

Jonas Andreas Powelske

## **MBSE-gestützte Methoden zur Strukturierung und Anwendung von Baukästen in der Frühen Phase der PGE – Produktgenerationsentwicklung mechatronischer Steuergeräte im Einsatz in Kleinantrieben im Automobilbereich**

MBSE-supported methods for Structuring and Application of Modular Kits in the Early Stages of PGE – Product Generation Engineering of Mechatronic Control Units in Small Drives in the Automotive Sector

**Band 164**

Systeme ■ Methoden ■ Prozesse

Univ.-Prof. Dr.-Ing. Dr. h.c. A. Albers  
Univ.-Prof. Dr.-Ing. S. Matthiesen  
(Hrsg.)

Copyright IPEK - Institut für Produktentwicklung, 2023  
Karlsruher Institut für Technologie (KIT)  
Die Forschungsuniversität in der Helmholtz-Gemeinschaft  
Alle Rechte vorbehalten

ISSN 1615-8113

**MBSE-gestützte Methoden zur Strukturierung und  
Anwendung von Baukästen in der Frühen Phase der  
PGE – Produktgenerationsentwicklung  
mechatronischer Steuergeräte im Einsatz in  
Kleinantrieben im Automobilbereich**

Zur Erlangung des akademischen Grades eines  
**DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)**  
von der KIT-Fakultät für Maschinenbau des  
Karlsruher Instituts für Technologie (KIT)

angenommene

**DISSERTATION**

von

M.Sc. Jonas Andreas Powelske

Tag der mündlichen Prüfung:	23.02.2023
Hauptreferent:	Univ.-Prof. Dr.-Ing. Dr. h. c. Albert Albers
Korreferent:	Univ.-Prof. Dr.-Ing. Nikola Bursac





# Vorwort der Herausgeber

Wissen ist einer der entscheidenden Faktoren in den Volkswirtschaften unserer Zeit. Der Unternehmenserfolg wird mehr denn je davon abhängen, wie schnell ein Unternehmen neues Wissen aufnehmen, zugänglich machen und verwerten kann. Die Aufgabe eines Universitätsinstitutes ist es, hier einen wesentlichen Beitrag zu leisten. In den Forschungsarbeiten wird ständig Wissen generiert. Dieses kann aber nur wirksam und für die Gemeinschaft nutzbar werden, wenn es in geeigneter Form kommuniziert wird. Diese Schriftenreihe dient seit mehr als 20 Jahren als eine Plattform zum Transfer und macht damit das Wissenspotenzial aus aktuellen Forschungsarbeiten am IPEK - Institut für Produktentwicklung Karlsruhe\* am Karlsruher Institut für Technologie (KIT) verfügbar. Die Forschung des IPEK ist dabei strukturiert in die Kategorien Systeme, Methoden und Prozesse, um so der Komplexität heutiger Produktentwicklung ganzheitlich gerecht zu werden. Erst die Verknüpfung dieser drei Kategorien ermöglicht die Synthese innovativer Systeme durch Nutzung neuester Methoden und Prozesse. Gleichzeitig werden durch die Systemsynthese die erforschten neuen Methoden und Prozesse validiert und deren Mehrwert für die Praxis abgesichert. Dieses Forschungskonzept prägt nicht nur das IPEK-Leitbild, sondern auch den Charakter dieser Schriftenreihe, da immer alle drei Kategorien und deren Wechselwirkungen berücksichtigt werden. Jeder Band setzt hier individuelle Schwerpunkte und adressiert dabei folgende Forschungsgebiete des IPEK:

- das Entwicklungs- und Innovationsmanagement,
- die Entwicklungs- und Konstruktionsmethodik,
- der Leichtbau von der Ebene des ganzen Systems bis hinunter zur Optimierung des Bauteils,
- die Validierung technischer Systeme auch unter Berücksichtigung der NVH Aspekte (Noise, Vibration, Harshness) mit dem Fokus auf Schwingungen und Akustik an Komponenten und in den Gesamtsystemen sowie deren subjektiver Beurteilung durch den Menschen,
- die Antriebssystemtechnik mit den Schwerpunkten komplette Antriebslösungen für Fahrzeuge und Maschinen,
- das Design, die Tribologie und Erprobung von Kupplungen und Bremsen sowie
- die Gerätetechnik mit dem Schwerpunkt auf Power-Tools.

Die Forschungsberichte stellen Ergebnisse unserer Forschung sowohl anderen Wissenschaftlern als auch den Unternehmen zu Verfügung, um damit die Produktentwicklung in allen ihren Facetten mit innovativen Impulsen zu optimieren.

Albert Albers und Sven Matthiesen

\* Eh.: Institut für Maschinenkonstruktionslehre und Kraftfahrzeugbau, Universität Karlsruhe (TH)



## Vorwort zu Band 164

Die Aufgabe der Produktentwicklung ist es, zukünftige Kunden- und Anwenderbedürfnisse zu erkennen, vorherzusagen und mit neuen Produktgenerationslösungen zu antizipieren, um so erfolgreich im Markt die eigene Position im Wettbewerb durch Innovationen zu stärken. Diese grundlegende Aufgabenstellung realisieren Produktentwickelnde in allen Bereichen technischer Maschinen, Apparate und Fahrzeuge. Der permanente kreative Wettbewerb um die Innovationen hat dazu geführt, dass – zunächst beginnend im Bereich mechanischer Lösungen – vielfältige Produktlösungen entwickelt und in den Märkten etabliert wurden. Spätestens seit Ende des 20. Jahrhunderts kommt ein entscheidender Bereich in der Produktentwicklung hinzu. Durch die Potenziale der Elektronik und Informatik wurden die Lösungen im Maschinen- und Fahrzeugbau immer stärker zu mechatronischen Produktlösungen, bei denen die Nutzung der Elektronik und der zugehörigen Programmierung die möglichen Lösungsräume für Funktionen extrem erweitert haben. Heute findet man sowohl im Bereich der Konsumerprodukte als auch in den Produktlösungen der Investitionsgüterindustrie eine sehr enge Symbiose aus mechanischen, elektronischen und informationstechnischen Teilsystemen, die dann zu einem Gesamtsystem zusammen komponiert werden. Die Architektur solcher moderner Produktlösungen beinhaltet dabei immer auch umfangreiche elektronische Hardware in Form von Steuergeräten, die dann die, in den Programmen abgelegten Steuerungs- und Regelungsalgorithmen abarbeiten und so die Steuerungssignale für die mechanischen Funktionsträger erzeugen. Dabei entstehen zum Teil sehr hohe Anforderungen auch an die Steuergeräte, zum Beispiel bezogen auf Speicher, aber auch der Rechengeschwindigkeit, die realisiert werden müssen. Gleichzeitig führt die zunehmende Individualisierung von Lösungen dazu, dass unterschiedliche Steuergeräte in sehr großer Stückzahl verbaut werden. Dies gilt dabei im Bereich der Antriebssysteme sowohl für die Leistungsantriebe als auch für den Bereich der Nebenantriebe. Besonders im Fahrzeugbau sind in modernen Fahrzeugen eine große Anzahl an Steuergeräten verbaut, die die Kundenfunktionen realisieren helfen. Diese Steuergeräte methodisch und strukturiert zu entwickeln und dabei auch die richtige Balance zwischen Individualität, Performance und Kosten zu realisieren, stellt eine große Herausforderung für die Entwickelnden im Bereich der Steuergeräte dar.

Ein Ansatz, der im klassischen Maschinenbau schon sehr weit vorangetrieben worden ist, um die Herausforderungen äußerer Vielfalt durch möglichst geringe innere Vielfalt zu realisieren, ist das Konzept der Baukastenentwicklung. Dabei ist die Grundstrategie, durch eine geschickte Komposition der Bausteine des Baukastens eine möglichst große äußere Vielfalt mit möglichst wenig unterschiedlichen Bausteinen darzustellen. Um im Bereich der Steuergeräte die Potenziale des Baukastenkonzeptes zu nutzen, müssen die Ansätze erweitert und gezielt auf die Anwendung in der Produktgenerationsentwicklung mechatronischer Steuergeräte adaptiert werden. Damit kann ein hohes Potenzial zu einer

schnellen, effizienten und effektiven und auch kostengünstigen Entwicklung unterschiedlicher Steuergerätgenerationen gewonnen werden.

In seiner wissenschaftlichen Arbeit hat sich Herr Dr.-Ing. Jonas Powelske genau dieser Herausforderung gestellt. Auf Basis des Konzeptes der PGE – Produktgenerationsentwicklung nach Albers erforscht und entwickelt er eine praxisorientierte Methode zu Unterstützung der Baukastenentwicklung für Steuergeräte auf Basis des Modell-based-Systems-Engineering.

März, 2023

Albert Albers

# Kurzfassung

Die Entwicklung des Automobils hin zu einem HiTech Multimediaprodukt ist nicht erst durch die neusten Erkenntnisse im Bereich des autonomen Fahrens abzusehen. Bereits seit dem Einzug der Mechatronik ins Fahrzeug ist eine stete Zunahme an Funktionalität zu beobachten. Im Zentrum des Geschehens stehen dabei die Steuergeräte, die als intelligente Schaltzentralen außerhalb oder innerhalb der elektrischen Komponenten des Fahrzeuges agieren. Der Bereich der elektrischen Kleinantriebe stellt bis heute einen der bedeutendsten Sektoren dieser Komponenten dar. Durch den zunehmenden globalen Wettbewerb sind auch in diesem Segment Individualität und Differenzierung zu unverzichtbaren Verkaufsargumenten geworden. Standardisierungsmethoden allen voran die Baukastenentwicklung bieten an dieser Stelle die Aussicht, sich langfristig am Markt zu behaupten. Anders als in der Automobilindustrie, deren Baukästen über Jahre bestehen bleiben, zeichnet sich die Steuergeräteentwicklung aufgrund der unterschiedlichen Entwicklungszyklen der beteiligten Domänen, durch eine sehr hohe Innovationsdynamik aus. Gepaart mit den Trends der Hochintegration und der zunehmenden Funktionsdichte führt dies zu einem drastischen Anstieg der Produktkomplexität. Gerade in der frühen Phase eines Produktentstehungsprozesses, die sich ohnehin durch eine hohe Unsicherheit auszeichnet, gilt es geeignete Wege zum Umgang mit der vorherrschenden Komplexität zu ermöglichen. Hieraus entsteht der Bedarf nach geeigneten Ansätzen die spezifischen Herausforderungen der Steuergeräteentwicklung im Einsatz elektrischer Kleinantriebe mittels Baukastenentwicklung adressieren zu können.

Im Rahmen der vorliegenden Arbeit werden, an die Spezifika der frühen Phase der PGE – Produktgenerationsentwicklung mechatronischer Steuergeräte angepasste Methoden zur Baukastenstrukturierung und -anwendung entwickelt. Methodisches und modelltheoretisches Fundament des Forschungsvorhabens bilden dabei die Erkenntnisse der KaSPro – Karlsruher Schule für Produktentwicklung. Mittels der Strukturierungsmethode wird eine detaillierte Vorgehensweise beschrieben, die den Leser ausgehend von einem Produktportfolio mechatronischer Steuergeräte zur Erstellung eines MBSE-gestützten Baukastenmodells führt. Um die Anwendung des Baukastens in der industriellen Praxis zu ermöglichen, wird eine Toolkette entwickelt, die dazu in der Lage ist, basierend auf realen Kundenanforderungen Produktempfehlungen aus dem Baukastenmodell heraus zu generieren und gegeneinander abzuwägen. Die unter der Durchführung von Fallstudien und Experteninterviews generierten Evaluationsergebnisse bestätigen die Eignung der Methoden zum praktischen Einsatz. Die automatisierte Baukastenentwicklung zeichnet sich zusätzlich durch einen signifikanten Beitrag hinsichtlich der Effizienzsteigerung in der Ableitung von Produktkonzepten in der frühen Entwicklungsphase aus.



# Abstract

The development of the automobile into a high-tech multimedia product is not only foreseeable due to the latest findings in the field of autonomous driving. Ever since mechatronics entered the vehicle, a steady increase in functionality has been observed. At the center of the action are the control units, which act as intelligent control centers outside or inside the vehicle's electrical components. The field of small electric drives still represents one of the most important sectors of these today. Due to increasing global competition, individuality and differentiation have become indispensable sales arguments in this segment as well. Standardization methods, above all modular kits, offer the prospect of long-term market success. Unlike in the automotive industry, where modular kits remain in place for years, ECU development is characterized by very high innovation dynamics due to the different development cycles of the domains involved. Coupled with the trends of high integration and increasing functional density, this leads to a drastic increase in product complexity. Particularly in the early phase of a product development process, which is characterized by a high degree of uncertainty anyway, it is important to enable suitable ways of dealing with the prevailing complexity. This leads to the need for suitable approaches to address the specific challenges of ECU development in the use of small electric drives by means of modular development.

In the context of the present work, methods for modular structuring and application adapted to the specifics of the early phase of the PGE – Product Generation Engineering of mechatronic ECUs are developed. The methodological and model-theoretical foundation of the research project is formed by the findings of the KaSPro - Karlsruhe School for Product Development. By means of the structuring method, a detailed procedure is described that leads the reader from a product portfolio of mechatronic control units to the creation of an MBSE-supported modular kit model. In order to enable the application of the modular kit in industrial practice, a tool chain is developed which is able to generate product recommendations from the modular kit based on real customer requirements and to weigh them against each other. The evaluation results generated by conducting case studies and expert interviews confirm the suitability of the methods for practical use. The automated modular kit application also makes a significant contribution to increasing efficiency in the derivation of product concepts in the early development phase.





# Danksagung

Die Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Produktentwicklung (IPEK) des Karlsruher Instituts für Technologie (KIT) in Kooperation mit der Robert Bosch GmbH.

Im Besonderen möchte ich mich bei meinem Doktorvater Herrn Prof. Dr.-Ing. Dr. h.c. Albert Albers für das entgegengebrachte Vertrauen und seine herzliche Art bedanken. Die gemeinsamen wissenschaftlichen Gespräche waren für mich stets eine Inspiration und hatten grundlegenden Anteil an meiner fachlichen und menschlichen Entwicklung.

Für die Übernahme des Korreferats bedanke ich mich bei Herrn Prof. Dr.-Ing. Nikola Bursac, der meine Forschung bereits als Oberingenieur inhaltlich begleiten durfte. Großer Dank gilt ebenfalls Herrn Prof. Dr. sc. techn. Thomas Koch für die Übernahme des Prüfungsvorsitzes.

Ein großer Dank gilt meinem ehemaligen Vorgesetzten bei der Robert Bosch GmbH Herrn. Vincent Rieger für die spannende und lehrreiche Zeit. Besonderer Dank gilt ebenfalls meinem inhaltlichen Mentor Herrn Prof. Dr.-Ing. Patrick König für seine Unterstützung und die Verbundenheit über die gemeinsame Zeit im Unternehmen hinaus.

Mein Dank gilt ebenfalls den Kollegen und Kolleginnen der Forschungsgruppe Entwicklungsmethodik und -management für den wissenschaftlichen Austausch und die großartige Zusammenarbeit. In diesem Zusammenhang möchte ich mich besonders bei Frau Dr.-Ing. Natalie Balle für die gemeinsame Zeit und die entstandene Freundschaft bedanken. Besonderer Dank gebührt ebenfalls Herrn Constantin Mandel für seine Unterstützung und die Verbundenheit über die Zeit am IPEK hinaus.

Weiterhin möchte ich mich bei allen Studierenden bedanken, die zum Gelingen dieser Arbeit beigetragen haben: Moritz Zanger, Christian Martens, Peter Schad, Johannes Galenzowski, Lena Hinterseh, Jörg Neye, Daniel Baumann und Yannick Rauch.

Von Herzen möchte ich mich bei meinen Eltern Reinhard und Anna Maria Powelske bedanken, die mir durch ihren Einsatz meine Ausbildung und berufliche Laufbahn ermöglicht haben. Darüber hinaus gilt mein Dank meinen Geschwistern Julia und Christian Powelske die immer ein offenes Ohr für mich haben und mir das Gefühl geben getragen zu sein. Zuletzt gilt mein größter Dank meiner Freundin Michaela Nawrath die mich durch ihre liebevolle Art an guten wie auch schlechten Tagen unterstützt hat und während dieser Zeit mein größter Resilienzfaktor war.

Karlsruhe, den 05. März 2023

Jonas Powelske



Denn Gott ist's, der in euch wirkt beides, das Wollen und das Vollbringen, nach  
seinem Wohlgefallen.

Philipper 2:13



# Inhaltsverzeichnis

<b>Kurzfassung</b> .....	<b>i</b>
<b>Abstract</b> .....	<b>iii</b>
<b>Abbildungsverzeichnis</b> .....	<b>xiii</b>
<b>Tabellenverzeichnis</b> .....	<b>xxvii</b>
<b>Abkürzungsverzeichnis</b> .....	<b>xxix</b>
<b>Formelzeichen</b> .....	<b>xxxiii</b>
<b>1 Einleitung</b> .....	<b>1</b>
1.1 Fokus der Arbeit.....	2
1.2 Aufbau der Arbeit .....	4
<b>2 Grundlagen und Stand der Forschung</b> .....	<b>7</b>
2.1 Mechatronische Steuergeräte.....	7
2.1.1 Mechatronische Systeme .....	8
2.1.2 Grundlagen mechatronischer Steuergeräte .....	15
2.1.3 Fazit.....	31
2.2 Model Based Systems Engineering .....	32
2.2.1 Systemtheorie.....	35
2.2.2 Modelltheorie .....	39
2.2.3 Prozessmodelle .....	47
2.2.4 Produktarchitektur .....	64
2.2.5 Funktions-Gestalt-Zusammenhang.....	71
2.2.6 Systems Modelling Language.....	73
2.2.7 MBSE in der Anwendung.....	78
2.2.8 Fazit.....	81
2.3 PGE – Modell der Produktgenerationsentwicklung nach Albers .....	82
2.3.1 Das Modell der PGE – Produktgenerationsentwicklung nach Albers .....	82
2.3.2 Frühe Phase der PGE .....	88
2.3.3 Fazit.....	95
2.4 Baukastenentwicklung als Standardisierungsmethode .....	96
2.4.1 Grundlagen der Standardisierung modularer Produktarchitekturen.....	96
2.4.2 MBSE-gestützte Baukastenentwicklung .....	103
2.4.3 Kompatibilität von Bausteinen.....	115

2.4.4	Fazit.....	118
<b>3</b>	<b>Zielsetzung und Vorgehensweise .....</b>	<b>119</b>
3.1	Zielsetzung .....	119
3.1.1	Forschungsbedarf.....	119
3.1.2	Forschungshypothesen.....	122
3.2	Forschungsfragen.....	123
3.3	Forschungsmethode und Vorgehensweise .....	124
3.3.1	Forschungsmethode .....	124
3.3.2	Empirische Methoden .....	127
3.4	Forschungsumgebung.....	130
3.4.1	Entwicklungstätigkeiten und Einordnung in die Phasen von Forschung & Entwicklung.....	130
3.4.2	Systemverständnis der AE/ENS und organisatorische Rahmenbedingungen.....	134
<b>4</b>	<b>Modellierung der Baukastenstrukturierung .....</b>	<b>139</b>
4.1	Ermittlung des Zielsystems für die Baukastenstrukturierung .....	139
4.1.1	Studiendesign .....	140
4.1.2	Studienergebnisse .....	144
4.2	Entwicklung einer Strukturierungsmethode.....	156
4.2.1	Ableitung der Strukturierungskriterien .....	161
4.2.2	Herleitung der Strukturierungsmethode.....	165
4.3	Implementierung der Strukturierungsmethode mittels MBSE .....	176
4.3.1	Analyse bestehender Produkte .....	177
4.3.2	Anwendung der Strukturierungsmethode .....	179
4.3.3	Erstellung des Eigenschafts-Abhängigkeits-Netzwerks (EAN) .....	188
4.3.4	Ermittlung der Bausteindaten .....	192
4.3.5	Erstellung des Baukastenmodells in IBM Rhapsody .....	196
4.4	Diskussion und Zwischenfazit.....	205
<b>5</b>	<b>Modellierung der Baukastenanwendung.....</b>	<b>209</b>
5.1	Ermittlung des Zielsystems für die Baukastenanwendung .....	209
5.1.1	Ermittlung der Bedarfe an die Baukastenentwicklung .....	211
5.1.2	Ermittlung der gewünschten Stakeholderinteraktion .....	216
5.2	Entwicklung einer Methode zur Baukastenanwendung .....	223
5.2.1	Gestaltung der Kompatibilitätsprüfung .....	231
5.2.2	Identifikation potenzieller Lücken des Baukastens .....	243
5.3	Implementierung der Baukastenanwendung mittels MBSE.....	245
5.3.1	Aufbau einer Toolkette zur automatisierten Baukastenanwendung.....	245
5.3.2	Aufbau der zentralen Benutzeroberfläche .....	247

5.3.3	Realisierung der Stakeholder Use-Cases .....	250
5.4	Diskussion und Zwischenfazit.....	265
<b>6</b>	<b>Validierung der entwickelten Methoden.....</b>	<b>269</b>
6.1	Impact Models .....	269
6.1.1	Aufbau der Impact Models .....	270
6.1.2	Impact Model Baukastenstrukturierung.....	273
6.1.3	Impact Model Baukastenanwendung .....	276
6.2	Übersicht der Validierungsschritte .....	280
6.3	Ergebnisse der Validierungsaktivitäten.....	292
6.4	Diskussion der Ergebnisse .....	298
<b>7</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>301</b>
7.1	Zusammenfassung .....	301
7.2	Ausblick.....	306
	<b>Literaturverzeichnis.....</b>	<b>I</b>
	<b>Glossar .....</b>	<b>XXXV</b>
	<b>Anhang A.....</b>	<b>XXXIX</b>
	Interviewleitfaden DS I – Baukastenstrukturierung.....	XXXIX
	Interviewleitfaden DS I – Baukastenanwendung .....	XLII
	Interviewleitfäden DS II – Validierungsaktivitäten.....	XLV
	<b>Liste der wissenschaftlichen Veröffentlichungen des Autoren .....</b>	<b>LVII</b>





# Abbildungsverzeichnis

Abbildung 1.1:	Darstellung des Forschungsfeldes nach Blessing & Chakrabarti (2009) im Kontext der Ontologie der KaSPro - Karlsruher Schule für Produktentwicklung .....	3
Abbildung 1.2:	Elemente und Zusammenhänge des IPEK-X-in-the-Loop-(XiL)-Ansatzes (Albers & Düser, 2010). .....	4
Abbildung 2.1:	Mechatronische Systeme im Automobil. Eigene Darstellung in Anlehnung an Isermann (1999), Czichos (2015), Reif (2011). Erstellt unter Verwendung von Darstellungen der Robert Bosch GmbH (Robert Bosch GmbH, 2022).....	10
Abbildung 2.2:	Grundstruktur mechatronischer Systeme. Darstellung in Anlehnung an die Neuauflage der VDI 2206 im November 2021 (VDI, 2021).....	12
Abbildung 2.3:	Übersicht des Grundtyps mechatronischer Funktionsmoduli nach Steffen (2007). Geordnet nach Profil der Aufgabenstellung, Produktbeispielen und des resultierenden Produktstrukturtyps. ....	14
Abbildung 2.4:	Elemente eines elektrischen Antriebssystems am Beispiel eines Scheibenwischersystems. Eigene Darstellung in Anlehnung an Stöltzing, Kallenbach & Amrhein (2011).....	18
Abbildung 2.5:	Vier-Quadranten-Betrieb eines E-Antriebs illustriert anhand der Richtungen von Drehzahl bzw. Spannung und Drehmoment bzw. Stromstärke (eigene Darstellung).....	20
Abbildung 2.6:	Übersicht und Einordnung elektrischer Kleinantriebe. Das Diagramm zeigt verschiedene Produkte, in denen die entwickelten Kleinantriebe enthalten sind. Diese sind entsprechend ihrer Betriebsspannung (x-Achse) und deren Ausgangsleistung (y-Achse) geordnet. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022).....	21
Abbildung 2.7:	Grundtypen elektrischer Synchronmaschinen (Reluktanzmaschine, permanentmagneterregte Synchronmaschinen und fremderregte Synchronmaschine) in Anlehnung an Winzer (2017).....	23

Abbildung 2.8:	Einsatzbereiche mechatronischer Steuergeräte in Kleinantrieben im Automobilbereich. Eigene Darstellung nach Reif (2011), unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022).....	25
Abbildung 2.9:	Grundbestandteile eines Steuergerätes im Kontext der vorliegenden Arbeit, in Anlehnung an (Streichert & Traub, 2012, sowie Wallentowitz & Reif, 2011).....	26
Abbildung 2.10:	Übersicht verschiedener Substratanbindungen und Entwärmungskonzepte am Beispiel eines Zwischenkreiskondensators (Darstellung in Anlehnung an Powelske et al., 2017) .....	29
Abbildung 2.11:	Gegenüberstellung modellbasierte und dokumentenzentrierte Entwicklung auf Basis der beteiligten Domänen in Anlehnung an Alt (2012).....	34
Abbildung 2.12:	Die drei Kernaspekte des Systembegriffs, Darstellung nach der dritten Auflage (Ropohl, 2009).....	36
Abbildung 2.13:	Merkmale von Modellen (eigene Darstellung nach Stachowiak (1973)).....	40
Abbildung 2.14:	Modelling Triple (Sprache-Tool-Prozess) nach Matthiesen et al. (2014) .....	43
Abbildung 2.15:	Das Sichtenkonzept des MBSE nach Bursac (2016) und Alt (2012).....	44
Abbildung 2.16:	Ansätze zur Modellbildung (integrierter vs. förderierter Ansatz) nach Kleiner (2003). .....	45
Abbildung 2.17:	Das V-Modell nach VDI 2206 (2004). .....	49
Abbildung 2.18:	Übersicht des ISO 26262 Standards für sicherheitsrelevante elektrische/elektronische Systeme in Kraftfahrzeugen (ISO 26262, 2018). .....	52
Abbildung 2.19:	Übersicht des ASPICE Rahmenprozesses (ASPICE, 2017).. ..	54
Abbildung 2.20:	Allgemeines Modell der Produktentwicklung 1993 (VDI 2221, 1993). .....	55
Abbildung 2.21:	Allgemeines Modell der Produktentwicklung 2018 (VDI 2221 Blatt 1, 2018). .....	57
Abbildung 2.22:	Das erweiterte ZHO-Modell (Albers, Lohmeyer & Ebel, 2011; Albers, Behrendt et al, 2013). .....	60
Abbildung 2.23:	Das iPeM – integrierte Produktentstehungsmodell (Albers, Reiß, Bursac & Richter, 2016). .....	63

Abbildung 2.24:	Darstellung der Produktarchitektur auf Basis der Transformation einer Funktionsstruktur in eine Produktstruktur nach Göpfert (1998).....	66
Abbildung 2.25:	Funktionale Dekomposition eines Systems (Feldhusen, Grote, Göpfert & Tretow, 2013). Gezeigt ist die Zerlegung einer Gesamtfunktion in Teilfunktionen. ....	67
Abbildung 2.26:	Ausschnitt der vertikalen Strukturierung eines elektrischen Antriebs bestehend aus Steuergerät, Elektromotor und Getriebe. Eigene Darstellung nach Pahl, Beitz et al. (2007) ....	69
Abbildung 2.27:	C&C <sup>2</sup> -A Modell am Beispiel eines Hybridantriebes nach Albers & Wintergerst (2014). ....	72
Abbildung 2.28:	SysML-Diagrammtaxonomie nach Object Management Group (2007).....	75
Abbildung 2.29:	Nutzung von bdd und ibd am Beispiel des elektrischen Antriebs bestehend aus ECU, Motor und Getriebe (Darstellung in IBM Rhapsody).....	76
Abbildung 2.30:	Attribute und Ports in SysML am Beispiel eines Systems bestehend aus ECU und Motor (Darstellung in IBM Rhapsody). ....	77
Abbildung 2.31:	Darstellung der Variationsanteile Prinzipvariation und Gestaltvariation am Beispiel der Endstufengestaltung. Eigene Darstellung entsprechend den Grundlagen von Albers et al. (2020) ...	84
Abbildung 2.32:	Risikobewertung auf Basis von Neuentwicklungsanteil (GV + PV) und Herkunft des Referenzprodukts am Beispiel des ZMS nach Albers, Rapp et al. (2017).....	86
Abbildung 2.33:	Möglichkeit von Kostenbeeinflussung und -beurteilung im Produktentwicklungsprozess nach Ehrlenspiel & Meerkamm (2013).....	89
Abbildung 2.34:	Zusammenhang zwischen Theorie, Technologie und Technik nach Specht et al. (2002). ....	92
Abbildung 2.35:	Schematische Darstellung der „Frühen Phasen“ einer Produkt-Generationsentwicklung und der dazugehörigen Vorentwicklung eines (Teil-)Systems nach Albers, Rapp et al. (2017).....	93
Abbildung 2.36:	Aktivitäten der Forschung & Entwicklung erweiterte Darstellung nach Specht et al. (2002). ....	94
Abbildung 2.37:	Fraktaler Charakter der Standardisierungsmethoden am Beispiel der Baukastenentwicklung mechatronischer	

	Steuergeräte im Einsatz in Kleinantrieben im Automobilbereich. Darstellung in Anlehnung an Albers, Scherer et al. (2015). ..	102
Abbildung 2.38:	Vorgehen beim Entwickeln eines Baukastensystems nach Ehrlenspiel (2009). .....	104
Abbildung 2.39:	Herausforderungen der Baukastenentwicklung in Anlehnung an Bursac (2016) dort nach Albers, Bursac & Scherer (2014). ..	106
Abbildung 2.40:	Abstraktionsgrade der Produktmodellierung (Bursac, 2016).	107
Abbildung 2.41:	Framework der Baukastenentwicklung in Anlehnung an Bursac (2016). .....	109
Abbildung 2.42:	Elemente der Baukastenentwicklung im Produktmodellierungs-Framework nach Bursac (2016).....	110
Abbildung 2.43:	Übersicht der Baukastenarchitekturen: singuläre, hybride und autarke Architektur (Powelske et al., 2017). .....	113
Abbildung 3.1:	Forschungsvorgehen (genutzte Methoden, Phasen und jeweiliges Ergebnis) der vorliegenden Arbeit entsprechend der DRM nach Blessing & Chakrabarti (2009). .....	125
Abbildung 3.2:	Methodensteckbrief (teil-)strukturiertes Interview nach Marxen (2011). .....	128
Abbildung 3.3:	Methodensteckbrief retrospektives Protokoll nach Marxen (2011).....	128
Abbildung 3.4:	Methodensteckbrief teilnehmende Beobachtung nach Marxen (2011). .....	129
Abbildung 3.5:	Methodensteckbrief Fallstudie nach Marxen (2011) .....	129
Abbildung 3.6:	Auszug des Betätigungsfeldes der ENS. Das Diagramm zeigt verschiedene Produkte, in denen die entwickelten Steuergeräte enthalten sind. Diese sind entsprechend ihrer Betriebsspannung (x-Achse) und deren Ausgangsleistung (y-Achse) geordnet. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022).....	131
Abbildung 3.7:	„Innovation-Pipeline“ der ENS. Darstellung der geplanten Nutzung von Technologien als Elemente des Referenzsystems aktueller und kommender Produktgenerationen, entsprechend der Nomenklatur der PGE – Produktgenerationsentwicklung. ....	132

Abbildung 3.8:	Ebenenmodell des ENS-Systemverständnisses inklusive Fokus- und Schnittstellenthemen. Levelstruktur aufgeteilt in Produkt-, System-, Sub-System- und Domänen-Level. ....	135
Abbildung 3.9:	Vereinfachtes Organigramm der ENS als Matrixorganisation in Anlehnung an Schlick et al. (2018). Projektorganisation (links) und Linienorganisation (rechts) sind dabei zur Matrix verwoben. ....	137
Abbildung 4.1:	Übersicht der, zur Ermittlung des Zielsystems angewandten empirischen Methoden in Abhängigkeit der Teilforschungsfragen (vgl. Kapitel 3.2) .....	141
Abbildung 4.2:	Übersicht der Domänenzugehörigkeit der Interviewpartner aufgeteilt in Management- und Entwicklungsdisziplinen.....	144
Abbildung 4.3:	Übersicht der Studienergebnisse abhängig von den gewählten empirischen Methoden und deren Beitrag zur Beantwortung der Teilforschungsfragen.....	145
Abbildung 4.4:	Reference Model der Strukturierungsmethode der vorliegenden Arbeit entsprechend der DRM von Blessing & Chakrabarti (2009).....	154
Abbildung 4.5:	Ergebnisse der deskriptiven Studie I geordnet anhand der entsprechenden Teilforschungsfragen (FF1.1 – FF1.4).....	155
Abbildung 4.6:	Prinzipien der Funktionsintegration und Funktionsdifferenzierung abhängig von der gewählten Art der Produktarchitektur (Ponn & Lindemann, 2008, S. 149).....	158
Abbildung 4.7:	Erläuterung der Architekturprinzipien integral und modular am Beispiel der Gestaltung einer Endstufe. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022).....	158
Abbildung 4.8:	Kriterien der Baukastenstrukturierung für Steuergeräte im Einsatz in Kleinantrieben im Automobilbereich (Powelske et al., 2017).....	161
Abbildung 4.9:	Veränderungen in der Modulbildung im Rahmen der Baukastenentwicklung mechatronischer Steuergeräte im Einsatz von Kleinantrieben im Automobilbereich, durch ganzheitliche Betrachtung am Beispiel des Zwischenkreiskondensators nach Powelske et al. (2017) .....	163
Abbildung 4.10:	Prinzip der Bildung einer User-Funktion in Anlehnung an Pohl (2012).....	166

Abbildung 4.11:	Gruppierung von Funktionen zu einem Funktionscluster entsprechend der späteren technischen Umsetzung. ....	167
Abbildung 4.12:	Illustration der Wirkzusammenhänge von Funktionsclustern über Funktionsstrukturausschnitte (FSAs) innerhalb der Funktionsstruktur mechatronischer Steuergeräte. F1, F2 sowie F1a und F2b stellen dabei Funktionscluster unterschiedlicher hierarchischer Strukturierungstiefe dar. ....	168
Abbildung 4.13:	Anbindung integraler Bausteinvarianten an die Funktionscluster, entsprechend der technischen Umsetzung in realen Produkten. ....	170
Abbildung 4.14:	Darstellung eines Bausteinsteckbriefs der integralen Bausteinvariante eines Zwischenkreiskondensators inklusive verwendeter Elemente der Aufbau und Verbindungstechnik. Enthalten sind neben designabhängigen Attributen (z.B. Gewicht), funktionale Parameter (z.B. Spannung) und Informationen über das Bausteinverhalten.....	171
Abbildung 4.15:	Bildung modularer Bausteinvarianten auf Basis der motivierten Struktur von Funktionsclustern und Funktionsstrukturausschnitten. Die aus der Kombination entstandenen modularen BV (mBV1.1 – mBV1.3) entsprechen dabei dem Funktionsumfang der integralen BV des übergeordneten Funktionsclusters (F1). ....	172
Abbildung 4.16:	Umgesetzte (mBV 1.1) und mögliche (mBV1.2) modulare Bausteinvarianten am Beispiel einer Endstufe. Die umgesetzte modulare BV enthält zusätzliche Informationen aus deren Einsatz in realen Produkten (P1). ....	173
Abbildung 4.17:	Parallele Dekompositionspfade bei der Baukastenstrukturierung. Das initiale Funktionscluster lässt sich dementsprechend auf verschiedene Weise in Funktionscluster (F1+F2 bzw. F3+F4) dekomponieren.....	175
Abbildung 4.18:	Vorgehensweise zur Implementierung der Strukturierungsmethode für Baukästen mechatronischer Steuergeräte bestehend aus den fünf angeführten Schritten. ....	177
Abbildung 4.19:	Eigenschaften der Antriebe des gewählten Beispielportfolios. Diese sind unter anderem durch geringe Ausgangsleistungen und geringe Anforderungen an die Regelalgorithmen gekennzeichnet. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022). ....	178

Abbildung 4.20:	Übersicht der schematischen Blockschaltbilder zweier Produkte des gewählten Beispielportfolios. Hervorgehoben sind die unterschiedlichen technischen Lösungen für das Logikmodul eines mechatronischen Steuergerätes. ....	180
Abbildung 4.21:	Gegenüberstellung der diskreten Lösungen zweier Produkte des gewählten Beispielportfolios für das Modul Mikrocontroller + Treiberbaustein. ....	181
Abbildung 4.22:	Prüfung des Varianzkriteriums am Beispiel des zuvor betrachteten Logikmoduls und dessen unterschiedlichen technischen Lösungen. ....	182
Abbildung 4.23:	Prüfung des Abhängigkeitskriteriums am Beispiel des Logikmoduls und dessen unterschiedlichen technischen Lösungen. ....	183
Abbildung 4.24:	Prüfung des Verhältnismäßigkeitskriteriums am Beispiel des Logikmoduls und dessen unterschiedlichen technischen Lösungen. ....	184
Abbildung 4.25:	Einführung einer neuen Strukturierungsebene für das Logikmodul in Form eines neuen Funktionsstrukturausschnittes (FSA) und den Funktionsclustern „Mikrocontroller“ und „Treiber“. ....	185
Abbildung 4.26:	Bildung der modularen Bausteinvarianten für das Logikmodul durch Kombination der integralen Bausteinvarianten der Funktionscluster „Mikrocontroller“ und „Treiber“. ....	186
Abbildung 4.27:	Beispielhafte Modulstruktur eines mechatronischen Steuergerätes als Ergebnis der Strukturierungsmethode. Gezeigt sind resultierende Module, Sub-Module und optionale Module. ....	187
Abbildung 4.28:	Ausschnitt des Moduls Zwischenkreis inkl. Key-Parameter für Eingangsstrom, Ausgangsstrom und Motorphasenanzahl. ....	189
Abbildung 4.29:	Erstellung eines Eigenschafts-Abhängigkeits-Netzwerks (EAN): Regeln für die Verbindung benachbarter Module. ....	190
Abbildung 4.30:	Erstellung eines Eigenschafts-Abhängigkeits-Netzwerks (EAN): Regeln für die Vorgehensweise zur Verbindung nicht benachbarter Module. ....	191
Abbildung 4.31:	Übersicht des resultierenden Eigenschafts-Abhängigkeits-Netzwerks eines mechatronischen Steuergerätes am Beispiel des Mikrocontroller Moduls und dessen Schnittstellen. ....	192

Abbildung 4.32:	Exemplarischer Aufbau eines Bausteinsteckbriefs. Enthalten sind Attributname, Attributtyp, Einheit des Attributs und die Quantifizierung entsprechend des Einsatzes in realen Produkten. ....	193
Abbildung 4.33:	Nachverfolgbarkeit zwischen Anforderungen und technischer Umsetzung. Die Anforderungen sind an die Systemfunktionen gestellt, welche wiederum an ein Funktionscluster allokiert sind. Die Prüfung der quantifizierten Anforderungen erfolgt anhand der quantifizieren Key-Parameter der Bausteinvariante im Bausteinsteckbrief. ....	195
Abbildung 4.34:	Fünf Schritte zur Erstellung des Baukastenmodells für mechatronische Steuergeräte im Einsatz in Kleinantrieben in IBM Rhapsody. ....	196
Abbildung 4.35:	Schematische Darstellung der Struktur des DOORS Funktionskataloges. Der Aufbau der Struktur orientiert sich dabei an der ENS Systemstruktur (vgl. Abbildung 3.8, Kapitel 3.3.2) .....	197
Abbildung 4.36:	Schematischer Aufbau der Referenz-Funktionsstruktur in IBM Rhapsody. Eine übergeordnete Systemfunktion wird dabei mittels Kompositionsbeziehung in Domänenfunktionen aufgeteilt. Eine Anforderung ist über <<satisfy>>Beziehung angehängt. ....	198
Abbildung 4.37:	Übersicht der Flussattribute und stationären Attribute eines Funktionsclusters in IBM Rhapsody. Die Flussattribute weisen zusätzlich eine Flussrichtung auf. ....	200
Abbildung 4.38:	Umsetzung der Beziehungen zwischen FSA und Funktionscluster mittels <<Allocation>>Beziehung in IBM Rhapsody. ....	201
Abbildung 4.39:	Übersicht eines Funktionsstrukturausschnittes für das Modul „System Basis Chip“. Enthalten sind stationäre Attribute (Verbindung über Linien dargestellt), sowie Flussattribute (Verbindung über Pfeile dargestellt).....	202
Abbildung 4.40:	Übersicht einer erstellten Bausteinvariante inklusive Attribute und <<ImplementationOf>> Beziehung in IBM Rhapsody. ...	203
Abbildung 4.41:	Übertrag der quantifizierten Attributwerte aus den Bausteinsteckbriefen in IBM Rhapsody. ....	204
Abbildung 4.42:	Fünf Schritte zur Implementierung der Strukturierungsmethode .....	206



Abbildung 5.1:	Übersicht der deskriptiven Studie I der Baukastenanwendung anhand genutzter Grundlagen, empirischer Methoden und zugrundeliegender FF. ....	210
Abbildung 5.2:	Übersicht des Zusammenspiels aus Baukastenstrukturierung und Baukastenanwendung für mechatronische Steuergeräte. ....	210
Abbildung 5.3:	Potenzielle Schlüsselfaktoren der Baukastenanwendung für mechatronische Steuergeräte. Ermittelt auf Basis (teil-)strukturierter Interviews mit Vertretern des Managements, der System- und Modulentwicklung. ....	212
Abbildung 5.4:	Reference Model der Baukastenanwendung (Managementsicht) entsprechend der DRM nach Blessing & Chakrabarti (2009). ....	214
Abbildung 5.5:	Reference Model der Baukastenanwendung (Entwicklersicht) entsprechend der DRM nach Blessing & Chakrabarti (2009). ....	215
Abbildung 5.6:	Interaktion des Nutzers mit der entwickelten Baukasten-anwendung.....	216
Abbildung 5.7:	Übersicht der stakeholderspezifischen User-Stories für die entwickelte Baukastenanwendung. ....	217
Abbildung 5.8:	Zuordnung der abgeleiteten Use-Cases zu den User-Stories für die entwickelte Baukastenanwendung.....	218
Abbildung 5.9:	Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 1 .....	219
Abbildung 5.10:	Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 2. ....	220
Abbildung 5.11:	Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 3. ....	221
Abbildung 5.12:	Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 4. ....	222
Abbildung 5.13:	Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 5. ....	223
Abbildung 5.14:	Übersicht der Ausgangslage nach Anwendung und Implementierung der Strukturierungsmethode für mechatronische Steuergeräte in IBM Rhapsody. ....	224
Abbildung 5.15:	„Interne“ und „externe“ Anforderungen an ein Produkt als Grundlage der Kompatibilitätsbetrachtung. Interne bzw. externe	

	Anforderungen sind dabei Elemente des Zielsystems von Teilsystemen bzw. übergeordneten Systemen. ....	225
Abbildung 5.16:	Verortung der internen Kompatibilitätsbetrachtung entsprechend der Methode zu Baukastenstrukturierung für mechatronische Steuergeräte. Modulare BV eines untergeordneten FSA werden nach deren Prüfung eine ebene höher wiederum zur Kombination herangezogen.....	226
Abbildung 5.17:	Externe und interne Kompatibilität eines mechatronischen Steuergerätes am Beispiel einer spezifischen Bausteinkombination (A1, B3, C1) der enthaltenen Module..	228
Abbildung 5.18:	Einfluss von Baukastenstrukturierung und -anwendung für mechatronische Steuergeräte auf den Zyklus aus Induktion und Deduktion im Framework für die Baukastenentwicklung nach Bursac (2016). ....	230
Abbildung 5.19:	Beispielanwendung des Kompatibilitätstypen „logische Schnittmenge“ für das Attribut „Kommunikationsschnittstelle“ der Bausteine A1 und B3 bzw. A2 und B3. ....	232
Abbildung 5.20:	Beispielanwendung des Kompatibilitätstypen „Vereinigungsmenge“ für das Attribut „Substratanbindung“ der Bausteine A und B. ....	233
Abbildung 5.21:	Beispielanwendung des Kompatibilitätstypen „SensorTypAnzahl“ für das Attribut „SensorTypAnzahl“ unterschiedlicher Kombinationen für Quelle und Senke.....	234
Abbildung 5.22:	Bestimmung aller Kombinationen eines FSA .....	235
Abbildung 5.23:	Ablauf der Kompatibilitätsprüfung für einen FSA in sechs Schritten .....	237
Abbildung 5.24:	Anwendung der Intervallverrundung nach Zanger (2018). Beispiel anhand des Intervalls eines Drehmomentes in Bezug auf die Kompatibilität eines Motors und eines Getriebes. ....	239
Abbildung 5.25:	Anpassung von Intervallen bei der Intervallverrundung nach Zanger (2018). Da das Drehmoment vom Eingangsstrom abhängig ist, muss dessen Intervall analog zum Drehmomentintervall angepasst werden. ....	240
Abbildung 5.26:	Überblick der entwickelten Toolkette zur Baukastenentwicklung .....	246
Abbildung 5.27:	Aufbau der zentralen Benutzeroberfläche der Baukastenentwicklung. ....	247

Abbildung 5.28:	Toolkettenzugriff für System- und Modulentwickler. Diese ist durch den direkten Zugriff auf das initiale Baukastenmodell charakterisiert.....	249
Abbildung 5.29:	Toolkettenzugriff für das Management. An dieser Stelle wird eine Nutzung der Anwendung ohne Zugriff auf das initiale Baukastenmodell realisiert. ....	250
Abbildung 5.30:	Übersicht der Eingabemaske der Baukastenanwendung. Diese ist entsprechend der zugrundeliegenden Attributstruktur in Vier Sektoren aufgeteilt. ....	252
Abbildung 5.31:	Ausgabemaske der externen Kompatibilitätsbewertung. ....	253
Abbildung 5.32:	Bewertung der externen Kompatibilität einer Kombination durch Gegenüberstellung der Eingabewerte mit den errechneten Kombinationswerten.....	254
Abbildung 5.33:	Übersicht der enthaltenen Bausteine einer Kombination. ....	256
Abbildung 5.34:	Übersicht eines Bausteinsteckbriefs der Baukastenanwendung. ....	257
Abbildung 5.35:	Darstellung der Gesamtkompatibilität inklusive Überblick der KPIs Kosten und Bauraum. ....	258
Abbildung 5.36:	Filtermöglichkeiten der berechneten Kombinationen .....	259
Abbildung 5.37:	Ausgabe spezifizierter Bausteingrenzen entsprechend der Anforderungen in Kapitel 5.2.2. ....	260
Abbildung 5.38:	Auswertungsmaske zur Prüfung des Beitrags einer Bausteinvariante. Auf der linken Seite ist dargestellt, welche der Bausteinvarianten den Eingabewert erfüllen bzw. nicht erfüllen können und welche der BV nicht vom Eingabewert betroffen sind. Die rechte Seite dient der Navigation durch die einzelnen Parameter. ....	261
Abbildung 5.39:	Schaltfläche zur Vorauswahl einer Bausteinvariante .....	262
Abbildung 5.40:	Übersicht der vorhandenen integralen Bausteinvarianten eines gewählten Technologiesatzes. Durch Auswahl einer Bausteinvariante erscheint diese in der Liste am rechten Bildrand.....	263
Abbildung 5.41:	Ergebnis der Kompatibilitätsbewertung bei getätigter Bausteinvorauswahl. Die vorausgewählte Bausteinvariante ist farblich hervorgehoben und weist die Namensweiterung (Preselect) auf.....	264

Abbildung 5.42:	Übersicht der entwickelten Methoden zur Baukasten- strukturierung und -anwendung für mechatronische Steuergeräte.....	265
Abbildung 5.43:	Initiale Bedarfe der Stakeholder hinsichtlich der entwickelten Baukasten-anwendung .....	266
Abbildung 6.1:	Aufbau eines Impact Models in Anlehnung an die DRM (Blessing & Chakrabarti, 2009).....	271
Abbildung 6.2:	Ausprägungen der Einflussfaktoren und Unterstützungen der vorliegenden Arbeit in Anlehnung an die DRM (Blessing & Chakrabarti, 2009).....	272
Abbildung 6.3:	Rechter Teil des Impact Models der Baukastenstrukturierung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).....	273
Abbildung 6.4:	Linker Teil des Impact Models der Baukastenstrukturierung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).....	274
Abbildung 6.5:	Impact Model der Managementsicht auf die Baukasten- anwendung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).....	277
Abbildung 6.6:	Linker Teil des Impact Models der Entwicklersicht auf die Baukasten-anwendung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).....	278
Abbildung 6.7:	Rechter Teil des Impact Models der Entwicklersicht auf die Baukasten-anwendung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).....	280
Abbildung 6.8:	Gegenüberstellung der Anforderungen der untersuchten 12 V und 48V Antriebe.....	282
Abbildung 6.9:	Übersicht der Validierungsaktivitäten entsprechend deren Verwendung für die entwickelten Methoden und der involvierten Stakeholder.....	283
Abbildung 6.10:	Studiensteckbrief: Validierung der Anwendung der entwickelten Strukturierungsmethode.....	284
Abbildung 6.11:	Studiensteckbrief: Evaluation der Schnittstellendefinition /Moduleingrenzung der entwickelten Baukastenstrukturierungs- methode.....	286
Abbildung 6.12:	Studiensteckbrief: Der verwendeten Validierungsmaßnahme Back-Testing.....	287

Abbildung 6.13:	Studiensteckbrief: Der verwendeten Validierungsmaßnahme Parallel-Testing. ....	289
Abbildung 6.14:	Studiensteckbrief: Evaluation der Bedienerfreundlichkeit der entwickelten Toolkette. ....	290
Abbildung 6.15:	Studiensteckbrief: Evaluation der Lücken des Baukastens. ..	291
Abbildung 6.16:	Kompatibilitätsübersicht der implementierten Baukasten-anwendung. ....	296
Abbildung 7.1:	Vorgehensweise zur Beantwortung der Forschungsfragen in Anlehnung an die DRM (Blessing & Chakrabarti, 2009). ....	304



# Tabellenverzeichnis

Tabelle 2.1:	Übersicht bestehender Definitionen der frühen Phase nach Bursac (2016, S.44) .....	90
Tabelle 3.1:	Arten technischer Forschungsprojekte und deren Schwerpunkte (Blessing & Chakrabarti, 2009, S. 18) .....	126





# Abkürzungsverzeichnis

ADC	Analog-Digital Wandler
AE	Automotive Electronics (Name eines Geschäftsbereiches der Robert Bosch GmbH)
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
ASE	Advanced Systems Engineering
ASPICE	Automotive Software Process Improvement and Capability Determination
ASW	Anwendungssoftware
AV	Ausprägungsvariation
AVT	Aufbau- und Verbindungstechnik
bdd	Block Definition Diagram
BEMF	Back Electromagnetic Force
BK	Baukasten
BLAC	bürstenloser Wechselstrommotor
BLDC	bürstenloser Gleichstrommotor
BMBF	Bundesministerium für Bildung und Forschung
BSW	Basissoftware
BV	Bausteinvariante
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAN	Controller Area Network
C&C <sup>2</sup> -A	Contact and Channel Approach
DC-Link	Zwischenkreiskondensator
DFG	Deutsche Forschungsgemeinschaft
DRM	Design Research Methodology

DS I	Deskriptive Studie I
DS II	Deskriptive Studie II
DSM	Design Structure Matrix
EAN	Eigenschafts-Abhängigkeits-Netzwerk
ECU	Elektronisches Steuergerät
EDS	Elektrisches Antriebssystem
E/E	Elektrik/Elektronik
Elko	Elektrolytkondensator
EMV	Elektromagnetische Verträglichkeit
ENS	Systems Engineering Drives and Control Units (Name einer Abteilung der Robert Bosch GmbH)
F	Funktionscluster
FF	Forschungsfrage
FAS	Functional Architectures of Systems
FAS4M	Functional Architectures of Systems for Mechanical Engineers
FESM	fremderregte Synchronmaschine
F&E	Forschung und Entwicklung
FSA	Funktionsstrukturausschnitt
GDU	Gate Driver Unit
GfSE	Gesellschaft für Systems Engineering
$G_n$	aktuell in der Entwicklung befindliche Produktgeneration
$G_{n+1}$	zukünftige Produktgeneration
$G_i$	Produktgenerationen
GV	Gestaltvariation
HV	Hochvolt
ibd	internal Block Diagram
INCOSE	International Council on Systems Engineering
IPEK	Institut für Produktentwicklung

iPeM	integriertes Produktentstehungsmodell
KaSPro	Karlsruher Schule für Produktentwicklung
KPI	Key-Performance Indicator
LIN	Local Interconnect Network
LSS	Leit-Stütz-Struktur
MBSE	Modellbasierte Systementwicklung
mBV	modulare Bausteinvariante
MI	Match Index
MOSFET	Metall-Oxid-Halbleiter-Feldeffekttransistor
MoSyS	Menschorientierte Gestaltung komplexer System of Systems
MR	Match Range
OMG	Object Management Group
OOSEM	Object-Oriented Systems Engineering Method
PEP	Produktentstehungsprozess
PGE	Produktgenerationsentwicklung
PLM	Project-Lifecycle-Management
PMSM	permanentmagneterregte Synchronmaschine
PS	Präskriptive Studie
PV	Prinzipvariation
RFLP	Requirements, Functions, Logics, Physics
RTE	Run-Time-Environment
SBC	System-Basis-Chip
SE	Systems Engineering
SENT	Single Edge Nibble Transmission
SMD	Surface-Mounted Device
SOD	Start of Development
SOP	Start of Production

SPES	Software Platform Embedded Systems
SPI	Serial Peripheral Interface
SysML	Systems Modelling Language
SYSMOD	System Modelling Toolbox
THT	Through-Hole Technology
TS	Technologieset (Bezeichnung der ENS für anwendungsfallspezifische Vorzugs- lösungen auf Systemebene)
UML	Unified Modelling Language
ÜV	Übernahmevariation
VDA	Verband der Automobilindustrie
V-SUM	Virtual Single Underlying Model
VR	Virtual Reality
WFP	Wirkflächenpaar
ZHO	Zielsystem, Handlungssystem, Objektsystem
ZMS	Zweimassenschwungrad

## Formelzeichen

$G_n$	aktuell in der Entwicklung befindliche Produktgeneration
$\ddot{U}S_n$	durch Übernahmevariation entwickelte Teilsysteme in $G_n$
$GS_n$	durch Gestaltvariation entwickelte Teilsysteme in $G_n$
$PS_n$	durch Prinzipvariation entwickelte Teilsysteme in $G_n$
$\delta_{\ddot{U}S_n}$	Anteil an Übernahmevariation in $G_n$
$\delta_{GS_n}$	Anteil an Gestaltvariation in $G_n$
$\delta_{PS_n}$	Anteil an Prinzipvariation in $G_n$
MI	Match Index
s	betrachtetes Element bei der Kompatibilitätsberechnung
utility(s)	spezifische Gewichtung eines Elementes s
M(s)	Match Koeffizient (Kennzahl für die Kompatibilität eines Elementes s)
MR	Match Range (Intervall des geringsten und höchsten Match Koeffizienten)
$I_V$	verrundetes Intervall
$I_{Ku}$	Kundenintervall
$I_{Kombi}$	Intervall der vom Baukasten bereitgestellten Kombination



# 1 Einleitung

In den vergangenen 40 Jahren hat nichts die Automobilindustrie so sehr verändert wie die Mechatronik. Während zu Beginn hauptsächlich mechanische Funktionsträger im Automobil substituiert wurden (Reif, 2011), sind über die Jahre Unmengen neuer Funktionen hinzugekommen. Dank der Integration vieler Sensoren und Kameras erlauben wir es den heutigen Modellen sogar zu Sehen und zu Fühlen. Das Automobil ist damit längst vom bloßen Fortbewegungsmittel zum HiTech Multimediaprodukt avanciert und zählt somit zur Klasse der sogenannten Advanced Systems (Dumitrescu et al., 2021). Im Zuge dessen sind Motorenleistung und Design nicht mehr länger die einzigen Kaufargumente (Johanning & Mildner, 2015). Durch die Megatrends Digitalisierung und Vernetzung sieht sich die Automobilindustrie nun einer neuen informationstechnischen Revolution gegenüber. Im Zentrum dieses Geschehens stehen die intelligenten Schaltzentralen des Automobils – die Steuergeräte. Bereits heute kontrollieren bis zu 100 verschiedene Steuergeräte (Broy, 2013) im Mittel über 200 elektrische Komponenten (Reuss, Diesner & Thoben, 2004) innerhalb eines Mittelklassefahrzeuges, Tendenz steigend. Durch den zunehmenden globalen Wettbewerb haben sich Individualität und Differenzierung auch in diesem Segment zu den bestimmenden Elementen am Käufermarkt entwickelt. Auch im Bereich der Steuergeräteentwicklung elektrischer Kleinantriebe im Automobilbereich stellt diese Entwicklung die Konzerne vor große Herausforderungen. Viele der Innovationen auf diesem Gebiet zählen heutzutage aus Käufersicht zu absoluten Basismerkmalen<sup>1</sup>. Auch für langjährig etablierte Konzerne wird es zunehmend schwieriger sich an diesem hart umkämpften Markt zu behaupten. In diesem Zusammenhang ist der Bedarf an Standardisierungsmethoden, allen voran an der Baukastenentwicklung, daher so hoch wie noch nie. Durch die Erhöhung der externen Vielfalt (in Form unterschiedlicher Produkte) bei gleichzeitiger Reduktion der internen Vielfalt (durch standardisierte Bausteine) kann sowohl dem Wunsch nach Individualität des Kunden entsprochen als auch ein Preisvorteil gegenüber den Wettbewerbern erzielt werden. Im Gegensatz zur Automobilindustrie, in der die Baukastenentwicklung seit geraumer Zeit etabliert ist und in der, Baukästen meist über Jahre hinweg bestehen bleiben, ist die Steuergeräteentwicklung aufgrund der unterschiedlichen Entwicklungszyklen der beteiligten Domänen, einem deutlich dynamischeren

---

<sup>1</sup> vgl. den Begriff des Basismerkmals im Kano-Modell nach Kano et al. (1984)

Wandel unterworfen. In der Domäne der Elektronik lässt sich dies anhand des Moore'schen Gesetzes verdeutlichen, wonach sich die Leistungsfähigkeit eines Mikrocontrollers alle 18 Monate verdoppelt. Die Entwicklungszyklen der reinen Softwarekomponenten gestalten sich sogar noch dynamischer (eNOVA, 2016). Weiterhin tragen der Trend der Hochintegration elektronischer Bauteile und die zunehmende Funktionsdichte zu einem überproportionalen Anstieg der Produktkomplexität bei (vgl. Kapitel 2.1.2). Die Baukastenentwicklung ist nach Albers, Bursac, Scherer, Birk, Powelske & Muschik (2019) als kontinuierlicher, den Produktentstehungsprozessen paralleler Prozess zu verstehen. Folglich ergeben sich aus den spezifischen Rahmenbedingungen im Bereich der Steuergeräteentwicklung neue, grundlegende Herausforderungen an die Baukastenentwicklung (vgl. Kapitel 2.1.2). Gerade in der sogenannten Frühen Phase der PGE – Produktgenerationsentwicklung nach Albers, die sich durch eine hohe Unsicherheit auszeichnet, gilt es den Umgang mit der vorherrschenden Komplexität zu ermöglichen (vgl. Kapitel 2.3.2). Hierin liegt der Grundstein, um dem Baukasten kontinuierlich innovative Produktkonzepte und Funktionalitäten zuzuführen und diese beim Deduktionsprozess von Produkten aus dem Baukasten frühzeitig validieren zu können (vgl. Kapitel 2.4.2).

### 1.1 Fokus der Arbeit

Die vorliegende Forschungsarbeit fügt sich als fester Bestandteil in den, vom Verständnis der KaSPro - Karlsruher Schule für Produktentwicklung - aufgespannten, methodischen und modelltheoretischen Rahmen. Insbesondere sind sämtliche Begrifflichkeiten und Definitionen entsprechend der Ontologie der KaSPro - Karlsruher Schule für Produktentwicklung zu verstehen.

Der inhaltliche Fokus der Arbeit liegt auf der Frühen Phase der Entwicklung mechatronischer Steuergeräte.<sup>2</sup> Hierbei werden zunächst die Einflüsse der spezifischen Herausforderungen in der Entwicklung mechatronischer Steuergeräte auf die Baukastenentwicklung untersucht. Auf dieser Basis neue, menschenzentrierte Methoden zur Strukturierung und Anwendung von Baukästen für mechatronische

---

<sup>2</sup> Um die ganzheitliche Betrachtungsweise der an der Entwicklung elektronischer Steuergeräte beteiligten Domänen im Rahmen der vorliegenden Arbeit zu unterstreichen, wurde anstatt dessen der Begriff der mechatronischen Steuergeräte eingeführt (vgl. Kapitel 2.1).



Steuergeräte abgezielt. Mit diesen sollen vorwiegend die Produktentwickler<sup>3</sup> aber auch andere in der Entwicklungspraxis beteiligte Stakeholder, durch den gezielten Einsatz des Ansatzes der Modellbasierten Systementwicklung (MBSE) beim Induktions- und Deduktionsprozess unterstützt werden. Als wesentliche Grundlagen dienen die Vorarbeiten aus der KaSPro - Karlsruher Schule für Produktentwicklung in den Bereichen der Baukastenentwicklung (Kapitel 2.4), des Model Based Systems Engineering (Kapitel 2.2), der Mechatronischen Systeme (Kapitel 2.1) und der PGE – Produktgenerationsentwicklung (Kapitel 2.3). Im Speziellen werden der Induktions- und Deduktionsprozess, auf Basis des von Bursac (2016) abgeleiteten Makro-Prozesses zur Unterstützung der Baukastenentwicklung in der Frühen Phase (Kapitel 2.4.2), zielgerichtet auf die Bedarfe der mechatronischen Steuergeräteentwicklung dilatiert. Aufbauend auf dem Modell der PGE werden durch die Anwendung des so entstandenen MBSE-gestützten Baukastenmodells Wege aufgezeigt, wie der Baukasten aktiv zu dessen Weiterentwicklung genutzt werden kann. Abbildung 1.1 zeigt die wesentlichen Forschungsfelder der vorliegenden Arbeit im Überblick:

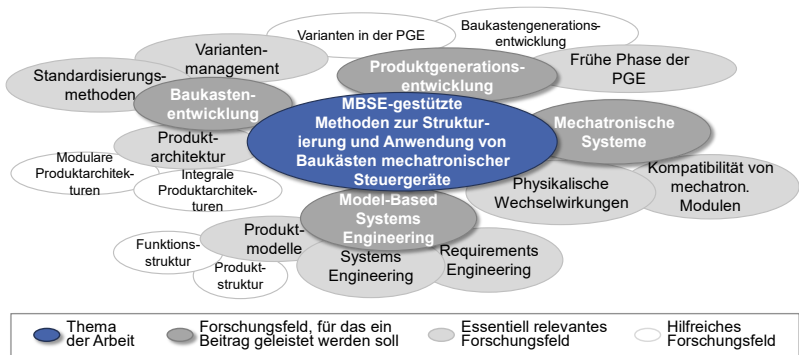


Abbildung 1.1: Darstellung des Forschungsfeldes nach Blessing & Chakrabarti (2009) im Kontext der Ontologie der KaSPro - Karlsruher Schule für Produktentwicklung

<sup>3</sup> Um die Lesbarkeit der vorliegenden Arbeit zu erhöhen, wird ausschließlich das generische Maskulinum verwendet. Dies bezieht sich auf Personen beiderlei Geschlechts.

## 1.2 Aufbau der Arbeit

Im folgenden Abschnitt wird der inhaltliche Aufbau der vorliegenden Arbeit erläutert (Abbildung 1.2):

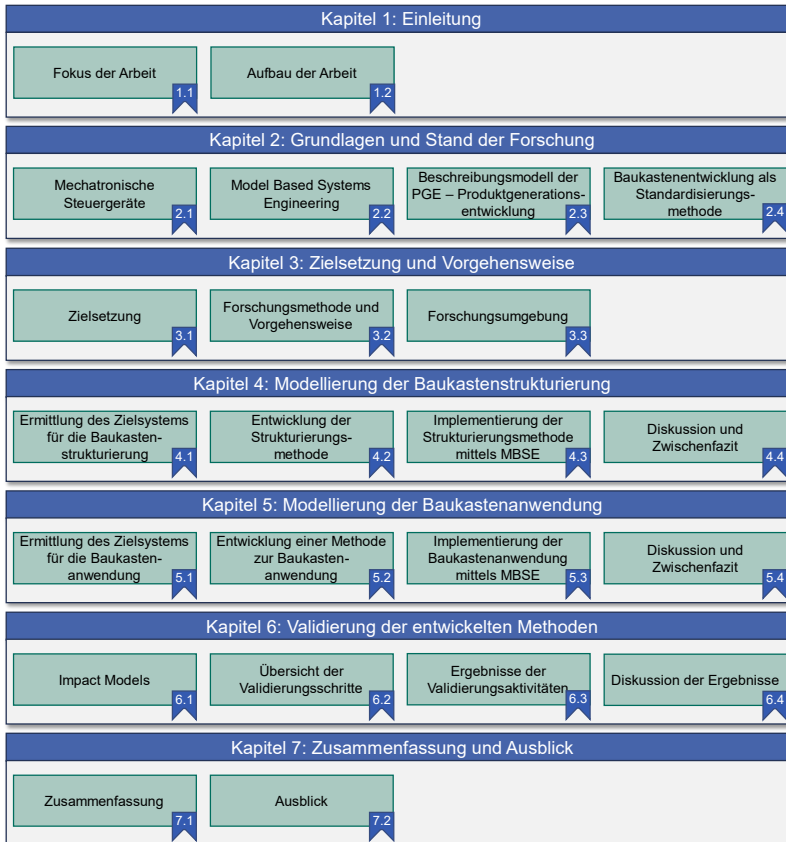


Abbildung 1.2: Elemente und Zusammenhänge des IPEK-X-in-the-Loop-(XiL)-Ansatzes (Albers & Düser, 2010).

In **Kapitel 2** werden dem Leser die Grundlagen, der für diese Arbeit wesentlichen Forschungsfelder präsentiert und eine Übersicht des aktuellen Forschungsstandes vermittelt. Zu Beginn werden die Eigenschaften der mechatronischen Steuergeräte

beleuchtet und diese entsprechend der Verortung in Kleinantrieben im Automobilbereich präzisiert. Im Anschluss wird der Ansatz des Model Based Systems Engineering vorgestellt. In diesem Zusammenhang werden sowohl system- und modelltheoretische Erkenntnisse als auch geltende Prozessmodelle und Produktarchitekturmodelle - diskutiert. Darauf aufbauend wird in das Modell der PGE – Produktgenerationsentwicklung nach Albers mit Fokus auf der Frühen Phase eingeführt. Abschließend werden die Grundlagen zur Baukastenentwicklung dargelegt. Besonderes Augenmerk liegt hierbei auf MBSE-gestützten Ansätzen mit Schwerpunkt auf dem, im Rahmen der KaSPro – Karlsruher Schule für Produktentwicklung entwickelten Framework (Albers, Scherer, Bursac & Rachenkova, 2015).

**Kapitel 3** dient der Darlegung des auf Grundlage des Standes der Forschung ermittelten Forschungsbedarfes. Dieser wird zur Zielsetzung der vorliegenden Arbeit kondensiert und die zugehörigen Forschungshypothesen mittels Forschungsfragen operationalisiert. Anschließend wird das Forschungsvorgehen in Anlehnung an die Design Research Methodology (DRM) vorgestellt und die zur Erhebung empirischer Daten verwendeten Forschungsmethoden illustriert. Zum Abschluss des Kapitels wird die der Arbeit zugrundeliegende Untersuchungsumgebung genauer beleuchtet.

**Kapitel 4** umfasst die Entwicklung und Implementierung der Methode zur Baukastenstrukturierung basierend auf den Konzepten und Modellen der KaSPro – Karlsruher Schule für Produktentwicklung. Entsprechend der DRM wird das Zielsystem der Methode zunächst mittels Durchführung von Interviews empirisch ermittelt, um Schlüsselfaktoren für die methodische Unterstützung herauszuarbeiten. Daraufhin wird die Strukturierungsmethode in ihren Grundzügen theoretisch hergeleitet und ein Regelwerk zu deren Anwendung ermittelt. Die Vorstellung der Implementierung der Strukturierungsmethode unter Verwendung von MBSE Tools entsprechend den Randbedingungen der Untersuchungsumgebung erfolgt im Anschluss.

In **Kapitel 5** wird die Entwicklung und Implementierung der Methode zur Baukastenentwicklung thematisiert. Hierzu werden analog zu Kapitel 4 zunächst die Schlüsselfaktoren des Zielsystems an die Methode ermittelt. Anschließend erfolgt die Identifikation konkreter Use-Cases der einzelnen Stakeholdergruppen für die Interaktion mit der, im Rahmen der Implementierung entwickelten Toolkette, zur teilautomatisierten Baukastenentwicklung. Darauffolgend werden die zur Prüfung der internen und externen Kompatibilität entwickelten Mechanismen dargelegt. Die abschließende Ausführung der Implementierung innerhalb der

Untersuchungsumgebung orientiert sich an den abgeleiteten Use-Cases der Stakeholder.

**Kapitel 6** dient der Vorstellung, der zur Validierung der entwickelten Methoden durchgeführten Aktivitäten und deren Ergebnisse. Die zur Validierung herangezogenen, messbaren Erfolgskriterien werden hierzu, in sogenannten Impact Models in Bezug gesetzt.

In **Kapitel 7** wird eine Zusammenfassung der vorliegenden Arbeit, sowie ein Ausblick auf die Anknüpfung zukünftiger Forschungsvorhaben gegeben.

## 2 Grundlagen und Stand der Forschung

Im Stand der Forschung werden nacheinander, die im Bereich der in Kapitel 1.2 bereits eingeführten Schwerpunktthemen (vgl. Abbildung 1.1) als Basis und Ausgangspunkt der Forschung, im Rahmen der vorliegenden Arbeit diskutiert. Zu Beginn (Kapitel 2.1) wird eine Definition des Begriffs der mechatronischen Steuergeräte vorgenommen und an die spezifischen Rahmenbedingungen bei deren Einsatz in Kleinantrieben im Automobilbereich herangeführt. Kapitel 2.2 umfasst die Einführung in das Gebiet des Model Based Systems Engineering (MBSE) mit besonderem Fokus auf den Erkenntnissen der System- und Modelltheorie. Auf dieser Basis werden anschließend das Modell der PGE – Produktgenerationsentwicklung nach Albers (Kapitel 2.3) und die Baukastenentwicklung im Rahmen der Standardisierungsmethoden (Kapitel 2.4) beleuchtet.

### 2.1 Mechatronische Steuergeräte

Bei **elektronischen Steuergeräten** handelt es sich um eine spezifische Klasse **mechatronischer Systeme**, deren Hauptfunktion in der Informationsverarbeitung und Bereitstellung von Ansteuersignalen für Aktoren liegt. In den folgenden Abschnitten wird zunächst in das Gebiet der Mechatronik eingeführt und ein Überblick der unterschiedlichen **Klassen mechatronischer Systeme** gegeben. Anschließend erfolgt die Einordnung des Begriffes der **mechatronischen Steuergeräte**<sup>1</sup> in den Kontext **elektrischer Kleinantriebe im Einsatz im Automobilbereich**, was dem Untersuchungsgegenstand der vorliegenden Arbeit entspricht. Zum Abschluss werden die spezifischen **Herausforderungen im Entwicklungsprozess** mechatronischer Steuergeräte umrissen.

---

<sup>1</sup> Der Begriff mechatronisches Steuergerät stellt im Rahmen dieser Arbeit, eine sprachliche Präzisierung der allgemein anerkannten Produktklasse der elektronischen Steuergeräte dar.

## 2.1.1 Mechatronische Systeme

Kaum ein Begriff hat den technologischen Fortschritt unserer Gesellschaft in den letzten Jahrzehnten so geprägt wie der, der Mechatronik. Seit der initialen Wortschöpfung im Jahre 1969 durch das japanische Unternehmen Yaskawa (Harashima, Tomizuka & Fukuda, 1996) ist die Mechatronisierung und der damit einhergehende Funktionszuwachs moderner<sup>2</sup> Produkte nicht mehr wegzudenken. Dank komplexer, mechatronischer Systeme sind technologische Fortschritte wie bspw. autonom manövrierende PKWs in erreichbare Nähe gerückt (eNOVA, 2016; Gausemeier, Anacker, Czaja, Wassmann & Dumitrescu, 2013; Gausemeier, Dumitrescu, Echterfeld, Pfänder, Steffen & Thielemann, 2019). Vor diesem Hintergrund, so möchte man meinen, herrscht ein klares Bild davon was in der Wissenschaft unter dem Begriff der Mechatronik zu verstehen ist. Tatsächlich existiert jedoch bis heute keine einheitliche Definition. Im Gegenteil, es wurden über die Jahre zahlreiche Definitionen abgefasst, die unter anderem sehr stark an spezifische Problemstellungen angelehnt sind. (Bolton, 2003; Isermann, 1999; Kyura & Oho 1996) Die Vielzahl an unterschiedlichen Definitionen zum Thema Mechatronik lassen sich jedoch zu einer Kernaussage subsumieren, welche aus der Definition nach Harashima et al. (1996) ersichtlich wird. Diese wurde später vom VDI (VDI 2206, 2004) in der folgenden Form übernommen:

### Definition 1: Mechatronik

[Mechatronik ist] „das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse, sowie bei der Prozessgestaltung“

Das synergetische Wechselspiel bei der Integration disziplin-spezifischer Konzepte führt dazu, dass mechatronische Produkte, neben einem Funktionszuwachs, auch deutliche Vorteile in Sachen Effizienz und Kosten gegenüber klassischen, rein mechanischen Lösungen aufweisen können (Qamar, During & Wikander, 2009; Isermann, 1999). Man spricht bei dieser Art des Zusammenwirkens auch vom Emergenzbegriff, welcher im Rahmen der Systemtheorie (vgl. Kapitel 2.2.1) weiter ausgeführt wird. In realen Produktentstehungsprozessen entstehen bei der Kombination von Software, Elektronik und Mechanik allerdings auch Effekte, die sich nachteilig auf die Funktionalität eines Produktes auswirken können (Göpfert,

---

<sup>2</sup> Nach Albers sind moderne technische Systeme immer auch mechatronische Systeme. (Albers im Vorwort zu Oerding, 2009)

1998). Diese Effekte, zu denen bspw. die Phänomenologie rund um die elektromagnetische Verträglichkeit (EMV) zählt, können sich unter emergenten Gesichtspunkten negativ gestalten (Göpfert, 1998). Negativ-emergente Effekte führen in der Entwicklung mechatronischer Systeme zu einem deutlichen Anstieg der Produktkomplexität (Powelske et al., 2017). Darüber hinaus erhöht sich die Komplexität des gesamten Entwicklungsprozesses durch die erforderliche fachdisziplin-übergreifende Diskussion und Abstimmung bei der Problemlösung. Der dieser Arbeit zugrundeliegende Komplexitätsbegriff orientiert sich an der Definition von Weber (2005), welcher die Anzahl von Systemelementen und deren Beziehungen als Messgrößen der Komplexität ausweist. Die klassische Entwicklungsmethodik stößt im Umgang mit der erhöhten Komplexität von mechatronischen Produkten und deren Entwicklungsprozess an ihre Grenzen (Albers & Braun, 2011a). Zur Abbildung realer Produktentstehungsprozesse<sup>3</sup> mechatronischer Produkte bedarf es neben einer entsprechenden Ontologie vor allem einer ganzheitlichen Betrachtungsweise. Hierin liegt ein Kernaspekt der Forschung der KaSPro – Karlsruher Schule für Produktentwicklung (Albers im Vorwort zu Gladysz, 2019). Eine Übersicht verschiedener Prozessmodelle findet sich in Kapitel 2.2.3. Die konkrete Umsetzung der mechatronischen Prinzipien in Form technischer Elemente wird im Folgenden anhand mechatronischer Systeme<sup>4</sup> im Automobil erläutert:

Wie aus Abbildung 2.1 ersichtlich wird, ist das Spektrum unterschiedlicher Systeme, die auf dem Zusammenwirken mechanischer, elektronischer und informationstechnischer Komponenten beruhen, im Automobil bereits vielfältig. Neben zahlreichen Steuergeräten findet sich eine Vielzahl elektrischer Antriebe (Wallentowitz & Reif, 2011; Reuss, Diesner & Thoben, 2004). Die Steuergeräte und Antriebe sind wiederum mit anderen mechatronischen Komponenten aus dem Bereich der Sensorik und Aktorik zu komplexeren mechatronischen Systemen z.B. einem elektrischen Lenksystem zusammengesetzt (Isermann, 1999; Czichos, 2015; Reif, 2011). Im Fokus der einzelnen Systeme steht dabei die Erfüllung gewisser Funktionalitäten für den Nutzer bzw. den Kunden des Automobilherstellers (Albers, Haug, Heitger, Fahl & Hirschter, 2019). Vor diesem Hintergrund lässt sich auch das Automobil selbst, als eigenes mechatronisches System mit festgelegtem

---

<sup>3</sup> Nach Albers & Gausemeier (2012) ist die Produktentstehung als Teil des Produktlebenszyklus definiert und beschreibt den grundsätzlichen Ablauf von der Produkt- bzw. Geschäftsidee bis zum Serienanlauf.

<sup>4</sup> Eine detaillierte Einführung in die Systemtheorie und deren Ontologie erfolgt in Kapitel 2.2.1

Funktionsumfang betrachten. Dieses ist durch die zunehmende Vernetzung, beispielsweise über Cloudsysteme wiederum Teil übergeordneter Kommunikationssysteme (Dumitrescu et al., 2021). Anhand dieser Charakteristik wird deutlich, dass mechatronische Systeme gleichzeitig isoliert als auch in Form eines Bestandteils eines oder mehrerer übergeordneter mechatronischer Systeme betrachtet werden können<sup>5</sup>. In der Wissenschaft spricht man hierbei vom sogenannten **System-of-Systems (SoS)** Ansatz (Maier, 1998).

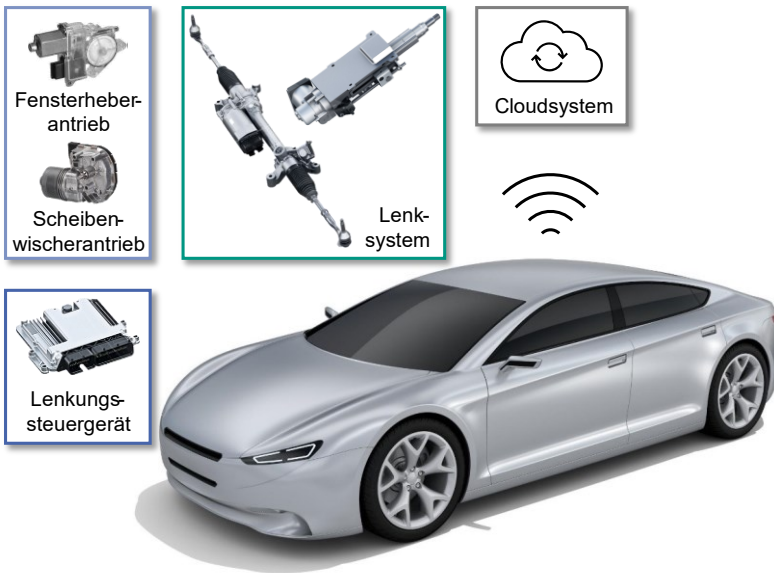


Abbildung 2.1: Mechatronische Systeme im Automobil. Eigene Darstellung in Anlehnung an Isermann (1999), Czichos (2015), Reif (2011). Erstellt unter Verwendung von Darstellungen der Robert Bosch GmbH (Robert Bosch GmbH, 2022)

Komplexe SoS äußern sich durch besondere Herausforderungen im Produktentstehungsprozess (Albers, Reussner, Kurrle, Burger, Moeser, Bursac, Klingler &

---

<sup>5</sup> Diese Eigenschaft wird im Zusammenhang mit der Systemtheorie auch als hierarchisches Konzept von Systemen bezeichnet Kapitel 2.2.1



Behrendt, 2017) sowohl entwicklungsseitig (Albers, Peglow, Powelske, Birk & Bursac, 2018) als auch validierungsseitig (Albers, Mandel, Yan & Behrendt, 2018). Abhängig von der Wahl der Systemgrenze des mechatronischen Systems variiert das Verständnis seines mechanischen Anteils (Isermann, 1999; Czichos, 2015; Reif, 2011). Da, nach Albers (2010) jeder Produktentstehungsprozess individuell und einzigartig ist, verschiebt sich ebenfalls der Schwerpunkt der Entwicklungstätigkeiten in Abhängigkeit der Systemgrenze. Betrachtet man Steuergeräte isoliert so werden diese von den Entwicklungsartefakten der Software und der Halbleiterindustrie bestimmt (Deicke, 2018; Hüning, 2019; Schmidt, Buhlmann, Ficek & Richter, 2012; Schäuffele & Zurawka, 2010). Der mechanische Anteil der Steuergeräte wird von den Technologien der Aufbau- und Verbindungstechnik (AVT) dominiert (Hainzel & Nicolics, 2001; Chamas, Hopfensitz, Obergfell, Oszwald & Traub, 2019; Steffen, 2007)<sup>6</sup>. Bei einem Schweißenwischsystem hingegen, ist das Gesamtsystem deutlich stärker an den mechanischen Eigenschaften des Aktuators orientiert, da dieser den Ort der letztlichen Funktionserfüllung vor Kunde bedeutet (Albers, Haug, Heitger, Fahl & Hirschler, 2019). Das System-of-Systems eines Gesamtfahrzeugs als Teil hochvernetzter Cloudsysteme wiederum hat einen deutlichen Schwerpunkt auf Seiten der informationstechnischen Entwicklung. Handelt es sich bei den Kommunikationsnetzwerken um internet- und plattformbasierte, folglich globale Ansätze („den Cyberspace“) und weisen die Systeme zusätzlich eine hohe interaktive, soziotechnische Integration auf, spricht die aktuelle Forschung von sogenannten **cyber-physischen Systemen**. (Dumitrescu et al., 2021; VDI 2206, 2021)

Trotz der großen Vielfalt an mechatronischen Systemen wohnt allen eine gleiche Referenzarchitektur inne. In ihrem Zentrum liegt der aus den vier Elementen: Grundsystem, Sensoren, Aktoren und Informationssystem bestehende Regelkreis (VDI 2206, 2021; Isermann, 1999; Czichos, 2015; Reif, 2011; Wallaschek, 1995). Einflüsse der Systemumgebung werden über die Sensorik detektiert. Zur Kommunikation mit anderen technischen Systemen als auch zur direkten Mensch-Maschine-Interaktion existieren optionale Schnittstellen. Die Leistungsversorgung der Aktorik kann sowohl intern als auch extern erfolgen. (VDI 2206, 2021) Einzelne mechatronische Systeme können dabei wiederum aus mechatronischen Systemen

---

<sup>6</sup> Wird im weiteren Verlauf der vorliegenden Arbeit von der Mechanik im Kontext der Steuergeräteentwicklung gesprochen, so ist damit immer der Anteil der Aufbau- und Verbindungstechnik gemeint.

