt werden soll, ist eine effiziente Abkündigung von Produktfamilien möglichst im Ganzen. Hierfür wird eine Visualisierung gewünscht, aus der die **Verflechtung der Abkündigungskandidaten mit anderen Produktfamilien** hervorgeht. In der Abkündigungsentscheidung möchte die Elektronik GmbH diese Abhängigkeiten berücksichtigen. Daher wurde der konsolidierte Graph als Ausgangsbasis herangezogen und hinsichtlich produktfamilien-übergreifender Gleichteile ausgewertet.

Im ersten Schritt der Analyse wurden alle Teile entfernt, die ausschließlich in einer Produktfamilie vorkommen. Folgende Regelsequenz beschreibt diese Operation:

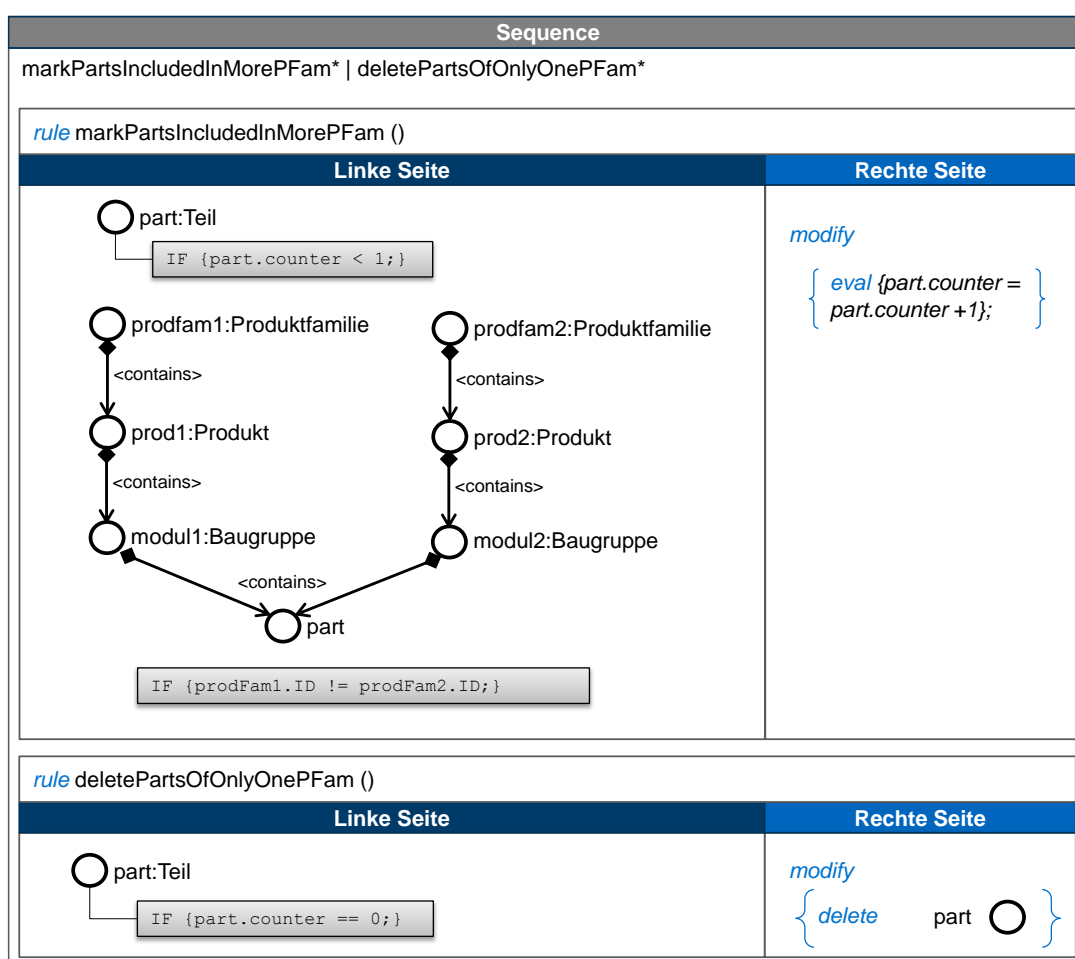


Abbildung 6-18: Reduzierung der Module und Teile auf produktfamilien-übergreifende Gleichteile

Eine einfache Struktursuche nach Teilen oder Modulen, auf die mehrere *contains*-Kanten zeigen, ist in diesem Fall nicht ausreichend. Hier müssen mehrere Fälle berücksichtigt werden, nämlich Gleichteile, die

- (1) nur innerhalb einer Produktfamilie,
- (2) über zwei oder mehrere Produktfamilien hinweg oder auch
- (3) innerhalb einer Produktfamilie *und* über mehrere Produktfamilien hinweg

vorkommen können. Lediglich Fall (1) und Nicht-Gleichteile sollen gelöscht werden. Daher wird eine Regelsequenz in zwei Stufen formuliert. Es wird ein Hilfsattribut *counter* verwendet, um den logischen Ablauf zu unterstützen. Auf der Linken Seite der Regel wird nach einem Knoten *part* vom Typ Teil gesucht. Solange das Attribut *counter* Null ist, wird nach einem Muster gesucht, in dem das Teil in zwei unterschiedlichen Produktfamilien – also mit verschiedenen IDs – enthalten ist. Wenn ein solches Muster gefunden wurde, wird das Hilfsattribut auf 1 gesetzt. Somit werden auch Teile auf 1 gesetzt, wenn sie öfters innerhalb einer Produktfamilie und mindestens einmal in einer weiteren Produktfamilie eingesetzt werden. Teile mit Wiederverwendung innerhalb nur einer Produktfamilie bleiben auf Null gesetzt. Diese Logik wird im nächsten Schritt dazu genutzt, alle Teile mit Hilfsattribut *counter* gleich Null zu löschen.

Im nächsten Schritt werden die verbliebenen Teile mit den Produktfamilien verknüpft. Alle übrigen Knoten (Module und Produkte) werden gelöscht, um die Übersichtlichkeit bei der Visualisierung zu gewährleisten. Nachfolgende Regelsequenz beschreibt die zweite Operation:

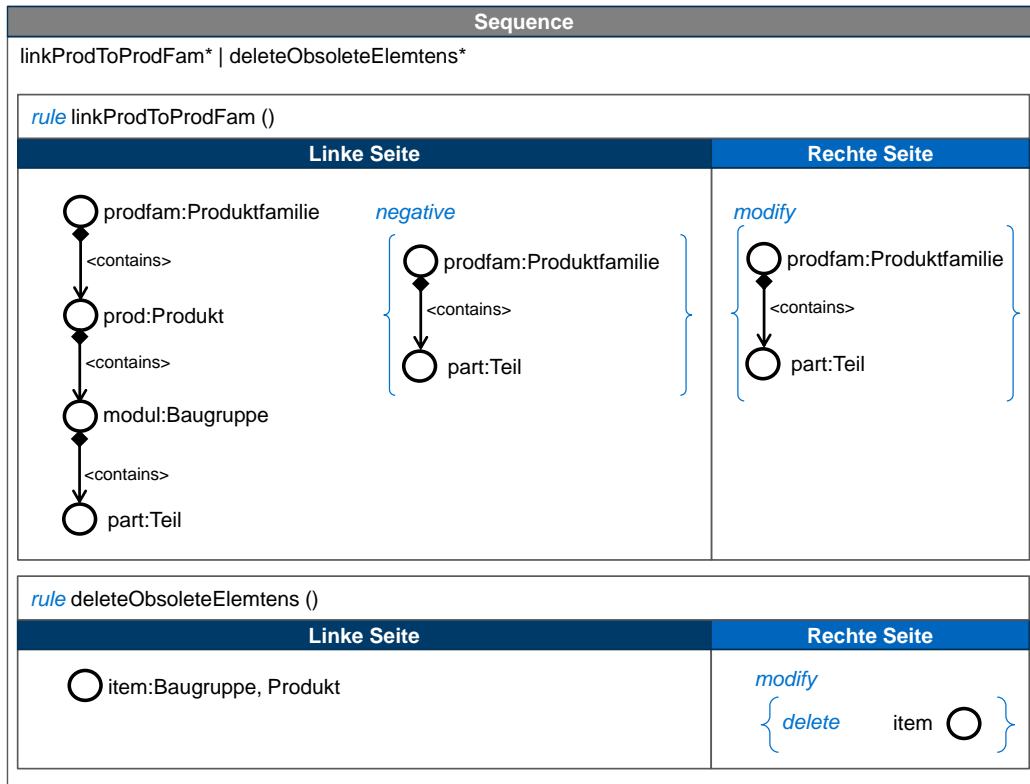


Abbildung 6-19: Reduzierung des Graphs um Produkte und Module

Auf der Linken Seite wird nach einer Knotenkette zwischen Produktfamilien und Teilen gesucht. Solange noch keine neue contains-Kante zwischen Produktfamilie *prodfam* und Teil *part* gesetzt wurde, wird eben jene auf der Rechten Seite generiert. Im zweiten Schritt werden alle Baugruppen und Produkte gelöscht, so dass nur noch Produktfamilien und jene Teile im Graph sind, die produktfamilien-übergreifend eingesetzt werden. Nachfolgende Abbildung 6-20 zeigt den resultierenden Graphen. Weiter oben wurden Produktfamilien identifiziert, die abgekündigt werden sollen (vgl. Abbildung 6-15). Diese wurden rot hervorgehoben.

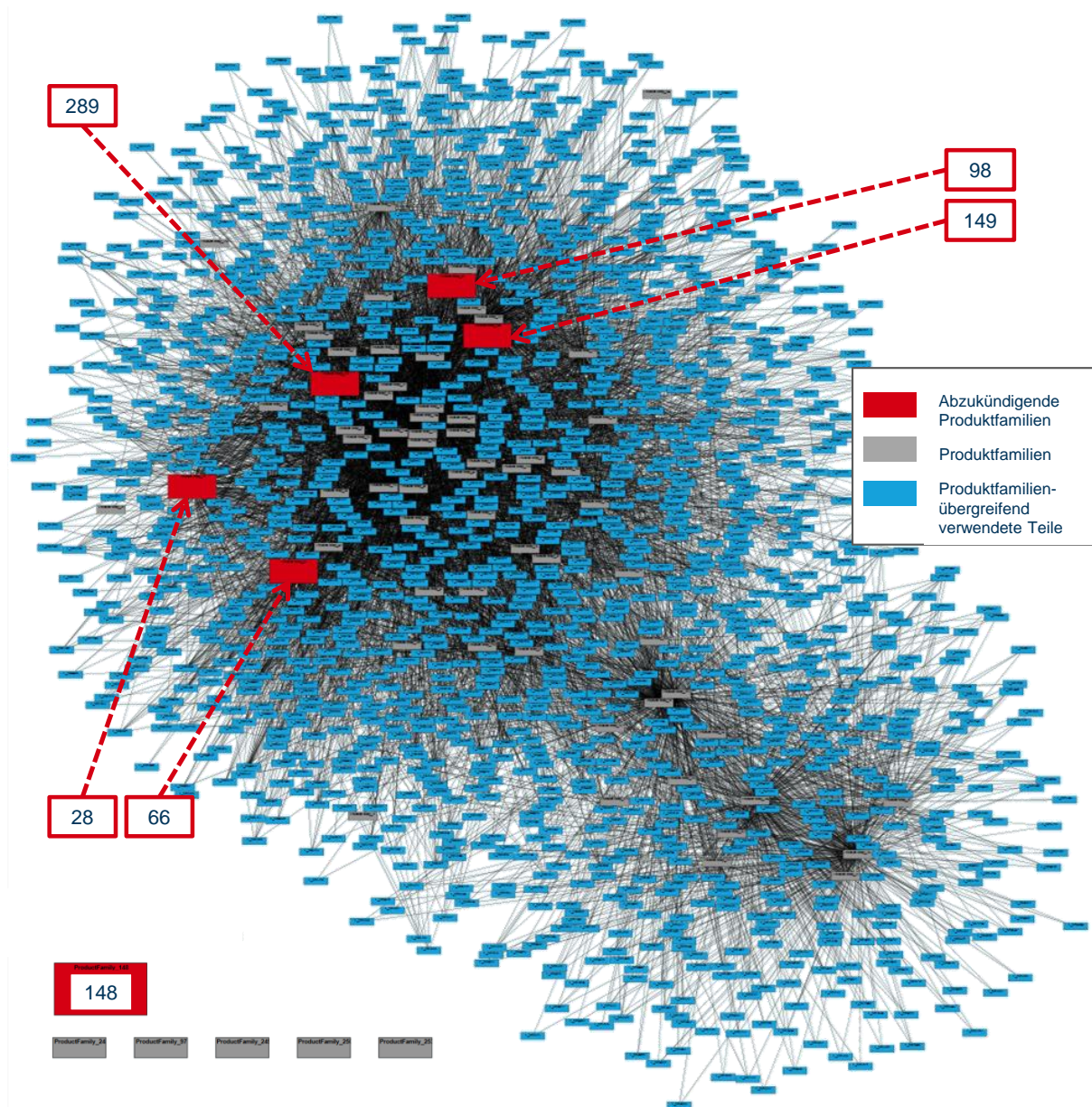


Abbildung 6-20: Produktfamilien und produktfamilien-übergreifend eingesetzte Teile

Die Produktfamilien sind eng miteinander verknüpft. Lediglich sechs Produktfamilien teilen keine Teile mit anderen (nämlich Produktfamilien 24, 97, 148, 249, 250 und 253). Im Sinne der Wiederverwendung ist dieses Bild positiv zu bewerten. Insgesamt werden 1.461 Teile produktfamilien-übergreifend verwendet. Im Sinne einer Abkündigung ergeben sich allerdings zahlreiche mögliche Wechselwirkungen mit anderen Produktfamilien, die geprüft werden müssen, um eine Entscheidung vorbereiten zu können. Anhand dieses Graphs können nun Detaildiskussionen geführt werden. Hierfür können jeweils die diskutierten Produktfamilien hervorgehoben und nicht verknüpfte Elemente ausgeblendet werden.

## 6.2.4 Diskussion der Fallstudie 2

In der Fallstudie bei der Elektronik GmbH wurde das Ziel verfolgt, die Abkündigungsentscheidungen mit graphenbasierten Analysen zu unterstützen. Die gestellte Aufgabe war mit konventionellen Mitteln nicht durchführbar. Daher wurde nach alternativen Methoden gesucht, die Problemstellung zu lösen. Angesichts des großen Datenvolumens eignet sich Methodik der Mustererkennung in komplexen Produktportfolios dazu, die aufwendigen Analysen effektiv und effizient durchzuführen. In der Fallstudie wurden die Methoden zur

- Zusammenführung relevanter Daten in graphenbasierter Form,
- Formalisierung repetitiver Aufgaben im Produktportfoliomanagement,
- automatischen Muster- und Fehlererkennung und
- intuitiv verständlichen Visualisierung relevanter Zusammenhänge

erfolgreich eingesetzt und evaluiert.

Die Elektronik GmbH pflegt zahlreiche Datenbanksysteme. Die Daten mussten also nicht nachträglich aufwendig erhoben, sondern konnten direkt in einen Graphen zusammengeführt werden. Bei der Konsolidierung der Datenbasis hat sich gezeigt, dass eine regelbasierte Transformation des Graphen es erlaubte, effizient isolierte Knoten und Knoten mit definierten Eigenschaften (hier „Handelsware“) vor der Analyse heraus zu löschen und falsch typisierte Knoten entsprechend ihrer Einbettung im Graph neu zu typisieren. Auch wenn der Datensatz aktualisiert wird, können diese Konsolidierungsregeln vorab mit einer Regelsequenz abgearbeitet werden und der Datensatz automatisch für weitere Analysen vorbereitet werden. Soll bei einer Analyse beispielsweise auch Handelsware berücksichtigt werden, kann die Konsolidierung einfach angepasst werden, indem die entsprechende Regel nicht angewendet oder angepasst wird. Zudem können weitere Datenquellen auch nachträglich im Graph mit aufgenommen und in die Analyse eingebunden werden.