(auch anderer Klassen) bestehen. Abbildung 2.2 zeigt die Grundstruktur eines mechatronischen Systems in Anlehnung an die VDI Norm 2206 (2021):

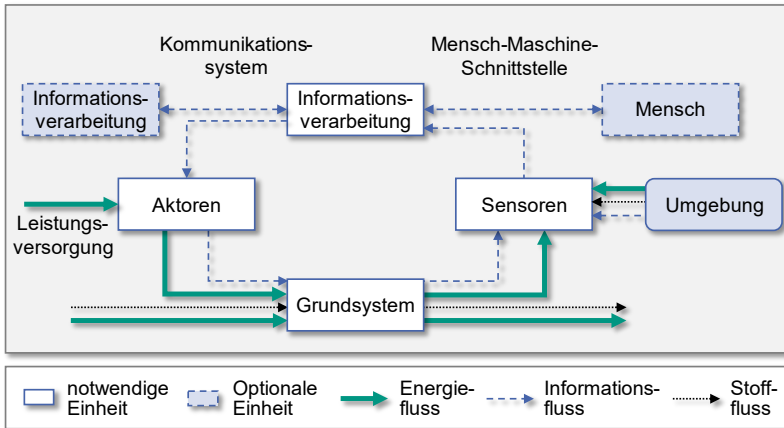


Abbildung 2.2: Grundstruktur mechatronischer Systeme. Darstellung in Anlehnung an die Neuauflage der VDI 2206 im November 2021 (VDI, 2021)

Nachfolgend werden die Elemente der Grundstruktur (VDI 2206, 2021; Isermann, 1999; Czichos, 2015; Reif, 2011; Wallaschek, 1995) am Beispiel eines mechatronischen Steuergerätes im Kontext ihres Einsatzes in Kleinantrieben erläutert:

Bei dem **Grundsystem** handelt es sich um eine mechanische, elektromagnetische, hydraulische oder pneumatische Struktur. Im Falle des Steuergerätes fällt beispielsweise das bestückte Substrat (in der Regel eine klassische Leiterplatte), unter die Elemente des Grundsystems. Zur Bestimmung ausgewählter Zustandsgrößen<sup>7</sup> des Grundsystems bedient man sich der **Sensorik** (Czichos,

---

<sup>7</sup> Zustandsgröße: physikalische Größe durch deren Wert zu einem beliebigen Zeitpunkt  $t_0$  der Ablauf des Systems für  $t > t_0$  eindeutig bestimmt ist, sofern die Eingangsgrößen des Systems für  $t > t_0$  gegeben sind (VDI 2206, 2004 dort in Anlehnung an Föllinger & Dörrscheidt, 1994)

2015; Isermann, 1999; VDI 2206, 2021; Stölting, Kallenbach & Amrhein, 2011). In handelsüblichen Steuergeräten für Kleinantriebe finden sich beispielsweise Sensoren zur Bestimmung der Rotorlage, der Temperatur und zur Bestimmung von Strömen und Spannungen. Auf Basis der erhobenen Sensorsignale werden in der **Informationsverarbeitung** die erforderlichen Einwirkungen auf das System bestimmt (Czichos, 2015; Isermann, 1999; VDI 2206, 2021; Stölting, Kallenbach & Amrhein, 2011). Diese dienen dazu, die Zustandsgrößen in gewünschter Form anzupassen. Im Steuergerät erfolgt die zentrale Informationsverarbeitung digital im Mikrocontroller (Wallaschek, 1995). In sicherheitskritischen Anwendungen werden zusätzlich Überwachungsfunktionen für die Controlleraktivität in andere informationsverarbeitende Bausteine ausgelagert (Abouda et al., 2021). Im Bereich der intelligenten Sensorik herrscht zunehmend das Bestreben die Digitalisierung und Interpretation der Sensordaten direkt am Sensor durchzuführen. Entsprechende, dadurch ungenutzte Kapazitäten des Mikrocontrollers können somit für andere Berechnungen herangezogen werden. (Czichos, 2015; Isermann, 1999; VDI 2206, 2021; Stölting, Kallenbach & Amrhein, 2011). Die Aufgabe der **Aktorik** besteht in der Umsetzung der mittels Informationsverarbeitung festgelegten Einwirkungen auf das Grundsystem (VDI 2206, 2021; Isermann, 1999; Czichos, 2015). Im Rahmen der Kleinantriebe sind dies die unterschiedlichen Arten an Elektromotoren (Stölting, Kallenbach & Amrhein, 2011). Diese werden auf Basis komplexer Algorithmen über die Leistungsendstufe der Steuerelektronik entsprechend mit Strom und Spannung versorgt (Brown, 2001; Heumann, 1996). Somit können Größen wie beispielweise Momenten- und Drehzahlverlauf sowie die Rotorposition an den gewünschten Wert angepasst werden. Betrachtet man das Steuergerät isoliert vom Motor, so kann die Leistungselektronik ebenfalls als Aktor (Beeinflussung der Zustandsgrößen Strom und Spannung) betrachtet werden (VDI 2206, 2004).

Die Beziehungen zwischen den Bestandteilen eines mechatronischen Systems werden in Anlehnung an Pahl, Beitz, Feldhusen & Grothe (2007) in Form von Energie-, Informations- und Stoffflüssen dargestellt. Je nach Ausprägung eines mechatronischen Systems ergeben sich wie bereits erwähnt unterschiedliche Herausforderungen in dessen Entwicklung (VDI 2206, 2021; Isermann, 1999; Czichos, 2015, Wallaschek, 1995). In diesem Kontext leitet Steffen (2007) sogenannte „Grundtypen von Entwicklungsaufgaben“ für mechatronische Systeme ab. Hier exemplarisch am Grundtyp der mechatronischen Funktionsmoduli dargestellt (Abbildung 2.3):

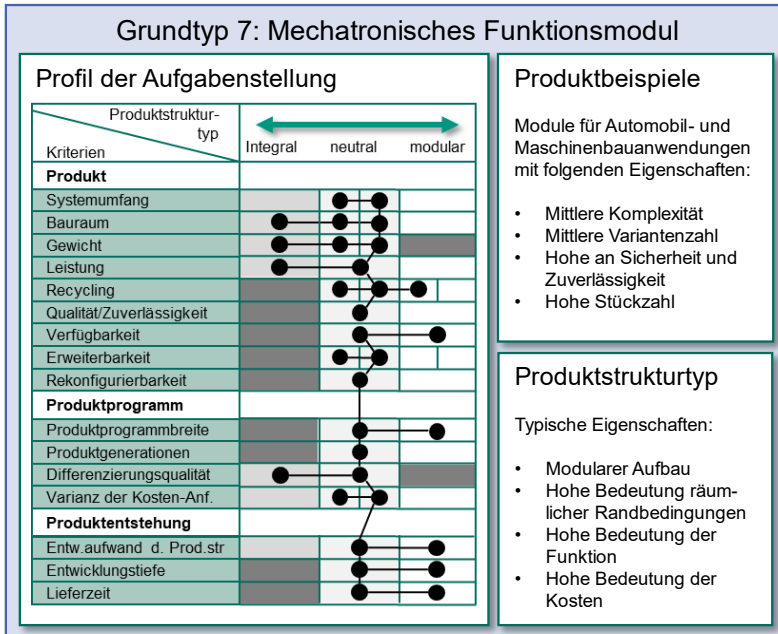


Abbildung 2.3: Übersicht des Grundtyps mechatronischer Funktionsmoduli nach Steffen (2007). Geordnet nach Profil der Aufgabenstellung, Produktbeispielen und des resultierenden Produktstrukturtyps.

Innerhalb der definierten Grundtypen werden die jeweiligen produktbestimmenden Eigenschaften, der Produktstrukturtyp und die reale Umsetzung gemäß festgelegten Kriterien beschrieben. Für die Forschungsaktivitäten im Rahmen dieser Arbeit liefern die Eigenschaften des Produktstrukturtypen, der nach Steffen definierten mechatronischen Funktionsmoduli, als hilfreiche Zusammenfassung des bestehenden Standes der Forschung, eine visuelle Grundlage. In Kapitel 2.2.4 dieser Arbeit werden diese Eigenschaften näher beleuchtet.

Nachdem die Grundbegriffe der Mechatronik und eine Einteilung mechatronischer Systeme dargelegt sind, widmet sich der nachfolgende Abschnitt der Begriffsdefinition, dem Aufbau und dem Entwicklungsprozess mechatronischer Steuergeräte.

## 2.1.2 Grundlagen mechatronischer Steuergeräte

Der Begriff des mechatronischen Steuergerätes wird im Rahmen dieser Arbeit anstelle der allgemein anerkannten Bezeichnung für elektronische Steuergeräte (engl. ECU = electronic control unit) verwendet. Die Substitution von „elektronisch“ durch „mechatronisch“ geschieht in der Absicht, den Fokus auf die ganzheitliche, disziplin-übergreifende Betrachtung des Steuergeräts in seinem Produktentstehungsprozess als mechatronisches System zu legen. Zunächst wird ein Überblick über die bestehende Bedeutung und Definition elektronischer Steuergeräte gegeben.

Über die letzten 40 Jahre der Automobilentwicklung nimmt die Funktionsdichte im Fahrzeug immer weiter zu. (Reif, 2011; Automobil-Elektronik, 2011) Neben der Umsetzung neuer bisher fahrzeug-fremder Funktionen erfolgt die Substitution bisher rein mechanischer Steuer- und Regelfunktionen durch „intelligente“ elektronische Einheiten, sogenannte „elektronische Steuergeräte“. Die Einsatzzwecke der Steuergeräte sind dabei mannigfaltig. So kontrollieren bis zu 100 verschiedene Steuergeräte (Broy, 2013; Wallentowitz & Reif, 2011) im Mittel über 200 elektrische Komponenten (Reuss, Diesner & Thoben, 2004) innerhalb eines Mittelklassefahrzeuges. Sie übernehmen beispielsweise die Auslösung und Überwachung von Airbags, dienen der Bequemlichkeit in Form automatischer Sitzeinstellungen, kontrollieren die Zündung des Motors, oder steuern die Multimedia- und Kommunikationseinrichtungen. Trotz der großen Bedeutung für die Funktionsbereitstellung im Kraftfahrzeug und der großen Bandbreite unterschiedlichster Steuergeräte existieren in der wissenschaftlichen Literatur unterschiedliche Definitionen. Während die Betrachtungen von Gausemeier et al. (2014) Steuergeräte, einer Klasse mechatronischer Systeme zuordnen und damit als eigenständige mechatronische Systeme ansehen, weisen diverse andere Quellen (Deicke, 2018; Hüning, 2019; Schmidt, Buhlmann, Ficek & Richter, 2012) die Steuergeräte lediglich als in mechatronische Systeme „eingebettete Systeme“<sup>8</sup> (engl. Embedded Systems) aus. Gemäß letzterer Sichtweise werden die Steuergeräte lediglich der Informationsverarbeitung aus der allgemeinen Definition mechatronischer Systeme gleichgesetzt. Diese Sichtweise lässt sich anhand folgender, weit verbreiteter Definition elektronischer Steuergeräte illustrieren (Reif, 2011, S.198):

---

<sup>8</sup> Eine Definition für eingebettete Systeme liefert Weik (2000).

**Definition 2: Elektronisches Steuergerät/Electronic Control Unit (ECU)<sup>9</sup>**

Das Steuergerät empfängt die elektrischen Signale der Sensoren, wertet sie aus und berechnet die Ansteuersignale für die Stellglieder (Aktoren). Das Steuerungsprogramm – die „Software“ – ist in einem Speicher abgelegt. Die Ausführung des Programms übernimmt ein Mikrocontroller. Die Bauteile eines Steuergerätes werden als „Hardware“ bezeichnet.

Unabhängig von den unterschiedlichen systemischen Betrachtungsweisen des elektronischen Steuergerätes herrscht in der Literatur gegenüber seinen Funktionalitäten jedoch Konsens (Deicke, 2018; Hüning, 2019; Schmidt, Buhlmann, Ficek & Richter, 2012). Die Wissenschaft ist sich darüber einig, dass die Steuerung externer Aktoren und damit der unmittelbare Zweck elektronischer Steuergeräte maßgeblich von den Entwicklungsartefakten der Software bestimmt wird (Deicke, 2018; Hüning, 2019; Schmidt, Buhlmann, Ficek & Richter, 2012; Schäuffele & Zurawka, 2010). Fakten belegen beispielsweise, dass die Gesamtheit aller Steuergeräte eines Fahrzeuges ca. 100 Millionen Zeilen (Broy, 2013) ausführbaren Softwarecode umfasst. Neben der Steuergerätesoftware spielt die Halbleitertechnologie eine weitere zentrale Rolle für die Steuergeräte- und Automobilentwicklung. Aktuell werden ca. 80% der Automobilinnovationen durch mechatronische Systeme unter Verwendung von Halbleitern bestritten. (eNOVA, 2016) Diese werden hauptsächlich über die Technologieentwicklung in den Bereichen der leistungselektronischen Bauteile und der Mikrocontroller erwirkt. (eNOVA, 2016) Neben der eigentlichen Bauteilentwicklung zählt die Aufbau- und Verbindungstechnik (AVT) zu den Kerntätigkeiten im Bereich der Halbleitertechnik. Der Aufgabenbereich der AVT umfasst, ihrem Namen entsprechend, sämtliche Tätigkeiten der mechanischen und elektrischen Verbindung elektronischer Bauteile (Hainzel & Nicolics, 2001; Chamas, Hopfensitz, Oberfell, Oszwald & Traub, 2019). Hierzu gehört neben der Substratbestückung ebenfalls die Entwicklung applikationsspezifischer Entwärmungskonzepte. Die Auswahl der AVT hat einen direkten Einfluss auf die zu verwendende Fertigungstechnologie und das umfassende Produktkonzept. Sie wird somit zu einem der entscheidenden Kostentreiber der Steuergeräteentwicklung (Becker, 2007; Steffen, 2007). Im Gegensatz zu der isolierten Betrachtung von Funktionen in der Softwareentwicklung spielt bei der Entwicklung und Umsetzung von AVT-Konzepten, als mechanischer Anteil der Mechatronik eines Steuergerätes, der Funktions-Gestalt-Übergang eine maßgebliche Rolle. Im Kontext dieser Arbeit wird die AVT folglich dem

---

<sup>9</sup> Abweichend auch als „electronically controlled unit“ bezeichnet (Reif, 2011)

Zuständigkeitsbereich der Mechanikentwicklung im Produktentstehungsprozess mechatronischer Steuergeräte zugeordnet. Ergo ist die Erfüllung der von Software dominierten Hauptfunktionalität eines elektronischen Steuergerätes in großem Maße von den elektrotechnischen und mechanischen Entwicklungskonzepten abhängig (Becker, 2007; Schäuffele & Zurawka, 2010). Die Entwicklung eines solchen, von interdisziplinären Wechselwirkungen bestimmten Systems, erfordert ganzheitliche Ansätze und demnach spezifische Entwicklungsmethoden (Albers im Vorwort zu Gladysz, 2019; Albers, Matthiesen, Bursac, Moeser, Klingler, Schmidt, Munker et al., 2016). Der Begriff des mechatronischen Steuergerätes soll für eben diesen Umstand sensibilisieren.

Im Folgenden wird der Fokus der vorliegenden Arbeit, entsprechend dem zugrundeliegenden Untersuchungsgegenstand, auf die mechatronischen Steuergeräte für den Einsatz in elektrischen Kleinantrieben im Automobilbereich gelegt. Hierzu wird zunächst ein allgemeiner Überblick über die Charakteristika elektrischer Kleinantriebe gegeben. Im Anschluss werden der Aufbau und die Herausforderungen im Produktentstehungsprozess der dort eingesetzten mechatronischen Steuergeräte beleuchtet.

### **Verortung im Rahmen elektrischer Antriebe**

Elektrische Antriebssysteme<sup>10</sup> (engl. Electrical/electromagnetic drive system EDS) dienen der Erzeugung mechanischer Bewegungsenergie (Stölting, Kallenbach & Amrhein, 2011; Binder, 2012). Im Kern bestehen sie aus einer **informationsverarbeitenden Steuereinheit** und einem **energieumformenden Teilsystem** zur Wandlung elektrischer Energie in mechanische Energie. Das informationsverarbeitende Teilsystem beinhaltet die **Steuereinheit** und ein, über Sensoren realisiertes **Messsystem**. Die Bestandteile des energieumformenden Teilsystems sind das **Stellelement** zur elektrischen Energiewandlung, das **Antriebselement** zur elektromechanischen Energieumformung und das **Übertragungselement** zur Wandlung der mechanischen Energie (Schröder, 2013). Außerhalb des Antriebssystems befindet sich das **Wirkelement**, welches letztlich die gewünschte Bewegung ausführt (Stölting, Kallenbach & Amrhein, 2011; Binder, 2012; Schröder & Böcker, 2021).

---

<sup>10</sup> In der Literatur häufig auch als elektromagnetische Antriebssysteme bezeichnet (Schulze, 2008; Stölting, Kallenbach & Amrhein, 2011)

Abbildung 2.4 zeigt die Elemente eines elektrischen Antriebssystems am Beispiel eines Scheibenwischersystems:

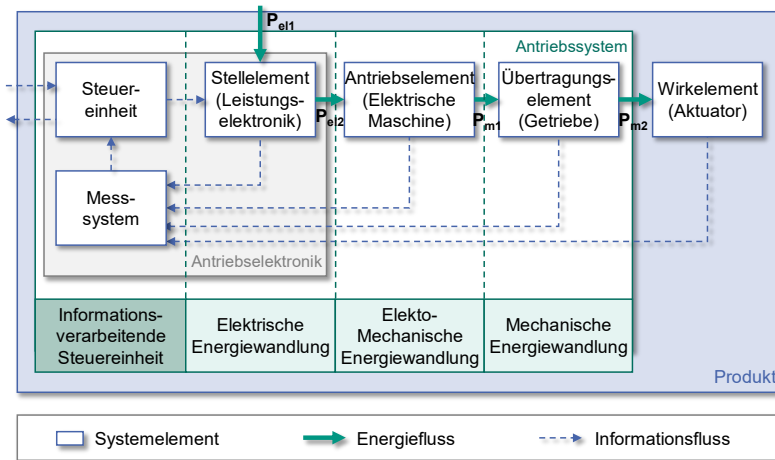


Abbildung 2.4: Elemente eines elektrischen Antriebssystems am Beispiel eines Scheibenwischersystems. Eigene Darstellung in Anlehnung an Stöltling, Kallenbach & Amrhein (2011).

Innerhalb des Antriebssystems sind das **informationsverarbeitende Teilsystem** und die elektrische Energiewandlung durch das mechatronische Steuergerät realisiert (Stöltling, Kallenbach & Amrhein, 2011; Binder, 2012; Schröder & Böcker, 2021). Das **Stellelement**, in Form der leistungselektronischen Schaltung, ist hierbei als Akteur des Steuergerätes zu verstehen (Schäuffele & Zurawka, 2010). Das Steuergerät ist in diesem Kontext an der Erfüllung folgender Aufgaben maßgeblich beteiligt:

Informationsverarbeitendes Teilsystem:

- Vergleich der vorgegebenen Zustandsgrößen mit durch Sensoren erfassten, entsprechenden Messwerten
- Bildung sich aus der Abweichung ergebender Stellgrößen
- Erfassung von Störgrößen inklusive Überwachung und Schutz von Antriebssystem und Wirkelement
- Ausgabe von Meldegrößen an übergeordnete Systeme (Kommunikation)



Stellglied (elektrischer Energiewandler):

- Wandlung der elektrischen Energie (bspw. Gleichstrom in Wechselstrom)
- Anpassung der Eingangsspannungen des elektromechanischen Wandlers an die Versorgungsspannung des Antriebssystems
- Umsetzung der Stellgröße in eine, vom Antriebselement nutzbare Form (bspw. Wandlung eines Eingangsstromes in getaktete Phasenströme zum Betrieb eines Elektromotors). Hierbei wird der elektrische Energiefluss (Strom) mit einem Informationsfluss (Taktung) überlagert.

Das Hauptaugenmerk liegt seitens der Funktionalität demnach auf der bedarfsgerechten Ansteuerung des Antriebselementes. Mittels elektromechanischer Wandlung erfolgt dort die Bereitstellung mechanischer Energie in Form einer Translation oder einer Rotation. Diese kann kontinuierlich oder schrittweise wie z.B. in einem Schrittmotor erfolgen (Binder, 2012; Schulze, 2008). An dieser Stelle sei erneut darauf hingewiesen, dass die Bereitstellung der Translation bzw. Rotation, im Falle der Kleinantriebe im Automobilbereich, lediglich den ersten Schritt zur Erfüllung der Funktionalität vor Kunde darstellt (vgl. Kapitel 2.1.1). Diese wird maßgeblich durch die mechanischen Eigenschaften der Wirkelemente (Aktuatoren) beeinflusst. Im Hinblick auf den Forschungsgegenstand der vorliegenden Arbeit werden nachfolgend lediglich diejenigen Antriebssysteme näher betrachtet, welche mittels Rotation und elektromagnetischer Energiewandlung die Erzeugung stetiger Bewegungen gewährleisten. Elektrische Schrittantriebe, piezoelektrische Antriebe oder elektromagnetische Linearantriebe bleiben demnach außen vor. Innerhalb dieser Antriebe<sup>11</sup> werden die sogenannten Dauerläufer, wie beispielsweise in Motorkühlgebläsen angewendet, von den Servoantrieben zur präzisen Winkel-, Drehzahl- oder Beschleunigungsstellung unterschieden (Stölting, Kallenbach & Amrhein, 2011; Binder, 2012; Schröder & Böcker, 2021; Schröder, 2013). Letztere werden aufgrund ihrer Funktion häufig auch als Stellantriebe bezeichnet.<sup>12</sup> Sie zeichnen sich durch besondere Anforderungen an Sensorik und Regelgüte aus (Schulze, 2008; Stölting, Kallenbach & Amrhein, 2011). Prominentes Beispiel dieser Klasse ist ein

---

<sup>11</sup> Im Folgenden auch als E-Antriebe bezeichnet

<sup>12</sup> Eine Verwendung von Schrittmotoren zur präzisen Positionssteuerung ist ebenfalls möglich. Allerdings überwiegen bei hohen Lasten die Nachteile ggü. den Servomotoren. Diese entstehen aus dem Verzicht auf Sensoren und Regelung (Schulze, 2008).

Lenkunterstützungsantrieb für den Automobileinsatz. Beim Einsatz rotierender elektrischer Maschinen ist es möglich, diese die mittels Veränderung von Drehzahl- und Drehmomentenrichtung in verschiedenen Bereichen, sogenannten Quadranten zu betreiben (Abbildung 2.5). Abhängig davon, ob die elektrische Maschine vorwiegend mechanische Bewegungsenergie abgibt (Motorbetrieb), oder zur Wandlung in elektrische Energie aufnimmt (Generatorbetrieb) wird sie in der Literatur als Antriebs- oder Arbeitsmaschine bezeichnet (Binder, 2012). Hierbei sei auf die Definition der Antriebssysteme nach Albers verwiesen, nach der die leistungsübertragende Mechanik der Antriebssysteme grundsätzlich aus den drei Subsystemen: Antriebs- oder Kraftmaschine, Antriebsstrang und Arbeitsmaschine besteht (Albers, 2008).

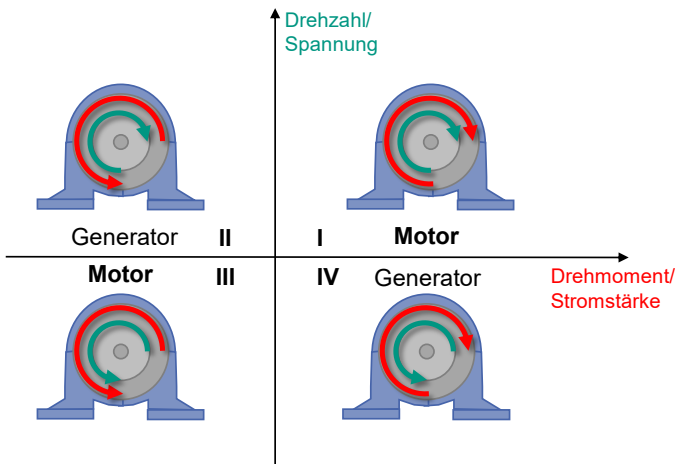


Abbildung 2.5: Vier-Quadranten-Betrieb eines E-Antriebs illustriert anhand der Richtungen von Drehzahl bzw. Spannung und Drehmoment bzw. Stromstärke (eigene Darstellung)

Ein Mehrquadrantenbetrieb ermöglicht es in der Realität zusätzliche Funktionalitäten mit einem Antriebssystem abzubilden (Kremser, 2004; Schröder & Böcker, 2021). So wird beispielsweise in elektrischen Traktionsantrieben im Automobil, neben dem motorischen Betrieb, eine generatorische Rückspeisung elektrischer Energie (Rekuperation) ins Batteriesystem vorgesehen (Tschöke, Gutzmer & Pfund, 2019). Je nach Betriebsart und entsprechendem Verhalten, ergeben sich ergänzende Anforderungen an die Steuergeräte. Um eine detailliertere Auseinandersetzung mit den Steuergeräten der vorliegenden Arbeit zu

ermöglichen, erfolgt im nächsten Schritt die Eingrenzung der elektrischen Antriebssysteme auf die spezifische Gruppe der elektrischen Kleinantriebe für den Einsatz im Automobil. Abbildung 2.6 zeigt hierzu einen exemplarischen Ausschnitt des Portfolios im Automobil eingesetzter, rotatorischer E-Antriebe:

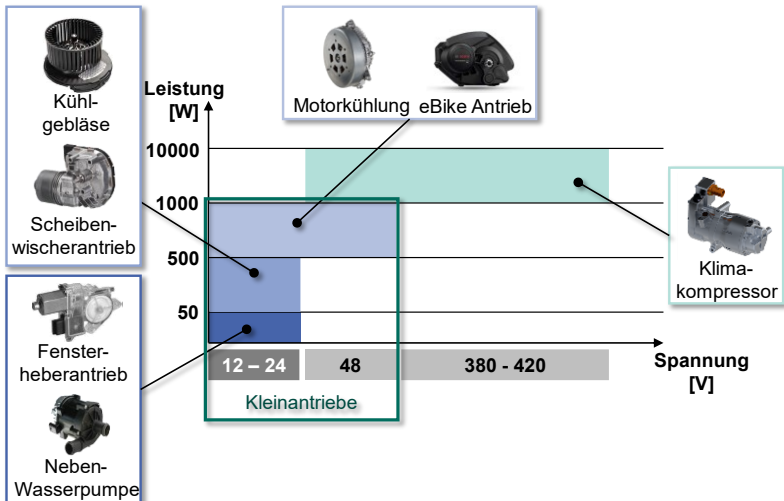


Abbildung 2.6: Übersicht und Einordnung elektrischer Kleinantriebe. Das Diagramm zeigt verschiedene Produkte, in denen die entwickelten Kleinantriebe enthalten sind. Diese sind entsprechend ihrer Betriebsspannung (x-Achse) und deren Ausgangsleistung (y-Achse) geordnet. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022)

Die Einteilung erfolgt in dieser Darstellung anhand der mechanischen Ausgangsleistung des Systems (Ordinate), sowie dessen nominaler Eingangsspannung (Abszisse). Die Eingangsspannungen sind dabei auf Basis der heute gängigen Bordnetzspannungen in drei Bereiche unterteilt. Man unterscheidet die Bereiche von 12V (PKW) und 24V (LKW), 48V (als zweites leistungsfähigeres Bordnetz für PKW) und 400V (als zweites Bordnetz für Elektrofahrzeuge). Sämtliche im Automobil eingesetzten Bordnetze werden dabei ausschließlich mit Gleichspannung betrieben (Reif, 2011). Diese Eigenschaft hat einen großen Einfluss auf den inneren Aufbau der betrachteten Antriebssysteme respektive ihrer Steuergeräte. Ihrer Definition folgend, sind elektrische Kleinantriebe (engl. fractional

horsepower drive) anhand ihrer oberen Leistungsgrenze von ca. 1kW charakterisiert (Stölting, Kallenbach & Amrhein, 2011; Schröder, 2013). Im Rahmen des Forschungsgegenstandes der Steuergeräte für den Einsatz in Kleinantrieben im Automobilbereich wird der betrachtete Spannungsbereich auf Bordnetzspannungen im sogenannten Niedervoltbereich (also 12, 24 und 48 Volt) beschränkt.

Die im Rahmen dieser Arbeit untersuchten Kleinantriebe (rote Markierung) zeichnen sich zusätzlich durch eine außerordentliche Vielfalt ihrer Einsatzgebiete und sehr große Stückzahlen (>1 Mio. Stück pro Jahr) aus. Diese Charakteristika bedingen zugleich eine möglichst kosteneffiziente Fertigung, als auch eine hohe Anpassung an die jeweiligen Anwendungsfälle (Stölting, Kallenbach & Amrhein, 2011; Schröder, 2013). Somit sind sowohl Steuergerät als auch Motor für eine Standardisierung in Form der Baukastenentwicklung (vgl. Kapitel 2.4) prädestiniert. Seitens der Elektromotoren führt dies zu einer weiteren Eingrenzung des Untersuchungsgegenstandes. Die Verwendung mechanisch kommutierter Gleichstrommotoren ist im Bereich der Kleinantriebe im Automobileinsatz gängig (Stölting, Kallenbach & Amrhein, 2011; Reif, 2011). Aufgrund der zunehmenden Bedeutung elektronisch kommutierter Drehfeldmotoren innerhalb der Kleinantriebe (Stölting, Kallenbach & Amrhein, 2011; Reif, 2011) wird im Folgenden näher auf deren Differenzierung eingegangen. Bei diesen unterscheidet man Asynchron- und Synchronmotoren, je nachdem wie sich Drehfeld des Stators und Rotor zueinander bewegen. Die elektronisch-kommutierten Kleinantriebe im Automobilbereich werden nahezu ausschließlich mit Synchronmotoren umgesetzt (Wallentowitz & Reif, 2011). Diese lassen sich auf Basis der Materialzusammensetzung, des geometrischen Aufbaus und der elektromagnetischen Wirkungsweise ihrer Rotoren weiterhin in Reluktanzmaschinen, permanentmagneterregte Synchronmaschinen (PMSM), hybride Maschinen mit Reluktanz- und Magnetanteil, sowie elektrisch erregte Synchronmaschinen unterteilen (Kremser, 2004; Nikowitz, Hofer & Schrödl, 2019; Winzer, 2017).

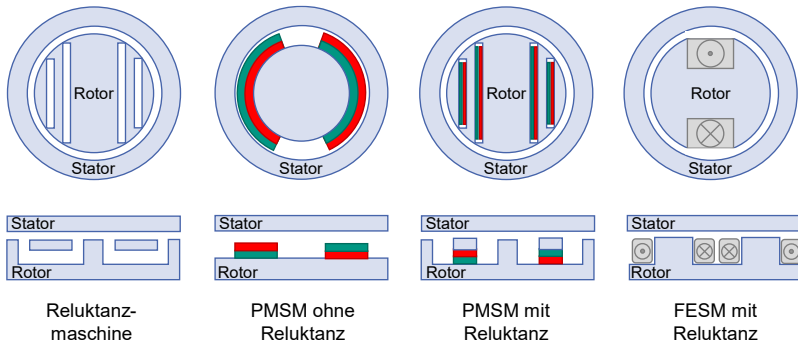


Abbildung 2.7: Grundtypen elektrischer Synchronmaschinen (Reluktanzmaschine, permanentmagneterregte Synchronmaschinen und fremderregte Synchronmaschine) in Anlehnung an Winzer (2017)

Aus diesem Kanon findet die permanentmagneterregte Synchronmaschine im Kontext der elektronisch-kommutierten Kleinantriebe im Automobilbereich am häufigsten ihren Einsatz (Stölting, Kallenbach & Amrhein, 2011). In der Literatur (Stölting, Kallenbach & Amrhein, 2011; Reif & Dietsche, 2018) finden sich des Weiteren die Begriffe der bürstenlosen Gleichstrom- (BLDC) bzw. der bürstenlosen Wechselstrommotoren (BLAC). Hierbei handelt es sich ebenfalls, um den elektronisch kommutierten Motoren zugehörige, kleine permanentmagneterregte Synchronmotoren. Wobei der BLDC Motor für den Betrieb mit blockförmiger Kommutierung und der BLAC Motor für die sinusförmige Strom- und Spannungsversorgung eingesetzt wird. Um maschinenseitig eine möglichst kostengünstige Standardisierung zu erreichen, werden häufig drei- oder sechsfasige Motoren zu verwendet (Stölting, Kallenbach & Amrhein, 2011; Reif & Dietsche, 2018; Kremser, 2004). Bereits mit einer Anzahl von zwei Motorphasen ließen sich Drehfelder konstanter Amplitude erzeugen, welche den Einsatz der oben erwähnten PMSM ermöglichen. Die Entscheidung für dreiphasige Systeme beruht auf der Tatsache, dass dort die Summe der resultierenden Phasenströme jederzeit null ergibt (bei zweiphasigen Systemen, ist dies nicht gegeben). Dies ermöglicht den Einsatz deutlich einfacherer leistungselektronischer Schaltungen, was in einen Kostenvorteil des Gesamtantriebes gegenüber der zweiphasigen Lösung mündet (Stölting, Kallenbach & Amrhein, 2011; Reif & Dietsche, 2018). Sechsfasige Motoren finden häufig in sicherheitskritischen Antrieben, in denen etwa Redundanz erforderlich ist, ihren Einsatz. (Stölting, Kallenbach & Amrhein, 2011; Reif & Dietsche, 2018)

Im Folgenden sind die zur Eingrenzung der elektrischen Kleinantriebe für den Einsatz im Automobilbereich vorgenommenen Schritte abschließend vermerkt:

- Art der Energiewandlung: rotatorisch, elektromagnetisch
- Spannungsklasse: 12 – 48 V
- Leistungsklasse:  $\leq 1000$  W
- Anzahl der Motorphasen: 3 oder 6
- Maschinentyp: Fokus auf elektronisch kommutierten, permanentmagnet-erregten Synchronmaschinen

### **Aufbau mechatronischer Steuergeräte**

Analog zu den betrachteten Kleinantrieben, weisen auch die zugehörigen mechatronischen Steuergeräte eine große bauliche Varianz auf (Becker, 2007). Diese begründet sich in dem breiten Einsatzgebiet und den, aus hohen Stückzahlen resultierenden, anwendungsspezifischen Anpassungen. Zusätzlich herrschen hohe Anforderungen an Sicherheit und Zuverlässigkeit, sowie ein enormer Kostendruck (Steffen, 2007; Reif, 2011; Streichert & Traub, 2012; Becker, 2007; Schäuuffele & Zurawka, 2010). Abbildung 2.8 illustriert einen Ausschnitt der breiten Einsatzmöglichkeiten der mechatronischen Steuergeräte in Kleinantrieben im Automobilbereich:

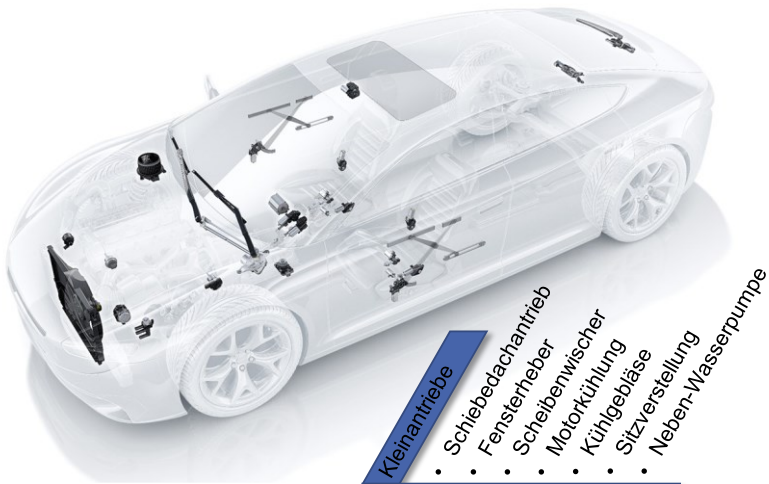


Abbildung 2.8: Einsatzbereiche mechatronischer Steuergeräte in Kleinantrieben im Automobilbereich. Eigene Darstellung nach Reif (2011), unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022)

Zum Schutz der Steuerelektronik vor Umwelteinflüssen und zur Fixierung des Steuergerätes am jeweiligen Einbauort dient das **Gehäuse**. Gehäuseform und Materialien sind dabei abhängig von der gewählten Antriebstopologie (Wallentowitz & Reif, 2011). Man unterscheidet zwischen vom Motor separierten Wegbausteuergeräten, am Motorgehäuse befestigten Steuergeräten und im Motorengehäuse integrierten Steuergeräten (Reif, 2011). Im betrachteten Leistungssegment wird in der Regel eine Integration des Steuergerätes ins Motorengehäuse angestrebt (Stölting, Kallenbach & Amrhein, 2011; Reif & Dietsche, 2018). Die Wahl der Antriebstopologie hat neben dem Einbauort im Fahrzeug großen Einfluss auf die Gestaltung der Entwärmungskonzepte (Stölting, Kallenbach & Amrhein, 2011; Reif & Dietsche, 2018). Diese sind zusammen mit der Aufbau- und Verbindungstechnik zentrale Einflussgrößen für das Produktkonzept (Becker, 2007; Schäuffele & Zurawka, 2010). Trotz der großen Unterschiede im Aufbau und den Einsatzorten der mechatronischen Steuergeräte im Einsatz in Kleinantrieben im Automobilbereich, können auf Ebene der elektronischen Funktionsträger gewisse strukturelle Ähnlichkeiten festgestellt werden (Wallentowitz & Reif, 2011). Abbildung 2.9 zeigt anhand einer Beispielarchitektur die im Rahmen

dieser Arbeit wichtigsten Hardware-Grundbestandteile eines mechatronischen Steuergerätes im Einsatz in Kleinantrieben im Automobilbereich auf.

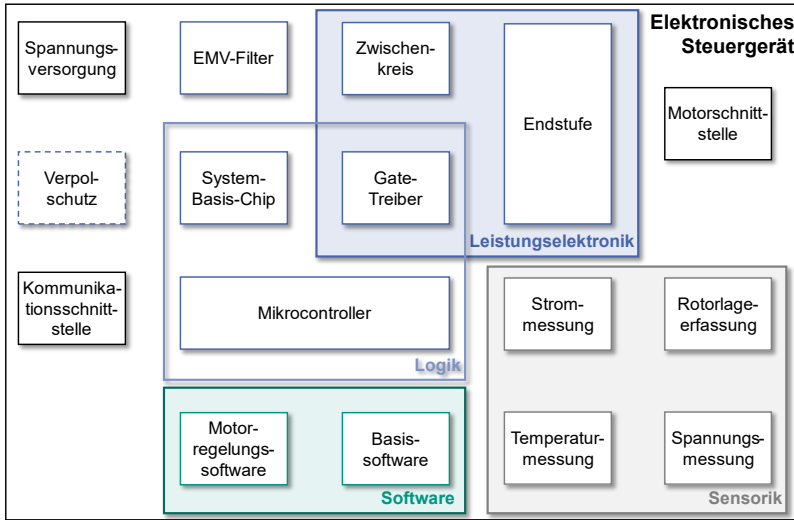


Abbildung 2.9: Grundbestandteile eines Steuergerätes im Kontext der vorliegenden Arbeit, in Anlehnung an (Streichert & Traub, 2012, sowie Wallentowitz & Reif, 2011)

Die **Spannungsversorgung** der Steuergeräte erfolgt mittels Anschluss an das jeweilige Batteriesystem. Je nach Anwendung können auch mehrere Bordnetze (z.B. 12V und 48V) angeschlossen werden (Streichert & Traub, 2012; Wallentowitz & Reif, 2011; Reif, 2011). Die Anbindung erfolgt über besonders gekennzeichnete Klemmen. Wird der Einsatz des Steuergerätes an die Zündung des Automobils gekoppelt so wird ein zusätzliches Signal (Klemme 15) ins System eingespeist. Zur **Kommunikation** mit anderen Elementen der übergeordneten E/E-Architektur, beispielsweise anderen Steuergeräten sind standardisierte Schnittstellen (z.B.: LIN, CAN) vorgesehen (Streichert & Traub, 2012; Wallentowitz & Reif, 2011; Reif, 2011; Schäuffele & Zurawka, 2010). Für Onboard-Diagnose und die Anbindung an Cloud-/Internetdienste werden zusätzliche Schnittstellen genutzt. Der **Mikrocontroller** stellt die zentrale Einheit des Steuergerätes dar (Streichert & Traub, 2012; Wallentowitz & Reif, 2011; Reif, 2011; Schäuffele & Zurawka, 2010). Hier werden die im Messsystem und über die Kommunikationskanäle ermittelten Eingangsgrößen gespeichert, plausibilisiert und weiterverarbeitet. Auf dieser Basis werden die Ausgangsgrößen zur Ansteuerung der Aktoren errechnet und



beispielsweise in Form pulswidenmodulierter Signale an die Stellglieder ausgegeben (Streichert & Traub, 2012; Wallentowitz & Reif, 2011; Reif, 2011; Schäuuffele & Zurawka, 2010). Neben dem Datenspeicher beherbergt der Mikrocontroller ebenfalls einen Programmspeicher für die Anwendungssoftware (Streichert & Traub, 2012; Wallentowitz & Reif, 2011; Reif, 2011; Schäuuffele & Zurawka, 2010). Die Anforderungen an den vorhandenen Speicherplatz und die Rechnerperformance steigen mit zunehmender Anzahl an gewünschter Funktionalität. Die Bereitstellung verschiedener Versorgungsspannungsniveaus für angrenzende Logikbausteine und Sensoren findet häufig im sogenannten **System-Basis-Chip** (SBC) mittels integriertem Spannungswandler statt (Abouda et al., 2021). Im Falle sicherheitskritischer Anwendungen überwacht der SBC die Signalerfassung und Signalverarbeitung im Mikrocontroller. Der SBC beinhaltet die Wake-Up Logik sowie häufig die Kommunikations-Bus-Schnittstellen (Abouda et al., 2021; Elektroniknet, 2021). Zur Erfassung der relevanten Zustandsgrößen des Antriebssystems finden **Spannungs-, Temperatur-, Strom- und optional Sensoren zu Rotorlage-, Drehzahl- oder Momentenerfassung** Anwendung. Diese werden bei Bedarf in unterschiedlicher Anzahl innerhalb und außerhalb des Steuergerätes (beispielsweise an der Motorwelle) angebracht (Streichert & Traub, 2012; Wallentowitz & Reif, 2011; Reif, 2011; Schäuuffele & Zurawka, 2010). Die Sensoren liefern ihre Signale entweder in analoger oder bereits in digitaler Form an den Mikrocontroller zur Verarbeitung. Die aktuelle Rotorposition kann alternativ zur Erfassung mittels magnetischer Sensoren, anhand der vom Motor erzeugten rückwirkenden elektromagnetischen Kraft (engl. back electromagnetic force BEMF) berechnet werden (Genduso, Miceli, Rando & Galluzzo, 2010; Shrutika, Matani, Chaudhuri, Gupta, Gupta & Singh, 2021). Dazu können Phasenspannungen und -ströme herangezogen werden. Die als Stellglieder bezeichneten **Leistungsschalter** sind im Falle der betrachteten 3-phasigen Kleinantriebe als B6-Brückenschaltung von MOSFETs realisiert (Stölting, Kallenbach & Amrhein, 2011). Dies ermöglicht einen Vier-Quadrantenbetrieb der elektrischen Maschine (Kremser, 2004; Schröder & Böcker, 2021). Die Steuerung der Leistungsschalter erfolgt mittels über die Software berechneter Signale. Als **Gate-Treiber** (engl. gate-driver unit GDU) bezeichnet man in dieser Signalkette die integrierte, elektronische Schaltung zur Ansteuerung der Leistungsschalter (Brown, 2001; Heumann, 1996). Dieser empfängt die Steuersignale direkt vom Mikrocontroller und gibt diese mittels höherer Ströme an die Leistungsendstufe weiter. Die Gestaltung der mechanischen **Motorschnittstelle** in Form der Verbindung der Wicklungen der einzelnen Motorphasen mit dem Steuergerät fällt unter die Aufgaben der Aufbau- und Verbindungstechnik und kann anwendungsfallspezifisch erfolgen (Hainzel & Nicolics, 2001; Chamas, Hopfensitz, Obergefell, Oszwald & Traub, 2019). Als bedeutendster Energiespeicher des leistungselektronischen Systems gilt die **Zwischenkreiskapazität**. Diese wird in direkter Nähe zu den Leistungsschaltern

platziert, um Einflüsse parasitärer Induktivitäten zu minimieren (Brown, 2001; Heumann, 1996; Reif, 2011). Durch den transienten Wechsel zwischen Auf- und Entladevorgängen ist die Zwischenkreiskapazität hohen Belastungen in Form von Strom- und Spannungsrippeln ausgesetzt. Zusammen mit der Zwischenkreiskapazität unterdrückt der **EMV-Filterkreis** die Fortpflanzung leitungsgebundener Störspannungen aus dem Bordnetz in das Steuergerät. Gleichzeitig werden durch die Schaltvorgänge der MOSFETs erzeugte Ripplespannungen an der Verbreitung ins Bordnetz gehindert (Streichert & Traub, 2012; Wallentowitz & Reif, 2011; Reif, 2011). Diverse weitere Funktionen zum Schutz des Steuergerätes beispielsweise ein **Verpolschutz** können optional innerhalb des Steuergerätes realisiert werden.

Zur Umsetzung der Steuerungs- bzw. Regelungsfunktionen beherbergt das Steuergerät einen nicht unerheblichen Anteil an Software. Diese wird auf Basis der AUTOSAR-Richtlinien in **Applikationssoftware** (ASW), eine **Laufzeitumgebung** (RTE) und eine **Basissoftware** (BSW) unterteilt (AUTOSAR, 2020; Schäuffele & Zurawka, 2010; Wallentowitz & Reif, 2011). Die Basissoftware fungiert hierbei als Betriebssystem für die übergeordneten Softwareschichten und stellt eine standardisierte Schnittstelle zur Hardware (Mikrocontroller) dar (AUTOSAR, 2020; Schäuffele & Zurawka, 2010; Wallentowitz & Reif, 2011). Die Applikationssoftware enthält sämtliche Softwaremodule, die zur übergeordneten Funktionserfüllung der ECU dienen, wie beispielsweise die Algorithmen zur Motorregelung (AUTOSAR, 2020; Schäuffele & Zurawka, 2010; Wallentowitz & Reif, 2011). Die Laufzeitumgebung ermöglicht sämtliche Kommunikation zwischen BSW und ASW Komponenten sowie die Kommunikation der ECU internen und externen ASW Komponenten (AUTOSAR, 2020; Schäuffele & Zurawka, 2010; Wallentowitz & Reif, 2011). In Abbildung 2.9 nicht erfasst, jedoch den Grundbestandteilen eines Steuergerätes zugehörig ist das **Substrat**. Im Leistungsbereich der betrachteten mechatronischen Steuergeräte ist das Substrat in Form einer Leiterplatte realisiert (Streichert & Traub, 2012; Reif, 2011). Die Leiterplatte übernimmt als Schaltungsträger sowohl mechanische (Fixierung der Elektronikbauteile und Abstützung im Gehäuse) als auch elektrische (durch aufgeprägte Leiterbahnen) Funktionen. Neben der funktionalen Sicht in Form eines Blockschaltbildes sind hinsichtlich des Zusammenspiels von AVT, Elektronikbausteinen und deren Peripherie Querschnittsansichten hilfreich<sup>13</sup> (Powelske et al., 2017). Gerade im Bereich der Leistungselektronik können somit komplexe, aus Schaltplansicht nicht

---

<sup>13</sup> Hierbei in Anlehnung an den Funktions-Gestalt-Zusammenhang der KaSPro – Karlsruher Schule für Produktentwicklung

ersichtliche, interdisziplinäre Wechselwirkungen illustriert werden (Powelske et al., 2017). Abbildung 2.10 zeigt verschiedene Arten der Leiterplattenanbindung und der Entwärmung einer Zwischenkreiskapazität. Die Gestaltung der Anbindung an Substrat und Wärmesenke haben einen großen Einfluss auf die elektrische Funktionalität und die Lebensdauer der eingesetzten Kondensatoren.

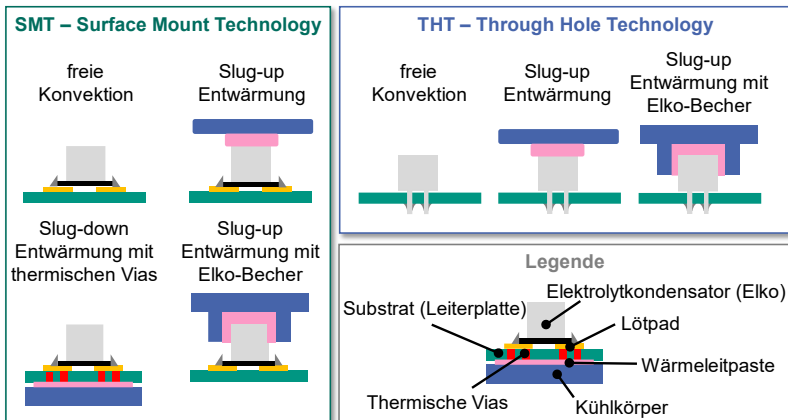


Abbildung 2.10: Übersicht verschiedener Substratanbindungen und Entwärmungskonzepte am Beispiel eines Zwischenkreiskondensators (Darstellung in Anlehnung an Powelske et al., 2017)

Der Trend der Hochintegration (Hainzel & Nicolics, 2001; Chamas, Hopfensitz, Obergfell, Oszwald & Traub, 2019) und die normativen Bestimmungen zur funktionalen Sicherheit führen zu einer anhaltenden Zunahme baulicher und funktionaler Varianz (Volpato & Stocchetti, 2008, eNOVA, 2016). Innerhalb der Steuerelektronik finden sich in gewissen Anwendungen hochintegrierte, applikationsspezifische Mikrocontroller (Kappos & Kinniment, 1996). Diese umfassen SBC, Mikrocontroller und Gate-Treiber in einer Ein-Chip-Lösung. Ähnliche Trends sind auf Seite der Leistungselektronik in Form sogenannter Mold-Module – hochintegrierter B6-Brückenschaltungen zu beobachten (Steffen, 2007). Die Nutzung hochintegrierter Bausteine und diskreter Lösungen im gleichen Portfolio stellt eine große Herausforderung an die Vergleichbarkeit und die Schnittstellenkompatibilität der einzelnen Bausteine (Reif, 2011; Powelske et al., 2017).

## Entwicklung mechatronischer Steuergeräte

Der Produktentstehungsprozess mechatronischer Steuergeräte ist geprägt von einer **hohen Variantenzahl** und einer **komplexen interdisziplinären Entwicklung** (Volpato & Stocchetti, 2008). Getrieben von den vorherrschenden Methoden im Bereich mechatronischer Systeme (vgl. Kapitel 2.2.3 das V-Modell) findet die Entwicklung von Gesamtsystem und domänenspezifischer Teilsysteme **sequenziell** statt. Im Entwicklungsprozess der Steuergeräte, sofern unabhängig von ihrem übergeordneten mechatronischen System z.B. einem mechatronischen Lenksystem betrachtet, dominieren die Anforderungen der **Disziplinen Software und Elektronik** (Dombrowski, Schmidt & Wrehde, 2006). Demnach steht die Orientierung an lösungsneutralen Funktionen zur Systemdekomposition und als Grundlage der Lösungsfindung im Zentrum der Aktivitäten (VDI 2206, 2004). Die Bedeutung der Funktionszentrierung in der Steuergeräteentwicklung lässt sich anhand deren Zuordnung zur Klasse der mechatronischen Funktionsmoduli, also der Betrachtung unabhängig von ihrem übergeordneten System, durch Steffen (2007) ablesen. Der Fokus auf den Entwicklungskonzepten aus Software und Hardware führt zu einer im Produktentstehungsprozess **nachgelagerten Betrachtung des mechanischen Anteils** der Steuergeräteentwicklung, der Aufbau- und Verbindungstechnik (Hainzel & Nicolics, 2001; Chamas, Hopfensitz, Obergfell, Oszwald & Traub, 2019)<sup>14</sup>. Diese nachgelagerte Betrachtung wird zusätzlich durch die von Felsch (2010) beschriebene **Pfadabhängigkeit** begünstigt. Als Pfadabhängigkeit wird in diesem Zusammenhang die historisch bedingte Ausrichtung einer Entwicklungssparte auf die Bereiche, in denen die größte und langjährigste Expertise vorliegt, bezeichnet (Felsch, 2010; Stetter, Rauchenberger, Spiegelberger, Müller, Pfeifer, Witzel & Gideon, 2009). Der Umstand der nachgelagerten Betrachtung einer der beteiligten Disziplinen innerhalb der Steuergeräteentwicklung birgt jedoch die Gefahr die bereits motivierten **negativ-emergenten Wechselwirkungen** (vgl. Kapitel 2.1.1) nicht in erforderlichem Maße zu berücksichtigen. Weiterhin spricht die zunehmende Rolle der AVT als zentraler und über das Produktkonzept entscheidender Kostenfaktor (Becker, 2007; Schäuuffele & Zurawka, 2010), für eine ganzheitliche Betrachtung der einzelnen Disziplinen. Aus der **Interdisziplinarität** ergeben sich zusätzliche Herausforderungen beispielsweise an die Darstellung und Kommunikation der Entwicklungsergebnisse (Albers, Matthiesen, Bursac, Moeser, Klingler, Schmidt,

---

<sup>14</sup> An dieser Stelle sei erneut darauf hingewiesen, dass dies lediglich auf den Kontext des Steuergerätes als eigenständiges mechatronisches System zu beziehen ist und nicht auf das Steuergerät als Teil eines übergeordneten mechatronischen Systems.

Munker, F. et al., 2016). Diese liegen vornehmlich im Umgang mit den verschiedenen Sichten auf das zu entwickelnde System. Die Sichten ergeben sich zwangsläufig aus der unterschiedlichen persönlichen Vorprägung und Expertise der am Produktentstehungsprozess beteiligten Personen (Albers, Ebel & Lohmeyer 2012).

Eine weitere Herausforderung bedingt durch die unterschiedliche Entwicklungspraxis der beteiligten Disziplinen, stellt die kontinuierliche Validierung der Systemkonzepte besonders in frühen Entwicklungsphasen dar (Albers, Rapp et al. 2017; Klingler, 2017). In der Domäne der Software gehört die kontinuierliche Validierung der Entwicklungsartefakte mittels testbasierter Entwicklung bereits zum Stand der Technik. In den Domänen der Hardware und der Mechanik und vor allem auf Ebene des Gesamtsystems bedarf es hierzu geeigneter Methoden (Albers, Behrendt et al., 2016). Beispielhaft für Lösungen zur frühzeitigen und kontinuierlichen Validierung mechatronischer Systeme lassen sich die Veröffentlichungen im Rahmen der KaSPro – Karlsruher Schule für Produktentwicklung von Albers, Fischer, Klingler & Behrendt (2014), Albers, Behrendt et al. (2016), Albers, Behrendt et al. (2017) sowie Albers, Reinemann et al. (2019) anführen. Neben der hohen Varianz und der zunehmenden Funktionsintegration spielen die Technologiezyklen der beteiligten Domänen eine große Rolle für die Steuergeräteentwicklung (Albers, Peglow, Powelske, Birk & Bursac, 2018; Johanning & Mildner, 2015). In der als Trägersystem geltenden Automobilbranche bleiben Entwicklungen ganzer Modellreihen und Funktionsbaukästen auf vier bis acht Jahre fixiert. Die Entwicklungszyklen der einzelnen Disziplinen der Steuergeräteentwicklung sind im Vergleich weitaus schnelllebiger (Johanning & Mildner, 2015). Während die Halbleiterindustrie innerhalb von 12 bis 18 Monaten<sup>15</sup> eine Innovationsschleife durchläuft erneuert sich die Software zur Komponentensteuerung binnen weniger Monate (eNOVA, 2016).

### 2.1.3 Fazit

Mechatronische Systeme haben einen entscheidenden Beitrag zum technologischen Fortschritt der letzten Jahrzehnte beigetragen (Albers im Vorwort zu Gladysz, 2019). Als Vertreter einer Klasse mechatronischer Systeme mit vornehmlich informationsverarbeitenden Funktionen sind die mechatronischen

---

<sup>15</sup> zeitintensive Qualifikationsprozesse für den Einsatz im Automobilbereich sind bereits eingerechnet

Steuergeräte von der funktionsorientierten Sicht der Software- und Hardwareentwicklung geprägt (Deicke, 2018; Hüning, 2019; Schmidt, Buhlmann, Ficek & Richter, 2012; Schäuffele & Zurawka, 2010). Durch die zunehmende Funktionsdichte im Automobil bspw. in Zusammenhang mit der Forschung zu autonomem Fahren, bieten sich gerade im Bereich der elektrischen Kleinantriebe zahlreiche neue Anwendungsmöglichkeiten entsprechender Steuergeräte. Der Trend der Hochintegration verstärkt die ohnehin starken interdisziplinären Wechselwirkungen und führt zu einer drastischen Zunahme der Komplexität im Entwicklungsprozess (Hainzel & Nicolics, 2001; Chamas, Hopfensitz, Obergfell, Oszwald & Traub, 2019). Die kurzen Innovationszyklen von Software- und Halbleiterindustrie erweisen sich als weiterer Komplexitätstreiber (Johanning & Mildner, 2015). Vor diesem Hintergrund erscheint es empfehlenswert, den einzigartigen Entwicklungsprozess mechatronischer Steuergeräte mittels ganzheitlicher Ansätze zu unterstützen. Hinsichtlich der steigenden Variantenvielfalt birgt der Einsatz geeigneter Standardisierungsmethoden für den erfolgreichen Umgang mit der vorherrschenden Produkt- und Prozesskomplexität besonders hohe Potentiale.

## 2.2 Model Based Systems Engineering

Die steigende Produktkomplexität mechatronischer Produkte und die zunehmende Internationalisierung im Produktentwicklungsprozess erfordern neue, auf die Tauglichkeit im interdisziplinären Umfeld optimierte Systementwicklungsansätze (Albers, 2003). Eine wesentliche Herausforderung stellt hierbei die Synchronisation beteiligter Disziplinen mittels eines transparent gestalteten Gesamtsystems dar (Matthiesen et al., 2014). Im Rahmen der Ansätze steht jedoch der Mensch als Bestandteil eines interdisziplinären Problemlösungsteams, stets im Zentrum der Produktentstehung (Albers, 2011). Ein prominenter Vertreter eines solchen Ansatzes verbirgt sich hinter dem Begriff des Model Based Systems Engineering (MBSE). Seit seiner Einführung im Rahmen der „INCOSE Model Based Systems Engineering Initiative“ im Jahre 2007 (Friedenthal, Griego & Sampson, 2007) hat sich MBSE zu einem der vielversprechendsten Forschungsfelder im Bereich der Entwicklung komplexer mechatronischer Systeme entwickelt (Albers & Zingel, 2013a; Gausemeier, Czaja et al., 2013). Erklärtes Ziel von MBSE ist es, die in der Produktentwicklung historisch gewachsenen, dokumentenzentrierten und heterogenen Produktmodelle durch konsistente, vernetzte und ganzheitliche Produktmodelle abzulösen (Alt, 2012). Nach dem International Council on Systems Engineering (INCOSE) ist MBSE wie folgt definiert:

### **Definition 3: Model Based Systems Engineering**

Model Based Systems Engineering ist die formalisierte Anwendung der Modellbildung zur Unterstützung der Aktivitäten im Rahmen der Anforderungsdefinition, des Systemdesigns, der Systemanalyse sowie der Verifikation und Validierung von Systemen, beginnend mit der Konzeptfindung über die Entwicklung und spätere Phasen des Lebenszyklus hinaus. (Übersetzt nach Friedenthal, Griego & Sampson (2007))

Mit dem MBSE Ansatz werden die bewährten Methoden und Prozesse des Systems Engineering (SE) aufgegriffen und um das Modell als zentrale und steuernde Instanz ergänzt. (Estefan, 2008) Auf diesem Wege soll die Vielzahl der traditionell im Entwicklungsprozess anfallenden Dokumente minimiert werden. (Alt, 2012; Albers, Bursac, Scherer, Birk, Powelske & Muschik, 2019) Gleichzeitig werden die Aufwände zur Konsistenzwahrung durch ein zentrales Systemmodell reduziert (Albers & Zingel, 2013a). Abbildung 2.11 illustriert die unterschiedliche Herangehensweise des MBSE gegenüber der, des traditionellen dokumentenzentrierten Arbeitens.

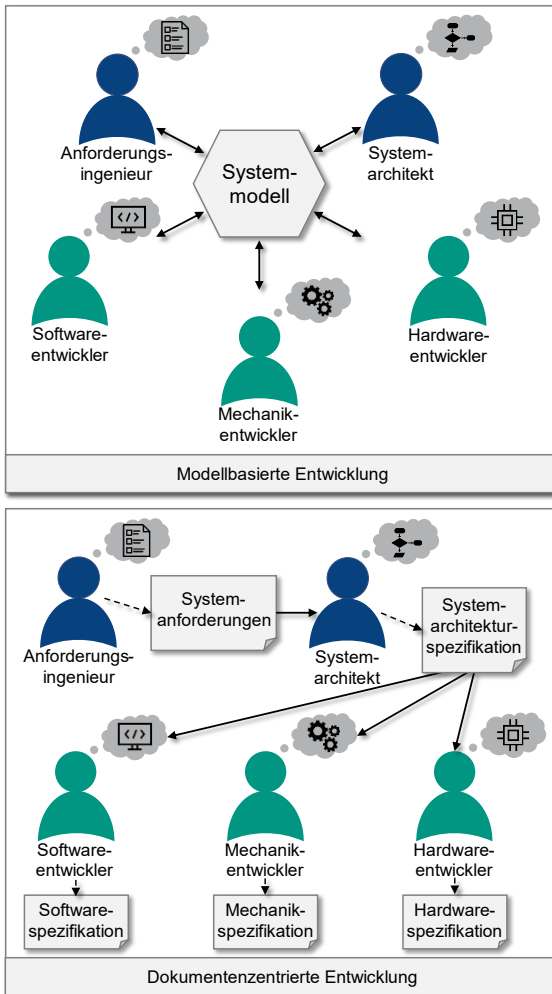


Abbildung 2.11: Gegenüberstellung modellbasierte und dokumentenzentrierte Entwicklung auf Basis der beteiligten Domänen in Anlehnung an Alt (2012)

Im Fokus der Aktivitäten des SE steht die Ganzheitlichkeit der Problembetrachtung (Albers, Ebel & Lohmeyer, 2012; Walden et al., 2015; Albers, Bursac & Scherer,



2014). Mittels des Systemdenkens gelingt es komplexe Wechselwirkungen durch Modelle abzubilden und somit im interdisziplinären Umfeld ein ganzheitliches Denken zu unterstützen (Albers & Lohmeyer, 2012; Habermeyer et al., 2012). Aufgrund der vielfältigen Einsatzmöglichkeiten haben sich über die Jahre hinweg verschiedene Definitionen des SE etabliert. Eine detaillierte Übersicht der Anwendungsbereiche und Paradigmen findet sich bei Hossain & Jaradat (2018). Dem Verständnis dieser Arbeit liegt folgende Definition des SE übersetzt nach Walden et al. (2015), dort in Anlehnung an INCOSE (2011) zugrunde:

#### **Definition 4: Systems Engineering**

Systems Engineering ist ein interdisziplinärer Ansatz mit dem Ziel, die Entwicklung erfolgreicher Systeme zu ermöglichen. Systems Engineering konzentriert sich dabei auf die Definition und Dokumentation der Systemanforderungen in der frühen Entwicklungsphase, die Erarbeitung eines Systemdesigns und die Überprüfung des Systems auf Einhaltung der gestellten Anforderungen über alle Phasen des Lebenszyklus hinweg. Hierfür stellt das Systems Engineering geeignete Methoden und Prozesse bereit.

Als Indikator des Systemerfolgs nennen Sillitto et al. (2019) den Erfüllungsgrad sämtlicher technischer und wirtschaftlicher Anforderungen aller beteiligten Kunden. Der MBSE Ansatz stützt sich auf Vorarbeiten auf den Gebieten der Systemtheorie und der allgemeinen Modelltheorie (Walden et al., 2015; Albers, Scherer, Bursac & Rachenkova, 2015). In den folgenden Abschnitten werden daraus für die vorliegende Arbeit relevante Aspekte skizziert. Darauf aufbauend erfolgt eine Übersicht für die Entwicklung mechatronischer Produkte relevanter Prozessmodelle. Weiterhin wird das für die Methodenentwicklung im Rahmen der Baukastenstrukturierung und -anwendung essenzielle Vokabular der Produktstruktur, mit Fokus auf den strukturellen Spezifika mechatronischer Steuergeräte eingeführt.

### **2.2.1 Systemtheorie**

Der Grundgedanke, dass ein Ganzes stets mehr als die bloße Summe seiner Einzelteile ist, existiert bereits seit der Antike (Wuketits, 1979; Bursac, Albers & Ölschläger, 2016). Erste umfassende und allgemein anerkannte Überlegungen diese und weitere „System-Gedanken“ zu einer Theorie zusammenzufassen haben ihren Ursprung auf dem Gebiet der Biologie. Der Biologe Bertalanffy begründet seine „Allgemeine Systemtheorie“ (Bertalanffy, 1968) auf der Notwendigkeit eines wissenschaftlichen Ansatzes, der eine „Formulierung und Ableitung jener allgemeinen Prinzipien..., die für „Systeme“ überhaupt gelten“, ermöglicht (Ropohl,

1975 dort nach Bertalanffy, 1949). Er verfolgt das Ziel eine Übertragbarkeit domänenspezifischen Verständnisses zu erhöhen und eine Disziplinintegration in der Wissenschaft anzustoßen. Die vorherrschenden Ansätze der damaligen Zeit, Teilaspekte stets isoliert zu betrachten, waren laut Bertalanffy (1949) nicht dazu geeignet Phänomene eines Ganzen zu erklären, die sich aus der Kombination der Einzelteile und deren Beziehungsgeflecht ergeben. Diese Eigenschaft von Systemen wird mittels des aus der Philosophie entliehenen Begriffs der „Emergenz“ (Lewes, 1875)<sup>16</sup> bezeichnet.

Eine Übertragung und Anpassung jenes Systemverständnisses auf die Gegebenheiten technischer Systeme liefert Ropohl (1975). Im Rahmen der „Systemtheorie der Technik“ (Ropohl, 2009; dort aufbauend auf Ropohl, 1975) benennt er drei Kernaspekte von Systemen: das funktionale, das strukturelle und das hierarchische Konzept (Abbildung 2.12).

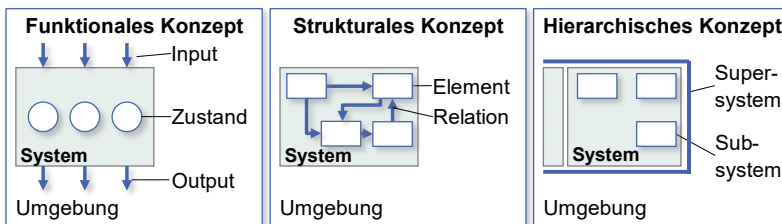


Abbildung 2.12: Die drei Kernaspekte des Systembegriffs, Darstellung nach der dritten Auflage (Ropohl, 2009)

Dem **funktionalen Konzept** folgend, lässt sich ein System durch seine Ein- und Ausgangsgrößen sowie unterschiedliche, anzunehmende Zustände darstellen. Im Zentrum der Betrachtung steht das Verhalten des Systems in Form seiner Interaktion mit der Umwelt. Der innere Aufbau bleibt dabei unberücksichtigt. In der Wissenschaft spricht man vermehrt von einem „black box“ Ansatz (Bursac, 2016; Ropohl, 2009; Gausemeier, Czaja & Dülme, 2015; Daenzer & Huber, 1994).

Das **strukturelle Konzept** besteht in der Beschreibung, der von Bertalanffy (1968) bereits beschriebenen emergenten Eigenschaft, das System als Ganzheit

---

<sup>16</sup> Quelle zur erstmaligen Erwähnung des Emergenzbegriffes

untereinander verknüpfter Elemente zu betrachten. Die Berücksichtigung der Relationen zwischen den Elementen ist dabei Voraussetzung, um Eigenschaften eines Systems umfänglich analysieren zu können (Ropohl, 2009).

Das **hierarchische Konzept** illustriert den Sachverhalt, dass die Elemente eines Systems ihrerseits wieder als Systeme betrachtet werden können. Der gleichen Logik folgend ist jedes System selbst Teil eines übergeordneten „Supersystems“. Albers, Braun & Muschik (2010) zeigen auf Basis des hierarchischen Konzepts, dass sich systemtheoretische Ansätze vom System auf die darin enthaltenen Subsysteme übertragen lassen. Diese Eigenschaft wird als fraktaler Charakter bezeichnet.

Ropohl (2009) kondensiert die Aussagen der drei miteinander vereinbaren Konzepte und verbindet sie in einer Definition des Begriffs „System“. Diese Definition (Ropohl, 2009) bildet die Basis des Systemverständnisses der KaSPro – Karlsruher Schule für Produktentwicklung und soll der vorliegenden Arbeit als Grundlage dienen.

#### **Definition 5: System**

Ein System ist das Modell einer Ganzheit, die (a) Beziehungen zwischen Attributen (Inputs, Outputs, Zustände etc.) aufweist, die (b) aus miteinander verknüpften Teilen bzw. Subsystemen besteht, und die (c) von ihrer Umgebung bzw. von einem Supersystem abgegrenzt wird.

Auf dieser Betrachtung basierend klassifiziert Ropohl (1975) verschiedene Arten von Systemen, wobei das technische System als künstlich entstandenes, zweckorientiertes System ausgeprägt ist. Ferner hat das technische System den Zweck die Durchführung gewünschter, vom Menschen spezifizierter Handlungen zu unterstützen (Zingel, 2013). Vor diesem Hintergrund ist eine vom Menschen losgelöste Betrachtung des technischen Systems „Produkt“ schlichtweg nicht zielführend. Das Produkt als technisches System wird daher in die sozio-technischen Systeme der Produktentstehung und -verwendung eingebettet. Albers & Lohmeyer (2012) entwickeln diesen Gedanken weiter und stellen den Menschen bewusst ins Zentrum der Produktentstehung. Daraus ergibt sich die Notwendigkeit, dass sämtliche Methoden der Produktentstehung nur unter Berücksichtigung des Menschen zu entwickeln sind (Albers, 2003; Albers 2010). Gerade im Zusammenhang interdisziplinärer Produktentstehungsprozesse, wie im Falle mechatronischer Produkte, bedarf die Ressource Mensch besonderer Beachtung. Hier treffen neben verschiedenen Charakteren zusätzlich verschiedene Systemansichten aufeinander. (Ehrlenspiel, 2009; Diehl, 2009). Diese gilt es zur erfolgreichen Systementwicklung entsprechend zu berücksichtigen und zu

harmonisieren. Eine Übersicht menschenzentrierter Methoden der Produktentwicklung wird im Kapitel der Prozessmodelle gegeben.

### **Der Funktionsbegriff im Rahmen der Systemtheorie**

In Zusammenhang mit der Zweckerfüllung technischer Systeme spielt der Funktionsbegriff eine zentrale Rolle. Im Falle von Produkten (als technische Systeme) sind diese in ihrem Entwicklungsprozess zahlreichen Anforderungen und Randbedingungen unterworfen. Der letztendliche Zweck eines Produktes und damit seine auszuübende Wirkung besteht allerdings in der Funktionserfüllung (Feldhusen, Grote, Nagarajah et al., 2013; Alink, 2010). In der Entwicklung findet daher zunächst eine Übersetzung der Anforderungen in Funktionen des gewünschten technischen Systems statt (Pahl, Beitz et al., 2007; VDI 2206, 2004; VDI 2221 Blatt 1, 2018). In der Literatur finden sich zahlreiche Ansätze zur lösungsneutralen Beschreibung von Funktionen. Diese Herangehensweise hat sich aus den Bedürfnissen der Software- und Elektronikentwicklung auch im Bereich des Maschinenbaus etabliert (VDI 2206, 2004; VDI 2221 Blatt 1, 2018). Über der allgemeinen Beschreibung einer Funktion als Beziehung zwischen Eingangs- und Ausgangsgrößen herrscht weitestgehend Konsens (Albers & Wintergerst, 2014; Pahl, Beitz et al., 2007; Erden et al., 2008; Ropohl, 2009; Feldhusen, Grote, Göpfert & Tretow, 2013). Nachfolgend ist die Definition entsprechend dem Verständnis der KaSPro – Karlsruher Schule für Produktentwicklung gegeben (Matthiesen, 2021):

#### **Definition 6: Funktion**

Der Begriff Funktion ist für den Produktentwickler nur im Umgang mit Modellen von technischen Systemen anwendbar. Sie ist die Umwandlung von Input zu Output, die in dem modellierten System abgebildet wird. Sie entsteht aus der Interaktion von Gestaltelementen und kann nicht direkt beeinflusst werden.

Betrachtet man die Funktion jedoch hinsichtlich ihrer Unterscheidung vom Begriff des Verhaltens, so herrschen je nach Autor und Kontext unzählige Definitionen. Eine detaillierte Übersicht diverser Sichtweisen findet sich bei Erden et al. (2008). Im Zuge dieser Arbeit haben sich diejenigen Definitionen des Verhaltensbegriffes bewährt, die im Rahmen der KaSPro - Karlsruher Schule der Produktentwicklung entstanden sind. Diese lauten:

**Definition 7: beabsichtigtes Verhalten**

Beabsichtigtes Verhalten beschreibt die gewünschten Reaktionen eines Produkts oder seiner Teilsysteme auf Einflüsse, die durch Interaktionen mit Umgebungssystemen induziert werden. (Wintergerst, 2015, dort nach Albers & Düser, 2010)

**Definition 8: tatsächliches Verhalten**

Tatsächliches Verhalten beschreibt die in der Erprobung oder Anwendung festgestellten erwünschten oder unerwünschten Wirkungen. (Alink, 2010, dort nach Albers & Düser 2010)

Gegenüber der lösungsneutralen Funktionsbeschreibung und deren abstrakter Definition weist der Verhaltensbegriff auf eine untrennbare Verbindung von Funktion und tatsächlicher Gestalt hin. Dieser sogenannte Funktions-Gestalt-Zusammenhang wird in Kapitel 2.2.5 auf Basis der Forschung von Albers & Wintergerst (2014) gesondert vorgestellt. Vor dem Hintergrund der mechatronischen Steuergeräteentwicklung, die die verschiedenen Sichten auf den Entwicklungsprozess der beteiligten Disziplinen umfasst, müssen stets beide Ansätze Beachtung finden. In Zusammenhang mit dem Verhalten mechatronischer Systeme bedarf es weiterhin der Erläuterung der negativen Emergenz (Göpfert, 1998). Diese beschreibt unerwünschte Effekte, die sich aus dem tatsächlichen Verhalten von technischen Systemen bzw. der realen Wechselwirkungen zwischen Systemelementen ergeben. Albers & Wintergerst (2014) sprechen hierbei auch von gewolltem und ungewolltem Verhalten. Negativ emergente Phänomene erfordern zusätzliche Aufmerksamkeit und erhöhen die Komplexität des Entwicklungsprozesses deutlich. Anschauliche Beispiele aus dem Umfeld mechatronischer Steuergeräte sind die elektromagnetische Verträglichkeit oder die Erwärmung und Erzeugung von Verlustleistung in Leistungselektronikbauteilen.

Im Rahmen der Systemtheorie ist die Notwendigkeit von Modellen zur Explikation und Abbildung von Systemen jeglicher Art impliziert. Der nächste Abschnitt widmet sich daher eingangs, den Grundlagen der allgemeinen Modelltheorie. Darauf aufbauend wird der eigentliche Erstellungsprozess von Modellen - die sogenannte Modellierung - näher beleuchtet.

**2.2.2 Modelltheorie**

Die Bedeutung von Modellen für unsere Gesellschaft als Grundlage des menschlichen Erkenntnisprozesses wird von Stachowiak (1973) in seiner

„allgemeinen Modelltheorie“ betont. Demnach „...ist alle Erkenntnis in Modellen oder durch Modelle, und jegliche menschliche Weltbegegnung überhaupt bedarf des Mediums ‚Modell‘...“ (Stachowiak, 1973, S. 56). Die Modellbildung als zentralen Bestandteil in die Produktentstehung zu integrieren, erscheint vor diesem Hintergrund die logische Konsequenz. Nachfolgend wird ein Überblick des dieser Arbeit zugrundeliegenden Modellverständnisses gegeben:

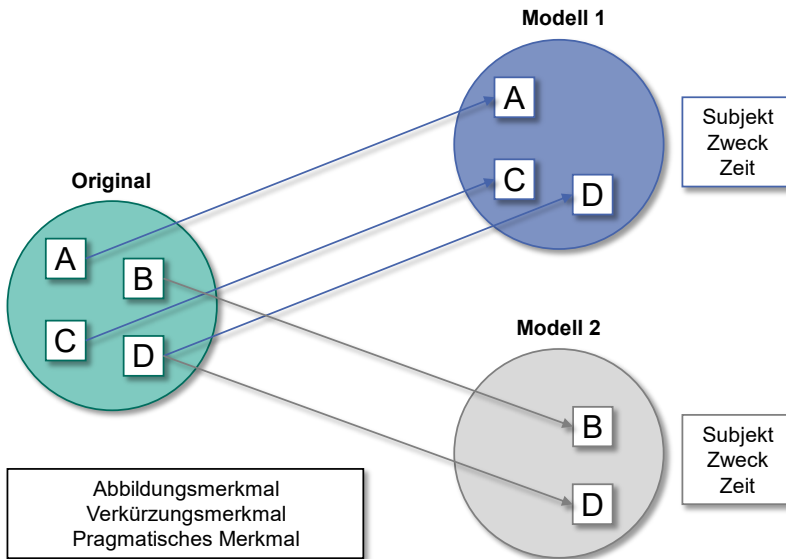


Abbildung 2.13: Merkmale von Modellen (eigene Darstellung nach Stachowiak (1973))

Die Beschreibung der Modellbildung erfolgt auf Basis des Abbildungsmerkmals, des Verkürzungsmerkmals und des pragmatischen Merkmals (Abbildung 2.13). Hinter dem **Abbildungsmerkmal** verbirgt sich die Aussage, dass jedes Modell die Übersetzung eines natürlichen oder künstlichen Originals darstellt. Darin ist enthalten, dass Modelle selbst als Grundlage für Modelle fungieren können. Das **Verkürzungsmerkmal** beschreibt den Sachverhalt, dass die Menge der Attribute, welche in einem Modell enthalten sind, stets nur eine Untermenge der Attribute des Originals sein kann. Die abgebildeten Attribute sind dabei diejenigen, die für den Modellersteller oder –nutzer gleichermaßen als relevant erachtet werden. Dem

**pragmatischen Merkmal** folgend sind Modelle ihren Originalen nicht eindeutig zugeordnet. Sie erfüllen vielmehr innerhalb bestimmter Zeitintervalle und unter Einschränkung auf bestimmte Operationen einen Zweck für einen ausgewählten Kreis an Personen (Stachowiak, 1973). Folglich sind Modelle auf Basis dieser Kriterien immer als zweckorientierte, verkürzte Darstellungen eines Originals zu sehen. Einem Streben nach Vollständigkeit in Bezug auf die Abbildung realer Prozesse mittels Modellen, ist daher per Definition, ein Riegel vorgeschoben. Um Missverständnissen vorzubeugen, muss klar zwischen Vollständigkeit und dem im MBSE genutzten Begriff der Ganzheitlichkeit unterschieden werden. Ganzheitlichkeit versteht sich als Betrachtung einer Sache in der systemischen Vollständigkeit aller Teilsysteme, nicht aber mit dem Anspruch der Vollständigkeit hinsichtlich der Modellbildung (Friedenthal, Griego & Sampson, 2007).

Modelle sind im Produktentwicklungsprozess omnipräsent (Albers, Matthiesen, Bursac, Moeser, Klingler, Schmidt, Munker et al., 2016; Braun, 2013) Sie liegen entweder in Form mentaler Modelle (ausschließlich im Kopf der Entwickler (Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013 dort nach Meboldt, 2008), oder analog (z.B. Handskizzen) bzw. digital (z.B. CAD-Modelle, Berechnungsmodelle, ...) expliziert, in unterschiedlichsten Abstraktionsgraden vor. Explizierte Modelle haben den Mehrwert, den Informationsaustausch zu unterstützen und somit das übergeordnete Systemverständnis des Entwicklerteams zu steigern (Lamm & Weilkiens, 2014). Da alle Erkenntnis an die Modellbildung und folglich an die subjektive Verkürzung der Realität gekoppelt ist (Stachowiak, 1973), entsteht im Laufe des Entwicklungsprozesses eine Vielzahl verschiedener Modelle. Aufgrund ihres unterschiedlichen Zwecks sind die Modelle selbst bei gleichem, zugrundeliegendem Original auf unterschiedliche Weise verkürzt und daher häufig nicht miteinander kompatibel. Man spricht in diesem Falle vom sogenannten Reduktions-Dilemma (Lohmeyer, 2013; Bursac, 2016). Das Vorhandensein inkompatibler, nicht verbundener Partialmodelle steht zunächst im Widerspruch zur durchgängigen Verwendung eines ganzheitlichen Systemmodells und damit zum Kerngedanken des MBSE.

In der Entwicklung mechatronischer Systeme treffen in Form der beteiligten Disziplinen zusätzlich unterschiedliche Denkweisen und Systemverständnisse aufeinander (Ehrlenspiel, 2009; Diehl, 2009). Folglich erhöht sich im interdisziplinären Entwicklungsprozess die Anzahl verschiedener, mentaler oder explizierter Produkt- bzw. Systemmodelle. Ohne eine Zusammenführung und Harmonisierung dieser Modelle mittels durchgängiger und ganzheitlicher Modellbildung kann das Auftreten inkompatibler Modelle ein und desselben Produktes nicht ausgeschlossen werden (Klare et al., 2021). Gründe dafür liegen beispielsweise in der hohen Komplexität der mechatronischen Wechselwirkungen,

die ein einheitliches Verständnis des Gesamtsystems erschwert. Das Reduktions-Dilemma begründet sich auf der Individualität der Denk- und Handlungsweisen der Entwickler (Lohmeyer, 2013) und stellt einen Widerspruch zur Forderung nach ganzheitlichen Modellen dar. Die Auflösung dieses Widerspruchs gelingt daher nur über einen menschenzentrierten Ansatz. Dementsprechend leiten Albers & Lohmeyer (2012) klare Richtlinien über die Modellerstellung und deren Verwendung im Rahmen des MBSE ab und formulieren folgende Qualitätskriterien: Ganzheitlichkeit, Reduktion, Durchgängigkeit, Kontinuität, Skalierbarkeit und Akzeptanz (Albers & Lohmeyer, 2012; Badke-Schaub, Daalhuizen & Roozenburg, 2011). Das Kriterium der Durchgängigkeit kann dabei in die horizontale Durchgängigkeit, die vertikale Durchgängigkeit und die Konsistenz der Modelle unterteilt werden (Albers & Lohmeyer, 2012). Die Horizontale Durchgängigkeit erfordert die zeitlich durchgängige Verwendbarkeit der Modelle. Sie bedingt deren kontinuierliche Erweiterbarkeit und Aktualisierbarkeit. Im Rahmen der Vertikalen Durchgängigkeit ist der kontinuierliche Wechsel von Hierarchieebenen und unterschiedlichen Blickwinkeln durch die Modellbildung zu gewährleisten. Die Konsistenz ist dann gewährleistet, wenn sich zwei unterschiedliche Modelle desselben Produktes nicht widersprechen, sondern ergänzen (Albers & Lohmeyer, 2012).

Die Erstellung durchgängiger Modelle in der Produktentwicklung erfordert neben geeigneten **Modellierungsmethoden**, eine geeignete **Sprache** zur Abbildung der Modellelemente und ein geeignetes **Tool** zur Umsetzung (Lohmeyer, 2013). Man spricht hierbei vom sogenannten „Modeling Triple“<sup>17</sup> (Matthiesen et al., 2014 dort nach Friedenthal, Moore & Steiner, 2008; Albers, Munker, Zingel & Behrendt, 2013) (Abbildung 2.1.4). Die vorliegende Arbeit beschränkt sich in ihrer Betrachtung explizit auf die Modellierung rechnerbasierter, objektorientierter Systemmodelle. Das Verständnis des Toolbegriffs umfasst vor diesem Hintergrund ausschließlich Softwarewerkzeuge zur Modellbildung.

---

<sup>17</sup> in Zusammenhang mit der interdisziplinären Systemmodellierung auch als drei Säulen bezeichnet (Albers, Munker, Zingel & Behrendt, 2013)



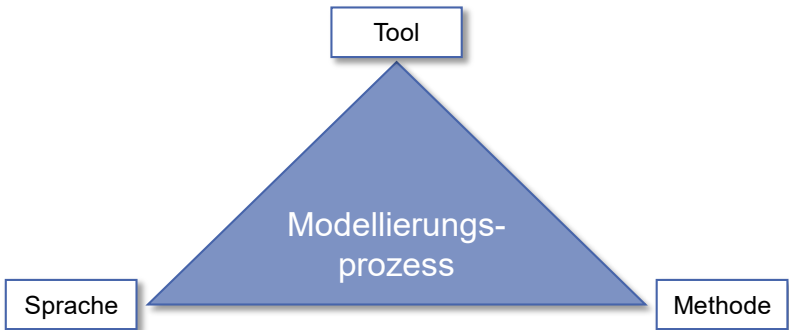


Abbildung 2.14: Modelling Triple (Sprache-Tool-Prozess) nach Matthiesen et al. (2014).