Der Abkündigungsprozess und insbesondere die Vorbereitung von Abkündigungsentscheidungen sind repetitive Aufgaben im Produktportfoliomanagement. Die hier vorgestellten Analysemethoden wurde regelbasiert formalisiert, so dass sie sich immer wieder auf neue Graphen, die mit aktualisierten Datensätzen instanziiert wurden, anwenden lassen. Besonders die Suche nach Mustern gestaltet sich ohne entsprechende Rechnerwerkzeuge als schwierige Aufgabe. Knotenkette als wiederkehrende Muster in einem Graphen, wie in Abbildung 6-18 oder Abbildung 6-19 dargestellt, sind „von Hand“, mit Tabellenkalkulationsprogrammen oder mit konventionellen Datenbankabfragen nicht auffindbar. Hier kann die musterbasierte Suche in Verbindung mit der Graphtransformation genutzt werden, neue Analyseverfahren zu entwickeln und einzusetzen.

Die Mustererkennung erlaubt es auch, neuartige Sichten auf den Graphen zu generieren, die als Unterstützung in Entscheidungssituationen genutzt werden können. In dieser Fallstudie wurden exemplarisch

- eine umsatz- und produktstatus-basierte Übersicht über das Portfolio (vgl. Abbildung 6-15),
- eine Portfoliodarstellung von Produkten sortiert nach Umsatz und Deckungsbeitrag (vgl. Abbildung 6-17) und

- eine Darstellung von produktfamilien-übergreifender Wiederholteile (vgl. Abbildung 6-20)

vorgestellt. Die Darstellungen erlauben es, Produkte gezielt für Abkündigungsentscheidungen zu priorisieren. Mit den Verantwortlichen der Elektronik GmbH wurden die Ergebnisse diskutiert und die Darstellungsformen evaluiert. Hierbei wurde angemerkt, dass die Visualisierungen neue Einblicke erlauben, die in der Vorbereitung und Durchführung von Abkündigungsentscheidungen herangezogen werden sollten. Die Darstellungsformen und die verwendeten Kriterien müssen weiter verfeinert und auf die Bedürfnisse der jeweiligen Entscheidungssituation detaillierter angepasst werden. Die Methodik zur Mustererkennung in komplexen Produktportfolios eignet sich dafür, weitere Anforderungen an die Entscheidungsunterstützung umzusetzen und neue Ideen für die Analyse und Bewertung von Produktportfolios zu entwickeln.

## 7. Diskussion des Beitrags der Arbeit und Implikationen

**Ziel der Arbeit** war es, Antworten darauf zu finden, wie sich in dem Entwicklungsumfeld der Systemarchitektur das Arbeiten mit dem Produktportfolio angesichts der hohen Komplexität effektiver und effizienter gestalten lässt. Daraus wurden die drei Hauptziele formuliert, nämlich eine Methodik zu erarbeiten, die in einem komplexen Produktportfolio (1) Transparenz schafft sowie (2) effektiveres und (3) effizienteres Arbeiten ermöglicht. Die Methodik zur Mustererkennung in komplexen Produktportfolios als neuer Ansatz für die Unterstützung des Systemarchitekten in komplexen Entscheidungssituationen sollte diese Ziele adressieren. In Abbildung 7-1 wird der Argumentationsstrang, der zur Entwicklung dieser Methodik geführt hat, zusammengefasst.

Ausgehend von den Fundamentalzielen eines Unternehmens (Zeit-Kosten-Qualität), wurden die **Fundamental- und Instrumentalziele** des Systemarchitekten abgeleitet. Voraussetzungen für eine erfolgreiche Zielerfüllung sind sich aufwandsarm einen Überblick über das Portfolio machen zu können, klare und konsistente Strukturen pflegen zu können und effektiv kommunizieren zu können. Dies ist in Anbetracht der hohen Komplexität von Produktportfolios oft schwierig. Aus einer umfangreichen Literaturrecherche ging hervor, dass die Ursachen für die stetig wachsende Komplexität primär in der Gestaltung interner Unternehmensprozesse zu finden sind. Es fehlt an geeigneten Werkzeugen, systematischen Informationsmanagement und effektiven Suchsystemen. Hinzu kommen Gestaltungsregeldefizite und hoher Zeitdruck.

Dies führt zu den **Hürden** Mangel an Transparenz, hoher Aufwand für die Informationsbeschaffung, hoher Aufwand für die Analyse und Bewertung sowie unklare Handlungsfelder und ineffektive Fehlersuche. Diese Hürden wirken sich negativ auf die Erreichung der Ziele des Systemarchitekten aus. Mit dem Erreichen der oben genannten Forschungsziele, können diese Hürden abgemildert oder sogar überwunden werden.

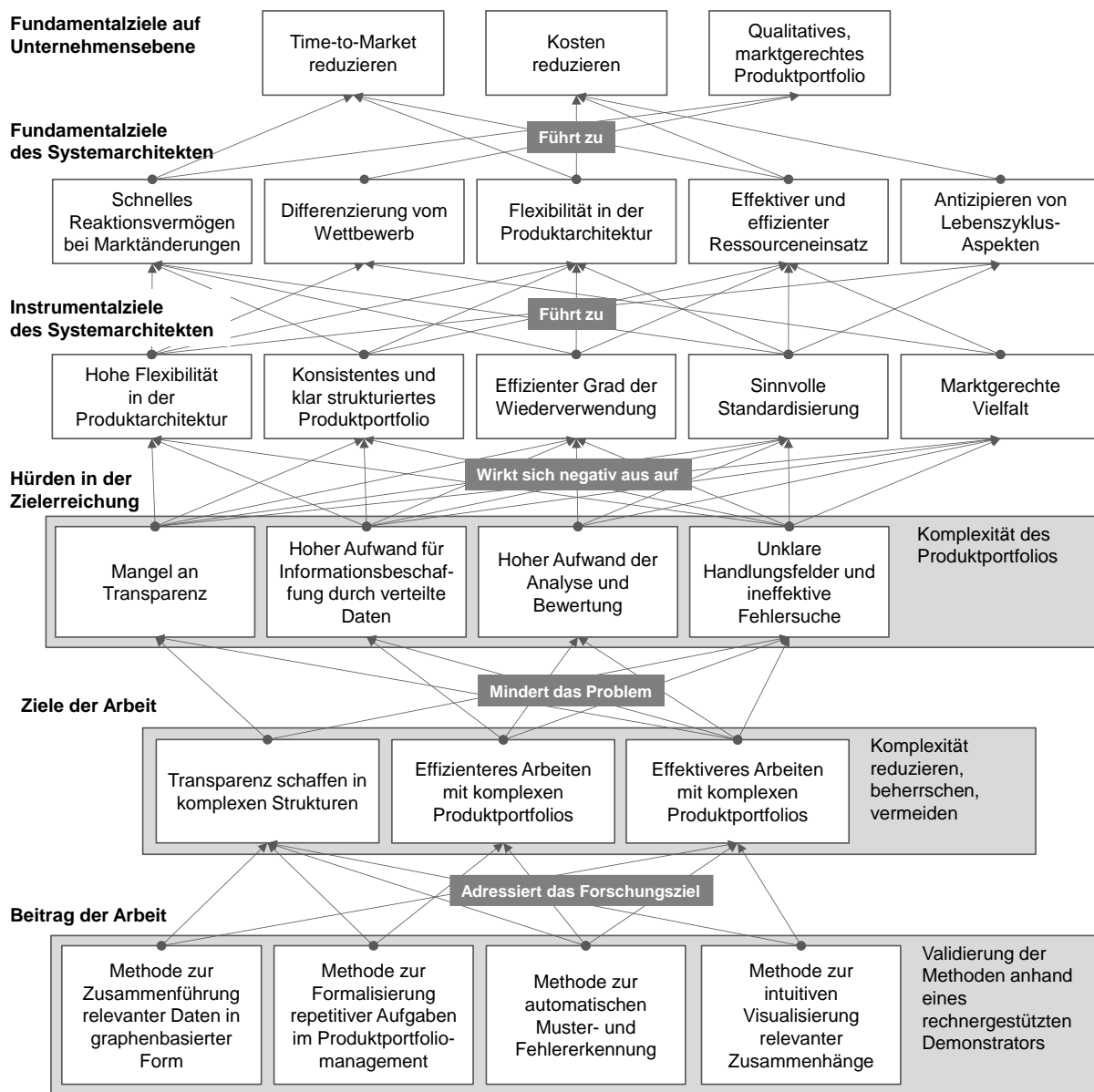


Abbildung 7-1: Zielestruktur der Arbeit

Zunächst sollen allgemeine Schlussfolgerungen aus den beiden Fallstudien gezogen werden. Im Anschluss daran werden der **Beitrag der entwickelten Methoden** und Implikationen für die praktische Anwendung diskutiert.

## 7.1 Schlussfolgerungen aus den Fallstudien

In beiden Industriestudien konnte die Anwendbarkeit und Nützlichkeit der Methodik in der Praxis umfassend evaluiert und die unterschiedlichen Aspekte der Mustererkennung an praktischen Beispielen aufgezeigt werden. In beiden Fällen wurde durch Einsatz der Methodik mehr Transparenz geschaffen und Wege aufgezeigt, die Arbeit der Produktportfolioverantwortlichen effektiver und effizienter zu gestalten.

Sowohl in der Fallstudie bei Hilti als auch bei der Elektronik GmbH konnte gezeigt werden, dass sich Regeln prinzipiell wiederverwenden oder aufwandsarm anpassen lassen. Je spezieller eine Auswertung ist, desto kleiner wird der **Grad der Wiederverwendbarkeit von allgemeinen Regelwerken**. Die Beispiele zeigen, dass sich Analyseschritte in Regelwerken zusammenfassen lassen, die im Prinzip wiederverwendbar sind und somit auch auf andere Daten angewendet werden können. Voraussetzung ist allerdings, dass die Strukturierung der Datenbasis den Grundanforderungen der verwendeten Regeln gerecht wird, also die entsprechenden Attribute, Knoten- und Kantentypen aufweist. Die Methodik ermöglicht demnach, Expertenwissen zu dokumentieren, zu formalisieren und wiederverwendbar zu machen.

In beiden Fallstudien wurde dasselbe Metamodell aus Kapitel 5.2.2 verwendet. Dieses wurde an die jeweiligen Zielstellungen und erforderlichen Analysen in wenigen Details angepasst. Große Teile des Metamodells konnten unverändert eingesetzt werden. Dieser Aspekt kann künftig dazu genutzt werden, **wiederverwendbare Vorlagen („Templates“)** zu entwickeln. Dies gilt auch für die Entwicklung von Regelwerken. Durch die Verwendung von Metamodell- und Regel-Templates vermindert sich der Aufwand in Folgeprojekten. Es ist zu erwarten, dass sich das hier entwickelte Metamodell und die erarbeiteten Regelwerke in wesentlichen Bestandteilen auch in anderen Fallstudien wiederverwenden lassen, wenn damit ähnliche Fragestellungen in anderen Unternehmen bearbeitet werden sollen.

Explizit formalisierte Regelwerke erlauben zudem eine bessere **Nachvollziehbarkeit der Analyseschritte**. Bei der Verarbeitung großer Datenmengen in der Fallstudie der Elektronik GmbH war dies essentiell, um bei der Abkündigung von Produkten rationale Entscheidungen unter Berücksichtigung der gesamten Datenbasis vorbereiten zu können. Mit Regelwerken in Code-Form und der Dokumentation von Zwischenergebnissen durch die Generierung von Reports können die zugrundeliegenden Annahmen explizit und nachvollziehbar dokumentiert werden.

Zudem können die Analysen im Zweifel aufwandsarm und schnell wiederholt werden. Mit einer gezielten Variation von Annahmen oder der Entwicklung von Regelvarianten kann die **Sensitivität der Ergebnisse** getestet werden. Durch solch eine Form der Sensitivitätsanalyse können Ergebnisse umfassender geprüft und die Analysen verbessert werden.

In beiden Fallstudien konnte das **Prinzip der Trennung von Datenbasis, Analyse und Visualisierung** aufgezeigt werden. Im Fall der Elektronik GmbH konnten mit einer Datenbasis Analysen auf unterschiedlichen Detaillierungsstufen und Visualisierungen in unterschiedlichen Darstellungsformen realisiert werden. Mit der Trennung von Datenbasis, Analyse und Visualisierung können unterschiedliche Sichten auf das Problem freier gestaltet und generiert werden. Durch eine einfache Variation der Sichtweisen können umfassendere Erkenntnisse über das behandelte Problem gewonnen werden.

In den Fallstudien hat sich zudem gezeigt, dass viele der **Aufgaben des Systemarchitekten**, wie sie in Kapitel 2 beschrieben wurden, nicht ohne Weiteres durchgeführt werden können. Die Optimierung der Wirtschaftlichkeit eines Produktportfolios durch Wiederverwendung, Standardisierung und marktgerechte Vielfalt, die Entwicklung einer flexiblen Produktportfolioarchitektur, die Pflege konsistenter und klarer Strukturen, die Bewertung des Produktportfolios zur Unterstützung von Steuerungsaufgaben in Entwicklungsprozessen und



Dergleichen lassen sich ohne die nötige Transparenz über die komplexen Zusammenhänge kaum bewerkstelligen. Daher wird die Überwindung der identifizierten Hürden durch Schaffung von Transparenz und Erhöhung der Effektivität und Effizienz als erster nötiger Schritt bewertet. Die vorgestellte Methodik bietet dafür einen praktikablen Ansatz. Der Beitrag der einzelnen Methoden und die Implikationen für die Verwendung der Methodik in der Industrie werden im nächsten Kapitel diskutiert.

## 7.2 Beitrag der Methodik und Implikationen für die praktische Anwendung

Es wurden vier **Methoden** entwickelt, um die Forschungsziele zu adressieren. Diese Methoden wurden mit einem rechnerbasierten Demonstrator mit dem Programm Soley Studio umgesetzt. Im Folgenden werden die vier Methoden diskutiert.

### 7.2.1 Zusammenführung relevanter Daten in graphenbasierter Form

Mit der Methode zur Zusammenführung relevanter Daten in graphenbasierter Form wird ein Beitrag insbesondere zur Transparenzsteigerung und zu einem effektiveren Arbeiten geleistet.

- Die Methode ermöglicht es, verteilte, in gebrauchstüblichen Formaten vorliegende Daten zusammenzuführen und in einem graphenbasierten Modell zu speichern. Zuvor war es meist nötig, sehr viel Aufwand für die Datenbeschaffung, Datenerhebung und konsistente Pflege zu investieren.
- Der große Freiheitsgrad bei der Datenbeschreibung durch eine objekt-orientierte Sprache sowie die freie Typisierung und Attribuierung in der Metamodellierung erlauben es, individuelle Sachverhalte im Unternehmen flexibel abbilden zu können.
- Daten über bestehende, über die Jahre gewachsene Strukturen können in Graphen überführt und konsolidiert werden. Zudem können die Daten weiterhin verteilt gepflegt und bei möglichen Aktualisierungen der Analysen aufwandsarm wieder auf dem letzten Stand zusammengeführt werden.
- Die Mustererkennung kann dazu genutzt werden, durch logische Schlussfolgerungen implizite Zusammenhänge in den importierten Daten explizit nachzupflegen (vgl. Beispiel (14) aus Kapitel 5). Auch eine Retypisierung von Elementen konnte in den Fallstudien exemplarisch gezeigt werden. Zudem können Datenfehler sowie redundante und inkonsistente Informationen über logische Abfragen nachträglich korrigiert werden.

Als **limitierende Faktoren** müssen die Verfügbarkeit von Informationen, die Zugriffsrechte, die Datenqualität und die Rechnerleistung des verarbeitenden Systems angeführt werden.

Das Basis-Metamodell der vorgestellten Methodik umfasst einige Elementtypen, die in den Fallstudien nicht modelliert und somit nicht weiter ausgewertet werden konnten (z. B. Anforderungen, Funktionen, Testfälle oder auch Stakeholder wie Lieferanten oder Kunden). Zum einen lag das an der Zielstellung der jeweiligen Fallstudie, zum anderen auch an einem **Mangel an explizit dokumentierten Daten**. Für eine Verbesserung der Kommunikation im Unternehmen, die Erzeugung stakeholdergerechter Ansichten und die Überwachung der

Einhaltung von Gestaltungsprinzipien muss eine Datenbasis aufgebaut werden, die über die Informationstiefe von Stücklisten hinausgeht. Beispielsweise werden für eine technisch-funktionale Modularisierung Funktionen und Informationen über die Funktionszuordnung zu Komponenten benötigt. Funktionen und deren Zuordnung zu Prozessen und verantwortlichen Mitarbeitern können für die Gestaltung einer funktionalen Ausrichtung der Prozesse und der Organisation im Unternehmen genutzt werden. Mit einer Beziehungsstruktur von der Anforderungsermittlung bis hin zum Recycling könnte ein durchgängiges, produktzentriertes Modell aufgebaut werden, mit dem übergreifende Auswertungen durchgeführt werden könnten.

Im Rahmen der Datenbereitstellung und -pflege muss über ein ausgeglichenes **Kosten-Nutzen-Verhältnis** diskutiert werden. Mit den neuen Möglichkeiten der Formalisierung von Expertenwissen wird der Nutzen einer gut gepflegten, breiteren und detaillierten Datenbasis deutlich erhöht. In diesem Zusammenhang muss aber auch abgewogen werden, wer die Daten bereitstellen und pflegen muss und wer davon spürbar profitiert. Um den Prozess der Datenpflege lebendig zu halten und die **Datenqualität** langfristig hoch zu halten, sollte der, der die Daten pflegt, auch einen unmittelbaren Nutzen davon haben. Zudem müssen Zugriffsrechte auf Daten überdacht werden, wenn diese eine Einschränkung der wirtschaftlichen Nutzung von (nicht personengebundenen) Produktdaten bedeuten.

Bei der Verarbeitung von großen Datenmengen ist ein weiterer limitierender Faktor die **Rechenleistung** eingesetzter Hardware. Große Datenmengen im Produktportfoliomanagement erfordern eine hohe Leistungsfähigkeit der Datenverarbeitung. Durch die Nutzung von GrGen.NET in Soley Studio konnten umfangreiche Transformationen der Datenbasis selbst mit großen Datenmengen in kurzer Zeit (wenige Sekunden) durchgeführt werden. Die Verwendung einer graphenbasierten Datenbasis ermöglicht leistungsfähige Abfragen, da sowohl Graph als auch Regelwerk als ausführbare Dateien im Arbeitsspeicher verfügbar sind und somit keine Abfragen über langsamere Netzwerke getätigt werden müssen.

Die Vorteile einer **graphenbasierten Datenbasis** beschreiben auch ROBINSON ET AL. (2013, S. 8f). Graphenbasierte Datenbasen zeichnen sich durch höhere Leistungsfähigkeit gegenüber relationalen Datenbanksystemen oder NoSQL<sup>37</sup>-Speichern aus. Abfragen können wesentlich schneller durchgeführt werden. In einem Benchmark haben PARTNER ET AL. (2013, S. 8f) die Leistungsfähigkeit eines relationalen (MySQL) mit einem graphenbasierten (Neo4j) Datenbanksystem verglichen. Bei Abfragen direkter Relationen ist noch kein großer Unterschied erkennbar. Sobald aber eine Tiefensuche über mehr als zwei Relationen getätigt wird, zeigen sich die Vorteile des graphenbasierten Systems gegenüber dem relationalen Datenbanksystem. Bei einer Abfrage über vier Relationen hinweg war das graphenbasierte System um Faktor 1000 schneller.

Zudem zeichnen sich graphenbasierte Datenbasen durch ihre **hohe Flexibilität und Agilität** aus (ROBINSON ET AL. 2013, S. 8f). Sie können auch nachträglich um neue Knoten- und Kanten Typen erweitert und angepasst werden ohne existierende Strukturen, Abfragen oder

---

<sup>37</sup> Not only SQL (NOSQL) sind nicht-relationale Datenbanken, auch *structured storages* genannt (<http://de.wikipedia.org/wiki/NoSQL>; entnommen am 01.01.14).

Funktionen zu beeinträchtigen. Somit bietet der Ansatz, eine graphenbasierte Datenbasis zu nutzen, große Vorteile gerade im Umfeld des Produktportfoliomanagements.

## 7.2.2 Formalisierung repetitiver Aufgaben

Die **Methode zur Formalisierung repetitiver Aufgaben im Produktportfoliomanagement** schafft mehr Transparenz in komplexen Strukturen und ermöglicht ein effizienteres Arbeiten.

- Durch die Formulierung von Regeln und Regelsequenzen kann Expertenwissen formalisiert und einem breiteren Anwenderkreis zur Verfügung gestellt werden. Dieser Aspekt kann genutzt werden, das Wissensmanagement in Unternehmen zu unterstützen. Dies erhöht allerdings auch Anforderungen an die Datensicherheit.
- Die Arbeitsschritte, die nötig sind, die Ordnung im Produktportfolio zu erhöhen, können formalisiert und somit die Frequenz von Datenpflegemaßnahmen gesteigert werden.
- Der Aufwand für Klassifizierungs- und Strukturierungsmaßnahmen sinkt durch den Einsatz der Graphtransformation. Regeln können dazu verwendet werden, die Ordnung im Produktportfolio zu erhöhen, beispielsweise durch Bildung logischer Bauteilgruppen.
- Aufwendige Bewertungsverfahren können automatisiert und somit eine kontinuierliche Überwachung des Produktportfolios realisiert werden.
- Mit Regeln können logische Zusammenhänge auch in großen Datenbasen automatisiert erkannt und daraus ableitbare Aktionen durchgeführt werden. Das Problem der Datenmenge verliert somit an Gewicht hinsichtlich des Arbeitsaufwandes.

Viele der hier vorgestellten Regeln konnten nur exemplarisch dargestellt werden. Wie in Kapitel 5.1.2 angemerkt wurde, haben individuelle Rahmenbedingungen in den Unternehmen großen Einfluss darauf, welche Regeln wie eingesetzt und welche Ergebnisaufbereitung am sinnvollsten ist. Bei einer größeren Verbreitung dieser Methodik könnte sich mit der Zeit herausstellen, welche Regeln und Regelsequenzen besonders hilfreich sind und sich in der Breite anwenden lassen. Diese könnten dann als **Standardmethoden** in einem Softwarewerkzeug eingebunden werden.

Das Durchführen einer **regelmäßigen Bewertung** eines komplexen Produktportfolios ist zwar aus planungstechnischen und produktstrategischen Gesichtspunkten erstrebenswert, allerdings auch mit hohem Aufwand verbunden. Angesichts der benötigten Zeit und der damit verbundenen Kosten müssen Umfang und Häufigkeit einer „manuellen“ Bewertung wirtschaftlichen Kriterien standhalten. Im Abschnitt 3.2.3 wurden Bewertungsmethoden ausführlich beleuchtet. Probleme und Defizite der Bewertungsverfahren wurden von zahlreichen Autoren angeführt (vgl. Junge05, Heilemann, Hölttä-Otto2012 etc.). Die Datenbeschaffung, die Analyse und Aufbereitung der Bewertung erfordern zahlreiche Einzelschritte. Das Wissen über die Auswertungsmethoden und das Vorgehen ist nur selten so dokumentiert, dass auch Nicht-Experten die Bewertung durchführen könnten. Mit der in dieser Arbeit vorgestellten Methode konnte gezeigt werden, dass sich dieses Fach- und Methodenwissen formalisieren lässt und in Regelwerken abgebildet werden kann. Nach einem initialen Einrichtungsaufwand, kann das Verfahren beliebig oft angewendet werden. Somit bietet diese Methode die Möglichkeit, den Aufwand der Bewertung drastisch zu reduzieren und

die Anzahl und Häufigkeit der Bewertung ganz an die Erfordernisse der zugrundeliegenden Planungsprozesse anzupassen.

Der **Experte** lässt sich durch die Formalisierung **nicht ersetzen**. Nur mit seinem nötigen Fingerspitzengefühl lassen sich planungsrelevante Zusammenhänge und Einblicke gewinnen, die sich nicht in vorgefertigten Regelwerken definieren lassen. Der Experte kann entlastet werden, indem er durch die Methode von den wiederkehrenden, monotonen Datenaufbereitungen und Auswertungen entlastet wird. Dies verschafft ihm mehr Freiräume für kreative Aufgaben, umfassendere Analysen und zielgerichtete Maßnahmenentwicklung.

### 7.2.3 Automatische Muster- und Fehlererkennung

Die Methode zur automatischen Muster- und Fehlererkennung leistet ihren Beitrag zu allen drei Zielen dieser Arbeit.

- Durch die Kombination von Attribut- und Struktursuche können Zusammenhänge in komplexen Netzwerken effektiver gesucht und gefunden werden. Dadurch werden die Abfragemöglichkeiten im Produktportfoliomanagement erweitert (z. B. durch die Ähnlichkeitssuche).
- Durch die automatische Muster- und Fehlererkennung kann das Suchen in komplexen Produktportfolios effizienter durchgeführt werden. Eine graphenbasierte Mustersuche hat signifikante Vorteile gegenüber der Mustersuche in relationalen Datenbanksystemen (vgl. Abschnitt 7.2.1).
- Kritische Handlungsfelder können durch die Definition von Fehlermustern identifiziert werden. Abweichungen von Sollmustern können somit schnell erkannt und Maßnahmen eingeleitet werden.
- Mit einer Mustersuche über die gesamte Datenbasis können neuartige, muster-basierte Kennzahlen entwickelt und aufwandsarm ermittelt werden. Dies erweitert die Bewertungsmöglichkeiten.

Die Methode zur Muster- und Fehlererkennung kann als Teil der Methode zur Automatisierung repetitiver Aufgaben angesehen werden. Einige der formalisierten Tätigkeiten beinhalten mustergestützte Suchvorgänge. Dennoch soll ihre Bedeutung der Muster- und Fehlersuche hier noch mal hervorgehoben werden.

Durch die Kombination von Attribut- und Struktursuche kann die Transparenz in komplexen Systemen effektiv verbessert werden, indem gezielt nur entscheidungsrelevante Informationen dargestellt wird. Dadurch können komplexe Zusammenhänge einfach **auf relevante Informationen reduziert** werden. Dies erhöht die Übersichtlichkeit und entlastet den Entscheider.

Durch ein Anwachsen des Regelwerkes können die Methoden weiter professionalisiert und die Vorteile der Rechnerwerkzeuge für eine **Unterstützung der Systemarchitektur** genutzt werden. Mit effektiver Such- und Fehler-Mustererkennung können kritische Handlungsfelder gezielt erkannt werden. Durch die Identifikation von Fehlern und effizienter Bewertung lassen sich schneller Maßnahmen einleiten. Die Wirksamkeit ergriffener Maßnahmen kann mit kontinuierlicher Überwachung nachhaltig festgestellt werden. Planungen für mögliche

strukturelle Änderungen im Produktportfolio können simuliert werden, indem der prognostizierte Zustand im Graph abgebildet und bewertet wird.

Die Verwendung von Mustern bringt zusätzlichen Erkenntnisgewinn in der **Bewertung** von Systemen. Mit attributbasierten Messwerten können die Eigenschaften von Systemen quantifiziert werden. Eine solche Bewertung täuscht falsche Objektivität vor, da die Beziehungsstruktur außer Acht gelassen wird. Von einer Änderung der Eigenschaften muss die Beziehungsstruktur eines Elements nicht zwangsläufig betroffen sein. Eine rein attributorientierte wie auch eine rein strukturorientierte Bewertung eines Systems vernachlässigen jeweils wichtige Erkenntnisanteile (VESTER 2000, S. 21).

Die Mustersuche kann dazu genutzt werden, neuartige Kennzahlen zu ermitteln. Als hypothetisches Beispiel könnte bei der Elektronik GmbH eine **muster-basierte Kennzahl** für produktfamilien-übergreifende Wiederverwendung entwickelt werden. Im Rahmen der Abkündigungsentscheidung wurde nach Teilen gesucht, die produktfamilien-übergreifend eingesetzt werden. Die über die Mustersuche gefundenen Teile können ins Verhältnis zu der Anzahl aller Teile im Portfolio gesetzt werden. Der produktfamilien-übergreifende Wiederverwendungsgrad (*Product Family Commonality Index*) berechnet sich dann wie folgt:

$$\text{PFCI} = \frac{\text{Anzahl der Teile, die in mindestens zwei Produktfamilien eingesetzt werden}}{\text{Anzahl aller Teile im Produktportfolio}}$$

$$\text{PFCI}_{\text{Elektronik GmbH}} = \frac{1.461}{16.901} = 8,64\%$$

Im Sinne einer wirtschaftlichen Wiederverwendung wäre ein möglichst hoher Wert anzustreben. Im Kontext einer Abkündigung wie bei der Elektronik GmbH sind hohe Anteile von produktfamilien-übergreifend verwendeten Teilen eher ungünstig. Durch die regelbasierte Ermittlung kann eine Kennzahl wie diese automatisch berechnet werden und somit aufwandsarm beobachtet und für Managementaufgaben eingesetzt werden.

#### 7.2.4 Intuitiv verständliche Visualisierung relevanter Zusammenhänge

Die Methode zur intuitiv verständlichen Visualisierung relevanter Zusammenhänge trägt zur Schaffung von Transparenz bei und unterstützt ein effektiveres Arbeiten mit komplexen Produktportfolios. Folgende Aspekte werden von der Methode adressiert:

- Mit der Methode können Elementtypen, Attribute und Strukturmerkmale gezielt für Visualisierungen gefiltert und sortiert werden.
- Die Methode erlaubt einen freien und flexiblen Einsatz von Formen, Farben, Größen und Positionierung von Knoten und Kanten.
- Entscheidungen können durch die Nutzung von Kombinationen relevanter Kriterien in der Darstellung der Ergebnisse effektiver unterstützt werden (z. B. Knotengröße skaliert nach Umsatz und Knoten eingefärbt nach Stückzahlen).

- Nach erstmaligen Erstellungsaufwand einer Visualisierungsform (z. B. die Tapete in der Hilti-Fallstudie) ergibt sich bei weiteren Einsätzen ein Zeitgewinn durch die automatische Erzeugung von Graphiken und Reports.
- Änderungen der Datenbasis durch Löschen, Hinzufügen oder Verändern von Elementen können effizient, ohne einen Mehraufwand zu erzeugen, durch Aktualisierung der Visualisierung berücksichtigt werden.

Informationen können in unterschiedlichen Formen wie z. B. Tabellen, Diagrammen, Graphen oder auch Textform dargestellt werden. Mit der Wahl einer Repräsentationsform wird in der Regel ein Zweck verfolgt. So besteht beispielsweise bei der Ergebnisdarstellung einer Quality Function Deployment (QFD) die Übersichtlichkeit über die Relationen zwischen kundenwerten Eigenschaften und Produktmerkmalen im Vordergrund. Die erstellten Informationen liegen nun in jenen Matrizen als Ergebnis vor. Wenn die Informationen oder auch nur ein Teilaspekt der Informationen für einen anderen Zweck genutzt werden sollen, müssen diese manuell extrahiert und neu aufbereitet werden. Wenn die benötigten Informationen hingegen in Form eines typisierten, attribuierten Graphs vorliegen, kann das Modell mit den entsprechenden Visualisierungsregeln in beliebiger Form dargestellt werden.

Somit ist eine Wieder- und Weiterverwendung von Informationen in Modellen für andere Aufbereitungen möglich. Auf Seite der Datenpflege und -bereitstellung kann man sich auf die Konsistenz der Datenbasis konzentrieren. Änderungen werden an zentraler Stelle vorgenommen. Sämtliche abgeleitete Modelle und deren Visualisierungen lassen sich aufwandsarm aktualisieren und somit auf dem neusten Stand halten.

### 7.3 Limitationen des Lösungsansatzes

Es wurde eine neue Methodik entwickelt und **prototypisch** in zwei Fallstudien in der Industrie eingesetzt. Empirische Beobachtungen beschränken sich daher auf wenige Unternehmen und Anwendungsszenarios. Dadurch lässt sich noch **keine Allgemeingültigkeit** der Methodik feststellen. Der Einsatz der Methodik in einem breiteren Anwenderkreis, in anderen Branchen und für andere Problemstellungen konnte im Rahmen dieser Forschungsarbeit nicht evaluiert werden.