Die **Methoden** zur Modellierung dienen im Zuge des menschenzentrierten Ansatzes, als Unterstützung des Entwicklers bei der Modellbildung. Hierbei liegt die zentrale Herausforderung in der angemessenen Abstraktion (bzw. Reduktion) bei der Überführung komplexer Systemzusammenhänge zu einem Systemmodell (Albers, Matthiesen et al., 2014). Die Entscheidung gewisse Informationen ins Modell zu überführen oder entfallen zu lassen verändert lediglich den Blickwinkel des Entwicklers auf das System und damit die Sicht auf die verbundene Problemstellung. Dies wiederum führt zu unterschiedlichen Ergebnissichten (Albers, Matthiesen et al., 2014). Die Komplexität des zugrundeliegenden Systems bleibt davon unberührt. In der Literatur herrscht die weitverbreitete Meinung mittels geeigneter Methoden die Komplexität im Entwicklungsprozess mechatronischer Produkte zu reduzieren (Schuh & Riesener, 2018; Butz, 2011). Tatsächlich blendet man mittels Modellbildung gezielt Informationen eines beliebig komplexen Systems aus, um den Umgang des Entwicklers mit der Komplexität und damit seine individuelle Problemlösung zu unterstützen (Albers, 2010). Eine weitere Herausforderung bei der Modellbildung im Kontext interdisziplinärer Zusammenarbeit liegt in der Bereitstellung der vertikalen Durchgängigkeit eines Systemmodells. Dabei gilt es die Vielzahl der Nutzer und deren verschiedene Blickwinkel auf das Modell, sowie verschiedene Sprachen zur Darstellung zu berücksichtigen. Im Rahmen des MBSE hat sich dazu ein Konzept zur Ableitung spezifischer Sichten durchgesetzt (Abbildung 2.15).

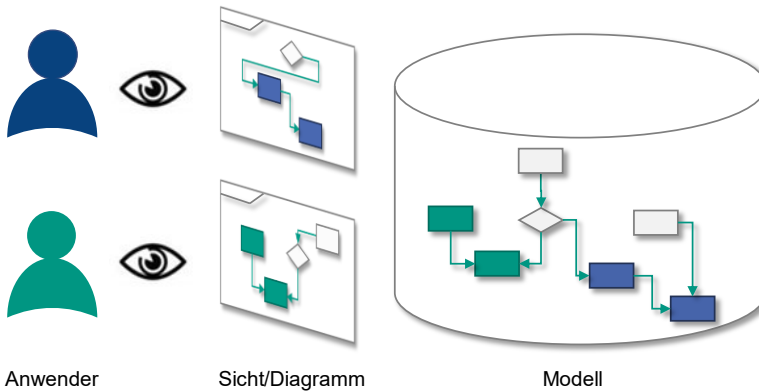


Abbildung 2.15: Das Sichtenkonzept des MBSE nach Bursac (2016) und Alt (2012)

Grundlage bildet hierbei ein in der Regel komplexes Systemmodell, welches sämtliche Modellelemente und deren Beziehungen beinhaltet. Über die Modellierungssprache und die verwendeten Tools lassen sich einzelne Sichten (bspw. in Form von Diagrammen) ableiten. Mittels dieser Diagramme können gezielt spezifische Modellelemente und Beziehungen, also Ausschnitte des zugrundeliegenden Gesamtsystems visualisiert und bearbeitet werden. Die einzelnen Sichten können zur besseren Handhabung auf einen bestimmten Anwenderkreis eingeschränkt werden. Fügt man innerhalb einer Sicht neue Elemente oder Beziehungen hinzu wird diese Veränderung automatisch und konsistent ins Gesamtmodell übernommen. Gegenüber der klassischen, dokumentenzentrierten Entwicklung erspart dies den redundanten Übertrag von Änderungen. Gleichzeitig erhöht sich allerdings die Komplexität der Modellbildung des Gesamtmodells unter Gewährleistung widerspruchsfreier Sichten (Albers, Matthiesen, Bursac, Moeser, Schmidt & Lüdcke, 2014). Bei der Erstellung von Modellen wird der Entwickler durch spezifische Tools unterstützt. Diese sind jeweils auf den Modellierungsfokus (Berechnung, Gestaltmodellierung, usw.) ausgerichtet. Bei der Zusammenführung zu ganzheitlichen Systemmodellen liegen nicht selten verschieden geartete Partialmodelle (Berechnungsmodelle, Gestaltmodelle, ...) aus

unterschiedlichen Toolumgebungen vor (Eigner et al., 2012)<sup>18</sup>. Um die im MBSE geforderte Konsistenz bei der sichtenbasierten Modellierung zu gewährleisten, müssen die Daten der zugrundeliegenden Partialmodelle von Widersprüchen befreit werden. Da sich dies mit steigender Anzahl beteiligter Domänen und Sprachen innerhalb der Modelle immer komplexer gestaltet empfehlen Klare et al. (2021) den Einsatz eines sogenannten V-SUM (Virtual Single Underlying Model). Dieses ermöglicht anhand seines modularen Aufbaus eine Wiederverwendung von Modellelementen, Modellen, Metamodellen und Sichten. Darüber hinaus bietet es die Möglichkeit zur bedarfsgerechten Definition von Maßnahmen zur Konsistenzwahrung v.A. bei der Überführung verschiedener Modellelemente in Sichten mit unterschiedlichen zugrundeliegenden Sprachen. Bisher wird das V-SUM Modell vorwiegend im Zusammenhang mit der Forschung auf dem Gebiet der Informatik verwendet, die Nutzung im Rahmen mechatronischer Systeme ist aktueller Gegenstand der Grundlagenforschung.

Zur durchgängigen Modellbildung finden sich in der Literatur zwei verschiedene Philosophien, der integrierte und der föderierte Ansatz (Grundel et al., 2014; Kleiner, 2003) (Abbildung 2.16):

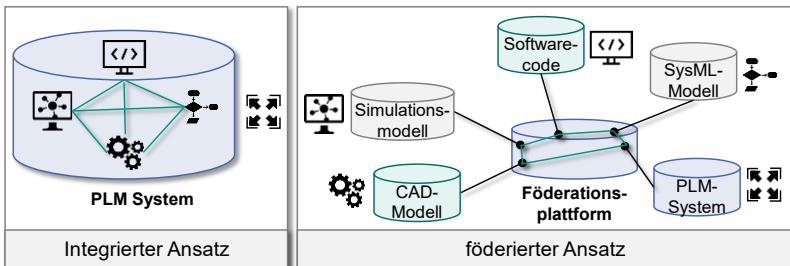


Abbildung 2.16: Ansätze zur Modellbildung (integrierter vs. föderierter Ansatz) nach Kleiner (2003).

<sup>18</sup> Dort beispielsweise: qualitative Modelle zur Modellbildung und Spezifikation, quantitative Modelle zur Simulation, Geometrie oder CAE-Modelle für die disziplinspezifische Modellbildung

Typische Beispiele für integrierte Ansätze in der Produktentwicklung sind PLM (Product Lifecycle Management) Ansätze. Im Rahmen komplexer mechatronischer Systeme gewinnen zudem SysML- (System Lifecycle Management) Ansätze vermehrt an Bedeutung. Beide Formen dienen der Verwaltung und Verteilung sämtlicher relevanter und digitaler Informationen, die entlang der Wertschöpfungskette entstehen und aufeinander abgestimmt werden müssen (Eigner, 2014). Die Modellierung und die zugrundeliegende Datenstruktur erfolgen beim integrierten Ansatz innerhalb einer durchgängigen Toolumgebung. Eigner (2014) spricht in diesem Kontext vom zentralen Modell als der „Single source of truth“ (Eigner, 2014, S. 279). Zu den Vorteilen integrierter Ansätze zählen die zentrale Versionierung, sowie die uniformen Datentypen innerhalb des Tools. Grundlegende Veränderungen der Datenstruktur und die Integration neuer Tools sind hingegen mit großen Aufwänden verbunden und bedürfen einer durchgängigen Betreuung seitens des Toolherstellers.

Beim föderierten Ansatz werden existierende Partialmodelle, die in spezifischen Tools vorliegen, über individualisierte Schnittstellen innerhalb einer sogenannten Förderierungsplattform verbunden (Grundel et al., 2014; Kleiner, 2003). Über einen solchen Ansatz ermöglicht man den Entwicklern ihre über die Jahre hinweg etablierten Tools weiterhin zu verwenden. Besonders innerhalb domänenübergreifender Expertenorganisationen erweist sich der Einsatz spezifischer Tools als zwingend notwendig (Grundel et al., 2014). Die Vielzahl verschiedener zu harmonisierender Datentypen und die Versionierung des aus Partialmodellen zusammengesetzten Systemmodells bilden den Hauptkritikpunkt an dieser Vorgehensweise. Der Aufwand liegt dementsprechend in der Gestaltung konsistenter Schnittstellen (Powelske et al., 2018). Mittels der individuellen Verknüpfung zu durchgängigen Modellen über die Toolgrenzen hinweg bewahrt man sich allerdings eine hohe Flexibilität gegenüber der Integration neuer Tools und Methoden (Powelske et al., 2018; Stark et al., 2010). Zudem kann, durch die individuelle Verknüpfung von Partialmodellen, der Einzigartigkeit und Individualität des Produktentstehungsprozesses (Albers, 2010) besser entsprochen werden als mit integrierten Ansätzen (Powelske et al., 2018).

In der Produktentwicklung spielt die Modellbildung neben der Abbildung des technischen Systems „Produkt“ gerade bei der Explikation menschzentrierter Methoden in Zusammenhang des sozio-technischen Systems der Produktentstehung eine große Rolle (Albers & Lohmeyer, 2012; Gausemeier, Czaja et al., 2013). Diese Methoden werden in sogenannten Prozessmodellen aggregiert. Eine Übersicht verschiedener, im Bereich der mechatronischen Steuergeräte relevanter Prozessmodelle und damit verbundener verpflichtender Standards wird im nächsten Kapitel gegeben.

### 2.2.3 Prozessmodelle

Im folgenden Abschnitt werden verschiedene, vornehmlich entwicklerorientierte Prozessmodelle der Produktentstehung vorgestellt. Zunächst wird auf die, im Bereich mechatronischer Produkte bedeutendste Norm, die VDI-Richtlinie 2206 eingegangen. Diese bildet unter anderem die Grundlage für die in der Steuergeräteentwicklung verbindlichen Richtlinien der ISO 26262 und ASPICE. Nach einem kurzen Überblick beider, erfolgt die Vorstellung des Modells der Produktentwicklung der VDI-Richtlinie 2221 Blatt 1. Das abschließend behandelte integrierte Produktentstehungsmodell - iPeM ergänzt die entwicklungspezifische Sicht um die Schnittstellen zum operativen Management. Es bietet somit einen weitaus umfassenderen Blick auf das sozio-technische System der Produktentstehung.

#### **Entwicklungsmethodik für mechatronische und cyber-physische Systeme: VDI 2206**

Die im Rahmen der VDI Richtlinie 2206 vorgestellte Methodik bildet bis heute eine der wichtigsten Grundlagen zur Entwicklung mechatronischer Produkte im industriellen Umfeld (Cadet et al., 2017).

Die Richtlinie<sup>19</sup> umfasst in ihrem Kern drei Elemente:

- Den Problemlösungszyklus als Mikrozyklus
- Das V-Modell als Makrozyklus
- Die Prozessbausteine für wiederkehrende Arbeitsschritte

Der innerhalb der Richtlinie vorgestellte **Mikrozyklus** basiert auf dem allgemeinen Problemlösungszyklus von Daenzer & Huber (1994) und wurde zur Verwendung lediglich um eine Phase ergänzt. Er umfasst folgende Schritte:

- Situationsanalyse bzw. Zielübernahme
- Analyse und Synthese
- Analyse und Bewertung

---

<sup>19</sup> An dieser Stelle wird maßgeblich auf die Version der Norm aus dem Jahre 2004 eingegangen, da die Auswirkungen der 11/2021 erschienenen Version auf die industrielle Praxis zum gegenwärtigen Zeitpunkt noch nicht erfasst werden können

- Entscheidung
- Planen des weiteren Vorgehens bzw. Lernen<sup>20</sup>

Weitaus bekannterer Bestandteil der Richtlinie ist allerdings das V-Modell. Dieses wurde aus der Softwareentwicklung übernommen und zur Darstellung des domänenübergreifenden Makrozyklus an den mechatronischen Entwicklungsprozess angepasst (Abbildung 2.17). Eingangsgröße des Makrozyklus stellen die **Anforderungen** an das zu entwickelnde Produkt dar. Auf deren Basis werden, die zur späteren **Eigenschaftsabsicherung** notwendigen Testfälle spezifiziert. Die Anforderungen werden im anschließenden **Systementwurf** zunächst in möglichst lösungsneutrale Funktionen (Pahl, Beitz et al., 2007) übersetzt und in Teilfunktionen heruntergebrochen. Dies geschieht so lange, bis zu deren Umsetzung geeignete Wirkprinzipien und Lösungselemente gefunden sind und zugeordnet werden können. Ist ausgewählte Lösungskonzept ausreichend konkretisiert wird im sogenannten **domänenspezifischen Entwurf** die Umsetzung der Funktionserfüllung entsprechend auf die beteiligten Domänen verteilt.

---

<sup>20</sup> Ergänzung gegenüber dem initialen Modell von Daenzer & Huber (1994)

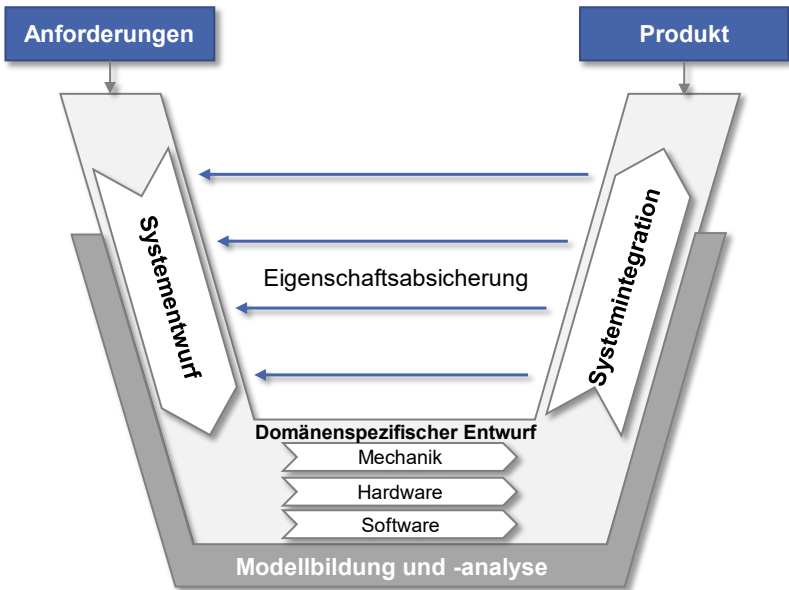


Abbildung 2.17: Das V-Modell nach VDI 2206 (2004).

Nachdem die einzelnen Funktionen innerhalb der Domänen umgesetzt sind, werden diese sukzessive zum Gesamtsystem integriert. Während der **Systemintegration** erfolgt eine konstante Absicherung der innerhalb der Anforderungen abgefassten Eigenschaften in Form einer Verifizierung und Validierung. Sowohl die Entwurfs- als auch die Integrationsaktivitäten werden mittels der **Bildung und Analyse von Modellen** unterstützt. An dieser Stelle finden die Methoden des MBSE immer stärkeren Eingang in die Entwicklungspraxis. Einer der Gründe dafür, ist die Forderung diverser Normen und Richtlinien, eine durchgängige Nachverfolgbarkeit (**Traceability**) von Kundenanforderungen zu deren Realisierung in der technischen Architektur bereit zu stellen (ASPICE, 2017; ISO 26262, 2018). Dokumentenzentrierte Ansätze eignen sich aufgrund geringer Flexibilität und hohem Anpassungsaufwand zur Konsistenzsicherung nur bedingt zur Darstellung der Traceability (Kaufmann & Schuler, 2016). Daher werden die im Makrozyklus des V-Modells vorgesehenen Schritte vermehrt in durchgängigen Modellen dargestellt (Bartelt, Ternité & Ziegler, 2005). Die Phasen des Makrozyklus resultieren schließlich im **Produkt**. Dieses ist im Sinne der Richtlinie nicht zwingend als reales

Erzeugnis zu verstehen, sondern kann verschiedene Reifegrade in Form von Musterständen abbilden. Im Falle komplexer mechatronischer Produkte wird von vorneherein ein repetitives, sequenzielles Durchlaufen des Makrozyklus nahegelegt (VDI 2206, 2004).

Das letzte Element der Richtlinie „Prozessbausteine für wiederkehrende Arbeitsschritte“ beinhaltet die Konkretisierung der im Makrozyklus beschriebenen Phasen: Systementwurf, Modellbildung und –analyse, domänenspezifischer Entwurf, Systemintegration und Eigenschaftsabsicherung. Dort dargestellte methodische Ansätze basieren auf den allgemeinen und brachenunabhängigen Grundlagen methodischen Entwickelns, wie sie in der VDI-Richtlinie 2221 (VDI 2221, 1993) abgefasst sind.

Im praktischen Einsatz hat sich das sequenzielle Abarbeiten der im V-Modell<sup>21</sup> vorgestellten Vorgehensweise jedoch häufig als nicht zielführend herausgestellt. Nachfolgend werden einige Ergebnisse der kritischen Auseinandersetzung diverser Forschungsarbeiten mit dem V-Modell vorgestellt. Vornehmlich im Bereich der Anforderungen weicht das im Makrozyklus beschriebene Idealbild eines vollständigen Zielsystems stark von der Realität im Produktentwicklungsprozess ab. Die geforderte Anforderungsspezifikation kann zu Beginn der Entwicklung aufgrund mangelnder Wissensbasis lediglich rudimentär erfolgen (Muschik, 2011). Das V-Modell sieht weiterhin keine Iterationen zwischen den einzelnen Phasen vor. Der im Bereich des SE entwickelte SIMILAR-Prozess betont, dass ohne die Iterativität und Parallelisierbarkeit der einzelnen Phasen eine erfolgreiche Systementwicklung nicht denkbar ist (Bahill & Gissing, 1998). So können beispielsweise Anforderungen, welche sich aus einer Konzeptentscheidung innerhalb der technischen Architektur ergeben, frühestens im zweiten Durchlauf des V-Zyklus Berücksichtigung finden. Eine kontinuierliche Weiterentwicklung des Zielsystems, wie nach Albers (2010) gefordert, ist nach dieser Vorgehensweise nur bedingt möglich. Auf den Anforderungen software- und hardwaregetriebener Branchen ist die Idee einer getrennten Entwicklung von System- und Domänenebene entstanden. Zusammen mit der Trennung von Funktion und Gestalt sorgt diese gerade im Bereich mechanischer Integration für Vorbehalte (Albers & Wintergerst, 2014, Matthiesen, 2002). Der Berücksichtigung der Validierung als Absicherung gewünschter Soll-Eigenschaften (Albers, Ebel & Lohmeyer, 2012) kommt im V-Modell eine zentrale Rolle zugute. Diese findet nach der Ausarbeitung der domänenspezifischen Lösungen während der Systemintegration statt. Die aktuelle Forschung zeigt

---

<sup>21</sup> Der Version der Norm von 2004



jedoch, dass die Validierung begleitend zum gesamten Produktentstehungsprozess iterativ erfolgen sollte. Gerade in den frühen Entwicklungsphasen können Ergebnisse einer Validierung einen essenziellen Beitrag zur Konkretisierung des vagen Zielsystems leisten. (Albers, Behrendt, Klingler & Matros 2017). Basierend auf der umfangreichen Kritik und dem immer größer werdenden Einfluss der datenbasierten Vernetzung mechatronischer Systeme wurde die Norm im November 2021 grundlegend überarbeitet und um die Klasse der cyber-physischen Systeme ergänzt. (vgl. Kapitel 2.1.1) Maßgebliche Veränderungen liegen bspw. in der explizit berücksichtigten Verschachtelung des V-Modells, um der Iterativität des Produktentwicklungsprozesses Rechnung tragen zu können.

Trotz umfangreicher Kritik am V-Modell<sup>22</sup> existieren im Bereich der mechatronischen Steuergeräteentwicklung einige verbindliche Standards, die auf dem Makrozyklus aufbauen. Nachfolgend werden die beiden bedeutendsten Richtlinien vorgestellt.

### **Road-Vehicles – Functional Safety: ISO 26262**

Mit der zunehmenden Funktionalität im Automobil lässt sich ein immer höherer Substitutionsgrad rein mechanischer Funktionsträger/-prinzipien durch Software- und Hardwarekomponenten beobachten (Reif, 2011; Automobil-Elektronik, 2011). Als Folge, ergibt sich ein interdisziplinärer Entwicklungsprozess mit steigendem Vernetzungsgrad einzelner Steuergeräte, im Rahmen der gesamten E/E-Architektur (Hillenbrand, 2011). Dies resultiert in einer Komplexitätszunahme innerhalb der Produkte und ihres Netzwerks. Eine steigende Komplexität erhöht wiederum das Risiko möglicher Fehlfunktionen im Betrieb (Matheis, 2010). Sind die betroffenen Funktionen unmittelbar an eine Gefährdung des Produkthanwenders und seiner Umwelt gekoppelt, gelten sie als sicherheitsrelevant. Das Konzept der funktionalen Sicherheit schreibt vor, dass dem Ausfall sicherheitsrelevanter Funktionen mittels entsprechender Maßnahmen bereits im Entwicklungsprozess vorgebeugt werden muss (ISO 26262, 2018). Eine Definition funktionaler Sicherheit mit Fokus auf E/E-Systemen nach der ISO 26262 (2018) sei im Folgenden gegeben:

#### **Definition 9: Funktionale Sicherheit**

Die Abwesenheit unzumutbarer Risiken, welche aus dem Fehlverhalten von E/E-Systemen hervorgerufen werden können.

---

<sup>22</sup> Der Version der Norm von 2004

Als verbindliche Richtlinie zur Gewährleistung der funktionalen Sicherheit in Kraftfahrzeugen gilt die ISO 26262. Sie definiert ein Vorgehensmodell, dass die Dokumentation und Umsetzung geeigneter Maßnahmen zur Vermeidung von Fehlverhalten umfasst.

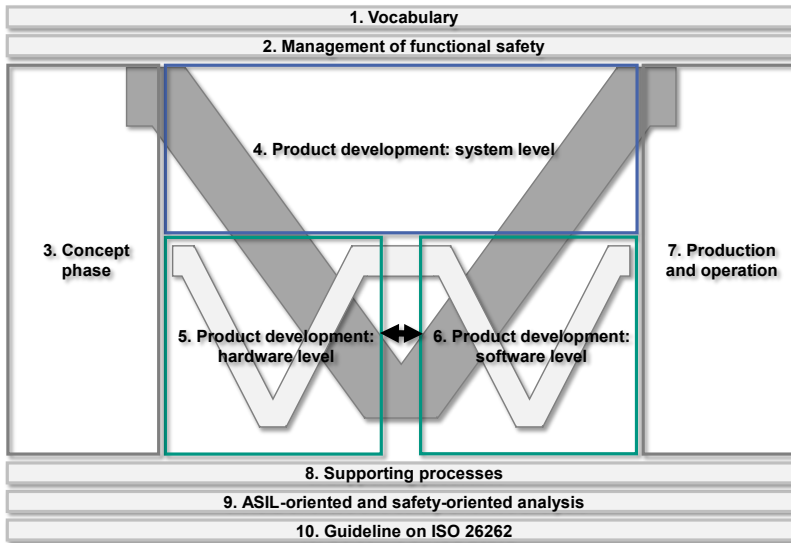


Abbildung 2.18: Übersicht des ISO 26262 Standards für sicherheitsrelevante elektrische/elektronische Systeme in Kraftfahrzeugen (ISO 26262, 2018).

Sämtliche Entwicklungsschritte, nicht als sicherheitsrelevant gekennzeichnete Funktionen und Elemente, finden keine Berücksichtigung. Abbildung 2.18 zeigt eine Übersicht der aktuellen Richtlinie. Zum Kern der Norm zählen die Konzeptphase, die Produktentwicklung (auf System-, Software- und Hardware-Level) und die Produktion und Inbetriebnahme. Die Vorgehensweise zur Umsetzung eines Sicherheitskonzeptes basiert auf dem bereits bekannten V-Modell. Alle Aktivitäten beziehen sich allerdings ausschließlich auf die Gewährleistung der funktionalen Sicherheit. Nicht-sicherheitsrelevante Funktionen erfahren keine Beachtung. Eine Anpassung des V-Modells gegenüber der VDI Richtlinie 2206 ist das Durchlaufen separater „V“s für die Domänen der Software und Hardware. Darüber hinaus sind innerhalb der „Domänen-V“s ebenfalls weitere „V“s möglich, wobei stets alle Phasen durchlaufen werden. Die Domäne der Mechanik findet im Rahmen der ISO 26262

keine gesonderte Beachtung, da alle sicherheitsrelevanten Funktionen im Steuergerät software- oder hardwareseitig realisiert werden. Mittels der Betrachtung der funktionalen Sicherheit wird der Fokus innerhalb des Produktentstehungsprozesses von Steuergeräten noch stärker auf die Rolle der Software und der Hardware gelegt. Die in Kapitel 2.1.2 bereits diskutierte Nachlagerung der Aufbau- und Verbindungstechnik wird durch diese Art der Betrachtung noch weiter forciert.

Die Einhaltung der ISO 26262 ist für alle Automobilhersteller und Zulieferer, mit ihrer Veröffentlichung, entsprechend dem Produkthaftungsgesetz zur Verbindlichkeit geworden (Löw, Pabst & Petry, 2010). Eine Hauptforderung liegt dabei in der nachverfolgbaren Dokumentation (Traceability) der funktionalen Sicherheitskonzepte und deren letztendlich technische Umsetzung auf Basis der Anforderungen und Ziele. Bedenkt man, dass sich der Großteil in der Entwicklung verwendeter Prozess- und Produktentstehungsmodelle über die Jahre hinweg etablieren musste, stellt die Integration neuer Prozesse im Rahmen der funktionalen Sicherheit keine triviale Aufgabe dar. Die ISO 26262 beschreibt lediglich einen eigenen Lebenszyklus, dessen Phasen im Produktentstehungsprozess mechatronischer Steuergeräte bisher trotz Anlehnung an bekannte Prozessmodelle oftmals nicht mit der geforderten Ausführlichkeit Rechnung getragen werden (Hillenbrand, 2012). Neben der Integration der ISO 26262 stellen die Regularien des Automotive SPICE (ASPICE) eine zweite große Herausforderung für den Entwicklungsprozess mechatronischer Steuergeräte dar. Im folgenden Abschnitt werden die Inhalte von ASPICE kurz umrissen.

### **Automotive SPICE: ASPICE**

Hinter dem Namen ASPICE verbirgt sich ein branchenspezifischer Standard des Verbandes der Automobilindustrie (VDA). Sein Ziel ist die Bewertung der Leistungsfähigkeit von Entwicklungsprozessen der software-basierten Systementwicklung in der Automobilindustrie. Die Bewertung einzelner Prozessschritte erfolgt dabei anhand spezifischer Indikatoren. Auch hier liegt der Fokus auf der Nachverfolgbarkeit der technischen Konzeptentscheide, die zur Erfüllung der Produktanforderungen getroffen wurden. Im Kern der Richtlinie stehen das ASPICE Prozess-Referenz-Modell und das ASPICE Assessment Modell. Beide orientieren sich stark an der Vorgehensweise der VDI 2206, wobei der System- und der Softwareentwicklungsprozess im Fokus stehen. Das Prozess-Referenz-Modell gibt die zur Bewertung relevanten Prozessschritte vor. System- und Softwareentwicklung sind entsprechend dem V-Modell partitioniert. Die Rahmenprozesse werden vergleichbar zur ISO 26262 dargestellt (Abbildung 2.19).

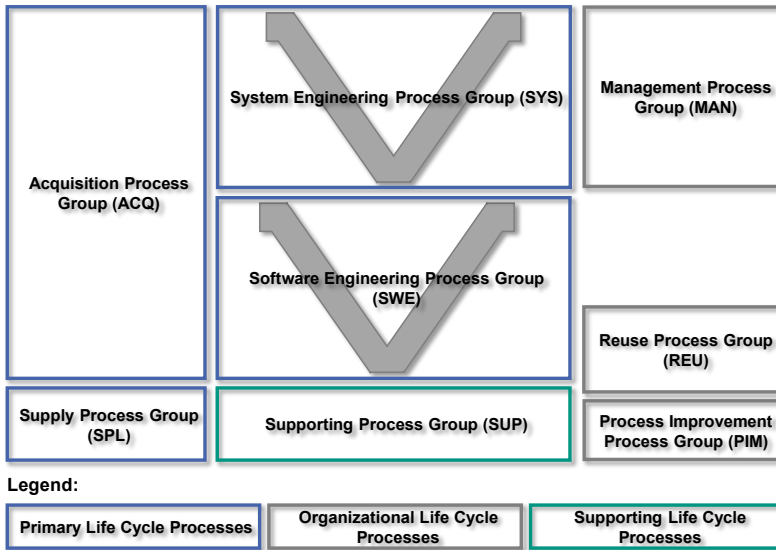


Abbildung 2.19: Übersicht des ASPICE Rahmenprozesses (ASPICE, 2017).

Basierend auf dem im Prozess-Referenz-Modell abgeleiteten Schritte gibt das ASPICE Assessment-Modell Indikatoren vor, die zur Bewertung der Prozessfähigkeit herangezogen werden. Die Bewertung resultiert in der Vergabe eines Prozessreifegrades für jeden der geprüften Schritte. Dieser reicht von „0 = unvollständige Erfassung“ der benötigten Prozesse bis zu „5 = statistisch gemessene und optimierte“ Prozesse. (ASPICE, 2017)

Ansätze zur Erweiterung des ASPICE Assessments auf den Bereich der Hardware existieren bereits, eine offizielle Integration der Konzepte auf Basis des „Plug-in“ Ansatzes steht allerdings noch aus (Schlager et al., 2018). Neben der ISO26262 findet die Mechanikentwicklung des Steuergerätes auch innerhalb der Regularien von ASPICE wenig Beachtung<sup>23</sup>.

<sup>23</sup> vgl. Kapitel 2.1.2 Nachlagerung der Mechanik, sowohl Konstruktion als auch AVT

## Entwicklung technischer Produkte und Systeme: VDI 2221

Die 1993 veröffentlichte Version der VDI Richtlinie 2221 stellt bis heute das am weitesten verbreitete Vorgehensmodell zur systematischen Entwicklung und Konstruktion im Maschinenbau dar (Mantwill, 2014). Die dort aufgezeigte Methode unterteilt den Produktentwicklungsprozess in einzelne, je nach Branche und Projekt anzupassende, Phasen. Diesen Phasen werden, ausgehend von der initialen Entwicklungsaufgabe sieben sequenziell durchzuführende Arbeitsschritte inklusive ihrer Ergebnisse, zugeordnet (Abbildung 2.20).

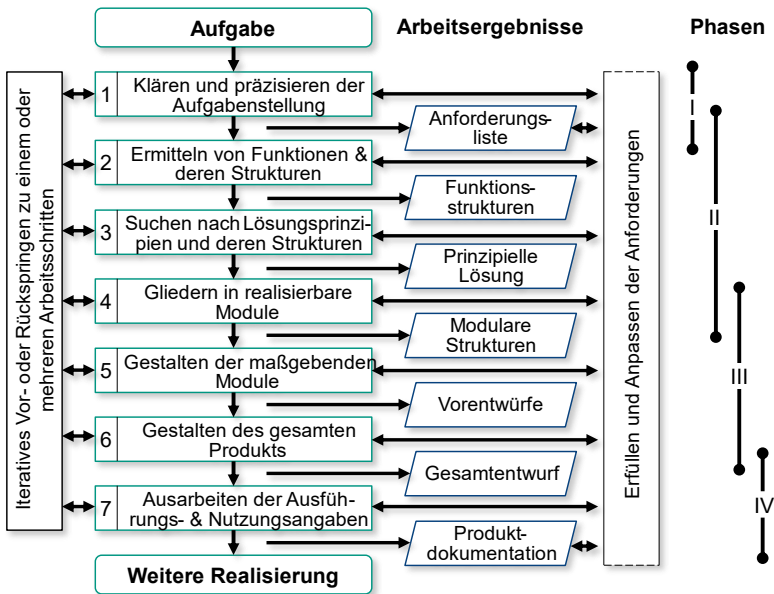


Abbildung 2.20: Allgemeines Modell der Produktentwicklung 1993 (VDI 2221, 1993).

Aufgrund vielfacher Kritik aus der industriellen Praxis und der Wissenschaft wurde die Richtlinie grundlegend überarbeitet und im März 2018 neu veröffentlicht. Im Zentrum der Kritik stand hauptsächlich die sequenzielle Darstellung der einzelnen Prozessschritte. Trotz bereits im Modell vorgesehener Iterationen suggeriert die Darstellungsform dem Anwender eine gewisse Linearität im Entwicklungsprozess.

Zusätzliche Kritik ergab sich an der strikten Trennung von Funktion und Gestalt<sup>24</sup> die nachweislich den Erkenntnisgewinn in der Entwicklung erschwert. (Albers & Wintergerst, 2014; Matthiesen 2002) Neben diesen und weiteren Punkten fand im Zuge der Überarbeitung das Prinzip der kontinuierlichen Weiterentwicklung des Zielsystems ebenfalls Berücksichtigung. Die Ergebnisse der generischen Aktivitäten werden dazu im Sinne einer prozessbegleitenden Verifizierung und Validierung mit den iterativ weiterentwickelten Anforderungen abgeglichen. Dies kann sowohl simulativ, durch reale Tests oder eine Mischform erfolgen. Für die Anwendung in realen Produktentwicklungsprozessen wurde somit die Möglichkeit geschaffen eine situationsspezifische Auswahl der Aktivitäten zu treffen und diese in vollem oder reduziertem Umfang, ein oder mehrere Male durchzuführen. Somit sind ganz wesentliche Kernbausteine der KaSPro – Karlsruher Schule für Produktentwicklung in den Standard eingeflossen. Mit der Überarbeitung wurde die Beschreibung der Entwicklungsphasen in einen eigenen Abschnitt der Norm verschoben (VDI 2221 Blatt 2, 2018). Abbildung 2.21 zeigt die aktuelle Version des allgemeinen Modells der Produktentwicklung:

---

<sup>24</sup> hier existieren einige Parallelen zur VDI Richtlinie 2206 deren die 2221 zugrunde liegt

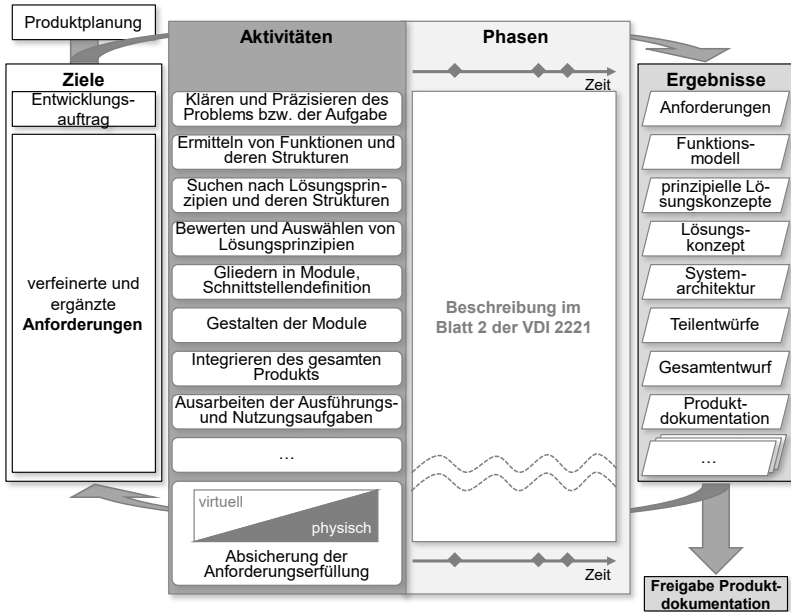


Abbildung 2.21: Allgemeines Modell der Produktentwicklung 2018 (VDI 2221 Blatt 1, 2018).

Die bisher vorgestellten Prozessmodelle tragen in ihrer ursprünglichen Form der realen Produktentstehungsprozessen intrinsischen, Iterativität (Wynn, Eckert & Clarkson, 2007) lediglich auf makroskopischer Ebene Rechnung. Zudem beschränken sie sich entweder ausschließlich auf die Produktentwicklung oder wie im Falle der ISO 26262 auf disziplinspezifische Ausschnitte des Produktentstehungsprozesses. Weitere weitverbreitete Produktentstehungsmodelle, sind bspw.:

- das **3-Zyklen-Modell** nach Gausemeier, Plass & Wenzelmann (2009), welches den Zusammenhang von Produkt- und Technologieplanung, Produktentwicklung und Produktionssystementwicklung beschreibt
- das **Münchener Vorgehensmodell** nach Lindemann (2009), welches den PEP als Netzwerk der folgenden sieben Elemente betrachtet: Ziel planen,

Ziel analysieren, Eigenschaften ermitteln, Zielerreichung absichern, Entscheidungen herbeiführen, Probleme strukturieren und Lösungsideen ermitteln

- das **erweiterte V-Modell** nach Eigner et al. (2012)

Um den Produktentstehungsprozess in seiner Tiefe beschreiben zu können bedarf es jedoch eines ganzheitlichen, individuellen Ansatzes. Das im Folgenden vorgestellte integrierte Produktentstehungsmodell – iPeM stellt ein generisches Metamodell dar, welches die geforderte Ganzheitlichkeit inkludiert.

### **Das integrierte Produktentstehungsmodell iPeM**

Das integrierte Produktentstehungsmodell - iPeM nach Albers ergänzt die Aktivitäten der Produktentwicklung, um die Schnittstellen zum operativen Management, wie beispielsweise im Stage-Ansatz nach Cooper (1990), und bildet somit ein Framework, das sich zur Beschreibung des gesamten Prozesses der Produktentstehung eignet. (Albers & Braun, 2011a) Es basiert auf der Grundhypothese nach Albers (2010), dass jeder Entwicklungsprozess individuell und einzigartig ist und ermöglicht es jederzeit flexibel und individuell auf die Gegebenheiten des abzubildenden Produktentstehungsprozesses einzugehen. Mit dem iPeM lassen sich sowohl, die in realen Produktentstehungsprozessen weitverbreiteten Iterationen, als auch sequenziell verlaufende Entwicklungsphasen abbilden. (Klingler, 2016)

### **Das ZHO-Modell**

Die Grundlage des iPeM bildet das sogenannte ZHO-Modell, welches auf den systemtheoretischen Überlegungen von Ropohl (2009) aufbaut. Das ZHO-Modell beschreibt das sozio-technische System der Produktentstehung anhand der Wechselwirkungen der von Ropohl (1975) definierten Begriffe des Zielsystems, des Handlungssystems und des Sachsystems (Objektsystems). Die Beziehungen zwischen den einzelnen Systemen und deren Bedeutung für den Produktentstehungsprozess greift Albers (2010) in seiner zweiten Hypothese auf. Demnach kann die Produktentstehung als die Transformation eines (anfangs vagen) Zielsystems in ein Objektsystem durch ein Handlungssystem beschrieben werden. (Albers & Braun 2011b)

Das **Zielsystem** umfasst alle Ziele, Anforderungen und deren Verknüpfungen, sowie geltende Randbedingungen und Restriktionen, die zur Entwicklung des Produktes erforderlich sind (Ebel, 2015). Nachfolgend sind die Elemente des



Zielsystems auf Basis der Ontologie der KaSPro – Karlsruher Schule für Produktentwicklung definiert:

**Definition 10: Ziel**

Ein Ziel ist die bewusste gedankliche Vorwegnahme eines zukünftigen Soll-Zustands, der gemeinsam vereinbart und durch aktives Handeln angestrebt oder vermieden wird. (Lohmeyer, 2012; dort nach Eiletz, 1999)

**Definition 11: Anforderung**

Eine Anforderung ist...

- eine Eigenschaft oder Bedingung, die ein System oder eine Person benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen,
- eine Eigenschaft oder Bedingung, die ein System oder eine Systemkomponente aufweisen muss, um einen Vertrag, eine Norm, eine Spezifikation oder ein anderes formell vorgegebenes Dokument zu erfüllen,
- eine dokumentierte Repräsentation einer Eigenschaft oder Bedingung, wie in den ersten beiden Punkten beschrieben

(Ebel, 2015; dort nach IEEE, 1990)

**Definition 12: Randbedingung**

Eine Randbedingung ist ein aus dem Umfeld der Entwicklung resultierender äußerer Einfluss, der von anderen verantwortet wird und daher zwar ermittelt, aber nicht eigenständig definiert oder geändert werden kann. (Albers, Ebel & Lohmeyer, 2012; Lohmeyer, 2013)

Die Beschreibung gewünschter Produkteigenschaften und –zustände innerhalb des Zielsystems erfolgt lösungsneutral. Mit der Konkretisierung von Anforderungen und Zielen, beispielsweise in Folge einer kontinuierlichen Validierung, entwickelt sich das Zielsystem ebenfalls stetig weiter (Meboldt, 2008). Es enthält neben dem geforderten Zweck des Objektsystems auch strategische Ziele und Anforderungen des Unternehmens sowie Regularien für das Handlungssystem.

Das **Objektsystem** beinhaltet sämtliche Artefakte die als (Teil-)Ergebnisse des Produktentstehungsprozesses anfallen, inklusive deren Dokumentation (Albers & Braun 2011b). Dazu zählen neben Prototypen und dem fertigen Produkt, ebenfalls

die während der Entwicklung angefallenen Skizzen, Modelle und Projektpläne (Albers & Braun, 2011a, 2011b).

Das **Handlungssystem** dient der Überführung des Zielsystems ins Objektsystem (Albers & Braun 2011b; Muschik 2011). Dies geschieht mittels geeigneter Aktivitäten, Methoden und Prozesse. Den zur Durchführung benötigten Ressourcen, wie beispielsweise Personen, Ausstattung, Entwicklungsmethoden, etc., ist dabei ein eigener Teil des Handlungssystems vorgehalten: das sogenannte Ressourcensystem.

Die erfolgreiche Entwicklung komplexer, mechatronischer Produkte setzt eine ganzheitliche Betrachtung des interdisziplinären Arbeitens voraus. Innerhalb interdisziplinärer Teams finden verschiedenste disziplinspezifische Methoden, Expertisen und Modelle Anwendung. Als logische Konsequenz entwickeln die beteiligten Disziplinen ihre eigenen Ziele und Objekte (Albers, Lohmeyer & Ebel, 2011). Mittels des ZHO-Modells lassen sich die verschiedenen Handlungsweisen unter Berücksichtigung ihrer Wechselwirkungen modellieren und im gemeinsamen Ziel- und Objektsystem harmonisieren.

### Das erweiterte ZHO-Modell

Zur Darstellung der kontinuierlichen Weiterentwicklung von Ziel- und Objektsystem als Folge eines menschenzentrierten Entwicklungsprozesses kann das erweiterte ZHO-Modell nach Albers, Lohmeyer & Ebel (2011) herangezogen werden (Abbildung 2.22).

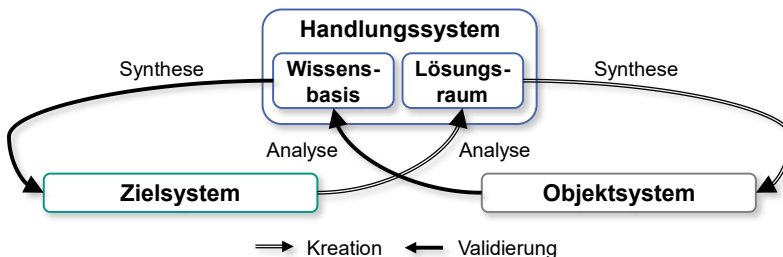


Abbildung 2.22: Das erweiterte ZHO-Modell (Albers, Lohmeyer & Ebel, 2011; Albers, Behrendt et al, 2013).

Dieses ist im Handlungssystem um die Elemente der **Wissensbasis** und des **Lösungsraums** ergänzt, die in einen fortwährenden Kreislauf aus Analyse- und Syntheseaktivitäten eingebettet sind. Die Wissensbasis beschreibt sowohl domänen- als auch fallspezifisches Wissen aus der Sicht individueller Personen. Der Lösungsraum stellt eine mentale Repräsentation der gesamten Lösungsmöglichkeiten einer spezifischen Problemstellung dar. Er wird durch die Analyse des Zielsystems konkretisiert und bildet die Basis der Erstellung neuer Elemente des Objektsystems (Lohmeyer, 2013). Eine Erweiterung der Wissensbasis kann entweder durch Nutzung zusätzlicher Ressourcen oder aus der Analyse des Objektsystems erfolgen. Ein Erkenntnisgewinn infolge der Analyse des Objektsystems kann auf Basis von Albers dritter Hypothese zur Produktentstehung nur durch Validierung erzielt werden. Der Validierung kommt somit eine zentrale Rolle in der Produktentstehung zugute (Albers, 2010). Die wachsende Wissensbasis kann mittels eines Syntheseschrittes zur weiteren Präzisierung des Zielsystems genutzt werden. Die Prozesskette ist somit geschlossen. Ein iteratives Durchlaufen des Prozesses führt zu einer kontinuierlichen Konkretisierung von Objekt- und Zielsystem und gipfelt schließlich in der Realisierung des Produktes.

Aus dem erweiterten ZHO-Modell geht hervor, dass die kontinuierliche Weiterentwicklung des Zielsystems nur auf Basis einer kontinuierlichen Validierung möglich ist (Albers, Ebel & Lohmeyer 2011). Im Falle mechatronischer Steuergeräte herrscht gerade in den frühen Phasen der Entwicklung eine Unsicherheit in Bezug auf die Verlässlichkeit von Anforderungen und Produktzielen. Diese Unsicherheit begründet sich auf der Interdisziplinarität im Entwicklungsprozess (Ehrlenspiel, 2009), der hohen Systemkomplexität (Haberfellner et al., 2012) und auf dem immer größer werdenden Funktionsumfang der Steuergeräte (Reif, 2011). Vor diesem Hintergrund ergibt sich die Forderung bereits zu Beginn der Produktentwicklung geeignete Validierungsaktivitäten vorzusehen. Erkenntnisse neuester Forschungsarbeiten der Karlsruher Schule der Produktentwicklung - KaSPro zeigen beispielsweise die Bedeutung einer frühen Validierung für den Produktentstehungsprozess und deren mögliche Umsetzung mittels VR-Technologien (Reinemann, 2021).

### **Allgemeiner Problemlösungsprozess SPALTEN**

Um der Iterativität der einzelnen Aktivitäten des Produktentstehungsprozesses gerecht zu werden, bedarf es eines flexiblen Problemlösungsprozesses, der zur Abbildung der Iterationen geeignet ist. Ein solcher Problemlösungsprozess verbirgt sich hinter dem von Albers et al. (2005) entwickelten SPALTEN. Der Begriff SPALTEN steht dabei als Akronym für die sieben Aktivitäten der Problemlösung. Diese sind:

- **Situationsanalyse**
- **Problemeingrenzung**
- **Alternative Lösungen aufzeigen**
- **Lösungsauswahl**
- **Tragweitenanalyse**
- **Entscheiden und Umsetzen**
- **Nachbereiten und Lernen**

Vor dem systematischen Durchlaufen der Aktivitäten ist zunächst die Formation eines geeigneten Problemlösungsteams (PLT) vorzusehen. Die einzelnen Prozessschritte lassen sich wiederum selbst mittels SPALTEN strukturieren und lösen (dabei müssen allerdings nicht alle sieben Aktivitäten erneut durchlaufen werden). Diese Eigenschaft bezeichnen Albers, Braun & Muschik (2010) als fraktalen Charakter von SPALTEN. Der fraktale Charakter und darin das situationsbedingte Durchlaufen einzelner SPALTEN-Schritte verdeutlicht die Eignung von SPALTEN zur Anwendung im iterativen Produktentstehungsprozess (Albers, Reiß, Bursac & Breitschuh, 2016). Hiermit grenzt sich der SPALTEN-Prozess zudem entscheidend von anderen gängigen Problemlösungsmethoden ab. (Haberfellner et al. 2012; Daenzer & Huber, 1994; Ehrlenspiel, 2009)

Im Kern seines integrierten Produktentstehungsmodells iPeM verbindet Albers auf Basis seiner fünf Hypothesen über den Entwicklungsprozess (Albers, 2010) die Überlegungen zum erweiterten ZHO-Modell mit dem universellen auf Iterativität ausgelegten Problemlösungsprozess SPALTEN. Durch die Integration der Aktivitäten der Produktentwicklung und der Sicht des operativen Managements auf den Produktentstehungsprozess wird das iPeM zu einem ganzheitlichen Modell der Produktentstehung (Albers & Braun, 2011a). Die ganzheitliche Denkweise des iPeM stellt die kontinuierliche Validierung ins Zentrum der Aktivitäten und berücksichtigt gleichzeitig die Interdisziplinarität und Iterativität moderner Entwicklungsprozesse. Es eignet sich daher besonders zur Abbildung des Produktentstehungsprozesses komplexer, interdisziplinärer mechatronischer Produkte und dient der vorliegenden Arbeit als wichtige Grundlage.

Abbildung 2.23 illustriert den schematischen Aufbau des iPeM und seiner Bestandteile:

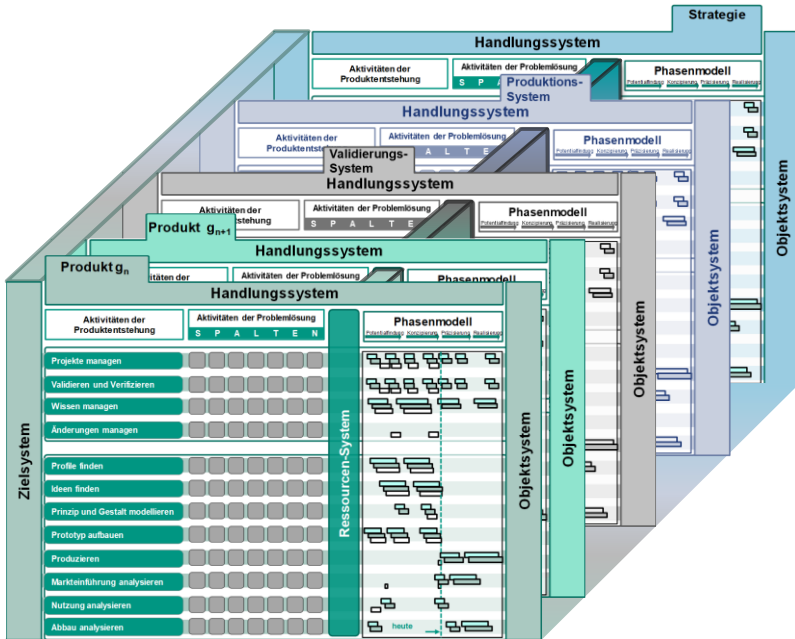


Abbildung 2.23: Das iPeM – integrierte Produktentstehungsmodell (Albers, Reiß, Bursac & Richter, 2016).

Im Zentrum des iPeM steht das Handlungssystem, welches, gemäß der Grundlagen des ZHO-Modells, der Transformation eines Zielsystems in ein Objektsystem dient. Das Handlungssystem selbst, ist in einen statischen und einen dynamischen Teil untergliedert. Der statische Teil ergibt sich zum einen durch das Ressourcensystem und zum anderen durch die sogenannte **Aktivitätenmatrix**. Diese verknüpft die Aktivitäten der Produkterstellung (Makroaktivitäten) mit den Aktivitäten des Problemlösungsprozesses SPALTEN (Mikroaktivitäten). Die Aktivitätenmatrix stellt darüber hinaus keine zeitliche Vorgabe der Aktivitäten dar und ist daher sowohl generisch als auch projektungebunden zu betrachten (Albers, Reiß, Bursac & Richter, 2016; Reiß, 2018).

Zur zeitlichen Planung, Terminierung und Verknüpfung der einzelnen Aktivitäten untereinander und zur Einordnung in einen konkreten Projekthintergrund dient das, im dynamischen Teil des Handlungssystems verortete **Phasenmodell**. Es

veranschaulicht die Verknüpfung der Entwicklungstätigkeiten mit der Sicht des operativen Managements anhand diverser Projektverläufe. Wilmsen, Dühr, Heimicke & Albers (2019) unterscheiden hier zwischen dem **Referenz-**, dem **SOLL-** und dem **IST-Prozess**. Mittels des Referenzprozesses und dem dazugehörigen Referenzprozessmodell können die Erfahrungen bereits erfolgreich durchgeführter Projekte im Sinne der kontinuierlichen Weiterentwicklung, direkt in zukünftige Projekte einfließen. Auf dieser Basis kann, der für ein konkretes Entwicklungsprojekt erwartbare SOLL-Prozess im entsprechenden Modell aufgeplant werden. Die Visualisierung des tatsächlichen Projektverlaufs findet im Rahmen des IST-Prozesses statt. (Albers & Braun 2011a; Albers, Reiß, Bursac & Richter 2016; Reiß, 2018; Wilmsen et al., 2019)

Die Logik der iterativen Problemlösung anhand vorgegebener Aktivitäten und deren Aufplanung anhand eines Phasenmodells kann vom **Produkt (G<sub>n</sub>)** und dessen **Generationen (G<sub>i</sub>)** auf die Entwicklung des **Validierungssystems**, des **Produktionssystems** als auch der **Unternehmensstrategie** übertragen werden. Im iPeM ist dies mittels verschiedener Ebenen berücksichtigt. (Albers, Reiß, Bursac & Richter, 2016)

Die Betrachtung des System-Tripels illustriert den Produktentstehungsprozess als Übergang der Informationen des Zielsystems in die materiellen Elemente des Objektsystems. In diesem Transformationsprozess spielt die Gestaltung der Produktarchitektur eine zentrale Rolle. Sie verknüpft technische Funktionen mit physischen Umsetzungen bzw. Lösungen. Der folgende Abschnitt dient der Definition der Produktarchitektur und gibt eine Einführung in deren verschiedene Dimensionen und Ausprägungen.

## 2.2.4 Produktarchitektur

Mit der Festlegung der Produktarchitektur wird der spätere wirtschaftliche Erfolg eines Produktes maßgeblich beeinflusst (Feldhusen, Grote, Göpfert & Tretow, 2013). Sie umfasst neben geforderten Produktfunktionen und –eigenschaften, auch Modulstrukturen und prägt darüber die spätere Varianz des Produktes. Für eine detaillierte Übersicht relevanter Literatur im Bereich der Produktarchitekturen sei auf Jiao et al. verwiesen (Jiao et al., 2007). Die Auswirkungen bei der Auswahl einer geeigneten Produktarchitektur erstrecken sich über die komplette Wertschöpfungskette (Feldhusen, Grote, Göpfert & Tretow, 2013). Auf Basis des ZHO-Modells ist der Produktentstehungsprozess durch die Überführung von Informationen (Zielsystem) in eine reale Umsetzung (Elemente des Objektsystems) charakterisiert. Es existieren folglich zwei unterschiedliche Sichten auf ein Produkt. Zum einen immateriell in Form von Erfordernissen und zum anderen materiell

anhand realer Baugruppen und Komponenten. Mittels beider Sichten lässt sich ein Produkt je nach Zielsetzung clustern und hierarchisch strukturieren. In der Funktionsstruktur findet sich die Strukturierung des Produktes anhand geforderter Funktionen wieder (Albers & Wintergerst, 2014). Die physische Clusterung des Produktes anhand von Baugruppen und Komponenten bezeichnet man als Produkt- bzw. Baustruktur. Während die Funktionsstruktur weitestgehend abstrakt und neutral gegenüber anzuwendender Wirkprinzipien gehalten wird, stellt die Produktstruktur genau eine konkrete Lösung zur Realisierung der geforderten Produktfunktionen dar. Für die Umsetzung einer Funktionsstruktur sind meist mehrere Produktstrukturen möglich (Pahl, Beitz et al., 2007). Der Transformationsprozess und damit die eindeutige Verbindung einer abstrakten Funktionsstruktur mit genau einer konkreten Produktstruktur gilt als Kernaufgabe der Produktarchitekturgestaltung (Göpfert, 1998). Ihre Definition entsprechend der KaSPro – Karlsruher Schule für Produktentwicklung sei im Folgenden gegeben:

**Definition 13: Produktarchitektur**

Die Produktarchitektur umfasst die Funktionsstruktur, die Produktstruktur (bzw. Baustruktur), sowie sämtliche Beziehungen zur Transformation zwischen den beiden Strukturen. (in Anlehnung an Albers & Wintergerst, 2014)

Die Gestaltung der Produktarchitektur beruht auf dem Dekompositionsprinzip der Funktionsstruktur und der Zusammenfassung von Komponenten zu Baugruppen in der Produktstruktur (Ulrich, 1995; Feldhusen, Grothe, Göpfert & Tretow, 2013; Van Wie et al, 2003) (Abbildung 2.24).

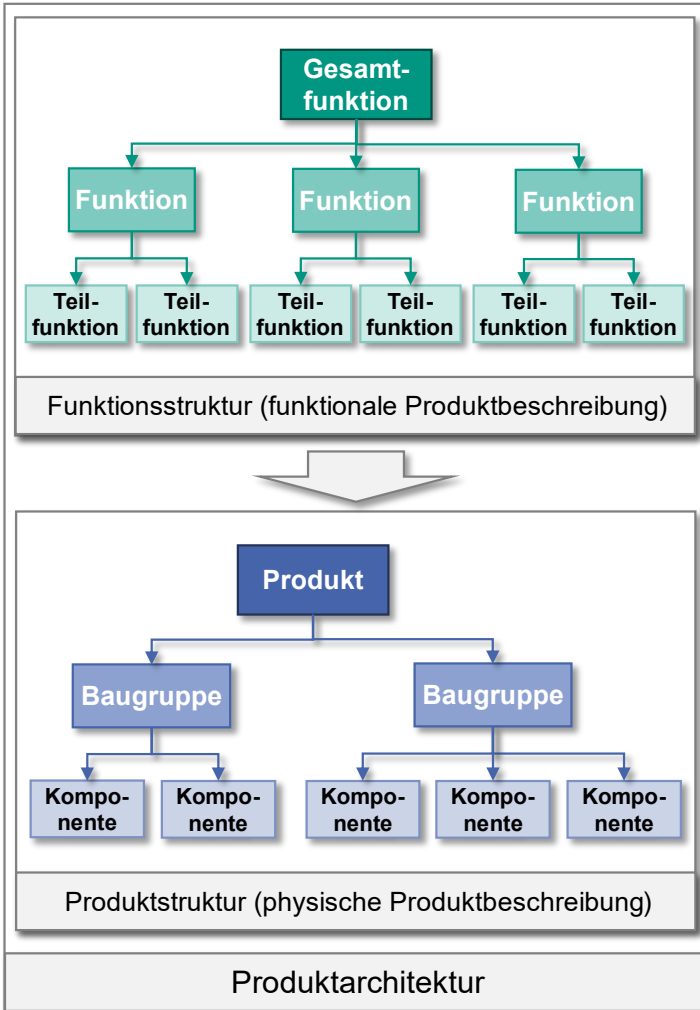


Abbildung 2.24: Darstellung der Produktarchitektur auf Basis der Transformation einer Funktionsstruktur in eine Produktstruktur nach Göpfert (1998).



## Gestaltung der Funktionsstruktur

Zur Dekomposition eines Produktes auf funktionaler Ebene wird dessen Gesamtfunktion so weit in Teilfunktionen untergliedert bis eine technische Umsetzung jener Funktionalität in Form von Komponenten und Baugruppen bzw. Wirkprinzipien denkbar ist. Die Aufteilung einer Gesamtfunktion in ihre Teilfunktionen ist gemäß der Systemtheorie, durch unterschiedliche Möglichkeiten der Elementpartitionierung sowie deren Verknüpfung von Grund auf variant. Die Verbindung von Teilfunktionen einer Ebene erfolgt mittels Flüssen von Energie, Stoff oder Information (Pahl, Beitz et al., 2007; Steffen, 2007; Stone, 1997). Diese kann je nach funktionaler Partitionierung auf unterschiedliche Weise erfolgen (Abbildung 2.25):

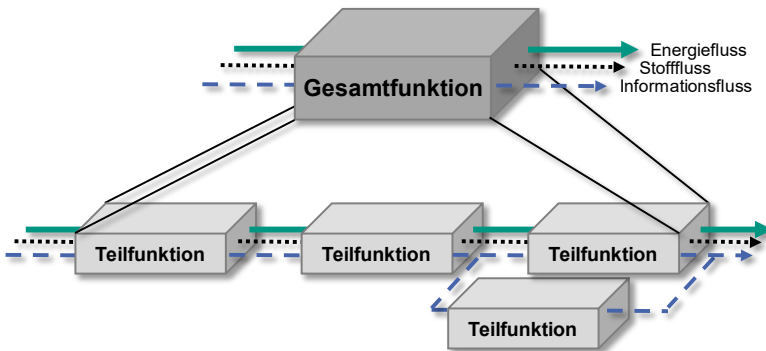


Abbildung 2.25: Funktionale Dekomposition eines Systems (Feldhusen, Grote, Göpfert & Tretow, 2013). Gezeigt ist die Zerlegung einer Gesamtfunktion in Teilfunktionen.

In der Literatur finden sich umfassende Arbeiten zur methodischen Ableitung funktionaler Strukturen siehe Stone (1997), Svendsen & Thorp Hansen (1993), Pahl, Beitz et al. (2007), Mistree et al. (1993), um nur einige zu nennen. Letztendlich bleibt der Dekompositionsprozess aber eine zentrale Entwicklungstätigkeit. Stone (1997) spricht im Zusammenhang von funktionsbasierter Entwicklung sogar vom wichtigsten Entwicklungsschritt überhaupt. Dieser ist dem subjektiven Verständnis der beteiligten Entwickler unterworfen. Ausgangspunkt der funktionalen Dekomposition sind die Produktziele und -anforderungen des Kunden, die auf Basis von Referenzsystemen in möglichst lösungsneutrale, technische Funktionen übersetzt werden. (Albers & Wintergerst, 2014; Moeser, Albers & Klingler, 2015;

Pahl, Beitz et al., 2007; VDI 2206, 2004; Yu, Gonzalez-Zugasti & Otto, 1999). An dieser Stelle sei nochmals auf die Kritik an der lösungsneutralen Formulierung von Produktfunktionen hingewiesen. Taylor & Henderson (1994) sehen in der Dekomposition bereits eine Entscheidung, die eine bestimmte Lösung impliziert und andere Lösungsprinzipien ausschließt. Zusätzliche Kritik besteht an der klaren Zuordnung von Funktionsumfängen zu einzelnen Komponenten und Baugruppen. Albers & Wintergerst (2014) erkennen, dass Teilsysteme ihre angedachte Funktionalität nur im Zusammenspiel mit ihren umgebenden Systemen erfüllen. Daher ist eine Zuordnung von Funktionen zu Bauteilen im Rahmen dieser Arbeit, nur im Sinne einer maßgeblichen Beteiligung an der Funktionserfüllung zu treffen.

### **Gestaltung der Produktstruktur**

Die Ableitung einer geeigneten Produktstruktur muss im Entwicklungsprozess stets mit Hinblick auf die gesamte Wertschöpfungskette sowie unter Berücksichtigung des Portfolios erfolgen (Steffen, 2007; Pahl, Beitz et al., 2007; Zamirowski & Otto, 1999). Diese Einflüsse wirken sich sowohl auf die horizontalen als auch die vertikalen Beziehungen innerhalb der Produktstruktur aus. In der Literatur werden die verschiedenen Produktstrukturtypen häufig auf die gesamte Produktarchitektur bezogen (Steffen, 2007; Jiao et al., 2007)<sup>25</sup>. Im Folgenden soll ein kurzer Überblick grundlegender Produktstrukturtypen gegeben werden. Unter der vertikalen Strukturierung eines Produktes versteht man die hierarchischen Beziehungen zwischen einzelnen Komponenten und deren übergeordneten Baugruppen. Zur Illustration diene die exemplarische Gliederung eines elektrischen Antriebs (Abbildung 2.26). Während die vertikale Strukturierung die Aggregation von Elementen über die Hierarchieebenen beschreibt, finden die Beziehungen der Elemente einer Hierarchieebene im Rahmen der horizontalen Strukturierung Beachtung. Die hierarchische Zuordnung des Produktes ist somit direkt von seiner horizontalen Partitionierung abhängig.

Der horizontale Aspekt einer Produktstruktur dient der Beschreibung der Wechselwirkungen verschiedener Systemelemente, die sich auf einer Hierarchieebene befinden. Diese Sichtweise findet sich wiederum im strukturellen Konzept der Systemtheorie nach Ropohl (2009).

---

<sup>25</sup> Steffen (2007) spricht hier beispielsweise von modularen Produktstrukturen, während Jiao et al. (2007) den Begriff der modularen Produktarchitektur verwenden.

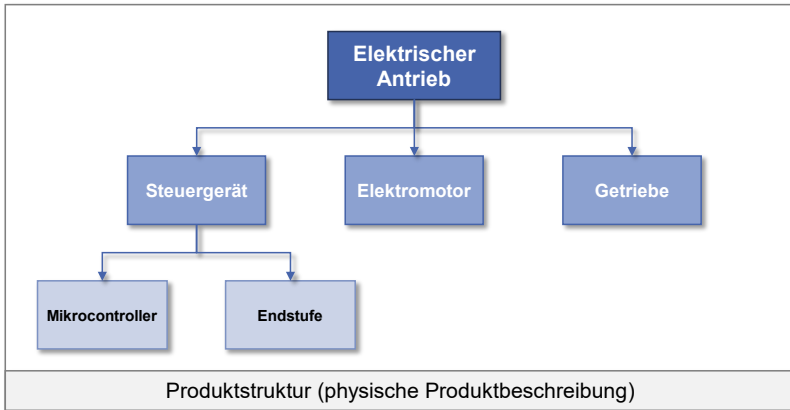


Abbildung 2.26: Ausschnitt der vertikalen Strukturierung eines elektrischen Antriebs bestehend aus Steuergerät, Elektromotor und Getriebe. Eigene Darstellung nach Pahl, Beitz et al. (2007)

Hierbei können auf Basis der entwickelten Funktionsstruktur grundsätzlich zwei Wege beschriftet werden:

- **Integrale Produktstrukturen**

Integrale Produktstrukturen verfügen durchgängig über starke funktionale und geometrische Abhängigkeiten. Die Beziehungen der einzelnen Elemente sind dabei entlang der Produktstruktur verworren bzw. unstrukturiert. Eine physikalische Trennung einzelner Elemente kann in der Regel nicht erfolgen (Pahl, Beitz et al., 2007; Steffen, 2007). Aus diesem Grund ist eine hierarchische Gliederung innerhalb der integralen Struktur nur im Groben möglich. Ebenso ist eine Zuordnung maßgeblich an der Funktionserfüllung beteiligter Systemelemente zu ihren Funktionen nur bedingt möglich. An dieser Stelle sei besonders auf die Arbeiten in Zusammenhang mit dem Funktions-Gestalt-Übergang der KaSPro – Karlsruher Schule für Produktentwicklung hingewiesen (vgl. Kapitel 2.2.5).

Auf dem Bereich der Steuergeräteentwicklung können sogenannte Molded Interconnect Devices als Beispiele integraler Strukturen dienen. Mittels dieser Technologie können ganze Leistungsendstufen (z.B. B6-Bückenschaltung: 6 MOSFETS inklusive der Leiterbahnen und deren Beschaltung) in einem Bauteil verschmelzen (Steffen, 2007 dort auf Basis von Forschungsvereinigung Räumliche Elektronische Baugruppen, 2004 und Meier, 2002). Eine Trennung der

Komponenten ist im Nachgang nicht mehr möglich. Hieraus wird die klare Optimierung integraler Produktstrukturen auf spezifische Anforderungen bzw. Anwendungsfälle ersichtlich. Diese liegen beispielsweise in einer Bauraumoptimierung durch Miniaturisierung, Leichtbau oder einer Minimierung der Herstellkosten. Die Wiederverwendbarkeit integraler Strukturen oder Systemelemente ist nur innerhalb enger Grenzen möglich.

- **Modulare Produktstrukturen**

Im Sinne einer modularen Produktarchitektur erfolgt die Zusammenfassung einzelner Systemelemente zu in sich geschlossenen Einheiten. Diese geschlossenen Einheiten zeichnen sich durch starke interne Abhängigkeiten aus, sind vom Verhalten benachbarter Elemente jedoch weitestgehend unabhängig (Simon, 1962; Göpfert, 1998). Simon (1962) hat in diesem Zusammenhang den Begriff „nearly decomposable“ geprägt. Die Abhängigkeiten können sowohl auf Basis der zugeordneten Funktionalitäten im Sinne der Produktarchitektur existieren oder durch Folgeprozesse entlang der Wertschöpfungskette bestimmt werden. In modular konzipierten Produkten sind die Schnittstellen zwischen den in sich abgeschlossenen Elementen gering. Benachbarte Elemente werden im Systemkontext als Black-Box betrachtet. Aufgrund der klaren Abgrenzung der verschiedenen Systemelemente ist eine hierarchische Aufteilung modularer Produkte ohne weiteres möglich. Die klare Abgrenzung der weitgehend unabhängigen Systemelemente beeinflusst ebenfalls die Anpassung der Organisationsstruktur entlang der Wertschöpfungskette (Steffen, 2007; Göpfert, 1998). Im Gegensatz zu integralen Produktstrukturen weisen modulare Produktarchitekturen eine hohe Flexibilität und Wiederverwendbarkeit auf. Sie sind daher Grundlage diverser Standardisierungsmethoden im Bereich des Variantenmanagements (Albers, Scherer, Bursac & Rachenkova, 2015). Eine detaillierte Übersicht der Vor- und Nachteile modularer und integraler Produktstrukturen findet sich bei Steffen (2007) sowie bei Pahl, Beitz et al. (2007). In der Literatur existieren umfassende Arbeiten zur Ableitung modularer Produktarchitekturen (vgl. Kapitel 2.4.1). Um eine Vorwegnahme zu vermeiden, werden diese erst an späterer Stelle eingeführt. Aus dem gleichen Grund wurde hier von einer Verwendung des Modulbegriffes abgesehen. Dieser wird im Kapitel der Baukastenentwicklung als Standardisierungsmethode definiert und im Zuge relevanter Methoden zur modularen Produktstrukturierung entsprechend des Verständnisses der KaSPro – Karlsruher Schule für Produktentwicklung näher beleuchtet.

Erscheinen die vorgestellten Methoden der horizontalen Produktstrukturierung zunächst als Gegensätze, schließen sie sich in der Anwendung keinesfalls aus. Bei modular aufgebauten Systemen lässt sich ein gewisser fraktaler Charakter

erkennen. Dieser ist der Natur, dass die Umsetzung der in sich abgeschlossenen Systemelemente entweder integral erfolgt oder sich ebenfalls modular auf den nachfolgenden Hierarchieebenen fortsetzt. Gründe für die Vermischung der Strukturtypen sind charakteristisch für mechatronische Steuergeräte. So sprechen vorherrschende Bauraum-, Gewicht- und Leistungseinschränkungen für eine integrale Umsetzung der Produkte, die hohen Stückzahlen und die mittlere Variantenanzahl entlang des Portfolios allerdings für einen modularen Aufbau (Steffen, 2007) (vgl. Kapitel 2.1.1 Abbildung 2.3).

### **2.2.5 Funktions-Gestalt-Zusammenhang**

Im Rahmen der Produktarchitekturgestaltung wird der Übergang von Funktionsstrukturen zur physischen Umsetzung als Kernaktivität im Entwicklungsprozess identifiziert. Gerade in den Bereichen der Software- und Hardwareentwicklung ist die Auffassung einer strikten Trennung von Funktionsmodellen und deren Umsetzung weit verbreitet (Dröschel & Wiemers, 1999)<sup>26</sup>. Auf dieser Basis soll ein von den Gestaltvorstellungen losgelöster Entwicklungsvorgang gewährleistet werden (Roth, 2000; Koller, 1998; Ropohl, 2009, Feldhusen, Grote, Nagarajah et al., 2013). Albers & Wintergerst (2014) hinterfragen diese Denkweise in Berufung auf die Forschung von Eckert. Dieser vertritt die Annahme, dass jegliche Vorstellung einer Funktion in der Konstruktionspraxis mit einer Idee der physischen Umsetzung einhergeht (Eckert et al., 2011). Auf dieser Basis entwickeln Albers et al. (Albers & Wintergerst 2014; Albers, Braun et al., 2009), einen Modellansatzes, der dem menschlichen Gedankengang Rechnung trägt, indem Funktionen und konkrete Gestalt unter ständiger gegenseitiger Berücksichtigung in integrierten Produktmodellen beschrieben werden. Der Modellansatz trägt den Namen Contact and Channel Approach (C&C<sup>2</sup>-A).

Der C&C<sup>2</sup>-A beschreibt, dass zur Erfüllung einer Funktion mindestens zwei Wirkflächenpaare (WFP) und eine die WFP's verbindende Leitstützstruktur (LSS) erforderlich sind (Albers & Wintergerst, 2014). Einflüsse benachbarter Systemelemente oder der Systemumgebung werden mittels sogenannter Connectoren abgebildet. Alle Wirkflächenpaare, Leitstützstrukturen und Connectoren, die zur Erfüllung einer Funktion in einem bestimmten Betriebszustand beitragen bilden ein sogenanntes Wirk-Netz. Betrachtet man alle möglichen Wirk-

---

<sup>26</sup> In Bezug auf die Softwareentwicklung

Netze eines Systems spricht man von seiner Wirk-Struktur (Albers & Wintergerst, 2014). Abbildung 2.27 illustriert den C&C<sup>2</sup>-A am Beispiel eines Hybridantriebes:

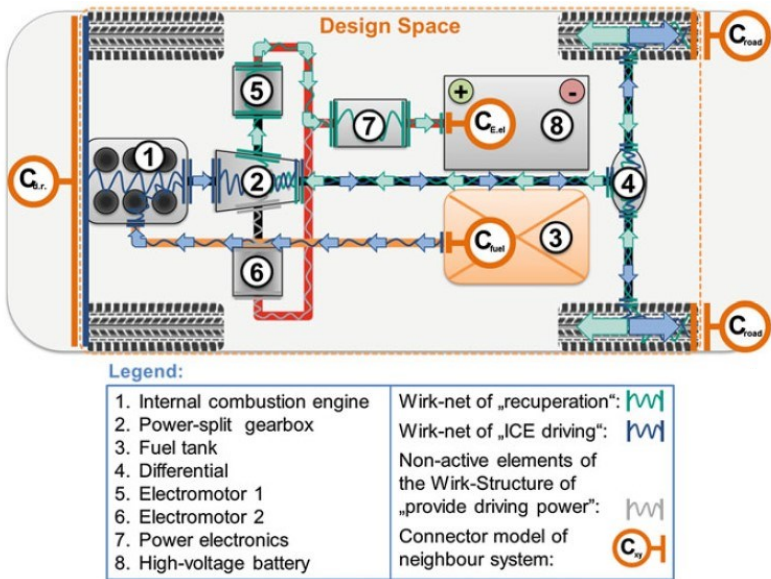


Abbildung 2.27: C&C<sup>2</sup>-A Modell am Beispiel eines Hybridantriebes nach Albers & Wintergerst (2014).

Eine große Stärke des C&C<sup>2</sup>-A ist seine Eignung zur Abbildung verschiedenster Abstraktionsgrade von Funktions-Gestalt-Zusammenhängen. Das Verständnis der Gestalt eines Produktes als die quantitative Beschreibung der Summe aller Merkmale<sup>27</sup> und Struktureigenschaften der Produktstruktur umfasst sehr detaillierte, aber auch sehr abstrakte Darstellungen (Zingel, 2013). Neben der Anwendung im Rahmen realer Umsetzungen oder detaillierter Produktmodelle kann der C&C<sup>2</sup>-A folglich auch zur Analyse deutlich abstrakterer Produktarchitekturen herangezogen werden. Aktuelle Forschung auf dem Gebiet des C&C<sup>2</sup>-A findet sich beispielsweise

<sup>27</sup> Merkmale werden hierbei als Attribute eines technischen Systems, die der Entwickler direkt festlegen kann, verstanden (Zingel, 2013)

bei Tröster et al. (2021). In Bezug auf die in Kapitel 2.2.4 angesprochene Produktarchitektur und deren Aufteilung in integrale und modulare Produktstrukturen kann der C&C<sup>2</sup>-A ebenfalls dazu herangezogen werden, den unterschiedlichen Aufbau anhand der höheren Anzahl an WPF's der modularen Produktstruktur und dem unterschiedlichen resultierenden Wirk-Netz zu erklären.

## 2.2.6 Systems Modelling Language

Zur Darstellung rechnerbasierter Systemmodelle wurde 2007 im Zuge der MBSE Initiative<sup>28</sup> (Friedenthal, Griego & Sampson, 2007) eine deskriptive, zur grafischen Abbildung aufbereitete Modellierungssprache eingeführt: die sogenannte Systems Modelling Language (SysML) (Object Management Group, 2007). Ziel der Unternehmung war es, mit der SysML einen Sprachstandard<sup>29</sup> zu schaffen, welcher die Bildung eines einheitlichen Systemverständnisses unterstützt und die Kommunikation in interdisziplinären Teams erleichtert (Albers, Munker, Zingel & Behrendt, 2013; Alt, 2012). Damit die Sprache zur Abbildung der Systemsichten aller, an der Entwicklung mechatronischer Produkte beteiligter Disziplinen geeignet ist, muss diese möglichst unabhängig von spezifischen Disziplinen und daher abstrakt gestaltet sein (Weilkiens, 2007; Ehrlenspiel & Meerkamm, 2013). In Bezug auf die Modellierung von Funktionsstrukturen existieren diverse Studien, die die Eignung und Vorteile der SysML auf diesem Gebiet betonen (Albers & Zingel 2013a). Da mechatronische Steuergeräte im Entwicklungsprozess vornehmlich funktionsorientiert entwickelt werden zeichnet sich die Eignung der SysML zur Abbildung der Produktarchitektur ab. Die Verwendung der SysML als Abbildungssprache für Produktmodelle im Rahmen der Baukastenentwicklung wurde in diversen Forschungsarbeiten am IPEK – Institut für Produktentwicklung Karlsruhe untersucht (Albers, Rapp et al., 2017). Besonders hervorzuheben sind dabei die Arbeiten von Bursac (2016) und Scherer (2016). Für die Baukastenentwicklung mechatronischer Steuergeräte bleibt allerdings noch zu prüfen, ob sich die SysML dazu eignet die Strukturierung und Anwendung von Baukästen sinnvoll zu unterstützen.

---

<sup>28</sup> Die Entwicklung wurde hierbei von INCOSE in Zusammenarbeit mit der Object Management Group (OMG) geleitet

<sup>29</sup> Standards jeglicher Art sind im SE von besonderer Bedeutung, um Konsistenz durch Projekte und Organisationen zu wahren (Reichwein & Paredis, 2011)

Im Bereich mechatronischer Systeme finden neben der SysML diverse weitere Sprachen ihre Anwendung zur Systemmodellierung. Nachfolgend sei eine Übersicht der aus Sicht dieser Arbeit relevantesten Vertreter gegeben<sup>30</sup>:

Sowohl im industriellen als auch universitären Umfeld findet die Entwicklungsumgebung **MATLAB** zahlreiche Anwendung. Diese dient vornehmlich der numerischen Berechnung matrixbasierter Ausdrücke. Die Modellbildung erfolgt mittels einer eigenen Programmiersprache. Die Erweiterung **Simulink** ermöglicht die grafische Modellierung, Simulation und Analyse dynamischer Systeme in Form von Blockschaltbildern. Im Bereich der Steuergeräteentwicklung werden Simulink-Modelle bspw. zur Modellierung von Regelkreisen genutzt. Der Fokus von MATLAB/Simulink liegt klar auf der quantitativen Darstellung von Simulationsmodellen. Die quantitative Darstellung von Modellen folgt dabei der Definition von Eigner et al. (2012). Durchgängige, deskriptive Systemdarstellungen anhand von Anforderungen, Funktionen und der technischen Umsetzung sind für diese Entwicklungsumgebung weniger vorgesehen (Zingel, 2013; Alt, 2012; Qamar et al., 2009). Da es sich bei MATLAB bzw. Simulink um ein proprietäres Werkzeug handelt, ist eine Verknüpfung dort explizierter, domänenspezifischer (Qamar et al., 2009) Modelle zu übergeordneten Systemmodellen erschwert und lediglich über einen föderierten Ansatz möglich. (Zingel, 2013) Im Rahmen der Spezifizierungstechnik für komplexe mechatronische Systeme **CONSENS** (Conceptual Design Specification Technique for the Engineering of Complex Systems) wurde eine SysML-ähnliche Sprache entwickelt. Diese liefert zusätzlich spezifische, zur Abbildung von Wirkstrukturen optimierte Systemelemente. Aufgrund des ohnehin geringen Unterschiedes zur SysML und der fehlenden Standardkonformität, wurde im Nachgang von Iwanek et al. (2013) ein SysML-Profil entwickelt, welches die Abbildung CONSENS-konformer Modelle in SysML ermöglicht. Zur Erleichterung des Austausches von Softwarearchitekturen zwischen Automobil-, Steuergeräte- und Mikrocontrollerherstellern wurde die modellbasierte Konfigurations- und Beschreibungssprache **AUTOSAR** (Automotive Open System Architecture) geschaffen. Mittels AUTOSAR lassen sich Software-Komponenten und Systemkonfigurationen mit besonderem Hinblick auf deren Skalierbarkeit und Übertragbarkeit abbilden. Aufbauend auf dem AUTOSAR-Metamodell existieren diverse Sprachen zur Beschreibung der Wechselwirkungen innerhalb eines gesamten E/E-Systems. Eine detaillierte Übersicht der existierenden Sprachen liefert Zingel (2013) sowie Albers & Zingel (2013a). Neben den Vorgaben tool- und

---

<sup>30</sup> Detaillierte Übersichten über verwendete Sprachstandards finden sich ebenfalls bei Reichwein & Paredis (2011), sowie Zingel (2013)



domänenspezifischer Sprachstandards bilden sich in interdisziplinären Entwicklungsprozessen häufig **eigene Sprachstandards** aus. Diese sind im Allgemeinen mit einer Abbildung in verschiedenen Microsoftanwendungen (z.B. Powerpoint, Visio) (Alt, 2012) verbunden und somit Überbleibsel dokumentenzentrierter Entwicklung.

Als Grundlage der SysML gilt die Unified Modelling Language (UML). Diese findet ihren Ursprung in der Softwareentwicklung und ist dort bereits seit Jahrzehnten anerkannter Standard zur objektorientierten Modellierung (Korff, 2008; Weilkiens, 2007). In der SysML wurde sich größtenteils der Elemente und der Syntax der UML 2 bedient. Die SysML wird daher häufig als UML-Dialekt bezeichnet (Kaffenberger et al., 2012). Um den Anforderungen an die Modellierung mechanischer und mechatronischer Systeme zu entsprechen, wurden die aus der UML übernommenen Elemente, um eigene, zur Abbildung von Anforderungen, Funktions- und Baustrukturen geeignete Sprachelemente und Diagramme ergänzt (Zingel, 2013; Alt, 2012). Die graphische Modellierung findet in der SysML in anwendungsspezifischen Diagrammen statt. Diese dienen der objektorientierten Modellbildung der Systemelemente und deren Beziehungen und sind gleichzeitig als entwicklerspezifische Sichten (vgl. Kapitel 2.2.2) auf das Gesamtsystem zu verstehen (Weilkiens, 2007). Die in der SysML verwendeten Diagramme lassen sich in drei verschiedene Diagrammtypen untergliedern: Verhaltensdiagramme, Strukturdiagramme und Anforderungsdiagramme. Abbildung 2.28 zeigt eine Übersicht aller in der SysML enthaltenen Diagrammtypen und deren Unterschiede gegenüber der Verwendung in der UML.

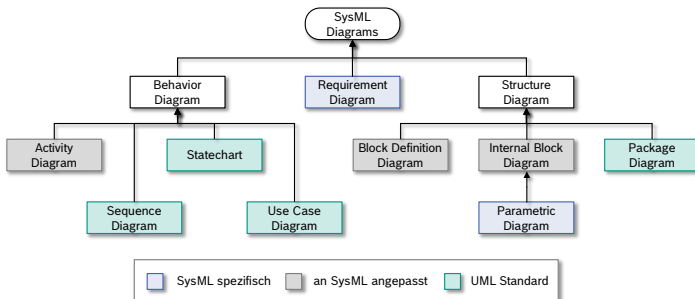


Abbildung 2.28: SysML-Diagrammtaxonomie nach Object Management Group (2007).

In Folge der zentralen Bedeutung von Strukturdarstellungen im SE finden, mit dem Block Definitions Diagramm (bdd) und dem Internal Block Diagram (ibd), zwei Strukturdiagramme die häufigste Anwendung in der SysML (Holt & Perry, 2014; Bone & Cloutier 2010). Das bdd dient vornehmlich der Modellierung von Hierarchien, ferner der Abbildung der vertikalen Strukturierung eines Systems (vgl. Kapitel 2.2.4; Weilkiens, 2007). Dazu zählt beispielsweise die Darstellung der funktionalen Dekomposition (vgl. Kapitel 2.2.4). Die Abbildung horizontaler Beziehungen zwischen Systemelementen - im Sinne des strukturalen Konzeptes nach Ropohl - erfolgt im ibd (Alt, 2012; Weilkiens, 2007). Das hierarchisch übergeordnete System ist im ibd durch die Diagrammgrenze repräsentiert. In Abbildung 2.29 wird der Einsatz beider Diagramme am Beispiel eines elektrischen Antriebs vorgestellt.