Die Ergebnisse der Analysen konnten den Verantwortlichen in der Industrie im Rahmen von Workshops zurückgespiegelt werden. Regeln aus den fünf Kategorien der Wissensformalisierung konnten exemplarisch angewendet werden und führten zu nachvollziehbaren Ergebnissen. Die **Nützlichkeit** der Methodik und der unmittelbare **Mehrwert** der Ergebnisse wurden von den Industriepartnern bestätigt. Allerdings lassen sich die mittel- und langfristigen Wirkungen der abgeleiteten Maßnahmen nicht mehr im Rahmen dieser Arbeit prüfen. Somit ist letztlich auch nicht der Beweis erbracht, ob tatsächlich die richtigen Handlungsfelder aufgezeigt wurden. Es konnten nur exemplarisch die Möglichkeiten der Regelanwendung im Rahmen der Graphtransformation aufgezeigt werden. Das hier vorgestellte Metamodell und Regelwerk sollen daher als erster Schritt und „Kristallisationspunkt“ verstanden werden.

Die **Anwendbarkeit** der Methodik kann lediglich aus Sicht des Autors bestätigt werden. Programmierung, Datenverarbeitung und Darstellung der Ergebnisse wurden ausschließlich

durch ihn durchgeführt. Es konnte nicht festgestellt werden, ob ein ungeschulter Nutzer ohne Hilfestellung zu den gleichen Ergebnissen kommen könnte. Die Formulierung von Regelwerken erfordert das Einarbeiten in Skriptsprachen. Um die Benutzerfreundlichkeit der Methode zu verbessern, könnte der Schritt der Regelformulierung durch einen graphischen Regeleditor unterstützt und somit auf Programmierung weitestgehend verzichtet werden.

Mit der Verwendung von Soley Studio wurde eine **Software** eingesetzt, deren Leistungsfähigkeit, Verfügbarkeit, Flexibilität in der Anpassbarkeit und persönliche Vorerfahrungen des Autors entscheidungsrelevante Kriterien bei der Auswahl darstellten. Es wurde auf ein Benchmark von Softwarealternativen verzichtet, da dies außerhalb des Fokus der Untersuchungen lag. Es kann daher nicht ausgeschlossen werden, dass alternative Software-Werkzeuge noch leistungsfähigere Lösungen darstellen.





## 8. Zusammenfassung und Ausblick

Abschließend werden die wesentlichen Erkenntnisse dieser Forschungsarbeit zusammengefasst. Zudem wird ein Ausblick auf weitere Schritte gegeben.

### 8.1 Zusammenfassung

#### Problemstellung und Ziele der Arbeit

Das Arbeiten mit komplexen Produktportfolios stellt Systemarchitekten vor große Herausforderungen. Steigende Vielfalt, zahlreiche Abhängigkeiten innerhalb des Produktportfolios und Wechselwirkungen mit Stakeholdern machen das Produktportfoliomanagement zu einer komplexen Aufgabe. Doch es fehlen geeignete Werkzeuge und Methoden, die mangelnde Transparenz zu reduzieren, den hohen Aufwand für die Informationsverarbeitung zu senken und die mühsamen Bewertungsverfahren zu vereinfachen.

Ziel der Arbeit ist es, eine Methodik zu entwickeln, die den Systemarchitekten dabei unterstützt, einerseits sich effektiv und effizient ein Bild des Gesamtsystems zu machen und andererseits schnell kritische Handlungsfelder im Produktportfolio identifizieren und bewerten zu können. Die zentrale Forschungsfrage lautet demnach, wie sich im Entwicklungsumfeld der Systemarchitektur das Arbeiten mit dem Produktportfolio angesichts der hohen Komplexität effektiver und effizienter gestalten lässt. Die drei abgeleiteten Forschungsziele sind (1) Transparenz schaffen in einem komplexen Produktportfolio, (2) effektiveres und (3) effizienteres Arbeiten mit komplexen Produktportfolios. Die Methodik soll mittels eines rechnergestützten Demonstrators umgesetzt werden. Die Arbeit verbindet die Themenfelder Systems Engineering und variantenreiche Produktgestaltung mit der Graphtransformation. Die Graphtransformation wird eingesetzt, um in den beiden Ingenieursthemen zu Effektivitäts- und Effizienzsteigerungen zu gelangen.

Zunächst wurden in der Aufbereitung der Problemstellung relevante Begrifflichkeiten abgegrenzt. Neben dem allgemeinen Systemverständnis wurde die Komplexität von Systemen definiert. Sie drückt sich durch die Konnektivität (Beziehungsarten und Verknüpfungsdichte) und die Varietät (Unterschiedlichkeit der Elemente und Menge an Elementen) der in einem System enthaltenen Elemente und Beziehungen aus. Das Systems Engineering bietet als Disziplin Vorgehensmodelle, Methoden und Prozessbausteine, um komplexe Systeme zu entwickeln. Zentraler Aspekt ist dabei die Systemarchitektur. Jedes System hat eine Architektur. Somit können einzelne Produkte oder auch das Produktportfolio eines Unternehmens eine Architektur aufweisen. Im Rahmen dieser Forschungsarbeit wurde Produktportfolioarchitektur wie folgt definiert: *Sie beschreibt die Elemente der Produktstruktur, deren horizontalen wie vertikalen Relationen sowie die zugrundeliegenden Gestaltungsprinzipien.* Um mit Architekturen effektiv und effizient arbeiten zu können, bietet sich die Mustererkennung an. Sowohl attribut-basierte als auch strukturelle Beschreibungen können als Bedingungen für die Definition eines Musters genutzt werden. Muster können mit

der Graphtransformation effizient in komplexen Strukturen gefunden und weiterverarbeitet werden.

Zentrale Rolle im Systems Engineering nehmen Systemarchitekten ein, auf deren Unterstützung die hier entwickelte Methode abzielt. Sie überführen im Entwicklungsprozess die Anforderungen an ein System in Teilprobleme, die dann von Fachabteilungen entwickelt werden. Zudem machen sie Vorgaben zur Integrationsstrategie und zur Definition von Testfällen. Innerhalb der Entwicklung nehmen sie eine Überblicks- und Koordinationsfunktion wahr. Die Systemarchitekten haben eine Reihe von Zielen. Fundamentalziele sind

- Schnelles Reaktionsvermögen bei Marktänderungen,
- Differenzierung vom Wettbewerb,
- Flexibilität in der Produktarchitektur,
- effektiver und effizienter Ressourceneinsatz und
- das Antizipieren von Lebenszyklusaspekten.

Instrumentalziele, um die obenstehenden Ziele zu erreichen, sind:

- Hohe Flexibilität in der Produktarchitektur,
- ein konsistentes und klar strukturiertes Produktportfolio,
- die Wahl eines effizienten Grads der Wiederverwendung,
- sinnvolle Standardisierung und
- die Pflege einer marktgerechten Vielfalt.

Um diese Ziele erreichen zu können, muss sich der Systemarchitekt jederzeit einen Überblick über den Ist-Zustand des gesamten Produktportfolios machen können. Zudem muss er aufwandsarm relevante Problemstellungen analysieren und beurteilen sowie Fehler und Handlungsfelder erkennen können. Daher muss er über die Mittel und Werkzeuge verfügen, Ordnung im Produktportfolio zu halten, klare und konsistente Strukturen zu pflegen und ungewollte Vielfalt zu vermeiden.

In der Literatur finden sich zahlreiche Strukturierungsmöglichkeiten und Methoden, ein komplexes System beherrschbarer zu machen. Neben Produkten weisen auch Produktportfolios typische Strukturen auf. Mögliche Strukturierungsmöglichkeiten sind Baureihen, Plattformen, Baukastensysteme und die Klassifizierung von Bausteinen. Methoden zur Komplexitätsreduktion umfassen Methoden aus der variantengerechten Produktgestaltung (z. B. Variant Mode and Effect Analysis oder der Integrierte PKT-Ansatz zur Entwicklung modularer Produktfamilien). Zudem gibt es zahlreiche Bewertungsmethoden, die die Entscheidungsfindung in komplexen Systemen unterstützen sollen. Werkzeuge des Komplexitätsmanagements sind beispielsweise die matrix-basierten Methoden des Structural Complexity Managements. Des Weiteren gibt es die Ansätze Neuronale Netze und Data Mining.

Trotz der Möglichkeiten, Komplexität zu reduzieren und zu beherrschen, tragen unternehmensinterne (z. B. organisatorische Defizite) wie externe Ursachen (z. B. Markt, Kunde, Wettbewerb) dazu bei, die Vielfalt und somit die Komplexität im Produktportfolio zu erhöhen. Diese Arbeit zielt auf übergeordnete, unternehmens-interne Ursachen ab, d. h. methodische Defizite, Transparenz- und Informationsdefizite, Gestaltungsregeldefizite sowie

Zeitproblematik. Durch sie entstehen grundlegende Hürden, die der Systemarchitekt überwinden muss, um seine Ziele erreichen zu können:

- Mangel an Transparenz,
- hoher Aufwand für Informationsbeschaffung durch verteilte Daten,
- hoher Aufwand für Analysen und Bewertungen sowie
- unklare Handlungsfelder und ineffektive Fehlersuche

Aus dieser Analyse wurden Anforderungen an den Lösungsansatz abgeleitet und im Folgenden eine Methodik zur Lösung der Problemstellung erarbeitet.

### **Lösungsansatz**

Der Lösungsansatz basiert auf einer graphenbasierten Repräsentation und Manipulation von komplexen Strukturen. Die Methodik ermöglicht es, verteilte und heterogene Produktdaten in einem Graphen zusammenzuführen und auf diesem formalisiertes Expertenwissen zur Analyse, Bewertung und Visualisierung von Produktportfolios anzuwenden. Graphen bestehen aus Knoten und Kanten. Diese Elemente des Graphs können attribuiert und typisiert werden. Grundlage hierfür ist ein Metamodell, in dem das Datenschema des Graphs definiert wird. Das Metamodell wurde im Rahmen dieser Forschungsarbeit mit einer Domänen-Spezifischen Sprache („DSL“) entwickelt und definiert. Somit können komplexe Strukturen, wie die eines Produktportfolios, mit allen relevanten Aspekten und Zusammenhängen modelliert werden.

Kern der Methodik ist die Graphtransformation. Mittels Regeln können auf der Linken Seite einer Regel Muster definiert werden. Entsprechen Teile des Graphs den Bedingungen der Linken Seite, wird auf der Rechten Seite mit den gefundenen Objekten eine definierte Aktion ausgeführt. Durch diesen einfachen Aufbau können mächtige Auswertungen aufgebaut werden. Die Methodik wurde mit einem rechnergestützten Demonstrator entwickelt. Für die Entwicklung des Metamodells, der Regelwerke und für die Durchführung der Fallstudien wurde Soley Studio verwendet.

Die Methodik zur Mustererkennung in komplexen Produktportfolios umfasst vier Methoden:

- Zusammenführung relevanter Daten in graphenbasierter Form
- Formalisierung repetitiver Aufgaben im Produktportfoliomanagement
- Automatische Muster- und Fehlererkennung
- Intuitiv verständliche Visualisierung relevanter Zusammenhänge

Verteilte Daten werden in einem Graphen als Datenbasis für weitere Analysen zusammengeführt. Hierfür wird das entwickelte Metamodell eingesetzt. Die Formalisierung der Aufgaben des Systemarchitekten erfolgt durch Erstellung von Regelwerken in den Aufgabenfeldern (1) Datenkonsolidierung, (2) Suchmuster und Diagnose von Systemfehlern, (3) Bewertung und Überwachung, (4) Design und Strukturierung und (5) Planung und Prognose. Zu jedem dieser Themenfelder wurden Beispiele für Regeln und Regelsequenzen entwickelt, die auf den Graph als Datenbasis angewendet werden können. Im letzten Schritt werden die Ergebnisse visualisiert. Die Darstellung des Ergebnisgraphs ist voll beeinflussbar. Es können Form, Farbe, Größe und Position der Knoten und Kanten bestimmt oder alternativ automatische Layouter genutzt werden.

## Ergebnisse

Die Methodik wurde anschließend in zwei Fallstudien in der Industrie eingesetzt und evaluiert. In der Fallstudie bei Hilti war es das Ziel, die Transparenz im Produktportfolio der Antriebsentwicklung zu erhöhen, um somit Entscheidungssituationen effizienter unterstützen zu können. Die Erstellung der bereits eingesetzten Überblicksdarstellungen über das Portfolio („Tapeten“) ist mit sehr hohem Aufwand verbunden. Mit der Methodik konnte das Expertenwissen zur Erstellung der Tapete formalisiert und somit automatisiert werden. Dies führte zu einer signifikanten Effizienzsteigerung und zur aufwandsarmen Schaffung von mehr Transparenz.

In der Fallstudie bei der Elektronik GmbH mussten Abkündigungsentscheidungen innerhalb eines sehr großen über Jahre gewachsenen Produktportfolios (ein Graph mit etwa 30.000 Knoten und 150.000 Relationen) unterstützt werden. Es konnte gezeigt werden, dass die Methodik Ergebnisse erzeugen konnte, die mit manuellem Aufwand und Verwendung konventioneller Hilfsmittel nicht umsetzbar gewesen wären. Die erzeugten Visualisierungen konnten die Entscheidungen signifikant unterstützen.

Das Metamodell und Teile der entwickelten Regelwerke konnten innerhalb der Fallstudien weitgehend wiederverwendet werden. Durch die Formalisierung können Analyseschritte beliebig oft und aufwandsarm wiederholt werden. Metamodell wie auch Regelwerke sind flexibel anpassbar und erweiterbar. Eine Übertragung auf andere Fragestellungen im Umfeld des Produktportfoliomanagements ist denkbar, wurde aber im Rahmen dieser Forschungsarbeit nicht validiert.

Mit der Methodik zur Mustererkennung in komplexen Produktportfolios konnte ein Lösungsansatz präsentiert werden, die Transparenz in komplexen Strukturen systematisch zu verbessern und eine Effektivitäts- und Effizienzsteigerung in der Arbeit des Systemarchitekten zu erzielen.

## 8.2 Ausblick

### Berücksichtigung von Dynamik

In dieser Arbeit wurde der Aspekt der Dynamik nicht detaillierter betrachtet, um den Fokus der Arbeit nicht zu breit zu setzen. Nichtsdestotrotz spielen Veränderungen eine bedeutende Rolle. Gerade die Abbildung von **Änderungen** in netzwerkartigen Strukturen und deren Beobachtung über die Zeit bergen ein großes Potential für Planungs- und Steuerungswerkzeuge in sich. Die Werkzeuge, die in dieser Arbeit vorgestellt wurden, können als Grundstock hierfür herangezogen werden. Ein Beispiel hierzu war die Zeitreihenanalyse in Beispiel (17) in Kapitel 5 (S. 121). Jedem Elementtyp können entsprechende Datenfelder zugeordnet werden, die mit einem Zeitstempel aktualisiert werden, sobald sich ein Knoten dieses Typs oder Relationen dieses Knotens verändern. Regelmäßige Zwischenspeicherungen des Gesamtgraphen können dann genutzt werden, um Zeitreihen mit entsprechenden Regeln analysieren und auswerten zu können. Erforderlich wäre hierfür allerdings eine performantere Rechnerarchitektur mit geeigneten Speichermöglichkeiten.

## Weiterentwicklung hin zu einem Expertensystem für Systemarchitekten

Die hier entwickelte Methodik und deren Umsetzung in einem rechnergestützten Demonstrator kann ein erster Schritt sein, um ein Expertensystem für die Systemarchitektur aufzubauen. Expertenwissen kann mit Regeln formalisiert werden und ist unter Einhaltung der Kompatibilitätsbedingungen, die sich aus den Definitionen im Metamodell ergeben, wiederverwendbar. Diese Eigenschaften können genutzt werden, um ein Expertensystem für das Produktportfoliomanagement aufzubauen. Expertensysteme unterscheiden sich von Datenbanksystemen durch die Möglichkeit „vom Computer interpretierbare Anleitungen über [die Wissens-] Verwendung“ zur Problemlösung verwenden zu können (PUPPE 1988, S. 11). PUPPE (1988, S. 4f) beschreibt notwendige Eigenschaften eines Expertensystems. Dazu gehören:

- Transparenz
- Flexibilität
- Kompetenz
- Benutzerfreundlichkeit

Mit dem Proof of Concept wurde gezeigt, dass die **Transparenz** im komplexen System eines Produktportfolios gesteigert werden konnte und Entscheidungen rationaler vorbereitet werden können. Weiterhin konnte die **Flexibilität** der Methodik gezeigt werden, dadurch dass sich Wissensbausteine relativ einfach hinzufügen, verändern oder löschen lassen. Die Problemlösungsfähigkeit („**Kompetenz**“) des Demonstrators ist im Rahmen dieser Arbeit noch auf die individuellen Zielstellungen der Fallbeispiele beschränkt. Durch die Wiederverwendbarkeit von Regelwerken können die Wissensbibliotheken über die Zeit wachsen. Somit können die Regelwerke um Lösungsbausteine aus dem Architekturumfeld, aber auch um weitere Aspekte des Systems Engineerings und anderen Unternehmensbereichen erweitert werden. Je abstrakter Lösungen in Regeln formalisiert werden können, desto größer ist die Chance der Wiederverwendbarkeit. Das Ziel ist hierbei wie in technischen Baukastensystemen auch, den Anteil spezifischer Lösungsbausteine möglichst gering und den Anteil wiederverwendbarer Bausteine möglichst groß zu halten.