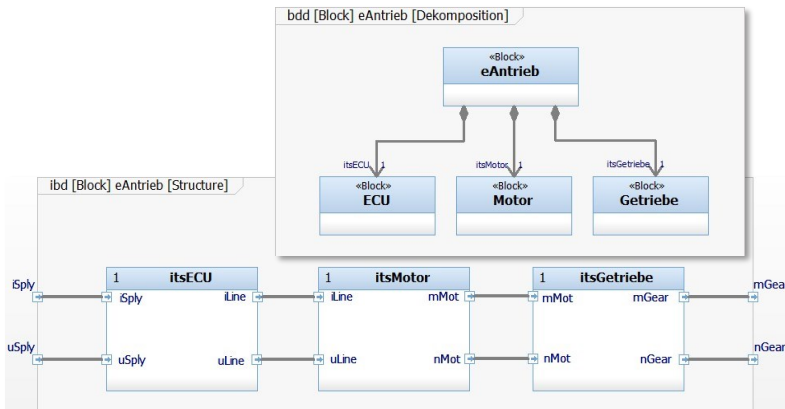


Abbildung 2.29: Nutzung von bdd und ibd am Beispiel des elektrischen Antriebs bestehend aus ECU, Motor und Getriebe (Darstellung in IBM Rhapsody).

Die aus der UML bekannten Struktur- und Verhaltensdiagramme wurden im Zuge der Definition der SysML durch einen neuen Diagrammtyp ergänzt: das Anforderungsdiagramm. Dieses dient vornehmlich der Modellierung der Elemente des Zielsystems hinsichtlich ihrer Wechselwirkungen mit der technischen Architektur. Die nachvollziehbare Dokumentation der Realisierung von Anforderungen durch die technische Umsetzung wird im Entwicklungsprozess mechatronischer Steuergeräte unter dem Begriff der Traceability explizit gefordert (ASPICE, 2017; ISO 26262, 2018). Dem Anforderungsdiagramm kommt dort folglich eine besonders wichtige Rolle zu.

Eine Sprache hat im Verständnis des Modellierungstriplets die Aufgabe, die zur Modellierung zulässigen Elemente und deren Beziehungen in Form einer Syntax vorzugeben. Ähnlich den Diagrammtypen orientieren sich die Modellelemente der SysML stark an den bestehenden Konzepten der UML. Eine detaillierte Übersicht der Gemeinsamkeiten und Unterschiede beider Sprachen findet sich u.a. bei Zingel (2013). Wie aus den Diagrammtypen des bdd und des ibd entnommen werden kann, spielt der Block in der SysML eine zentrale Rolle. Es handelt sich hierbei um eine Art Universalelement zur Modellierung vergleichbar mit der Rolle der Klasse in der objektorientierten Modellbildung der UML (Weilkiens, 2007). Der Block kann beispielsweise zur Darstellung von Funktionen, Produktkomponenten oder von generischen Systemelementen herangezogen werden. Die Eigenschaften und Merkmale des Blocks werden in Form sogenannter Attribute dokumentiert. Diese bestehen aus einem Attributnamen, Attributstyp und dem zugehörigen Attributwert. Zur Modellierung von Ein- und Ausgangsgrößen und deren Vernetzung bspw. in Form von Stoff-, Energie- oder Informationsflüssen zwischen Funktionen (Pahl, Beitz et al., 2007; Stone, 1997) übernimmt die SysML das Konzept der Ports aus der UML. Diese Ports dienen als Schnittstellen zwischen Blöcken und deren Umgebung und können je nach Bedarf entweder gerichtet oder ungerichtet implementiert werden. Eine Solche Vernetzung findet ausschließlich in ibds statt. Abbildung 2.30 zeigt die Zusammenhänge zwischen Blöcken, deren Attribute und Ports:

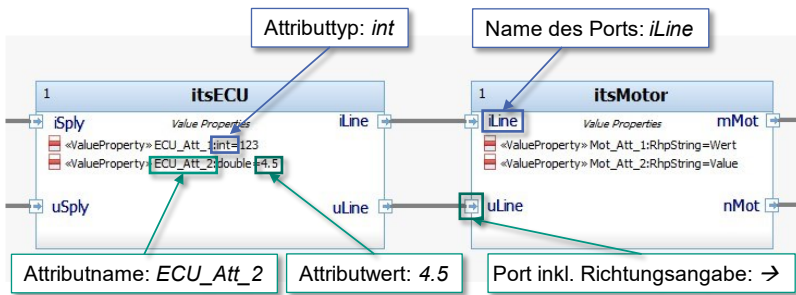


Abbildung 2.30: Attribute und Ports in SysML am Beispiel eines Systems bestehend aus ECU und Motor (Darstellung in IBM Rhapsody).

Um das Sprachprofil der SysML an den individuellen Einsatzzweck verschiedener Modelle anzupassen sind sogenannte Stereotypen vorgesehen. Mithilfe der Stereotypen können Blöcke beispielsweise konkret als Funktionen ausgezeichnet und auf die spezifischen Anforderungen der Funktionsmodellierung hin modifiziert

werden. Neben den Modellelementen existieren in der SysML verschiedene Beziehungstypen zu deren Verknüpfung. Eine detaillierte Übersicht in der SysML enthaltener Elemente und Beziehungen findet sich bei Weilkiens (2007), Zingel (2013) der SysML Spezifikation der Object Management Group (2007). Zur Abbildung von existenzabhängigen Hierarchien dient die Komposition. Diese kann als klassische „A ist-Teil-von B“ Beziehung interpretiert werden. Entfällt das hierarchisch übergeordnete Element, so werden die mittels Komposition verbundenen Elemente ebenfalls aus dem Modell entfernt. Verbindet man Elemente über Kompositionen wird im übergeordneten der Elemente ein sogenannter Part des untergeordneten Elements erstellt. Mit Hilfe der Parts kann die interne Struktur des übergeordneten Elements im Sinne des strukturalen Konzeptes nach Ropohl modelliert werden. Eine Auswahl spezifischer an die Systementwicklung angepasster Beziehungstypen etwa zur Sicherstellung der Traceability (Rückverfolgbarkeit) - wie in den internationalen Standards des ASPICE (2017) oder der ISO 26262 (2018) gefordert (vgl. Kapitel 2.2.3) - sind in der SysML ebenfalls enthalten. Beispielsweise kann die Zuordnung einer spezifischen Anforderung durch die sie erfüllende technische Umsetzung oder Funktion mittels einer **satisfy** Beziehung modelliert werden.

In der Theorie stellt die SysML einen bewusst abstrakt gehaltenen Sprachstandard für die Systemmodellierung nach dem MBSE Ansatz dar. Durch die individuelle Erweiterbarkeit über Profile und Stereotypen kann die abstrakte Sprache an die spezifischen Herausforderungen eines jeden Projektes angepasst werden. Wie im Modellierungs-Tripel (vgl. Kapitel 2.2.2) bereits anklingt ist eine erfolgreiche Modellbildung allerdings nur durch das Zusammenspiel aus Sprache, geeigneten Tools und allen voran, geeigneten, daher mensch-zentrierten Methoden zu gewährleisten. Erfolgreich ist hierbei gleichzusetzen mit akzeptiert und ins Tagesgeschäft der Produktentwickler integriert. Inwiefern der MBSE-Ansatz im industriellen Umfeld Anwendung und Zuspruch findet, wird im nächsten Abschnitt genauer beleuchtet.

### 2.2.7 MBSE in der Anwendung

Die Notwendigkeit eines Ansatzes, zum strukturierten Umgang mit der Komplexität mechatronischer Systeme, ist bereits seit geraumer Zeit, ausgehend von der Industrie ins Bewusstsein unserer Gesellschaft gelangt (INCOSE, 2007). Seither hat sich das Bild des Systems Engineering im Zuge globaler Megatrends: Digitalisierung, Globalisierung und Nachhaltigkeit, stetig weiterentwickelt. Im zentralen Fokus steht der Wandel der mechatronischen Systeme hin zu intelligenten, hochvernetzten, sogenannten Advanced Systems (Dumitrescu et al., 2021). Unter dem Leitbild des Advanced Systems Engineering werden die Ansätze

des MBSE nicht nur als mögliche Lösungen betrachtet, sondern bewusst ins Zentrum der Maßnahmen gerückt (Dumitrescu et al., 2021). Bereits in der Vergangenheit belegten zahlreiche Studien, dass die traditionellen, dokumentenbasierten Ansätze keine probaten Lösungen zur Problembewältigung in komplexen Entwicklungsprozessen darstellen (Saunders, 2011; Delegatti, 2014; Hossain & Jaradat, 2018). Als Hauptgrund wird der Aufwand zur Konsistenzwahrung im Falle verteilter abliegender Informationen genannt. Dieser wird hinsichtlich des Einpflegens von Änderungen mit zunehmender Produktkomplexität immer schwerer beherrschbar (Alt, 2012). An dieser Stelle sehen die Unternehmen ein großes Potential des MBSE (Saunders, 2011; Hossain & Jaradat, 2018; Dumitrescu et al., 2021; Gausemeier, Anacker et al., 2013; Bone & Cloutier, 2010; Albers, Lohmeyer & Radimersky, 2012; Albers & Zingel 2013a). Mittels durchgängiger Systemmodelle bleibt die Konsistenz im Entwicklungsprozess erhalten und ein strukturierter Umgang mit der Produktkomplexität wird ermöglicht. Gerade im Bereich mechatronischer Produkte, liegt eine nach Weber (2005) (vgl. auch Luhmann, 1991; Reichwein & Paredis, 2011) über die Anzahl von Elementen und Beziehungen definierte, sehr hohe Komplexität vor (Gausemeier, Anacker et al., 2013). Ein Einsatz von Systems Engineering auf Basis durchgängiger, interdisziplinärer Systemmodelle scheint auf diesem Gebiet nahezu alternativlos zu sein.

Wie bereits in Kapitel 2.2.6 diskutiert wurde, ist die bloße Festlegung einer standardisierten Sprache zur erfolgreichen Darstellung interdisziplinärer Modelle keineswegs ausreichend. Ohne geeignete Modellierungsmethoden und -tools kann der MBSE Ansatz nicht zielführend umgesetzt werden (Matthiesen et al., 2014). Im Bereich der Systemmodellierung wurden in den vergangenen Jahren zahlreiche **Methoden** geschaffen. Diese dienen der Unterstützung der Systementwickler bei der ganzheitlichen Modellierung komplexer Systeme. Als Beispiele sind der SysMOD-Ansatz (Weilkiens, 2016) - inklusive der Erweiterungen FAS (Lamm & Weilkiens, 2010; Lamm & Weilkiens, 2014; Lamm & Weilkiens, 2015) und FAS4M (Moeser et al., 2016) – der OOSEM Ansatz von Lykins et al. (2000) der RFLP-Ansatz nach Kleiner & Kramer (2013) und der SPES-Ansatz nach Pohl et al. (2012) zu erwähnen. Die genannten Ansätze zielen auf eine durchgängige und rückführbare Modellierung von Anforderungsspezifikationen und Systemarchitekturen mit Fokus auf der funktionsorientierten Entwicklung ab. Durch den zunehmenden Einfluss von KI und globalen Datennetzwerken nimmt die Bedeutung informationstechnischer Darstellungen in der Systemmodellierung zu (Dumitrescu et al. 2021) Dabei wird die Harmonisierung der domänenspezifischen Sichten in einem interdisziplinären Systemmodell zu einer zunehmenden Herausforderung der aktuellen Forschung. Erste Ergebnisse zeigt der V-SUM Ansatz von Klare et al. (2021).

Neben der steigenden Anzahl an Methoden hat sich im Zuge des MBSE eine breite **Toollandschaft** ausgebildet (Bone & Cloutier, 2010). Die toolgestützte Systemmodellierung ist dabei zunehmend an die Nutzung der **SysML** gebunden. Die am häufigsten genutzten Vertreter der SysML-basierenden Tools sind mit großem Abstand IBM Rational Rhapsody und NoMagic MagicDraw (Bone & Cloutier, 2010). In den Bereichen der domänenspezifischen Entwicklung ist weniger die Sprache als der benötigte Funktionsumfang des Tools auswahlrelevant. Der Prämisse zur Erstellung durchgängiger Modelle folgend existieren vermehrt Ansätze zur Toolintegration. Über den Einsatz sogenannter API's (Application Programming Interfaces) bieten die Toolhersteller eine Möglichkeit zur Integration ihres Tools im Rahmen eines föderierten Ansatzes.

Trotz der großen Potentiale und der immer wichtigeren Rolle des MBSE in unserer Gesellschaft haben die Ansätze zur durchgängigen Systemmodellierung noch immer keine flächendeckende Durchdringung im industriellen Umfeld erreicht (Tazir, 2011; Gausemeier, Anacker et al., 2013; Albers, Lohmeyer & Radimersky, 2012; Bone Cloutier, 2010; Delligatti, 2014; Dumitrescu et al., 2021). Lediglich im forschungsnahen Umfeld und im Rahmen einzelner Pilotprojekte kann eine dauerhafte Nutzung registriert werden (Bone & Cloutier, 2010; Dumitrescu et al., 2021). Seit der Einführung des MBSE haben es sich daher zahlreiche Studien zum Ziel gemacht, Gründe für die fehlende Akzeptanz und Anwendung zu erforschen. Regelmäßig wird dort das Fehlen geeigneter Methoden und die unzureichende Toolunterstützung als einer der Hauptgründe ausgemacht (Albers & Zingel, 2013a; Gausemeier, Anacker et al., 2013). Die Einzigartigkeit und Individualität eines jeden Entwicklungsprozesses (Albers, 2010) hat zur Folge, dass die generischen Methoden zur Systemmodellierung (vgl. vorheriger Abschnitt) den Projektspezifika angepasst werden müssen. Eine Übertragbarkeit auf andere Branchen, Unternehmen oder Produktentwicklungsprozesse ist daher mit sehr hohen Aufwänden verbunden (Scherer, 2016). Des Weiteren kristallisiert sich der hohe Initialaufwand zur Systemmodellierung gegenüber ihrem nur schwerlich quantifizierbaren, daher oft subjektiv empfundenen Nutzen im Entwicklungsprozess, als große Hürde für die Akzeptanz des MBSE heraus (Powelske et al., 2018; Delligatti, 2014). Bone & Cloutier (2010) identifizieren an dieser Stelle die starke Verankerung traditioneller Arbeitsweisen in der Unternehmenskultur als bedeutendes Hindernis. Beim Übergang von der dokumentenzentrierten Arbeitsweise zur durchgängigen Systemmodellierung sind viele der späteren Systemelemente bereits in Dokumenten vorhanden. Diese werden zur Übertragung häufig erneut und damit redundant erzeugt. Powelske et al. (2018) sehen große Potentiale in einem automatisierten Informationstransfer zur drastischen Senkung des initialen Modellierungsaufwands. Die Arbeiten von Götz & Donges (2016) sowie von Müller et al. (2017) zeigen bereits wie mittels einer bewussten Nutzung von

Referenzmodellen eine Wiederverwendung bestehender Informationen gelingen und die Aufwände bei der Modellierung reduziert werden können. Weiteres Hindernis für die flächendeckende Einführung des MBSE stellt die geringe Erfahrung der Systementwickler und der Unternehmen mit den Inhalten des MBSE dar (INCOSE, 2011; Reichwein & Paredis, 2011). Der subjektive Nutzen des MBSE für die Tätigkeiten beim Systementwurf wird daher nicht in voller Gänze erfahren. Beides wirkt sich negativ auf die organisatorische als auch die individuelle Akzeptanz des Ansatzes aus (Scherer, 2016). Viele Unternehmen sehen die Aufgabe interne Standards für Modelle und MBSE-gestützte Entwicklungsprozesse als besonders herausfordernd an (INCOSE, 2011). Durch die starke Kopplung der MBSE Methoden an die Nutzung der SysML, spielt diese in Bezug auf die Akzeptanz des gesamten Ansatzes ebenfalls eine große Rolle. Hierbei wird der Sprache SysML ein großes Potential nachgesagt; eine Durchdringung der industriellen Prozesse hat bisher allerdings nur stellenweise stattgefunden (Zingel, 2013). Gründe dafür liegen weniger an der Sprache an sich, sondern an der wenig anwenderfreundlichen Umsetzung innerhalb der Tools und der fehlenden Schnittstellen zu anderen Produktmodellen (Albers & Zingel 2013a; Albers & Lohmeyer, 2012; Zingel, 2013; Bone & Cloutier 2010). Die **Akzeptanz des MBSE Ansatzes** lässt sich resümierend am Verhältnis des voraussichtlichen Nutzens zu den tatsächlichen Aufwänden zu Umsetzung festmachen. Während der Nutzen fallspezifisch nur subjektiv ermittelt werden kann, zeigen sich große Potentiale in der Reduktion des Aufwandes zur initialen Modellerstellung. Die Methodenentwicklung sollte sich daher vermehrt der Aufwandsvermeidung durch Nutzung von Referenzsystemen (vgl. Kapitel 2.3.1) und deren nutzerfreundliche Anpassung an branchen- und projektspezifische Problemstellungen widmen. An dieser Stelle kann der gezielte Einsatz der im Rahmen des ASE als Potentialträger identifizierten KI (Dumitrescu et al., 2021) zum Schlüsselfaktor werden.

## 2.2.8 Fazit

Model Based Systems Engineering (MBSE) beschreibt einen menschenzentrierten Ansatz, der zur Erzeugung eines umfassenden Systemverständnisses, konsistente, vernetzte Systemmodelle ins Zentrum der zunehmend interdisziplinären Produktentwicklung rückt. Im Umfeld mechatronischer Produkte ermöglicht der Einsatz dieser Modelle die allgemeinverständliche Abbildung komplexer Wechselwirkungen, was eine massive Unterstützung des ganzheitlichen Denkprozesses darstellt. Als universeller Sprachstandard für die Modellierung im interdisziplinären Umfeld hat sich dabei die SysML etabliert. Der Trend der Hochintegration und die hohen Stückzahlen in der Entwicklung mechatronischer Steuergeräte führen zu Mischformen modularer und integraler Produktstrukturanteile. Für die Entwicklung neuer Standardisierungsmethoden

erscheint es daher sinnvoll zunächst eine Vergleichbarkeit dieser Strukturen zu schaffen. Trotz der großen Potentiale für den branchenübergreifenden Einsatz des MBSE Ansatzes ist dessen Etablierung im industriellen Umfeld weiterhin nicht flächendeckend. Aktuelle Studien unter dem Leitbild des ASE zeigen jedoch deutlich, dass die Bedarfe für den Wandel des Engineerings erkannt sind und globale Lösungen in den Bereichen Forschung, Lehre und Wirtschaft angestrebt werden. In der Entwicklung neuer Methoden empfiehlt es sich nach wie vor, diese explizit auf branchen- und projektspezifische Bedarfe und die Minimierung des Modellierungsaufwandes auszurichten. Eine (Teil-) Automatisierung von Modellierungsschritten bspw. beim Übertrag von Informationen aus bestehenden Dokumenten scheint hierbei vielversprechend.

## **2.3 PGE – Modell der Produktgenerationsentwicklung nach Albers**

Der folgende Abschnitt widmet sich der Einführung des Modells der PGE – Produktgenerationsentwicklung (Albers, Bursac & Wintergerst, 2015). Dazu wird eingangs, ein grundlegender Überblick des Modells und seiner zentralen Hypothesen gegeben. Im Anschluss wird, auf Basis des systemtheoretischen Verständnisses des vorangegangenen Kapitels, die Erweiterung des Modells um das Element des Referenzsystems vorgestellt (Albers, Rapp, et al., 2019). Hiernach wird mit der Frühen Phase der PGE derjenige Abschnitt des Produktentstehungsprozesses definiert, der sich durch ein besonders hohes Maß an Unsicherheit auszeichnet. Abschließend findet eine Einordnung der Frühen Phase der PGE in die allgemeinen Aktivitäten der Forschung und Entwicklung (F&E) statt. Dies dient der späteren Klassifizierung von Prozessen innerhalb der zugrundeliegenden Untersuchungsumgebung.

### **2.3.1 Das Modell der PGE – Produktgenerationsentwicklung nach Albers**

Das Modell der PGE dient der Modellierung von Beobachtungen im Entwicklungsprozess neuer technischer Systeme. Es wurde von Albers mit dem Zweck entwickelt die beobachteten Phänomene der entwicklungsmethodischen Forschung zugänglich zu machen (Albers, Bursac & Wintergerst, 2015). Das Modell berücksichtigt im Gegensatz zu den Theorien der klassischen Konstruktionsmethodik (Pahl, Beitz et al., 2007) und des Innovationsmanagements (Specht et al., 2002; Henderson & Clark, 1990) den Einfluss von Referenzlösungen. Es trägt damit der Beobachtung, dass Produkte niemals komplette



Neuentwicklungen darstellen (Eckert et al., 2010)<sup>31</sup> Rechnung und ermöglicht eine eindeutige Zuordnung von Entwicklungsanteilen zu den Teilsystemen der neuen Produktgeneration (Albers, Bursac & Wintergerst, 2015) in realen Produktentwicklungsprojekten. Im Kern des Modells stehen zwei grundlegende Hypothesen (Albers, Bursac & Wintergerst 2015). Der **ersten Hypothese** folgend, basiert jede Produktentwicklung immer auf einem oder mehreren Referenzprodukten. Bei diesen handelt es sich beispielsweise um Wettbewerberprodukte, um Produkte aus Vorgängergenerationen, oder um branchenfremde Produkte. Die Entwicklung eines neuen Produktes ist immer abbildbar bzw. beschreibbar durch die Überführung von Elementen eines Referenzsystems in eine neue Produktgeneration. Das Referenzsystem inklusive seiner Elemente ist dabei der Kern und wird zusammengestellt aus Teilsystemen von bestehenden Produkten oder Artefakten der Forschung. Die **zweite Hypothese** konkretisiert die Art und Weise wie die Teilsysteme der Referenzprodukte zur Synthese des neuen Produktes herangezogen werden. Dem Modell der PGE folgend, beruht die Entwicklung neuer Produktgenerationen ausschließlich auf der Kombination dreier Variationsarten. Die Übernahme eines Teilsystems mit lediglich geringen Anpassungen von Schnittstellen wird als Übernahmeveriation (ÜV) bezeichnet. Die Neuentwicklung von Teilsystemen des neuen Produktes teilt sich in Prinzip- (PV) und Ausprägungsvariation (AV) (ehemals Gestaltvariation (GV) nach Albers, Bursac & Wintergerst (2015)) auf (Albers et al., 2020). Wobei die Ausprägungsvariation einer generalisierten Auffassung entspricht, die z.B. auch in Funktions- oder Eigenschaftssicht eine zutreffende Beschreibung ist. Für die Ausprägung physischer Teile mechatronischer Systeme, kann nach wie vor der Terminus Gestaltvariation verwendet werden. Die Variationsanteile dienen vornehmlich der Differenzierung gegenüber der Wettbewerberprodukte und der Vorgängergenerationen. In Anlehnung an das Kano-Modell (Kano et al., 1984) bedarf es über die Zeit immer wieder neuer Begeisterungsanforderungen, um eine ausreichende Differenzierung zu gestalten und den Produkterfolg zu sichern (Albers, 1994; Bailom et al., 1996). Zur Erklärung der Variationsanteile soll im Folgenden ein Beispiel aus der Leistungselektronik mechatronischer Steuergeräte dienen (Abbildung 2.31):

Wird zur Motoransteuerung statt einer H-Brückenschaltung (bspw. der Vorgängergeneration) eine B6-Brückenschaltung eingesetzt, spricht man von einer Prinzipvariation (Albers et al., 2020). Dies ist auf den Umstand zurückzuführen, dass

---

<sup>31</sup> entsprechende Beobachtungen auf systemischer Ebene existieren auch auf dem Gebiet des Design Reuse (Wyatt et al., 2009)

beide Schaltungen ein anderes Lösungsprinzip zur Bereitstellung von Strömen und Spannungen an die einzelnen Motorphasen nutzen. Wählt man hingegen innerhalb einer B6-Brückenschaltung leistungstärkere MOSFETs aus, so behält man das gleiche Lösungsprinzip bei, verändert jedoch einzelne Attribute bzw. Gestaltparameter. Eine solche Ersetzung bezeichnet man dementsprechend als Ausprägungs- bzw. Gestaltvariation.

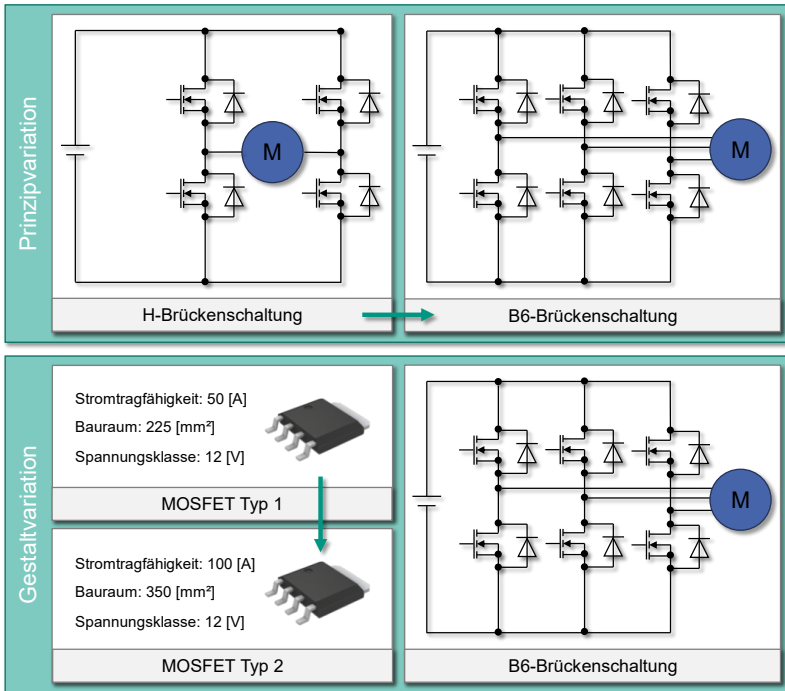


Abbildung 2.31: Darstellung der Variationsanteile Prinzipvariation und Gestaltvariation am Beispiel der Endstufengestaltung. Eigene Darstellung entsprechend den Grundlagen von Albers et al. (2020)

Gemessen an der Gesamtzahl von Teilsystemen einer Produktgeneration lassen sich die prozentualen Anteile auf Basis von Referenzprodukten angepasster oder neuentwickelter Teilsysteme errechnen. Zur Beschreibung dieser sogenannten Variationsanteile kann folgendes mathematisches Modell herangezogen werden:

Eine neue, gerade in der Entwicklung befindliche Produktgeneration ( $G_n$ ) stellt die Vereinigungsmenge der Teilsysteme dar, welche auf der Übernahmevariation ( $\ddot{U}S_n$ ), Gestaltvariation ( $GS_n$ ) und Prinzipvariation ( $PS_n$ ) der Teilsysteme eines der Referenzprodukte beruhen (Albers, Bursac & Wintergerst, 2015).

$$G_n = \ddot{U}S_n \cup GS_n \cup PS_n \quad (1)$$

Die jeweiligen Variationsanteile ergeben sich demnach zu:

$$\delta_{\ddot{U}S_n} = \frac{|\ddot{U}S_n|}{|G_n|} = \frac{|\ddot{U}S_n|}{|\ddot{U}S_n \cup GS_n \cup PS_n|} [\%] \quad (2)$$

$$\delta_{GS_n} = \frac{|GS_n|}{|G_n|} = \frac{|GS_n|}{|\ddot{U}S_n \cup GS_n \cup PS_n|} [\%] \quad (3)$$

$$\delta_{PS_n} = \frac{|PS_n|}{|G_n|} = \frac{|PS_n|}{|\ddot{U}S_n \cup GS_n \cup PS_n|} [\%] \quad (4)$$

Auf Basis der Beschreibung existierender Produktgenerationen und deren Variationsanteile, können die Erkenntnisse der PGE bewusst zur Planung und Steuerung von Produktentwicklungsprozessen eingesetzt werden (Albers, Haug et al, 2016). In der Praxis herrscht das Bestreben die Aufwände und Risiken bei der Entwicklung zukünftiger Produktgenerationen sinnvoll zu beschränken (Albers, Haug et al, 2016; Deubzer & Lindemann, 2009). Albers et al. liefern ein Instrument, um die zu erwartenden Entwicklungsrisiken bereits in einer Frühen Phase abzuschätzen. Hierzu wird die Herkunft der berücksichtigten Referenzprodukte dem zu erwartenden Entwicklungsaufwand, in Form der Neuentwicklungsanteile ( $\delta_{GS_n} + \delta_{PS_n}$ ) gegenübergestellt (Albers, Haug et al., 2016; Albers, Bursac & Rapp 2016; Albers, Rapp, Birk & Bursac, 2017; Albers, Rapp et al., 2018). Als Resultat der Einordnung ergibt sich eine Tendenz des zu erwartenden Risikos in der Entwicklung der geplanten Produktgeneration. Abbildung 2.32 zeigt ein sogenanntes Risikoportfolio für die Entwicklung einer neuen Produktgeneration des Zweimassenschwungrads (ZMS) auf Basis des Neuentwicklungsanteils und der Herkunft des Referenzproduktes.

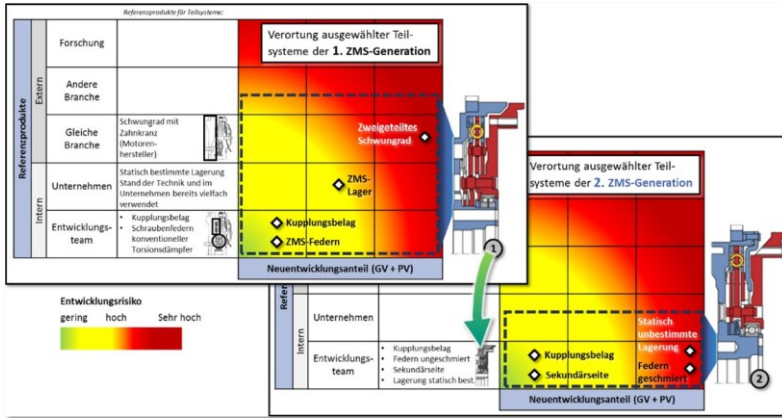


Abbildung 2.32: Risikobewertung auf Basis von Neuentwicklungsanteil (GV + PV) und Herkunft des Referenzprodukts am Beispiel des ZMS nach Albers, Rapp et al. (2017).

Wird die Erhöhung des Übernahmevariationsanteils systematisch durchgeführt, kann damit eine Verringerung der unternehmensinternen Variantenvielfalt verwendeter Teilsysteme einhergehen (vgl. Kapitel 2.4). Auf diesem Gedanken begründen sich Standardisierungsmethoden wie beispielsweise die Baukastenentwicklung. Diverse Forschungsarbeiten am Institut für Produktentwicklung Karlsruhe (IPEK) haben bereits gezeigt, wie die Erkenntnisse des Modells der PGE zur Unterstützung der Baukastenentwicklung verwendet werden können (Bursac, 2016; Scherer, 2016; Scherer, Albers & Bursac, 2017). Weiterhin sind unter Verwendung des Modells der PGE umfassende Methoden und Prozesse zur Unterstützung verschiedenster Aspekte der Produktentwicklung entstanden. Albers et al. nennen hierfür beispielhaft die Bildung initialer Zielsysteme, die Ableitung zielführender Variationen, Methoden der Strategiebildung, die Wirtschaftlichkeitseinschätzung von Produktvarianten, sowie den Umgang mit komplexen Systems-of-Systems, die im Zuge der Advanced Systems immer relevanter werden (Albers, Rapp et al., 2019 dort in Bezug auf Arslan, Bursac et al., 2018; Albers, Heitger et al., 2018; Rapp, Moeser et al., 2018; Albers, Walch & Bursac, 2016; Peglow et al., 2017; Albers, Peglow, Powelske, Birk & Bursac, 2018). Die Entwicklung der Methoden beruhte dabei stets auf dem Kern des Modells der PGE: der Berücksichtigung geeigneter Referenzprodukte. Aktuelle Beobachtungen zeigen, dass neben Teilsystemen realer, am Markt verfügbarer Referenzprodukte auch andere Elemente in die Entwicklung einer neuen

Produktgeneration einfließen können. Diese lassen sich auf Basis der Erkenntnisse der Systemtheorie am treffendsten als existierende technische Systeme klassifizieren. Dazu zählen beispielsweise Konzepte unterschiedlicher Detaillierungsgrade auf Ebene der Funktions- als auch der Produktstruktur, welche im Entwicklungsprozess anfallen, jedoch nicht zwingend zu einem am Markt realisierten Produkt führen (Albers, Rapp, et al., 2019). Des Weiteren kann mittels des Systembegriffes die Verwendung einer branchenfremden Technologie als Referenz zur Entwicklung einer neuen Produktgeneration erklärt werden (Albers, Rapp, et al., 2019). Eine Erweiterung des Modells der PGE hinsichtlich eines Elementes, welches sämtliche Objekte umfasst, die die Grundlage der Entwicklung einer Produktgeneration bilden, erscheint vor diesem Hintergrund sinnvoll. Die Überlegungen wurden mit der Einführung des Referenzsystems manifestiert. Albers et al. (2019) formulieren hierfür folgende Definition:

**Definition 14: Referenzsystem**

Das Referenzsystem für die Entwicklung einer neuen Produktgeneration ist ein System, dessen Elemente aus bereits bestehenden oder bereits geplanten sozio-technischen Systemen und der zugehörigen Dokumentation stammen und die Basis und den Ausgangspunkt für die Entwicklung der neuen Produktgeneration bilden. (übersetzt nach Albers, Rapp, et al., 2019)

Zu jeder Produktgeneration existiert folglich genau ein Referenzsystem. Dieses muss im Laufe des Entwicklungsprozesses kontinuierlich angepasst werden. Die Elemente des Referenzsystems werden wiederum als dessen Teilsysteme bezeichnet. Das Referenzsystem einer Produktgeneration kann – entsprechend dem Modell der PGE – in Bezug auf seine Teilsysteme, Schnittmengen zu den Referenzsystemen anderer Produktgenerationen aufweisen. Auf dieser Basis weiten Albers, Kürten, Rapp, Birk, Hünemeyer & Kempf (2022) die Produktgenerationsentwicklung auf die Beschreibung der Entwicklung von Systemen im Allgemeinen zum sogenannten Modell der SGE – Systemgenerationsentwicklung aus.

Mit der Erweiterung der PGE um den Begriff des Referenzsystems bzw. der Ausweitung auf die SGE – Systemgenerationsentwicklung bietet sich das Modell ebenfalls zur Beschreibung der Aktivitäten in Zusammenhang mit dem Systems Engineering an. Die Forschung bezüglich des Referenzsystems und seiner Bedeutung für die Aktivitäten der Produktentstehung steht allerdings noch am Anfang. Die Eignung zur umfassenderen Betrachtung der Einflüsse von Variationsanteilen für den gesamten Produktentstehungsprozess ist bereits anhand diverser Fallstudien erkennbar (Albers, Rapp et al., 2019). Das Referenzsystem

bietet die Möglichkeit die Einflüsse von Produkt, Produktgeneration, Unternehmensstrategie, Produktions- und Validierungssystem auf die Variationsanteile einer neuen Produktgeneration detailliert zu beschreiben. Beispielsweise kann untersucht werden, inwiefern existierende oder fehlende Elemente des Validierungssystems die Entscheidung bezüglich Übernahme oder Neuentwicklung von Teilsystemen einer Produktgeneration beeinflussen. Im Rahmen der mechatronischen Steuergeräteentwicklung bietet das Modell der PGE die Möglichkeit die Übernahme von Teilen der Funktions- oder der Produktstruktur zur Sicherstellung der durchgängigen Rückverfolgbarkeit in der Entwicklung neuer Produktgenerationen zu beschreiben. Die Potentiale zur Zeitersparnis im Entwicklungsprozess erweisen sich hierbei mitunter als enorm (Powelske et al., 2018). Die Tatsache, dass neben Teilsystemen aus Referenzprodukten auch Erfahrungswissen abgelaufener Studien oder durch das bisherige Modell nicht erfasster, konzeptueller Lösungsbeschreibungen, also Elemente des Referenzsystems, aktiv bei der Entwicklung berücksichtigt werden, birgt große Potentiale für die Systementwicklung (Powelske et al., 2017)<sup>32</sup>. Auf dieser Basis ergibt sich als Ziel weiterer Forschungsarbeiten, die genannten Erkenntnisse über Referenzsysteme für die Planung und Entwicklung neuer Produktgenerationen systematisch nutzbar zu machen. Dazu wird unter anderem untersucht wie mittels des Einsatzes von Szenarien Referenzsysteme zukünftiger Produktgenerationen konkretisiert werden können und somit die strategische Produktplanung unterstützt werden kann. Zu erwartende Ergebnisse im Bereich der Zukunftsvorausschau sind kurz-, mittel- und langfristige Handlungsempfehlungen der PGE (Albers, Dumitrescu et al. 2018; Marthaler et al., 2019).

### **2.3.2 Frühe Phase der PGE**

Die Frühe Phase der Produktentwicklung ist nicht nur einer hohen Unsicherheit (Verworn, 2005) unterworfen, sondern beherbergt zusätzlich das von Grabowski & Geiger (1997) spezifizierte Paradoxon der Konstruktion. Dieses besagt, dass in der Frühen Phase des Entwicklungsprozesses zwar großer Freiraum bezüglich Veränderungen herrscht, die Auswirkungen getroffener Entscheidungen jedoch kaum bekannt sind. Dem gegenüber stehen im späteren Verlauf der Produktentwicklung Phasen, in denen die Auswirkungen beurteilt werden können, die Möglichkeiten zur Veränderung allerdings limitiert sind. Die zentrale Bedeutung der Entscheidungen innerhalb der Frühen Phase für den Produkterfolg und deren

---

<sup>32</sup> hinsichtlich der Baukastenentwicklung mechatronischer Steuergeräte

tiefgreifenden Auswirkungen auf gesamten Produktlebenszyklus, wird durch zahlreiche Untersuchungen untermauert (Cooper & Kleinschmidt, 1993). Meerkamm (1995) quantifiziert den Einfluss, indem er die Festlegung von zwei Dritteln aller späteren Produkteigenschaften in der Frühen Phase verortet. Zusätzlich dient die Beeinflussung von Kosten und deren tatsächliche Entstehung entlang des Produktentstehungsprozesses als Indikator des Einflusses der Frühen Phase auf alle nachfolgenden Aktivitäten (Abbildung 2.33).

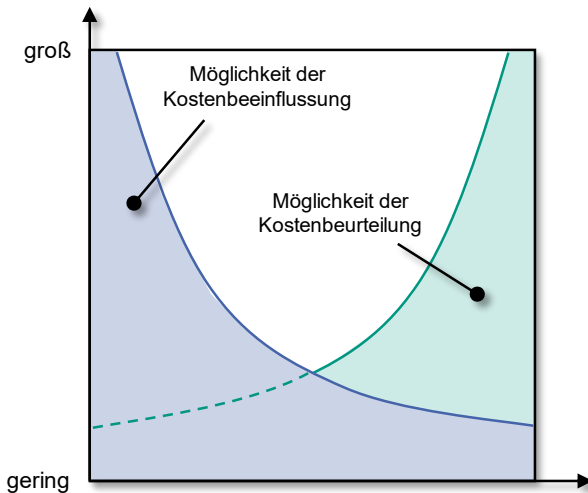


Abbildung 2.33: Möglichkeit von Kostenbeeinflussung und -beurteilung im Produktentwicklungsprozess nach Ehrlenspiel & Meerkamm (2013).

Die Ausrichtung der Entwicklungstätigkeiten späterer Entwicklungsphasen gründet auf der Bildung des initialen Zielsystems in der Frühen Phase (Heitger, 2019). Muschik (2011) identifiziert die Bildung und damit das Wesen jenes Zielsystems, aufgrund der Schwierigkeiten bei der Anforderungsableitung, in der Frühen Phase als besonders vage. Albers et al. schließen, dass die vage Beschaffenheit ihres initialen Zielsystems, der Frühen Phase der Produktentwicklung einen gering strukturierten Charakter verleihen. Hierbei beziehen sie sich auf eine Liste von Merkmalen, welche Verwon (2005) neben der Tragweite gefällter Entscheidungen zur Bewertung der Frühen Phase heranziehen (Albers, Rapp et al, 2017). Die geringe Strukturierung und das vage Zielsystem erweisen sich als Folgen der,

eingangs erwähnten Unsicherheit. Diese konkretisiert Galbraith (1973) als die Differenz zwischen benötigten und vorhandenen Informationen. Das Fehlen einer ausreichenden Datenbasis kann weiterhin auf das Auftreten einer Wissenslücke oder einer Definitionslücke zurückgeführt werden (McManus & Hastings, 2005). Da beide Phänomene in Bezug auf deren Auflösung eng miteinander verknüpft sind, schließt Albers auf folgendes Dilemma: Die Generierung zielgerichteten Wissens bedarf vorhandener Definitionen, während das Schließen von Definitionslücken nur, mittels problemspezifischen Wissens gelingen kann (Albers, Lohmeyer & Ebel, 2011). Die zeitliche Einordnung der Frühen Phase in den Produktentwicklungsprozess erweist sich vor diesem Hintergrund als schwierig (Heitger, 2019). Gängige Definitionen stehen häufig in Bezug zum Ansatz der klassischen Konstruktionsmethodik nach Pahl, Beitz et al. (2007), speziell zum Kontext der Neukonstruktion. Im Fokus der Tätigkeiten der Frühen Phase stehen die Ideengenerierung und deren Bewertung (Khurana & Rosenthal, 1997). Definitionen aus dem englischen Sprachraum nutzen daher den Begriff des „front end of innovation“. Die Ergebnisse der Frühen Phase in Form einer positiven Entscheidung für ein Entwicklungsprojekt stellen den Übergang zum klassischen, strukturierten Entwicklungsprozess dar (Koen et al. 2001). Für eine detaillierte Gegenüberstellung verschiedener Definitionen der Frühen Phase und deren Vergleich sei an dieser Stelle auf Bursac (2016) verwiesen. Folgende Darstellung (Tabelle 2.1) dient der Explikation differenzierender Aspekte verschiedener Definitionen:

**Tabelle 2.1: Übersicht bestehender Definitionen der frühen Phase nach Bursac (2016, S.44)**

Autor	Spezifikum	Visualisierung
Khurana & Rosenthal (1997)	Finanzierung und Start der Produktentwicklung auf Basis einer „Go / No-Go“ Entscheidung	
Koen et al. (2001)	Alle Aktivitäten vor dem Beginn des formalen und strukturierten Produktentwicklungsprozesses	
Jetter (2005)	Brücke zwischen der strategischen Produktplanung und der Produktentwicklung	
Muschik (2011)	Zeitraum von der Initiierung eines Projektes bis zur Formulierung einer ersten Produktspezifikation	



Mittels der Elemente des Modells der PGE und unter Einbezug einer frühen Validierung im Sinne des erweiterten ZHO-Modells kann der Konkretisierungsprozess des Zielsystems bereits früh erfolgen. Dies resultiert in einer Reduktion der Unsicherheit was letztlich den Weg dazu ebnet das gleichnamige Dilemma zu überkommen. Bursac (2016) folgert weiterhin, dass mittels des Ansatzes der PGE der zweiten großen Herausforderung in der Frühen Phase begegnet werden kann – dem Paradox der Konstruktion nach Grabowski & Geiger (1997). Albers kondensiert diese und weitere Überlegungen zur folgenden Definition der Frühen Phase im Modell der PGE:

#### **Definition 15: Frühe Phase der PGE – Produktgenerationsentwicklung**

Die „Frühe Phase“ der Produktgenerationsentwicklung ist eine Phase im Entwicklungsprozess einer neuen Produktgeneration, die mit der Initiierung eines Projektes beginnt und mit einer bewerteten technischen Lösung endet, die schließlich das initiale Zielsystem hinsichtlich seiner wesentlichen Elemente abdeckt. Die zur technischen Lösung gehörende Produktspezifikation als Teil des Zielsystems enthält u.a. Informationen bzgl. der verwendeten Technologien und Subsysteme sowie deren Übernahme- und Neuentwicklungsanteile. Sie ermöglicht eine valide Bewertung des zu entwickelnden technischen Systems hinsichtlich der relevanten Parameter wie beispielsweise der Produzierbarkeit, der notwendigen Ressourcen oder des technischen und ökonomischen Risikos. (Albers, Rapp et al., 2017)

Die bewertete technische Lösung und damit das Ergebnis der Frühen Phase der PGE, wird im Falle der zugrundeliegenden Untersuchungsumgebung für mechatronische Steuergeräte, (vgl. Kapitel 3.3) als Funktionsmuster bezeichnet. Dieses dient dem Test einzelner Teilfunktionen im Sinne der frühen Validierung und damit der Konkretisierung des initialen Zielsystems. Zur Bewertung des zu entwickelnden technischen Systems hinsichtlich technischer und ökonomischer Risiken können die Erkenntnisse nach Albers, Rapp et al. (2018) herangezogen werden. Als mögliche Maßnahme zur Senkung der Unsicherheit und den damit verbundenen Entwicklungsrisiken der Frühen Phase können über Forschungs- und Vorentwicklungsprojekte gezielt Erkenntnisse für künftige Produktgenerationen aufgebaut werden (Albers, Rapp et al., 2017 dort in Bezug auf Heismann & Maul, 2012). Specht et al. (2002) konkretisieren das Vorhaben derartiger Forschungs- und Vorentwicklungsprojekte als Transformation von Theorien in Technologien und deren anschließende Anwendung in technischen Systemen (Abbildung 2.34).

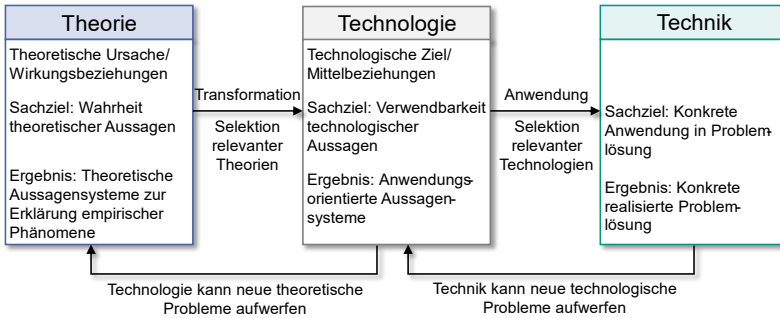


Abbildung 2.34: Zusammenhang zwischen Theorie, Technologie und Technik nach Specht et al. (2002).

Im Transformations- und Anwendungsprozess gewonnene Erkenntnisse lassen sich verschiedenen Aktivitäten der Forschung und Entwicklung (Specht et al., 2002) zuordnen und als Elemente des Referenzsystems für neue Produktgenerationen nutzbar machen (vgl. Kapitel 2.3.1). Im Rahmen der zeitlichen Betrachtung von Vorentwicklungsprojekten zeigen Albers, Rapp et al. (2017), dass diese Projekte ebenfalls Abschnitte aufweisen können, die die Merkmale einer Frühen Phase (Verwon, 2005) erfüllen (Abbildung 2.35). Daneben existieren jedoch zusätzliche Phasen, die die angesprochenen Kriterien (hier in Bezug auf die „Frühe Phase“) nicht erfüllen und trotzdem vor der Frühen Phase der Entwicklung einer Produktgeneration ablaufen (Albers, Rapp et al., 2017).

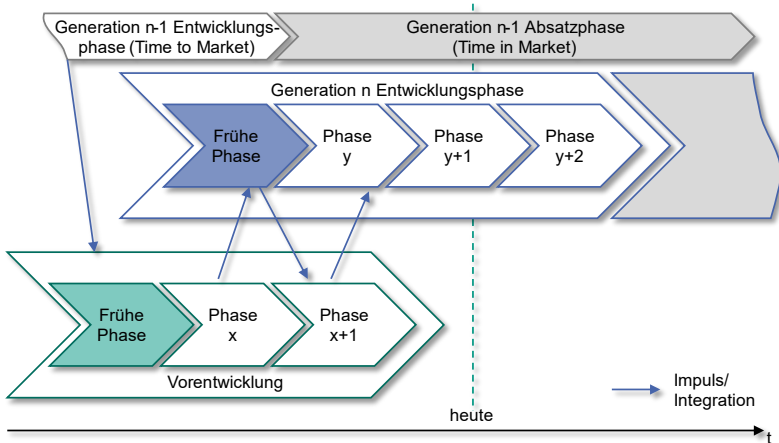


Abbildung 2.35: Schematische Darstellung der „Frühen Phasen“ einer Produkt-Generationsentwicklung und der dazugehörigen Vorentwicklung eines (Teil-)Systems nach Albers, Rapp et al. (2017).

Zur besseren zeitlichen und inhaltlichen Einordnung der Forschungs- und Entwicklungsaktivitäten wird nachfolgend eine Vorstellung und Gliederung ebendieser auf Basis der Arbeiten von Specht et al. (2002) sowie Albers, Kürten et al. (2022) vorgenommen. Specht et. al unterscheidet in Anbetracht des jeweiligen Anwendungsbezuges vier verschiedene Aktivitäten (Abbildung 2.36): Die Grundlagenforschung, die Technologieentwicklung, die Vorentwicklung und die Produkt-/Prozessentwicklung (Specht et al., 2002).

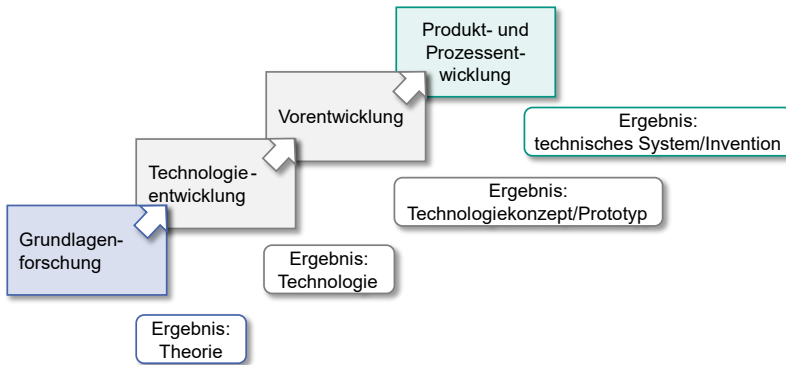


Abbildung 2.36: Aktivitäten der Forschung & Entwicklung erweiterte Darstellung nach Specht et al. (2002).

Unter der **Grundlagenforschung** wird jener Erkenntnisgewinn verstanden, der sich in Form abgeleiteter Theorien äußert und keinen unmittelbaren praktischen Bezug aufweisen muss. Die Ergebnisse der Grundlagenforschung stellen jedoch die wichtigste Referenz und Basis für anwendungsorientiertes Wissen dar (Specht et al., 2002, S. 15). Die darauf aufbauende **Technologieentwicklung** umfasst alle Aktivitäten in Zusammenhang mit der Gewinnung und Weiterentwicklung des Wissens hinsichtlich dessen praktischer Anwendbarkeit. Ziel der Unternehmungen ist die Bildung technologischer Leistungspotentiale und Kernkompetenzen. Diese werden im Zuge der **Vorentwicklung** auf deren produktspezifische Anwendbarkeit geprüft und folglich anwendungsorientiert weiterentwickelt. Die Ergebnisse der Vorentwicklungsaktivitäten zielen dabei auf die Definition verschiedener Produktkonzepte sowie rudimentärer Funktionsnachweise in Form erster Prototypen ab. Zu diesem Zeitpunkt umgesetzte Produktspezifikationen sind lediglich im Rahmen des Vorentwicklungsprojektes bewertet (Albers, Rapp et al., 2017) Dies steht in klarer Abgrenzung zur Produktspezifikation die innerhalb der anschließenden Frühen Phase der Produktgenerationsentwicklung als Element des initialen Zielsystems erstellt wird. In dieser Phase wird zunächst geprüft, inwieweit die Ergebnisse der Vorentwicklung und anderer Aktivitäten zur Integration in die neue Produktgeneration geeignet sind (Albers, Rapp et al. 2017 dort nach Heismann & Maul, 2012). Die von Specht et al. (2002) als **Produkt- und Prozessentwicklung** deklarierte Aktivität umfasst die Entwicklung eines konkreten Produktes/Prozesses. Sie findet ihren Abschluss in der unmittelbaren Einführung

eines neuen Produktes am Markt und der damit verbundenen Inbetriebnahme des zugrundeliegenden Produktionssystems. Die Entwicklungsphasen der aktuellen Produktgeneration ( $G_n$ ) lassen sich folglich jener F&E-Aktivität zuordnen. Für den Ablauf der F&E-Aktivitäten ist keine verbindliche, zeitliche Abfolge festgelegt. Vielmehr können die unterschiedlichen Aktivitäten zeitweise parallel erfolgen.<sup>33</sup> Ebenfalls herrscht keine strenge Kopplung der Aktivitäten, so muss beispielsweise eine Vorausbildung nicht zwingend die Folge einer spezifischen Technologieentwicklung sein. Im Entwicklungsprozess können somit Ergebnisse verschiedenster F&E-Aktivitäten innerhalb weiterer Aktivitäten genutzt werden und als Elemente des Referenzsystems einer Produktgeneration bzw. einer Systemgeneration dienen (Albers, Kürten et al., 2022). Ein klarer Produktbezug der Ergebnisse wird letzten Endes erst durch die Vorausbildungstätigkeiten und die Entwicklung einer Produktgeneration geschaffen.

### 2.3.3 Fazit

Das Modell der PGE nach Albers ermöglicht es, aus der Praxis gewonnene Erkenntnisse über die generationenübergreifende Nutzung von Wissen, aktiv für den Syntheseprozess menschenzentrierter Entwicklungsmethoden nutzbar zu machen. Mittels der bewussten Anwendung von Übernahmevariation kann, der gerade in frühen Entwicklungsphasen vorherrschenden Unsicherheit entgegengewirkt werden. Durch die Erweiterung des Vokabulars um den Begriff des Referenzsystems lässt sich erklären, dass auch bloße Entwicklungsartefakte, beispielsweise aus Konzeptstudien früher Phasen der Forschung und Entwicklung, in die Produktentwicklung einfließen können. Im Rahmen der mechatronischen Steuergeräteentwicklung kann diese Erkenntnis u.a. dazu genutzt werden, große Teile der geforderten Systemdokumentation inklusive der Maßnahmen zur Sicherstellung einer durchgängigen Traceability, für Folgegenerationen zu übernehmen. Hierin liegt das Potential bedeutende Effizienzgewinne zu erzielen. Durch die Betrachtung einzelner Phasen der Forschung und Entwicklung im Kontext der PGE, lassen sich zusätzlich die unternehmensseitigen Tätigkeiten im Rahmen dieser Arbeit entsprechend der wissenschaftlichen Grundlagen abbilden.

---

<sup>33</sup> Beispielhaft hierfür sei die Beobachtung von Albers et al. im Entwicklungsprozess von Smartphones erwähnt (Albers et al. 2017) dort anhand der Vorentwicklung und der Produktgenerationsentwicklung

## **2.4 Baukastenentwicklung als Standardisierungsmethode**

Auf Basis der in Kapitel 2.2 illustrierten Grundlagen von Systemtheorie und Produktarchitekturen wird im Folgenden die Baukastenentwicklung als eine der bekanntesten, produktseitigen Standardisierungsmethoden vorgestellt (Albers, Bursac, Scherer, Birk, Powelske & Muschik, 2019). Anknüpfend an die Motivation zur Einführung standardisierter modularer Produktarchitekturen werden deren grundlegende Begrifflichkeiten dargelegt. Darauf aufbauend erfolgen die Darstellung der spezifischen Herausforderungen der Baukastenentwicklung und die Diskussion der besonderen Eignung von MBSE-Methoden zur Unterstützung in der Frühen Phase der PGE. Letztere wird am Beispiel bestehender Ansätze erläutert. Der abschließende Abschnitt widmet sich dem Verständnis des Begriffs der Kompatibilität im Rahmen dieser Arbeit und präsentiert einen Ansatz zur Kompatibilitätsbewertung, der die Nutzung einer bestehenden Wissensbasis im Sinne der PGE ermöglicht.

### **2.4.1 Grundlagen der Standardisierung modularer Produktarchitekturen**

Mit Beginn der industriellen Massenproduktion der 50er Jahre und den anschließenden Absatzengpässen in den 60er Jahren hat ein stetiger Wandel der Märkte auf die Bedürfnisse des Kunden hin stattgefunden (Sander, 2001). Man spricht hierbei von einem sogenannten Käufermarkt. Heutzutage erschweren gestiegene, häufig individualisierte Kundenansprüche und ein gleichzeitig zunehmend homogenes, austauschbares Produktangebot die Absatzsituation von Unternehmen. Das Eindringen immer neuer Anbieter in angestammten Märkten infolge der allgemeinen Internationalisierung, führt zu einer zusätzlichen Verschärfung des Wettbewerbes (Sander, 2001). Die Befriedigung der heterogenen Kundenanforderungen resultiert dabei für die Unternehmen in einer gesteigerten Anzahl am Markt angebotener, individualisierter Produktvarianten. In der Literatur wird dieser Vorgang als Maximierung der externen Vielfalt bezeichnet. Um gleichzeitig, hinsichtlich der Entwicklungs- und Produktionskosten wettbewerbsfähig zu bleiben bedarf es Maßnahmen, die unternehmensinterne Vielfalt (z.B. verwendeter Teile und Prozesse) zu minimieren (Kersten, 1999; Piller & Waringer, 1999; Schuh & Riesener, 2018; Renner, 2007; Franke et al., 2002). Bewährte Maßnahme der Produktentwicklung ist die Anwendung von Standardisierungsmethoden in Kombination mit der Nutzung vorwiegend modularer Produktarchitekturen (Franke et al., 2002). Im Kern dieser Strategie liegt die Beschränkung der Elementanzahl und deren Wiederverwendung zur Bereitstellung einer großen Bandbreite individualisierter Produkte (Schuh, Rudolf, Thomas et al.,

2014; Franke et al., 2002). Die Wiederverwendung standardisierter Elemente selbst hat eine Vielzahl an verschiedenen Ansätzen hervorgebracht. Diese reichen von einer Größen- und Leistungsskalierung von Produkten gleicher Funktion, bis hin zu einer individuellen Rekombination austauschbarer Grundkomponenten zur Darstellung neuer Funktionsumfänge. Letztere stellt den Grundgedanken der Baukastenentwicklung dar. Bedeutung und Potentiale der Standardisierungsmethoden und der Modularisierung für die Produktentwicklung lassen sich anhand der intensiven, seit vielen Jahrzehnten andauernden Forschung auf diesem Gebiet illustrieren.<sup>34</sup> Der folgende Abschnitt, soll einem kurzen Überblick der umfangreichen Forschungsarbeiten mit Fokus auf modell-basierte systemtheoretische Ansätze dienen:

Die Überlegungen zur Bildung modularer Strukturen existieren bereits lange vor deren Einzug in die Produktentwicklung. Ansätze dieser Natur finden sich bereits in den frühen 60er Jahren auf dem Gebiet der Sozialforschung (Simon, 1962) und der Architektur (Alexander, Ishikawa & Silverstein, 1977). Im Mittelpunkt der Untersuchungen steht die Strukturierung eines Systems in, in sich abgeschlossene, weitestgehend unabhängige Teilsysteme. Dieser Vorgang hat das Ziel einen erfolgreichen Umgang mit der steigenden systemischen Komplexität zu ermöglichen. Seitens der Produktentwicklung sind es 30 Jahre später Forscher wie Stone und Pimmler & Eppinger die diesen Gedanken bei der Dekomposition integraler Produkte zu einer Struktur funktional und baulich autonomer Subsysteme aufgreifen (Pimmler & Eppinger, 1994; Stone, 1997; sowie Eilmus et al., 2012). Die daraus abgeleiteten Designregeln wie Funktions- und Flussorientierung bilden bis heute Grundlage einer Vielzahl an Modularisierungsansätzen (Renner, 2007). Ein detaillierter Vergleich vorwiegend produktarchitekturbezogener Methoden findet sich bei Blees (2011). Die Einführung standardisierter, modularer Architekturen hat weitreichende Auswirkungen auf die unternehmensinterne Prozesslandschaft (Ulrich, 1994) und den gesamten Produktentstehungsprozess. Göpfert beschäftigt sich mit der Entwicklung einer geeigneten Organisationsstruktur, die die geforderte Minimierung der internen Vielfalt für Teile und Prozesse gleichermaßen zulässt (Göpfert, 1998). Hinsichtlich der Maximierung der externen Vielfalt erweisen sich Standardisierung und Modularisierung für die Breite des Produktprogrammes, Lieferzeiten und die generelle Verfügbarkeit als sehr positiv. Gemessen am benötigten Bauraum, dem Gewicht und der Leistungsbereitstellung schneiden

---

<sup>34</sup> Eine intensive Auseinandersetzung mit der Literatur der Standardisierungsmethoden findet sich bei Jiao et al. (2007), sowie in Bezug auf modulare Architekturen bei Albers, Bursac, Scherer, Birk, Powelske & Muschik (2019)

modulare Produkte jedoch meist deutlich schlechter ab als vergleichbare integral gestaltete Produkte. Zusätzlich wird die mangelnde Differenzierung über die angebotene Variantenvielfalt in einigen Forschungsarbeiten als Herausforderung der Modularisierung herausgestellt (Steffen, 2007; Renner, 2007; Scherer, 2016; Blees, 2011; Hölttä-Otto & deWeck, 2007a). Eine detaillierte Auseinandersetzung mit den allgemeinen Effekten von Individualisierung auf die Produktkosten und den wirtschaftlichen Auswirkungen mangelnder Differenzierung eines mittels Standardisierung geschaffenen Portfolios findet sich bei Robertson & Ulrich (1998). Über die Betrachtung der Auswirkungen von Modularisierung auf die Zusammenarbeit innerhalb eines interdisziplinären Entwicklungsteams nähert sich Sosa der Bedeutung von Wechselwirkungen zwischen Modulen und der Herausforderungen in deren Abbildung (Sosa et al., 2000; Sosa et al., 2004). Zur Organisation und Darstellung der produktinternen Wechselwirkungen von Teilsystemen bilden die von Steward (1981) entwickelten sogenannten Design-Structure-Matrizen (DSM) eine wichtige Basis diverser Ansätze für Standardisierungsmethoden (Pimmler & Eppinger, 1994; Lindemann, 2009; Sosa et al., 2003; Van Beek et al., 2010; Hölttä-Otto & deWeck, 2007b; Steffen, 2007). Eine detaillierte Zusammenfassung über Einsatzzwecke der DSM liefern Eppinger & Browning (2012). In der Literatur finden sich darüber hinaus extensive Arbeiten zur Rolle und der Ableitung eindeutiger Schnittstellen zur Unterstützung der Austauschbarkeit von Teilsystemen (Blackenfelt & Sellgren, 2000; Salvador, 2007). Erixon illustriert in seinem Ansatz zum "Modular Function Deployment" (Erixon, 1998), dass abhängig von der jeweiligen Phase im Produktentstehungsprozess verschiedene Schnittstellen zum Tragen kommen. Diese bezieht er ausgehend von den Kundenanforderungen bis zum Produktrecycling in seinen Modularisierungsprozess mit ein. Ähnliche auf die ganzheitliche Betrachtung des Produktlebenszyklus ausgelegte Ansätze sind beispielsweise die „METUS“-Methode von Göpfert (1998) oder die Methoden zur Entwicklung modularer Produktstrukturen von Schuh, Rudolf & Vogels (2014). Ausgehend von der Vielzahl an Definitionen der zentralen Mechanismen auf dem Gebiet der modularen Bauweise haben Albers, Scherer et al. (2015) eine auf der Systemtheorie basierende Ontologie trennscharfer Begriffe im Bereich der Standardisierungsmethoden gebildet. Diese ist zur expliziten Verwendung im Bereich der modellbasierten Entwicklung mechatronischer Systeme bereitgestellt (Bursac, 2016). Im Folgenden wird das auf dem Kanon der Ontologie gründende Verständnis der Standardisierungsmethoden im Kontext dieser Arbeit dargelegt:



### **Definition 16: Modul**

Ein Modul ist ein technisches Subsystem, das sich durch andere technische Subsysteme ersetzen lässt, so, dass die Menge aller Funktionen oder Attribute des technischen Systems variiert wird. (Albers, Scherer et al., 2015)

Demnach entscheiden Auswahl und Kombination der Subsysteme über den Funktionsumfang des modularen Gesamtsystems. Die Einführung weniger standardisierter Schnittstellen und die Strukturierung des Systems in abgeschlossene, möglichst unabhängige Teilsysteme sind dabei unmittelbare Voraussetzung für eine zielführende Substitutionsfähigkeit der Module. (vgl. Kapitel 2.2.4 Produktarchitekturen) Als besondere Ausprägungen dieser Substitution gelten das gezielte Weglassen und Hinzufügen eines Moduls. Im interdisziplinären Umfeld mechatronischer Systeme stellt die Modulableitung, also die Identifikation der unabhängigen Teilsysteme und die Definition kompatibler Schnittstellen unter Berücksichtigung negativer Emergenzen eine besondere Herausforderung dar (Powelske et al., 2017). Die Festlegung der Modulgrenzen kann im Allgemeinen zwar methodisch unterstützt werden, beruht im Grunde jedoch auf der subjektiven Einschätzung von Experten. Je nach Kontext ergeben sich abhängig vom zugrundeliegenden Portfolio und der betrachteten Phasen des Produktentstehungsprozesses unterschiedliche Schnittstellen, die es zu harmonisieren gilt. (Blees, 2011; Steffen, 2007; Göpfert, 1998; Erixon, 1998; Powelske et al., 2017)

### **Definition 17: Plattform und Hut**

Eine Plattform ist die Menge jener Subsysteme, die in unterschiedlichen technischen Systemen unverändert zum Einsatz kommt. Der Hut hingegen umfasst die restlichen Subsysteme, die in den unterschiedlichen technischen Systemen die Menge aller Funktionen oder Attribute variieren. (Albers, Scherer et al., 2015)

Die Subsysteme einer Plattform müssen dabei nicht zwingend physisch miteinander verbunden sein. Damit sind Kombinationen aus Software- und Hardwarekomponenten, wie in der Steuergeräteentwicklung in Form von Mikrocontrollern und deren Basissoftware alltäglich, bei durchgängiger Nutzung ebenfalls als Plattform zu verstehen. Im Falle ausnahmslos physisch verbundener Plattformensubsysteme und gleichzeitig standardisierten Schnittstellen zu den Hutssystemen sprechen Albers, Scherer et al. (2015) von einem Schnittstellen-Träger. Dieser lässt sich beispielhaft anhand eines Substrates in Form einer Leiterplatte in der Steuergeräteentwicklung illustrieren.

**Definition 18: Baureihe**

Eine Baureihe bezeichnet mehrere technische Systeme, die eine ähnliche Produktarchitektur aufweisen. Die jeweiligen technischen Systeme unterscheiden sich dabei durch die Ausprägung einzelner Attribute, die durch Skalieren variiert werden. (Albers, Scherer et al., 2015)

Die Skalierung der Subsysteme einer Baureihe kann auf Basis der Ähnlichkeitsgesetze beispielsweise anhand einer Variation der Größe, der Leistung oder einer Kombination erfolgen. Exemplarisch dafür folgen die, in der Steuergeräteentwicklung eingesetzten MOSFETs häufig einer leistungs- und größenskalierten Baureihe.

**Definition 19: Baukasten**

Ein Baukasten ist die Menge aller technischen Subsysteme, die dem zugehörigen Baukasten-Regelwerk folgen, mit dem Ziel, aus diesen Subsystemen technische Systeme mit jeweils unterschiedlicher Menge aller Funktionen konfigurieren zu können. (Albers, Scherer et al., 2015)

Die flexible Konfigurierbarkeit des Funktionsumfangs abgeleiteter Systeme, stellt eine deutlich größere Herausforderung an die Kompatibilität der beteiligten Subsysteme dar, als es in den bereits thematisierten Standardisierungsmethoden der Fall ist. Sie wird über klare Vorgaben an die Schnittstellen der Module als zulässige Architektur (funktional und baulich) der abzuleitenden Produkte im Baukasten-Regelwerk festgelegt. Der Baukasten selbst umfasst die Gesamtheit der zur Konfiguration bereitstehenden Subsysteme - auch Bausteine genannt<sup>35</sup> - stellt in diesem Zusammenhang jedoch ein rein abstraktes Gebilde dar. Dieses Charakteristikum führt zu der unmittelbaren Notwendigkeit der Nutzung von Modellen im Rahmen der Baukastenentwicklung, welche im nächsten Abschnitt anhand vorhandener MBSE-gestützter Methoden erläutert wird. Zu den Tätigkeiten der Baukastenentwicklung zählen das Aufstellen des Baukasten-Regelwerks, die Entwicklung der Subsysteme für den Baukasten, sowie die Konfiguration spezifischer Produkte aus dem Baukasten. Abhängig davon, wer den Konfigurationsschritt eines Produktes aus dem Baukasten vornimmt, unterscheidet

---

<sup>35</sup>Demnach sind alle Elemente eines Baukastens Bausteine, jedoch nur diejenigen, die sich durch andere Subsysteme substituieren lassen auch Module. Tiefere Erläuterungen hierzu finden sich bei Bursac (2016).

man Anwender- (z.B. im Bereich der Power-Tools) von Herstellerbaukästen (z.B. in der Automobilherstellung).<sup>36</sup>

Betrachtet man den Einsatz der Standardisierungsmethoden auf Basis modularer Produktarchitekturen in realen Produktentstehungsprozessen, so finden sich in der Regel Kombinationen dieser Methoden auf unterschiedlichen Abstraktionsebenen wieder. (Albers Scherer et al., 2015; Ehrlenspiel, 2009, Schuh, 2005; Hoffmann & Vietor, 2014; Scherer, 2016; Bursac, 2016)

Am Beispiel der Steuergeräteentwicklung soll dies illustriert werden:

Um die unterschiedlichen Einsatzzwecke der Steuergeräte abzudecken, folgt deren Entwicklung auf Systemebene einem Baukastenansatz. Die einzelnen elektronischen Bauteile sind wiederum in Form eines Plattformansatzes mit dem zugrundeliegenden Substrat kombiniert. Die Bauteile selbst können als Baureihenansatz realisiert sein (Abbildung 2.37).

---

<sup>36</sup> Eine detailliertere um die Parameter „Systemabgrenzung“, „Baukastenstruktur“ und „Reinheit“ ergänzte Klassifizierung von Baukästen findet sich bei Ponn & Lindemann (2008).

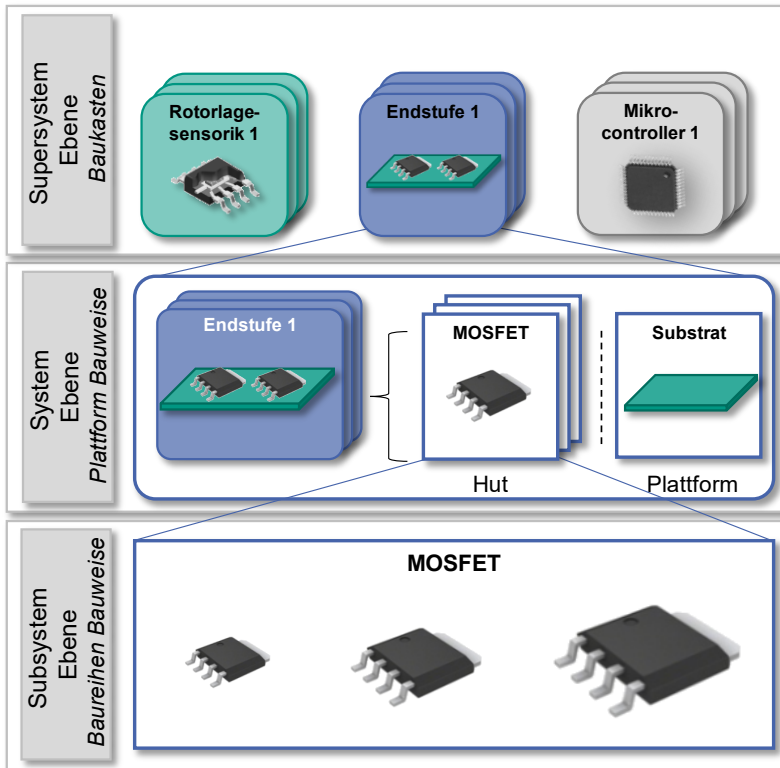


Abbildung 2.37: Fraktaler Charakter der Standardisierungsmethoden am Beispiel der Baukastenentwicklung mechatronischer Steuergeräte im Einsatz in Kleinantrieben im Automobilbereich. Darstellung in Anlehnung an Albers, Scherer et al. (2015).

Entsprechend der hergeleiteten Definitionen erfolgt die Zuordnung eines Systems zu einer der Standardisierungsmethoden anhand der Eigenschaften seiner Subsysteme. Unter Beachtung des hierarchischen Konzepts der Systementwicklung (vgl. Kapitel 2.2.1) lässt sich dieser Sachverhalt ebenfalls auf die Ebenen von Super- und Subsystemen übertragen. Albers, Scherer et al. (2015) sprechen hierbei vom fraktalen Charakter der Standardisierungsmethoden.

## 2.4.2 MBSE-gestützte Baukastenentwicklung

Im folgenden Abschnitt wird ein MBSE-gestützter Ansatz zur Baukastenentwicklung vorgestellt. Dieser ermöglicht es durch die Nutzung von Wissen aus Produktgenerationen im Kontext der PGE den Herausforderungen der Baukastenentwicklung in realen Produktentstehungsprozessen effizient zu begegnen. Die Motivation des Ansatzes erfolgt in zwei Schritten. Zunächst werden die einzelnen Aktivitäten der Baukastenentwicklung anhand der Einordnung des weit verbreiteten Begriffes der Modularisierung vorgestellt. Anschließend erfolgt eine Auseinandersetzung mit den angesprochenen Herausforderungen realer Produktentstehungsprozesse und der damit verbundenen Erkenntnis, warum nur kontinuierliche Ansätze in der Lage sind, diese erfolgreich zu adressieren.

Betrachtet man die Vielzahl der im vorhergehenden Abschnitt vorgestellten Methoden zur Modularisierung (vgl. Kapitel 2.4.1) so erscheint eine Unterstützung des Entwicklers bei der Definition geeigneter Schnittstellen und Modulausprägungen in ausreichendem Maße vorhanden zu sein. Tatsächlich bieten die verschiedenen Ansätze ausführliche Hilfestellungen in Form von Regelwerken zur Überführung nicht-modularer Produktgenerationen in tragfähige Modulkonzepte. Dabei wird der Fokus entweder auf einzelne Phasen des Produktentstehungsprozesses (Lindemann, 2009; Stone, 1997; Zamirowski & Otto, 1999; Pimpler & Eppinger, 1994; Steffen, 2007; Salvador, 2007, Kopenhagen, 2004; Bles, 2011, um hier nur einige zu nennen) oder den gesamten Produktlebenszyklus einer Produktgeneration (Erixon, 1998; Göpfert, 1998) gelegt. Stellt man diesen Ansätzen die Definition des Baukastens nach Albers, Scherer, Bursac & Rachenkova (2015) gegenüber, wird ersichtlich, dass die Erstellung eines Regelwerkes zur Ableitung modularer Subsysteme lediglich einen Teil der Baukastenentwicklungsaktivitäten umfasst (vgl. Kapitel 2.4.1). Überlegungen, die eine ganzheitliche Betrachtung der Baukastenentwicklung beinhalten existieren beispielsweise in den Ansätzen von Schuh (2015). Diese berücksichtigen neben der Erstellung des Baukastenregelwerkes (der Modularisierung) und der daraus folgenden Ableitung der Subsysteme auch die spätere Konfiguration von Produkten aus dem Baukasten. Ergebnis dieser Ansätze ist ein in sequenzielle Phasen aufgeteiltes Prozessmodell der Baukastenentwicklung. Aus dem Phasenmodell geht hervor, dass die Zielsystembildung des Baukastens vor den parallelen Entstehungsprozessen der abgeleiteten Produkte, abgeschlossen wird. Ehrlenspiel (2009) und Kohlhasse (1997) identifizieren bei einer solchen Betrachtung jedoch einen Konflikt mit der Entwicklungspraktik realer Produktentstehungsprozesse.

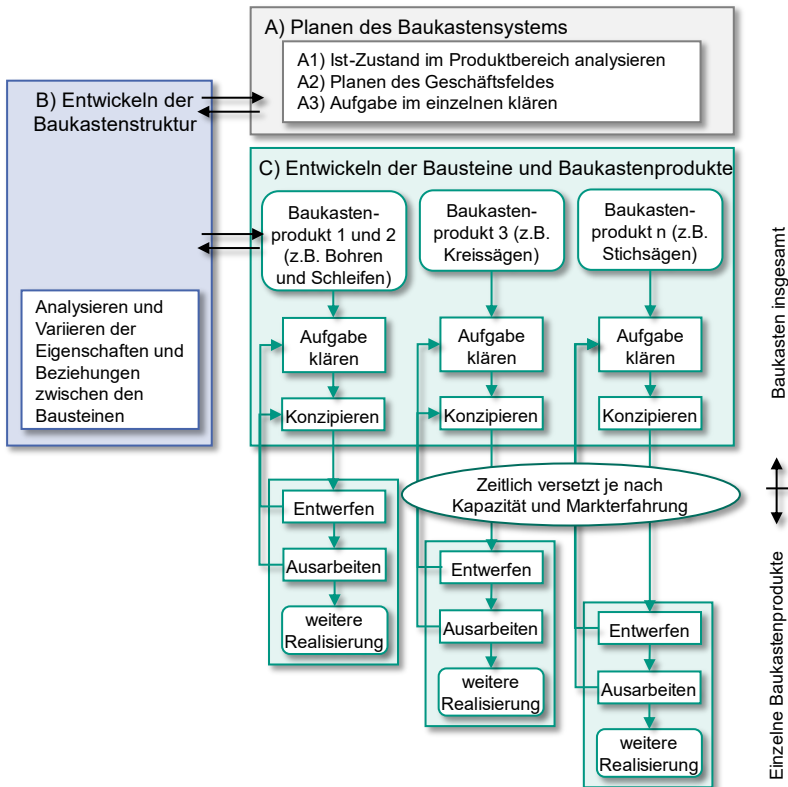


Abbildung 2.38: Vorgehen beim Entwickeln eines Baukastensystems nach Ehrlenspiel (2009).

Um ihre Ressourcen (Fertigungslinien, Mitarbeiter, ...) optimal auslasten zu können, starten Unternehmen die Entwicklungsprozesse der einzelnen Produkte aus dem Baukasten mit zeitlichem Versatz, welcher je nach Branche auch mehrere Jahre betragen kann.<sup>37</sup> Vor diesem Hintergrund verortet Ehrlenspiel (2009) die Erstellung

<sup>37</sup> Als Paradebeispiel kann an dieser Stelle die Automobilindustrie herangezogen werden

seines Baukastenzielsystems parallel und nicht sequenziell zur Zielsystembildung der abgeleiteten Produkte (Abbildung 2.38). Beide Aktivitäten sind seinem Modell entsprechend, vor der Ableitung<sup>38</sup> des ersten Produktes abgeschlossen. Albers et al. stellen auf Basis der gegenwärtigen Forschung zu Zielsystemen (Albers, Ebel & Lohmeyer, 2012; Albers, Lohmeyer & Ebel, 2011; Ebel, 2015 und Lohmeyer, 2013) jedoch fest, dass sich aus dem zeitlichen Versatz zusätzliche Herausforderungen an die Baukastenentwicklung ergeben. Entsprechend der Frühen Phase der PGE besteht für den Baukasten beispielsweise eine hohe Unsicherheit gegenüber den Zielsystemen seiner zukünftig abgeleiteten Produktgenerationen. Aufgrund des großen Zeitversatzes konkretisieren<sup>39</sup> sich deren zunächst vage Anforderungen und Ziele häufig erst, nachdem bereits diverse Produkte aus dem Baukasten abgeleitet wurden. Gleichzeitig bestehen durch die Nutzung von Modulen als Grundlage der Produktableitung sehr hohe Abhängigkeiten zwischen den Produkt- und Modulzielsystemen, was in einer steigenden Komplexität für das Anforderungsmanagement der Module äußert (Renner, 2007; Scherer, 2016). Die Konkretisierung einzelner Produktziele kann zudem die Anpassung bestehender oder die Entwicklung neuer Module zur Erfüllung der Kundenwünsche erfordern. Veränderungen an den Modulen des Baukastens haben eine direkte Kopplung zum Baukasten selbst und bedingen beispielsweise direkte Maßnahmen zur Kompatibilitätswahrung im Baukastenregelwerk. Albers et al. stellen fest, dass diese Mechanismen über die ganze Dauer zwischen dem Entwicklungsbeginn des ersten Baukastenproduktes bis zum Abschluss des letzten Produktes auftreten (Abbildung 2.39).

---

<sup>38</sup> Konkret spricht er hierbei von der Aktivität des Entwerfens (vgl. Abbildung 2.38)

<sup>39</sup> Die Konkretisierung des Zielsystems ist entsprechend des erweiterten ZHO-Modells als kontinuierlicher Prozess zu verstehen, der nur mittels kontinuierlicher Validierung gewährleistet werden kann (vgl. Kapitel 2.2.3)

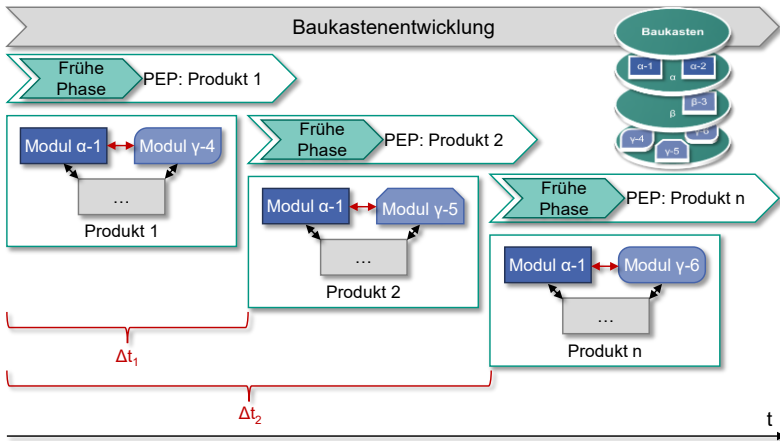


Abbildung 2.39: Herausforderungen der Baukastenentwicklung in Anlehnung an Bursac (2016) dort nach Albers, Bursac & Scherer (2014).

Um den Herausforderungen des Zeitversatzes und der steigenden Abhängigkeiten erfolgreich begegnen zu können, muss die Baukastenentwicklung demnach als kontinuierliche Aktivität im Kontext der PGE betrachtet werden (Albers, Bursac, Scherer, Birk, Powelske & Muschik, 2019; Bursac, 2016).

### Modellierungs-Framework zur Unterstützung der Baukastenentwicklung

Die in Kapitel 2.4.1 beschriebene Reduktion der unternehmensinternen Teilevielfalt, führt aus Perspektive der Baukastenentwicklung zu einem drastischen Anstieg der Komplexität im Entwicklungsprozess. Die Komplexitätszunahme lässt sich nach Weber (2005) mit dem bereits dargelegten Anstieg von Elementen und Verbindungen innerhalb und zwischen den Zielsystemen von Baukasten, Modulen und Produkten begründen. Die hohe systemische Komplexität des Entwicklungsprozesses und der abstrakte Charakter des Baukastens bieten eine ideale Grundlage für den Einsatz MBSE-gestützter Methoden (vgl. Kapitel 2.2). Ein entsprechender MBSE-gestützter Ansatz für die Baukastenentwicklung findet sich bei Bursac (2016) Dieser stützt sich auf die Aussage, dass die Baukastenentwicklung nur unter Nutzung bereits vorhandenen Wissens aus vorhergehenden Produktgenerationen, nicht aber auf der „grünen Wiese“ gelingen kann. Das vorhandene Wissen liegt dabei in Form von spezifischen Produktmodellen vor. Dieses Wissen nutzt Bursac (2016) aktiv dazu, um daraus



schrittweise das abstrakte Modell eines Baukastens zu synthetisieren. Innerhalb des Syntheseprozesses ist es unerlässlich die unterschiedlichen Abstraktionsgrade der Produktmodellierung<sup>40</sup> zu kennen und deren Zusammenhänge ergünden zu können. Bursac (2016) nutzt daher das Produktmodellierungsframework von Albers et al. als Grundlage seiner Arbeit. Dieses wurde auf Basis des Phasenmodells des iPem (vgl. Kapitel 2.2.3) geschaffen und ermöglicht nicht nur eine Beschreibung verwendeter Modelle auf verschiedenen Abstraktionsgraden, sondern erleichtert die Wiederverwendung von Modellinhalten und –strukturen im Kontext der PGE (Albers, Matthiesen, Bursac, Moeser, Schmidt & Lüdcke, 2014). Zum weiteren Verständnis wird der Aufbau des Frameworks in groben Zügen dargelegt (Abbildung 2.40):

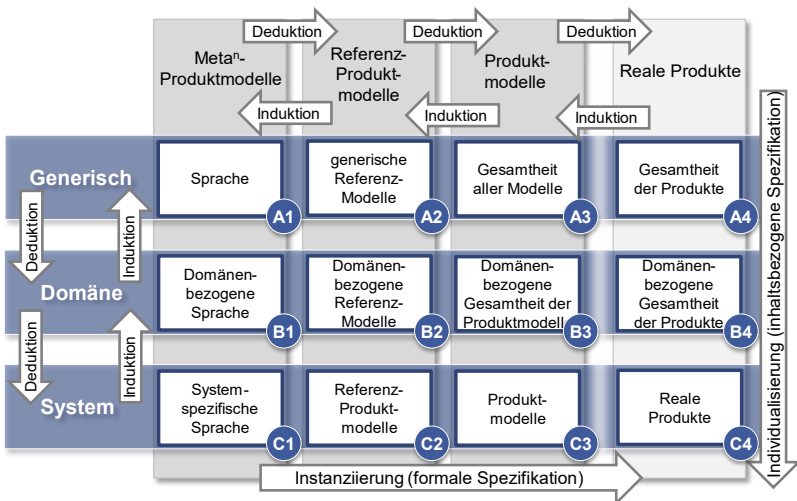


Abbildung 2.40: Abstraktionsgrade der Produktmodellierung (Bursac, 2016).