Die **Benutzerfreundlichkeit** zu erhöhen, gilt als weitere Herausforderung. Ziel muss es hier sein, normalen Nutzern wie auch Experten den Umgang mit der Software auch ohne programmiersprachliches Vorwissen zu ermöglichen. Hierzu kann das System durch Weiterentwicklung eines Graphischen Nutzer Interface (GUI) und der Anbindung von gängigen Datenbanksystemen weiter ausgebaut werden.



## 9. Literaturverzeichnis

ABERNATHY & UTTERBACK 1978

Abernathy, W. J.; Utterback, J. M.: Patterns of Industrial Innovation. In: Technology Review. 80, 1978. 7.

ABRAMOVICI & SCHULTE 2005

Abramovici, M.; Schulte, S.: Lifecycle Management von Produkt-Service-Systemen (PSS) für einen maximierten Kundennutzen. Veröffentlichung des Lehrstuhls. Ruhr-Universität Bochum. Lehrstuhl für Maschinenbauinformatik (ITM). Bochum: 2005.

ANDREASEN & HEIN 1987

Andreasen, M. M.; Hein, L.: Integrated product development. Bedford: IFS (Publications) 1987. ISBN: 9780387166797.

BALDWIN & CLARK 2000

Baldwin, C. Y.; Clark, K. B.: Design rules. The power of modularity, volume 1. Cambridge: MIT Press 2000.

BARDOHL ET AL. 2003

Bardohl, R.; Ehrig, H.; Lara, J. de; Runge, O.; Taentzer, G.; Weinhold, I.: Node Type Inheritance Concept for Typed Graph Transformation. Berlin: 2003.

BEIERLE & KERN-ISBERNER 2008

Beierle, C.; Kern-Isberner, G.: Methoden wissensbasierter Systeme. Grundlagen, Algorithmen, Anwendungen. 4., verb. Aufl. Wiesbaden: Vieweg + Teubner 2008. ISBN: 9783834805041.

BEMBENNECK 2005

Bembenneck, J.: Zauberwort verhilft zur Kostensenkung. Wiederholteilestrategie bietet hohes Einsparungspotential. In: Intelligenter produzieren, 2005. 2.

BIEDERMANN ET AL. 2012

Biedermann, W.; Kirner, K.; Kissel, M.; Langer, S.; Wickel, M.: Forschungsmethodik in den Ingenieurwissenschaften. 2012.

URL: <http://www.pe.mw.tum.de/studium/semesterarbeiten/skript-forschungsmethodik-in-den-ingenieurwissenschaften/> - zuletzt geprüft am: 30.12.2013.

BIEDERMANN & LINDEMANN 2012

Biedermann, W.; Lindemann, U.: Correlation of Structural Characteristics of Product Design Structure Matrices. In: Proceedings of the International Design Conference 2012, Dubrovnik, 2012.

BIEGERT 1971

Biegert, H.: Die Baukastenbauweise als technisches und wirtschaftliches Gestaltungsprinzip. Dissertation.

Universität Karlsruhe (TH). Karlsruhe: 1971.

BLACKENFELT 1999

Blackenfelt, M.: On the Development of Modular Mechatronic Products. Dissertation. The Royal Institute of Technology. Stockholm: 1999.

BLEES ET AL. 2010

Blees, C.; Kipp, T.; Beckmann, G.; Krause, D.: Development of Modular Product Families: Integration of Design for Variety and Modularization. In: Proceedings of NordDesign Conference 2010, Göteborg, Schweden, 2010.

BLEES 2011

Blees, C.: Eine Methode zur Entwicklung modularer Produktfamilien. Technischen Universität Hamburg-Harburg. Institut für Produktentwicklung und Konstruktion. Hamburg: 2011.

BLESSING & CHAKRABARTI 2009

Blessing, L. T. M.; Chakrabarti, A.: DRM, a design research methodology. Dordrecht, New York: Springer 2009. ISBN: 9781848825864.

BLOMER ET AL. 2013

Blomer, J.; Geiss, R.; Jakumeit, E.: The GrGen. NET User Manual. 2013.  
URL: [www.grgen.net/](http://www.grgen.net/) - zuletzt geprüft am: 15.11.2013.

BONGULIELMI 2002

Bongulielmi, L.: Die Konfigurations- & Verträglichkeitsmatrix als Beitrag zur Darstellung konfigurationsrelevanter Aspekte im Produktentstehungsprozess. Dissertation. Eidgenössische Technische Hochschule. Zürich: 2002.

BRAUN ET AL. 2013

Braun, B.; Kissel, M.; Förg, A.; Kreimeyer, M.: Referenzmodell zur Analyse von Baukastensystematiken in der Nutzfahrzeugindustrie. In: Tagungsband zum 24. DfX-Symposium, Hamburg, 2013.

BROWNING 2001

Browning, T. R.: Applying the design structure matrix to system decomposition and integration problems: a review and new directions. In: IEEE Trans. Eng. Manage. 48, 2001. 3.

BROWNING & HONOUR 2008

Browning, T. R.; Honour, E. C.: Measuring the Life-Cycle Value of Enduring Systems. In: Systems Engineering. 11, 2008. 3.

BRUNS 1991

Bruns, M.: Systemtechnik. Ingenieurwissenschaftliche Methodik zur interdisziplinären Systementwicklung. Berlin: Springer 1991. ISBN: 978-3540534679.

BULLINGER ET AL. 2002

Bullinger, H.-J.; Warnecke, H.-J.; Westkämper, E.: Neue Organisationsformen im Unternehmen. Ein Handbuch für das moderne Management. 2., neu bearb. u. erw. Aufl. Berlin: Springer 2002. ISBN: 9783540676102.

CAESAR 1991

Caesar, C.: Kostenorientierte Gestaltungsmethodik für variantenreiche Serienprodukte.



Dissertation.

RWTH. Aachen: 1991.

CHOMSKY 1956

Chomsky, N.: Three models for the description of language. In: IRE Transactions on Information Theory. 2, 1956.

CLANCEY 1985

Clancey, W. J.: Heuristic classification. In: Artificial Intelligence. 27, 1985. 3.

CLARKSON ET AL. 2004

Clarkson, P. J.; Simons, C.; Eckert, C.: Predicting Change Propagation in Complex Design. In: J. Mech. Des. 126, 2004. 5.

COLFER & BALDWIN 2010

Colfer, L.; Baldwin, C.: The Mirroring Hypothesis: Theory, Evidence and Exceptions. In: Harvard Business School Working Paper, 2010. 10-058.

CORRADINI ET AL. 1996

Corradini, A.; Montanari, U.; Rossi, F.: Graph Processes. In: Fundamenta Informaticae - Special issue on graph transformations. 26, 1996. 3 - 4.

CRAWLEY ET AL. 2004

Crawley, E.; Weck, O. de; Eppinger, S.; Magee, C.; Moses, J.; Seering, W.; Schindall, J.; Wallace, D.; Whitney, D.: The influence of architecture in engineering systems. 2004.  
URL: <http://esd.mit.edu/symposium/pdfs/monograph/architecture-b.pdf/> - zuletzt geprüft am: 01.01.2014.

DAENZER (Hrsg.) 1977

Daenzer, W. F.: Systems engineering. Leitfaden zur methodischen Durchführung umfangreicher Planungsvorhaben. Köln: Hanstein [u.a.] 1977. ISBN: 3775662006.

DAHL 1990

Dahl, B.: Entwicklung eines Konstruktionssystems zur Unterstützung der montagegerechten Produktgestaltung. Dissertation.  
RWTH. Aachen: 1990.

DANILOVIC & BROWNING 2004

Danilovic, M.; Browning, T. R.: A Formal Approach for Domain Mapping Matrices (DMM) to Complement Design Structure Matrices (DSM). In: Proceedings of the 6th Design Structure Matrix (DSM) International Workshop, EDC, Cambridge, 2004.

DANNER 1996

Danner, S.: Ganzheitliches Anforderungsmanagement für marktorientierte Entwicklungsprozesse. Aachen: Shaker 1996. ISBN: 9783826519086.

DE WECK ET AL. 2003

de Weck, O. L.; Suh, E. S.; Chang, D.: Product Family and Platform Portfolio Optimization. In: Proceedings of DETC'03 2003 ASME Design Engineering Technical Conferences, 2003.

DE WECK ET AL. 2012

de Weck, O. L.; Ross, A. M.; Rhodes, D. H.: Investigating Relationships and Semantic Sets

amongst System Lifecycle Properties (Ilities ). In: Third International Engineering Systems Symposium, Delft University of Technology, 2012.

DE WECK ET AL. 2011

de Weck, O. L.; Roos, D.; Magee, C. L.; Vest, C. M.: Engineering systems. Meeting human needs in a complex technological world. Cambridge, Mass: MIT Press 2011. ISBN: 9780262016704.

DEIMLING 2013

Deimling, C.: Konzept zur Analyse und Gestaltung von Systemarchitekturen multivarianter Produktprogramme. Diplomarbeit.

Technische Universität München. Lehrstuhl für Produktentwicklung. München: 2013.

Department of Defense 2008

Department of Defense: Systems Engineering Guide for Systems of Systems. 2008.

URL: <http://www.acq.osd.mil/se/docs/SE-Guide-for-SoS.pdf/> - zuletzt geprüft am: 23.12.2013.

DIN 2002

DIN: DIN 199-1. Technische Produktdokumentation - CAD-Modelle, Zeichnungen und Stücklisten. Teil 1: Begriffe. Berlin: Beuth 2002.

DÖRNER 2003

Dörner, D.: Die Logik des Misslingens. Strategisches Denken in komplexen Situationen. Erw. Neuausg. Reinbek bei Hamburg: Rowohlt-Taschenbuch-Verl. 2003. ISBN: 9783499615788.

DU ET AL. 2002a

Du, X.; Jiao, J.; Tseng, M. M.: Graph Grammar Based Product Family Modeling. In: Concurrent Engineering: Research and Application. 10, 2002. 2.

DU ET AL. 2002b

Du, X.; Jiao, J.; Tseng, M. M.: Product family modeling and design support: An approach based on graph rewriting systems. In: AIEDAM. 16, 2002. 02.

DYM & LEVITT 1991

Dym, C. L.; Levitt, R. E.: Knowledge-based systems in engineering. New York: McGraw-Hill 1991. ISBN: 0070185638.

EHRIG ET AL. 1999

Ehrig, H.; Engels, G.; Kreowski, H. J.: Handbook of graph grammars and computing by graph transformation. Volume 2: Applications, Languages and Tools. 1999. ISBN: 9789810240202.

EHRENSPIEL ET AL. 2005

Ehrlenspiel, K.; Kiewert, A.; Lindemann, U.: Kostengünstig entwickeln und konstruieren. Kostenmanagement bei der integrierten Produktentwicklung. 5., bearb. Aufl. Berlin, Heidelberg, New York: Springer 2005. ISBN: 3540251650.

EHRENSPIEL & MEERKAMM 2013

Ehrlenspiel, K.; Meerkamm, H.: Integrierte Produktentwicklung. München: Carl Hanser Verlag GmbH & Co. KG 2013. ISBN: 978-3-446-43548-3.

EILMUS ET AL. 2013

Eilmus, S.; Ripperda, S.; Krause, D.: Towards the development of commonal product programs. In: Proceedings of the International Conference on Engineering Design, Seoul, Korea, 2013.

EILMUS & KRAUSE 2012

Eilmus, S.; Krause, D.: An Approach for reducing Variety across Product Families. In: Proceedings of NordDesign Conference 2012, Aalborg, 2012.

EISENFÜHR ET AL. 2010

Eisenführ, F.; Langer, T.; Weber, M.: Rationales Entscheiden. 5., überarb. u. erw. Aufl. Berlin [u.a.]: Springer 2010. ISBN: 978-3-642-02848-9.

ENGEL & BROWNING 2008

Engel, A.; Browning, T. R.: Designing systems for adaptability by means of architecture options. In: Syst. Engin. 11, 2008. 2.

EPPINGER & BROWNING 2012

Eppinger, S. D.; Browning, T. R.: Design structure matrix methods and applications. Cambridge, Mass: MIT Press 2012. ISBN: 0262301423.

ERIXON 1998

Erixon, G.: Modular Function Deployment - A Method for Product Modularisation. Dissertation.

The Royal Institute of Technology. Dept. of Manufacturing Systems. Stockholm: 1998.

EVERSHEIM 1989

Eversheim, W.: Organisation in der Produktionstechnik, Band 4: Fertigung und Montage. Düsseldorf: VDI-Verlag 1989. ISBN: 3-18-.

EVERSHEIM 1999

Eversheim, W.: Produktion und Management "Betriebshütte". 7. Aufl. Berlin [u.a.]: Springer 1999. ISBN: 9783540654681.

FAYYAD ET AL. 1996

Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.: From Data Mining to Knowledge Discovery in Databases. In: AI Magazine. 17, 1996. 3.

FEIGENBAUM 2014

Feigenbaum, L.: SPARQL by Example. 2014.

URL: <http://www.cambridgesemantics.com/semantic-university/sparql-by-example/> - zuletzt geprüft am: 01.01.2014.

FIORE 2005

Fiore, C.: Accelerated product development. Combining lean and six sigma for peak performance. New York, NY: Productivity Press 2005. ISBN: 9781563273100.

FIRCHAU 2003

Firchau, N. L.: Variantenoptimierende Produktgestaltung. 1. Aufl. Göttingen: Cuvillier 2003. ISBN: 9783898739344.

FIRCHAU ET AL. 2002

Firchau, N. L.; Franke, H.-J.; Huch, B.; Menge, M., 2002 Variantenmanagement: Variantenvielfalt in Produkten und Prozessen erfolgreich beherrschen. In: Hans-Joachim Franke, Jürgen Hesselbach, Burkhard Huch und Norman Lee Firchau (Hrsg.): Variantenmanagement in der Einzel- und Kleinserienfertigung. München: Hanser. ISBN: 9783446217300.

FITCH & COOPER 2005a

Fitch, P.; Cooper, J. S.: Life cycle modeling for adaptive and variant design. Part 2: Case Study. In: Research in Engineering Design. 15, 2005. 4.

FITCH & COOPER 2005b

Fitch, P.; Cooper, J. S.: Life-cycle modeling for adaptive and variant design. Part 1: Methodology. In: Research in Engineering Design. 15, 2005. 4.

FIXSON 2005

Fixson, S. K.: Product architecture assessment: a tool to link product, process, and supply chain design decisions. In: Journal of Operations Management. 23, 2005. 3-4.

FRANK 2006

Frank, U.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme. Paderborn: Heinz-Nixdorf-Inst 2006. ISBN: 3935433840.

FRANKE 2002

Franke, H.-J.: Variantenmanagement in der Einzel- und Kleinserienfertigung. Mit 33 Tabellen. München [u.a.]: Hanser 2002. ISBN: 9783446217300.

FRICKE & SCHULZ 2005

Fricke, E.; Schulz, A. P.: Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle. In: Syst. Engin. 8, 2005. 4.

FRIEDENTHAL ET AL. 2011

Friedenthal, S.; Moore, A.; Steiner, R.: A practical guide to SysML. The systems modeling language. [2nd ed.]. Amsterdam, Boston: Morgan Kaufmann 2011. ISBN: 0123852064.

FUJITA & YOSHIDA 2004

Fujita, K.; Yoshida, H.: Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes. In: Concurrent Engineering. 12, 2004. 2.

GAUSEMEIER ET AL. 2008

Gausemeier, J.; Frank, U.; Donoth, J.; Kahl, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus. In: Konstruktion, 2008. 7/8-2008.