Das Framework umfasst zwei Achsen. Auf der horizontalen Achse wird zwischen Meta-Modellen, Referenzproduktmodellen, Produktmodellen und zusätzlich den realen Produkten unterschieden. Entlang der Vertikalen erfolgt die Konkretisierung

<sup>40</sup> Gerade durch die hohe Komplexität interdisziplinärer Entwicklungsprozesse kommt diesem Punkt eine zusätzliche Bedeutung zu

von der generischen Ebene bis zur Systemebene. Dazwischen findet sich die Ebene der Domänen, die je nach gewünschtem Detaillierungsgrad, weiter unterteilt werden kann. Mittels der Domänenebene lassen sich beispielsweise die unterschiedlichen Disziplinen der Steuergeräteentwicklung fassen: Mechanikentwicklung, Elektronikentwicklung und Softwareentwicklung. Die entstandenen zwölf Felder ermöglichen eine Zuordnung der in der Produktentwicklung anfallenden Modelle. Diese können durch Induktions- und Deduktionsschritte innerhalb des Frameworks ineinander überführt werden. Die Transformation kann beispielhaft an der Verwendung der SysML zur Abbildung des C&C<sup>2</sup>-Ansatzes (Zingel, 2013) illustriert werden. Hierbei wird durch das gezielte Hinzufügen von Sprachelementen das generische Meta-Modell der Sprache, mittels Deduktion in einen für die Domäne der Mechanik geltenden Sprachstandard überführt. Einordnung und Transformation ermöglichen, die Wiederverwendung der Modelle im Kontext der PGE. Den Referenzmodellen kommt dabei eine besondere Bedeutung zu. Durch deren Nutzung können nachweislich unnötige Aufwände in der Entwicklung komplexer Produkte vermieden werden (Müller et al., 2017). Gleichzeitig lassen sich über den Wiederverwendungsgedanken, die in der Anwendung von MBSE-Methoden bemängelten, hohen Initialaufwände drastisch reduzieren (vgl. Kapitel 2.2.7). Durch das vorliegende Produktmodellierungs-Framework wurde ein bedeutender Beitrag zur Akzeptanzsteigerung der MBSE-Methoden geschaffen. Diversen Forschungsarbeiten im Bereich des MBSE (Scherer, 2016; Zingel, 2013; Bursac, 2016; Heitger, 2019) bietet es eine wichtige Grundlage, so auch der vorliegenden Arbeit.

Für die Unterstützung der Baukastenentwicklung appliziert Bursac das vorgestellte Framework (Abbildung 2.41).

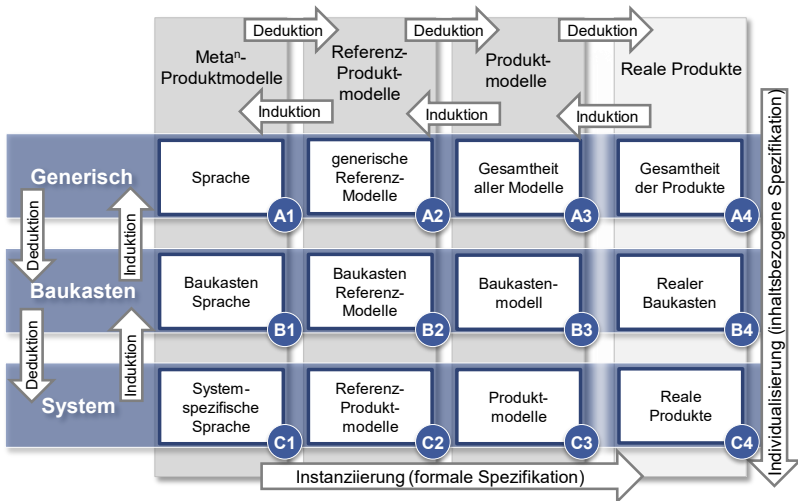


Abbildung 2.41: Framework der Baukastenentwicklung in Anlehnung an Bursac (2016).

In vertikaler Richtung verortet er den Baukasten auf Ebene der Domänen. Aus der Kreuzung mit den Spalten der Produktmodelle und der Referenz-Produktmodelle auf der horizontalen Achse ergeben sich die für seinen Ansatz relevanten Elemente des Frameworks: Die systemspezifischen Produktmodelle (C3) und Referenzproduktmodelle (C2), sowie die Baukastenreferenzmodelle (B2) und das Baukastenmodell (B3). Besonderen Schwerpunkt in seiner Betrachtung hat die aktive Überführung der einzelnen Modelle mittels Induktion und Deduktion. Diese Vorgänge ermöglichen den Übertrag der Aktivitäten der Baukastenentwicklung auf die Modellwelt. Ergo lassen sich die Synthese des Baukastenmodells aus initialen Produktmodellen (Modularisierung + Baukastenregelwerk) sowie die Ableitung neuer Generationen von Produktmodellen aus dem Baukasten unter Anwendung des Frameworks abbilden. Zur Erläuterung der theoretischen Zusammenhänge soll



mittels eines Induktionsschrittes sogenannte Referenz-Produktmodelle (C2) erzeugen. Diese beinhalten neben Informationen über die Struktur der Referenzprodukte ebenfalls konkrete Informationen bspw. zu Bauraumvorgaben oder Typ-Teilenummern. Die Referenz-Produktmodelle bilden eine Wissensbasis, die jederzeit entsprechend den Erfordernissen eines Projektes deduktiv wieder zu einem Produktmodell konkretisiert werden kann. Unterzieht man die gebildeten Referenz-Produktmodelle einem Induktionsschritt, so können diese zu einem Baukasten-Referenzmodell (B2) zusammengefasst werden. Hierzu werden die produktübergreifenden Gemeinsamkeiten der Referenz-Produktmodelle analysiert und von produktspezifischen Umfängen unterschieden. Bei der Zusammenführung im Baukasten-Referenzmodell wird anschließend eindeutig festgelegt, was als Element des späteren Baukastens gezählt wird und welche produktspezifischen Anpassungen dem Baukastenregelwerk widersprechen. Um den steigenden Anforderungen an die Individualität der Produkte zu entsprechen, werden vermehrt für Kunden sichtbare Komponenten aus der Baukastenentwicklung bzw. den Baukastenreferenzmodellen ausgenommen. Mit Verlassen der systemspezifischen Ebene des Frameworks ändert sich ebenfalls der Charakter des abgebildeten Referenzmodells. Während die Referenz-Produktmodelle mit Fokus auf der vollständigen Abbildung von Produkten erstellt werden, müssen die Baukasten-Referenzmodelle gleichzeitig den Austausch und die Vergleichbarkeit der Komponenten ermöglichen. Damit die unterschiedlichen Zielstellungen miteinander vereinbar bleiben und die Modelle sowohl bei der induktiven als auch der deduktiven Überführung konsistent bleiben, ist es erforderlich, dass diese aufeinander basieren und nicht beliebig festgelegt werden. Wie bereits beschrieben, kann es allerdings notwendig sein, Elemente der Referenz-Produktmodelle bewusst aus dem Baukastenreferenzmodell auszuschließen. Für den Zusammenhang zwischen Produkt und Baukasten schließt Bursac (2016) auf folgende Gesetzmäßigkeit: Der Baukasten ist nicht in der Lage ein Produkt vollständig abzubilden, allerdings ist die Gesamtheit der Produkte in der Lage den Baukasten abzubilden. Die deduktive Ableitung des Baukastenmodells aus dem entsprechenden Referenzmodell erfolgt durch die Zuordnung der einzelnen Fahrzeug-Komponenten. Zur Veranschaulichung eignet sich das Beispiel eines Setzkastens. Das Baukasten-Referenzmodell beinhaltet die Unterteilung der in den Referenz-Produktmodellen enthaltenen Komponenten. Es stellt somit den Setzkasten selbst und seine Unterteilung in die entsprechenden Fächer bereit. Das Baukastenmodell wiederum wird durch die Einordnung der Komponenten in die entsprechenden Fächer gebildet. Es vereint sowohl die Informationen über die gebildete Struktur als auch die Komponenten. Als Resultat kann die Vielfalt der einzelnen Komponenten zentral über das Baukastenmodell gesteuert werden. Neuentwicklungen von Komponenten finden gezielt im Rahmen der Entwicklung eines Produktes, allerdings unter strenger Beachtung des Baukasten-Regelwerkes statt. Ohne eine kontinuierliche,

parallel stattfindende Baukastenentwicklung sind die grundlegenden Aktivitäten der Baukastenentwicklung folglich nicht realisierbar (Bursac, 2016). Aus dem Baukastenmodell können jederzeit neue Produktmodelle abgeleitet werden. Um diesen Vorgang hinsichtlich der Herausforderungen der Frühen Phase zu unterstützen und eine erhebliche Effizienzsteigerung zu ermöglichen können die relevanten Referenz-Produktmodelle zusätzlich als Wissensbasis herangezogen werden. Die Referenz-Produktmodelle geben dabei einen Bauplan des Produktes<sup>42</sup> vor der mittels der Komponenten des Baukastenmodells individuell an die Projekterfordernisse angepasst werden kann. Rein produktspezifische Anpassungen (z.B. zur optischen Differenzierung) werden über das Regelwerk aus dem Baukasten ausgeschlossen und nicht zurück gespeist.

### **Baukastenarchitekturen und deren Entwicklung in Generationen**

Nach Powelske et al. (2017) können bei der Ableitung eines Produktes und der Entwicklung neuer Komponenten auch mehrere Baukastenmodelle bzw. deren Regelwerke betroffen sein. Hierzu wird auf die Unterscheidung verschiedener Baukastenarchitekturen verwiesen. Entsprechend des vorherrschenden Produktportfolios können die einzelnen Produktgenerationen im Sinne des fraktalen Charakters der Standardisierungsmethoden aus verschiedenen gekoppelten Baukästen abgeleitet werden. Drei Architekturtypen sind dabei denkbar (Abbildung 2.43):

- **Singuläre Architektur:** Ein Baukasten für alle Elemente des Produktportfolios, falls Produkte in Funktions- und Produktstruktur ausreichend hohe Ähnlichkeiten haben.
- **Hybride Architektur:** Ein Baukasten je unterschiedlicher Produktreihen, wobei die Grundstruktur des Baukastens und einzelne Elemente von einem übergeordneten Baukasten übernommen werden.
- **Autarke Architektur:** Eigenständige Baukästen für alle Produktlinien (hauptsächlich, wenn keine funktionale oder bauliche Übereinstimmung existiert).

---

<sup>42</sup> Inklusive detaillierter Zusatzinformationen (Bauraum etc.)

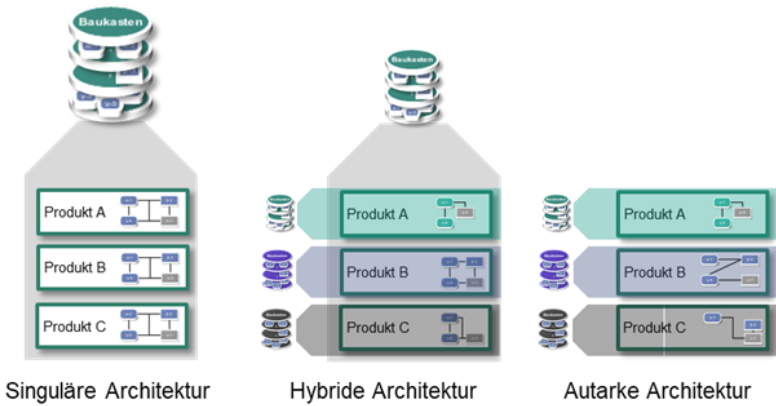


Abbildung 2.43: Übersicht der Baukastenarchitekturen: singuläre, hybride und autarke Architektur (Powelske et al., 2017).

Aufgrund zusätzlicher baukastenübergreifender Abhängigkeiten ergeben sich je nach Architekturtyp zusätzliche Herausforderungen. Voraussetzung hierfür ist es, dass zur Abbildung eines Portfolios funktional oder baulich ähnlicher Produkte mehrere Baukastenmodelle existieren können. Die Ausbildung eines der beschriebenen Architekturtypen beruht auf der Auswahl der zur Baukastensynthese verwendeten Produkte. Deren Auswahl erfolgt vorrangig anhand der Eigenheiten der jeweiligen Produktstrukturen, welche wiederum vom Einfluss bestehender Organisationsstrukturen des betrachteten Unternehmens geprägt sind.

Im Verständnis der PGE lösen sich aufeinanderfolgende Generationen eines Produktes vom Markt ab. Im Sinne des fraktalen Charakters – Module folgen ebenfalls dem PGE Verständnis – lässt sich dieser Gedanke gleichfalls auf Baukästen erweitern. Von Baukastengenerationen lässt sich entsprechend dann sprechen, wenn ein Baukasten  $BK_{gn+1}$  einem Baukasten  $BK_{gn}$  folgt (Powelske et al., 2017). Die Frequenz, in der sich die unterschiedlichen Baukastengenerationen ablösen ist dabei abhängig von den Marktzyklen der abgeleiteten Produkte und kann sich je nach Sparte deutlich unterscheiden.<sup>43</sup> Im Falle der mechatronischen

<sup>43</sup> Während man in der Automobilbranche von Zyklen im Bereich von 4-6 Jahren spricht, sind Halbleiter oder Softwarebaukästen meist nur wenige Monate aktuell.

Steuergeräte sind zusätzlich die unterschiedlichen Entwicklungszyklen der beteiligten Domänen z.B. Halbleiter- oder Softwareentwicklung zu beachten (vgl. Kapitel 2.1.2). Bei der Baukastenentwicklungsentwicklung lassen sich die Übergänge zwischen den einzelnen Baukastengenerationen vorrangig bei autarken Architekturen als solche beobachten. Singuläre oder hybride Architekturen weisen eher kontinuierlich wahrnehmbare Veränderungen auf. Auf Generationswechsel des Baukastens kann im Allgemeinen erst retrospektiv anhand der abgeleiteten Produkte geschlossen werden. Dabei gilt jedoch zu beachten, dass neue aus einem Baukasten abgeleitete Produktgenerationen nicht zwangsläufig auf eine generationsbedingte Weiterentwicklung des Baukastens hindeuten müssen. Eine neue Baukastengeneration ist umgekehrt jedoch immer mit neuen Produktgenerationen verknüpft.<sup>44</sup> Die stärkeren Abhängigkeiten zwischen den Produktgenerationen unterschiedlicher Produktreihen singulärer oder hybrider Architekturen müssen als Abhängigkeiten zwischen den Baukastenreferenzmodellen Berücksichtigung finden. Diese resultieren in zusätzlichen Anforderungen an die Kompatibilität der Komponenten und der Modelle sowie deren Weiterentwicklung über die Generationen hinweg. Bereits ohne die zusätzlichen Beziehungen in Abhängigkeit des Baukastenarchitekturtyps ist die kontinuierliche Wahrung der Kompatibilität zwischen Komponenten und Modellen vielleicht die bedeutendste Herausforderung in der Baukastenentwicklung. Denn nur im Falle einer effizienten Wiederverwendung kann die Baukastenentwicklung ihre Vorteile gegenüber der Entwicklung unabhängiger Produkte ausspielen. Bei komplexen von interdisziplinären Wechselwirkungen bestimmten Produkten, wie mechatronischen Steuergeräten, kann die Kompatibilität allerdings nicht allein durch eine Standardisierung der Schnittstellen sichergestellt werden. Dies liegt an der Tatsache, dass sich dort einzelne Effekte nur aus dem Zusammenwirken der Subsysteme und deren Umwelt – also deren Verhalten (vgl. Kapitel 2.2.1) ergeben. Anschauliche Beispiele verhaltensinduzierter Wechselwirkungen aus dem Feld der mechatronischen Steuergeräte sind die bereits thematisierte elektromagnetische Verträglichkeit oder die temperaturabhängige Stromtragfähigkeit von Elementen der Leistungselektronik (vgl. Kapitel 2.1.2).

---

<sup>44</sup> vgl. hierzu die Feststellung von Bursac (2016), dass der Baukasten nicht vollständig zur Beschreibung eines Produktes ausreicht, die Gesamtheit der abgeleiteten Produkte allerdings den Baukasten abbildet



### 2.4.3 Kompatibilität von Bausteinen

Nachdem in Kapitel 2.4.2 die zentrale Bedeutung der Kompatibilität von Bausteinen für die Baukastenentwicklung hervorgehoben wurde, gibt dieser Abschnitt einen Überblick der verschiedenen Facetten von Kompatibilität, die im Rahmen der vorliegenden Arbeit Berücksichtigung finden. Daran anknüpfend wird ein Ansatz zur Kompatibilitätsbeschreibung vorgestellt, der auf der Wiederverwendbarkeit von Wissen aus Referenzprodukten aufbaut.

Der Begriff der Kompatibilität begegnet uns in seinen verschiedensten Formen tagtäglich. Über 34.000 Normen und Standards regeln wie die Kompatibilität von Elementen in den uns umgebenden technischen, sozio-technischen und sozialen Systemen gewährleistet werden kann. Eine allgemeine Definition der Kompatibilität beschreibt diese als Verträglichkeit von Objekten und Sachverhalten (Lackes & Siepermann, 2018). Inkompatibilitäten, also das Fehlen dieser Verträglichkeit bedeutet zusätzliche Aufwände, Effizienzverluste oder resultiert beispielsweise in Konflikten mit dem Gesetz<sup>45</sup>. Wendet man sich ingenieurtechnischen Definitionen zu, so wird ersichtlich, dass sich die unterschiedlichen Domänen (bspw. Mechanik, Elektrotechnik und Software) dem Kompatibilitätsbegriff jeweils kontextspezifisch nähern. Während in der Konstruktion häufig Designaspekte eine große Rolle spielen, beschäftigt sich die Softwareentwicklung intensiv mit generationenübergreifender Kompatibilität von Programmen und Algorithmen.<sup>46</sup> Beim Zusammentreffen von Hardware- und Softwarekomponenten definiert sich die Kompatibilität auch über die Möglichkeit verschiedene Komponenten (auch unterschiedlicher Hersteller) zusammen oder aufeinander abgestimmt zu benutzen (Lackes & Siepermann, 2018). Mit zunehmender Interdisziplinarität steigen dabei die Aufwände in der Gewährleistung von Kompatibilität der Systemelemente enorm. Aus der Diskussion der komplexen Wechselwirkungen mechatronischer Steuergeräte wissen wir bereits, dass eine verlässliche Bewertung von Kompatibilität nur auf dem Vergleich von gewünschtem und tatsächlichem Verhalten erfolgen kann. (vgl. Kapitel 2.1.2) Um das Systemverhalten in der Entwicklung neuer Produkte zuverlässig und effizient vorhersagen zu können, bedient man sich MBSE-gestützten Simulationsmodellen oder expliziertem Expertenwissen. Die Anwendung des Produktmodellierungs-Frameworks zeigt, dass dieses Wissen in Form von

---

<sup>45</sup> Unterschiede zwischen metrischem und imperialem System, verschiedene Spannungslevel der internationalen Stromnetze

<sup>46</sup> Je nach Richtung, in der die Verträglichkeit gegeben ist, spricht man hier von Aufwärts- oder Abwärtskompatibilität

Referenz-Produktmodellen für die Entwicklung folgender Produktgenerationen nutzbar gemacht werden kann (vgl. Kapitel 2.4.2). Damit Aussagen zur Kompatibilität von Systemelementen wiederverwendbar gemacht werden können, bedarf es nachvollziehbarer Methoden zur Kompatibilitätsbewertung. Anhand des Zusammenhangs von Ziel- und Objektsystem im ZHO-Modell wird deutlich, dass die Kompatibilität zwischen Systemelementen stets am resultierenden Erfüllungsgrad von Systemanforderungen und übergeordneten Produkt- oder Unternehmenszielen bewertet werden sollte. Der Nachweis eben dieser Kompatibilität zu den Elementen des Zielsystems<sup>47</sup> ist heute als Teil einer Vielzahl rechtlich verbindlicher Normen wie bspw. der ISO26262 oder ASPICE zu einem der zentralen Themen im Systems Engineering aufgestiegen. Dass die unterschiedlichen Ziele an ein Produkt ebenfalls auf Kompatibilität zu prüfen sind, wird aus einer Definition im Bereich der Wirtschaftswissenschaften ersichtlich. Diese fasst den Kompatibilitätsbegriff als Vereinbarkeit unterschiedlicher Zielstellungen auf (Lackes & Siepermann, 2018). In Bezug auf die multiplen Zielstellungen, denen moderne Produkte unterworfen sind erweist sich die Bewertung von Kompatibilität als äußerst kontextsensitiv. Abhängig davon, welche an das System angelegte Zielstellung betrachtet wird, kann die bewertete Kompatibilität der Systemelemente unterschiedlich ausfallen. Beispielhaft sei der Einsatz von Nutzwertanalysen im SPALTEN-Schritt der Lösungsauswahl angeführt. Die zunehmende Funktionsdichte innerhalb mechatronischer Steuergeräte und der damit verbundene Anstieg an Produktanforderungen gilt als weiterer Komplexitätstreiber in der Ableitung von Baukastenelementen. Diese sogenannte Unschärfe eines logischen Problems wird anhand reeller Zahlen zwischen 0 und 1 abgebildet. In der Interpretation entsprechen 0 und 1 dem klassischen Verständnis von wahr und falsch, die dazwischenliegenden Werte beschreiben teilwahre Zustände. Diese Zwischenwerte spielen vor allem in der Beschreibung realer Prozesse, also im Bereich der angewandten Wissenschaften eine große Rolle. Beispielsweise lassen sich steigende oder fallende Flanken in der Regelungstechnik nur schwerlich mittels boolescher Logik abbilden. Unter Betrachtung teilwahrer Zustände kann zusätzlich wertvolle Information transferiert werden. Böhme (1993) illustriert dies, indem er die Fuzzy-Logik zur Veranschaulichung des Ladezustandes einer Batterie verwendet. Wird dieser logisch mit einem Wert von 0,9 beschrieben, ist der Zustand voll zu 90% „wahr“. Im Ansatz von Ishii et al. (1988) werden zunächst qualitative Kompatibilitätsbewertungen durch Übertragung auf den Wertebereich 0 bis 1, quantifizierbar dargestellt.

---

<sup>47</sup> vgl. Kapitel 2.2.3 Begriff der Traceability

Auf Basis der quantifizierten Kompatibilitätsbewertungen einzelner Systemelemente lässt sich die resultierende Systemkompatibilität, der sogenannte Match Index, folgender Formel entsprechend berechnen.

$$MI = \sum_k utility(s) \cdot M(s), \quad s \in K \quad (5)$$

Der Match Index MI eines Systems mit K Elementen bildet sich als gewichtete ( $utility(s)$ ) Summe von Match Koeffizienten  $M(s)$ . Der Match Koeffizient entspricht dabei einer Kennzahl für die Kompatibilität des betrachteten Elementes  $s$  innerhalb einer vorgegebenen Systemkonfiguration unter Beachtung der Systemanforderungen. Die Bestimmung des Match Koeffizienten beinhaltet die Konvertierung einer initial vorgenommenen quantitativen Bewertung bereits bekannter Komponentenpaarungen. Die quantitative Bewertung erfolgt von Experten anhand der Adjektive (exzellent, sehr gut, gut, gering, schlecht, sehr schlecht). Dabei müssen sowohl zeitlich unveränderliche als auch dynamische, vom Verhalten geprägte Eigenschaften berücksichtigt werden. Mittels konsequenter Anwendung wird das gesammelte Erfahrungswissen aus Referenzprodukten erfasst und in Form einer Wissensdatenbank zur Wiederverwendung abgelegt. Die anschließende Konvertierung der adjektivischen Bewertung erfolgt entsprechend der Fuzzy-Logik in die Werte (1; 0,8; 0,6; 0,4; 0,2; 0). Da gewissen Systemelementen erfahrungsgemäß ein größerer Beitrag zur Erreichung der Systemziele zukommt, werden die einzelnen Match Koeffizienten über die  $utility(s)$ -Funktion entsprechend gewichtet. Für die Gewichtungsfunktion gilt dabei:

$$\sum utility(s) = 1 \quad (6)$$

Als Resultat kann der Match-index MI nur Werte zwischen 0 und 1 annehmen, wobei 1 der vollständigen Kompatibilität und 0 der völligen Inkompatibilität aller betrachteten Systemelemente entspricht. Die Gewichtung mittels  $utility(s)$ -Funktion birgt allerdings die Gefahr, dass ein hoher allgemeiner MI erreicht werden kann, obwohl das System eine Vielzahl an Inkompatibilitäten gering gewichteter Elemente beherbergt. Ishii et al. (1988) führen daher mit der Match Range (MR) eine weitere Größe zur Aufdeckung möglicher Dysbalancen ein.

$$MR = [\min \{M(s) \mid s \in K\}, \max \{M(s) \mid s \in K\}] \quad (7)$$

Die Match Range bildet ein geschlossenes Intervall, welches auf Basis des geringsten und des höchsten Match Koeffizienten dessen Abweichung in der Berechnung des Match Index widerspiegelt. Zur verlässlichen Aussage über die Kompatibilität sind beide Größen MI und MR zu betrachten. Besonders kompatible Lösungen zeichnen sich als Folge durch einen möglichst hohen MI und eine möglichst kleines Intervall MR aus. Als Richtwert bezüglich der potentiellen Eignung einer bewerteten Systemkonfiguration für die Entwicklung des betrachteten Produktes gilt ein  $MI \geq 0,5$ . Komponentenpaarungen, zu denen keine Experteneinschätzungen vorliegen werden, automatisch mit einem MI von 0,5 belegt, da eine mögliche Eignung nur auf Basis weiterer Untersuchungen überprüft werden kann.

#### 2.4.4 Fazit

Durch die zunehmende Individualisierung von Produkten bietet die Baukastenentwicklung, die als bereits etabliertes Mittel zur Reduktion der unternehmensinternen Vielfalt gilt, bis heute Basis unzähliger Forschungsarbeiten. Die in der Literatur ausgiebig behandelte Aktivität der Modularisierung stellt dabei nur einen Teil der Baukastenentwicklung dar. In der industriellen Praxis machen es der mitunter große Zeitversatz und die hohen Abhängigkeiten notwendig die Baukastenentwicklung als kontinuierliche, den Produktentwicklungszyklen parallel geschaltete, Aktivität zu verstehen. Bestehende Ansätze im Bereich der MBSE-gestützten Baukastenentwicklung bieten gerade in der Frühen Phase eine bedeutende Unterstützung. Im Bereich der mechatronischen Steuergeräte muss zusätzlich den kurzen Entwicklungszyklen der beteiligten Domänen, als auch steigender Integrationsvorhaben innerhalb modularer Produktstrukturen, Rechnung getragen werden. Es ergeben sich neue Herausforderungen an eine stetige, generationenübergreifende Weiterentwicklung des Baukastens, der selbst verschiedenen Architekturtypen folgen kann. Daraus resultierende Abhängigkeiten erschweren den Prozess der Kompatibilitätswahrung von Bausteinen zusätzlich. Dieser kann im Umfeld komplexer mechatronischer Produkte nicht durch bloße Standardisierung von Schnittstellen, sondern nur auf einer verhaltensbasierten Betrachtung der Modulzusammenhänge erfolgen. Für die Strukturierung und Anwendung von Baukästen scheint es daher sinnvoll bereits vorhandene Informationen über die Modulkompatibilität in die Referenz-Produktmodelle einfließen zu lassen und diese computergestützt für Induktions- und Deduktionsschritte verfügbar zu machen.

## 3 Zielsetzung und Vorgehensweise

In diesem Kapitel werden die Zielsetzung der Arbeit, sowie die methodische Vorgehensweise zur Zielerreichung vorgestellt. Dazu werden zunächst die Erkenntnisse des dargelegten Standes der Forschung kondensiert und daran der Forschungsbedarf abgeleitet. Daraufhin erfolgt die Formulierung der übergeordneten Zielstellung, sowie die Darstellung der dabei zugrundeliegenden Forschungshypothesen. Die anschließend vorgestellten Forschungsfragen dienen der Operationalisierung der Zielerreichung (Kapitel 3.1). Im folgenden Teilkapitel wird die systematische Vorgehensweise zur Methodenentwicklung erörtert und die, zur Bestätigung des Forschungsbedarfes in der industriellen Praxis herangezogenen empirischen Methoden, präsentiert (Kapitel 3.2). Abschließend erfolgt die Vorstellung der Untersuchungsumgebung, anhand derer große Teile des Erkenntnisgewinns dieser Arbeit erzielt und die entwickelten Ansätze validiert wurden (Kapitel 3.3).

### 3.1 Zielsetzung

Die Zielsetzung der vorliegenden Arbeit wird zunächst anhand der Diskussion des Forschungsbedarfes abgeleitet. Im Anschluss erfolgt deren Fokussierung mittels Hypothesen und eine Operationalisierung der Zielstellung anhand geeigneter Forschungsfragen.

#### 3.1.1 Forschungsbedarf

Aus der Betrachtung der Grundlagen und des Standes der Forschung erweisen sich die **mechatronischen Steuergeräte** als Vertreter einer spezifischen Klasse mechatronischer Systeme, die auf der **räumlichen Integration** mechanischer und elektronischer Bauteile beruhen, deren **mechanischer Anteil** fast ausschließlich im Bereich der **Aufbau- und Verbindungstechnik** liegt (vgl. Kapitel 2.1.1). Der hohe Anteil informationsverarbeitender Aufgaben äußert sich in einer **Dominanz funktionsorientierter Denkweisen** aus den Domänen der Software- und Hardwareentwicklung. Die **kurzen Entwicklungszyklen** der Halbleiterindustrie führen zu einer raschen Zunahme an baulicher und funktionaler Varianz (vgl. Kapitel 2.1.2). In Zusammenhang mit dem Trend der Hochintegration bilden sich als Folge **Mischformen modularer und integraler Produktstrukturanteile** aus, die eine Vergleichbarkeit der ohnehin vielfältigen Steuergerätearchitektur erschweren (vgl.

Kapitel 2.1.2; Kapitel 2.2.4). Die steigende Anzahl an Integrationsvorhaben zur Bauraumersparnis resultiert in einer Verstärkung **interdisziplinärer Wechselwirkungen** (vgl. Kapitel 2.1.2). Daraus ergibt sich besonders im Bereich der Aufbau- und Verbindungstechnik (AVT) ein stetiger Bedarf an technologischer Innovation. Die AVT selbst und damit die Integration der Funktionsträger ins Steuergerät, wird, trotz ihrer zunehmend ausgeprägten Rolle als zentraler, das Produktkonzept beeinflussender Kostenfaktor, im Entwicklungsprozess häufig nachgelagert, betrachtet. Gründe dafür liegen in der bereits thematisierten **Pfadabhängigkeit im Entwicklungsprozess mechatronischer Steuergeräte** (vgl. Kapitel 2.1.2). Verstärkt wird dieser Mechanismus durch die Anwendung sequenzieller Prozessmodelle - bspw. des V-Modells - welches trotz bekannter Kritik an mangelnder Iterativität, in der Praxis noch immer weit verbreitet ist. Im Automobilbereich sind auf dieser Basis, für die Entwicklung mechatronischer Steuergeräte, verbindliche Standards wie ASPICE und die ISO26262 entstanden (vgl. Kapitel 2.2.3). Durch die zunehmende Automatisierung sicherheitskritischer Funktionen, spielen diese Standards eine immer entscheidendere Rolle im interdisziplinären Entwicklungsprozess. Die geforderte **Durchgängigkeit in der Dokumentation** der Systementwicklungsartefakte erweist sich als zusätzliche Herausforderung für den Entwickler (vgl. Kapitel 2.1.2, Kapitel 2.2.1). Im Entwicklungsprozess mechatronischer Steuergeräte besteht aufgrund der dargelegten Spezifika der dringende Bedarf **ganzheitlicher Ansätze**, die auf **iterativen Prozessmodellen** wie dem iPeM beruhen (vgl. Kapitel 2.2.3).

Für diesen Einsatzzweck wurde bereits die Eignung der **Modellbasierten Systementwicklung** nachgewiesen (vgl. Kapitel 2.2). Mittels der dort propagierten, konsistenten Modelle können komplexe Wechselwirkungen allgemeinverständlich abgebildet und eine durchgängige Dokumentation der Entwicklungsartefakte gewährleistet werden (vgl. Kapitel 2.2.2). Dies führt zu einer **Unterstützung des ganzheitlichen Denkprozesses** (vgl. Kapitel 2.2). Aufgrund der zunehmenden Individualisierung der Produkte und der Vielzahl neuer Einsatzmöglichkeiten für elektrische Kleinantriebe im Automobilbereich ist eine steigende Variantenzahl mechatronischer Steuergeräte zu beobachten (vgl. Kapitel 2.1.2). Der Einsatz ganzheitlicher Ansätze zur Adressierung der spezifischen Charakteristika mechatronischer Steuergeräte konnte daher im Bereich der Standardisierungsmethoden als besonders vielversprechend identifiziert werden (vgl. Kapitel 2.4.2).

Im Kanon der Standardisierungsmethoden gilt die **Baukastenentwicklung** als anerkannte Maßnahme zur **Vergrößerung der externen Variantenvielfalt** bei gleichzeitiger **Reduktion der internen Teilevielfalt** (vgl. Kapitel 2.4.1). Umfassende Forschungsergebnisse existieren hier vor allem im Teilbereich der

Modularisierungsmethoden (vgl. Kapitel 2.4.1). Bei ganzheitlicher Betrachtung der Baukastenentwicklung erweisen sich deren Aktivitäten als in hohem Maße iterativ. Dementsprechend wurde die Eignung sequenzieller Modelle zu deren Beschreibung widerlegt (vgl. Kapitel 2.4.2). Im Kontext der PGE-Produktgenerationsentwicklung ist die Baukastenentwicklung daher als **kontinuierliche, den Produktentwicklungszyklus begleitende, Aktivität** zu betrachten (vgl. Kapitel 2.4.2, Kapitel 2.3.1). Im Bereich der Frühen Phase und der Serienentwicklung von Hybridtriebstrangsystemen existieren bereits Methoden, die dieser Sichtweise entsprechen. Bei diesen Ansätzen erfährt der Entwickler durch den Einsatz MBSE-gestützter Methoden Unterstützung in Form eines iterativen Vorgehens zur Baukastenentwicklung (vgl. Kapitel 2.4.2). Im Ansatz zur Anwendung innerhalb der Frühen Phase zeigt sich, dass die **bewusste Nutzung von (Referenz-)Modellen** entsprechend der Erkenntnisse der PGE, einen erfolgreichen Umgang mit der Komplexität innerhalb der Baukastenentwicklung ermöglichen (Albers, Scherer, Bursac & Rachenkova, 2015). Der daraus entstandene Makroprozess beschreibt die Mechanismen zur induktiven und deduktiven Überführung von Produkt- und Baukastenmodellen (vgl. Kapitel 2.4.2). Verfolgt man diesen Prozess, so ergeben sich aus den Spezifika der Entwicklung mechatronischer Steuergeräte neue Forschungsbedarfe.

Der Makroprozess beschreibt, dass eine Vielzahl an Informationen in die Referenzmodellbildung miteinbezogen werden können. Hinsichtlich der negativ-emergenten Wechselwirkungen kommt den **Mechanismen zur Kompatibilitätswahrung** eine besondere Rolle zu (vgl. Kapitel 2.4.3). Diese kann nicht allein durch die Ableitung standardisierter Schnittstellen abgedeckt werden und bedarf daher eigener Ansätze zur effizienten Integration in den Kreislauf aus Induktion und Deduktion. Vor allem im Bereich der deduktiven Ableitung von Produktmodellen, also der aktiven Anwendung eines Baukastenmodells ist eine **systematische Betrachtung der Kompatibilität** von Bausteinen erforderlich (vgl. Kapitel 2.4.3).

Die zahlreichen Ansätze im Bereich der Modularisierung geben zwar Aufschluss über die Modellierung komplexer Produktmodelle, sind aber aufgrund der häufig isolierten Betrachtung der Induktion hinsichtlich ihrer Wiederverwendbarkeit in Form von Referenzmodellen nur bedingt nutzbar (vgl. Kapitel 2.4.1, Kapitel 2.4.2). Fragen zum Umgang mit den Mischformen modularer und integraler Produktstrukturanteile innerhalb der Referenzmodellbildung bieten dementsprechend weiteren Nährboden wissenschaftlicher Auseinandersetzung (vgl. Kapitel 2.2.4). Die kurzen Entwicklungszyklen der Halbleiterindustrie gepaart mit den verschiedenen Baukastenarchitekturen zur Abbildung des Produktportfolios machen eine kontinuierliche Anpassung des Baukastenmodells in kurzen Intervallen notwendig.

Hier bietet wiederum die PGE eine geeignete Grundlage die **Entwicklung des Baukastens in Generationen** zu betrachten (vgl. Kapitel 2.4.2). Durch die Erweiterung der PGE hinsichtlich der Referenzsysteme ist eine Nutzung von Entwicklungsartefakten aus den Phasen der Forschung und Entwicklung gerade in der Frühen Phase ein vielversprechender Ansatz für weitere Forschung (vgl. Kapitel 2.3.2).

Während sich die Ansätze auf dem Gebiet der MBSE nachweislich zur ganzheitlichen Lösung komplexer Problemstellungen mechatronischer Produkte und der Unterstützung der Baukastenentwicklung hervorgerufen haben, besteht weiterhin **kein flächendeckender Einsatz** im industriellen Umfeld (vgl. Kapitel 2.2.7). Mit der SysML existiert hier bereits ein, trotz abstrakter Charakteristik, anerkannter Sprachstandard (vgl. Kapitel 2.2.6). Ausschlaggebend für die mangelnde Akzeptanz bleiben nach wie vor die **zu generischen Methoden und der hohe Modellierungsaufwand** bei deren Anwendung (vgl. Kapitel 2.2.7). Aus diversen Fallstudien wird allerdings ersichtlich, dass mittels sinnvoller Teil-Automatisierung in der Erstellung und Auswertung SysML-basierter Modelle, weitere Schritte in Richtung Akzeptanz getan werden können (vgl. Kapitel 2.2.7). Für die Baukastenentwicklung bieten sich gerade in der Frühen Phase Potentiale in der **Automatisierung der Baukastenentwicklung**.

Basierend auf dem dargelegten Forschungsbedarf lässt sich für die vorliegende Arbeit folgendes Ziel formulieren:

### Ziel der Arbeit

Im Rahmen dieser Arbeit wird das Ziel verfolgt, die Strukturierung und Anwendung von Baukästen in der Frühen Phase der PGE – Produktgenerationsentwicklung von mechatronischen Steuergeräten zu unterstützen, die Akzeptanz der Methodenanwendung des MBSE zu steigern und eine Zeitreduktion bis zur initialen Konzeptauswahl in der Frühen Phase der PGE zu erwirken.

### 3.1.2 Forschungshypothesen

Die zuvor dargelegte Zielstellung gründet auf Forschungshypothesen. Hierbei handelt es sich um zentrale Annahmen des Forschungsvorhabens, welche als Basis der folgenden Untersuchungen dienen und in deren Verlauf verifiziert bzw. falsifiziert werden.



**Forschungshypothese 1:**

Die einzigartigen Herausforderungen im Produktentstehungsprozess mechatronischer Steuergeräte erfordern hinsichtlich der Baukastenentwicklung flexible und auf die Kompatibilität der Bausteine ausgerichtete, ganzheitliche Methoden zur Strukturierung und Anwendung.

**Forschungshypothese 2:**

Mittels spezifischen Einsatzes des MBSE kann die Kompatibilität der Baukastenelemente über die Prozesse von Induktion und Deduktion bei der Strukturierung und Anwendung von Baukästen mechatronischer Steuergeräte gewährleistet werden.

**Forschungshypothese 3:**

Durch (teil-)automatisierte, MBSE-gestützte Methoden zur Baukastenstrukturierung und -anwendung kann die Frühe Phase der PGE mechatronischer Steuergeräte effizient unterstützt werden, dies bedeutet, die Zeit zwischen der Produkthanfrage durch den Kunden bis zum ersten Produktkonzept deutlich zu verringern.

## 3.2 Forschungsfragen

Entsprechend des Forschungsbedarfes und der zugrundeliegenden Hypothesen wird die Erreichung der abgeleiteten Zielstellung der vorliegenden Arbeit, anhand folgender Forschungsfragen konkretisiert:

**Forschungsfrage 1:**

*FF1: Welche Spezifika und Bedarfe im Bereich der Strukturierung und Anwendung von Baukästen existieren in der Frühen Phase der PGE von mechatronischen Steuergeräten und welches Zielsystem an entsprechende Methoden lässt sich daraus ableiten?*

### **Forschungsfrage 2:**

*FF2: Wie sind die Strukturierung und Anwendung eines Baukastens mechatronischer Steuergeräte auf Basis des ermittelten Zielsystems, unter Verwendung der Methoden des MBSE zu gestalten und zu integrieren?*

### **Forschungsfrage 3:**

*FF3: Welchen Beitrag leisten die Methoden zur Baukastenstrukturierung und -anwendung in der Frühen Phase der PGE von mechatronischen Steuergeräten zur Unterstützung der beteiligten Domänen und zur Steigerung der Effizienz im Entwicklungsprozess?*

Die methodische Vorgehensweise zur Beantwortung der Forschungsfragen, wird im folgenden Kapitel dargelegt.

## **3.3 Forschungsmethode und Vorgehensweise**

Im vorliegenden Unterkapitel wird zunächst die methodische Vorgehensweise unter Verwendung der Design Research Methodology (DRM) nach Blessing & Chakrabarti (2009) vorgestellt. Im Anschluss werden, die der vorliegenden Arbeit zugrunde gelegten, empirischen Methoden beleuchtet.

### **3.3.1 Forschungsmethode**

Entsprechend des zunehmenden Einflusses der Mechatronik in der Produktentwicklung erweist sich die Interdisziplinarität als Charakteristikum moderner ingenieurtechnischer Forschungsvorhaben. Weiterhin sind diese Vorhaben durch das iterative Durchlaufen von Analyse- und Syntheseschritten gekennzeichnet.

Nur wenige der bestehenden, strukturierten Vorgehensweisen zur Begleitung wissenschaftlicher Forschung bieten die Möglichkeit diese Punkte entsprechend zu adressieren und eignen sich daher für die Strukturierung technikorientierter Forschungsarbeiten. Für die Strukturierung der vorliegenden Arbeit wurde der von Blessing & Charkrabarti entwickelte Ansatz der Design Research Methodology (DRM) ausgewählt. Das dort präsentierte Framework ist aus dem Kanon der Vorgehensweisen am ehesten in der Lage, die Zielerreichung des dargelegten Forschungsvorhabens zu unterstützen.

Abbildung 3.1 zeigt das in Bezug auf die Arbeit angewendete Framework. Im Zentrum des Frameworks stehen die einzelnen Phasen der DRM inklusive deren Ausprägung als „Literatur-basiert“, „umfassend“ oder „initial“. Weiterhin sind die innerhalb der Phasen angewendeten Methoden, daraus erzielte Ergebnisse und deren Abhandlung in den Kapiteln der vorliegenden Arbeit dargestellt.

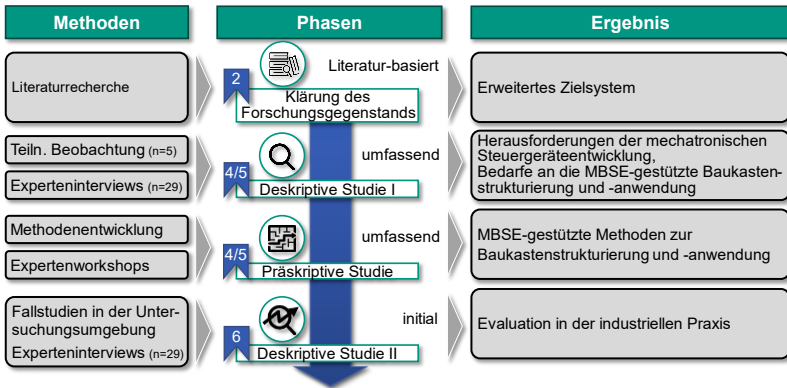


Abbildung 3.1: Forschungsvorgehen (genutzte Methoden, Phasen und jeweiliges Ergebnis) der vorliegenden Arbeit entsprechend der DRM nach Blessing & Chakrabarti (2009).

Basierend auf der jeweiligen Ausprägung, mit der die einzelnen Phasen durchgeführt werden, identifiziert die DRM sieben verschiedene Typen von Forschungsvorhaben (Tabelle 3.1).

**Tabelle 3.1: Arten technischer Forschungsprojekte und deren Schwerpunkte (Blessing & Chakrabarti, 2009, S. 18)**

Nr.	Klärung des Forschungsgegenstandes	Deskriptive Studie I	Präskriptive Studie	Deskriptive Studie II
1	literatur-basiert	umfassend		
2	literatur-basiert	umfassend	initial	
3	literatur-basiert	literatur-basiert	umfassend	initial
4	literatur-basiert	literatur-basiert	literatur-basiert initial/umfassend	umfassend
5	literatur-basiert	umfassend	umfassend	initial
6	literatur-basiert	literatur-basiert	umfassend	umfassend
7	literatur-basiert	umfassend	umfassend	umfassend

Wie Tabelle 3.1 zu entnehmen ist orientiert sich das dargelegte Forschungsvorhaben an der fünften Kategorie. Die Verortung und Abhandlung der Phasen in den Kapiteln der vorliegenden Arbeit wird im Folgenden umrissen:

Im Rahmen von Kapitel 2 wurde eine Literaturrecherche zur **Klärung des Forschungsgegenstandes** durchgeführt. Das Ergebnis dieser Aktivität mündet in die Formulierung des Forschungsbedarfes und der darauf gestützten Ableitung des erweiterten Zielsystems in Kapitel 3 der vorliegenden Arbeit. Die abgeleiteten Forschungshypothesen und Forschungsfragen dienen der Operationalisierung der entwickelten Zielstellung: die Strukturierung und Anwendung von Baukästen in der Frühen Phase der PGE – Produktgenerationsentwicklung mechatronischer Steuergeräte mittels gezielter Einsatzes von MBSE Methoden zu unterstützen.

Für die Beantwortung der Forschungsfragen werden die Aspekte der Baukastenstrukturierung und -anwendung über die folgenden Phasen der DRM sequenziell beleuchtet. Dies geschieht mit Hinblick auf die Tatsache, dass die Baukastenanwendung ein bereits strukturiertes Baukastenmodell voraussetzt. Kapitel 4 widmet sich dementsprechend der Methode zur Baukastenstrukturierung, während sich Kapitel 5 mit der Methode der Baukastenanwendung befasst. Die Vorgehensweise zur Betrachtung der einzelnen Aspekte und damit der Aufbau der zugrundeliegenden Studien erfolgt für beide Methoden analog. Dies äußert sich in einem vergleichbaren Aufbau der Unterkapitel: Zunächst werden im Rahmen der **deskriptiven Studie I** die jeweiligen Elemente des Zielsystems der zu

entwickelnden Methoden empirisch ermittelt (Kapitel 4.1 bzw. Kapitel 5.1). Für die spätere Validierung werden diese entsprechend der DRM in Schlüsselfaktoren und Erfolgsfaktoren aufgeteilt und deren Zusammenhänge in einem Reference Model illustriert. Anschließend erfolgt die Entwicklung der Methoden auf Basis des ermittelten Zielsystems innerhalb einer zweigeteilten **präskriptiven Studie**. Als Erstes wird dazu der theoretische Aufbau der Methoden in Kapitel 4.2 bzw. 5.2 hergeleitet. Im Anschluss findet in den Kapiteln 4.3 bzw. 5.3 die Implementierung der jeweiligen Methoden innerhalb der Rahmenbedingungen der Untersuchungsumgebung statt. Anhand der Ergebnisse wird dargelegt, welche Unterstützung der Anwender im Prozess der Baukastenentwicklung für mechatronische Steuergeräte durch die Methoden erfährt. Die Validierung der entwickelten Methoden wird im Rahmen der als initial ausgeprägten **deskriptiven Studie II** in Kapitel 6 vorgestellt. Im Bereich der Methodenentwicklung bedarf es nach Blessing & Chakrabarti (2009) besonderen Maßnahmen, um eine Bewertung der tatsächlichen Effekte der Unterstützungsansätze zu ermöglichen. Hierzu werden die in der DS I ermittelten Bedarfe in Form von Schlüsselfaktoren und messbaren Erfolgskriterien mit den, über die PS bereitgestellten Unterstützungsansätzen verknüpft. Dies geschieht mittels sogenannter Impact Models für die Baukastenstrukturierung und -anwendung. Anschließend werden die einzelnen Validierungsschritte zur Anwendungsevaluation und Erfolgsbewertung vorgestellt. Der tatsächliche Beitrag der entwickelten Methoden wird anhand der Ergebnisse der durchgeführten Validierungsmaßnahmen abschließend ausgewiesen.

### 3.3.2 Empirische Methoden

Die im Rahmen der DRM vorgenommene Strukturierung des Forschungsprojektes sieht im Anschluss an die Literaturrecherche eine Bestätigung des abgeleiteten Forschungsbedarfes vor. Innerhalb der vorliegenden Arbeit wurde hierzu eine umfassende deskriptive Studie I durchgeführt. Deren konkrete Forschungsaktivitäten wurden mittels Nutzung spezifischer, empirischer Methoden unterstützt. Die gewählten empirischen Methoden finden ebenfalls in der deskriptiven Studie II zur Validierung des Ansatzes Anwendung. In puncto Auswahl konkreter Forschungsmethoden, bspw. zur Gestaltung der Forschungsaktivitäten deskriptiver Studien, ist der Forscher seitens der DRM auf sich allein gestellt. Unterstützung findet er allerdings in dem von Marxen (2011) abgeleiteten Rahmenwerk, welches auf Basis des iPeM unterschiedliche Methoden vorstellt und für deren Einsatz kategorisiert. Die dem Kanon von Marxen zur Bearbeitung der vorliegenden Arbeit entnommenen Forschungsmethoden werden im Folgenden in Form von Methodensteckbriefen vorgestellt (Abbildungen 3.2 – 3.5). Diese umfassen neben dem vorgesehenen Anwendungsfeld, die jeweiligen Vor- und Nachteile in der Umsetzung.

(Teil-) strukturiertes Interview	
<b>Anwendungsfeld:</b> <ul style="list-style-type: none"><li>➤ Explorative Forschung mit dem Ziel, Forschungsbedarf zu identifizieren</li><li>➤ Empirische Forschung, Analyse von realen Entwicklungsprozessen</li><li>➤ Anwendungsstudien, Einsatz von Entwicklungsunterstützung in realen Prozessen</li></ul>	
<b>Vorteile:</b> <ul style="list-style-type: none"><li>➤ Direkte Datensammlung, da der Beobachter Daten aufnimmt</li><li>➤ Zugänglichkeit von Erfahrungswissen</li><li>➤ individuelle thematische Präferenz der Befragten kann beim (teil-) strukturierten Interview aufgenommen werden</li></ul>	<b>Nachteile:</b> <ul style="list-style-type: none"><li>➤ wissenschaftliche Akzeptanz</li><li>➤ die Befragten können ungenaue oder unvollständige Daten liefern</li><li>➤ Fragen müssen präzise und eindeutig formuliert werden</li></ul>

Abbildung 3.2: Methodensteckbrief (teil-)strukturiertes Interview nach Marxen (2011).

Retrospektives Protokoll	
<b>Anwendungsfeld:</b> <ul style="list-style-type: none"><li>➤ Empirische Forschung, Analyse von realen Entwicklungsprozessen</li><li>➤ Menschzentrierte Forschung an der Entwicklungsmethodik</li><li>➤ Wenn Zusammenfassung gegenüber detaillierter Beschreibung bevorzugt</li></ul>	
<b>Vorteile:</b> <ul style="list-style-type: none"><li>➤ Entwicklungsaktivitäten bleiben durch Beobachter unberührt</li><li>➤ Zugänglichkeit von realen Prozessen</li><li>➤ Daten sind bereits zusammengefasst wenn sie gesammelt werden</li></ul>	<b>Nachteile:</b> <ul style="list-style-type: none"><li>➤ produziert nicht viele Daten, sondern eine zusammengefasste Version</li><li>➤ indirekte Datensammlung</li><li>➤ die Befragten können ungenaue oder unvollständige Daten liefern</li></ul>

Abbildung 3.3: Methodensteckbrief retrospektives Protokoll nach Marxen (2011)

Teilnehmende Beobachtung	
<p><b>Anwendungsfeld:</b></p> <ul style="list-style-type: none"> <li>➤ Empirische Forschung, Analyse von realen Entwicklungsprozessen</li> <li>➤ Anwendungsstudien, Einsatz von Entwicklungsunterstützung in realen Prozessen</li> <li>➤ Langzeitprojekte, welche Einblick in Entwicklungsaktivitäten und Prozesse ermöglichen, indem die Datenquelle eine soziale Gruppe ist</li> </ul>	
<p><b>Vorteile:</b></p> <ul style="list-style-type: none"> <li>➤ Direkte Datensammlung, da der Beobachter ein Teil der Datenquelle ist</li> </ul>	<p><b>Nachteile:</b></p> <ul style="list-style-type: none"> <li>➤ Beobachtung kann die Entwickler von den Aktivitäten ablenken.</li> <li>➤ Abläufe entsprechen nicht der Realität, da die Beobachtungssituation bekannt ist</li> </ul>

Abbildung 3.4: Methodensteckbrief teilnehmende Beobachtung nach Marxen (2011).

Fallstudie	
<p><b>Anwendungsfeld:</b></p> <ul style="list-style-type: none"> <li>➤ Explorative Forschung mit dem Ziel die Forschungsfrage zu identifizieren</li> <li>➤ Empirische Forschung, Analyse von realen Entwicklungsprozessen</li> <li>➤ Untersuchung von komplexen Situationen, wenn das Ziel ein ganzheitliches Bild ist</li> <li>➤ Identifikation von Hypothesen &amp; Falsifizierung von Theorien</li> <li>➤ Zeigt Anwendbarkeit / Nutzen einer Entwicklungsunterstützung</li> </ul>	
<p><b>Vorteile:</b></p> <ul style="list-style-type: none"> <li>➤ Ganzheitlicher Ansatz</li> <li>➤ Funktioniert auch in komplexen Situationen</li> </ul>	<p><b>Nachteile:</b></p> <ul style="list-style-type: none"> <li>➤ Wissenschaftliche Akzeptanz</li> <li>➤ Aufwand, durch die notwendige Anwendung von verschiedenen Forschungsmethoden</li> </ul>

Abbildung 3.5: Methodensteckbrief Fallstudie nach Marxen (2011)

## **3.4 Forschungsumgebung**

Das folgende Kapitel widmet sich der Vorstellung der Forschungsumgebung innerhalb derer die Erkenntnisse der vorliegenden Arbeit geschaffen wurden. Der Fokus der Vorstellung liegt dabei auf den strukturellen und inhaltlichen Rahmenbedingungen, die einen entscheidenden Einfluss auf die spätere Implementierung der entwickelten Methoden und deren Validierung haben. In diesem Zusammenhang werden zunächst die Entwicklungstätigkeiten innerhalb der Untersuchungsumgebung knapp vorgestellt. Zur Unterstützung der unternehmensexternen Leserschaft findet eine begleitende Einordnung in die allgemeinen Phasen der Forschung und Entwicklung statt. Daran anknüpfend werden im zweiten Unterkapitel Organisation und Systemverständnis innerhalb der Untersuchungsumgebung, sowie dort geltende Vorgaben in Bezug auf die Nutzung von System Engineering-Tools in Kürze beleuchtet.

### **3.4.1 Entwicklungstätigkeiten und Einordnung in die Phasen von Forschung & Entwicklung**

Das der Arbeit zugrundeliegende Begleitprojekt fand im Zeitraum Oktober 2015 bis einschließlich März 2019 in Zusammenarbeit mit der Abteilung ENS (Systems Engineering Drives and Control Units) des Geschäftsbereichs Automotive Electronics (AE) der Robert Bosch GmbH statt. Die ENS bekleidet im Unternehmen eine zentrale Rolle als Kompetenz- und Standardisierungszentrum für Antriebselektroniken. Ihr Auftrag liegt in der Entwicklung innovativer Technologien auf den Gebieten der Antriebselektronik und der passiven Bauelemente<sup>1</sup> sowie deren konzernweite Bereitstellung in Form eines Baukastens. Im Rahmen der Aktivitäten von Forschung und Entwicklung (vgl. Kapitel 2.3.2) bildet die Abteilung das Bindeglied zwischen den Theorien der Grundlagenentwicklung und der Bereitstellung spezifischer Technologien für den Einsatz in Serienprodukten. Der Schwerpunkt der ENS-Entwicklung liegt in den Bereichen der Motorregelung, des Chipsatzes, der Sensorik, der Leistungselektronik und der Aufbau- und Verbindungstechnik welche als Komponenten des Baukastens bezeichnet werden. Die entwickelten Technologien finden sowohl einzeln als auch in Form von Systemlösungen ihren Eingang in die Serienentwicklung. Ein großer Teil, der von der ENS entwickelten Technologien findet, ihren Einsatz in den in Kapitel 2.1.2

---

<sup>1</sup> Hierbei findet eine Fokussierung auf kapazitive und induktive Bauteile in Form von Kondensatoren und Spulen statt.



beschriebenen elektrischen Kleinantrieben im Automobilbereich.<sup>2</sup> Abbildung 3.6 zeigt einen Auszug des Betätigungsfeldes der ENS im Rahmen der Entwicklung von Antriebselektroniken:

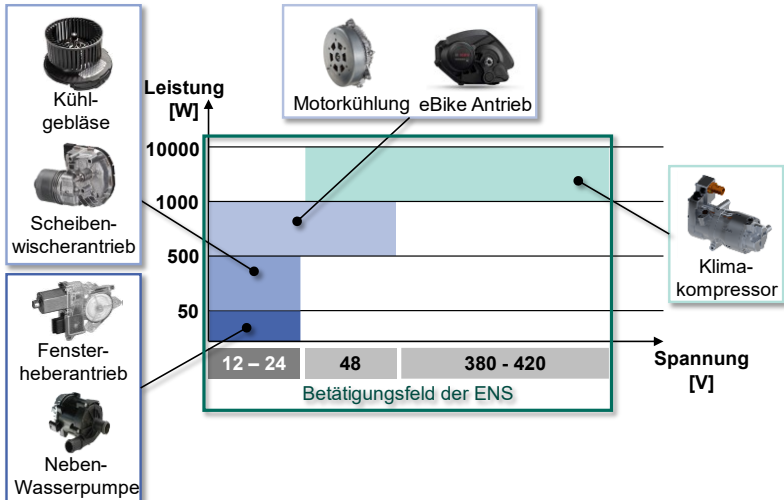


Abbildung 3.6: Auszug des Betätigungsfeldes der ENS. Das Diagramm zeigt verschiedene Produkte, in denen die entwickelten Steuergeräte enthalten sind. Diese sind entsprechend ihrer Betriebsspannung (x-Achse) und deren Ausgangsleistung (y-Achse) geordnet. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022)

Die Entwicklungstätigkeiten der Abteilung werden unternehmensintern in der sogenannten „**Innovation-Pipeline**“ dargestellt (Abbildung 3.7). Diese zeigt mittels eines weststromorientierten Ansatzes den o.g. Übergang von Theorie zu Technik. Die Standardisierungsaktivitäten der ENS werden im Folgenden als der Produktentwicklung paralleler Prozess anhand der Nomenklatur des Modells der

<sup>2</sup> Darüber hinaus werden ebenfalls non-Automotive-Anwendungen, HV-Anwendungen (400V) und Antriebe > 1kW Leistung mit den Technologien der ENS umgesetzt.

PGE – Produktgenerationsentwicklung nach Albers illustriert. Da innerhalb der ENS zwischen Produkt- und Technologiegenerationen unterschieden wird haben sich dort mit „X“ und „T“<sup>3</sup>, von der PGE<sup>3</sup> abweichende Bezeichner etabliert. Diese werden im nachfolgenden Abschnitt, der Vollständigkeit halber, in Klammern hinter der offiziellen PGE Bezeichnung angeführt.

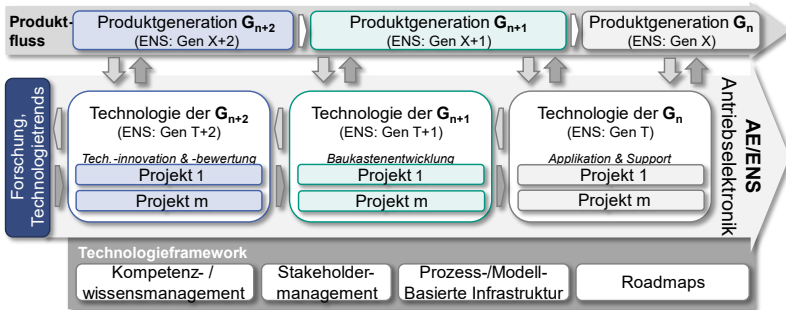


Abbildung 3.7: „Innovation-Pipeline“ der ENS. Darstellung der geplanten Nutzung von Technologien als Elemente des Referenzsystems aktueller und kommender Produktgenerationen, entsprechend der Nomenklatur der PGE – Produktgenerationsentwicklung.

Im Rahmen der „Innovation-Pipeline“ sind die Aufgabenbereiche der Abteilung sogenannten „**Technologiegenerationen**“ zugeordnet. Diese werden im Folgenden knapp vorgestellt:

Die Bezeichnung einer Technologiegeneration (ENS: Gen T+?) orientiert sich am Zeitpunkt deren Einsatzes in Serienprojekten, die entsprechend der PGE in Produktgenerationen  $G_n$  (ENS: Gen X+?) geplant werden. Produkte deren Serienproduktion bereits angelaufen ist werden als  $G_n$  (ENS: Gen X) bezeichnet. Die bereits in der Entwicklung befindliche nächste Produktgeneration heißt  $G_{n+1}$  (ENS: Gen X+1). Die darauffolgenden Konzepte der demnach übernächsten Produktgeneration fallen unter die Bezeichnung  $G_{n+2}$  (ENS: Gen X+2). Diesem Verständnis folgend werden jene Technologien, die in aktuellen Serienprodukten  $G_n$  eingesetzt sind, als **Gen T** bezeichnet. Die Tätigkeiten der Abteilung innerhalb der

<sup>3</sup> In der PGE wird die nächste am Markt verfügbare Generation immer als  $G_n$  bezeichnet.

**Technologieentwicklung für  $G_n$**  (ENS: Gen T) umfassen die Applikation im Baukasten befindlicher Antriebstechnologien in laufenden Serienprodukten  $G_n$  sowie Experten- und Bauteilsupport für laufende Entwicklungsprojekte  $G_{n+1}$ .

**Die Technologie für  $G_{n+1}$**  (ENS: Gen T+1) bezeichnet die Technologiegeneration, deren Technologien vornehmlich für den zukünftigen Einsatz in der **nächsten Generation** von Serienprodukten  $G_{n+1}$  vorgesehen sind. In diesem Zusammenhang prüft die Abteilung laufende Vorentwicklungsprojekte sowie neue Produktkonzepte auf die Eignung die entsprechenden Technologien einzusetzen. Die Vorentwicklungsprojekte können von der ENS zusätzlich durch den Aufbau von Elektronikprototypen, sogenannten Funktionsmustern (vgl. Kapitel 2.3.2) aus den bestehenden Baukastenelementen unterstützt werden. Für die  $G_{n+1}$  entwickelte Technologien (ENS: Gen T+1) stehen nach ihrer Entwicklung ebenfalls als standardisierte Elemente des Antriebselektronikbaukastens bereit. Basierend auf der langjährigen Erfahrung wurden die Baukastenelemente anwendungsfallsspezifischen Vorzugslösungen auf Systemebene sogenannten „**Technologiesets**“ (TS) zugeordnet. Die vier abgeleiteten Technologiesets unterteilen dabei das ENS Portfolio entsprechend diversen Einsatzkriterien. Die bedeutendsten Kriterien sind dabei die zugrundeliegende Bordnetzspannung, das bewertete ASIL-Level der Zielanwendung, die benötigte Rechnerleistung und die Antriebsleistung. Zur Bildung der TS wurde als Ausgangspunkt ein generischer Anforderungs- und Funktionskatalog erstellt, der die grundlegende Funktionalität mechatronischer Steuergeräte umfasst. Basierend auf diesem Katalog wurden TS-spezifische Funktionen und Anforderungen in getrennten Katalogen ergänzt. Die daraus gebildete Struktur einer gemeinsamen Basis, die anschließend für einzelne Produktgruppen konkretisiert wird, folgt der in Kapitel 2.4.2 beschriebenen hybriden Baukastenarchitektur. Allerdings finden die domänenspezifischen Wechselwirkungen innerhalb der TS aktuell nur in begrenztem Maße Berücksichtigung. Die Zusammenhänge zwischen Softwareelementen und den Elementen des Chipsatzes werden in den TS sehr detailliert betrachtet, wohingegen das systemische Zusammenspiel mit der Leistungsendstufe oder die Auswirkungen der AVT-Konzepte lediglich konzeptionell erfasst sind. Die vorliegende Arbeit soll genau an diesem Punkt ansetzen und eine ganzheitliche Betrachtung in Bezug auf die Bausteinkompatibilitäten schaffen. Dadurch soll die Funktionsmustererstellung weiter beschleunigt und damit die Untersuchung und frühzeitige Validierung neuer Produktkonzepte ermöglicht werden. Technologien die in der übernächsten Generation von Serienprodukten  $G_{n+2}$  zum Tragen kommen werden der entsprechenden Entwicklung (ENS: Gen T+2) zugeordnet. Die ENS betätigt sich hierzu in den Bereichen des Scoutings, der Evaluierung und Definition der Technologien auf Basis der Ergebnisse der Grundlagenforschung. Im Fokus der Aktivitäten liegen Machbarkeitsstudien zum Nachweis von Funktion und

Industrialisierbarkeit der Technologien. Als Rahmenbedingung der Untersuchungen gelten die Trends der Hochintegration und der Miniaturisierung. Am Ende der Technologieentwicklungsaktivitäten für die  $G_{n+2}$  steht der Transfer ausgewählter Technologiekonzepte zu den Standardisierungstätigkeiten in Bezug auf die Technologien der  $G_{n+1}$  (ENS: Gen T+1). Die Übergänge zwischen den Technologiegenerationen - in der Pipeline als Pfeile dargestellt - veranschaulichen den Einfluss der übergeordneten Innovationsprozesse des „Technology-push“ (von links nach rechts) und des „Market-pull“ (von rechts nach links) nach Baker et al. (1967).

Vergleicht man die Einteilung der Technologiegenerationen mit den Phasen der F&E und deren Ergebnisse (vgl. Kapitel 2.3.2) lässt sich folgendes feststellen:

Aufgrund der anteilig vorhandenen technology-push und market-pull Mechanismen finden sich in der Entwicklung der Technologien für  $G_{n+2}$  (ENS: Gen T+2) sowohl Aspekte der Technologieentwicklung und der Vorentwicklung wieder. Im Falle des technology-push werden die Technologien zunächst ohne Vorhandensein eines Produktkonzeptes untersucht, was der klassischen Technologieentwicklung entspricht. Beim market-pull hingegen werden bereits frühzeitig Produktkonzepte mitevaluiert, um weitere Technologiebedarfe zu erkennen. Hierbei handelt es sich definitionsgemäß um Vorentwicklungsaktivitäten. Die Tätigkeiten in Zusammenhang mit der Entwicklung von Technologien für  $G_{n+1}$  (ENS: Gen T+1) lassen sich ausnahmslos der F&E Aktivität der Vorentwicklung (vgl. Kapitel 2.3.2) zuordnen. Deren Umfang ist allerdings stark von den Vorarbeiten beim Transfer der  $G_{n+2}$  Technologien abhängig. Die Aktivitäten der im Rahmen der Technologieentwicklung für  $G_n$  bewegen sich in der Phase der Produkt- und Prozessentwicklung nach Specht et al. (2002). Bei den innerhalb der „Innovation-Pipeline“ vorgestellten Tätigkeiten handelt es sich um ein treffendes Beispiel, wie das Modell der PGE – Produktgenerationsentwicklung nach Albers in der industriellen Praxis bewusst eingesetzt wird. Die im Zuge der Technologieentwicklung erzielten Erkenntnisse stehen im Nachgang, als Elemente eines Referenzsystems auch produkt- und generationsübergreifend zur Verfügung.

### **3.4.2 Systemverständnis der AE/ENS und organisatorische Rahmenbedingungen**

#### **Systemverständnis der AE/ENS**

Dem Systemverständnis der ENS liegt ein Modell von vier Ebenen zugrunde (vgl. Abbildung 3.8). Dieses besteht aus der Produktebene, der Systemebene, der Sub-Systemebene und der Domänenebene. Alle Aktivitäten im Bereich von Produkt- bis

Sub-Systemebene (Level 0-2) werden dem Tätigkeitsbereich der Systementwicklung zugeordnet. Die eigentliche Domänenentwicklung findet demnach auf der Domänenebene (Level 3) statt. Im Zentrum der Systembetrachtung der ENS liegt der **elektrische Antrieb**, engl. Electronic Drive System (EDS) (Level 1). Diesen gilt es im Rahmen des Auftrags der standardisierten Antriebselektronikentwicklung in Gänze zu verstehen. Folglich ist ein gewisses Schnittstellenwissen bezüglich des Produktes (Level 0), welches den Konzern in Richtung des Kunden verlässt, unerlässlich. Das Produkt z.B. eine Kühlwasserpumpe liegt häufig in der Verantwortung eines anderen Geschäftsbereiches und wird als Supersystem des Antriebes verstanden. Der Antrieb teilt sich je nach Ausführung<sup>4</sup> in die drei Sub-Systeme **ECU**, **Motor** und **Getriebe** auf.

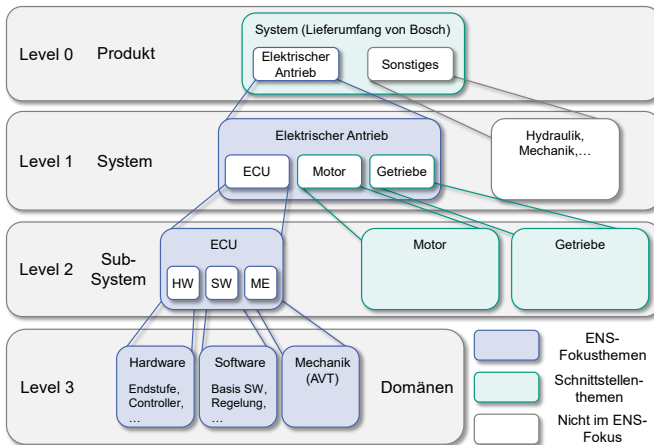


Abbildung 3.8: Ebenenmodell des ENS-Systemverständnisses inklusive Fokus- und Schnittstellenthemen. Levelstruktur aufgeteilt in Produkt-, System-, Sub-System- und Domänen-Level.

In der Sub-Systemebene liegt der Fokus der ENS auf der Entwicklung der Antriebselektronik im Rahmen dieser Arbeit auch als Steuergerät (ECU) bezeichnet

<sup>4</sup> Hierbei sei darauf hingewiesen, dass nicht jeder elektrische Antrieb eine mechanische Energieumformung in Form eines Getriebes beinhaltet. Dieses wird je nach Anwendungsfall als optional betrachtet.

(vgl. Kapitel 2.1.2). Hierfür ist eine genaue Kenntnis der internen Schnittstellen des Antriebs von der Elektronik zum Motor und zum Getriebe unabdingbar. Die ECU ist in der Domänenebene in Hardware, Software und Mechanik aufgeteilt. Auf dieser Ebene werden die einzelnen Technologien des Baukastens in den Bereichen Basissoftware, Motorcontrol (beide Domäne SW), Leistungselektronik, Chipsatz, Sensorik (Domäne HW) und der Aufbau- und Verbindungstechnik (Domäne Mechanik) entwickelt. Für die weitere Bearbeitung ergibt sich aus dem Systemverständnis folgende Präzisierung in Bezug auf die Begriffe der Kundenanforderung und der Kundenanfragen: Die direkten Kundenanforderungen richten sich wie beschrieben an den Lieferumfang eines elektrischen Kleinantriebes bzw. eines Produktes, in dem ein entsprechender Antrieb enthalten ist. Die behandelten mechatronischen Steuergeräte sind in beiden Fällen als Subsystem zu betrachten. Liegt vom unternehmensexternen Kunden kein dediziertes Lastenheft des mechatronischen Steuergerätes vor, wird vorausgesetzt, dass die Anforderungen an die ECU der ENS von der produktführenden Abteilung innerhalb des Unternehmens bereitgestellt werden. Die produktführende Abteilung fungiert als Empfänger des entwickelten Steuergerätes wiederum als interner Kunde der ENS. Je nach Betrachtung kann der Terminus Kundenanfrage demnach die Anfrage eines unternehmensexternen oder -internen Kunden bedeuten.

### Organisation der Abteilung

Entsprechend der Darstellung in der „Innovation-Pipeline“ und dem zugrundeliegenden Systemverständnis der ENS finden innerhalb der einzelnen Technologiegenerationen (ENS: Gen T+?) jeweils ähnliche Prozesse der System- und Domänenentwicklung statt. Als Organisationsform der Abteilung wurde daher die in Abbildung 3.9 dargestellte Matrixorganisation gewählt. In der horizontalen, der sogenannten objektbezogenen Dimension ist die Abteilung gemäß den einzelnen Technologiegenerationen in Programmen organisiert. Statt jede Technologiegeneration in einem eigenständigen Verantwortungsbereich zu organisieren, wurden die Aktivitäten der Technologieentwicklung für  $G_{n+1}$  und  $G_n$  (ENS: Gen T+1; Gen T) aufgrund ihrer engen inhaltlichen Verzahnung einem Programm zugeordnet. Als zusätzliches Programm wurde das Technologiemanagement und Benchmarking zur Abstimmung der Technologiestandardisierung Geschäftsbereich übergreifend installiert.

Die vertikale Dimension spiegelt den Funktionsbezug der Matrixorganisation wider. Hierbei sind die Mitarbeiter entsprechend ihrer Kompetenzen in drei Gruppen eingeteilt, welche wiederum inhaltlich zu einzelnen Teams organisiert sind. Die bisherigen Inhalte des Baukastens wurden gemäß der, in der Steuergeräteentwicklung vorherrschenden Pfadabhängigkeit (vgl. Kapitel 2.1.2) auf

das Zusammenspiel von Motorregelung, Sensorik, Basissoftware und Chipsatz also mit Fokus auf den Domänen der Software und Hardware optimiert. In Zusammenhang mit dem zunehmenden Kompetenzaufbau in den Bereichen der Leistungselektronik und AVT innerhalb der Abteilung sollte der bisherige Prozess der Baukastenstrukturierung hinsichtlich seiner ganzheitlichen Eignung für alle Domänen geprüft werden.

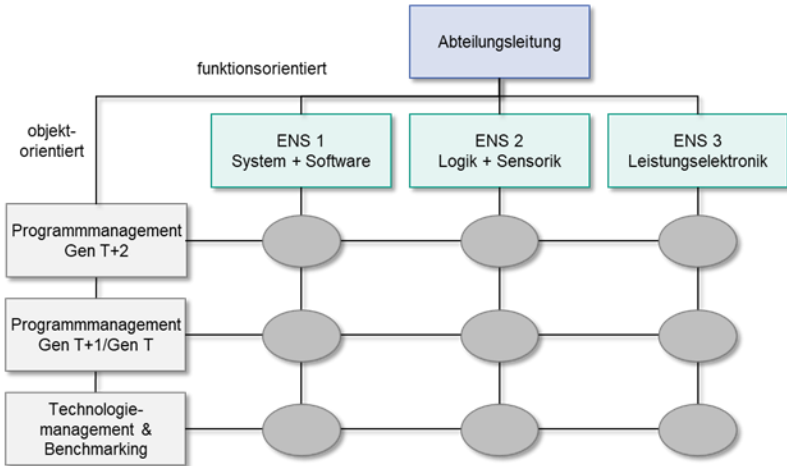


Abbildung 3.9: Vereinfachtes Organigramm der ENS als Matrixorganisation in Anlehnung an Schlick et al. (2018). Projektorganisation (links) und Linienorganisation (rechts) sind dabei zur Matrix verwoben.

## Toolingvorgaben und inhaltliche Vorarbeiten

Entsprechend der vielfältigen Einsatzmöglichkeiten der von der ENS bereitgestellten Technologien und Baukastenelemente erweist sich die Menge an produktspezifischen Anforderungen an jene Elemente als sehr umfangreich.<sup>5</sup> Um die Wiederverwendbarkeit der Baukastenelemente entsprechend des breiten Produktportfolios gewährleisten zu können wurde anhand bekannter Produkthanforderungen eine durchgängige Datenbank generischer Anforderungen und Funktionen erstellt. Diese erstreckt sich ausgehend vom Level des elektrischen Antriebs über die Sub-systeme ECU und Motor<sup>6</sup> bis in die Domänen (Software, Hardware und Mechanik) und diente der vorliegenden Arbeit als hilfreiches Nachschlagewerk. Zur Dokumentation dieses Kataloges wurde das Tool **DOORS classic** der Herstellers **IBM** verwendet. Innerhalb des Kataloges sind die einzelnen Anforderungen und Funktionen entsprechend ihrer Relevanz den Spalten der abgeleiteten Technologiesets zugeordnet und dort entsprechend quantifiziert. Zur Herstellung der geforderten Traceability durch **ASPICE** und der **ISO 26262** sind die Anforderungen und Funktionen levelübergreifend verlinkt. Zur Modellierung der SysML basierten Modelle (Produktreferenzmodelle, Baukastenreferenzmodelle, Baukastenmodelle) wurde das Systemarchitektur-Tool **Rational Rhapsody** von **IBM** verwendet. Die in der vorliegenden Arbeit entwickelten Methoden sind zunächst unabhängig von der Toollandschaft der Untersuchungsumgebung aufgestellt und werden mit dem Ziel der Baukastenanwendung erst in einem zweiten Schritt in die Softwarewerkzeuge implementiert.

---

<sup>5</sup> Die Anforderungssätze verschiedener Produkte können sich bspw. durch Hinzukommen oder Wegfallen von Anforderungen, sowie eine unterschiedliche Quantifizierung der Anforderungen unterscheiden.

<sup>6</sup> Die motorseitig beschriebenen Funktionen liegen dabei nicht im Fokus der ENS und dienen lediglich der Darstellung von Schnittstelleninformationen für die ECU-Entwicklung.



## 4 Modellierung der Baukastenstrukturierung

Das vorliegende Kapitel befasst sich mit der Entwicklung und Implementierung einer MBSE-gestützten Methode zur Baukastenstrukturierung für mechatronische Steuergeräte. Zunächst wird der ermittelte Forschungsbedarf im Rahmen der deskriptiven Studie I abgesichert und darauf aufbauend ein Zielsystem der zu entwickelnden Methode erstellt (Kapitel 4.1). Anschließend wird eine präskriptive Studie durchgeführt, welche im ersten Schritt der Entwicklung der Baukastenstrukturierung (Kapitel 4.2) und anschließend ihrer Implementierung innerhalb der Randbedingungen der Untersuchungsumgebung (Kapitel 4.3) dient. Die mittels Studien geschaffenen Ergebnisse werden im Nachgang reflektiert und diskutiert (Kapitel 4.4).

### 4.1 Ermittlung des Zielsystems für die Baukastenstrukturierung

Der DRM folgend werden mittels deskriptiver Studie I zunächst die komplexen Vorgänge bei der Entwicklung mechatronischer Steuergeräte auf Basis realer Produktentstehungsprozesse illustriert. Anschließend werden im Expertendialog Schlüsselfaktoren ermittelt, die auf mögliche Ansatzpunkte zur methodischen Unterstützung hinweisen. Im Kern dieser Studie steht die Beantwortung der in Kapitel 3.1.3 abgeleiteten Forschungsfrage hinsichtlich ihrer Aspekte der Baukastenstrukturierung<sup>1</sup> (vgl. Kapitel 3.2.1):

*FF1: Welche Spezifika und Bedarfe im Bereich der Strukturierung und Anwendung von Baukästen existieren in der Frühen Phase der PGE von mechatronischen Steuergeräten und welches Zielsystem an entsprechende Methoden lässt sich daraus ableiten?*

---

<sup>1</sup> Die im Rahmen von Kapitel 4.1 dargelegten Untersuchungen sind über folgende Publikationen aufgeteilt veröffentlicht worden:

Powelske et al. (2017), Albers, Peglow, Powelske, Birk & Bursac (2018), Peglow et al. (2017).

Gemäß den verschiedenen Facetten des in der Literaturrecherche ermittelten Forschungsbedarfes, wird die Forschungsfrage zunächst in Unterfragen unterteilt:

*FF1.1: Welche Herausforderungen lassen sich aus den Rahmenbedingungen des Produktentwicklungsprozesses mechatronischer Steuergeräte ableiten?*

*FF1.2: Welche Herausforderungen ergeben sich aus der Produktarchitektur mechatronischer Steuergeräte?*

*FF1.3: Welche Bedarfe werden aus Sicht der Entwicklung und des Managements definiert?*

*FF1.4: Welche Methoden des MBSE können zur Unterstützung der Baukastenstrukturierung und –anwendung herangezogen werden?*

#### **4.1.1 Studiendesign**

Zur Beantwortung der Teilforschungsfragen FF1.1 – FF1.3 werden aus dem in Kapitel 3.2.2 vorgestellten Kanon situationsgerecht empirische Methoden ausgewählt und eingesetzt (Abbildung 4.1).

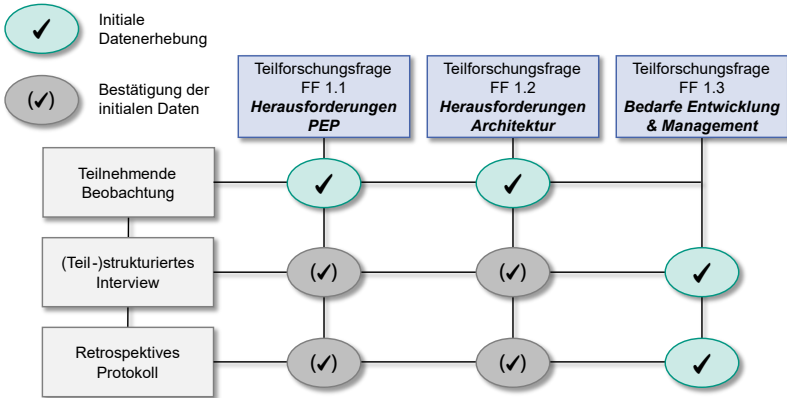


Abbildung 4.1: Übersicht der, zur Ermittlung des Zielsystems angewandten empirischen Methoden in Abhängigkeit der Teilforschungsfragen (vgl. Kapitel 3.2)

Entsprechend des in Abbildung 4.1 dargestellten Sachverhalts wurden zunächst über teilnehmende Beobachtungen prozess- und architekturseitige Herausforderungen erfasst. Anschließend wurden die Beobachtungsergebnisse im ersten Abschnitt der teilstrukturierten Interviews abgesichert. Im zweiten Abschnitt der Interviews wurden stakeholderspezifische Bedarfe der Entwicklung und des Managements erhoben und in Form retrospektiver Protokolle fixiert.

Eine detaillierte Beschreibung des Vorgehens zur Datenerhebung und zur Ausgestaltung der verwendeten Forschungsmethoden sei im folgenden Abschnitt gegeben:

Die Herausforderungen an die Baukastenstrukturierung aus dem PEP der mechatronischen Steuergeräte (Teilforschungsfrage FF1.1) wurden im Rahmen **teilnehmender Beobachtungen** ermittelt. Hierzu wurden über die gesamte Forschungsdauer insgesamt fünf Entwicklungsprozesse von Produkten unterschiedlicher Komplexität entlang ihrer Frühen Phase begleitet. Dem Autor der vorliegenden Arbeit wurde hierbei eine aktive Beobachtung (in Anlehnung an Marxen (2011)) des Entwicklungsgeschehens ermöglicht. Die Datenerhebung erfolgte in Form von Beobachtungsprotokollen, die während der Teilnahme an Meetings und Entwicklungswshops vom Forscher angefertigt wurden. Im Fokus der Datenerhebung stand die Hypothese, dass die aus dem Stand der Forschung bekannten Herausforderungen mechatronischer Steuergeräte innerhalb der

beobachteten Entwicklungsprojekte ebenfalls auftreten und dort zu Iterationen führen. Darüber hinaus wurde dem Autor Einsicht in die Entwicklungsartefakte (z.B. Konzeptentwürfe, Entscheidungsmatrizen) des jeweiligen Projektstandes gewährt. Auf dieser Basis konnten zusätzlich Erkenntnisse über den Einfluss der Produktarchitekturen mechatronischer Steuergeräte und deren Auswirkungen auf die Baukastenentwicklung gewonnen werden (Teilforschungsfrage FF1.2). Die vom Forscher ermittelten und in Form o.g. Protokolle erfassten Herausforderungen wurden anschließend im ersten Abschnitt der **teilstrukturierten Interviews** mit ausgewählten Stakeholdern abgesichert (vgl. Abbildung 4.1)<sup>2</sup>. Die Interviewpartner wurden dabei gebeten zu jeder der identifizierten Herausforderungen Stellung zu nehmen. Im zweiten Abschnitt der Interviews wurden die individuellen Bedarfe der Stakeholder an die Baukastenstrukturierung erfasst (Teilforschungsfrage FF1.3). Entsprechend dem Fokus der vorliegenden Arbeit wurden die Stakeholder als Vertreter aus den Bereichen Management und Entwicklung einer Vorausentwicklungsabteilung für mechatronische Steuergeräte identifiziert. (vgl. Kapitel 3.3) Beim Management sind sowohl die Interessen des Programm- bzw. Projektmanagements als auch der Linienführung zu berücksichtigen. Seitens der Entwicklung sind die Bedarfe der Systementwicklung sowie der Domänenentwicklung in den Bereichen der Software, Hardware und Mechanik ausschlaggebend.

Insgesamt wurden 29 Personen aus den Bereichen der Entwicklung und des Managements interviewt, die alle bereits mehrjährige (>3 Jahre) Erfahrung im Bereich der Baukastenentwicklung aufweisen. Da es sich bei der Untersuchungsumgebung, um eine geschäftsbereichszentrale Abteilung für die Baukastenentwicklung der Steuergeräte im Einsatz in Kleinantrieben handelt, sind deren Mitarbeiter dementsprechend stark mit den produktführenden Abteilungen verzahnt. Die Aussagen der Mitarbeiter geben demnach Aufschluss über die generelle, unternehmensinterne Meinung in Bezug auf die Bedarfe der Steuergeräteentwicklung im Einsatz in Kleinantrieben. Darüber hinaus wiesen einige der Mitarbeiter ebenfalls mehrjährige Berufserfahrung auf dem Gebiet der Steuergeräteentwicklung, außerhalb des untersuchten Konzerns auf. Die Interviewfragen wurden entsprechend konzern-unabhängig gestellt, so dass, auf Basis oben genannter Erfahrungen, auf eine generelle Übertragbarkeit der Aussagen auf die Steuergeräteentwicklung im Einsatz von Kleinantrieben geschlossen werden kann. Abbildung 4.2 gibt Aufschluss über die

---

<sup>2</sup> Der Interviewleitfaden kann dem Anhang der vorliegenden Arbeit entnommen werden

Domänenzugehörigkeit der Befragten. Die Ergebnisse der Interviews wurden im Anschluss in Form retrospektiver Protokolle<sup>3</sup> erfasst. Die initiale Erstellung der Protokolle wurde hierbei durch den Autor der vorliegenden Arbeit übernommen. Inhaltlich wurden dazu die Aussagen des jeweiligen Interviewpartners zusammengefasst und entsprechend der, mittels teilnehmender Beobachtungen ermittelten Herausforderungen sortiert. Während der Interviews von den jeweiligen Stakeholdern erstellte Grafiken wurden innerhalb der Protokolle ebenfalls berücksichtigt. Anschließend wurden die Protokolle den jeweiligen Interviewpartnern zum Review übergeben. Diese wurden dazu angehalten zu prüfen, ob durch die Zusammenfassung und Restrukturierung der Informationsgehalt weiterhin dem ihrer initialen Aussage entspricht. Zusätzlich wurden die Stakeholder gebeten sensible, firmeninterne Informationen zu kennzeichnen, damit diese aus dem Protokoll gestrichen werden können. Die gewonnenen Ergebnisse wurden im Nachgang zu einem Zielsystem für die methodische Unterstützung kondensiert.

---

<sup>3</sup> Die Nutzung des retrospektiven Protokolls ermöglicht es sensible, unternehmensinterne Inhalte bereits in der Dokumentation herauszufiltern.

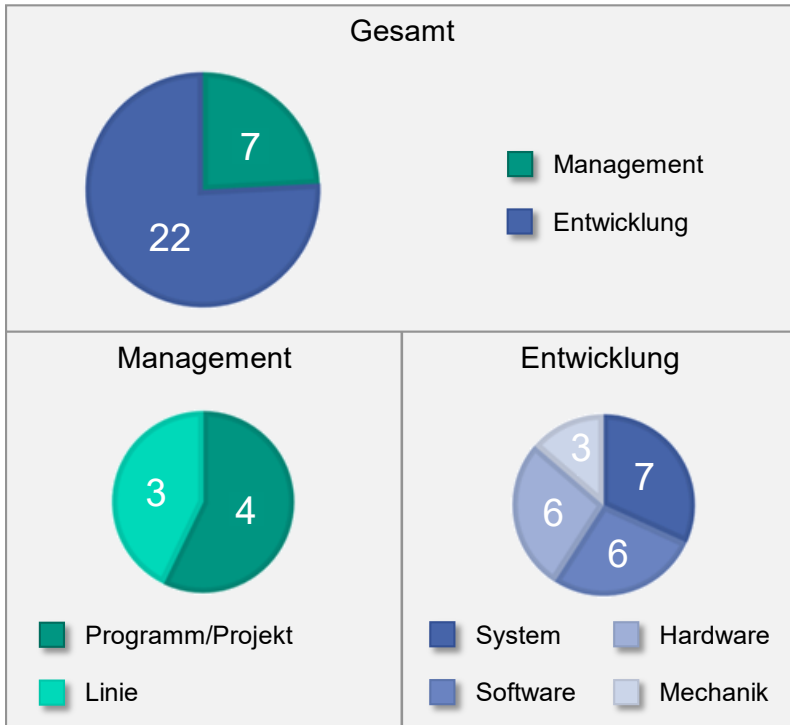


Abbildung 4.2: Übersicht der Domänenzugehörigkeit der Interviewpartner aufgeteilt in Management- und Entwicklungsdisziplinen.

Auf Basis des Zielsystems wurden abschließend bestehende Methoden aus dem Kanon des MBSE hinsichtlich ihrer Eignung als mögliche Anknüpfungspunkte ausgewählt. (Teilforschungsfrage FF1.4)

#### 4.1.2 Studienergebnisse

In Anlehnung an Forschungshypothese 1 (Kapitel 3.1.2), werden die Spezifika der mechatronischen Steuergeräteentwicklung auf deren Auswirkungen auf die Baukastenstrukturierung hin untersucht. Daraus resultierende Herausforderungen werden entsprechend der in Kapitel 4.1.1 dargelegten Vorgehensweise ermittelt. Die Ergebnisse der Studie werden nachfolgend anhand der betreffenden Teilforschungsfragen vorgestellt (Abbildung 4.3):

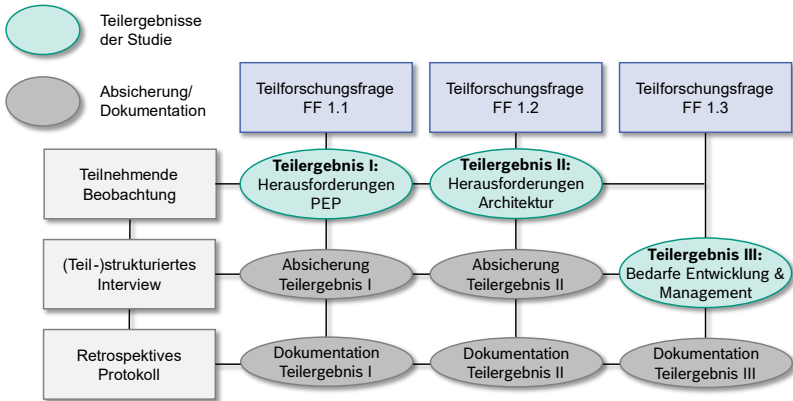


Abbildung 4.3: Übersicht der Studienergebnisse abhängig von den gewählten empirischen Methoden und deren Beitrag zur Beantwortung der Teilforschungsfragen

### Teilergebnis I: Herausforderungen des PEP mechatronischer Steuergeräte

*FF1.1 Welche Herausforderungen lassen sich aus den Rahmenbedingungen des Produktentwicklungsprozesses mechatronischer Steuergeräte ableiten?*

Im Zuge der teilnehmenden Beobachtungen erwiesen sich die Vorgänge innerhalb der Untersuchungsumgebung als stereotypisch für die Frühe Phase der Entwicklung mechatronischer Steuergeräte. Dementsprechend deutlich konnten die in Kapitel 2.1.2 vorgestellten spezifischen Charakteristika beobachtet werden. Hinsichtlich der Prozesslandschaft kristallisierte sich eine klare Orientierung an den dominanten Prozessmodellen (vgl. Kapitel 2.2.2) allen voran am V-Modell, heraus. Aufgrund der Beschaffenheit der mechatronischen Steuergeräte werden die Herausforderungen in Zusammenhang mit einem Herstellerbaukasten nach Ponn & Lindemann (2008) untersucht. Die beobachteten Herausforderungen werden im Folgenden hinsichtlich ihrer Relevanz für die Baukastenstrukturierung beleuchtet:

- **Pfadabhängigkeit:**

Aufgrund der langen Historie der Steuergeräteentwicklung in der Untersuchungsumgebung hat sich eine deutliche Pfadabhängigkeit (vgl. Kapitel 2.1.2) in Richtung Software- und Hardwareentwicklung herausgebildet.

Dementsprechend werden die funktionalen Anforderungen an die Domänen der Software- und Hardwareentwicklung priorisiert. Die Betrachtung der Aufbau- und Verbindungstechnik erfolgt nachgelagert. Diese hat jedoch einen starken Einfluss auf die Funktionsfähigkeit der Bauteile, was zu zahlreichen Iterationen im Entwicklungsprozess führt. In Bezug auf die Baukastenentwicklung ergibt sich aus diesem Umstand eine besondere Herausforderung. Entsprechend ihrer Definition (vgl. Kapitel 2.4.2) kann die Austauschbarkeit von Modulen nur dann gewährleistet werden, wenn eine gewisse Unabhängigkeit zwischen ihnen vorliegt. Durch die Pfadabhängigkeit wird genau dieser Prozess der Modulbildung erschwert. Es ist daher abzusehen, dass im Zuge der Baukastenstrukturierung Einfluss auf die bisherige Entwicklungspraxis genommen werden muss, um starke Abhängigkeiten innerhalb einzelner Module zu binden und eine ganzheitliche Modulbildung zu ermöglichen.

- **Sequenzielles Vorgehen nach dem V-Modell:**

Die Folgen eines sequenziellen Entwicklungsvorgehens nach dem V-Modell wurden in Kapitel 2.2.3 bereits beleuchtet. An dieser Stelle wird daher vorwiegend auf für mechatronische Steuergeräte spezifische Herausforderungen in Zusammenhang mit dem V-Modell eingegangen. Als besonders kritisch erweist sich hierbei die weitestgehend parallele und unabhängige Bearbeitung einzelner Arbeitspakete auf Domänenebene. Dies kann beispielhaft an der für mechatronische Steuergeräte charakteristischen Integration von Domänenlösungen auf möglichst geringem Bauraum verdeutlicht werden. Die räumliche Aneinanderreihung der mechatronischen Elemente führt zu einer Erhöhung negativ-emergenter Wechselwirkungen, deren Berücksichtigung teilweise nur mit gegenläufigen Maßnahmen erreicht werden kann. Soll die Gestaltung eines Filters aufgrund parasitärer leitungsgebundener Induktivitäten möglichst kompakt und in Nähe der Endstufe erfolgen, ist der Wärmeeintrag ebendieser Endstufe auf den Filter aus Gründen der Lebensdauer nicht erwünscht. Analog zur Pfadabhängigkeit scheint es, als läge die Herausforderung für die Baukastenstrukturierung vor allem in der Definition geeigneter Modulgrenzen und der Wahrung der Bausteinkompatibilitäten. Die bei der Zusammenführung, der domänenspezifischen Lösungen beobachteten, langwierigen Diskussionen über die Zuständigkeiten für einzelne Entwicklungsaufgaben, untermauern diese These.

Wird die Vorgehensweise des V-Modells streng sequenziell verfolgt, ist zunächst ein lösungsneutrales Herunterbrechen der Anforderungen und Funktionen von System- auf Domänenebene vorgesehen. In der Untersuchungsumgebung starten Systementwicklung und die Domänenentwicklung häufig zeitgleich ins Projekt. Da die Domänen entsprechend des V-Modells auf den Input des Systems angewiesen sind entsteht hieraus ein Konflikt. Seitens des Systems ist die Zeit zur Identifikation



eines möglichen Produktkonzeptes sehr gering und unmittelbar auf den Erfahrungsschatz des Systementwicklers gestützt. Seitens der Domänen werden in Ermangelung konkreter Anforderungen vermehrt Bottom-Up Ansätze gestartet. Als Resultat ist eine Tendenz zu lokalen domänenspezifischen Optimierungen, anstatt zur Erzielung eines globalen Optimums auf Steuergeräteebene zu beobachten.

Die Vorgaben des V-Modells werden durch die Anforderungen der ISO 26262 und der ASPICE Richtlinien hinsichtlich der Dokumentation der Arbeitsergebnisse verschärft. Die Baukastenstrukturierung muss demnach in der Lage sein, die dort formulierten Anforderungen z.B. nach der Einrichtung einer durchgängigen Traceability, zu erfüllen.

- **Frühe Phase der PGE:**

Im Zuge der teilnehmenden Beobachtung wurden nur Entwicklungsprozesse begleitet, die sich in der Frühen Phase befanden. In allen Fällen konnte das Fehlen eines konkreten Zielsystems und die Abhängigkeit erfahrungsgestützter Annahmen beobachtet werden. Im Gegensatz zur klassischen (Serien-)Produktentwicklung finden in der Vorentwicklung zusätzlich sogenannte Plattformentwicklungen statt. Unter dem Begriff Plattform wird in diesem Zusammenhang die Vorentwicklung von Produktkonzepten verstanden, bei denen Kundenaufträge antizipiert werden. Die Struktur des ohnehin vagen Zielsystems stützt sich hierbei noch ausgeprägter auf Annahmen. Der frühzeitigen Validierung von Produktkonzepten zur Konkretisierung des Zielsystems kommt an dieser Stelle eine noch bedeutendere Rolle zu. Die frühzeitige Validierung ist durch die Anwendung des V-Modells allerdings deutlich erschwert. Ein Baukasten könnte an dieser Stelle mittels geeigneter Kompatibilitätsbetrachtungen zu Vereinfachung der frühen Validierung der Produktkonzepte beitragen. Die Baukastenstrukturierung muss eine dementsprechende Beschaffenheit aufweisen.

Ein weiterer, zu beachtender Aspekt ist die aktive Nutzung von Referenzsystemen im Sinne der PGE. Innerhalb der Untersuchungsumgebung wurden bewährte Designlösungen wiederverwendet, allerdings waren diese meist auf Teilsysteme<sup>4</sup> beschränkt. Die Kompatibilität zwischen den Teilsystemlösungen musste für den vorliegenden Produktkontext erneut aufwändig untersucht werden. An dieser Stelle kann über die Baukastenstrukturierung eine Möglichkeit zur aktiven Nutzung der Wissensbasis berücksichtigt werden.

---

<sup>4</sup> Diese sind Elemente des Referenzsystems

## **Teilergebnis II: Herausforderungen der Produktarchitektur**

### *FF1.2 Welche Herausforderungen ergeben sich aus der Produktarchitektur mechatronischer Steuergeräte?*

Neben den prozessseitigen Herausforderungen konnten im Rahmen der teilnehmenden Beobachtungen wichtige Indizien für den Einfluss der Produktarchitektur mechatronischer Steuergeräte auf die Baukastenstrukturierung gesammelt werden. Diese sind nachfolgend erläutert:

- **Trend der Hochintegration:**

Die Produktarchitektur der mechatronischen Steuergeräte ist, zunehmend vom Trend der Hochintegration geprägt. Dies bedeutet, eine räumliche Zusammenführung verschiedener Funktionsträger zu einem Bauteil. Die Vorteile hochintegrierter Bauteile werden häufig in der Reduktion des benötigten Bauraums und der Stückkosten<sup>5</sup> gesehen. Im Gegenzug wird durch die Hochintegration eine gewisse funktionale Flexibilität eingebüßt, was die Einsatzmöglichkeiten entsprechender Bauteile stark auf einen spezifischen Anforderungsbereich eingrenzt. Trotz der Verfügbarkeit der integralen Lösungen kamen innerhalb der begleiteten Entwicklungsprozesse etwa zu gleichen Teilen weiterhin die modularen Lösungen zum Einsatz. Für das Technologieportfolio der Untersuchungsumgebung bedeutete dieser Umstand eine Koexistenz beider Architekturprinzipien und damit eine hohe Varianz verschiedener Bauteile.<sup>6</sup> Bei genauerer Betrachtung bilden die integralen Bauteile ihrerseits zusammen mit den sie umgebenden Elementen wiederum modulare Lösungen, deren Einsatz mit Bauteilen einer höheren Integrationsstufe abzuwägen ist. Entsprechend der Abstraktionsebenen der Produktstruktur lässt in Anlehnung an die Mechanismen der Standardisierungsmethoden ebenfalls ein fraktaler Charakter erkennen. Für einen Baukasten, genauer gesagt dessen Strukturierungsmethode, ergibt sich hieraus die Herausforderung die Auswahl zwischen modularen und integralen Lösungen zu unterstützen. Hierzu ist es zunächst erforderlich eine Vergleichbarkeit der verschiedenen technischen Herangehensweisen zu schaffen.

---

<sup>5</sup> Letztere wird durch die Nutzung von Skaleneffekten in der Produktion erzielt

<sup>6</sup> Entsprechend der gegenwärtigen Forschung gilt dies als charakteristisch für mechatronische Steuergeräte (Steffen, 2007)

- **Unterschiedliche Technologiezyklen der beteiligten Domänen:**

Bei der Betrachtung der Produktarchitektur mechatronischer Steuergeräte ist es ebenfalls auffällig, dass sich die Entwicklungszyklen der einzelnen Technologien je nach Domäne stark unterscheiden. Während die Halbleiterindustrie innerhalb von 12 bis 18 Monaten<sup>7</sup> eine Innovationsschleife durchläuft erneuert sich die Software zur Komponentensteuerung binnen weniger Monate. Was für die Entwicklung eines einzelnen Produktes den einmaligen Vergleich aktuell verfügbarer Bausteine bedeutet, wird für die Baukastenentwicklung zur echten Herausforderung. Alle in Produkten oder Studien eingesetzten Technologien, deren Wiederverwendung entsprechend der künftigen Anforderungslage absehbar ist, müssen auf deren Nutzung im Baukasten geprüft und bei Bedarf inkludiert werden können. Durch die ständig zunehmenden Funktionsumfänge und die Hochintegration entstehen mitunter große strukturelle Unterschiede über die Produktgenerationen hinweg. Starre über mehrere Jahre fixierte Strukturen, wie beispielsweise in der Baukastenentwicklung im Automobilsektor sind für die mechatronischen Steuergeräte aus diesem Grund undenkbar. Vielmehr sind flexible, mit geringem Aufwand erweiterbare Strukturen zu schaffen.

Zum Abschluss der Herausforderungsermittlung sind diese nachfolgend zusammengefasst:

- **Pfadabhängigkeit:** Software und Hardwareentwicklung bestimmend, Mechanikentwicklung in Bezug auf die Steuergeräte (hauptsächlich AVT) nachgelagert. Folge: Iterationen, da Abhängigkeiten nicht frühzeitig aufgedeckt werden können
- **Sequenzielles Vorgehen nach dem V-Modell:** lokale domänenspezifische Optimierung durch parallele, unabhängige Bearbeitung. Folge: Iterationen, da Zielkonflikte der Domänenlösungen zu spät erkannt werden
- **Frühe Phase der PGE:** hohe Unsicherheit gegenüber der Anforderungsbasis. Folge: Moving Targets, Aufwände zur Anforderungsabsicherung
- **Trend der Hochintegration:** modulare und integrale Modularchitekturen. Folge: Vergleichbarkeit zwischen den Architekturformen muss gewährleistet werden
- **Unterschiedliche Technologiezyklen der beteiligten Domänen:** hohe Innovationsdynamik. Folge: Erweiterbarkeit des Baukastens notwendig

---

<sup>7</sup> zeitintensive Qualifikationsprozesse für den Einsatz im Automobilbereich sind bereits eingerechnet

### **Teilergebnis III: Bedarfe Entwicklung & Management**

#### *FF1.3 Welche Bedarfe werden aus Sicht der Entwicklung und des Managements definiert?*

Die im Rahmen der teilnehmenden Beobachtungen ermittelten Herausforderungen wurden als Strukturierungsgrundlage der anschließenden Experteninterviews genutzt. Darauf aufbauend wurden die individuellen Bedarfe der einzelnen Stakeholder bezüglich der Baukastenstrukturierung aufgenommen. Die ermittelten Informationen werden im Folgenden zunächst vorgestellt und anschließend zu Schlüsselfaktoren des Zielsystems kondensiert:

- **Pfadabhängigkeit:**

Die in Zusammenhang mit der Pfadabhängigkeit abgeleiteten Herausforderungen wurden seitens der Entwicklung nahezu einstimmig bestätigt. Vor allem an den Schnittstellen von Hardware- und Mechanikentwicklung herrscht „der eindeutige Bedarf einer ganzheitlichen, domänenübergreifenden Betrachtung der Abhängigkeiten“<sup>8</sup>. Selbst aus Sicht der Softwareentwicklung, die sich zumindest „von der mechanischen Entwicklung als [weitestgehend] entkoppelt betrachtet“<sup>9</sup>, wurde die nachgelagerte Betrachtung der AVT als problematisch eingestuft. Hinsichtlich der Definition von Modulgrenzen besteht der Wunsch nach „eine[r] nachvollziehbare[n] Strukturierungsmethode, die [unter anderem] den domänenübergreifenden Abhängigkeiten Rechnung trägt“<sup>10</sup>. Damit verbundene etwaige Anpassungen der bisherigen Entwicklungspraxis werden als unkritisch eingestuft, sofern diese nachvollziehbar und transparent vermittelt werden. Seitens des Managements wurde diese Sichtweise in erster Linie von den Projektmanagern geteilt. Diese sehen in der ganzheitlichen Moduldefinition „eine Chance langwierige Iterationen zwischen den Domänen abzukürzen und einen zielführenden Umgang mit den [negativ-emergenten] Wechselwirkungen zu ermöglichen.“<sup>11</sup> Der Baukasten [und dessen Strukturierung] müsse[n] allerdings nachvollziehbare, verlässliche Aussagen in Bezug auf abzuleitende Produkte zulassen.“<sup>12</sup>

---

<sup>8</sup> Interviewpartner 3 (Entwicklung, Mechanik), 5 (Entwicklung, Hardware)

<sup>9</sup> Interviewpartner 10 (Entwicklung, Software)

<sup>10</sup> Interviewpartner 7 (Entwicklung, Hardware)

<sup>11</sup> Interviewpartner 27 (Management, Programm/Projekt)

<sup>12</sup> Interviewpartner 27 (Management, Programm/Projekt)

- **Sequenzielles Vorgehen nach dem V-Modell:**

Das V-Modell insbesondere der Makrozyklus hat sich für 90% der Befragten entlang der jahrelangen Nutzung als wichtiges Kommunikationsinstrument bewährt. Die sequenzielle Abarbeitung der einzelnen Schritte und die nachgelagerte Validierung werden von Seiten der Entwicklung allerdings als kritisch eingestuft, da diese „nicht dem iterativen Charakter der Produktentwicklung entsprechen“<sup>13</sup>. Gerade der Prozess des Herunterbrechens der Anforderungen von System- auf Domänenebene erweist sich für die Befragten als problematisch. Hier entsteht eine Lücke, die „nur mittels Nutzung von Erfahrungswissen aus anderen Produkten“<sup>14</sup> geschlossen werden kann. Der Baukasten „muss [folglich] eine Struktur aufweisen, die das Wissen aus Produkten nutzbar macht.“<sup>15</sup> Gleichzeitig ist die Kompatibilität der Elemente entsprechend nachvollziehbar zu gewährleisten. Die Systementwicklung setzt bezüglich der Baukastenstrukturierung „große Hoffnung in die Vergleichbarkeit verschiedener Produkt-/Architekturkonzepte“<sup>16</sup>. „Innerhalb der [Baukasten]Strukturierung sollen vor allem die bekannten Wechselwirkungen berücksichtigt werden.“<sup>17</sup> Das Management legt primär Wert auf die „prozesskonforme Dokumentation hinsichtlich ISO 26262 und ASPICE.“<sup>18</sup>

- **Frühe Phase der PGE:**

Aus Sicht des Managements spielt die Baukastenentwicklung eine maßgebliche Rolle bei der Konkretisierung des vagen Zielsystems in der Frühen Phase der PGE. „Wechselnde Anforderungen oder Moving-Targets sind regelrechte Effizienzkiller im Entwicklungsprozess.“ In der „[Baukastenstrukturierung und -]Anwendung wird [daher] die Chance gesehen flexibel auf die Volatilität der Anforderungen eingehen zu können.“<sup>19</sup> Weiterhin soll die Baukastenstrukturierung über eine geeignete Kompatibilitätsbetrachtung bereits eine Möglichkeit zur Validierung von Produktkonzepten vorsehen. Die Entwicklung sieht in der Frühen Phase, „dass bereits eine geringe Menge [...] [ausgewählter] Anforderungen über das spätere Produktkonzept entscheiden können.“<sup>20</sup> Diese gilt es bei der

---

<sup>13</sup> Interviewpartner 2 (Entwicklung, Mechanik)

<sup>14</sup> Interviewpartner 8 (Entwicklung, Hardware); In Anlehnung an die PGE würde man hierbei von Referenzsystemen sprechen

<sup>15</sup> Interviewpartner 13 (Entwicklung, Software)

<sup>16</sup> Interviewpartner 19 (Entwicklung, System)

<sup>17</sup> Interviewpartner 20 (Entwicklung, System)

<sup>18</sup> Interviewpartner 26 (Management, Programm/Projekt)

<sup>19</sup> Interviewpartner 23 (Management, Linie)

<sup>20</sup> Interviewpartner 9 (Entwicklung, Hardware)

Baukastenstrukturierung zu erfassen und bei der Moduldefinition zu berücksichtigen. Bezüglich der Nutzung von Informationen aus Referenzsystemen herrscht seitens des Managements die klare Vorgabe vorhandene Entwicklungserfahrung aktiv in die Baukastenstrukturierung mit einfließen zu lassen. Gleiches gilt für die Einbettung in vorhandene Tools und Prozesse.

- **Trend der Hochintegration:**

Bezüglich der Notwendigkeit einer Koexistenz integraler und modularer Bausteinstrukturen sind sich Entwicklung und Management einig. Hier liegen die Bedarfe eindeutig darin „die existierenden Lösungen vergleichbar zu machen“ und „den Auswahlprozess der Bausteine nachvollziehbar [zu] gestalten“<sup>21</sup> Aufgrund der unterschiedlichen Abstraktionsebenen, auf denen die Entwicklung stattfindet, wurde sowohl vom System als auch einzelnen Domänen die Frage nach der „Strukturierungstiefe des Baukastens“<sup>22</sup> gestellt. Eine „Nutzbarkeit der Baukastenstrukturierungsmethode [, entsprechend dem fraktalen Charakter der Standardisierungsmethoden,] auch auf Ebene der Modulentwicklung“<sup>23</sup> wurde seitens der Domänenentwicklung gewünscht.

- **Unterschiedliche Technologiezyklen der beteiligten Domänen:**

Aus Sicht des Managements sind vorrangig die schnelle Prüfbarkeit von Produktkonzepten, sowie die schnelle Erweiterbarkeit des Baukastens von großem Interesse. Da „innerhalb der Untersuchungsumgebung [...] sowohl Technologieentwicklung, Vorentwicklung und Produktentwicklung statt[finden]“, ist der „Baukasten so zu gestalten, dass [...] alle [F&E-] Phasen gleichermaßen unterstützt [werden].“<sup>24</sup> Seitens der Entwicklung ist das Einpflegen neuer Technologien und Funktionalitäten in den Baukasten besonders aufwandsarm zu gestalten. In diesem Zusammenhang wurde eine „klare Zuständigkeit für die Modulentwicklung und deren Dokumentation“<sup>25</sup> im Baukasten gefordert. Diese Punkte gilt es bereits bei der Strukturierung des Baukastens zu berücksichtigen.

---

<sup>21</sup> Interviewpartner 1 (Entwicklung, Mechanik), 29 (Management, Programm/Projekt)

<sup>22</sup> Interviewpartner 5 (Entwicklung, Hardware), 10 (Entwicklung, Software), 18 (Entwicklung, System)

<sup>23</sup> Interviewpartner 7 (Entwicklung, Hardware), 9 (Entwicklung, Hardware)

<sup>24</sup> Interviewpartner 27 (Management, Programm/Projekt)

<sup>25</sup> Interviewpartner 3 (Entwicklung, Mechanik)

Die erfassten Bedarfe wurden den Interviewpartnern anschließend in konsolidierter Form zur Verfügung gestellt. Folgende Themen wurden von den Befragten im Nachgang als potenzielle Schlüsselfaktoren<sup>26</sup> des Zielsystems an die Baukastenstrukturierung mechatronischer Steuergeräte formuliert:

- Vergleichbarkeit integraler und modularer Bausteinstrukturen
- Einfache Erweiterbarkeit und Anpassung der Struktur
- Effiziente Nutzung in der Frühen Phase der Steuergeräteentwicklung
- Klare Schnittstellen der Bausteine und daraus abgeleitet klare Zuständigkeiten
- Konsistenz und Transparenz im Prozess der Anforderungsableitung vom Gesamtsystem auf die Bausteine

Auf Basis der vorgestellten Studienergebnisse wurde ein Reference Model nach Blessing & Chakrabarti (2009) angefertigt (Abbildung 4.4). Dieses dient der Illustration des vom Forscher generierten Verständnisses über die ermittelten Zusammenhänge. Die identifizierten Schlüsselfaktoren sind dabei als besonders vielversprechende Ansatzpunkte für die methodische Unterstützung anzusehen. Neben den ermittelten Faktoren wird mit Hinblick auf die Validierung deren Einfluss auf die Akzeptanz der Strukturierungsmethode berücksichtigt. Diese wurde bereits in diversen Forschungsprojekten als zentrale Zielgröße zur Evaluation der erwirkten methodischen Unterstützung identifiziert. Eine detaillierte Auseinandersetzung mit dem Thema Methodenakzeptanz findet sich beispielsweise bei Scherer (2016). Abweichend von der Methode von Blessing & Chakrabarti wurden die ermittelten Herausforderungen zusätzlich ins Reference Model übernommen. Dies geschieht mit der Intention den Bezug zwischen Herausforderungen und Schlüsselfaktoren ersichtlich zu machen<sup>27</sup>.

---

<sup>26</sup> Begriff in Anlehnung an die DRM

<sup>27</sup> Aus Gründen der Übersichtlichkeit wurde das Reference Model in zwei Teile aufgeteilt.

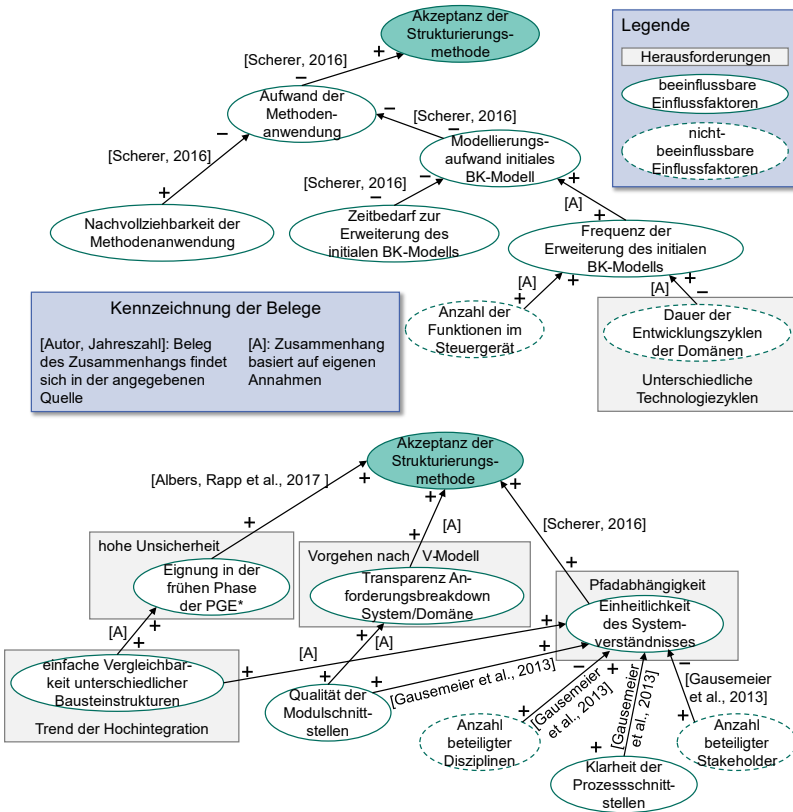


Abbildung 4.4: Reference Model der Strukturierungsmethode der vorliegenden Arbeit entsprechend der DRM von Blessing & Chakrabarti (2009)

**Unterstützung mittels Methoden des MBSE:**

*FF1.4 Welche Methoden des MBSE können zur Unterstützung der Baukastenstrukturierung und –anwendung herangezogen werden?*

Auf Basis der Schlüsselfaktoren wurden ausgewählte MBSE-Methoden hinsichtlich ihrer Nutzung als Anknüpfungspunkte der methodischen Unterstützung beleuchtet.



Durch den Fokus der vorliegenden Arbeit wurde die Unterstützung der Baukastenentwicklung auf die Frühe Phase der PGE mechatronischer Steuergeräte beschränkt. Mittels der Befragung konnte die in der Literaturrecherche aufgezeigte Forschungslücke an dieser Stelle durch Bedarfe aus der industriellen Praxis untermauert werden. Durch Bursac (2016) konnte im Zusammenhang mit der Baukastenentwicklung in der Frühen Phase bereits die Eignung der MBSE aufgezeigt werden. Aufbauend auf dessen Makroprozess, lassen sich die für mechatronische Steuergeräte spezifischen Herausforderungen ebenfalls zielgerichtet durch MBSE-gestützte Maßnahmen realisieren. Hinsichtlich der Vergleichbarkeit integraler und modularer Bausteine bietet der MBSE Ansatz durch die zeitgleiche Betrachtung von Anforderungen, Funktionen und Gestalt innerhalb eines durchgängigen Modells eine sinnvolle Unterstützung. Hierbei stellen bestehende Ansätze wie z.B. SYSMOD, SPES mögliche Anknüpfungspunkte dar. Die Berücksichtigung der negativ-emergenten, domänenübergreifenden Wechselwirkungen lässt sich hingegen ohne eine Form der Modellbildung nicht realisieren. Grundlage hierfür bieten beispielsweise flussbasierte Darstellungen von Funktionsstrukturen (Stone, 1997). Die Nutzung unterschiedlicher Sichten auf ein in sich konsistentes Modell bietet darüber hinaus die Möglichkeit komplexe Schnittstellen und deren Zuständigkeiten bedarfsgerecht zu illustrieren. Die Möglichkeit einer Teilautomatisierung MBSE-gestützter Modelle bietet vor allem hinsichtlich der Kompatibilitätsbetrachtung bzw. deduktiven Produktausleitung aus dem Baukasten große Potentiale (Powelske et al., 2018). Die Ergebnisse der deskriptiven Studie I sind in Abbildung 4.5 abschließend zusammengefasst.

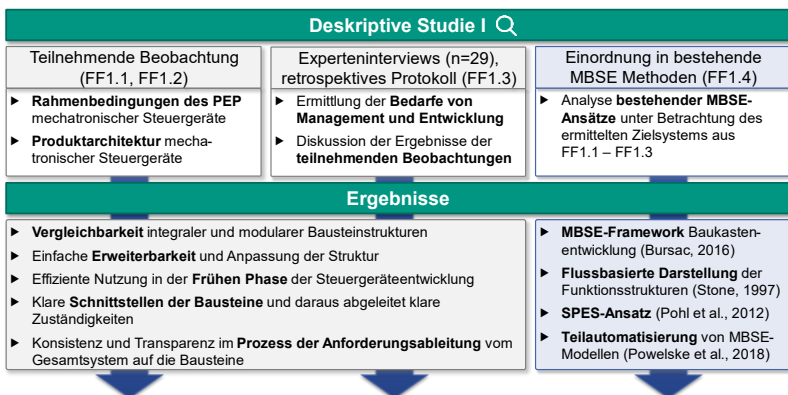


Abbildung 4.5: Ergebnisse der deskriptiven Studie I geordnet anhand der entsprechenden Teilforschungsfragen (FF1.1 – FF1.4)

## 4.2 Entwicklung einer Strukturierungsmethode

Aufbauend auf den Ergebnissen der deskriptiven Studie I wird nunmehr eine zweigeteilte präskriptive Studie zur Entwicklung und Implementierung der Strukturierungsmethode durchgeführt. Im Rahmen der präskriptiven Studie werden, die innerhalb der in Kapitel 3.1.3. abgeleiteten Forschungsfrage enthaltenen Aspekte der Baukastenstrukturierung beantwortet<sup>28</sup>:

*FF2: Wie sind die Strukturierung und Anwendung eines Baukastens mechatronischer Steuergeräte auf Basis des ermittelten Zielsystems mittels Methoden des MBSE zu gestalten und zu integrieren?*

In Bezug auf die Strukturierung des Baukastens besteht das Ziel, eine Methode zu entwickeln, die dazu in der Lage ist den aufgezeigten Erfordernissen mechatronischer Steuergeräte zu entsprechen. Indikatoren für eine zielgerichtete Unterstützung des Baukastenstrukturierungsprozesses sind die im Rahmen der deskriptiven Studie I ermittelten Schlüsselfaktoren<sup>29</sup>:

- Vergleichbarkeit integraler und modularer Baueinheiten
- Einfache Erweiterbarkeit und Anpassung der Struktur
- Effiziente Nutzung in der Frühen Phase der Steuergeräteentwicklung
- Klare Schnittstellen der Bausteine und daraus abgeleitet klare Zuständigkeiten
- Konsistenz und Transparenz im Prozess der Anforderungsableitung vom Gesamtsystem auf die Bausteine

Während die ersten beiden Punkte weitestgehend unabhängig von unternehmensinternen Randbedingungen adressiert werden können, bedürfen die Punkte drei bis fünf einer detaillierten Auseinandersetzung mit eben diesen. Es bietet sich an in einem ersten Schritt die Methoden des MBSE zu nutzen, um ein theoretisches Methodengerüst bzw. Framework zu erstellen. Dieses Framework soll zunächst frei von jeglicher Toolbindung in der Lage sein die Vergleichbarkeit modularer und integrierter Produktstrukturen zu ermöglichen, als auch die Frage nach einer geeigneten Strukturierungstiefe zu beantworten. In Zusammenhang mit

---

<sup>28</sup> Die im Rahmen von Kapitel 4.2 dargelegte Vorgehensweise ist zu Teilen bereits in Powelske et al. (2017) veröffentlicht.

<sup>29</sup> Begriff in Anlehnung an die DRM

der MBSE-gestützten Baukastenentwicklung bedeutet dies konkret ein entsprechendes Vorgehen für die induktive bzw. deduktive Überführung von Modellen mechatronischer Steuergeräte zu entwickeln. Das Framework wird anschließend an einem realen Beispiel aus dem Umfeld mechatronischer Steuergeräte angewendet. Hierbei wird es um die fehlenden Punkte ergänzt und mit seiner Umsetzung in der vorgegebenen Toolumgebung vorgestellt. Die theoretische Herleitung der Methode und deren praktische Anwendung inklusive Implementierung in der Untersuchungsumgebung werden demnach sequenziell erfolgen. Entsprechend des sequenziellen Vorgehens von Methodenentwicklung und darauffolgender Implementierung wird die initiale Forschungsfrage zunächst in zwei Teilforschungsfragen untergliedert<sup>30</sup>. Deren Beantwortung erstreckt sich ebenfalls sequenziell auf das vorliegende und das folgende Unterkapitel (Kapitel 4.3).

*FF2.1: Wie werden die Baukastenstrukturierung und –anwendung mechatronischer Steuergeräte mittels MBSE-Methoden entsprechend gestaltet?*

In puncto Vergleichbarkeit integraler und modularer Produkte erscheint es naheliegend sich mit der Analyse der zugrundeliegenden Produktarchitektur zu befassen. Gemäß der im Grundlagenkapitel gegebenen Definition (vgl. Kapitel 2.2.4) besteht die Produktarchitektur aus der Funktionsstruktur, der Produktstruktur und aus sämtlichen Transformationsbeziehungen zwischen den beiden Strukturen. Für die Baukastenstrukturierung ist es notwendig den Fokus vom Gesamtprodukt auf dessen Teilsysteme zu richten. Dementsprechend werden zur Vergleichbarkeit der Teilsysteme hauptsächlich Ausschnitte der Funktions- und Produktstruktur untersucht. Modulare und integrierte Teilsysteme unterscheiden sich anhand ihrer Zuordnung der Elemente der Funktionsstruktur zu technischen Bauteilen. Während bei der integralen Herangehensweise versucht wird, möglichst viele Funktionen auf möglichst wenige Bauteile zu aggregieren, sieht der modulare Ansatz eine möglichst feingranulare Aufteilung der Funktionen auf mehrere Bauteile vor (Abbildung 4.6).

---

<sup>30</sup> Die Implementierung wird anhand einer generalisierten Methode vorgestellt, die anschließend auf die, innerhalb der Untersuchungsumgebung vorliegende Toolumgebung, angepasst wird. Die unternehmensunabhängige Übertragbarkeit des Ansatzes ist somit sichergestellt.

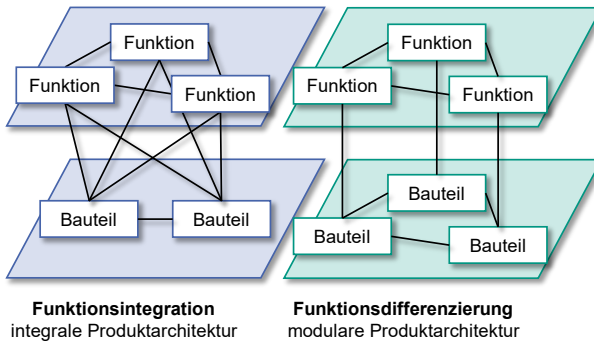


Abbildung 4.6: Prinzipien der Funktionsintegration und Funktionsdifferenzierung abhängig von der gewählten Art der Produktarchitektur (Ponn & Lindemann, 2008, S. 149)

Im folgenden Abschnitt wird an einem Beispiel mechatronische Steuergeräte ein zentrales Paradigma für die Baukastenstrukturierung im Rahmen der vorliegenden Arbeit abgeleitet:

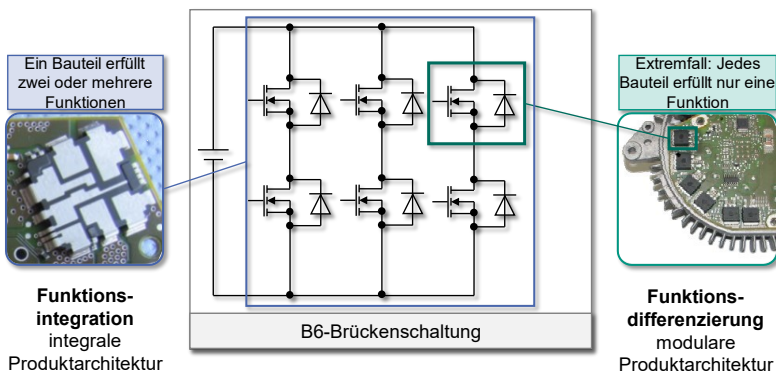


Abbildung 4.7: Erläuterung der Architekturprinzipien integral und modular am Beispiel der Gestaltung einer Endstufe. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022)

Abbildung 4.7 zeigt zwei Konzepte zur Realisierung einer Leistungsendstufe. Im linken Teil der Abbildung ist das Ergebnis eines hochintegrierten Ansatzes dargestellt. Hierbei ist die Endstufenfunktionalität inklusive der Phasenstrommessung in einem Bauteil realisiert. Im rechten Bildausschnitt findet sich eine modular aufgebaute Umsetzung desselben Funktionsumfangs. Beiden Realisierungen liegt ein vergleichbares technisches Lösungsprinzip, das einer B6 Brückenschaltung zugrunde. An der Umsetzung der Endstufenfunktionalität sind in beiden Varianten hauptsächlich 6 MOSFETs beteiligt, die Strommessung erfolgt durch hochgenaue Shuntwiderstände. Die baulichen Unterschiede bei gleichem funktionalem Umfang und vergleichbarem Lösungsprinzip lassen daher folgenden Schluss zu. Ein Vergleich integraler und modularer Strukturen kann lediglich anhand des zugrundeliegenden Funktionsumfangs, jedoch niemals anhand der technischen Realisierung widerspruchsfrei erfolgen. Als zentrales Paradigma der angestrebten Baukastenstrukturierung gilt folglich:

*Die Strukturierung des Baukastens muss auf Basis eines vorrangig **funktionszentrierten Ansatzes** erfolgen.*

Vorrangig deshalb, da der Vergleich technischer Realisierungen v.a. hinsichtlich quantitativer Unterschiede bei der Strukturierung nicht außen vorgelassen werden kann. Denken wir beispielsweise an den Vergleich zweier integral gestalteter Leistungsendstufen. Diese können entsprechend der Regeln einer Baureihe bei gleicher Funktion und Bauart in Größe und Leistung skaliert werden. Die Baukastenstrukturierung muss eine Unterscheidung auf Basis der Quantifizierung von Parametern selbstredend unterstützen.

Verbleiben wir beim Beispiel der Leistungsendstufen, um eine weitere bedeutende Prämisse der Baukastenstrukturierung aufzuzeigen:

Im Falle der integralen Lösung bleibt der Zusammenhang zwischen den Teilfunktionen der Endstufe und der Strommessung von außen für den Betrachter verborgen. Bei der modularen Realisierung hingegen offenbart die Verschaltung der einzelnen technischen Elemente die beabsichtigte Funktionalität. Den Grundlagen über Funktionsstrukturen (vgl. Kapitel 2.2.4) entnehmen wir, dass Funktionen ausgehend von der Gesamtfunktionalität eines Produktes über verschiedene Hierarchiestufen in feingranulare Teilfunktionen dekomponiert werden. Dies geschieht so lange, bis eine technische Realisierung der Funktionalität vorgenommen werden kann. Die Tiefe der funktionalen Dekomposition und die Einordnung des realisierenden Bauteils als Element modularer oder integraler Strukturierung ist eng mit dem Fokus und der Betrachtungsebene der jeweiligen Entwicklung verknüpft. Für den Schaltungsentwickler im oben aufgeführten

Sachverhalt stellt der MOSFET ein integral realisiertes Bauteil dar, welches als Element einer modularen Lösung zur Erfüllung eines übergeordneten Funktionsumfangs genutzt wird. Für den Entwickler des MOSFET hingegen deckt dieser unzählige weitere Teilfunktionen ab. Seine Teilsysteme können entsprechend des fraktalen Charakters der horizontalen Produktstrukturierung wiederum modular oder integral realisiert werden.

Es lässt sich feststellen, dass die Transformation von Funktions- in Produktstruktur bei integralen Strukturen abhängig von der individuellen Betrachtung auf einer höheren hierarchischen Ebene erfolgt. Nach Bursac ist der Syntheseprozess des Baukastens bzw. des Baukastenmodells maßgeblich durch die Induktion von Produktmodellen zu Produktreferenz- und Baukastenreferenzmodellen beeinflusst (Bursac, 2016). Durch diesen Vorgang wird die Granularität der späteren Baukastenelemente festgelegt. Hieraus ergeben sich zwei Forderungen:

- Damit die Induktion erfolgreich bewältigt werden kann, muss die Vergleichbarkeit der initialen Produktmodelle in Bezug auf deren hierarchische Dekomposition der Funktions- und Produktstruktur geschaffen werden.
- Aus Sicht der Baukastenstrukturierung ist es entscheidend die hierarchische Dekomposition von Funktions- und Produktstruktur so tief zu berücksichtigen, dass die gewünschten Funktionsumfänge hinsichtlich der Systemgrenze der ausgewählten Produkte sinnvoll abgebildet werden können. Dies bedeutet am Beispiel der Leistungsendstufe konkret, dass Funktionen innerhalb des MOSFET für einen Baukasten auf Bauteilebene essenziell, für einen Baukasten mechatronischer Steuergeräte jedoch von geringer Bedeutung sein können.

Führt man die bisherigen Erkenntnisse zusammen, bedarf die Baukastenstrukturierung für Steuergeräte im Einsatz in Kleinantrieben der Anwendung eines vorrangig funktionszentrierten Ansatzes. Dieser ist bis zu einer geeigneten Strukturierungstiefe fortzuführen. In der Literatur (vgl. Kapitel 2.4.1) existieren in Bezug auf die Strukturierungstiefe zahlreiche Überlegungen für ein systematisches Vorgehen. Im folgenden Teilkapitel werden drei Kriterien formuliert, welche verschiedene Ansätze aus der Literatur zusammenführen. Ziel dieser Kriterien ist es dem Anwender einen Leitfaden zur Verfügung zu stellen, mit dessen Hilfe eine geeignete Strukturierungstiefe für den Baukasten festgelegt werden kann.

## 4.2.1 Ableitung der Strukturierungskriterien

Die Kriterien zur Definition der Strukturierungstiefe (Abbildung 4.8) sind bei Betrachtung der Produktstrukturen für den Baukasten vorgesehener Produkte sequenziell zu durchlaufen.

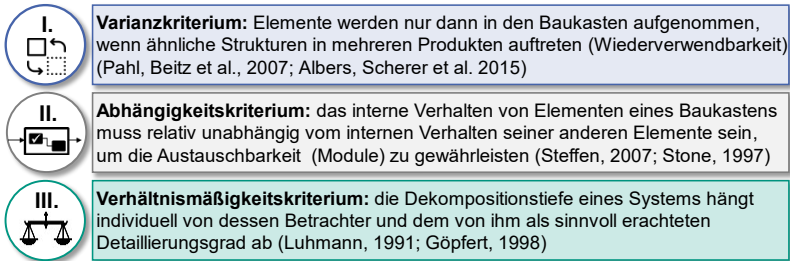


Abbildung 4.8: Kriterien der Baukastenstrukturierung für Steuergeräte im Einsatz in Kleinantrieben im Automobilbereich (Powelske et al., 2017)

Sie sind in großem Maße ausschlaggebend für die Gestaltung der Produktreferenz- bzw. Baukastenreferenzmodelle und gelten als ausschlaggebend für die Einführungen tieferer Strukturierungsebenen. Für die Anwendung der Kriterien gilt: Eine tiefere Strukturierungsebene wird nur bei Erfüllung aller Kriterien eingeführt. Die einzelnen Kriterien werden im Folgenden vorgestellt:

- **Varianzkriterium:** Elemente werden nur dann in den Baukasten aufgenommen, wenn ähnliche Strukturen in mehreren Produkten auftreten (Wiederverwendbarkeit) (Pahl, Beitz et al., 2007; Albers, Scherer et al. 2015)

Dem Grundgedanken der Baukastenentwicklung folgend (vgl. Kapitel 2.4) werden über das Varianzkriterium nur dann Elemente in den Baukasten aufgenommen, wenn deren durchgängige Wiederverwendbarkeit erkennbar ist. Dies gilt sowohl für bereits bestehende als auch in der Entwicklung befindliche Produkte. Im Kontext der Steuergeräteentwicklung wird demnach eine Leistungsendstufe z.B. in Form mehrerer MOSFETs, nur dann als Modul in den Baukasten aufgenommen, wenn diese bereits in einigen Produkten realisiert wurde. Applikationsspezifische Lösungen für einzelne Module sind jedoch trotzdem nicht völlig unabhängig von den Baukastenelementen. Nicht-Bausteine müssen bei deren Verwendung geeignete Schnittstellen zu den sie umgebenden Bausteinen aufweisen.

- **Abhängigkeitskriterium:** das interne Verhalten von Elementen eines Baukastens muss relativ unabhängig vom internen Verhalten seiner anderen Elemente sein, um die Austauschbarkeit<sup>31</sup> (Module) zu gewährleisten (Steffen, 2007; Stone, 1997)

Das Abhängigkeitskriterium begründet sich auf dem Kerngedanken der Baukastenentwicklung: durch die Austauschbarkeit von Modulen neue Funktionalitäten bereit stellen zu können. Dieser Grundgedanke ist für Baukästen aller Art generalisierbar. Aus dem Grundlagenkapitel der mechatronischen Steuergeräte (vgl. Kapitel 2.1.1) geht hervor, dass sich die domänenübergreifenden Wechselwirkungen häufig negativ auf das gewünschte Systemverhalten auswirken. Bei der Definition der Modulgrenzen ist darauf zu achten, dass starke, häufig nicht völlig ergründete Abhängigkeiten innerhalb eines Moduls verbleiben. Zwischen verschiedenen Modulen ist es von Vorteil möglichst wenige, weitestgehend beschreibbare Abhängigkeiten zu belassen. Die im Rahmen dieser Arbeit getätigten Untersuchungen zeigen, dass sich die Modulgrenzen im Rahmen der Baukastenentwicklung mechatronischer Steuergeräte, nur selten an der Beteiligung unterschiedlicher Domänen orientieren. Gerade dort, wo verschiedene Domänen aufeinandertreffen, ist eine ganzheitliche Betrachtung von Nöten, um eine geeignete Definition der Modulgrenzen vorzunehmen. Das folgende Beispiel soll diesen Sachverhalt erklären (Abbildung 4.9):

---

<sup>31</sup> Diese ist in Bezug auf die Definition eines Moduls nach Albers, Scherer et al. (2015) zu verstehen



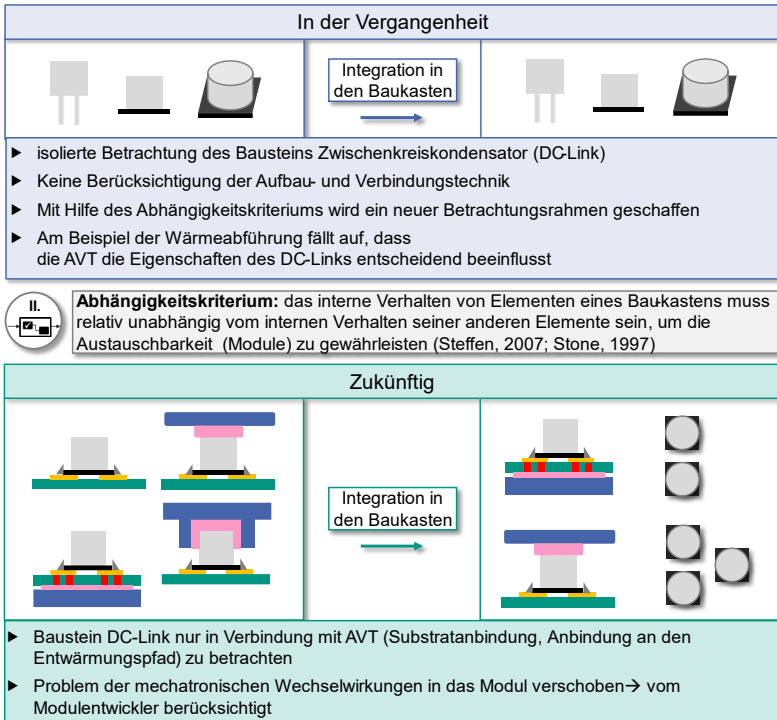


Abbildung 4.9: Veränderungen in der Modulbildung im Rahmen der Baukastenentwicklung mechatronischer Steuergeräte im Einsatz von Kleinantrieben im Automobilbereich, durch ganzheitliche Betrachtung am Beispiel des Zwischenkreiskondensators nach Powelske et al. (2017)

Die Hauptaufgabe des Zwischenkreiskondensators liegt neben seiner Filterwirkung in der Bereitstellung von Energie für die Endstufe. Diese Aufgabe wird maßgeblich durch die elektrischen Eigenschaften des Kondensators bestimmt. Aufgrund der getakteten Stromentnahme finden eine Vielzahl an Auf- und Entladevorgängen statt. In Abhängigkeit der Frequenz und des Innenwiderstandes des Kondensators kommt es zur Eigenerwärmung des Bauteils. Diese führt wiederum zu einer Verschlechterung der elektrischen Eigenschaften und gleichzeitig zu einer mechanischen Belastung, die sich in einer Reduktion der Lebensdauer des Bauteils äußert. Um diese Lebensdauer nicht zu gefährden kann die Verlustenergie durch eine Parallelschal-

tung mehrerer Kondensatoren gesenkt werden. Dies führt allerdings zu einem erhöhten Bauraumbedarf und höheren Kosten. Die Eigenerwärmung kann ebenfalls auf anderem Wege durch ein geeignetes Kühlkonzept reguliert werden. Die Erarbeitung des Kühlkonzeptes liegt jedoch in der Domäne der Aufbau- und Verbindungstechnik. Je nach Art des Kühlkonzeptes sind mehrere Bauteile mit deren elektrischen und mechanischen Eigenschaften davon betroffen und müssen auf die spezifischen Anforderungen des Systems individuell gestaltet werden. In der Vergangenheit wurde der elektrische Pfad des Kondensators unabhängig vom thermischen Pfad betrachtet, damit jede der Domänen unabhängig voneinander ihrer Arbeit nachgehen konnte. Folge waren zwei verschiedene Module mit sehr komplexen Schnittstellen. Die Frage nach der Kompatibilität der Modullösungen muss daher immer auf der Ebene des Gesamtbaukastens beantwortet werden, was langwierige Iterationen mit sich bringt. Mittels Nutzung des Abhängigkeitskriteriums wird die Modulgrenze von einem Kondensator auf die Parallelschaltung von Kondensatoren inklusive deren AVT ausgeweitet. Die bestehenden Abhängigkeiten werden damit in das Modul verschoben. Die letztendliche Konzeptauswahl kann folglich unabhängig vom restlichen System getroffen werden was, in einer Vereinfachung der Abhängigkeitsstruktur auf Baukastenebene resultiert.

- **Verhältnismäßigkeitskriterium:** die Dekompositionstiefe eines Systems hängt individuell von dessen Betrachter und dem von ihm als sinnvoll erachteten Detaillierungsgrad ab (Luhmann, 1991; Göpfert, 1998)

Das Verhältnismäßigkeitskriterium stellt das wohl kontroverseste der drei Kriterien dar. Während sich die Ingenieurwelt stets darum bemüht, Dinge zuverlässig quantifizieren zu können, zeichnet sich ab, dass die notwendige Strukturierungstiefe von Baukästen individuell vom Kontext der abzuleitenden Produkte abhängt. Tatsächlich sind Expertenaussagen über die Anzahl notwendiger Dekompositionsebenen eines Systems zu entnehmen, dass neben den Produkteigenschaften viele weitere Faktoren wie beispielsweise vorhandene Kompetenzen im Unternehmen oder der Aufbau der Organisationsstruktur eine Rolle spielen (Göpfert, 1998). Eine klare Vorgabe, dass beispielsweise die Betrachtung von zwei Hierarchieebenen für die Bildung eines Produkt- bzw. Baukastenreferenzmodells ausreichend ist, kann daher nicht für die Allgemeinheit der mechatronischen Steuergeräte ausgegeben werden. Vielmehr muss individuell geprüft werden, ob sich eine weitere Untergliederung als für den Baukasten relevant herausstellt oder nicht. Gleichzeitig erscheint es naheliegend einen Baukasten mechatronischer Steuergeräte nicht auf Ebene einzelner Widerstände aufzubauen. Da die ersten beiden Kriterien eher makroskopische Abbruchbedingungen für die weitere Strukturierung bedeuten allerdings klarer in der Bewertung sind, wird das

Verhältnismäßigkeitskriterium als finale Entscheidung ans Ende der Betrachtung gestellt.

Die abgeleiteten Kriterien bilden zusammen mit der im folgenden Unterkapitel hergeleiteten Methode das initiale Framework zur Baukastenstrukturierung.

#### **4.2.2 Herleitung der Strukturierungsmethode**

Wie bereits mehrfach motiviert, bedarf die Baukastenstrukturierung mechatronischer Steuergeräte der funktionalen Vergleichbarkeit der Produkte, des zugehörigen Baukastens und des abzubildenden Portfolios. Sämtliche Schritte der Strukturierung sind innerhalb des von Bursac (2016) motivierten Zyklus zur induktiven und deduktiven Modellüberführung (vgl. Kapitel 2.4.2) zu betrachten. Die hergeleiteten Kriterien und die Strukturierungsmethode, bilden dabei die konkrete Vorgehensweise beim Überführungsprozess. In Zusammenhang mit der Herleitung der Strukturierungsmethode spielt die gleichzeitige Betrachtung von Methoden zur Bereitstellung und Kategorisierung von Erfahrungswissen und deren aktive Nutzung im Induktionsprozess eine ausschlaggebende Rolle. Das iPeM bietet hierzu eine besonders geeignete Möglichkeit der Beschreibung. Demnach können die nachfolgenden Vorgänge als stetiges Wechselspiel der Basis-Aktivität „Wissen managen“ und der Produktentwicklungsaktivität „Prinzip und Gestalt modellieren“ betrachtet werden.

##### **Ermittlung einer Referenz-Funktionsstruktur:**

Als Startpunkt der Betrachtung wird die von Bursac (2016) geforderte Verfügbarkeit von Produktmodellen für die spätere Ableitung von Produktreferenzmodellen, zunächst bewusst auf die Betrachtung der Funktionsstrukturen reduziert. Konkret ist damit eine Zusammenführung der einzelnen Funktionsstrukturen der Produkte zu einer Referenz-Funktionsstruktur gemeint. Dieser Vorgang gilt in Bezug auf die Baukastenstrukturierung der vorliegenden Arbeit, als notwendige Vorarbeit ist jedoch kein Teil der Strukturierungsmethode selbst. Im Folgenden wird eine knappe Übersicht der Bildung einer solchen Referenz-Funktionsstruktur gegeben. Hierbei wird die notwendige Vorarbeit der Anwendung der Strukturierungsmethode beschrieben.

Die aktuelle Forschung im Bereich des MBSE zeigt, dass an oberster Stelle der Funktionsstruktur eines Produktes nicht nur eine Funktion, sondern mehrere gleichberechtigte Funktionen nebeneinander existieren können (Pohl, 2012). Diese Funktionen ergeben sich aus der Analyse des initialen Zielsystems. Ihre Inputs und Outputs sind in der Regel direkt an der Systemgrenze sichtbar und spiegeln die vom

Nutzer gewünschte Funktionalität wider. Die SPES Methodik spricht deshalb von sogenannten User-Funktionen (Pohl, 2012) (Abbildung 4.10). In Zusammenhang mit der Steuergeräteentwicklung ist als User-Funktion die Überführung der elektrischen Eingangsleistung in eine elektrische Ausgangsleistung mit einer entsprechenden Signalform zu verstehen.

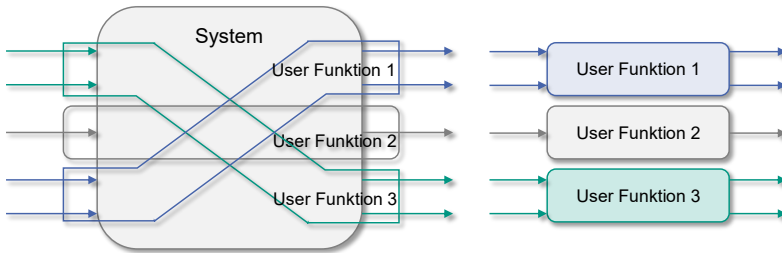


Abbildung 4.10: Prinzip der Bildung einer User-Funktion in Anlehnung an Pohl (2012)

Bei der Formulierung der User-Funktionen ist darauf zu achten, dass diese untereinander möglichst geringe Interaktionen aufweisen, damit der anschließende Dekompositionsprozess dieser Funktionen weitgehend unabhängig erfolgen kann. In der Theorie werden die Funktionen lösungsoffen bis zur Domänenebene heruntergebrochen, genau so weit, bis entsprechend der Definition der Produktarchitektur (vgl. Kapitel 2.2.4) passende technische Umsetzungen gefunden werden können. Der Funktionsumfang des Steuergerätes, wird hierbei z.B. so weit heruntergebrochen, bis die Funktionalität an deren Umsetzung ein hauptsächlich Mikrocontroller beteiligt ist, identifiziert werden kann. Die Zusammenführung der einzelnen Produktfunktionsstrukturen setzt voraus, dass diese nach einem vergleichbaren Prinzip heruntergebrochen wurden. Ist dies nicht der Fall, bietet sich das Anfertigen einer Hüllkurve aller für den Baukasten relevanten Produktanwendungsfälle an. Die Hüllkurve bietet anschließend die Möglichkeit eine Gesamtheit der für das Portfolio gewünschten User-Funktionen abzuleiten. Auf dieser Basis kann hiernach die beschriebene Dekomposition auf Domänenebene stattfinden. Eine auf dem geschilderten Wege erzeugte Referenz-Funktionsstruktur entspricht der für die Entwicklung mechatronischer Steuergeräte charakteristischen, lösungsneutralen Denkweise (vgl. Kapitel 2.2.3). Aus den Überlegungen zum Funktions-Gestalt-Übergang (vgl. Kapitel 2.2.5) geht allerdings klar hervor, dass eine von der technischen Umsetzung isolierte Funktionsbetrachtung einer ganzheitlichen, mechatronischen Sichtweise widerspricht. Dementsprechend bildet

die Referenz-Funktionsstruktur, wie eingangs erwähnt, lediglich eine Vorarbeit in Form eines Kataloges. Dieser erweist sich bei der Betrachtung der technischen Lösungen und der eigentlichen Baukastenstrukturierung als bedeutendes Hilfsmittel zur Schaffung einer Vergleichbarkeit modularer und integraler Strukturen.

### Gruppierung der Funktionen entsprechend der technischen Umsetzung:

Um einen einheitlichen Einstiegspunkt für die Baukastenstrukturierung festlegen zu können, werden alle User-Funktionen der lösungsneutralen Referenz-Funktionsstruktur zu einem sogenannten Funktionscluster kombiniert. Dieses entspricht auf oberster Ebene dem gesammelten Funktionsumfang einer über den Baukasten abzubildenden ECU, berücksichtigt demnach auch optionale Funktionsumfänge. Ausgehend von diesem Funktionscluster wird, unter Einbezug des technischen Anteils der Produktmodelle und der in Kapitel 4.2.1 abgeleiteten Kriterien, eine funktionale Restrukturierung vorgenommen.

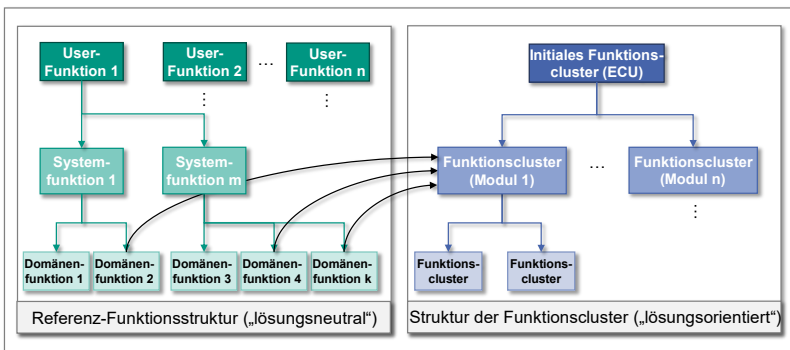


Abbildung 4.11: Gruppierung von Funktionen zu einem Funktionscluster entsprechend der späteren technischen Umsetzung.

Abbildung 4.11 zeigt das Zusammenspiel der Referenz-Funktionsstruktur und der auf Basis der technischen Realisierungen abgeleiteten Funktionscluster. Ein Funktionscluster definiert sich analog zum Funktionsbegriff (vgl. Kapitel 2.2.1) über die Beziehungen zwischen dessen Inputs und Outputs. Der Begriff des Funktionsclusters bietet sich an dieser Stelle an, da sich sein Funktionsinhalt aus der Clusterung einzelner Funktionen der Referenz-Funktionsstruktur bildet.

**Darstellung der Wirkzusammenhänge im Funktionsstrukturausschnitt (FSA):**

Wird im späteren Verlauf eine tiefere Strukturierungsebene eingeführt<sup>32</sup>, lassen sich die übergeordneten Funktionsumfänge wiederum als Kombination von Funktionsclustern und deren Verbindung bspw. über Stoff-, Informations- und Energieflüsse darstellen (horizontale Beziehungen).

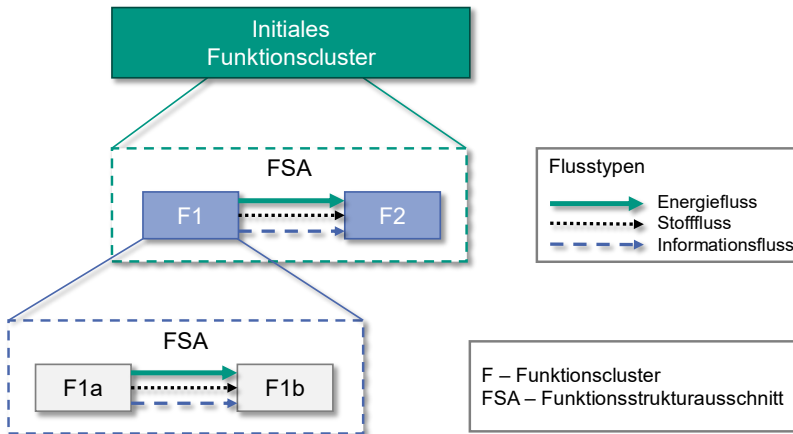


Abbildung 4.12: Illustration der Wirkzusammenhänge von Funktionsclustern über Funktionsstrukturausschnitte (FSAs) innerhalb der Funktionsstruktur mechatronischer Steuergeräte. F1, F2 sowie F1a und F2b stellen dabei Funktionscluster unterschiedlicher hierarchischer Strukturierungstiefe dar.

Die Darstellung des Zusammenwirkens der Funktionscluster zwischen deren Ein- und Ausgängen erfolgt in einem sogenannten Funktionsstrukturausschnitt (FSA) (Abbildung 4.12). Dieser ermöglicht es mit Hinblick auf den Einsatz im Rahmen mechatronischer Steuergeräte jedwede Art von Wechselwirkungen<sup>33</sup> im Strukturierungsprozess zu berücksichtigen. Zur Modellierung der Wechselwirkungen zwischen den Funktionsclustern wurde eine Methode entwickelt,

<sup>32</sup> Dies ist auf die korrekte Anwendung der Strukturierungskriterien zurückzuführen, welche in den folgenden Kapiteln praxisnah demonstriert wird.

<sup>33</sup> Hierbei sind explizit auch die domänenübergreifenden Wechselwirkungen gemeint

welche auf den Einsatz in der Frühen Phase der PGE optimiert ist. Diese Methode wird im Rahmen von Kapitel 4.3 genauer beleuchtet. Für die Erstellung eines FSA gilt, dessen Eingänge und Ausgänge sowie der durch ihn abgebildete Funktionsumfang sind identisch mit denen des übergeordneten Funktionsclusters.

### **Anbindung der technischen Lösungen an die Funktionscluster:**

Die mittels der Strukturierungskriterien gestützte Dekomposition der Funktionscluster bezieht, im Gegensatz zu den weitestgehend lösungsneutralen Ansätzen beim Zerlegen von Funktionen, direkt die technische Umsetzung des Funktionsumfangs mit ein. In anderen Worten gibt sie die Struktur zur Berücksichtigung der späteren Baukastenelemente, sogenannter Bausteine vor. Für die erste Ebene eingeführter FSA's gilt an dieser Stelle eine Besonderheit. Das initiale Funktionscluster vereint bekanntermaßen alle im vorgesehenen Portfolio beinhalteten Funktionalitäten in sich. Diese sind allerdings nicht in allen Produkten erforderlich. Dementsprechend bildet die oberste Ebene der FSA's gleichzeitig die Hüllkurve aller technischen Realisierungen und macht diese vergleichbar. Im Zusammenspiel mit den User-Funktionen entspricht diese Ebene den Produktreferenzmodellen. Für jedes Funktionscluster besteht die Möglichkeit, dass dessen Funktionsumfang in Form einer hochintegrierten technischen Lösung in einem der betrachteten Produkte umgesetzt ist.<sup>34</sup> An dieser Stelle eignet sich das Beispiel der hochintegrierten Leistungsendstufe im Vergleich zum diskreten Aufbau anhand von MOSFETs aus Abbildung 4.6. In der Entwicklung befindliche Technologien sind im Sinne des Referenzsystembegriffs der PGE ausdrücklich mit in die Betrachtung einzubeziehen. Allerdings ist dies nur dann sinnvoll, sofern deren Funktionsumfang, den durch das Cluster vorgegebenen eins zu eins abbildet. Ist eine technische Realisierung entweder im Produkt oder über die Entwicklung verfügbar wird diese als sogenannter integraler Baustein in die Struktur aufgenommen (Abbildung 4.13). Mit zunehmender Anzahl an analysierten Produkten existieren für die Funktionscluster innerhalb des zu strukturierenden Baukastens mehrere Lösungen; wir sprechen daher von integralen Bausteinvarianten. Diese unterscheiden sich lediglich anhand ihrer

---

<sup>34</sup> Präziser: für jedes Funktionscluster, welches sich unterhalb des initial gebildeten befindet, muss eine integrale Bausteinvariante existieren.

Parametrisierung, jedoch nicht anhand ihrer funktionalen Ein- und Ausgangsgrößen.<sup>35</sup>

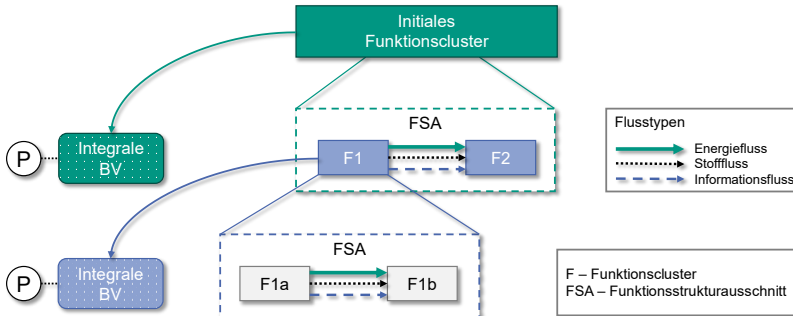


Abbildung 4.13: Anbindung integraler Bausteinvarianten an die Funktionscluster, entsprechend der technischen Umsetzung in realen Produkten.

### Dokumentation von Bausteinvarianten mittels Steckbriefen:

Zur Dokumentation der Bausteinvarianten werden diese ähnlich einem technischen Datenblatt mit Parametern versehen (Abbildung 4.14). Die Parameter umfassen dabei sowohl technisch-funktionale Größen wie beispielsweise zulässige Input- und Outputströme, als auch gestaltabhängige. Eingangs- und Ausgangsgrößen werden in Form von zulässigen Intervallen beschrieben, deren Zusammenhänge in Form von Messkurven, Wertetabellen oder ähnlichem. Ist eine Bausteinvariante bereits innerhalb eines oder mehrerer Produkte eingesetzt können den allgemeinen Bausteingrenzen die Parameter aus den spezifischen Produktkontexten zur Seite gestellt werden. Dadurch wird es möglich bisherige Erfahrungen aus den Produktentstehungsprozessen entsprechend der PGE gezielt für den Baukasten nutzbar zu machen. Allem voran stellen konkrete Schnittstellen- oder Platzierungsinformationen einen bedeutenden Mehrwert gegenüber reinen Datenblattwerten dar, da sie einen Überblick über das tatsächliche Verhalten des

<sup>35</sup> Der Begriff des Parameters ist hierbei als Oberbegriff für Merkmale und Eigenschaften technischer Systeme und deren Subsysteme zu verstehen (Zingel, 2013).



Bausteins in vergleichbaren Systemen geben. So können bei der Betrachtung eines Zwischenkreiskondensatorbausteins beispielsweise die Anzahl der verwendeten Kondensatoren, die Anbindung ins Kühlkonzept, geeignete Wärmeleitmaterialien oder die Platzierung auf der Leiterplatte als hilfreiche Informationen mitgegeben werden. Durch die direkte Verbindung der Funktionscluster mit ihrer technischen Umsetzung in Form der Bausteinvarianten wird es möglich die unterschiedlichen domänenspezifischen Entwicklungsartefakte zu harmonisieren. Dies entspricht dem in der Entwicklung mechatronischer Steuergeräte geforderten ganzheitlichen Ansatz. Neben den technischen Inhalten können ebenfalls Erkenntnisse über die Planung und Durchführung von Aktivitäten in Zusammenhang mit den Bausteinen für die Wiederverwendung von großem Interesse sein. Das iPeM bietet an dieser Stelle mit seinem Phasenmodell eine ideale Möglichkeit Referenzprozesse zu erfassen und diese für die Entwicklung neuer Produkt- bzw. Baustein generationen verfügbar zu machen.

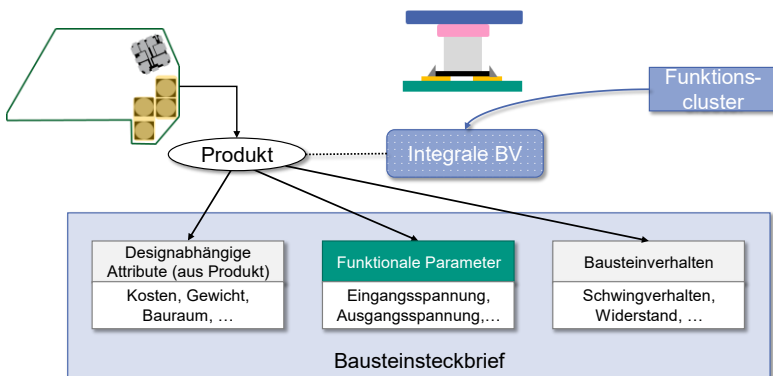


Abbildung 4.14: Darstellung eines Bausteinsteckbriefs der integralen Bausteinvariante eines Zwischenkreiskondensators inklusive verwendeter Elemente der Aufbau und Verbindungstechnik. Enthalten sind neben designabhängigen Attributen (z.B. Gewicht), funktionale Parameter (z.B. Spannung) und Informationen über das Bausteinverhalten.

### Einführung modularer Bausteinvarianten:

Während die Funktionscluster als Träger der integralen Bausteinvarianten dienen, stellen die Funktionsstrukturausschnitte den Aufhängepunkt für die zweite Klasse

der Bausteinvarianten, der sogenannten modularen Bausteinvarianten dar. Diese entstehen aus der Kombination integraler Bausteinvarianten der im FSA enthaltenen Funktionscluster. Hierbei werden die integralen Bausteinvarianten entsprechend der im FSA vorgegebenen Verbindungen auf Parameterebene zusammengeführt.<sup>36</sup> Auf diesem Wege entstandene modulare Bausteinvarianten bilden den Funktionsumfang des FSA ab und weisen dementsprechende Schnittstellen auf (Abbildung 4.15). Da ein FSA bekanntermaßen seinem übergeordneten Funktionscluster hinsichtlich Funktionsumfang und Schnittstellen gleicht, werden die modularen Bausteinvarianten ebenso direkt mit den übergeordneten integralen Vertretern vergleichbar.

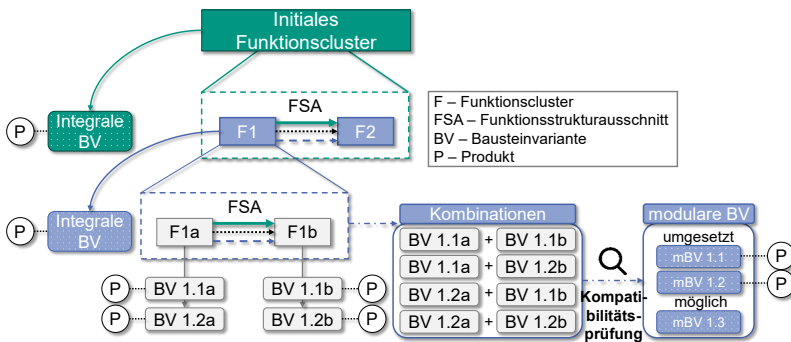


Abbildung 4.15: Bildung modularer Bausteinvarianten auf Basis der motivierten Struktur von Funktionsclustern und Funktionsstrukturausschnitten. Die aus der Kombination entstandenen modularen BV (mBV1.1 – mBV1.3) entsprechen dabei dem Funktionsumfang der integralen BV des übergeordneten Funktionsclusters (F1).

### Unterscheidung umgesetzter und möglicher modularer Bausteinvarianten:

Bei der Betrachtung modularer Bausteinvarianten gilt es grundsätzlich zwei verschiedene Arten zu unterscheiden. Jene, die bereits in Produkten eingesetzt

<sup>36</sup> Der Funktionsstrukturausschnitt kann damit als funktionaler Bauplan der technischen Umsetzung der untergeordneten Ebene betrachtet werden.

wurden (umgesetzte mBV) und Varianten, die sich zusätzlich aus der Kombinatorik ergeben (mögliche mBV). Zur Erläuterung greifen wir erneut auf das Beispiel zur Auswahl einer Leistungsendstufe aus Kapitel 4.2 zurück. Gegeben sei ein FSA der einen übergeordneten Funktionsumfang in die Funktionscluster der Endstufenfunktionalität und der Strommessung zerlegt.<sup>37</sup> (Abbildung 4.16)

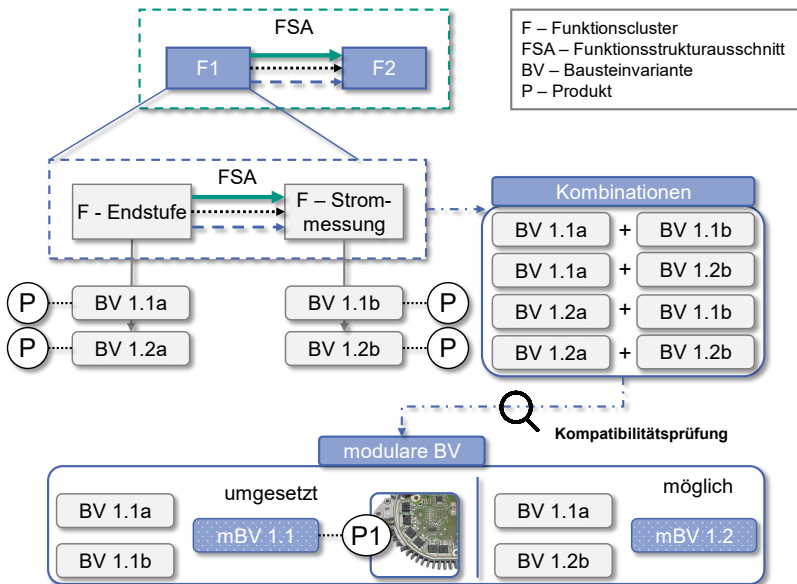


Abbildung 4.16: Umgesetzte (mBV 1.1) und mögliche (mBV 1.2) modulare Bausteinvarianten am Beispiel einer Endstufe. Die umgesetzte modulare BV enthält zusätzliche Informationen aus deren Einsatz in realen Produkten (P1).