GAUSEMEIER 2012

Gausemeier, J.: Maschinenbau braucht Systems Engineering. Fraunhofer Institut. Max-Syrbe-Kolloquium, Karlsruhe. 2012.

GAUSEMEIER ET AL. 2012

Gausemeier, J.; Lanza, G.; Lindemann, U.: Produkte und Produktionssysteme integrativ

konzipieren. Modellbildung und Analyse in der frühen Phase der Produktentstehung. München: Hanser Verlag 2012. ISBN: 3446428259.

GEIß & KROLL 2007

Geiß, R.; Kroll, M.: On Improvements of the Varró Benchmark for Graph Transformation Tools. Technical Report.

Universität Karlsruhe (TH). Institut für Programmstrukturen und Datenorganisation. Karlsruhe: 2007.

GEIß ET AL. 2006

Geiß, R.; Batz, G. V.; Grund, D.; Hack, S.; Szalkowski, A., 2006GrGen: A Fast SPO-Based Graph Rewriting Tool. In: Andrea Corradini (Hrsg.): Graph transformations. Third international conference, ICGT 2006, Natal, Rio Grande do Norte, Brazil, September 17-23, 2006 : proceedings. Berlin, New York: Springer. ISBN: 9783540388708.

GEMBRYNS 1998

Gembrys, S.-N.: Ein Modell zur Reduzierung der Variantenvielfalt in Produktionsunternehmen. Berlin: PTZ [u.a.] 1998. ISBN: 9783816751991.

GENNARI ET AL. 2003

Gennari, J. H.; Musen, M. A.; Ferguson, R. W.; Grosso, W. E.; Crubézy, M.; Eriksson, H.; Noy, N. F.; Tu, S. W.: The evolution of Protégé: an environment for knowledge-based systems development. In: International Journal of Human-Computer Studies. 58, 2003. 1.

GERHARD 1984

Gerhard, E.: Baureihenentwicklung. Grafenau/Württ: expert-Verlag 1984. ISBN: 3885088665.

GERSHENSON ET AL. 2003

Gershenson, J. K.; Prasad, G. J.; Zhang, Y.: Product modularity: Definitions and benefits. In: Journal of Engineering Design. 14, 2003. 3.

GERSHENSON ET AL. 2004

Gershenson, J. K.; Prasad, G. J.; Zhang, Y.: Product modularity: measures and design methods. In: Journal of Engineering Design. 15, 2004. 1.

GIFFIN ET AL. 2009

Giffin, M.; Weck, O. de; Bounova, G.; Keller, R.; Eckert, C.; Clarkson, P. J.: Change Propagation Analysis in Complex Technical Systems. In: J. Mech. Des. 131, 2009. 8.

GONZALEZ-ZUGASTI ET AL. 2000

Gonzalez-Zugasti, J. P.; Otto, K. N.; Baker, J. D.: A Method for Architecting Product Platforms. In: Research in Engineering Design. 12, 2000. 2.

GÖPFERT 1998

Göpfert, J.: Modulare Produktentwicklung. Zur gemeinsamen Gestaltung von Technik und Organisation. Wiesbaden: Dt. Univ.-Verl. 1998. ISBN: 9783824468270.

GRAESSLER

Graessler, I.: Warum Variantenbeherrschung alleine nicht mehr ausreicht – der neue Umgang mit gewollter Varianz in der Automobilindustrie. In: VDI-Gesellschaft Entwicklung,

- Konstruktion und Vertrieb (Hrsg.): Variantenvielfalt in Produkten und Prozessen. Erfahrungen, Methoden und Instrumente. Tagung, Kassel, 7. und 8. November 2001. 2001: VDI Verlag.
- GREISEL ET AL. 2013  
Greisel, M.; Kissel, M.; Spinola, B.; Kreimeyer, M.: Design for Adaptability in multi-variant product families. In: Proceedings of the International Conference on Engineering Design, Seoul, Korea, 2013.
- GRESHAKE 2011  
Greshake, P.-T.: Modularität – das richtige Maß entscheidet! In: Modularer. Kundenzeitung der 3D Systems Engineering GmbH, 2011. 5.
- GU ET AL. 2009  
Gu, P.; Xue, D.; Nee, a Y C: Adaptable design: concepts, methods, and applications. In: Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture. 223, 2009. 11.
- GU & SOSALE 1999  
Gu, P.; Sosale, S.: Product modularization for life cycle engineering. In: Robotics and Computer-Integrated Manufacturing. 15, 1999. 5.
- HABERFELLNER ET AL. 2012  
Haberfellner, R.; de Weck, O. L.; Fricke, E.; Vössner, S.: Systems Engineering - Grundlagen und Anwendung. 12. Auflag. Orell Füssli 2012. ISBN: 978-3280040683.
- HAN ET AL. 2012  
Han, J.; Kamber, M.; Pei, J.: Data mining. Concepts and techniques. 3rd ed. Burlington, MA: Elsevier 2012. ISBN: 9780123814791.
- HANSEN ET AL. 2012  
Hansen, C. L.; Mortensen, N. H.; Hvam, L.: Calculation of Complexity Costs: An Approach for Rationalizing a Product Program. In: Proceedings of NordDesign Conference 2012, Aalborg, 2012.
- HARLOU 2006  
Harlou, U.: Developing product families based on architectures. Contribution to a theory of product families. 1. Aufl. Lyngby: Department of Mechanical Engineering, Technical University of Denmark 2006. ISBN: 978-87-90130-07-7.
- HASHEMIAN 2005  
Hashemian, M.: Design for Adaptability. Dissertation. University of Saskatchewan. Department of Mechanical Engineering. Saskatoon: 2005.
- HASKINS ET AL. 2010JANUARY  
Haskins, C.; Forsberg, K.; Krueger, M.; Walden, D.; Hamelin, R. D.: INCOSE Systems Engineering Handbook v. 3.2. INCOSE, International Council on Systems Engineering 2010.
- HECKEL 2006  
Heckel, R.: Graph Transformation in a Nutshell. In: Electronic Notes in Theoretical Computer Science. 148, 2006. 1.

HEILEMANN ET AL. 2013

Heilemann, M.; Culley, S.; Schlüter, M.; Haase, H.-J.: Examination of modularization metrics in industry. In: Proceedings of the International Conference on Engineering Design, Seoul, Korea, 2013.

HEINA 1999

Heina, J.: Variantenmanagement. Integrierte Kosten-Nutzen-Bewertung zur Optimierung der Variantenvielfalt. Wiesbaden: Dt. Univ.-Verl. 1999. ISBN: 3-8244-7064-0.

HELLENBRAND 2013

Hellenbrand, D.: Transdisziplinäre Planung und Synchronisation mechatronischer Produktentwicklungsprozesse. Dissertation. Technische Universität München. Lehrstuhl für Produktentwicklung. München: 2013.

HELMS 2013

Helms, B.: Object-Oriented Graph Grammars for Computational Design Synthesis. Dissertation. Technische Universität München. Lehrstuhl Für Produktentwicklung. München: 2013.

HELMS & SHEA 2012

Helms, B.; Shea, K.: Computational Synthesis of Product Architectures Based on Object-Oriented Graph Grammars. In: J. Mech. Des. 134, 2012. 2.

HÖLTTA-OTTO & DE WECK 2007

Höltta-Otto, K.; de Weck, O. L.: Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints. In: Concurrent Engineering. 15, 2007. 2.

HÖLTTÄ-OTTO ET AL. 2012

Hölttä-Otto, K.; Chiriac, N. A.; Lysy, D.; Suk Suh, E.: Comparative analysis of coupling modularity metrics. In: Journal of Engineering Design. 23, 2012. 10-11.

HOMBURG & DAUM 1997

Homburg, C.; Daum, D.: Wege aus der Komplexitätsfalle. In: ZWF. 92, 1997. 7/8.

HUBKA & EDER 1992

Hubka, V.; Eder, E. W.: Einführung in die Konstruktionswissenschaft: Übersicht, Modell, Ableitungen. Berlin, Heidelberg: Springer 1992. ISBN: 978-3540548324.

IEEE 2005

IEEE: Adoption of ISO/IEC 15288:2002 systems engineering. System life cycle processes. New York, N.Y: Institute of Electrical and Electronics Engineers 2005. ISBN: 0-7381-4652-8 SS95313.

ISO 2004

ISO: ISO/IEC 11179-1. Information technology - Metadata registries (MDR)- Part 1: Framework. Genf: ISO 2004.

ISO 2012

ISO: ISO/IEC 19505-1. Information technology - Object Management Group Unified Modeling Language (OMG UML) - Part 1: Infrastructure. Genf: ISO 2012.

ISO 2012

ISO: ISO/IEC 19505-2. Information technology - Object Management Group Unified Modeling Language (OMG UML) - Part 2: Superstructure. Genf: ISO 2012.

JARRATT ET AL. 2011

Jarratt, T. a. W.; Eckert, C. M.; Caldwell, N. H. M.; Clarkson, P. J.: Engineering change: an overview and perspective on the literature. In: Res Eng Design. 22, 2011. 2.

JESCHKE 1997

Jeschke, A.: Beitrag zur wirtschaftlichen Bewertung von Standardisierungs-Maßnahmen in der Einzel- und Kleinserienfertigung durch die Konstruktion. Dissertation.

Technische Universität Carolo Wilhelmina zu Braunschweig. Braunschweig: 1997.

JIAO ET AL. 2006

Jiao, J.; Simpson, T. W.; Siddique, Z.: Product family design and platform-based product development: a state-of-the-art review. In: Journal of Intelligent Manufacturing. 18, 2006. 1.

JIAO & TSENG 2000a

Jiao, J.; Tseng, M. M.: Fundamentals of product family architecture. In: Integrated Manufacturing Systems. 11, 2000. 7.

JIAO & TSENG 2000b

Jiao, J.; Tseng, M.: Understanding product family for mass customization by developing commonality indices. In: J. of Eng. Design. 11, 2000. 3.

JIAO & TSENG 2004

Jiao, J.; Tseng, M. M.: Customizability analysis in design for mass customization. In: Computer-Aided Design. 36, 2004. 8.

JONAS ET AL. 2012

Jonas, H.; Gebhardt, N.; Krause, D.: Towards a Strategic Development of Modular Product Programs. In: Proceedings of the International Design Conference, Dubrovnik, 2012.

JUNGE 2005

Junge, M.: Controlling modularer Produktfamilien in der Automobilindustrie. Entwicklung und Anwendung der Modularisierungs-balanced-Scorecard. 1. Aufl. Wiesbaden: Dt. Univ.-Verl. 2005. ISBN: 3-8244-8226-6.

KAISER 1995

Kaiser, A.: Integriertes Variantenmanagement mit Hilfe der Prozesskostenrechnung. Dissertation.

Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften. St. Gallen: 1995.

KANO 1995

Kano, N., 1995 Upsizing the organization by attractive quality creation. In: Gopal K. Kanji (Hrsg.): Total quality management. Proceedings of the first world congress. London, New York: Chapman & Hall. ISBN: 9780902879355.

KARTIKA 2010

Kartika, S. I.: Methodik zur Analyse des Baukastenpotentials bei bestehender Produktvielfalt. Aachen: Shaker 2010. ISBN: 978-3-8322-9114-3.



KEIJZER & KREIMEYER 2006

Keijzer, W.; Kreimeyer, M.: Vernetzungsstrukturen in der Digitalen Fabrik. Status, Trend und Empfehlungen. München: Dr. Hut 2006. ISBN: 9783899633788.

KERZHNER & PAREDIS 2009

Kerzhner, A. A.; Paredis, C. J. J.: Using Domain Specific Languages to Capture Design Synthesis Knowledge for Model-Based Systems Engineering. In: Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2009, August 30- September 2, 2009, San Diego, California, USA, 2009.

KESPER 2012

Kesper, H.: Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden. Dissertation.

Technische Universität München. Lehrstuhl für Produktentwicklung. München: 2012.

KIM & CHHAJED 2001

Kim, K.; Chhajed, D.: An experimental investigation of valuation change due to commonality in vertical product line extension. In: Journal of Product Innovation Management. 18, 2001. 4.

KIPP ET AL. 2010

Kipp, T.; Bles, C.; Krause, D.: Anwendung einer integrierten Methode zur Entwicklung modularer Produktfamilien. In: Tagungsband zum 21. DfX-Symposium, Hamburg, 2010.

KIPP 2012

Kipp, T.: Methodische Unterstützung der variantengerechten Produktgestaltung. Dissertation. Technischen Universität Hamburg-Harburg. PKT. Hamburg: 2012.

KISSEL ET AL. 2011

Kissel, M.; Hellenbrand, D.; Lindemann, U.: A Methodology to Evaluate the Structural Robustness of Product Concepts. In: Proceedings of the International Conference on Engineering Design, Kopenhagen, Dänemark, 2011.

KISSEL ET AL. 2012

Kissel, M.; Schrieverhoff, P.; Lindemann, U.: Design for Adaptability – Identifying Potential for Improvement on an Architecture Basis. In: Proceedings of NordDesign Conference 2012, Aalborg, 2012.

KISSEL & HELMS 2014

Kissel, M.; Helms, B.: Komplexe Produktdaten automatisiert analysieren und visualisieren – Start-up Soley geht an den Start. In: Competence in Design and Development (CiDaD News). 11, 2014. 1.

KISSEL & LINDEMANN 2013

Kissel, M.; Lindemann, U.: System Architecture Change Decisions in Multi-Variant Product Portfolios. In: Proceedings of the International Conference on Engineering Design, Seoul, Korea, 2013.

KOH ET AL. 2012

Koh, E. C. Y.; Caldwell, N. H. M.; Clarkson, P. J.: A method to assess the effects of engineering change propagation. In: Res Eng Design. 23, 2012. 4.

## KÖHLHASE 1997

Kohlhase, N.: Strukturieren und Beurteilen von Baukastensystemen. Strategien, Methoden, Instrumente. Dusseldorf: VDI Verlag 1997. ISBN: 3183275015.

## KOMOTO &amp; TOMIYAMA 2010

Komoto, H.; Tomiyama, T.: Computational Support for System Architecting. In: Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2010, August 15-18, 2010, Montreal, Quebec, Canada, 2010.

## KÖNIGS 2013

Königs, S.: Konzeption und Realisierung einer Methode zur templategestützten Systementwicklung. Dissertation. Technische Universität Berlin. Fakultät V - Verkehrs- und Maschinensysteme. Berlin: 2013.

## KOSSIAKOFF ET AL. 2011

Kossiakoff, A.; Sweet, W. N.; Seymour, S. J.; Biemer, S. M.: Systems engineering. Principles and practice. 2nd ed. Hoboken, N.J: Wiley-Interscience 2011. ISBN: 9780470405482.

## KOTA ET AL. 2000

Kota, S.; Sethuraman, K.; Miller, R.: A Metric for Evaluating Design Commonality in Product Families. In: Journal of Mechanical Design. 122, 2000.

## KRATOCHVIL &amp; CARSON 2005

Kratochvil, M.; Carson, C.: Growing modular. Mass customization of complex products, services and software. Berlin: Springer 2005. ISBN: 9783540239598.

## KRAUSE &amp; RIPPERDA 2013

Krause, D.; Ripperda, S.: An assessment of methodical approaches to support the development of modular product families. In: Proceedings of the International Conference on Engineering Design, Seoul, Korea, 2013.

## KRAUSE ET AL. 2014

Krause, D.; Beckmann, G.; Eilmus, S.; Gebhardt, N.; Jonas, H.; Rettberg, R., 2014 Integrated Development of Modular Product Families: A Method Toolkit. In: Timothy W. Simpson (Hrsg.): Advances in product family and product platform design. Methods & applications: Springer. ISBN: 9781461479369.

## KRAUSE ET AL. 2013

Krause, D.; Eilmus, S.; Jonas, H., 2013 Developing Modular Product Families with Perspectives for the Product Program. In: Michael Abramovici und Rainer Stark (Hrsg.): Smart Product Engineering. Proceedings of the 23rd CIRP Design Conference, Bochum, Germany, March 11th - 13th, 2013. Berlin, Heidelberg: Springer. ISBN: 978-3-642-30816-1.