Sowohl für die Endstufenfunktionalität als auch die Strommessung existieren weiterhin jeweils zwei integrale Bausteinvarianten (BV1.1a und BV1.2a für die Endstufe sowie BV1.1b und BV1.2b für die Strommessung). Aus der Kombinatorik sind in Summe vier modulare Bausteinvarianten denkbar. Sind nun BV1.1a und BV1.1b im Produkt P1 bereits zusammen realisiert wird die entstandene modulare

<sup>37</sup> Beide sind aus der Zuordnung von Funktionen der Referenz-Funktionsstruktur in Abhängigkeit der technischen Realisierungen abgeleitet worden.

Bausteinvariante mBV1.1 den umgesetzten Varianten zugeordnet. Die umgesetzten mBV zeichnen sich ähnlich zum Fall der integralen Bausteinvarianten durch zusätzliche wichtige Informationen aus der Entwicklung in Zusammenhang mit dem Produktentstehungsprozess von P1 aus. Für bereits umgesetzte mBV lassen sich beispielsweise Restriktionen über relative Platzierungen der Bausteine zueinander sowie potenzielle oder verpflichtende Nachbarbausteine angeben, die im Laufe der Entwicklung aufgetreten sind. Entsprechend der PGE können somit Entwicklungsrisiken vermindert werden. Diese und weitere Informationen werden ebenfalls in einem Bausteinsteckbrief aggregiert. Betrachten wir nun die Kombination aus BV1.1a und BV1.2b. Hier gibt es noch kein entsprechendes Produkt, in dem beide Bausteine zusammen verwendet wurden. Die entstandene mBV1.2 ist daher zunächst nur als potenziell möglich einzustufen. Um die Möglichkeit der Kombination zu evaluieren ist im Sinne der frühen Validierung zunächst eine theoretische Kompatibilitätsprüfung vorgesehen. Dazu werden die Ein- und Ausgangsgrößen beteiligter Bausteine überprüft und gemäß der im FSA dargestellten Verbindungen miteinander verglichen. Je nach Art des zu betrachtenden Parameters kann dies beispielsweise auf Basis der Verrundung von Intervallen oder der Bildung logischer Schnittmengen erfolgen. Eine detaillierte Auseinandersetzung mit der Kompatibilitätsprüfung findet im Rahmen der Baukastenanwendung in Kapitel 5 statt. Die Parametrierung der möglichen mBV findet ebenfalls in Form eines Bausteinsteckbriefes statt. Führt man die theoretische Kompatibilitätsprüfung für umgesetzte mBV ebenfalls durch, so können berechnete und reale Werte gegenübergestellt werden. Auf diese Weise können die zur Kompatibilitätsprüfung herangezogenen Methoden kontinuierlich verbessert. Gleichzeitig kann eine Aussage über den tatsächlichen Einfluss des Produktdesigns auf das Bausteinverhalten getroffen werden.

Der Vorgang der Dekomposition kann sich beliebig oft wiederholen, so lange, bis die in Kapitel 4.2.1 abgeleiteten Kriterien eine weitere Strukturierungsebene verbieten. Auf der untersten Ebene zulässigen Strukturierungsebene existieren folglich nur integrale Bausteinvarianten für die dort dargestellten Funktionsumfänge. Beim Dekompositionsprozess kann es vorkommen, dass übergeordnete Funktionscluster in verschiedenen Produkten jeweils anders heruntergebrochen werden. Dies kann sich zum einen über eine unterschiedliche Aufteilung der Funktionsumfänge in untergeordnete Funktionscluster oder eine differente Flussverkettung zwischen den Funktionsclustern äußern (Abbildung 4.17). In beiden Fällen wird für jede der Umsetzungen ein eigener Funktionsstrukturausschnitt eingefügt (siehe FSA 1 und FSA 2 in Abbildung 4.17). Auf diese Weise können parallele Pfade in der Struktur entstehen, die sich spätestens auf Ebene des obersten Funktionsclusters wieder vergleichen lassen. Die parallelen Pfade sind klare Indizien für Variantentreiber innerhalb des analysierten Produktportfolios. Aufgrund der hohen Flexibilität bei der Bildung

von Funktionsclustern ist die Erweiterung eines auf diese Weise gebildeten Baukastens jederzeit möglich.

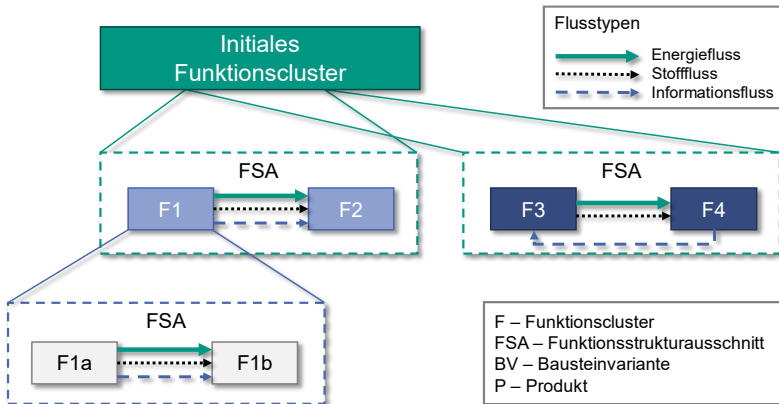


Abbildung 4.17: Parallele Dekompositionspfade bei der Baukastenstrukturierung. Das initiale Funktionscluster lässt sich dementsprechend auf verschiedene Weise in Funktionscluster (F1+F2 bzw. F3+F4) dekomponieren.

### Zusammenfassung und Einordnung in den Makroprozess nach Bursac (2016):

Betrachten wir die einzelnen Elemente der über die Methode definierten Struktur abschließend auf Basis des nach Bursac (2016) vorgegebenen Makroprozesses. Zunächst werden die verfügbaren Produktmodelle gemäß ihrem funktionalen und technischen Anteil aufgetrennt. Die Funktionsstrukturen werden anschließend zu einer Referenz-Funktionsstruktur zusammengesetzt. Dieser Prozess gilt als Vorarbeit der eigentlichen Methodenanwendung. Die Referenz-Funktionsstruktur wird zusammen mit dem technischen Anteil der Produktmodelle dazu verwendet Funktionscluster über Funktionsstrukturausschnitte herunterzubrechen, die sich an der technischen Umsetzung orientieren. Die gemeinsame Betrachtung der Referenz-Funktionsstruktur, genauer deren User-Funktionen und der ersten Ebene an Funktionsstrukturausschnitten kann in diesem Zusammenhang als Produktreferenzmodell verstanden werden. Betrachtet man die mittels Funktionscluster und FSA's aufgespannte Struktur über alle Ebenen als Andockpunkt der späteren Bausteinvarianten, kann dieses Konstrukt als

Baukastenreferenzmodell verstanden werden. Die anschließende Einbindung der realen Bausteinvarianten inklusive deren Steckbriefe entspricht der Konkretisierung des Baukastenreferenzmodells zu einem Baukastenmodell. Innerhalb des Baukastenmodells stellen die Funktionscluster die austauschbaren Module des Baukastens dar.<sup>38</sup> Die abgeleitete Methode zur Baukastenstrukturierung ist damit eine Konkretisierung des Makroprozesses hinsichtlich der Aktivitäten von Induktion und Deduktion für die Anwendung im Bereich der mechatronischen Steuergeräte. An dieser Stelle sei die besondere Eignung des iPeM zur Modellierung komplexer, iterativer Prozesse betont.

### 4.3 Implementierung der Strukturierungsmethode mittels MBSE

Nachdem die Frage nach der Gestaltung einer MBSE-gestützten Methode zur Baukastenstrukturierung im vorangegangenen Unterkapitel beantwortet wurde, wird das entwickelte Framework nun anhand eines realen Portfolioausschnittes mechatronischer Steuergeräte exemplarisch angewendet. Die Implementierung findet auf Basis der Rahmenbedingungen innerhalb der Untersuchungsumgebung statt. Mit der exemplarischen Anwendung der Strukturierungsmethode wird die zweite Teilforschungsfrage beantwortet:

*FF2.2: Wie werden die Baukastenstrukturierung und –anwendung mechatronischer Steuergeräte mittels MBSE-Methoden entsprechend integriert?*

Die Implementierung wird als mehrstufiger Prozess vorgestellt, der sich in folgende fünf Schritte unterteilen lässt (Abbildung 4.18):

- Analyse bestehender Produkte
- Anwendung der Strukturierungsmethode auf die ausgewählten Produkte<sup>39</sup>
- Erstellung eines Eigenschafts-Abhängigkeitsdiagramms
- Ermittlung der Bausteindaten<sup>40</sup>
- Erstellung Baukastenmodell in IBM Rational Rhapsody

---

<sup>38</sup> vgl. Modulbegriff Kapitel 2.4.2

<sup>39</sup> Ergebnis dieses Schrittes stellt die Identifikation der Modulstrukturen dar

<sup>40</sup> Hierbei werden lediglich die Daten der integralen Bausteinvarianten berücksichtigt

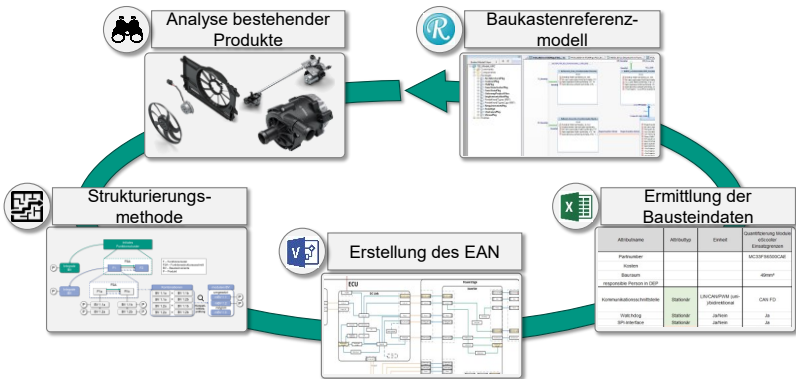


Abbildung 4.18: Vorgehensweise zur Implementierung der Strukturierungsmethode für Baukästen mechatronischer Steuergeräte bestehend aus den fünf angeführten Schritten.

Die einzelnen Schritte werden im Rahmen der folgenden Unterkapitel sequenziell durchlaufen.

### 4.3.1 Analyse bestehender Produkte

Der Schritt der Analyse bestehender Produkte dient zwei Zielen. Erstens der Festlegung des über den Baukästen abzubildenden Portfolios. Zweitens der Ableitung geeigneter Produktmodelle und der Zusammenführung der einzelnen Funktionsstrukturen zu einem Referenz-Funktionsmodell. Beide Ziele lassen sich entsprechend des Modells der PGE als Festlegung der Elemente des Referenzsystems für den Baukastens selbst, als auch die abzuleitenden Produkte beschreiben. In Bezug auf die Festlegung des Zielfortfolios spielt die Frage nach der geeigneten Baukastenarchitektur eine zentrale Rolle. Diese hängt stark von den unternehmensseitigen Gegebenheiten ab.<sup>41</sup> Im Rahmen der vorliegenden Untersuchungsumgebung wird, wie in Kapitel 3.3.1 beschrieben, die Strategie einer hybriden Baukastenarchitektur auf Basis von vier Technologiesets (TS) bereits verfolgt. Der in Zusammenhang mit den Technologiesets erstellte Anforderungs- und Funktionskatalog stellt für die vorliegende Arbeit eine wichtige

<sup>41</sup> Vgl Kapitel 2.4.2

unternehmensseitige Randbedingung dar. Genauer werden die für die TS spezifizierten Versionen dieses Katalogs im Folgenden als Referenz-Funktionsmodelle und damit als Ausgangspunkt der Baukastenstrukturierung angewendet. Der Katalog liegt im Anforderungsmanagementtool DOORS classic von IBM vor. Die notwendigen Indikationen über die technischen Umsetzungen werden beispielsweise in Form von Hardware-Blockschaltbildern, Software-Architekturdarstellungen und CAD Modellen zur Betrachtung herangezogen.

Eine weitere wichtige Randbedingung ergibt sich aus der Tatsache, dass der über die Strukturierung erzeugte Baukasten für die **Frühe Phase der PGE mechatronischer Steuergeräte** eingesetzt werden soll. Hierbei liegt der Fokus auf der schnellen Ableitung von Funktionsmustern. Da diese in der Regel ohne Gehäuse und mechanische Schnittstellen nach außen wie z.B. Stecker aufgebaut werden, können diese Elemente bei der Strukturierung entfallen. Soll die Methode ebenfalls auf die mechanische Einfassung der Elektronik im Gehäuse ausgeweitet werden, muss sie in diesem neuen Zusammenhang geprüft und gegebenenfalls erweitert werden. Für das folgende Beispiel wurde ein exemplarisches Portfolio auf Basis von fünf Steuergeräten des Technologiesatzes 1 ausgewählt. Die Steuergeräte des TS 1 sind ausnahmslos im Rahmen von Kleinantrieben im Automobilbereich im Einsatz. Sie zeichnen sich durch vergleichsweise geringe Ausgangsleistungen, geringe Anforderungen an die Regelungsalgorithmen und die Prozessorstärke, sowie eine niedrige Sicherheitsklasse in Zusammenhang mit der ISO26262 aus (Abbildung 4.19). Hieraus ergibt sich die Beschränkung, dass die Methode ebenfalls für den Einsatz leistungsstärkerer Steuergeräte separat geprüft werden muss.

**ANWENDUNG AN 12V ANTRIEBEN (AUTOMOTIVE)**

- ▶ Geringe Leistungen: 50-800W
- ▶ Geringe Anforderungen an die Regelung/  
Prozessorstärke ~0 .. 40 DMlps
- ▶ Niedrige Sicherheitsklasse



Abbildung 4.19: Eigenschaften der Antriebe des gewählten Beispielportfolios. Diese sind unter anderem durch geringe Ausgangsleistungen und geringe Anforderungen an die Regelalgorithmen gekennzeichnet. Eigene Darstellung unter Verwendung von Produktillustrationen der Robert Bosch GmbH (Robert Bosch GmbH, 2022).



### 4.3.2 Anwendung der Strukturierungsmethode

Der folgende Schritt der Anwendung der Strukturierungsmethode dient in erster Linie dem Verständnis zur Nutzung der Strukturierungskriterien an einem realen Portfolio. Das Zusammenspiel der Funktionscluster innerhalb eines Funktionsstrukturausschnittes und damit die Grundlage der späteren Kompatibilitätsbetrachtung wird zur Vereinfachung für das vorliegende Unterkapitel zunächst ausblendet. Ergebnis dieses Unterkapitels ist demnach ein Leitfaden zur Anwendung der Strukturierungsmethode für das Beispielportfolio, der in einer exemplarischen Modulstruktur eines Baukastens resultiert. Für die Anwendung der Strukturierungsmethode am Beispielportfolio werden die TS 1 Referenz-Funktionsstruktur, sowie die einzelnen Blockschaltbilder der ausgewählten Produkte als Grundlage genutzt. Im ersten Schritt werden sämtliche User-Funktionen dem Funktionscluster der ECU zugeordnet, um einen Startpunkt der Methodenanwendung zu generieren. Für die ECU existiert im gewählten Portfolio erwartungsgemäß noch keine integrale Bausteinvariante. Dies stellt in Zusammenhang mit allen weiteren, im Prozess des Herunterbrechens identifizierten Funktionsclustern eine Ausnahme dar.

Im nächsten Schritt wird das initiale Funktionscluster entsprechend den Umsetzungen aus den Produkten mittels Einsatzes der Strukturierungskriterien in weitere Funktionscluster unterteilt. Hierfür wird der erste Funktionsstrukturausschnitt eingeführt. Dieser entspricht gemäß der in Kapitel 4.2.2 vorgestellten Überlegungen dem technischen Anteil des Produktreferenzmodells. Er bildet gleichzeitig die oberste Vergleichsebene aller technischen Umsetzungen der über den Baukasten abzubildenden Lösungen. Die Identifikation der Funktionscluster wird nun unter Anwendung der Strukturierungskriterien am Beispiel des Logikmoduls aufgezeigt. Betrachten wir zunächst zwei der schematischen Blockschaltbilder (Abbildung 4.20).<sup>42</sup>

---

<sup>42</sup> Hierbei handelt es sich um vereinfachte Darstellungen.

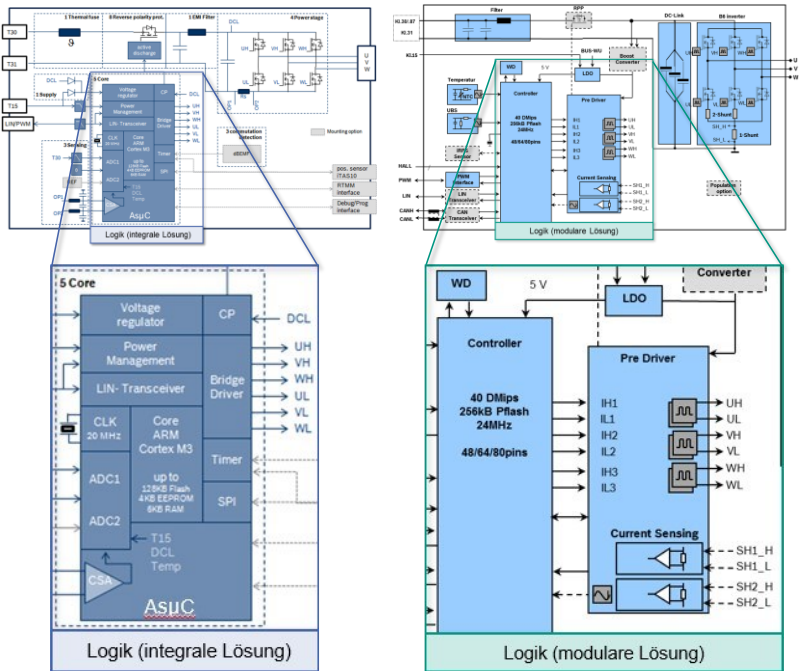


Abbildung 4.20: Übersicht der schematischen Blockschaltbilder zweier Produkte des gewählten Beispielfortfolios. Hervorgehoben sind die unterschiedlichen technischen Lösungen für das Logikmodul eines mechatronischen Steuergerätes.

Aus dem rechten Teil der Abbildung wird ersichtlich, dass die Gestaltung des Logikmoduls eindeutig einem diskreten Aufbau folgt. Das linke Blockschaltbild hingegen weist einen integralen Aufbau in Form eines applikationsspezifischen Mikrocontrollers (ASuIC) auf. Unter der Annahme, dass sich zunächst nur die beiden dargestellten Produkte im Portfolio befinden, wird das Varianzkriterium angelegt. Das Varianzkriterium lässt nur dann eine tiefere Strukturierungsebene zu, wenn sich die gleiche Untergliederung in mehreren Produkten befindet. An dieser Stelle kann ohne Prüfung weiterer Kriterien entschieden werden, dass keine tiefere Strukturierungsebene eingeführt wird. Ergebnis ist ein Funktionscluster, welches alle Funktionen der Logik in sich vereint und zwei entsprechende integrale Bausteinvarianten.

Hieraus wird ersichtlich, dass selbst diskret aufgebaute Strukturen im Falle eines Abbruchs der Strukturierungstiefe als integrale Bausteinvarianten betrachtet werden können. Die internen Beziehungen zwischen den diskreten Elementen spielen in diesem Fall keine Rolle auf der Ebene des Baukastens.

Im Portfolio befindet sich allerdings ein weiteres Produkt, welches einen vergleichbaren, diskreten Aufbau aufweist (Abbildung 4.21).

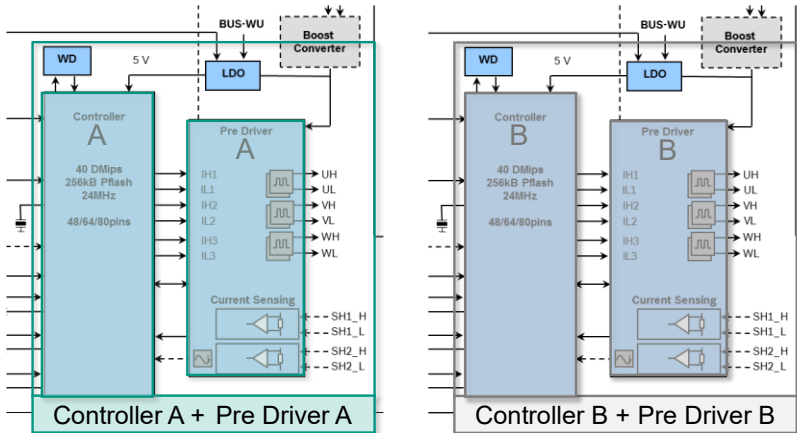


Abbildung 4.21: Gegenüberstellung der diskreten Lösungen zweier Produkte des gewählten Beispielportfolios für das Modul Mikrocontroller + Treiberbaustein.

Die beiden in Abbildung 4.21 dargestellten Produkte unterscheiden sich in der Realität durch die Wahl verschiedener Treiberbausteine und Mikrocontroller. Wenden wir nun erneut das Varianzkriterium an. Da sich beide Logikmodule sowohl in Funktion<sup>43</sup>, als auch in ihrem prinzipiellen Aufbau entsprechen gilt das Varianzkriterium als erfüllt (Abbildung 4.22).

<sup>43</sup> Aus der Betrachtung der Referenz-Funktionsstruktur ersichtlich

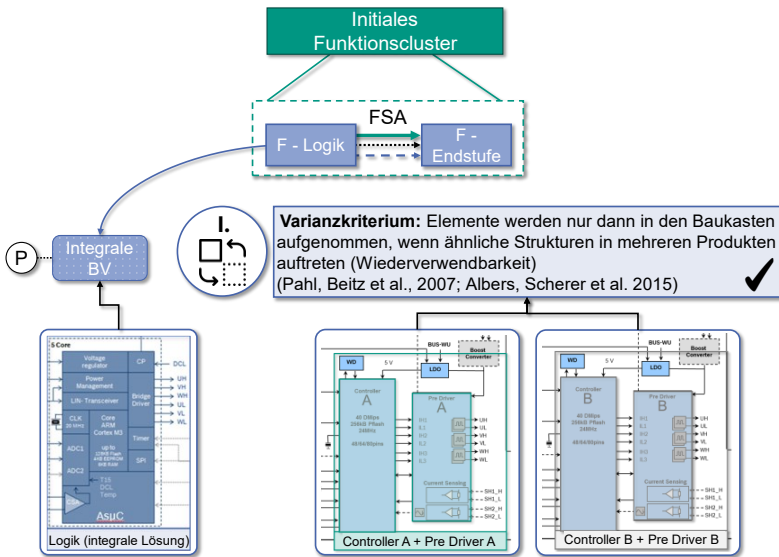


Abbildung 4.22: Prüfung des Varianzkriteriums am Beispiel des zuvor betrachteten Logikmoduls und dessen unterschiedlichen technischen Lösungen.

Die Strukturierungsmethode sieht demnach die anschließende Prüfung des Abhängigkeitskriteriums vor (Abbildung 4.23). Entsprechend der in Kapitel 4.2.1 gegebenen Definition muss hierbei geprüft werden, ob eine unabhängige Betrachtung der identifizierten Bausteine trotz deren Wechselwirkungen gelingen kann. Im vorliegenden Fall wurde diese Fragestellung im Expertenkreis des Chipsatz-Teams diskutiert. Die Diskussion kam zu dem Ergebnis, dass sich die Abhängigkeiten von Mikrocontroller und Treiberbaustein trotz getrennter Betrachtung beherrschen lassen. Das Abhängigkeitskriterium gilt somit als erfüllt.

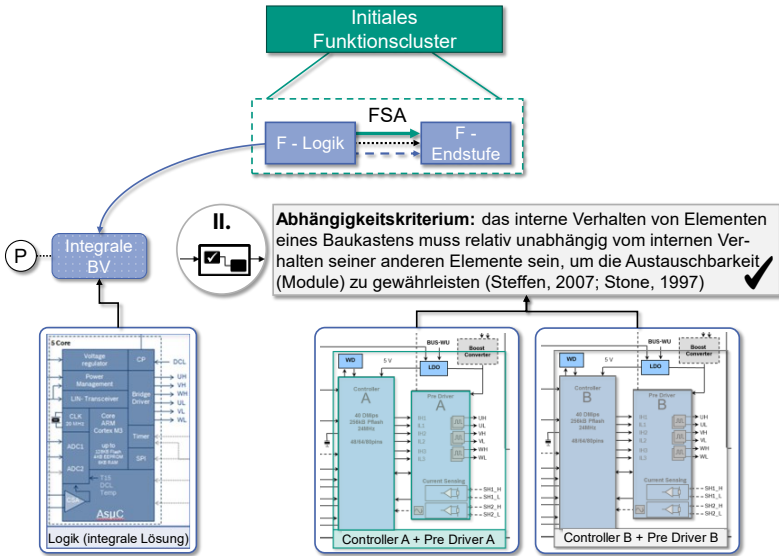


Abbildung 4.23: Prüfung des Abhängigkeitskriteriums am Beispiel des Logikmoduls und dessen unterschiedlichen technischen Lösungen.

Die anschließende Beantwortung des Verhältnismäßigkeitskriteriums wurde im Beisein von Systementwicklern geklärt (Abbildung 4.24). Hierbei herrschte ebenfalls Konsens über die Verhältnismäßigkeit der Einführung einer neuern Strukturierungsebene.



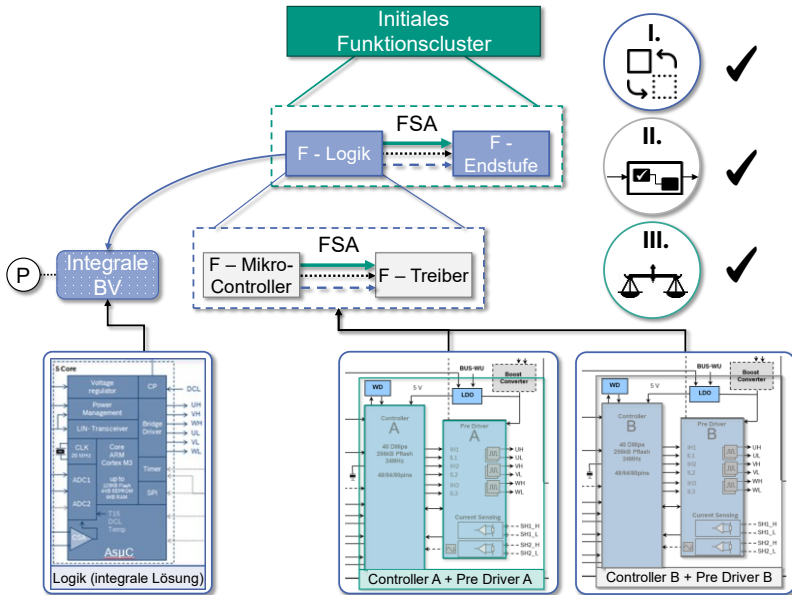


Abbildung 4.25: Einführung einer neuen Strukturierungsebene für das Logikmodul in Form eines neuen Funktionsstrukturausschnittes (FSA) und den Funktionsclustern „Mikrocontroller“ und „Treiber“.

Die vier Bausteinvarianten werden im Nachgang zunächst kombiniert und mittels Kompatibilitätsprüfung<sup>44</sup> in modulare Bausteinvarianten überführt (Abbildung 4.26). Die entstandenen modularen Bausteinvarianten sind auf diese Weise mit der integralen Bausteinvariante des applikationsspezifischen Mikrocontrollers vergleichbar. Dies gilt als wichtige Voraussetzung für die spätere Baukasten-anwendung. Analog zur Betrachtung des Logikmoduls wurde die Strukturierung mit allen weiteren Elementen der Produktstrukturen durchgeführt.

<sup>44</sup> Vgl. Kapitel 5.2.1

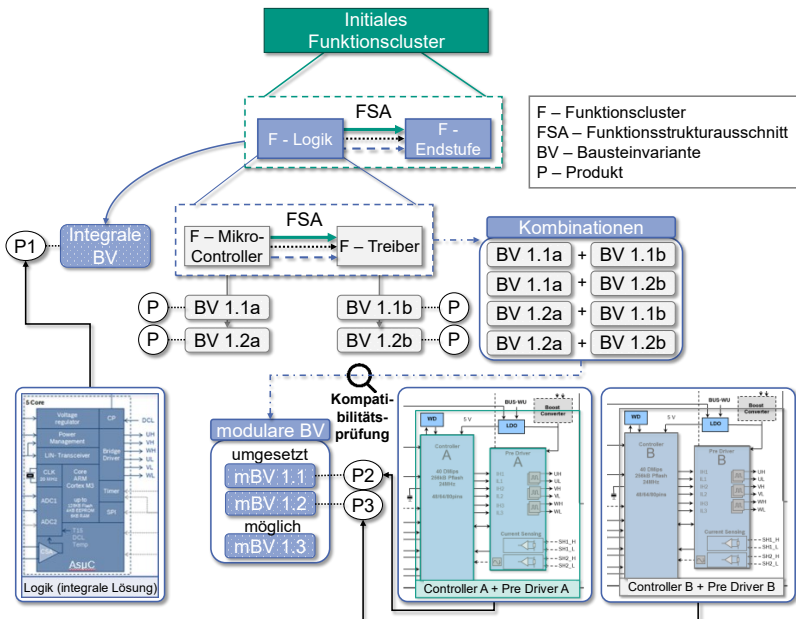


Abbildung 4.26: Bildung der modularen Bausteinvarianten für das Logikmodul durch Kombination der integralen Bausteinvarianten der Funktionscluster „Mikrocontroller“ und „Treiber“.

Dieselbe Vorgehensweise wird bei der Betrachtung des Softwareanteils eines Steuergerätes angewendet. Als Indikation der Produktstrukturen werden hierbei geeignete Darstellungen wie beispielsweise Softwarearchitekturen zu Rate gezogen. Wie in Kapitel 2.1.2 beschrieben, existieren Softwareanteile, die eine besonders starke Abhängigkeit zur verwendeten Hardware aufweisen. Dies gilt vor allem für die Basissoftware, welche eine standardisierte Schnittstelle zur Hardware des Mikrocontrollers darstellt. Jedoch auch in diesem Fall ist eine Untergliederung der Funktionscluster sinnvoll, da die starken Zusammenhänge bei der Kompatibilitätsbetrachtung ausreichend betrachtet werden können. Die entsprechenden Softwareelemente werden wiederum als eigenständige Bausteinvarianten im Baukasten aufgenommen.

Auf Basis der betrachteten, vereinfachten Blockschaltbilder lässt sich weiterhin der Umgang mit sogenannten optionalen Modulen illustrieren. Wie in Kapitel 4.2.2



erwähnt, beinhaltet das initiale Funktionscluster die Hüllkurve aller für das Portfolio möglichen Funktionalitäten mechatronischer Steuergeräte. Da diese nicht in allen Produkten benötigt werden, werden sie als optionale Elemente des ersten FSA gestaltet. Beispielsweise sind thermische Sicherungen, die im Falle einer Überhitzung den Stromkreis durchbrechen sollen, nicht von allen Kunden explizit gewünscht und daher bei der Betrachtung ausgespart werden. Als Ergebnis der Anwendung der Strukturierungsmethode lässt sich eine Übersicht der identifizierten Funktionscluster unter Beachtung der jeweiligen Hierarchieebene erstellen. Über die Anwendung der Strukturierungskriterien ist sichergestellt, dass immer mindestens zwei Bausteinvarianten für ein Funktionscluster existieren. Deren Substitution führt dazu, dass die Summe der Funktionalität bzw. der Attribute des Systems verändert werden. Die entstandene Übersicht kann somit als Modulstruktur bezeichnet werden.<sup>45</sup> Abbildung 4.27 zeigt eine beispielhafte Modulstruktur, in der die tieferen Strukturierungsebenen innerhalb der identifizierten Module angedeutet sind.

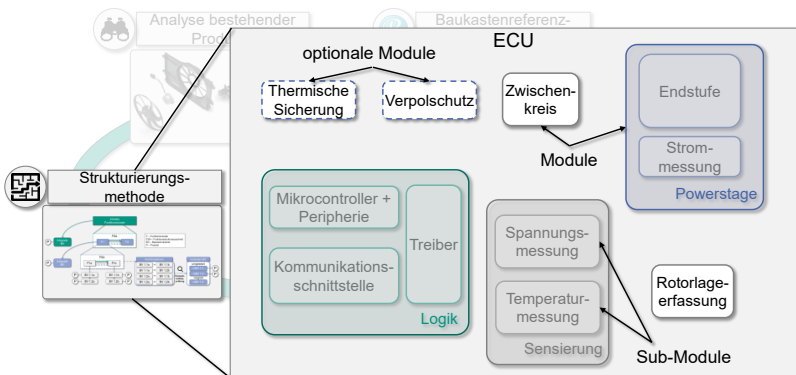


Abbildung 4.27: Beispielhafte Modulstruktur eines mechatronischen Steuergerätes als Ergebnis der Strukturierungsmethode. Gezeigt sind resultierende Module, Sub-Module und optionale Module.

<sup>45</sup> vgl. Kapitel 2.4.1 Definition des Modulbegriffs

### 4.3.3 Erstellung des Eigenschafts-Abhängigkeits-Netzwerks (EAN)

Um die vollständige Modellierung der Funktionsstrukturausschnitte zu ermöglichen und gleichzeitig die Grundlage der Kompatibilitätsprüfung zu schaffen, werden im nächsten Schritt die Beziehungen zwischen den Modulen des Baukastens untersucht. Diese Beziehungen entscheiden über die Güte der Produktkombinationen, die im Zuge der späteren Baukasten-anwendung empfohlen werden. An dieser Stelle ist es daher von höchster Bedeutung, möglichst viele Erfahrungsträger in die Methodenentwicklung einzubeziehen. Ausgehend von der ermittelten Modulstruktur wurden für alle identifizierten Module und Sub-Module die jeweiligen Experten hinzugezogen. Diese wurden im Rahmen einer Befragung dazu gebeten alle diejenigen Parameter zu benennen, welche sie für die initiale Auswahl der technischen Realisierung ihres Moduls als unmittelbar relevant erachten.

Damit die spätere Auswahl an Bausteinvarianten zur Realisierung vorgegebener Produkthanforderungen möglich ist, müssen neben den Parametern für die Module zusätzlich die relevanten Parameter für das Gesamtsystem ermittelt werden. Diese wurden mit den zuständigen Systementwicklern bestimmt. Die im Rahmen der vorliegenden Arbeit entwickelten Methoden kommen in der Frühen Phase der PGE zur Ableitung von Funktionsmustern zum Einsatz. Da die Funktionsmuster für die frühzeitige Validierung von Produktkonzepten dienen drängte sich die These auf, dass zur initialen Auswahl der Bausteine maximal 10-20 Parameter notwendig seien. Diese These konnte in Abstimmung mit den Experten verifiziert werden. Die auswahlentscheidenden Parameter der Module und des Gesamtsystems werden im Folgenden auch **Key-Parameter** genannt. Bei der Benennung der Key-Parameter wurden den Experten keine weitere Einschränkung mitgegeben. Im Vergleich zu klassischen Vorgehensweisen<sup>46</sup> beschränken sich die ermittelten Parameter dabei nicht ausschließlich auf Energie-, Stoff oder Signalflüsse, sondern umfassen weitere Eigenschaften wie beispielsweise Genauigkeiten von Sensormodulen. Diese nicht in Form von Flüssen abzubildenden Eigenschaften werden im Weiteren als **stationäre Attribute** bezeichnet. Bei den ermittelten Key-Parametern handelt es sich weiterhin ausschließlich um Schnittstelleninformationen, also Größen die sich auf das Modul bzw. das Gesamtsystem als Black-Box beziehen. Über die benannten Key-Parameter werden die in Kapitel 4.1 ermittelten Elemente des Zielsystems: die Forderung nach klar definierten Schnittstellen und einer geeigneten

---

<sup>46</sup> Diese weisen häufig eine strikte Orientierung an Energie-, Stoff- und Signalfüssen auf (Stone, 1997).

Unterstützung in der Frühen Phase der PGE, direkt adressiert. Abbildung 4.28 zeigt eine vereinfachte Darstellung des ermittelten Moduls „Zwischenkreis(-kondensator)“ inklusive seiner Key-Parameter.

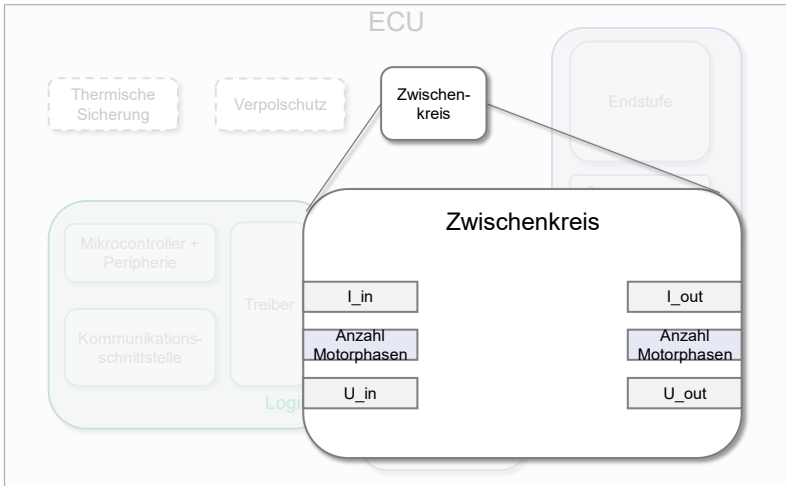


Abbildung 4.28: Ausschnitt des Moduls Zwischenkreis inkl. Key-Parameter für Eingangsstrom, Ausgangsstrom und Motorphasenanzahl.

Nach diesem Vorbild lassen sich die Ergebnisse der Expertengespräche anschließend in eine, um die Key-Parameter ergänzte Modulstruktur überführen.

Auf Basis der ergänzten Modulstruktur werden im nächsten Schritt die Zusammenhänge zwischen den einzelnen Modulen geschaffen. Hierzu werden weitere Informationen wie beispielsweise die Verknüpfung der Module im Schaltplan oder die Zuordnung der Softwaremodule zu den Chipsatzelementen herangezogen. Bezüglich der möglichen Beziehungen der Module gilt es folgendes zu prüfen:

- Besteht eine direkte physische Verbindung zwischen zwei Modulen bzw. sind diese im Schaltplan direkt verbunden?

Ist dies der Fall, so werden die Module über alle die Key-Parameter miteinander verbunden, die in beiden Modulen vorkommen (Abbildung 4.29).

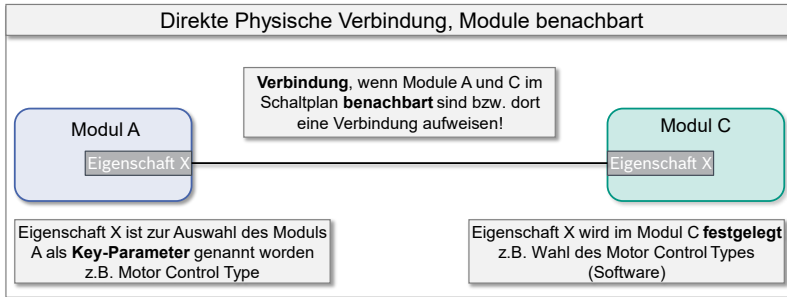


Abbildung 4.29: Erstellung eines Eigenschafts-Abhängigkeits-Netzwerks (EAN): Regeln für die Verbindung benachbarter Module.

- Besteht keine physische Verbindung zwischen den Modulen bzw. sind diese im Schaltplan nicht direkt verbunden?

Ist dies der Fall so muss geprüft werden, ob einer der Key-Parameter, die beide Module aufweisen über die Auswahl eines technischen Elementes für eines der Module festgelegt wird. Abbildung 4.30 zeigt dazu folgendes Beispiel. Modul A und Modul C weisen keine physische Verbindung auf noch sind diese im Schaltplan miteinander verbunden. In Modul C wird ein Key-Parameter festgelegt, welcher im Modul A zur Auswahl benötigt wird. Beispielsweise wird im Modul der Motor-Control der Regelungstyp und die Form der Kommutierung bestimmt. Diese Größen sind wiederum für die Auswahl der Leistungsendstufe relevant. Da beide Module allerdings keine direkte Schnittstelle aufweisen werden die Key-Parameter für alle dazwischen liegenden Elemente (im Beispiel Modul B) relevant. Für die Verbindungen zwischen den Modulen bedeutet dies zwei Dinge: Erstens nur direkt benachbarte Module können eine Verbindung besitzen. Zweitens Module können neben den von den Experten genannten Key-Parametern auch weitere Key-Parameter bekommen, wenn sie diese als Übermittler weitergeben müssen. Im Falle der Übermittlerfunktion besitzt der betreffende Key-Parameter zwei Verbindungen.

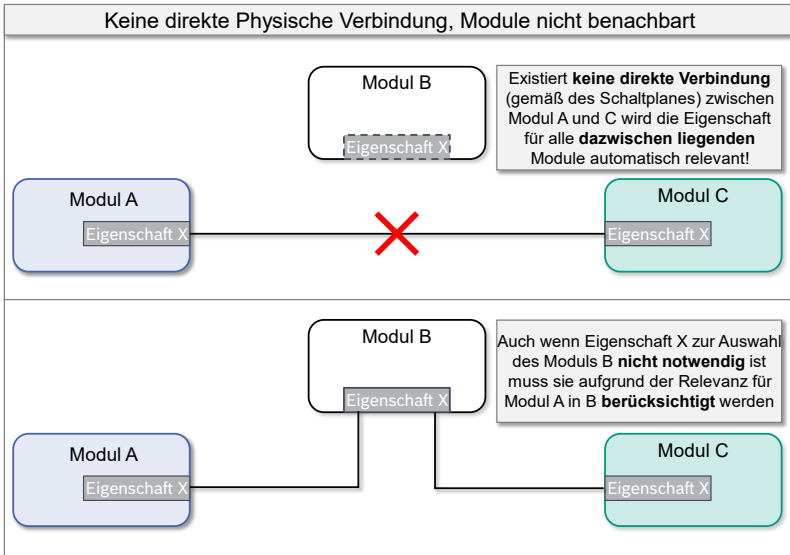


Abbildung 4.30: Erstellung eines Eigenschafts-Abhängigkeits-Netzwerks (EAN): Regeln für die Vorgehensweise zur Verbindung nicht benachbarter Module.

Da das Gesamtsystem ebenfalls mit Key-Parametern versehen wird, gibt es zwangsweise Verbindungen von Modulen und dem Gesamtsystem. Jede Verbindung zwischen zwei Modulen wird als **interne Beziehung** bezeichnet. Diese symbolisiert zum einen eine Wechselwirkung zwischen den Funktionsclustern eines Funktionsstrukturausschnittes und ist zum anderen Ansatzpunkt für die spätere Kompatibilitätsprüfung der Bausteinvarianten. Beziehungen zwischen den Key-Parametern des Gesamtsystems (der ECU) und einzelnen Modulen werden als **externe Beziehungen** bezeichnet. Werden auf diese Weise alle Module mit ihren Key-Parametern betrachtet kann als Ergebnis ein Netzwerk aller Module und Sub-Module inklusive deren Verbindungen erstellt werden. Da es sich wie eingangs erwähnt bei den Key-Parametern nicht nur um Flussgrößen, sondern allgemeine Eigenschaften und deren Abhängigkeiten handelt, wird die aggregierte Darstellung der Module mit ihren Beziehungen als **Eigenschafts-Abhängigkeits-Netzwerk (EAN)** bezeichnet. Abbildung 4.31 zeigt das resultierende EAN des TS1 Beispielfolios.

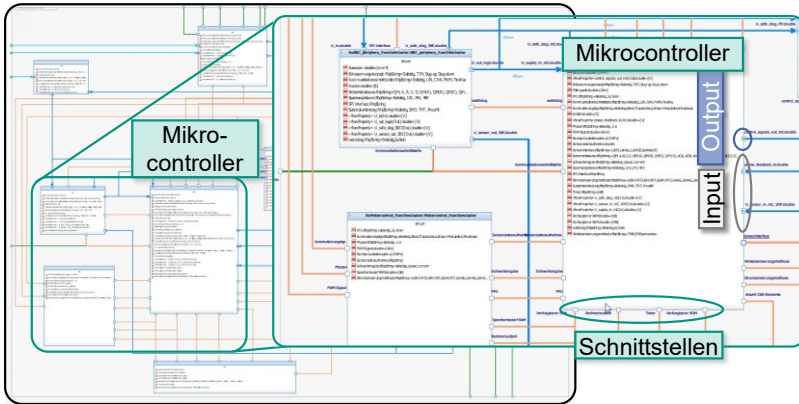


Abbildung 4.31: Übersicht des resultierenden Eigenschafts-Abhängigkeits-Netzwerks eines mechatronischen Steuergerätes am Beispiel des Mikrocontroller Moduls und dessen Schnittstellen.


Der EAN lässt sich als grafische Repräsentation technischer Umsetzungen zur Erfüllung eines definierten Funktionsumfangs betrachten. Die Darstellung entspricht dabei einem oder mehreren bestimmten Betriebszuständen. Der EAN stellt damit eine abstrakte Form eines im C&C<sup>2</sup>-A<sup>47</sup> beschriebenen Wirk-Netzes dar. Gleichzeitig entspricht er als Baukastenreferenzmodell einer Blaupause aller möglichen aus dem Baukasten abzuleitenden Elemente. Hinsichtlich der späteren Anwendung des Baukastens illustriert er die zur Kompatibilitätsprüfung benötigten Pfade.

### 4.3.4 Ermittlung der Bausteindaten

Im vorliegenden Schritt werden die mittels EAN vorgegebenen Key-Parameter der Module quantifiziert. Die Module sind dabei entsprechend des Baukastenreferenzmodells lediglich Platzhalter für ihre technischen Realisierungen. Die Quantifizierung ihrer Key-Parameter kann folglich nur für die einzelnen Bausteinvarianten erfolgen. Zur Quantifizierung der Bausteinvarianten werden, die in Kapitel 4.2.2 erwähnten Bausteinsteckbriefe erstellt. Der Aufbau eines solchen

<sup>47</sup> vgl. Kapitel 2.2.5

Steckbriefs ist in Abbildung 4.32 exemplarisch in Form einer Excel-Tabelle dargestellt.

 **Ermittlung der Bausteindaten**

Attributname	Attributtyp	Einheit	Quantifizierung Module einschließl. Einsatzgrenzen
Partnummer			MC33FS6500CAE
Kosten			
Bauraum			49mm²
responsible Person in DEP			
Kommunikationsschnittstelle	Stationär	LIN/CAN/PWM (uni- )bidirektional	CAN FD
Watchdog	Stationär	Ja/Nein	Ja
SPI-Interface	Stationär	Ja/Nein	Ja
U_safe_diag_SBC	Output (Fluss)	V	-0.3 .. 40
U_supply_in_SBC	Input (Fluss)	V	-1.0 .. 40
U_supply_out_SBC	Output (Fluss)	V	-2.0 .. 40

Abbildung 4.32: Exemplarischer Aufbau eines Bausteinsteckbriefs. Enthalten sind Attributname, Attributtyp, Einheit des Attributs und die Quantifizierung entsprechend des Einsatzes in realen Produkten.

Neben dem Namen des Key-Parameters wird ebenfalls seine Einordnung in die Attributkategorien Flussattribut oder stationäres Attribut<sup>48</sup> erfasst. Bei den stationären Attributen wird zwischen quantifizierbaren und nicht-quantifizierbaren Attributen unterschieden. Abhängig von der Attributkategorie werden entweder die Einheiten der physikalischen Größen (für Flussattribute oder quantifizierbare

<sup>48</sup> vgl. Kapitel 4.3.3

stationäre Attribute<sup>49</sup>) oder eine Liste möglicher Attributwerte (für die nicht-quantifizierbaren stationären Attribute) festgelegt. Beispielsweise wird dem nicht-quantifizierbaren stationären Attribut Kommunikationsschnittstelle eine Liste der im Beispielportfolio denkbaren Kommunikationsarten LIN, CAN oder PWM hinterlegt. Für optionale Module, die auf Ebene des Gesamtsystems gezielt an oder abgewählt werden sollen, sind im EAN zusätzliche Parameter eingefügt. Diese Parameter tragen einen nach dem Schema „Modul XY benötigt“ aufgebauten Attributnamen. Die entsprechenden möglichen Attributwerte sind demnach „Ja“ oder „Nein“. Im nächsten Schritt werden für die integralen Bausteinvarianten, unabhängig von ihrem Einsatz in realen Produkten, zunächst die minimalen und maximalen Grenzwerte ermittelt. Diese finden sich beispielsweise in Datenblättern oder bereits vorhandenen Modulbeschreibungen. Zusätzlich zu den theoretischen Grenzwerten werden, entsprechend den Produkten, in denen die Bausteinvariante eingesetzt wird, die dort messbaren realen Betriebsintervalle erfasst. Für Flussattribute und quantifizierbare stationäre Attribute bietet sich beispielsweise eine Erfassung in Form von Intervallen, Wertetabellen oder mathematischen Funktionen an (vgl. Kapitel 4.2.2). Der große Vorteil bei der Berücksichtigung realer Betriebsintervalle gegenüber den theoretischen Grenzwerten liegt darin, dass die realen Betriebsintervalle bereits detaillierte Informationen des Bausteinverhaltens in einer Referenzumgebung enthalten. Die Informationen über den Einsatz in den Referenzumgebungen beispielsweise z.B. die Platzierung des Bausteins oder andere im Rahmen der Entwicklung gesammelte Erkenntnisse werden den Bausteinsteckbriefen ebenfalls mitgegeben. Im späteren Verlauf werden sowohl für die theoretischen Grenzwerte als auch die spezifischen Einsatzgrenzen der Produkte eigene Bausteinvarianten im Baukastenmodell erstellt. Die Informationen aus den Referenzsystemen können im Zuge der Kompatibilitätsprüfung (vgl. Kapitel 5.2.1) dazu verwendet werden, die auf Basis der theoretischen Grenzwerte durchgeführten Berechnungen zu plausibilisieren, bzw. zu deren Verbesserung beitragen.

Im Folgenden werden die, bis zu diesem Punkt erzielten Ergebnisse anhand des Beispielportfolios rekapituliert. Im Zuge der Anwendung der Strukturierungskriterien wurde unter Berücksichtigung des Funktions-Gestalt-Zusammenhangs die Modulstruktur eines Baukastens zur Abbildung der ausgewählten Produkte des TS1 geschaffen. Diese wurde zunächst mittels Key-Parametern entsprechend des späteren Einsatzes in der Funktionsmustererstellung ergänzt. Darauf aufbauend

---

<sup>49</sup> Als quantifizierbares stationäres Attribut gilt beispielsweise die Genauigkeit eines Sensors, die in Prozent angegeben wird





### 4.3.5 Erstellung des Baukastenmodells in IBM Rhapsody

Der letzte Schritt der Methodenimplementierung widmet sich der Erstellung des Baukastenmodells im MBSE Tool IBM Rhapsody. Hierzu werden die Ergebnisse der letzten Schritte konsolidiert in das, innerhalb der Untersuchungsumgebung vorgegebene Tool übertragen. Für die Erstellung des Baukastenmodells werden dazu die folgenden fünf Schritte sequenziell durchlaufen (Abbildung 4.34).

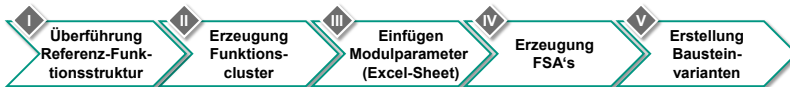


Abbildung 4.34: Fünf Schritte zur Erstellung des Baukastenmodells für mechatronische Steuergeräte im Einsatz in Kleinantrieben in IBM Rhapsody.

#### Überführung der Referenz-Funktionsstruktur in Rhapsody

Ausgangspunkt der Überführung ist der für die Produkte des TS1 angepasste Anforderungs- und Funktionskatalog in DOORS classic. Dieser ist entsprechend der in Kapitel 3.3.2 dargestellten Levelstruktur aufgeteilt und umfasst 6 Module. Innerhalb der einzelnen Module sind die Anforderungen den Funktionen direkt in Form einer Kapitelstruktur zugeordnet (Abbildung 4.35). Die Dekomposition der Hauptfunktionen in Teilfunktionen wird mittels modulübergreifenden Links realisiert.

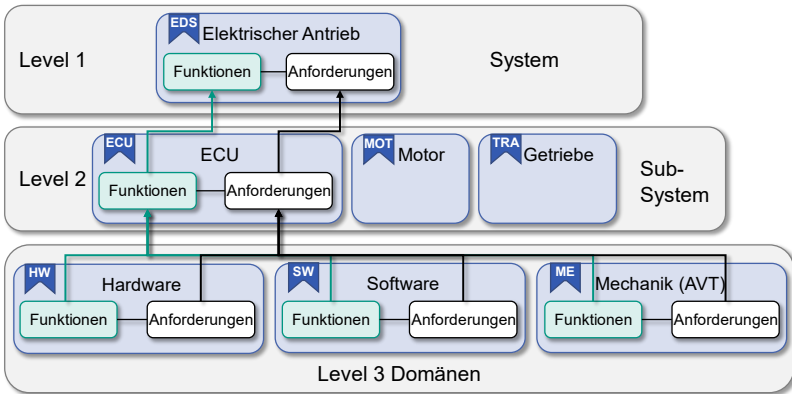


Abbildung 4.35: Schematische Darstellung der Struktur des DOORS Funktionskataloges. Der Aufbau der Struktur orientiert sich dabei an der ENS Systemstruktur (vgl. Abbildung 3.8, Kapitel 3.3.2)

Zur Modellierung des Baukastenmodells sollen sowohl die Anforderungen der einzelnen Funktionen als auch die Referenz-Funktionsstruktur nach Rhapsody überführt werden. Der Transfer der Daten von DOORS nach Rhapsody wurde mit der innerhalb der Installation von Rhapsody mitgelieferten Gateway-Anwendung, durchgeführt. Im Zielzustand soll die Abbildung der Funktionen in dem von der SysML dafür bereitgestellten Datentyp <<Block>> erfolgen. Die Verbindung der Anforderungen zu den Funktionsblöcken wird als <<satisfy>> Beziehung realisiert. Die hierarchischen Beziehungen zwischen den Funktionen der Referenz-Funktionsstruktur werden über Kompositionsbeziehungen (engl. directed composition) umgesetzt. Die Referenz-Funktionsstruktur kann anschließend in Form eines Block Definition Diagram (bdd) dargestellt werden (Abbildung 4.36).

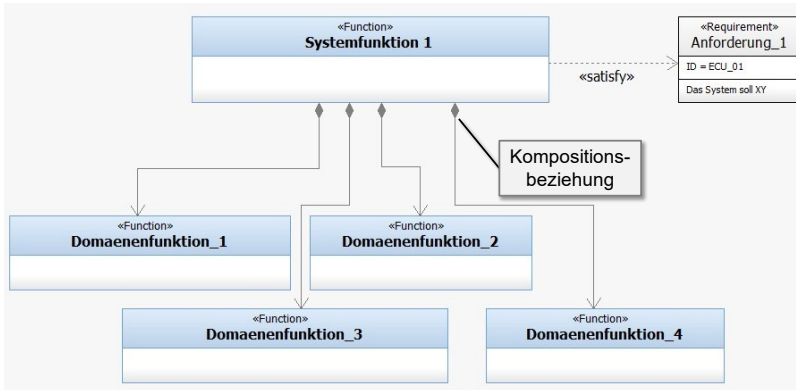


Abbildung 4.36: Schematischer Aufbau der Referenz-Funktionsstruktur in IBM Rhapsody. Eine übergeordnete Systemfunktion wird dabei mittels Kompositionsbeziehung in Domänenfunktionen aufgeteilt. Eine Anforderung ist über <<satisfy>>Beziehung angehängt.

Da sich der Umfang des Katalogs über mehrere 1000 Elemente erstreckt, ist eine manuelle Umstrukturierung der aus DOORS übertragenen Daten aufgrund des riesigen zeitlichen Aufwandes nicht wirtschaftlich. Durch die Nutzung einer Java-basierten Programmierschnittstelle, der sogenannten Java Rhapsody API konnte der manuelle Aufwand, mittels gezielten Einsatzes teilautomatisierter Modellierung, vollständig entfallen. Die auf diese Weise erzeugten helper-files erwiesen sich als wichtiger Faktor in der MBSE-gestützten Baukastenstrukturierung. Über die Baukastenstrukturierung hinaus konnten spezifische helper-files vermehrt dazu genutzt werden, mittels dokumentenzentrierter Arbeiten erzeugte Inhalte in zentrale Modelstrukturen nach dem Vorbild der MBSE zu überführen. An dieser Stelle sei noch einmal besonders auf die Arbeit von Powelske et al. (2018) hingewiesen. Auf Basis diverser Fallstudien konnte dort gezeigt werden, dass der gezielte Einsatz von helper-files signifikant zur Erhöhung der Akzeptanz von MBSE Methoden beitragen kann.

### Erzeugung der Funktionscluster/Mapping zur Referenzfunktionsstruktur

Widmen wir uns nun der Erzeugung der im Rahmen der Strukturierungsmethode identifizierten Funktionscluster. Für diese wurde in Rhapsody zunächst ein neuer Stereotyp eingeführt. Dieser ermöglicht es Modellelemente des SysML Datentyps <<Block>> in <<Funktionscluster>> zu überführen. Für die teilautomatisierte Zuordnung der Elemente der Referenz-Funktionsstruktur zu den Funktionsclustern wurde ein weiteres helper-file programmiert. Damit kann nach händischer Auswahl

der relevanten Elemente der Referenz-Funktionsstruktur im Modellbaum, per Rechtsklick ein entsprechendes Funktionscluster erstellt werden. Dieses wird gleichzeitig mittels Allokationsbeziehung an die ausgewählten Elemente verknüpft. Die Darstellung der Beziehungen erfolgt im ebenfalls durch Auslösung des helpers automatisch erstellten bdd. Um sich die spätere Erstellung der Funktionsstrukturausschnitte zu vereinfachen, wurden die hierarchischen Beziehungen zwischen den Funktionsclustern in Form von Kompositionen umgesetzt.

### **Einfügen der Modulparameter**

Nachdem die Funktionscluster erstellt sind, werden ihnen die ermittelten Key-Parameter zugewiesen. Die stationären Attribute werden hierzu als <<Value Properties>>, die als Flussattribute identifizierten Key-Parameter als <<Flow Properties>> innerhalb der Funktionscluster realisiert. Für die spätere Plausibilisierung der Attributwerte einzelner Bausteinvarianten sind an dieser Stelle bereits einige Vorkehrungen getroffen worden. Als Attributwerte der Funktionscluster werden im Falle von stationären Attributen beispielsweise zulässige Größen eingetragen. Für das Attribut der Kommunikationsschnittstelle sind dies im vorliegenden Portfolio LIN, CAN oder PWM (vgl. Kapitel 4.3.4). Für Flussattribute oder, über Zahlen zu quantifizierende, stationäre Attribute z.B. PWM Frequenz, werden die jeweiligen Einheiten als Attributwerte hinterlegt (Abbildung 4.37). Für die spätere Modellierung der Wirknetz-Darstellung innerhalb der FSAs werden für die Key-Parameter passende Schnittstellen an den Außengrenzen der Funktionscluster erstellt. Für die stationären Attribute werden <<Ports>> und für die Flussattribute <<Flow Ports>> verwendet. Sind alle Key-Parameter erzeugt können diese zur Erzeugung der vollständigen Traceability mit den entsprechenden Requirements verknüpft werden.

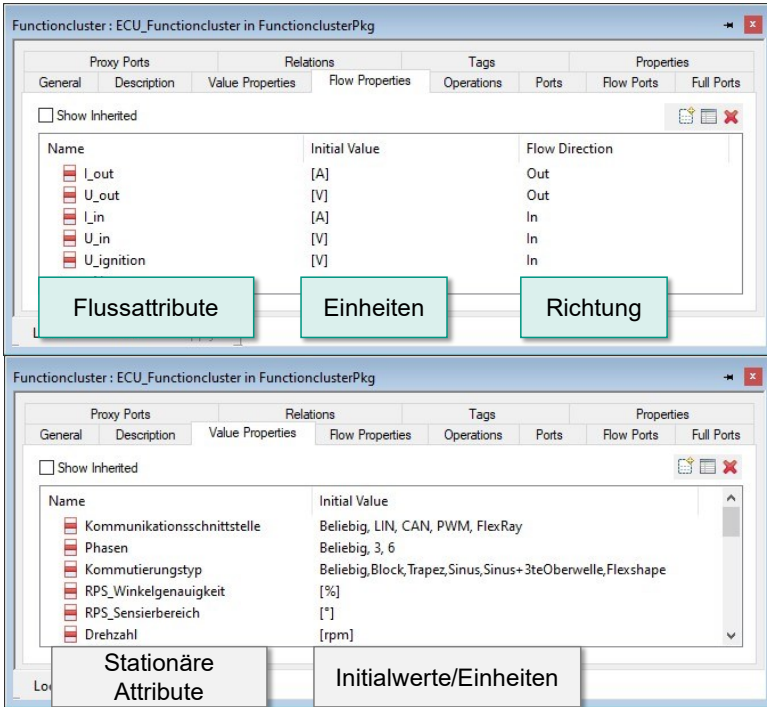


Abbildung 4.37: Übersicht der Flussattribute und stationären Attribute eines Funktionsclusters in IBM Rhapsody. Die Flussattribute weisen zusätzlich eine Flussrichtung auf.

### Erzeugen der Funktionsstrukturausschnitte

Im nächsten Schritt werden die Funktionsstrukturausschnitte, als zentrales Element zur Darstellung der Wirkzusammenhänge und damit unmittelbare Grundlage der späteren Kompatibilitätsprüfung, in Rhapsody modelliert. Analog zu den Funktionsclustern wird auch für die Funktionsstrukturausschnitte zunächst ein passender Stereotyp erstellt. Mittels helper-file kann der gewünschte FSA erstellt und automatisch mit den zur Ausgestaltung benötigten Funktionsclustern befüllt werden. Hierzu werden die Funktionscluster, deren Beziehungen im FSA visualisiert werden sollen, mittels Kompositionen als Teile des FSA realisiert. Die Darstellungen der Beziehungen zwischen FSA und den untergeordneten Funktionsclustern im Rahmen eines bdd

ist optional.<sup>50</sup> Wie der Strukturierungsmethode (vgl. Kapitel 4.2.2) zu entnehmen ist verstehen wir den FSA als Ausdetaillierung eines ihm übergeordneten Funktionsclusters.

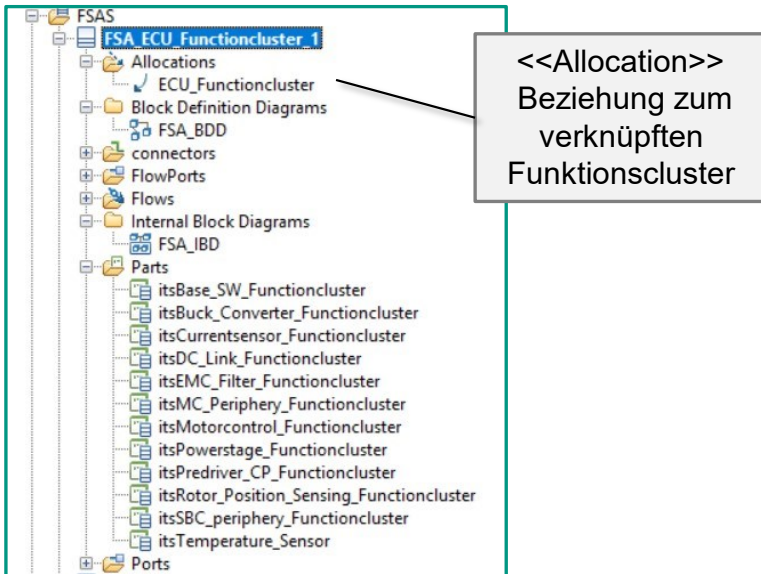


Abbildung 4.38: Umsetzung der Beziehungen zwischen FSA und Funktionscluster mittels <<Allocation>>Beziehung in IBM Rhapsody.

Um die Zugehörigkeit des FSA zu seinem übergeordneten Funktionscluster darzustellen, wird ausgehend vom FSA eine Allokationsbeziehung zum entsprechenden Funktionscluster erstellt (Abbildung 4.38). Neben den internen Beziehungen zwischen den Modulen sind die externen Beziehungen also die Verknüpfung von Modulen und FSA von großer Bedeutung. Damit diese

<sup>50</sup> Diagramme entsprechen in Zusammenhang mit einem ganzheitlichen Modellansatz lediglich verschiedenen Sichten auf das Gesamtmodell. Die einzelnen Modellelemente sind demnach nur dann innerhalb eines Diagramms zu erfassen, wenn dieses im Zuge des Entwicklungsprozesses einen definierten Nutzen aufweisen kann.

Verbindungen realisiert werden können, werden dem FSA die Schnittstellen seines übergeordneten Funktionsclusters als <<Ports>> und <<Flow Ports>> übergeben. Die Modellierung der Beziehungen selbst erfolgt in einem ibd. Als Vorlage der Modulverbindungen dient der bereits in Form von Visio-Diagrammen erstellten EAN des Beispielportfolios. Die Abhängigkeiten zweier Module auf Basis stationärer Attribute wird durch die Verbindung ihrer <<Ports>> durch <<Connectoren>> dargestellt. Beziehungen, die sich am Zusammenspiel von Flussattributen orientieren sind mittels <<Flows>> zwischen den <<Flow Ports>> der Module bzw. des FSA umgesetzt. Abbildung 4.39 zeigt den Funktionsstrukturausschnitt des Moduls „System Basis Chip“.

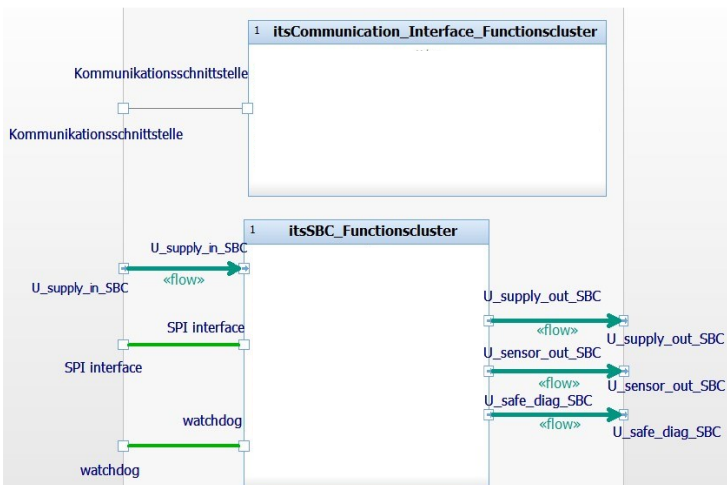


Abbildung 4.39: Übersicht eines Funktionsstrukturausschnittes für das Modul „System Basis Chip“. Enthalten sind stationäre Attribute (Verbindung über Linien dargestellt), sowie Flussattribute (Verbindung über Pfeile dargestellt).

Die Dekomposition eines Funktionsclusters wurde an dieser Stelle nicht direkt über die Nutzung eines ibd dargestellt, sondern stattdessen der FSA als eigener Datentyp eingeführt. Der Grund für diese Vorgehensweise liegt in der Tatsache, dass bei der Strukturierung des Baukastens durchaus parallele FSAs auftreten können. Diese können im Rahmen der späteren Kompatibilitätsberechnung auf diese Weise als unabhängige Elemente erfasst und evaluiert werden.

Sind alle Funktionsstrukturausschnitte nach dem oben genannten Vorbild modelliert liegt in Rhapsody bereits ein vollständiges Baukastenreferenzmodell vor. Dieses



wird durch die Erstellung der integralen Bausteinvarianten im nächsten Schritt zu einem initialen<sup>51</sup> Baukastenmodell erweitert.

### Erstellung der Bausteinvarianten

Der finale Schritt der Methodenimplementierung sieht die Modellierung der integralen Bausteinvarianten vor. Die Module des Baukastenreferenzmodells dienen hierbei als Blaupause der Bausteinvarianten und deren Verknüpfungen.

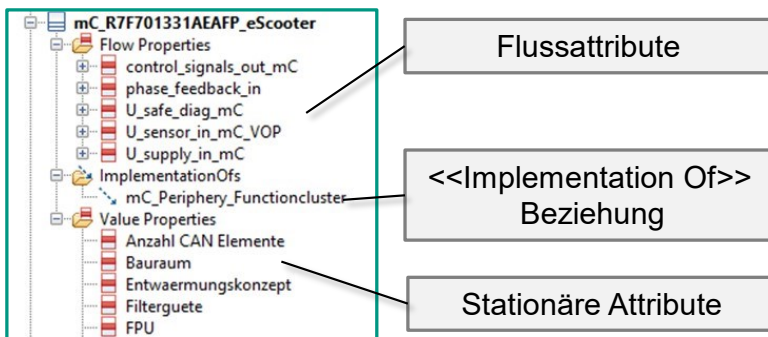


Abbildung 4.40: Übersicht einer erstellten Bausteinvariante inklusive Attribute und <<ImplementationOf>> Beziehung in IBM Rhapsody.

Die Bausteinvarianten werden dementsprechend nur mit den Attributen, jedoch nicht mit Schnittstellen in Form von <<Ports>> und <<Flow Ports>> versehen. Die Erstellung der Bausteinvarianten erfolgt erneut teilautomatisiert. Die Attribute des Funktionsclusters werden dabei der zu erstellenden Bausteinvariante übergeben. Gleichzeitig wird ein eigens dafür geschaffener Beziehungstyp <<ImplementationOf>> ausgehend von der BV an das Funktionscluster eingefügt (Abbildung 4.40). Im Anschluss an die Erstellung der integralen BV werden die

<sup>51</sup> Wir sprechen an dieser Stelle von einem initialen Baukastenmodell, da diesem zur Vollständigkeit die modularen Bausteinvarianten und die Produktkombinationen fehlen. Diese werden erst im Rahmen der Baukastenanwendung in Kapitel 5 ermittelt.

quantifizierten Attributwerte aus den gesammelten Bausteinsteckbriefen eingefügt (Abbildung 4.41).

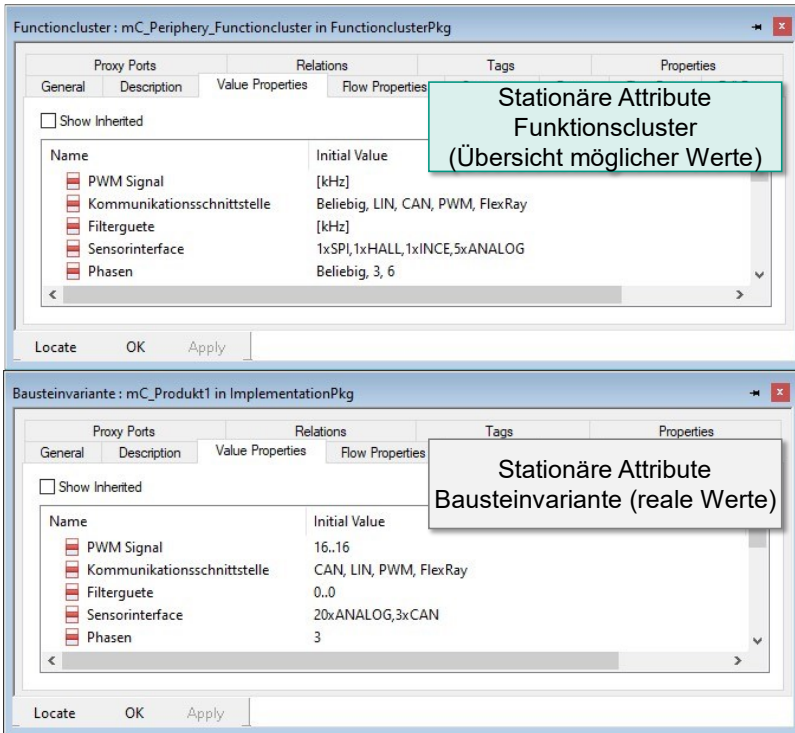


Abbildung 4.41: Übertrag der quantifizierten Attributwerte aus den Bausteinsteckbriefen in IBM Rhapsody.

Ergebnis der Implementierung der Strukturierungsmethode ist das initiale Baukastenmodell des ausgewählten Beispielportfolios der TS 1 Anwendungen.

## 4.4 Diskussion und Zwischenfazit

Die über die Literaturrecherche identifizierten Bedarfe nach einer Methode zur Baukastenstrukturierung wurden im Rahmen einer deskriptiven Studie auf deren Relevanz im industriellen Umfeld geprüft. Hierzu wurden die folgenden Teilforschungsfragen betrachtet:

*FF1.1: Welche Herausforderungen lassen sich aus den Rahmenbedingungen des Produktentwicklungsprozesses mechatronischer Steuergeräte ableiten?*

*FF1.2: Welche Herausforderungen ergeben sich aus der Produktarchitektur mechatronischer Steuergeräte?*

*FF1.3: Welche Bedarfe werden aus Sicht der Entwicklung und des Managements definiert?*

*FF1.4: Welche Methoden des MBSE können zur Unterstützung der Baukastenstrukturierung und –anwendung herangezogen werden?*

Als Ergebnis der empirischen Studie konnten die Bedarfe bestätigt und in Form des folgenden Zielsystems konkretisiert werden (vgl. Kapitel 4.2):

- Vergleichbarkeit integraler und modularer Bausteinstrukturen
- Einfache Erweiterbarkeit und Anpassung der Struktur
- Effiziente Nutzung in der Frühen Phase der Steuergeräteentwicklung
- Klare Schnittstellen der Bausteine und daraus abgeleitet klare Zuständigkeiten
- Konsistenz und Transparenz im Prozess der Anforderungsableitung vom Gesamtsystem auf die Bausteine

Im Anschluss an die deskriptive Studie I wurde auf Basis des ermittelten Zielsystems eine zweigeteilte präskriptive Studie durchgeführt. Diese beleuchtete zunächst die theoretischen Aspekte der Methodenentwicklung (Kapitel 4.2) und gab anschließend Aufschluss über deren praktische Umsetzung innerhalb der vorliegenden Untersuchungsumgebung (Kapitel 4.3). Die Vorgehensweise orientierte sich dabei sequenziell an den zu Grunde gelegten Teilforschungsfragen:

*FF2.1: Wie werden die Baukastenstrukturierung und –anwendung mechatronischer Steuergeräte mittels MBSE-Methoden entsprechend gestaltet?*



Realisierungen der im Portfolio enthaltenen Produkte ermöglicht es, nach klassischen Entwicklungsmethoden generierte Vorarbeiten in Form einer lösungsneutralen Referenz-Funktionsstruktur, unter Berücksichtigung des **Funktions-Gestalt-Zusammenhangs** in die Modulstruktur des Baukastens zu überführen. Hierbei spielt die **aktive Nutzung von Elementen des Referenzsystems** im Sinne der PGE die zentrale Rolle. Die im Rahmen der mechatronischen Steuergeräteentwicklung geforderte Traceability zwischen Anforderungen, Funktionen und technischer Realisierung kann mittels der abgeleiteten Struktur ohne weiteres erfüllt werden. Dabei bietet das iPeM eine ideale Möglichkeit die komplexen und iterativen Vorgänge der Baukastenstrukturierung anhand der Aktivitätenmatrix anschaulich zu beschreiben.

Bei der Implementierung der Strukturierungsmethode am Beispielpportfolio (Kapitel 4.3) erweist sich das **Eigenschafts-Abhängigkeits-Diagramm** (Kapitel 4.3.3), welches eine abstrakte Form eines **Wirk-Netzes** darstellt, als Element von zentraler Bedeutung. In diesem sind die **Schnittstellen der einzelnen Module** in Form von Key-Parametern klar abgegrenzt. Durch die Einschränkung der Anzahl zulässiger Key-Parameter können die Untersuchungen fokussiert werden, womit schnelle, fundierte Aussagen über die Eignung von Konzepten möglich werden. Dementsprechend wurde eine in der Anwendung schlanke, auf den Einsatz im Rahmen der **Frühen Phase der PGE mechatronischer Steuergeräte** ausgerichtete Methode geschaffen. Mittels direkter Einbindung der Experten konnte zudem die Korrektheit der Inhalte iterativ geprüft und somit bereits ein wichtiger Schritt zur Akzeptanz der Methodenanwendung gemacht werden. Darüber hinaus erweist sich das EAN als wichtiges Informationsinstrument der Systementwicklung im Produktentstehungsprozess, da die **Zusammenhänge zwischen den einzelnen Domänen** und deren **Auswirkungen auf das Gesamtsystem** übersichtlich dargestellt werden können. In Zusammenhang mit der Modellierung des initialen Baukastenmodells im branchenweit verbreiteten MBSE-Tool IBM Rhapsody (Kapitel 4.3.5), konnte die Bedeutung von teilautomatisierter Modellierung als wichtige Einflussgröße auf die **Akzeptanz von MBSE-Methoden** unter Nutzung der SysML nachgewiesen werden.

Im Folgenden sollen der **Einfluss der Untersuchungsumgebung** auf die Methodengestaltung sowie mögliche **Ansätze weiterer Forschungsvorhaben** in Kürze diskutiert werden. Durch die breite, bereits innerhalb der Untersuchungsumgebung verfügbare Informationsbasis konnte beim Entwurf, vor allem aber bei der Anwendung der Strukturierungsmethode am Beispielpportfolio viel Zeit gespart werden. Sämtliche, im Rahmen dieser Arbeit, als Vorarbeiten deklarierte Schritte zur Erstellung von Referenz-Funktionsstrukturen, gelten beispielsweise als besonders aufwändig. Weiterhin ist die Mitarbeit der Experten bei

der Methodenanwendung unmittelbare Voraussetzung für den Erfolg der Baukastenstrukturierung. Ohne fundierte Kenntnisse der Technologien ist eine sinnvolle Eingrenzung der Key-Parameter nicht möglich. Die Qualität der Key-Parameter ist letztlich für die Güte der aus dem Baukasten abgeleiteten Produkte verantwortlich. Bezüglich des Einflusses übergeordneter Baukastenarchitekturtypen auf die Methodenanwendung nachgewiesen werden konnte, drängt sich die Frage nach dem idealen Baukastenarchitekturtyp in Abhängigkeit des zu untersuchenden Portfolios auf. Da diese Rahmenbedingung innerhalb der Untersuchungsumgebung bereits vorgegeben war bieten sich hier weitere Nachforschungen an.

## 5 Modellierung der Baukastenanwendung

Dieses Kapitel beschreibt die Entwicklung einer Methode zur teilautomatisierten Anwendung eines Baukastenmodells mit dem Ziel den Schritt der Funktionsmustererstellung in der Frühen Phase der PGE mechatronischer Steuergeräte effizient zu unterstützen. Hierzu werden zunächst die konkreten Anforderungen an die Methode im Rahmen einer deskriptiven Studie ermittelt (Kapitel 5.1). Anschließend werden die theoretischen Grundlagen zur Baukastenanwendung vorgestellt (Kapitel 5.2). Als Basis dient hierbei ein, mittels der Strukturierungsmethode aus Kapitel 4 erstelltes, initiales Baukastenmodell. Die hergeleitete Methode zur Baukastenanwendung wird daraufhin innerhalb der Rahmenbedingungen der Untersuchungsumgebung implementiert (Kapitel 5.3). Zum Abschluss des Kapitels werden die geschaffenen Ergebnisse reflektiert und diskutiert (Kapitel 5.4).

### 5.1 Ermittlung des Zielsystems für die Baukastenanwendung

Die Ermittlung des Zielsystems der Baukastenanwendung erfolgt analog zur Baukastenstrukturierung im Rahmen einer deskriptiven Studie I. Leitfaden der Studie ist die Beantwortung der in Kapitel 3.1.3 hergeleiteten Forschungsfrage hinsichtlich der Aspekte zur Anwendung von Baukästen<sup>1</sup>:

*FF1: Welche Spezifika und Bedarfe im Bereich der Strukturierung und Anwendung von Baukästen existieren in der Frühen Phase der PGE von mechatronischen Steuergeräten und welches Zielsystem an entsprechende Methoden lässt sich daraus ableiten?*

Abbildung 5.1 veranschaulicht den Aufbau der deskriptiven Studie zur Ermittlung des Zielsystems der Baukastenanwendung. Im folgenden Abschnitt erfolgt ein kurzer Überblick über die Elemente der Studie, inklusive daraus abgeleiteter Prämissen: Die Beantwortung der Forschungsfrage erfolgt demnach auf Basis

---

<sup>1</sup> Die im Rahmen von Kapitel 5.1 dargelegten Untersuchungen sind über folgende Publikationen aufgeteilt veröffentlicht worden: Powelske et al. (2017), Powelske et al. (2018).

zweier Teilergebnisse. Diese sind in Zusammenhang mit den zur Ermittlung angewandten Forschungsmethoden und den genutzten Grundlagen dargestellt.

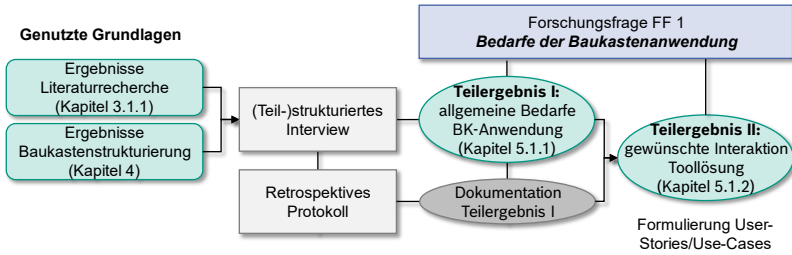


Abbildung 5.1: Übersicht der deskriptiven Studie I der Baukastenanwendung anhand genutzter Grundlagen, empirischer Methoden und zugrundeliegender FF.

Die Ergebnisse der Baukastenstrukturierung (Kapitel 4) stellen das Fundament zur Bedarfsermittlung der Baukastenanwendung dar.

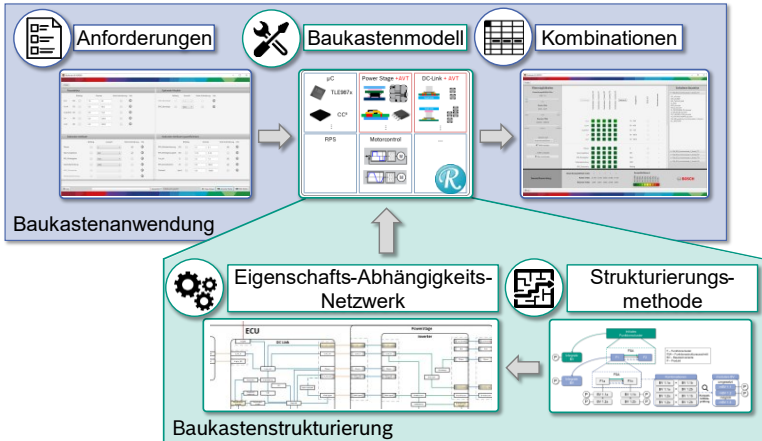


Abbildung 5.2: Übersicht des Zusammenspiels aus Baukastenstrukturierung und Baukastenanwendung für mechatronische Steuergeräte.



Dieser Sachverhalt ist auf Basis des Zusammenspiels beider Methoden in Abbildung 5.2 exemplarisch illustriert. Schnittstelle der Methoden bildet demnach ein MBSE-gestütztes Baukastenmodell mechatronischer Steuergeräte. Dieses wurde als Resultat der Anwendung der Strukturierungsmethode bereits in Kapitel 4 vorgestellt. Für die Baukastenanwendung, nach Bursac (2016) als Deduktionsprozess zu verstehen, lässt sich darauf aufbauend folgende Prämisse ausgeben: basierend auf quantifizierten Kundenanforderungen sollen passende Produktempfehlungen generiert werden. Vor diesem Hintergrund werden im ersten Schritt der Studie, die jeweiligen Bedarfe der Stakeholder an die Methode zur Unterstützung des Deduktionsprozesses ermittelt. Aus den Bedarfen werden, analog zur Baukastenstrukturierung, geeignete Ansatzpunkte für die methodische Unterstützung identifiziert. Anhand dieser Schlüsselfaktoren wird dem Leser, das vom Forscher erlangte Verständnis über die Zusammenhänge, in Form eines Reference Models präsentiert (**Teilergebnis I**). Im Gegensatz zur Baukastenstrukturierung weist die Methode der Baukastenanwendung eine deutlich stärkere Abhängigkeit von der späteren Umsetzung auf. In der Literaturrecherche konnten auf dem Gebiet des MBSE bereits zielführende Ansätze ermittelt werden. Demnach stellt der gezielte Einsatz einer Teilautomatisierung basierend auf der Nutzung eines föderativen Ansatzes eine sinnvolle Maßnahme für die Umsetzung der Baukastenanwendung dar. Diese Maßnahme wird zusätzlich zum initialen Baukastenmodell als Grundlage für die Bedarfserhebung genutzt. Als zweites Teilergebnis der Studie wird dementsprechend die gewünschte Interaktion der Stakeholder mit einer toolgestützten Baukastenanwendung ermittelt. In den folgenden Unterkapiteln werden die einzelnen Teilergebnisse und die Vorgehensweise zur Datenermittlung beleuchtet.