## KREIMEYER ET AL. 2006

Kreimeyer, M.; Herfeld, U.; Deubzer, F.; Dequidt, C.; Lindemann, U.: Function-driven Product Design in Virtual Teams through Methodical Structuring of Requirements and Components. In: Proceedings of the 8th Biennial Conference on Engineering Systems Design and Analysis (ASME-ESDA06), Torino, 2006.

KREIMEYER & LINDEMANN 2011

Kreimeyer, M.; Lindemann, U.: Complexity metrics in engineering design. Managing the structure of design processes. New York: Springer 2011. ISBN: 9783642209628.

KRISHNAN & ULRICH 2001

Krishnan, V.; Ulrich, K. T.: Product Development Decisions: A Review of the Literature. In: Management Science. 47, 2001. 1.

KUSIAK 2000

Kusiak, A.: Computational intelligence in design and manufacturing. New York: John Wiley & Sons 2000. ISBN: 9780471348795.

KÜSTNER ET AL. 2013

Küstner, C.; Breitsprecher, T.; Wartzack, S.: Die Auswirkung der Reihenfolge von Mess- und Simulationsdaten auf das Ergebnis der Kreuzvalidierung in KDD Prozessen. In: Tagungsband zum 24. DfX-Symposium, Hamburg, 2013.

KVIST 2010

Kvist, M.: Product Family Assessment. Dissertation. Danmarks Tekniske Universitet. DTU Management. Kopenhagen: 2010.

LAUX ET AL. 2012

Laux, H.; Gillenkirch, R. M.; Schenk-Mathes, H. Y.: Entscheidungstheorie. 8., erw. und vollst. überarb. Aufl. Berlin, Heidelberg: Springer Gabler 2012. ISBN: 3642235107.

LI ET AL. 2008

Li, Y.; Xue, D.; Gu, P.: Design for Product Adaptability. In: Concurrent Engineering. 16, 2008. 3.

LI ET AL. 2013

Li, Z.; Cheng, Z.; Feng, Y.; Yang, J.: An integrated method for flexible platform modular architecture design. In: Journal of Engineering Design. 24, 2013. 1.

LINDEMANN 2007

Lindemann, U.: Methodische Entwicklung technischer Produkte. Methoden flexibel und situationsgerecht anwenden. 2., bearb. Aufl. Berlin: Springer 2007. ISBN: 3-540-37435-3.

LINDEMANN ET AL. 2009

Lindemann, U.; Maurer, M.; Braun, T.: Structural complexity management. An approach for the field of product design. Berlin: Springer 2009. ISBN: 3540878882.

LINDEMANN ET AL. 2006

Lindemann, U.; Reichwald, R.; Zäh, M. F.: Individualisierte Produkte-Komplexität beherrschen in Entwicklung und Produktion. 1. Aufl. Berlin: Springer 2006. ISBN: 9781280616426.

LINGNAU 1994

Lingnau, V.: Variantenmanagement - Produktionsplanung im Rahmen einer Produktdifferenzierungsstrategie. Berlin: Erich Schmidt Verlag 1994. ISBN: 3503036199.

LOCK 2013

Lock, C.: Konzept zur Unterstützung von Entscheidungen mit Einfluss auf die

Systemarchitektur. Diplomarbeit.

Technische Universität München. Lehrstuhl für Produktentwicklung. München: 2013.

LUHMANN 1991

Luhmann, N.: Soziale Systeme - Grundriss einer allgemeinen Theorie. 4. Auflage. Suhrkamp 1991. ISBN: 3518282662.

MAIER 1993

Maier, T.: Gleichteileanalyse und Ähnlichkeitsermittlung von Produktprogrammen. Stuttgart: Univ., IMK, Forschungs- und Lehrgebiet Technisches Design 1993. ISBN: 3922823238.

MALMQVIST 2002

Malmqvist, J.: A Classification of Matrix-based Methods for Product Modeling. In: Proceedings of the International Design Conference, Dubrovnik, 2002.

MARTIN & ISHII 1997

Martin, M.; Ishii, K.: Design for variety: development of complexity indices and design charts. In: Proceedings of the ASME Design Engineering Technical Conference, Sacramento, USA, 1997.

MARTIN & ISHII 2002

Martin, M. V.; Ishii, K.: Design for variety: developing standardized and modularized product platform architectures. In: Research in Engineering Design. 13, 2002.

MAURER 2007

Maurer, M. S.: Structural Awareness in Complex Product Design. Dissertation. Technische Universität München. Lehrstuhl für Produktentwicklung. München: 2007.

MEYER & LEHNERD 1997

Meyer, M. H.; Lehnerd, A. P.: The power of product platforms: building value and cost leadership. New York: The Free Press 1997. ISBN: 0684825805.

MILES & HUBERMAN 1994

Miles, M. B.; Huberman, A. M.: Qualitative data analysis. An expanded sourcebook. 2nd ed. Thousand Oaks: Sage Publications 1994. ISBN: 9780803955400.

MÜLLER ET AL. 2013

Müller, P.; Pasch, F.; Drewinski, R.; Hayka, H.; Bedenbender, H.: Kollaborative Produktentwicklung und digitale Werkzeuge - Defizite heute, Potentiale morgen. 2013. ISBN: 9783000391118.

NASA 2007

NASA: NASA Systems Engineering Handbook. Revision 1. Hanover, MD: U.S. Government Printing Office 2007.

NEWCOMB ET AL. 1998

Newcomb, P. J.; Bras, B.; Rosen, D. W.: Implications of Modularity on Product Design for the Life Cycle. In: J. Mech. Des. 120, 1998. 3.

NURCAHYA 2009

Nurcahya, E.: Ein Produktdatenmodell für das rechnerunterstützte Variantenmanagement.

Dissertation.

RWTH. Aachen: 2009.

PAHL ET AL. 2007

Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H.: Pahl/Beitz Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung. 7. Auflage. Berlin, Heidelberg, New York: Springer 2007. ISBN: 978-3540340607.

PARTNER ET AL. 2013

Partner, J.; Vukotic, A.; Watt, N.: Neo4J in Action. O'Reilly Media 2013. ISBN: 9781617290763.

PATZAK 1982

Patzak, G.: Systemtechnik - Planung komplexer innovativer Systeme. Grundlagen, Methoden, Techniken. Berlin, Heidelberg, New York: Springer 1982. ISBN: 3540117830.

PILLER 1998

Piller, F. T.: Kundenindividuelle Massenproduktion. Die Wettbewerbsstrategie der Zukunft. München, Wien: Hanser 1998. ISBN: 9783446193369.

PILLER & WARINGER 1999

Piller, F. T.; Waringer, D.: Modularisierung in der Automobilindustrie - neue Formen und Prinzipien. Modular Sourcing, Plattformkonzept und Fertigungssegmentierung als Mittel des Komplexitätsmanagement. Aachen: Shaker 1999. ISBN: 9783826558276.

PIMMLER & EPPINGER 1994

Pimmler, T. U.; Eppinger, S. D.: Integration analysis of product decompositions. In: ASME Design Theory and Methodology Conference, Minneapolis, 1994. September.

PINE, II. 1993

Pine, B. J., II.: Mass Customization. The New Frontier in Business Competition. Boston: Harvard Business School Press 1993. ISBN: 0-87584-372-7.

PONN & LINDEMANN 2011

Ponn, J.; Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte. Systematisch von Anforderungen zu Konzepten und Gestaltungsformen. 2. Aufl. Berlin, Heidelberg: Springer 2011. ISBN: 3642205801.

PRASAD 1998

Prasad, B.: Designing products for variety and how to manage complexity. In: Journal of Product & Brand Management. 7, 1998. 3.

PULM 2004

Pulm, U.: Eine systemtheoretische Betrachtung der Produktentwicklung. Technische Universität München. Lehrstuhl für Produktentwicklung. München: 2004.

PUPPE 1988

Puppe, F.: Einführung in Expertensysteme. Berlin [u.a.]: Springer 1988. ISBN: 9780387194813.

RAPP 2010

Rapp, T.: Produktstrukturierung. Komplexitätsmanagement durch modulare

Produktstrukturen und -plattformen. 2. Aufl. Norderstedt: Books on Demand 2010. ISBN: 9783839136041.

RATHNOW 1993

Rathnow, P. J.: Integriertes Variantenmanagement. Bestimmung, Realisierung und Sicherung der optimalen Produktvielfalt. Göttingen: Vandenhoeck & Ruprecht 1993. ISBN: 9783525125694.

RENNER 2007

Renner, I.: Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil. Dissertation.

Technische Universität München. Lehrstuhl für Produktentwicklung. München: 2007.

REY & WENDER 2008

Rey, G. D.; Wender, K. F.: Neuronale Netze. Eine Einführung in die Grundlagen, Anwendungen und Datenauswertung. 1. Aufl. Bern: Huber 2008. ISBN: 9783456845135.

ROBERTSON & ULRICH 1998

Robertson, D.; Ulrich, K. T.: Planning for Product Platforms. In: Sloan Management Review. 39, 1998. 4.

ROBINSON ET AL. 2013

Robinson, I.; Webber, J.; Efrem, E.: Graph databases. Sebastopol, Calif: O'Reilly Media 2013. ISBN: 9781449356262.

ROPOHL 1975

Ropohl, G.: Systemtechnik, Grundlagen und Anwendung. München, Wien: Hanser 1975. ISBN: 3446118292.

ROSS ET AL. 2008

Ross, A. M.; Rhodes, D. H.; Hastings, D. E.: Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. In: Syst. Engin. 11, 2008. 3.

ROUSE & SAGE 2009

Rouse, W. B.; Sage, A. P.: Handbook of systems engineering and management. 2. ed. Hoboken, NJ: Wiley 2009. ISBN: 9780470083536.

ROZENBERG & EHRIG 1997

Rozenberg, G.; Ehrig, H.: Handbook of graph grammars and computing by graph transformation. Volume 1 - Foundations. Singapore: World Scientific 1997. ISBN: 9789810228842.

RUDE 1998

Rude, S.: Wissensbasiertes Konstruieren. Aachen: Shaker 1998. ISBN: 3-8265-3985-0.

SANCHEZ & MAHONEY 1996

Sanchez, R.; Mahoney, J. T.: Modularity, flexibility, and Knowledge Management in Product and Organization Design. In: Strategic Management Journal. 17, 1996. Winter Special Issue.

SCHMIDT 2009

Schmidt, S.: Die Diffusion komplexer Produkte und Systeme. Ein Systemdynamischer Ansatz. Wiesbaden: Gabler 2009. ISBN: 9783834914460.

SCHÖNSLEBEN 2011

Schönsleben, P.: Integrales Logistikmanagement. Operations und Supply Chain Management innerhalb des Unternehmens und unternehmensübergreifend. In: Integrales Logistikmanagement, 2011.

SCHRIEVERHOFF ET AL. 2012

Schrieverhoff, P.; Bender, T.; Kissel, M.; Lindemann, U.: Evaluation of Architecture Options in Systems Engineering. In: Proceedings of the International Design Conference, Dubrovnik, 2012.

SCHUH 1988

Schuh, G.: Gestaltung und Bewertung von Produktvarianten. Ein Beitrag zur systematischen Planung von Serienprodukten. Dissertation. RWTH. Aachen: 1988.

SCHUH 2005

Schuh, G.: Produktkomplexität managen - Strategien, Methoden, Tools. 2.Auflage. München: Hanser 2005. ISBN: 3-446-40043-5.

SIDDIQUE & BODDU 2005

Siddique, Z.; Boddu, K. R.: A mass customization information framework for integration of customer in the configuration/design of a customized product. In: Ai Edam. 18, 2005. 01.

SIMON 1962

Simon, H. A.: The Architecture of Complexity. In: Proceedings of the American Philosophical Society. 106, 1962. 6.

SIMPSON 2004

Simpson, T. W.: Product platform design and customization: Status and promise. In: AIEDAM. 18, 2004. 01.

SIMPSON & D'SOUZA 2004

Simpson, T. W.; D'Souza, B. S.: Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multiobjective Genetic Algorithm. In: Concurrent Engineering. 12, 2004. 2.

SOSA ET AL. 2004

Sosa, M. E.; Eppinger, S. D.; Rowles, C. M.: The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. In: Management Science. 50, 2004. 12.

STAKE 2000

Stake, R.: On conceptual development of modular products. Development of supporting tools for the modularisation process. Dissertation. Royal Institute of Technology. Stockholm: 2000.

STEFIK ET AL. 1982

Stefik, M.; Aikins, J.; Balzer, R.; Benoit, J.; Birnbaum, L.; Hayes-Roth, F.; Sacerdoti, E.: The Organization of Expert Systems, a Tutorial. In: Artificial Intelligence. 18, 1982.

STEWART 1981

Steward, D. V.: The Design Structure System - A Method for Managing the Design of Complex Systems. In: IEEE Transactions on Engineering Management. 28, 1981. 3.

SUH 2001

Suh, N. P.: Axiomatic design. Advances and applications. New York, NY: Oxford Univ. Press 2001. ISBN: 9780195134667.

TARASEWICH & NAIR 2001

Tarasewich, P.; Nair, S. K.: Designer-moderated product design. In: IEEE Trans. Eng. Manage. 48, 2001. 2.

THEVENOT & SIMPSON 2006

Thevenot, H. J.; Simpson, T. W.: Commonality indices for product family design: a detailed comparison. In: Journal of Engineering Design. 17, 2006. 2.

TITTMANN 2003

Tittmann, P.: Graphentheorie. Eine anwendungsorientierte Einführung ; mit zahlreichen Beispielen und 80 Aufgaben. München, Wien: Fachbuchverl. Leipzig im Carl-Hanser-Verl. 2003. ISBN: 9783446223431.

ULRICH 1995

Ulrich, K.: The role of product architecture in the manufacturing firm. In: Research Policy. 24, 1995. 3.

UMEDA ET AL. 1996

Umeda, Y.; Ishii, M.; Yoshioka, M.; Shimomura, Y.; Tomiyama, T.: Supporting conceptual design based on the function-behavior-state modeler. In: AIEDAM. 10, 1996. 04.

VAJNA ET AL. 2009

Vajna, S.; Weber, C.; Bley, H.; Zeman, K.: CAX für Ingenieure. Eine praxisbezogene Einführung. 2., völlig neu bearb. Aufl. Berlin, Heidelberg: Springer 2009. ISBN: 9783540360384.

VASANTHA ET AL. 2012

Vasantha, G. V. A.; Roy, R.; Lelah, A.; Brissaud, D.: A review of product-service systems design methodologies. In: Journal of Engineering Design. 23, 2012. 9.

VDI 2004

VDI: VDI 2206-2004. Entwicklungsmethodik für mechatronische Systeme. Düsseldorf: Verein Deutscher Ingenieure 2004.

VESTER 2000

Vester, F.: Die Kunst vernetzt zu denken. Ideen und Werkzeuge für einen neuen Umgang mit Komplexität. 6., durchges. und überarb. Aufl. Stuttgart: Dt. Verl.-Anst 2000. ISBN: 3421053081.



Volkswagen 2014

Volkswagen: Modulare Baukastenstrategie. 2014.

URL:

[http://www.volkswagenag.com/content/vwcorp/content/de/investor\\_relations/Warum\\_Volkswagen/MQB.html/](http://www.volkswagenag.com/content/vwcorp/content/de/investor_relations/Warum_Volkswagen/MQB.html/) - zuletzt geprüft am: 02.01.2014.

WEAVER 1948

Weaver, W.: Science and Complexity. In: American Scientist. 36, 1948. 4.

WEILKIENS 2008

Weilkiens, T.: Systems engineering mit SysML/UML. Modellierung, Analyse, Design. 2., aktualisierte u. erw. Aufl. Heidelberg: Dpunkt-Verl. 2008. ISBN: 3898645770.

WEINBERG 1975

Weinberg, G. M.: An introduction to general systems thinking. Gerald M. Weinberg. Silver anniversary ed. New York: Wiley 1975. ISBN: 0471925632.

WILDEMANN 1997

Wildemann, H.: Fertigungsstrategien. Reorganisationskonzepte für eine schlanke Produktion und Zulieferung. 3., überarb. Aufl. München: Transfer-Centrum-Verl. 1997. ISBN: 9783929918892.

WILDEMANN 2013

Wildemann, H.: Komplexitätsmanagement in Vertrieb, Beschaffung, Produkt, Entwicklung und Produktion. 14. Aufl. München: TCW Transfer-Centrum 2013. ISBN: 978-3-931511-30-2.

WITTEN ET AL. 2011

Witten, I. H.; Frank, E.; Hall, M. A.: Data mining. Practical machine learning tools and techniques. 3rd ed. Burlington, MA: Morgan Kaufmann 2011. ISBN: 9780123748560.

YIN 2013

Yin, R. K.: Case Study Research. Design and Methods. 5. ed. Los Angeles, Calif. [u.a.]: Sage Publ. 2013. ISBN: 9781452242569.

ZAGEL 2010

Zagel, M.: Übergreifendes Konzept zur Strukturierung variantenreicher Produkte und Vorgehensweise zur iterativen Produktstruktur-Optimierung. 2. Aufl. Kaiserslautern: Techn. Univ. Kaiserslautern 2010. ISBN: 9783939432265.

ZICH 1996

Zich, C.: Integrierte Typen- und Teileoptimierung. Neue Methoden des Produktprogramm-Managements. Wiesbaden, Wiesbaden: Dt. Univ.-Verl.; Gabler 1996. ISBN: 9783824463381.



## 10. Anhang

### 10.1 Metamodell

```
//Abstrakte Knoten

abstract node class Visueller_Knoten {
    x:double;
    y:double;
    label:string;
    color:string;
    height:double;
    width:double;
    diameter:double;
    shape:string;
}

abstract node class Portfolio_Element extends Visueller_Knoten {
    id:int;
    name:string;
    description:string;
    time_stamp:string;
    type_name:string;
}

abstract node class Produktstruktur_Element extends Portfolio_Element {
    status:string;
    revenue:double;
}

// Knoten des Produktportfolios und der Produktkonkretisierung

    node class Anforderung extends Produktstruktur_Element {
        requirement_class:string;}
    node class Funktion extends Produktstruktur_Element {
        function_class:string;}
    node class Testfall extends Produktstruktur_Element;

    node class ProduktPortfolio extends Produktstruktur_Element;
    node class ProduktFamilie extends Produktstruktur_Element;
node class Produkt extends Produktstruktur_Element {
    materialnumber:int;
    price:double;
    gross_margin:double;
    quantity:double;
    weight:double;
```

```

    }
    node class Baugruppe extends Produktstruktur_Element {
        materialnumber:int;
        building_block_class:string;
        weight:double;
    }
    node class Teil extends Produktstruktur_Element {
        materialnumber:int;
        building_block_class:string;
        weight:double;
        quantity:double;
    }
    node class Variant extends Produkt, Baugruppe, Teil{
        variantclass:string;
    }

// Stakeholder Knoten
abstract node class Stakeholder extends Portfolio_Element {
    classification:string;
}

    node class Markt extends Stakeholder {
        region:string;
        turnover:double;
        slToolTip = "Markt";}
    node class Kunde extends Stakeholder{
        turnover:double;
        slToolTip = "Kunde";}
    node class Gesetzgeber extends Stakeholder{
        slToolTip = "Gesetzgeber";}
    node class Entwickler extends Stakeholder{
        stakeholder_level:string;
        slToolTip = "Entwickler";}
    node class Lieferant extends Stakeholder{
        strategic_level:string;
        slToolTip = "Lieferant";}
    node class Produktion extends Stakeholder{
        slToolTip = "Produktion";}
    node class Prozessverantwortlicher extends Stakeholder{
        slToolTip = "Prozessverantwortlicher";}
    node class Projektleiter extends Stakeholder{
        slToolTip = "Projektleiter";}

// Gruppen Knoten
abstract node class Logische_Gruppe extends Portfolio_Element;

```

```
node class Technologie extends Logische_Gruppe {
    slToolTip = "Technologie";}
node class Bausteinklasse extends Logische_Gruppe {
    building_block_class:string;
    slToolTip = "Bausteinklasse";}
node class Schnittstelle extends Logische_Gruppe {
    interface_classification:string;
    slToolTip = "Schnittstelle";}
node class Änderung extends Logische_Gruppe {
    change_classification:string;
    slToolTip = "Änderung";}
node class logisches_Modul extends Logische_Gruppe {
    slToolTip = "Logisches Modul";}

// Kanten
abstract edge class Portfolio_Kante {
    name:string;
    weight:int;
    color:string;
    thickness:double;
}

edge class Contains extends Portfolio_Kante;
edge class ConsistsOf extends Portfolio_Kante;
edge class Defines extends Portfolio_Kante;
edge class Satisfies extends Portfolio_Kante;
edge class Fulfills extends Portfolio_Kante;
edge class Verifies extends Portfolio_Kante;
edge class MaterialFlow extends Portfolio_Kante;
edge class EnergyFlow extends Portfolio_Kante;
edge class InformationFlow extends Portfolio_Kante;
edge class IsConnectedTo extends Portfolio_Kante;
```

## 10.2 GrGen.NET

<code>s ;&gt; t</code>	Execute <code>s</code> then <code>t</code> . Success if <code>t</code> succeeded.
<code>s &lt;; t</code>	Execute <code>s</code> then <code>t</code> . Success if <code>s</code> succeeded.
<code>s   t</code>	Execute <code>s</code> then <code>t</code> . Success if <code>s</code> or <code>t</code> succeeded.
<code>s    t</code>	The same as <code>s   t</code> but with lazy evaluation, i.e. if <code>s</code> is successful, <code>t</code> will not be executed.
<code>s &amp; t</code>	Execute <code>s</code> then <code>t</code> . Success if <code>s</code> and <code>t</code> succeeded.
<code>s &amp;&amp; t</code>	The same as <code>s &amp; t</code> but with lazy evaluation, i.e. if <code>s</code> fails, <code>t</code> will not be executed.
<code>s ^ t</code>	Execute <code>s</code> then <code>t</code> . Success if <code>s</code> or <code>t</code> succeeded, but not both.
<code>if{r;s;t}</code>	Execute <code>r</code> . If <code>r</code> succeeded, execute <code>s</code> and return the result of <code>s</code> . Otherwise execute <code>t</code> and return the result of <code>t</code> .
<code>if{r;s}</code>	Same as <code>if{r;s;true}</code>
<code>!s</code>	Switch the result of <code>s</code> from successful to fail and vice versa.
<code>\$&lt;op&gt;</code>	Use random instead of left-associative execution order for <code>&lt;op&gt;</code> .
<code>s*</code>	Execute <code>s</code> repeatedly as long as its execution does not fail.
<code>s+</code>	Same as <code>s &amp;&amp; s*</code> .
<code>s[n]</code>	Execute <code>s</code> repeatedly as long as its execution does not fail but <code>n</code> times at most.
<code>s[m:n]</code>	Same as <code>s[n]</code> but fails if executed less than <code>m</code> times.
<code>s[m:*]</code>	Same as <code>s*</code> but fails if executed less than <code>m</code> times.
<code>?Rule</code>	Switches <code>Rule</code> to a test.
<code>%Rule</code>	This is the multi-purpose flag when accessed from LIBGR. Also used for graph dumping and break points.
<code>[Rule]</code>	Rewrite every pattern match produced by the action <code>Rule</code> .
<code>count[a]=&gt;v</code>	Rewrite every pattern match produced by the action <code>a</code> , and write the count of the matches found to the variable <code>v</code> .
<code>true</code>	A constant acting as a successful match.
<code>false</code>	A constant acting as a failed match.
<code>v</code>	A variable acting as a predicate.

Let `r, s, t` be sequences, `u, v, w` variable identifiers, `<op> ∈ {!, ^, &, ||, &&}`

Abbildung 10-1: Übersicht über Operatoren in Regelsequenzen (BLOMER ET AL. 2013, S. 98)

# 11. 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 BIERSACK, H.:  
Methode für Krafteinleitungsstellenkonstruktion 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 Hilfsmittelleinführung 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: Dr. Hut 2006. (Produktentwicklung München, Band 58). Zugl. 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.
- D87 JUNG, C.:  
Anforderungskklärung in interdisziplinärer Entwicklungsumgebung.  
München: Dr. Hut 2006. (Produktentwicklung München, Band 61). Zugl. München: TU, Diss. 2006.
- D88 HEBLING, T.:  
Einführung der Integrierten Produktpolitik in kleinen und mittelständischen Unternehmen.  
München: Dr. Hut 2006. (Produktentwicklung München, Band 62). Zugl. München: TU, Diss. 2006.
- D89 STRICKER, H.:  
Bionik in der Produktentwicklung unter der Berücksichtigung menschlichen Verhaltens.  
München: Dr. Hut 2006. (Produktentwicklung München, Band 63). Zugl. München: TU, Diss. 2006.
- D90 NIBL, A.:  
Modell zur Integration der Zielkostenverfolgung in den Produktentwicklungsprozess.  
München: Dr. Hut 2006. (Produktentwicklung München, Band 64). Zugl. München: TU, Diss. 2006.
- D91 MÜLLER, F.:  
Intuitive digitale Geometriemodellierung in frühen Entwicklungsphasen.  
München: Dr. Hut 2007. (Produktentwicklung München, Band 65). Zugl. München: TU, Diss. 2006.
- D92 ERDELL, E.:  
Methodenanwendung in der Hochbauplanung – Ergebnisse einer Schwachstellenanalyse.  
München: Dr. Hut 2006. (Produktentwicklung München, Band 66). Zugl. München: TU, Diss. 2006.
- D93 GAHR, A.:  
Pfadkostenrechnung individualisierter Produkte.  
München: Dr. Hut 2006. (Produktentwicklung München, Band 67). Zugl. München: TU, Diss. 2006.
- D94 RENNER, I.:  
Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil.  
München: Dr. Hut 2007 (Reihe Produktentwicklung) Zugl. München: TU, Diss. 2007.
- D95 PONN, J.:  
Situative Unterstützung der methodischen Konzeptentwicklung technischer Produkte.  
München: Dr. Hut 2007 (Reihe Produktentwicklung) Zugl. München: TU, Diss. 2007.
- D96 HERFELD, U.:  
Matrix-basierte Verknüpfung von Komponenten und Funktionen zur Integration von Konstruktion und numerischer Simulation.  
München: Dr. Hut 2007. (Produktentwicklung München, Band 70). Zugl. München: TU, Diss. 2007.
- D97 SCHNEIDER, S.:  
Model for the evaluation of engineering design methods.  
München: Dr. Hut 2008 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D98 FELGEN, L.:  
Systemorientierte Qualitätssicherung für mechatronische Produkte.  
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D99 GRIEB, J.:  
Auswahl von Werkzeugen und Methoden für verteilte Produktentwicklungsprozesse.  
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.

- D100 MAURER, M.:  
Structural Awareness in Complex Product Design.  
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D101 BAUMBERGER, C.:  
Methoden zur kundenspezifischen Produktdefinition bei individualisierten Produkten.  
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D102 KEIJZER, W.:  
Wandlungsfähigkeit von Entwicklungsnetzwerken – ein Modell am Beispiel der Automobilindustrie.  
München: Dr. Hut 2007 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2007.
- D103 LORENZ, M.:  
Handling of Strategic Uncertainties in Integrated Product Development.  
München: Dr. Hut 2009 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2008.
- D104 KREIMEYER, M.:  
Structural Measurement System for Engineering Design Processes.  
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2009.
- D105 DIEHL, H.:  
Systemorientierte Visualisierung disziplinübergreifender Entwicklungsabhängigkeiten mechatronischer Automobilsysteme.  
München: Dr. Hut 2009 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2009.
- D106 DICK, B.:  
Untersuchung und Modell zur Beschreibung des Einsatzes bildlicher Produktmodelle durch Entwicklerteams in der Lösungssuche.  
München: Dr. Hut 2009 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2009.
- D107 GAAG, A.:  
Entwicklung einer Ontologie zur funktionsorientierten Lösungssuche in der Produktentwicklung.  
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2010.
- D108 ZIRKLER, S.:  
Transdisziplinäres Zielkostenmanagement komplexer mechatronischer Produkte.  
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2010.
- D109 LAUER, W.:  
Integrative Dokumenten- und Prozessbeschreibung in dynamischen Produktentwicklungsprozessen.  
München: Dr. Hut 2010 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2010.
- D110 MEIWALD, T.:  
Konzepte zum Schutz vor Produktpiraterie und unerwünschtem Know-how-Abfluss.  
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D111 ROELOFSEN, J.:  
Situationsspezifische Planung von Produktentwicklungsprozessen.  
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D112 PETERMANN, M.:  
Schutz von Technologiewissen in der Investitionsgüterindustrie.  
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D113 GORBEA, C.:  
Vehicle Architecture and Lifecycle Cost Analysis in a New Age of Architectural Competition.  
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D114 FILOUS, M.:  
Lizenzierungsgerechte Produktentwicklung – Ein Leitfaden zur Integration lizenzierungsrelevanter Aktivitäten in Produktentstehungsprozessen des Maschinen- und Anlagenbaus.  
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.

- D115 ANTON, T.:  
Entwicklungs- und Einführungsmethodik für das Projektierungswerkzeug Pneumatiksimulation.  
München: Dr. Hut 2011 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2011.
- D116 KESPER, H.:  
Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden.  
München: Dr. Hut 2012 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2012.
- D117 KIRSCHNER, R.:  
Methodische Offene Produktentwicklung.  
München: TU, Diss. 2012.
- D118 HEPPERLE, C.:  
Planung lebenszyklusgerechter Leistungsbündel.  
München: Dr. Hut 2013 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2013.
- D119 HELLENBRAND, D.:  
Transdisziplinäre Planung und Synchronisation mechatronischer Produktentwicklungsprozesse.  
München: Dr. Hut 2013 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2013.
- D120 EBERL, T.:  
Charakterisierung und Gestaltung des Fahr-Erlebens der Längsführung von Elektrofahrzeugen.  
München: TU, Diss. 2014.
- D121 KAIN, A.:  
Methodik zur Umsetzung der Offenen Produktentwicklung.  
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D122 ILIE, D.:  
Systematisiertes Ziele- und Anforderungsmanagement in der Fahrzeugentwicklung.  
München: Dr. Hut 2013 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2013.
- D123 HELTEN, K.:  
Einführung von Lean Development in mittelständische Unternehmen - Beschreibung, Erklärungsansatz  
und Handlungsempfehlungen.  
TU München: 2013. (als Dissertation eingereicht)
- D124 SCHRÖER, B.:  
Lösungskomponente Mensch. Nutzerseitige Handlungsmöglichkeiten als Bausteine für die kreative  
Entwicklung von Interaktionslösungen.  
TU München: 2013. (als Dissertation eingereicht)
- D125 KORTLER, S.:  
Absicherung von Eigenschaften komplexer und variantenreicher Produkte in der Produktentwicklung.  
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D126 KOHN, A.:  
Entwicklung einer Wissensbasis für die Arbeit mit Produktmodellen.  
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D127 FRANKE, S.:  
Strategieorientierte Vorentwicklung komplexer Produkte – Prozesse und Methoden zur zielgerichteten  
Komponentenentwicklung am Beispiel Pkw.  
Göttingen: Cuvillier, E 2014.
- D128 HOOSMAND, A.:  
Solving Engineering Design Problems through a Combination of Generative Grammars and Simulations.  
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D129 KISSEL, M.:  
Mustererkennung in komplexen Produktportfolios.  
TU München: 2014. (als Dissertation eingereicht)

- D130 NIES, B.:  
Nutzungsgerechte Dimensionierung des elektrischen Antriebssystems für Plug-In Hybride.  
München: TU, Diss. 2014.
- D131 KIRNER, K.:  
Zusammenhang zwischen Leistung in der Produktentwicklung und Variantenmanagement –  
Einflussmodell und Analyseverfahren.  
München: Dr. Hut 2014 (Reihe Produktentwicklung). Zugl. München: TU, Diss. 2014.
- D132 BIEDERMANN, W.:  
A minimal set of network metrics for analysing mechatronic product concepts.  
TU München: 2014. (als Dissertation eingereicht)
- D133 SCHENKL, S.:  
Wissensorientierte Entwicklung von Produkt-Service-Systemen.  
TU München: 2014. (als Dissertation eingereicht)
- D134 SCHRIEVERHOFF, P.:  
Valuation of Adaptability in System Architecture.  
TU München: 2014. (als Dissertation eingereicht)
- D135 METZLER, T.:  
Models and Methods for the Systematic Integration of Cognitive Functions into Product Concepts.  
TU München: 2014. (als Dissertation eingereicht)