### **5.1.1 Ermittlung der Bedarfe an die Baukastenentwicklung**

Zur Ermittlung der stakeholderspezifischen Bedarfe an die Baukastenanwendung wurden die empirischen Methoden „teilstrukturiertes Interview“ und „retrospektives Protokoll“ angewendet. Der Teilnehmerkreis belief sich dabei auf die gleichen Personen und Stakeholdergruppen, der im Rahmen von Kapitel 4.1 präsentierten Studie (n=29). Zu Beginn der Interviews<sup>2</sup> wurden den Stakeholdern die Zusammenhänge zwischen Baukastenstrukturierung und -anwendung anhand Abbildung 5.2 vorgestellt. Davon ausgehend wurde gezielt nach deren individuellen Bedarfen an die Anwendung des Baukastenmodells bei der deduktiven Ableitung

---

<sup>2</sup> Der Interviewleitfaden kann dem Anhang der vorliegenden Arbeit entnommen werden

von Produkten gefragt. Um den Befragten möglichst viel Zeit zur Schilderung der Bedarfe zu geben, erfolgte die Ergebnissicherung vom Autor der vorliegenden Arbeit in Form retrospektiver Protokolle. Die Protokolle wurden im Nachgang mit den Befragten auf deren Vollständigkeit geprüft und bei Bedarf ergänzt. Anschließend wurden die genannten Bedarfe abhängig von den Stakeholdergruppen zusammengefasst und Schlüsselfaktoren der Baukastenanwendung formuliert. Die ermittelten Schlüsselfaktoren (Abbildung 5.3) und das darauf aufbauende Reference Model werden im folgenden Abschnitt vorgestellt:

**Teilergebnis I: allgemeine Bedarfe an die Baukastenanwendung**

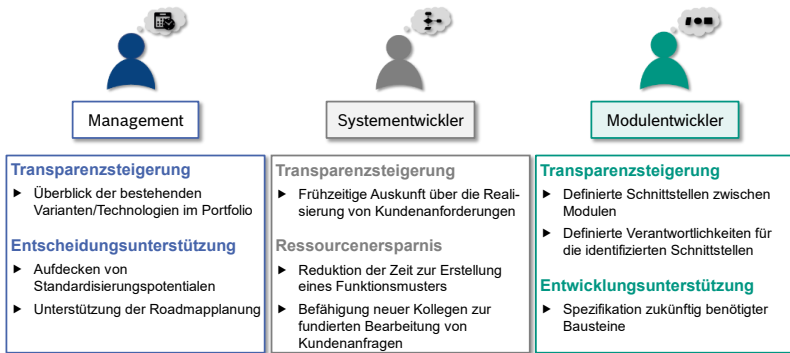


Abbildung 5.3: Potenzielle Schlüsselfaktoren der Baukastenanwendung für mechatronische Steuergeräte. Ermittelt auf Basis (teil-)strukturierter Interviews mit Vertretern des Managements, der System- und Modulentwicklung.

Das **Management** hat durch seine direkte Beteiligung an der Kundenkommunikation<sup>3</sup> den akuten Bedarf jederzeit einen transparenten Überblick der bestehenden Varianten/Technologien im Portfolio geben zu können. Für die Strategieentwicklung besteht darüber hinaus der Wunsch, mittels der Baukastenanwendung gezielt Standardisierungspotentiale aufdecken zu können. Hiermit sollen langfristig Kosten in Entwicklung und Produktion gesenkt werden.

<sup>3</sup> Dies kann entsprechend der Ausführungen in Kapitel 3.3.2 zu einem unternehmensexternen oder -internen Kunden realisiert werden.

Über die Identifikation neuer Produktkonzepte soll eine Erweiterung des Portfolios realisiert und damit eine direkte Entscheidungsunterstützung bei der Roadmapplanung geleistet werden.

Zusammen mit dem Management liegen die Bedarfe der **Systementwicklung** vor allem darin, eine frühestmögliche Aussage über die Realisierbarkeit der Kundenanfragen treffen zu können. Gerade in der Frühen Phase der PGE, in der die Anforderungen seitens des Kunden sehr vage sind, soll die Baukastenanwendung zur Konkretisierung des initialen Zielsystems beitragen. Als messbarer Erfolgsfaktor der Baukastenanwendung wurde an dieser Stelle eine Verkürzung der Zeit bis zur Funktionsmustererstellung identifiziert.<sup>4</sup> Neuen Kollegen soll durch die Baukastenanwendung die Möglichkeit gegeben werden Kundenanfragen, auf Basis des in den Baukasten eingeflossenen Expertenwissens, bereits nach kurzer Einarbeitungszeit bearbeiten zu können. Grundlage für die von der Systementwicklung adressierten Punkte ist eine verlässliche Methode zur teilautomatisierten Kompatibilitätsprüfung.

Aus Sicht der **Domänenentwickler**, die mit der Entwicklung der Module betraut sind, ist die Rückverfolgbarkeit der Schnittstelleninformationen und deren Quantifizierung bei der Baukastenanwendung zu gewährleisten. Weiterhin soll über die Anwendung des Baukastens die Entwicklung neuer Technologien unterstützt werden können. Beispielsweise durch eine Quantifizierung der Key-Parameter des entsprechenden Moduls.

Auf Basis der ermittelten Schlüsselfaktoren wurde analog zur Baukastenstrukturierung ein Reference Model der Baukastenanwendung erstellt. Entsprechend der unterschiedlichen Bedarfe der Stakeholdergruppen wurde das Reference Model in zwei Teile aufgeteilt: eine **Managementsicht** (Abbildung 5.4) und eine **Entwicklersicht** (Abbildung 5.5). Zur Wahrung der Übersichtlichkeit wurde im Falle von Literaturbelegen jeweils nur eine Quelle stellvertretend angegeben. Weitere zutreffende Belege für die dargestellten Zusammenhänge sind Kapitel 2 zu entnehmen. Anhand der Reference Models kann ebenfalls die direkte Abhängigkeit der Baukastenanwendung von den Schlüsselfaktoren der Baukastenstrukturierung illustriert werden.

---

<sup>4</sup> In Anlehnung an die DRM

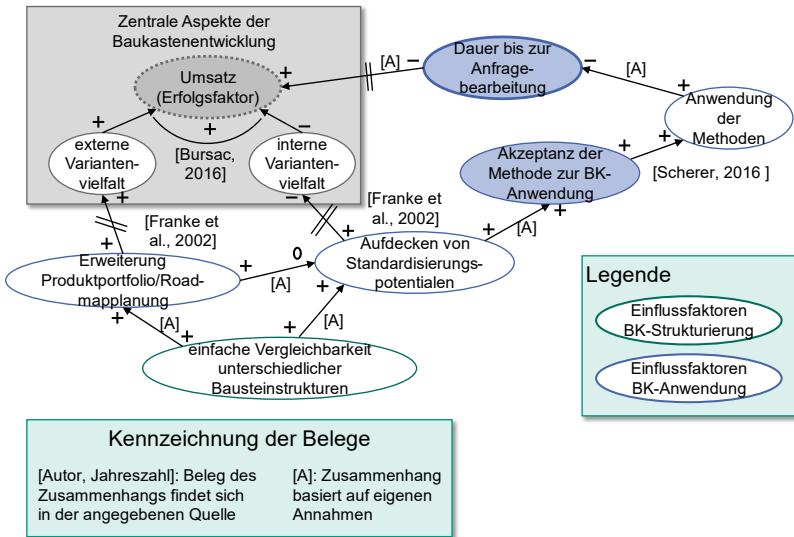


Abbildung 5.4: Reference Model der Baukastenanwendung (Managementsicht) entsprechend der DRM nach Blessing & Chakrabarti (2009).

Aus Sicht des Managements zählen die ökonomischen Aspekte der Baukastenentwicklung als übergeordnete Erfolgsfaktoren. Diese können auf Basis bestehender wissenschaftlicher Erkenntnisse, sowohl durch Beiträge zur Standardisierung als auch zur Technologieplanung positiv beeinflusst werden. Gleichzeitig erscheint die von den Entwicklern erwähnte Dauer zur Funktionsmustererstellung ein geeigneter Hebel zu sein. Zielt die zu erwirkende methodische Unterstützung auf eine Reduktion der genannten Zeit ab, ist diese ebenfalls von der Methodenakzeptanz abhängig. Aus der Entwicklersicht wird eine direkte Abhängigkeit von den Schlüsselfaktoren der Baukastenstrukturierung deutlich. Die Akzeptanz der Methode lässt sich an dieser Stelle durch die Erfüllung der ermittelten technischen Bedarfe beeinflussen.

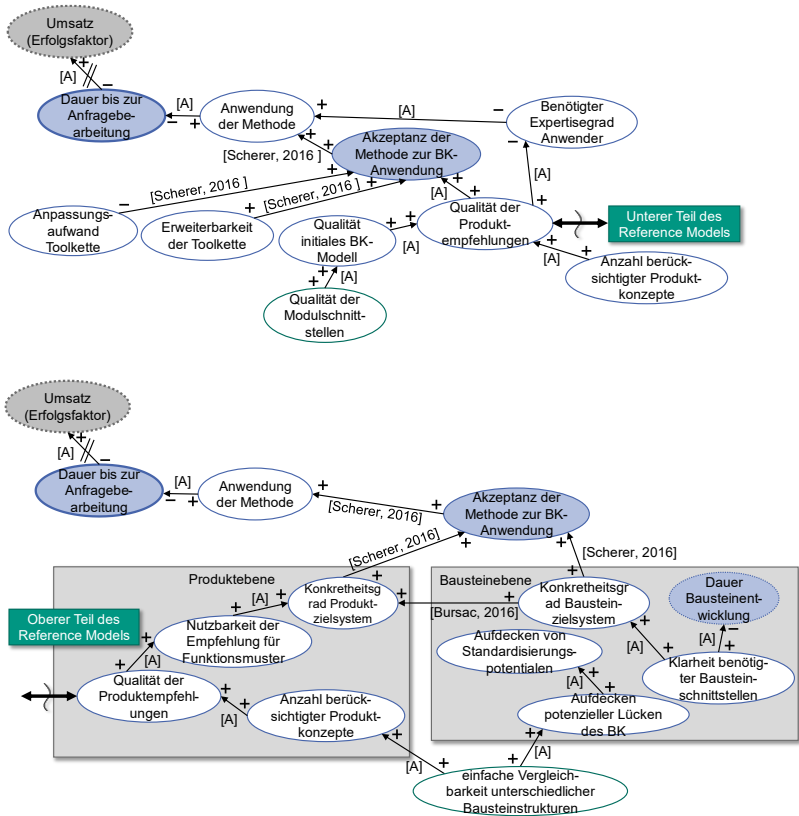


Abbildung 5.5: Reference Model der Baukastenanwendung (Entwicklersicht) entsprechend der DRM nach Blessing & Chakrabarti (2009).

### 5.1.2 Ermittlung der gewünschten Stakeholderinteraktion

Um die teilautomatisierte Anwendung des Baukastens hinsichtlich der späteren **Implementierung** (Kapitel 5.3) auf die genannten Bedürfnisse auszurichten, wurden zusammen mit den Stakeholdern User-Stories formuliert. Diese beschreiben die gewünschte Interaktion des Stakeholders mit der toolgestützten Baukastenanwendung aus dessen Blickwinkel. Hierzu wurden die Vertreter der Stakeholdergruppen gebeten, deren Interaktionswunsch in der Form „Als **<Stakeholder XY>** möchte ich **<Interaktionswunsch>**“ zu formulieren. Der Teilnehmerkreis gestaltet sich dabei analog zu den teilstrukturierten Interviews.

Die formulierten Interaktionswünsche wurden im Nachgang zusammen mit den Stakeholdern ausgewertet und eine entsprechende grafische Repräsentation der Interaktion erarbeitet. Für die Baukastenanwendung wurde im Vorfeld seitens der Stakeholder die Vorgabe gemacht, dass durch die Eingabe von Kundenanforderungen geeignete Modulkonfigurationen ausgegeben werden können. Bei der Formulierung der User-Stories wird daher vorausgesetzt, dass der Anwender über eine Eingabe- und eine Ausgabemaske mit dem entwickelten Baukastenmodell interagieren kann (Abbildung 5.6).

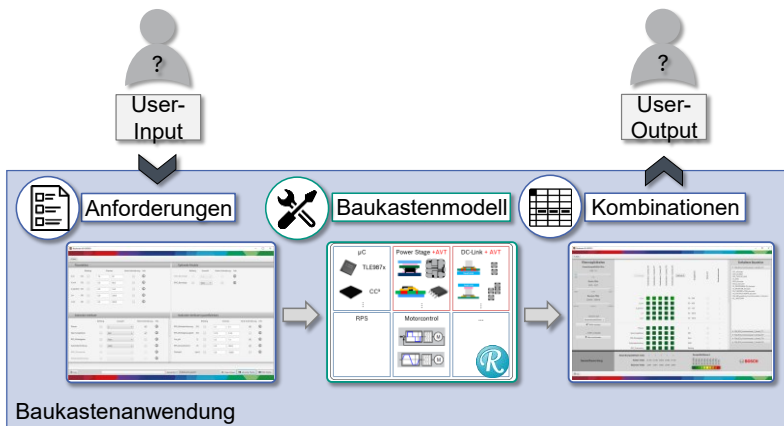


Abbildung 5.6: Interaktion des Nutzers mit der entwickelten Baukasten-anwendung.

**Teilergebnis II: Use-Cases für die teilautomatisierte Baukastenanwendung**

Abbildung 5.7 zeigt die mit den Stakeholdergruppen entwickelten User-Stories. Ausgehend von den User-Stories wurden Use-Cases für die Implementierung der Baukastenanwendung ermittelt.

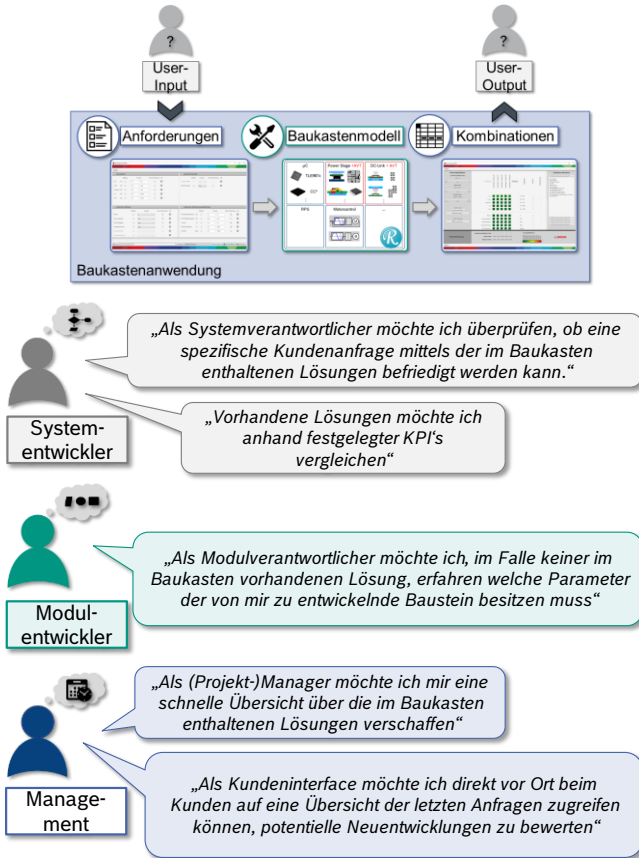


Abbildung 5.7: Übersicht der stakeholderspezifischen User-Stories für die entwickelte Baukastenanwendung.

Hierzu wurden die subjektiven Interaktionswünsche auf ihre Kernaussage hin analysiert und objektiviert. In Abbildung 5.8 ist die Zuordnung der Use-Cases zu den entsprechenden User-Stories illustriert:

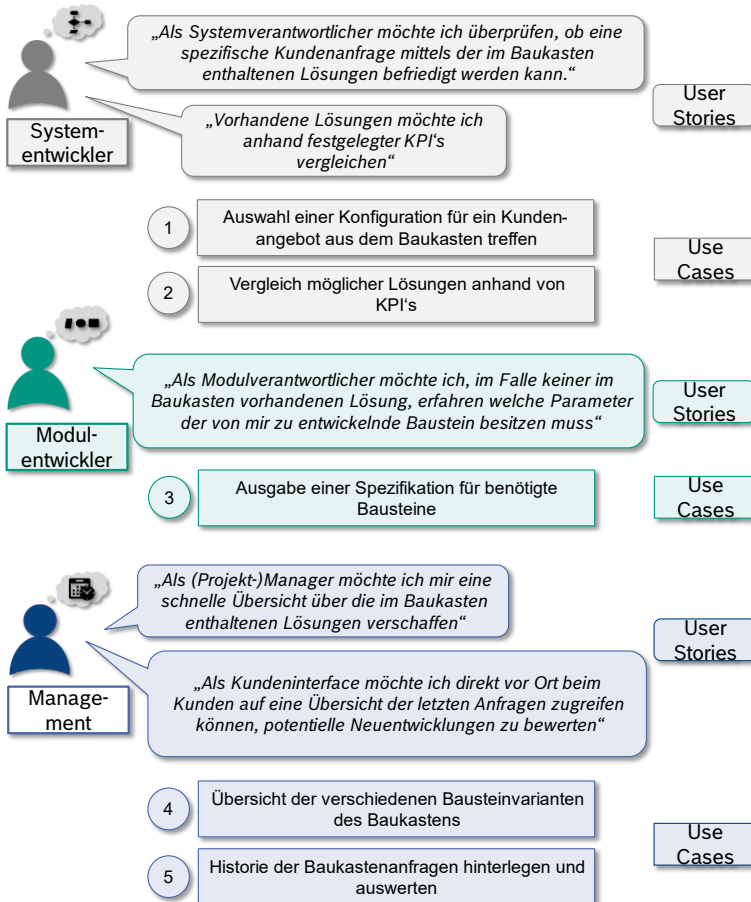


Abbildung 5.8: Zuordnung der abgeleiteten Use-Cases zu den User-Stories für die entwickelte Baukastenanwendung.



Betrachten wir im Weiteren die einzelnen Use-Cases hinsichtlich der agierenden Stakeholdergruppen:

**Use-Case 1 (Systementwicklung):**

Für die Auswahl einer Konfiguration aus dem Baukasten muss dem Systementwickler die Möglichkeit gewährt werden spezifische Kundenanforderungen in der Eingabemaske einzutragen. Auf dieser Basis müssen die Produktkonfigurationen bezüglich ihres jeweiligen Erfüllungsgrades der Kundenanforderungen ausgegeben werden. Empfänger dieser Empfehlung ist ebenfalls der Systementwickler (Abbildung 5.9).

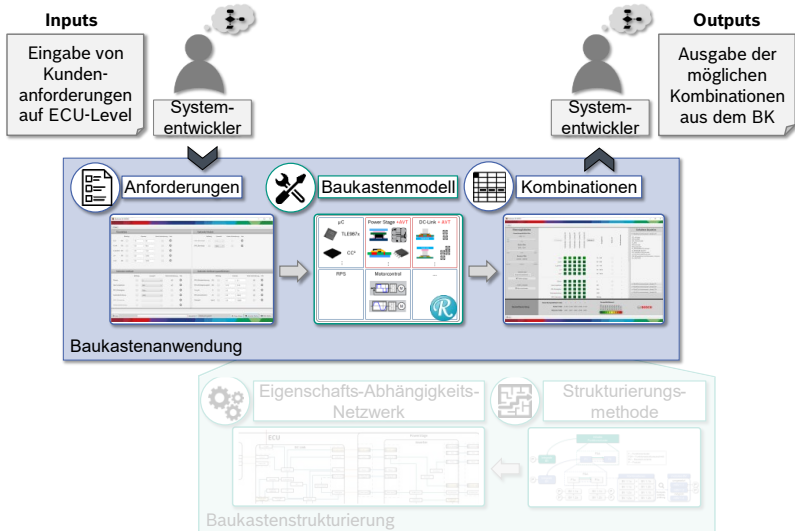


Abbildung 5.9: Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 1

### Use-Case 2 (Systementwicklung):

Um technisch mögliche Kombinationen gezielt anhand von KPI's vergleichen zu können, muss dem Systementwickler eine dementsprechende Sortierfunktion bereitgestellt werden. Voraussetzung hierfür ist eine Liste der zu berücksichtigenden KPI's (Abbildung 5.10).

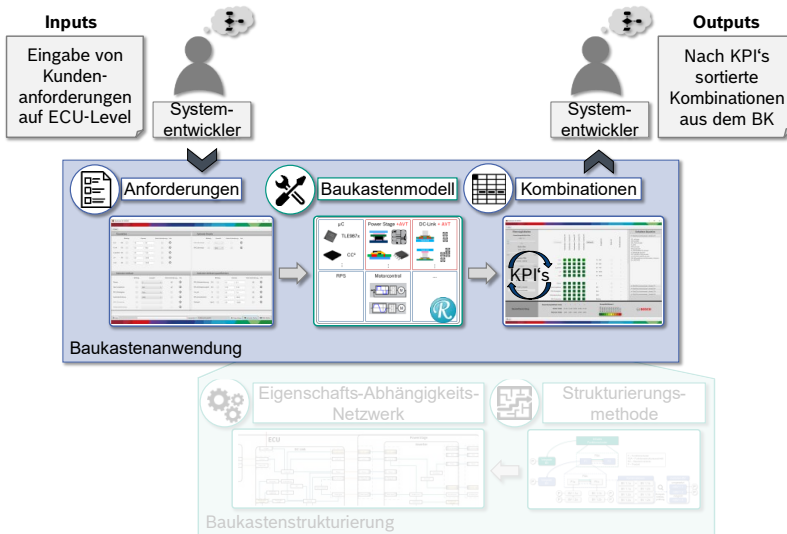


Abbildung 5.10: Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 2.

### Use-Case 3 (System-/Modulentwicklung):

Die Ausgabe von Spezifikationen für benötigte Bausteine soll dem Modulentwickler, im Falle fehlender oder inkompatibler Bausteine, im Anschluss an die Auswertung von Kundenanfragen bereitgestellt werden. Die Bearbeitung der Kundenanfrage erfolgt im Zuge der Use-Cases eins und zwei durch den Systementwickler (Abbildung 5.11).

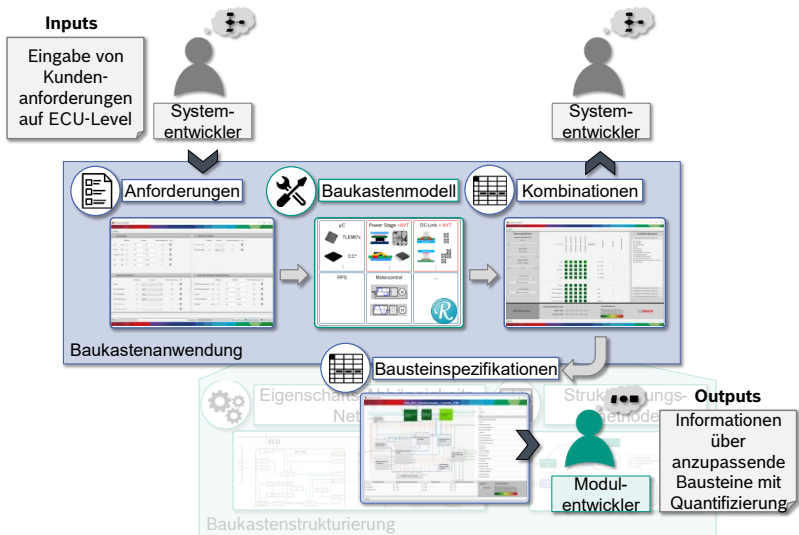


Abbildung 5.11: Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 3.

#### Use-Case 4 (Management):

Zur Erstellung der Variantenübersicht für das Management kann von der gleichen Fragestellung ausgegangen werden, die auch Use-Case 1 zugrunde liegt. Hierbei muss zusätzlich überprüft werden in welcher Form das benötigte Baukastenmodell dem Management vorliegt (Abbildung 5.12).

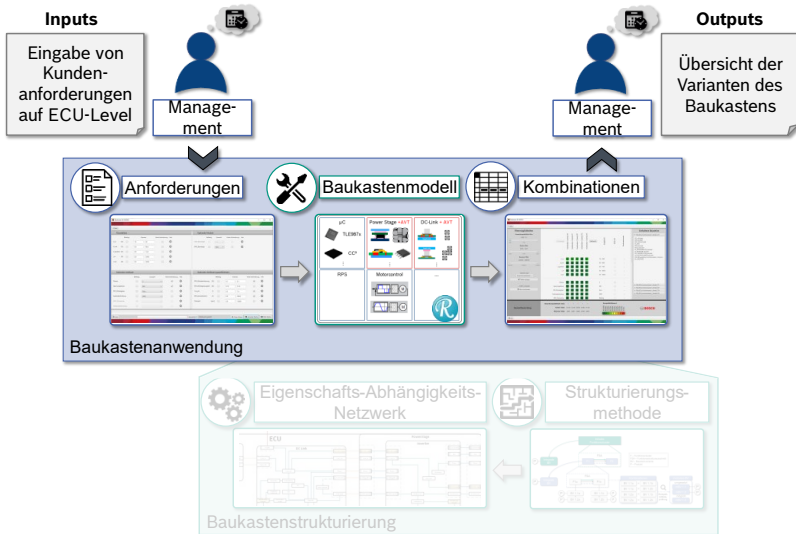


Abbildung 5.12: Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 4.

**Use-Case 5 (Management):**

Auf Basis der einzelnen Kundenanfragen ist eine Historie der Anfragen abzuleiten und dem Management in kondensierter Form zur Verfügung zu stellen (Abbildung 5.13). In Bezug auf die Implementierung der teilautomatisierten Baukastenanwendung unter den Rahmenbedingungen der Untersuchungs-umgebung, wurde die Nutzung eines föderierten Ansatzes von allen Stakeholdergruppen als zielführend eingeschätzt.

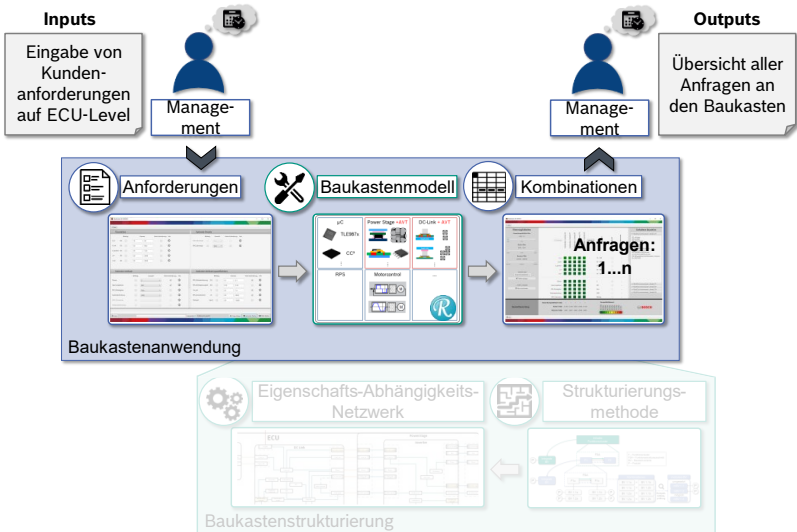


Abbildung 5.13: Interaktionskonzept des Benutzers mit der entwickelten Baukastenanwendung im Use-Case 5.

## 5.2 Entwicklung einer Methode zur Baukastenanwendung

Basierend auf den Erkenntnissen der deskriptiven Studie I wird analog zur Baukastenstrukturierung eine zweigeteilte präskriptive Studie zur Entwicklung und Implementierung der Baukastenanwendung durchgeführt. Dies geschieht mit dem Ziel, die folgende Forschungsfrage hinsichtlich der Aspekte der Baukastenanwendung zu beantworten:

*FF2: Wie sind die Strukturierung und Anwendung eines Baukastens mechatronischer Steuergeräte auf Basis des ermittelten Zielsystems mittels Methoden des MBSE zu gestalten und zu integrieren?*

Ausgangspunkt für die Baukastenanwendung ist, ein auf Basis der Strukturierungsmethode aus Kapitel 4, erstelltes Baukastenmodell. Dieses enthält

neben den identifizierten Modulen<sup>5</sup> und deren Key-Parameter, quantifizierte integrale Bausteinvaryanten, die den technischen Realisierungen der Module entsprechen (Abbildung 5.14). Die Modulverbindungen sind im Baukastenmodell entsprechend der im EAN ermittelten internen und externen Wirkzusammenhänge realisiert. Dabei sind die Wirkbeziehungen des Eigenschafts-Abhängigkeit-Netzwerks (EAN) entsprechend den unterschiedlichen Abstraktionsebenen auf die verschiedenen Funktionsstrukturausschnitte aufgeteilt. Durch die Nutzung des Baukastenmodells aus der Strukturierungsmethode befindet sich die Anwendung des Baukastens ebenfalls in direkter Abhängigkeit zum übergeordneten Baukastenarchitekturtypen (singulär, hybrid, autark) (vgl. Kapitel 2.4.2).

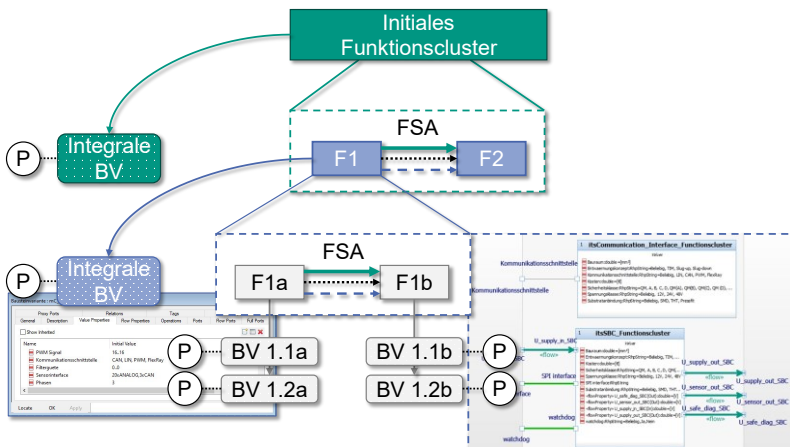


Abbildung 5.14: Übersicht der Ausgangslage nach Anwendung und Implementierung der Strukturierungsmethode für mechatronische Steuergeräte in IBM Rhapsody.

Aus der Stakeholderbefragung lässt sich das **primäre Ziel der Baukastenanwendung** formulieren. Dieses liegt darin, ausgehend von Kundenanforderungen, den Deduktionsprozess von Produkten aus dem Baukasten zu unterstützen. In Zuge dessen, sollen mögliche Realisierungen eines gewünschten Produktes, in Form ausgewählter Bausteinkombinationen

<sup>5</sup> In Form von Funktionsclustern

gegeneinander abgewogen werden können. Vergleichbar zur Entwicklung der Strukturierungsmethode fokussiert sich die Betrachtung auf ein Wechselspiel der iPeM Aktivitäten „Wissen managen“ und „Prinzip und Gestalt modellieren“. Mittels der Baukastenanwendung sind vorwiegend die SPALTEN Prozesse „alternative Lösungen aufzeigen“ und „Lösungsauswahl“ zu unterstützen. Wie bereits bei der Baukastenstrukturierung angeklungen ist, liegt der Schlüssel zur Baukastenanwendung in der Gestaltung einer verlässlichen Kompatibilitätsbetrachtung. Genauer betrachtet ist beim Deduktionsprozess sowohl die Kompatibilität zwischen einzelnen Modulen als auch zwischen einzelnen Kundenanforderungen und der Bausteinkombination einer ECU sicher zu stellen. Im Rahmen der Methodenentwicklung ist es gewissermaßen notwendig die beiden Arten der Kompatibilitätsbetrachtung zu unterscheiden. Da sich diese direkt an den internen und externen innerhalb eines EAN dargestellten Wirkbeziehungen orientieren, sprechen wir im Folgenden von interner und externer Kompatibilität (Abbildung 5.15). Während sich die Elemente der externen Kompatibilität aus den Zielsystemen der, das Steuergerät umgebenden Systeme ableiten lässt, beschreibt die interne Kompatibilität die Elemente der Zielsysteme im Steuergerät befindlicher Teilsysteme.

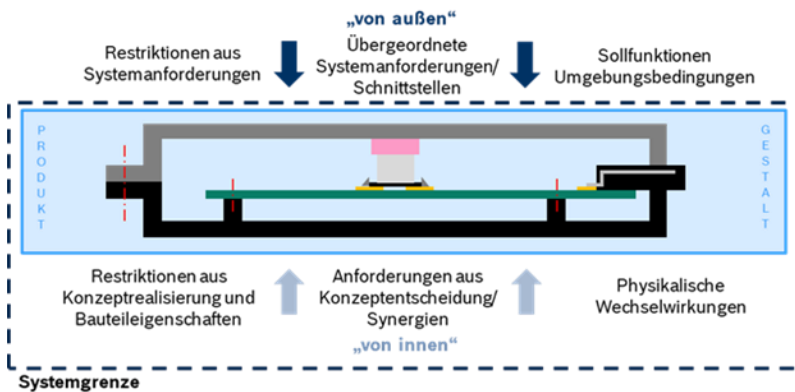


Abbildung 5.15: „Interne“ und „externe“ Anforderungen an ein Produkt als Grundlage der Kompatibilitätsbetrachtung. Interne bzw. externe Anforderungen sind dabei Elemente des Zielsystems von Teilsystemen bzw. übergeordneten Systemen.

Entsprechend der in Kapitel 4 beschriebenen Vorgehensweise werden bei der Vorstellung der Baukastenanwendung Theorie und praktische Umsetzung ebenfalls sequenziell behandelt. Das vorliegende Unterkapitel befasst sich daher mit der Beantwortung der folgenden Teilforschungsfrage in Bezug auf die Baukastenanwendung:

*FF2.1: Wie werden die Baukastenstrukturierung und –anwendung mechatronischer Steuergeräte mittels MBSE-Methoden entsprechend gestaltet?*

Um im Nachgang eine tiefere Auseinandersetzung mit der theoretischen Kompatibilitätsprüfung zu ermöglichen (Kapitel 5.2.1), werden die Charakteristika der Kompatibilitätsarten zunächst vorgestellt und entsprechend den Vorarbeiten der Baukastenstrukturierung eingeordnet:

- **Interne Kompatibilität:**

Die interne Kompatibilität beschreibt die Vorgänge zur Bildung modularer Bausteinvarianten entlang des Strukturbaums. Hierzu werden integrale Bausteinvarianten für die beteiligten Module ausgewählt und entsprechend der zugrunde liegenden Wirkkette kombiniert (Abbildung 5.16).

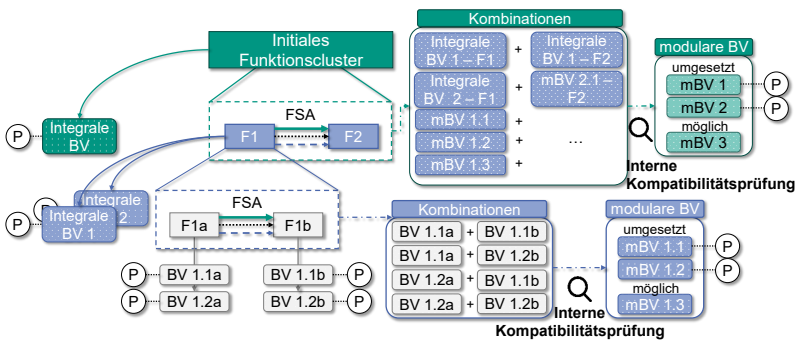


Abbildung 5.16: Verortung der internen Kompatibilitätsbetrachtung entsprechend der Methode zur Baukastenstrukturierung für mechatronische Steuergeräte. Modulare BV eines untergeordneten FSA werden nach deren Prüfung eine Ebene höher wiederum zur Kombination herangezogen.



Der Prozess der internen Kompatibilitätsprüfung dient dem automatisierten Aufzeigen alternativer Lösungen im Sinne der SPALTEN-Methode. Die Auswahl eines spezifischen integralen Bausteins für ein Modul stellt dabei das Treffen einer Designentscheidung dar. Diese Entscheidung schränkt den Freiheitsgrad der umgebenden Module bzw. deren integraler Bausteinvarianten hinsichtlich der möglichen Kompatibilität ein. Ergebnis einer erfolgreichen Kompatibilitätsprüfung ist eine modulare Bausteinvariante, die anhand der Grenzen des übergeordneten FSA parametrisiert ist. Diese steht im Nachgang wiederum für die Kombinatorik auf einer höheren Abstraktionsebene zu Verfügung. Wird die Prüfung der internen Kompatibilität über alle Ebenen des Baukastenmodells durchgeführt, sind am Ende alle technisch realisierbaren Bausteinkombinationen zur Abbildung einer ECU enthalten. Bei der Prüfung der internen Kompatibilität sind demnach neben den Schnittstellen zwischen den Modulen und deren Quantifizierung in Form ausgewählter integraler Bausteinvarianten, ebenfalls die Schnittstellen zur Systemgrenze zu betrachten. Die Sammlung aller technisch möglichen Kombinationen für eine ECU entspricht dem durch den Baukasten aufgespannten Lösungsraum<sup>6</sup>. Entsprechend der in Kapitel 4 vorgestellten Strukturierungsmethode sind viele der modularen Bausteinvarianten bereits in einem oder mehreren Produkten realisiert<sup>7</sup>. Existiert eine praktische Umsetzung einer modularen Bausteinvariante können deren Werte zur Kompatibilitätsprüfung mit herangezogen werden.

- **Externe Kompatibilität:**

Unter externer Kompatibilität wird im Rahmen der vorliegenden Arbeit der Abgleich des durch den Kunden aufgespannten Zielsystems mit den parametrisierten Kombinationen des Baukastens verstanden. Basierend auf dem Ergebnis der externen Kompatibilitätsprüfung kann dem Anwender, die als Ziel der Baukasten-anwendung proklamierte Unterstützung beim Deduktionsprozess zur Verfügung gestellt werden. Da die Kundenanforderungen an ein Produkt in der Regel eine Black-Box Beschreibung der gewünschten Funktionalität bzw. des gewünschten Verhaltens beinhalten, wird die Systemgrenze der ECU als Betrachtungsmittelpunkt der externen Kompatibilitätsprüfung definiert. Sämtliche Vorgaben oder Constraints an die externe Kompatibilität lassen sich aus dem

---

<sup>6</sup> Der Begriff des Lösungsraums entspricht dem von Albers im erweiterten ZHO-Modell entwickelten (Albers, Lohmeyer & Ebel, 2011)

<sup>7</sup> Gleiches gilt an dieser Stelle auch für Kombinationen auf ECU Ebene. In diesem Falle entspräche die gewählte Kombination exakt einem bereits umgesetzten Produkt.

Zielsystem des übergeordneten technischen Systems ableiten. In der Baukastenstrukturierung wurden neben den Key-Parametern der Module ebenfalls die Key-Parameter des Gesamtsystems ermittelt. Die im Lösungsraum des Baukastens enthaltenen möglichen Kombinationen sind auf Basis der internen Kompatibilitätsbetrachtung bereits auf Basis der Key-Parameter einer ECU quantifiziert. Um den gewünschten Abgleich mit den Kundenanforderungen vollziehen zu können müssen diese zunächst entsprechend der Key-Parameter aufbereitet werden. Die eigentliche Kompatibilitätsprüfung findet in der anschließenden Gegenüberstellung der Kundenanforderungen mit den Kombinationen aus dem Baukasten statt. Hierbei werden die enthaltenen Bausteinkombinationen auf deren Erfüllungsgrad bezüglich der Kundenanforderungen bewertet (Abbildung 5.17). Während die interne Kompatibilitätsprüfung die Lösungsalternativen aufzeigt, kann mittels externer Kompatibilitätsprüfung ein Beitrag zur Unterstützung der Lösungsauswahl geleistet werden.

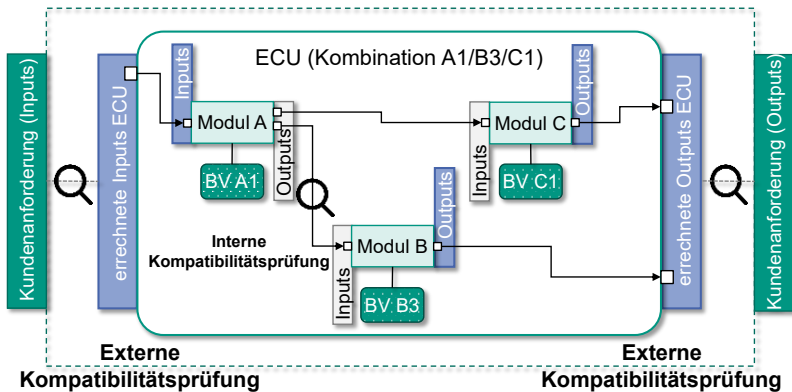


Abbildung 5.17: Externe und interne Kompatibilität eines mechatronischen Steuergerätes am Beispiel einer spezifischen Bausteinkombination (A1, B3, C1) der enthaltenen Module.

Die Ergebnisse beider Arten von Kompatibilitätsprüfungen können weiterhin dazu genutzt werden Lücken des Baukastens zu identifizieren und die Entwicklung neuer Technologien und Funktionalitäten aktiv zu steuern (Kapitel 5.2.2). Hierüber können die Bedarfe des Managements und der Modulentwicklung hinsichtlich der Baukastenanwendung adressiert werden. Anhand der externen Kompatibilitätsbewertung wird zunächst ermittelt, ob der Baukasten bereits

passende Kombinationen enthält. An dieser Stelle lassen sich funktionale und technologische Lücken voneinander unterscheiden. Nicht im Baukasten enthaltene Funktionalitäten, äußern sich, durch zusätzliche bisher nicht erfasste Schnittstellen an der Systemgrenze der ECU. Die Identifikation technologischer Lücken gelingt auf Basis der Kopplung beider Kompatibilitätsbetrachtungen. Kann die vom Kunden gewünschte Parametrierung durch die Kombinationen des Baukastens beispielsweise nicht oder nur teilweise erfüllt werden, können die internen Beziehungen zwischen den Modulen analysiert werden. Auf diese Weise wird festgestellt, welche der Bausteinvarianten an der Abweichung beteiligt sind. Die daraus resultierenden Informationen können anschließend zur Steuerung der Technologieentwicklung herangezogen werden.

Bis hierher gewonnene Erkenntnisse, zeigen den deutlichen Einfluss der Baukastenstrukturierung auf die Baukastenanwendung. An den Überlegungen zur Kompatibilitätsbetrachtung, kann nun entsprechend dem Makrozyklus von Bursac (2016) illustriert werden, welchen Einfluss die Baukastenanwendung auf die Strukturierung des Baukastens hat. Sobald ein Produkt aus dem Baukasten abgeleitet wurde, wird dieses inklusive der potenziellen technologischen und funktionalen Erweiterungen wieder in den Kreislauf aus Induktion und Deduktion aufgenommen. Hieraus erweitert sich das, in der Strukturierung des Baukastens, zu berücksichtigende Portfolio. Die Struktur des Baukastens selbst befindet sich also entsprechend der Definition der Baukastenentwicklung als kontinuierliche, der Produktentwicklung parallele Aktivität, in einem stetigen Weiterentwicklungsprozess (Abbildung 5.18). In den folgenden Unterkapiteln werden die theoretischen Grundlagen der Kompatibilitätsprüfungen (Kapitel 5.2.1) sowie die Indikation potenzieller Lücken des Baukastens (Kapitel 5.2.2) beleuchtet. Die konkreten Use-Cases der einzelnen Stakeholder (Kapitel 5.1.2) werden aufgrund der starken Abhängigkeit von der letztendlichen Implementierung in Kapitel 5.3 behandelt.

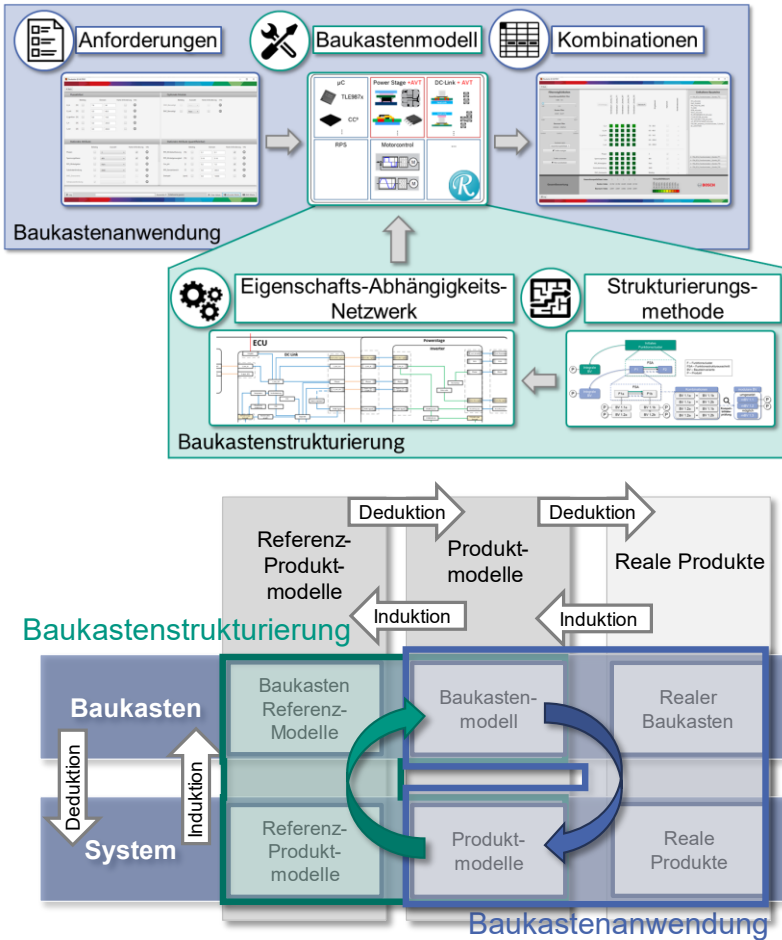


Abbildung 5.18: Einfluss von Baukastenstrukturierung und -anwendung für mechatronische Steuergeräte auf den Zyklus aus Induktion und Deduktion im Framework für die Baukastenentwicklung nach Bursac (2016).

## 5.2.1 Gestaltung der Kompatibilitätsprüfung

Entsprechend der verschiedenen Kompatibilitätsarten zur Baukastenanwendung werden die zugehörigen Prüfungen nachfolgend theoretisch hergeleitet. Die präsentierten Inhalte stützen sich dabei auf die Ergebnisse von Zanger (2018)<sup>8</sup>.

### Ableitung der Kompatibilitätstypen inklusive Auswahlkriterien

Im vorherigen Kapitel ist bereits angeklungen, dass die Kompatibilitätsprüfungen anhand der Key-Parameter, also den Schnittstellen zwischen Modulen und dem Gesamtsystems durchgeführt werden. Bei den einzelnen Key-Parametern handelt es sich um eine Vielzahl verschiedener Eigenschaften technischer Funktionsträger. Wie in Kapitel 4.3.4 beschrieben, wird im Rahmen der vorliegenden Arbeit zwischen Flussattributen, quantifizierbaren stationären Attributen und nicht-quantifizierbaren stationären Attributen unterschieden. Diese Attributklassen zeichnen sich durch ihre individuelle Beschaffenheit in Bezug auf die verknüpften Attributwerte aus. Um diesem Umstand Rechnung zu tragen, wurden sogenannte Kompatibilitätstypen eingeführt, denen die spezifischen Attribute zugeordnet werden können. Hinter jedem der Kompatibilitätstypen verbirgt sich eine eigene Methode zur Prüfung von Schnittstellenkompatibilitäten. Die einzelnen Kompatibilitätstypen können sowohl für die interne als auch externe Kompatibilitätsprüfung genutzt werden. In Zusammenhang mit den mechatronischen Steuergeräten wurden die folgenden vier Kompatibilitätstypen ermittelt:

- **Intervallverrundung:**

Die Intervallverrundung bezieht sich auf Attribute, deren Attributwerte einem Zahlenintervall<sup>9</sup> entsprechen. Dazu zählen wir Flussattribute, wie beispielsweise Ströme und Spannungen und quantifizierbare stationäre Attribute, wie z.B. Sensorgenauigkeiten (vgl. Kapitel 4.3.3). Zur Prüfung, ob die Attributwerte zweier Lösungen entsprechend der Intervallverrundung als kompatibel zählen, wird die Schnittmenge beider Intervalle gebildet. Das resultierende stellt den Bereich dar, in dem die Lösungen bezüglich des untersuchten Key-Parameters kompatibel sind. Dieser Vorgang ist am Beispiel des Key-Parameters Eingangs- bzw.

---

<sup>8</sup> Co-Betreute Abschlussarbeit

<sup>9</sup> Im Rahmen der vorliegenden Arbeit werden sämtliche Intervalle als im mathematischen Sinne geschlossen betrachtet. Dies bedeutet, dass die Grenzen des Intervalls im Zahlenbereich des Intervalls inkludiert sind.

Ausgangsstrom zweier über entsprechende Wirkbeziehungen verbundener integraler Bausteine dargestellt:

*Stromintervall Baustein A:  $\{x|1 \leq x \leq 5\}$*   
*Stromintervall Baustein B:  $\{x|3 \leq x \leq 10\}$*   
*resultierendes Stromintervall:  $\{x|3 \leq x \leq 5\}$*

Existiert keine Schnittmenge zwischen zwei Intervallen, so gelten die auf Kompatibilität geprüften Elemente als inkompatibel.

- **Logische Schnittmenge:**

Der Kompatibilitätstyp „logische Schnittmenge“ wird für all diejenigen Attribute angewendet, denen eine textuelle Liste möglicher Attributwerte zugeordnet ist. Hierzu zählen nicht-quantifizierbare stationäre Attribute und die zur Kennzeichnung optionaler Module verwendeten Attribute.

Kompatibilitätstyp logische Schnittmenge

Attribut Kommunikations-schnittstelle	Baustein A1	Baustein B3	Ergebnis logische Schnittmenge
	LIN CAN Flexray	LIN CAN	LIN CAN ✓ Kombination kompatibel
Attribut Kommunikations-schnittstelle	Baustein A2	Baustein B3	Ergebnis logische Schnittmenge
	Flexray	LIN CAN	✗ Kombination inkompatibel

Abbildung 5.19: Beispielanwendung des Kompatibilitätstypen „logische Schnittmenge“ für das Attribut „Kommunikationsschnittstelle“ der Bausteine A1 und B3 bzw. A2 und B3.

Entsprechend der mathematischen Definition einer logischen Schnittmenge werden die Listen bei der Prüfung auf gleiche Einträge hin untersucht. Für die Anwendung wird das in Kapitel 4.3.4 bereits angeklungene Beispiel der Kommunikationsschnittstelle herangezogen (Abbildung 5.19). Finden sich unter den zu vergleichenden Attributwerte keine gleichen Einträge, gelten die untersuchten Elemente als inkompatibel.

- **Vereinigungsmenge:**

Mit diesem Kompatibilitätstypen wird die logische Vereinigungsmenge zweier Attributwerte ermittelt. Beispielsweise werden unterschiedliche Verfahren zur Befestigung von Bauteilen auf einem Substrat in der Frühen Phase der Entwicklung zunächst gemeinschaftlich erfasst, anstelle diese als Ausschlusskriterien zu verwenden (Abbildung 5.20).

Kompatibilitätstyp Vereinigungsmenge

Attribut Substratanbindung	Baustein A	Baustein B	Ergebnis Vereinigungsmenge
	THT	SMD	THT SMD

Abbildung 5.20: Beispielanwendung des Kompatibilitätstypen „Vereinigungsmenge“ für das Attribut „Substratanbindung“ der Bausteine A und B.

Im Gegensatz zu den anderen Kompatibilitätstypen können zwei Elemente auf Basis der Vereinigungsmenge in keinem Fall als inkompatibel ausgewiesen werden.

- **SensorTypAnzahl:**

Hinter dem Kompatibilitätstypen SensorTypAnzahl verbirgt sich ein spezielles Auswerteverfahren für das Attribut Sensorinterface. Hierbei wird der Tatsache Rechnung getragen, dass die mittels Sensoren erfassten Daten im informationsverarbeitenden Teil des Steuergeräts, der sogenannten Steuergerätelektronik prozessiert werden. Damit die Sensorsignale die Logikbauteile erreichen können müssen diese über geeignete Schnittstellen verbunden sein. Ein Logikbauteil wird in diesem Zusammenhang dabei als Quelle betrachtet, die eine bestimmte Anzahl an verschiedener Sensorschnittstellentypen zur Verfügung stellt. Diese wird im Attribut Sensorschnittstelle gemäß nachfolgendem Schema erfasst:

Bild z.B. 2xSPI, 2x HALL, 2xSENT etc.

Die Sensoren hingegen stellen die Verbraucher der Schnittstellen dar. Auf deren Seite sind die benötigten Schnittstellentypen und Anzahlen analog zu den Logikbauteilen unter dem Attribut Sensorschnittstelle erfasst.

z.B. 5xANALOG, 2x HALL etc.

Im Zuge der Kompatibilitätsprüfung werden, die durch die Quelle verfügbaren Schnittstellen und die, durch die Verbraucher benötigten Schnittstellen betrachtet. Hierbei wird die benötigte Schnittstellenanzahl eines Verbrauchers in Abhängigkeit des jeweiligen Schnittstellentyps von der Menge, der auf Seiten des Logikbauteils verfügbaren subtrahiert. Dieser Prozess wird für jeden im Betrachtungsraum<sup>10</sup> vorhandenen Verbraucher sequenziell durchgeführt. Sind alle Verbraucher berücksichtigt wird anhand der übrigen Schnittstellen auf Seiten der Logikbauteile über die Kompatibilität der beteiligten Elemente entschieden. Fordert ein Verbraucher einen seitens der Quelle nicht angebotenen Schnittstellentyp, wird dieser an der Quelle eingefügt und mit einer entsprechend negativen Anzahl verfügbarer Schnittstellen gekennzeichnet. Eine Kombination zählt dann als kompatibel, wenn die Anzahl jedes Schnittstellentyps der Quelle  $\geq 0$  ist (Abbildung 5.21).

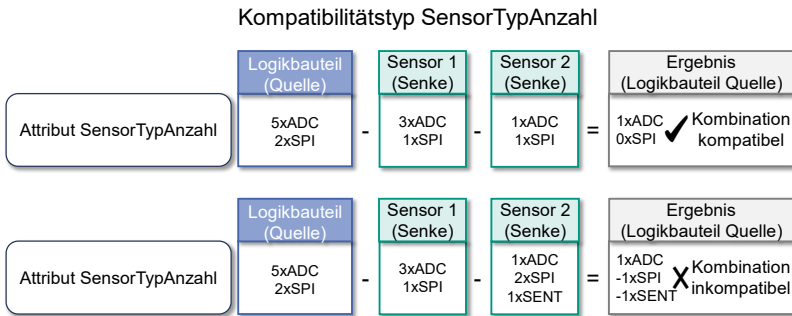


Abbildung 5.21: Beispielanwendung des Kompatibilitätstypen „SensorTypAnzahl“ für das Attribut „SensorTypAnzahl“ unterschiedlicher Kombinationen für Quelle und Senke.

### Prüfung der internen Kompatibilität innerhalb eines FSA

Die Prüfung der internen Kompatibilität stellt die unmittelbare Voraussetzung für die Darstellung, des über den Baukasten realisierbaren Lösungsraums dar. Hierzu werden Kombinationen integraler Bausteine anhand der, innerhalb eines FSA vorgegebenen Wirkkette zwischen den zugehörigen Modulen, evaluiert. Dieser

<sup>10</sup> Der Betrachtungsraum ist jeweils durch einen Funktionsstrukturausschnitt repräsentiert (vgl. Kapitel 5.2.1)



Vorgang wird im Zuge der Bildung der modularen Bausteinvarianten des Baukastens für jeden in der Struktur berücksichtigten FSA durchgeführt. Aus Kapitel 4.2.2 geht hervor, dass die modularen Bausteinvarianten der technischen Repräsentation des Funktionsumfang ihres zugrundeliegenden FSA entsprechen. Erweisen sich die kombinierten integralen Bausteinvarianten als kompatibel, wird die resultierende modulare Bausteinvariante auf Basis der Key-Parameter des betrachteten Funktionsstrukturausschnittes mit Werten versehen. Damit die Kompatibilität der integralen Bausteinvarianten innerhalb eines FSA geprüft werden kann, müssen zunächst sämtliche Kombinationen aufgestellt werden. Die Gesamtanzahl, der auf Kompatibilität zu untersuchenden Kombinationen, ergibt sich als Produkt der Anzahl integraler Bausteinvarianten entsprechend der einzelnen Module innerhalb eines FSA. Für einen FSA mit drei Modulen die fünf, vier und drei integrale Bausteinvarianten besitzen, existieren demnach 60 Bausteinvarianten.

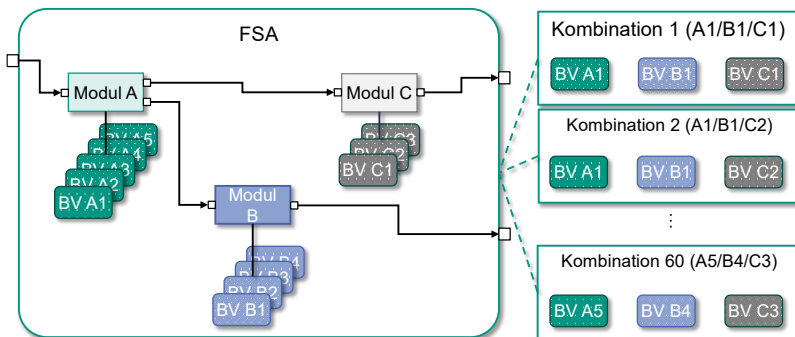


Abbildung 5.22: Bestimmung aller Kombinationen eines FSA

Jede der theoretisch ermittelten Bausteinvarianten stellt eine potenzielle modulare Bausteinvariante dar und wird im weiteren Verlauf als eigenständiges Element behandelt. Innerhalb einer Kombination werden die Module durch die spezifische für die Kombination ausgewählte Bausteinvariante ersetzt (Abbildung 5.22). Die Kompatibilität einer Kombination und damit verbunden, die Entscheidung zur Berücksichtigung der modularen Bausteinvariante im Baukasten, wird für jeden innerhalb der Kombination enthaltenen Key-Parameter geprüft. Dies geschieht unter Verwendung der in Kapitel 5.2.1 hergeleiteten Kompatibilitätstypen. Der

Ablauf der Kompatibilitätsprüfung für einen FSA wird am nachfolgenden Beispiel illustriert (Abbildung 5.23).

- Schritt 1: Ausgehend von den Schnittstellen der vorliegenden Kombination<sup>11</sup> wird zunächst eine benachbarte direkt, über die Wirkkette verbundene integrale Bausteinvariante ermittelt (BV1). Um mögliche Fehler in der Verbindung mittels der Key-Parameter aufzudecken, werden die Schnittstellen zwischen den Elementen zunächst anhand ihrer Attributtypen plausibilisiert.
- Schritt 2: Im Anschluss wird eine weitere, mit der ausgewählten Bausteinvariante in Beziehung stehende BV in die Betrachtung mit einbezogen (BV 2). Zwischen den Bausteinen werden die verbundenen Key-Parameter der Reihe nach entsprechend des zugrunde gelegten Kompatibilitätstypen abgearbeitet.
- Schritt 3a: Für den Fall, dass eine Key-Parameterpaarung entsprechend ihrer Attributwerte kompatibel ist, werden die initialen Attributwerte der Bausteinvarianten durch den ermittelten Kompatibilitätswert überschrieben.<sup>12</sup> Da die integralen Bausteinvarianten eines Moduls aufgrund der gewählten Kombinatorik stets in mehreren Kombinationen auf Kompatibilität untersucht werden, gelten die überschriebenen Werte jeweils ausschließlich für die gerade betrachtete Kombination.
- Schritt 3b: Sobald eine der untersuchten Key-Parameterpaarungen eine Inkompatibilität der beteiligten Bausteinvarianten aufzeigt, wird der Prozess der Kompatibilitätsprüfung direkt abgebrochen. Die Kombination und damit die potenzielle modulare Bausteinvariante werden dementsprechend verworfen und nicht für die weitere Betrachtung im Baukasten berücksichtigt. Schritt 1 wird mit einer neuen Kombination durchgeführt.

---

<sup>11</sup> An dieser Stelle wird bewusst von den Schnittstellen der Kombination bzw. der potenziellen modularen BV gesprochen, da der Begriff FSA nur im funktionalen Kontext und unter Verwendung von Modulen zu gebrauchen ist. Durch das Ersetzen der generischen Module mittels spezifischer Bausteinvarianten befinden wir uns allerdings im technischen Kontext.

<sup>12</sup> Im Falle des Kompatibilitätstypen `SensorTypAnzahl` wird, wie in Kapitel 5.2.1 beschrieben lediglich die Seite des Logikbauteils überschrieben.

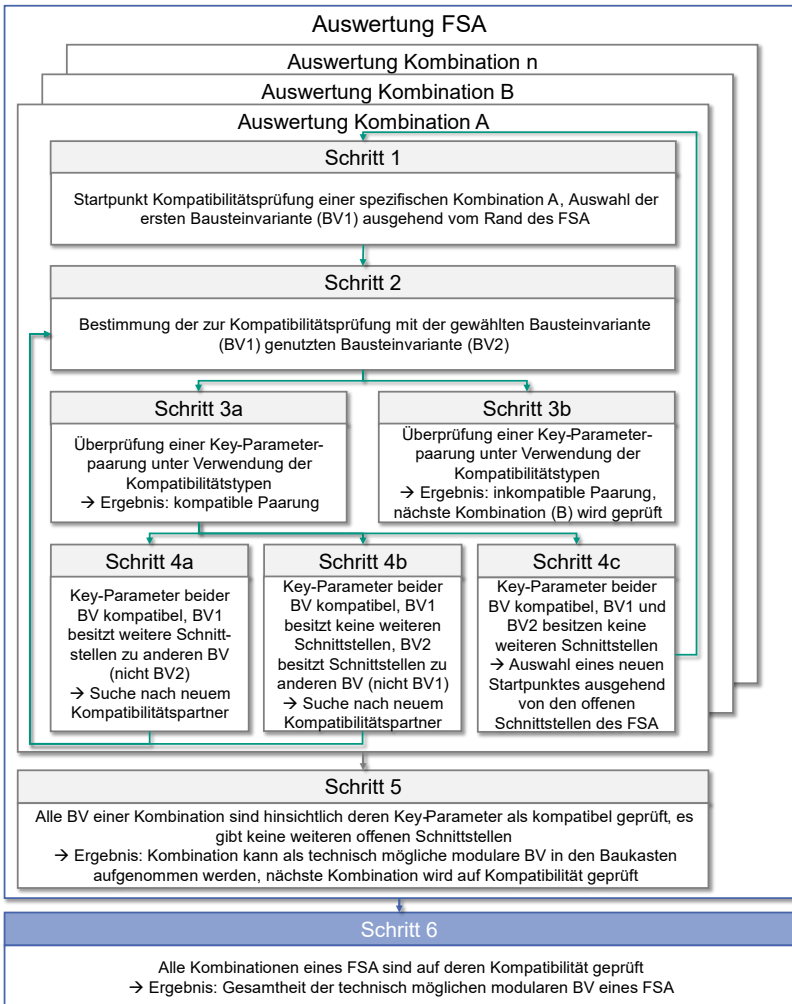


Abbildung 5.23: Ablauf der Kompatibilitätsprüfung für einen FSA in sechs Schritten

- Schritt 4a: Erweisen sich hingegen alle Schnittstellen zwischen zwei Bausteinvarianten als kompatibel, so wird ausgehend von der ersten Bausteinvariante (BV 1) geprüft, ob diese weiteren Schnittstellen zu anderen BVs (z.B. BV 3) besitzt. Ist dies der Fall werden die Schritte 1-4 wiederholt zwischen der neuen BV-Paarung (BV 1 – BV 3) untersucht.
- Schritt 4b: Existieren keine weiteren Schnittstellen für BV 1 die mit einer anderen Bausteinvariante verbunden sind wird BV 2 auf weitere Schnittstellen geprüft.
- Schritt 4c: Besitzt auch BV 2 keine weiteren bisher unbehandelten Beziehungen zu anderen Bausteinvarianten, wird erneut von den Schnittstellen der Kombination nach weiteren Bausteinvarianten gesucht.
- Schritt 5: Sind alle Bausteinvarianten abgearbeitet und demnach keine Inkompatibilitäten entdeckt worden, gilt die betrachtete potenzielle modulare Bausteinvariante als technisch umsetzbar. Damit diese für die weitere Betrachtung im Baukasten herangezogen werden kann, müssen abschließend deren Schnittstellen parametrisiert werden. Hierzu werden für jeden Key-Parameter einzeln die Werte der direkt mit den Schnittstellen der Kombination verbundenen integralen Bausteinvarianten übernommen.
- Schritt 6: Die Schritte 1-5 werden anschließend für jede, auf Basis eines FSA erstellte Kombination integraler Bausteinvarianten durchgeführt. Ergebnis der Kompatibilitätsprüfung innerhalb eines FSA ist die Gesamtheit, der seinem Funktionsumfang entsprechenden technisch möglichen modularen Bausteinvarianten.

Für die Kompatibilitätsprüfung mittels Intervallverrundung ergibt sich eine Abweichung zur beschriebenen Vorgehensweise. Entsprechend des Verhaltens mechatronischer Funktionsträger besteht zwischen den Flussattributen z.B. Strömen und Spannungen am Ausgang eines Moduls, in der Regel eine direkte Abhängigkeit zu entsprechenden Eingangsgrößen. Zur Veranschaulichung des Umgangs mit diesem Umstand wird ein Beispiel in Anlehnung an die Arbeit von Zanger (2018)<sup>13</sup> herangezogen. Das Beispiel zeigt eine mittels interner Kompatibilitätsprüfung auszuwertende Kombination integraler Bausteinvarianten und deren Beziehungen anhand festgelegter Key-Parameter. Der abgebildete Funktionsumfang entspricht dem eines vereinfachten elektrischen Kleinantriebs. Für die drei Module des betrachteten FSA: eine Motorregelung, einen DC-Motor und ein Getriebe, wird eine spezifische Kombination integraler Bausteinvarianten betrachtet (Abbildung 5.24).

---

<sup>13</sup> Co-Betreute Abschlussarbeit

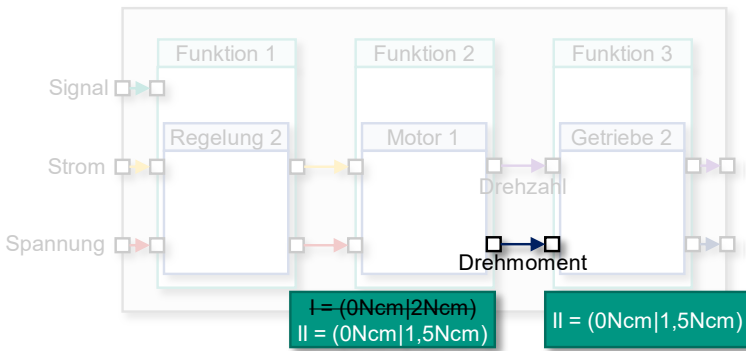


Abbildung 5.24: Anwendung der Intervallverrundung nach Zanger (2018). Beispiel anhand des Intervalls eines Drehmomentes in Bezug auf die Kompatibilität eines Motors und eines Getriebes.

Richten wir unsere Aufmerksamkeit zunächst die Ausgangsgröße Drehmoment des DC-Motors. Diese dient gleichzeitig als Eingangsgröße des Getriebes. Die Bausteinvariante des Motors weist ein maximal mögliches Drehmoment von  $2\text{Ncm}$  auf. Die für das Getriebe gewählte Bausteinvariante kann allerdings nur Drehmomente bis  $1,5\text{Ncm}$  aufnehmen. Entsprechend der Intervallverrundung wird das Kompatibilitätsintervall der betrachteten Bausteinvarianten von  $0$  bis  $1,5\text{Ncm}$  festgesetzt. Da das vom DC-Motor erzeugte Drehmoment ebenfalls von der Eingangsspannung und dem Eingangsstrom abhängig ist müssen die entsprechenden Kompatibilitätsintervalle ebenfalls angepasst werden (Abbildung 5.25). Die Beschreibung der Abhängigkeiten zwischen Ein- und Ausgangsgrößen der einzelnen integralen Bausteinvarianten müssen als Voraussetzung für die interne Kompatibilitätsberechnung vorliegen.<sup>14</sup> Der Vorgang der Intervallverrundung setzt sich auf alle davon abhängigen Größen fort. Erfolgt die Kompatibilitätsprüfung in Form eines Algorithmus entspricht die Anpassung der Kompatibilitätsintervalle einem rekursiven Aufruf der Intervallverrundung. Dieser wird so lange für alle beteiligten Key-Parameter durchgeführt, bis sich keines der betrachteten Intervalle mehr ändert, zwei der betrachteten Größen inkompatibel sind, oder die Schnittstellen der Kombination selbst erreicht sind. Wurde die Intervallverrundung

<sup>14</sup> Dies kann beispielsweise analytisch oder in Form toolgestützter Verhaltensmodelle geschehen

flächendeckend durchgeführt, sind alle Einflüsse der betrachteten Attribute bis ins Systeminnere berücksichtigt worden.

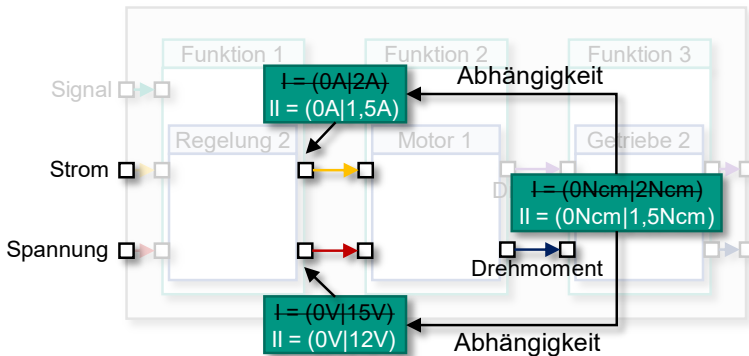


Abbildung 5.25: Anpassung von Intervallen bei der Intervallverrundung nach Zanger (2018). Da das Drehmoment vom Eingangsstrom abhängig ist, muss dessen Intervall analog zum Drehmomentintervall angepasst werden.

Sind alle Kombinationen eines FSA betrachtet worden ist die interne Kompatibilitätsprüfung für den entsprechenden Funktionsumfang abgeschlossen. Die ermittelten modularen Bausteinvarianten werden, wie in Kapitel 4.2.2 beschrieben, auf der nächsthöheren Abstraktionsebene zusammen mit den dort verorteten integralen Bausteinvarianten wiederum bei der Kombinatorik modularer Bausteinvarianten berücksichtigt. Sich daraus ergebende Rahmenbedingungen werden im folgenden Unterkapitel diskutiert.

### Berechnung der internen Kompatibilität entlang des Strukturbaumes

Im Zuge der vorliegenden Arbeit werden aus dem Baukasten abgeleitete mechatronische Steuergeräte als Kombinationen diverser Bausteinvarianten entsprechend der zugrundeliegenden Modulstruktur betrachtet. Entsprechend der funktionalen Betrachtung erfüllt eine Kombination den Funktionsumfang eines FSA. Dessen Funktionsumfang entspricht dabei dem eines übergeordneten Funktionsclusters. Innerhalb eines FSA befinden sich wiederum Funktionscluster, die auf einer tieferen Ebene durch einen oder mehrere FSAs beschrieben werden können. Dieser Prozess setzt sich bis zum Erreichen der maximalen Strukturierungstiefe fort (vgl. Kapitel 4.2.2). Entsprechend der funktionalen

Sichtweise auf Basis von Funktionsclustern/Modulen und FSAs, gestalten sich die Vorgänge auf technischer Ebene analog. Die Parametrierung einer modularen Bausteinvariante ergibt sich entsprechend des Bottom-Up Prinzips aus der Kombination untergeordneter integraler Bausteinvarianten. Damit auf Ebene des, aus dem Baukasten abzuleitenden Produktes alle als kompatibel ausgewiesenen modularen Bausteinvarianten berücksichtigt werden können muss der Strukturbaum von unten nach oben befüllt werden. Definitionsgemäß existieren immer dann ausschließlich integrale Bausteinvarianten, wenn beim Anlegen der Strukturierungskriterien an ein Funktionscluster eine Verletzung der Kriterien vorliegt. Aufgrund der Tatsache, dass für jedes Funktionscluster auch parallele FSAs entstehen können, ist die resultierende Baumstruktur mitunter stark verzweigt. Zur Berechnung der Parameter der modularen Bausteinvarianten werden müssen die einzelnen Abstraktionsebenen nacheinander betrachtet werden.

### Theoretisches Vorgehen zur externen Kompatibilitätsberechnung

Die Ausführungen zur internen Kompatibilitätsbetrachtung zeigen, dass hinsichtlich der Prüfung innerhalb eines FSA absolute Aussagen zur Kompatibilität ausreichend sind. Zur Betrachtung der externen Kompatibilität werden derartige Ansätze als nicht zielführend erachtet. Vielmehr gilt es die einzelnen Nuancen in der Befriedigung von Kundenanforderungen, in Form sogenannter Erfüllungsgrade zu berücksichtigen. Entsprechend der in Kapitel 2.4.3 präsentierten Grundlagen eignen sich hierzu, auf der Fuzzy-Logik basierende Ansätze zur Kompatibilitätsprüfung besonders. Im Rahmen der vorliegenden Arbeit wurde daher der von Ishii et al. entwickelte Ansatz der Design Compatibility Analysis (vgl. Kapitel 2.4.3) als Basis der externen Kompatibilitätsprüfung gewählt.

Resultat der externen Kompatibilitätsprüfung ist wie bereits diskutiert, die Ausgabe der bewerteten Gesamtkompatibilität jeder im Baukasten enthaltenen Kombination in Bezug auf einen ausgewählten Satz quantifizierter Kundenanforderungen. Entsprechend des von Ishii et al. definierten Match Index ergibt sich die Gesamtkompatibilität einer Kombination als gewichtete Summe einzelner Match Koeffizienten  $M(s)$  als Zahl im Wertebereich zwischen 0 und 1:

$$\text{Gesamtkompatibilität} = MI = \sum_k utility(s) \cdot M(s), \quad s \in K$$

Ein Match Koeffizient  $M(s)$  entspricht dabei dem Kompatibilitätswert einer im Baukasten enthaltenen Kombination, in Bezug auf einen einzelnen Key-Parameter. In Absprache mit den Experten wurde von einer unterschiedlichen Gewichtung der Match Koeffizienten zunächst abgesehen. Die Bestimmung der einzelnen Match

Koeffizienten wird im Gegensatz zu der von Ishii et al. durchgeführten Expertenbewertung, unter Verwendung der in Kapitel 5.2.1 vorgestellten Kompatibilitätstypen durchgeführt. Genauer, für die externe Kompatibilitätsprüfung werden lediglich die Typen Intervallverrundung und logische Schnittmenge zur Berechnung herangezogen. Auch an dieser Stelle wird zwischen den einzelnen Attributtypen unterschieden:

Für Flussattribute, sowie quantifizierbare stationäre Attribute wird zunächst eine Intervallverrundung durchgeführt. Das resultierende Intervall wird anschließend zu dem durch die Kundenanforderungen vorgegebenen Intervall in Relation gesetzt. Da eine Übererfüllung der Kundenanforderungen aus wirtschaftlicher Sicht meist nicht erwünscht ist, soll diese bei der Berechnung des Match Koeffizienten ebenfalls Berücksichtigung finden. Für die Berechnung werden daher drei Fälle unterschieden:

- Erster Fall: Ergibt sich das Verhältnis des verrundeten Intervalls ( $I_V$ ) zum Kundenintervall ( $I_{Ku}$ ) als  $\frac{I_V}{I_{Ku}} < 1$ , so entspricht der Match Koeffizient  $M(s)$  exakt dem berechneten Verhältnis  $M(s) = \frac{I_V}{I_{Ku}}$ .
- Zweiter Fall: Gilt:  $\frac{I_V}{I_{Ku}} = 1$  und ist das Verhältnis des Kundenintervalls zum Intervall der von Baukasten bereitgestellten Kombination ( $I_{Kombi}$ ) gleichzeitig  $\frac{I_{Ku}}{I_{Kombi}} = 1$ , so deckt die Kombination exakt den gewünschten Rahmen der Kundenanforderung ab. Dementsprechend ergibt sich der Match Koeffizient zu  $M(s) = 1$
- Dritter Fall: Gilt:  $\frac{I_V}{I_{Ku}} = 1$  und gleichzeitig  $\frac{I_{Ku}}{I_{Kombi}} < 1$ , so liegt eine Übererfüllung der Kundenanforderung vor. Der resultierende Match Koeffizient ergibt sich demnach als  $M(s) = 1 - \frac{I_{Ku}}{I_{Kombi}}$

Das Ergebnis eines Match Koeffizienten ergibt sich entsprechend der Fuzzy-Logik als Zahlenwert zwischen 0 und 1.

Für nicht-quantifizierbare stationäre Attribute, sowie die optionalen Module wird die logische Schnittmenge zwischen der Kundenparametrisierung des betrachteten Key-Parameters und den entsprechenden Werten der Kombination gebildet. Die Schnittmenge wird anschließend analog zur Intervallverrundung mit der Kundenvorgabe verglichen. Der resultierende Match Index inklusive Betrachtung der Übererfüllung ergibt sich ebenfalls analog zum Fall quantifizierter Key-Parameter.



Aus den betrachteten Entwicklungsprozessen mechatronischer Steuergeräte (vgl. Kapitel 4.1) zeigt sich, dass die Kundenanforderungen<sup>15</sup> einen gewissen Härtegrad aufweisen. Attribute wie die Spannungsklasse des vorgegebenen Bordnetzes sind beispielsweise nicht verhandelbar, während sich eine Abweichung der resultierenden Phasenströme mit den unternehmensinternen Motorenherstellern diskutieren lässt. Um diesem Umstand Rechnung zu tragen, wird die Betrachtung von sogenannten harten Anforderungen im Bewertungsprozess berücksichtigt. Deren Erfüllung wird dementsprechend zunächst mittels absoluter, also binärer Logik abgeprüft. Wird eine harte Anforderung durch die Parameter einer Kombination aus dem Baukasten nicht erfüllt, wird der Maximalwert der zu erreichenden externen Kompatibilität der Kombination auf 0,5 heruntergestuft. Dieser Wert entspricht gerade dem von Ishii et al. definierten Grenzwert, für die potenzielle Eignung einer bewerteten Systemkonfiguration. Anschließend wird der Kompatibilitätswert in Abhängigkeit des zugrundeliegenden Attributtypen wiederum durch intervallverrundung oder die Bildung der logischen Schnittmenge ermittelt.

## 5.2.2 Identifikation potenzieller Lücken des Baukastens

Bezüglich der Identifikation von etwaigen Lücken des Baukastens bzw. Entwicklungspotenzialen gilt es funktionale von technologischen Potenzialen zu unterscheiden:

### **Identifikation von Entwicklungspotenzialen hinsichtlich Funktionalität:**

Wie bereits in Kapitel 4.3.1 dargelegt, können in Abhängigkeit des gewählten Baukastenarchitekturtypen (singulär, hybrid, autark) mehrere Baukastenmodelle für ein abzubildendes Produktportfolio vorliegen (vgl. Kapitel 4.3.1). Die verschiedenen Baukastenmodelle weisen durch ihre unterschiedlichen Modulstrukturen einen unterschiedlichen funktionalen Umfang auf. Liegen mehrere Baukastenmodelle zur Anwendung bereit, muss vor der Prüfung der externen Kompatibilität das geeignete Baukastenmodell ausgewählt werden. Dieser Auswahlvorgang wird im Rahmen der vorliegenden Arbeit als funktionale Prüfung der Kundenanforderungen bezeichnet. Ist das passende Baukastenmodell ausgewählt können über die Baukasten-anwendung ebenfalls Entwicklungspotenziale im Bereich, der vom Baukasten erfassten Funktionen identifiziert werden. Dazu werden die vom Kunden adressierten Schnittstellen des Gesamtsystems mit den im vorliegenden

---

<sup>15</sup> Hierbei sind wiederum fallspezifisch interne von externen Kunden zu unterscheiden

Baukastenmodell realisierten Key-Parametern der ECU verglichen. Werden vom Kunden neue, bisher nicht enthaltene Schnittstellen adressiert, kann diese Information den Entwicklungsteams zur Verfügung gestellt werden. An dieser Stelle sei erneut auf die große Bedeutung einer kontinuierlichen Durchführung der iPeM Basis-Aktivität „Wissen managen“ hingewiesen. Auf diese Weise kann nach einer umfassenden Tragweitenanalyse der Prozess für eine Neuentwicklung angestoßen werden. Wird ein Produkt mit der neuen Funktionalität entwickelt, kann diese durch erneutes Durchführen der Strukturierungsmethode in den Baukasten aufgenommen werden<sup>16</sup>.

### **Identifikation technologischer Neuentwicklungspotenziale<sup>17</sup>:**

Wurde das entsprechende Baukastenmodell als Grundlage der Baukastenanwendung gewählt und wurden keine funktionalen Abweichungen identifiziert, können die Ergebnisse der externen Kompatibilitätsprüfung analysiert werden. Weisen die vom Baukasten angebotenen Kombinationen nicht die gewünschte Kompatibilität bezüglich der Kundenanforderungen auf, kann anhand einzelner Kombinationen nachvollzogen werden welche Bausteinvarianten maßgeblich an der Abweichung beteiligt sind. Dies lässt sich über einen Vergleich der, an den Schnittstellen der Kombination errechneten Key-Parameterwerte mit den entsprechenden Werten, der zur Kombination herangezogenen Bausteinvarianten bewerkstelligen. Auf diese Weise werden alle, an der Festlegung eines Kombinationsparameters beteiligten Bausteinvarianten, mit ihren Grenzen nebeneinander aufgeführt. Da der Wert des Key-Parameters der Kombination dem, des vermeintlich schwächsten Glied der Kette entspricht, kann dieses mühelos identifiziert werden. Ebenfalls wird ersichtlich, ob nur eine oder gleich mehrere Bausteinvarianten nicht die gewünschten Eigenschaften besitzen. Gleichzeitig kann aus der Abweichung der Gesamtkombination zu den Kundenanforderungen die Erkenntnis gewonnen werden, welche Eigenschaften eine potenziell neu zu entwickelnde Bausteinvariante besitzen muss. Diese Information kann anschließend in der F&E-Phase der Technologieentwicklung als wichtige Eingangsgröße genutzt werden. Über die Mechanismen zur Aufdeckung von etwaigen Lücken des Baukastens können auch die letzten der Stakeholderbedarfe

---

<sup>16</sup> Dies entspricht wiederum der SPALTEN Aktivität „Entscheiden und Umsetzen“

<sup>17</sup> Hierbei handelt es sich entsprechend der PGE um Gestalt- oder Prinzipvariationen

abgedeckt werden. Die theoretischen Grundlagen der Baukastenanwendung sind somit dargelegt.

### **5.3 Implementierung der Baukastenanwendung mittels MBSE**

Im Anschluss an die theoretischen Grundlagen der Baukastenanwendung wird im Folgenden vorgestellt, wie diese entsprechend den Rahmenbedingungen der vorliegenden Untersuchungsumgebung implementiert wurde. In diesem Zusammenhang werden die noch ausstehenden Aspekte der zweiten Teilforschungsfrage beantwortet:

*FF2.2: Wie werden die Baukastenstrukturierung und –anwendung mechatronischer Steuergeräte mittels MBSE-Methoden entsprechend integriert?*

Die Vorstellung der Implementierung erfolgt anhand der folgenden Schwerpunkte:

Zunächst wird die zur Implementierung genutzte Toolkette umrissen. Anschließend wird das zur Bedienung des Baukastens zentrale Element, die Benutzeroberfläche vorgestellt. Zum Abschluss des Kapitels wird auf die Umsetzung der in Kapitel 5.1.2 ermittelten Use-Cases eingegangen.

#### **5.3.1 Aufbau einer Toolkette zur automatisierten Baukastenanwendung**

Grundlage der Baukastenimplementierung stellt das in Kapitel 4.3 vorgestellte initiale Baukastenmodell in IBM Rhapsody dar. Dieses enthält bereits alle zur Baukastenanwendung benötigten strukturellen Elemente und stellt durch die integralen Bausteinvarianten die Datenbasis der Kompatibilitätsuntersuchungen dar. In Zusammenhang mit der Baukastenstrukturierung wurde bereits das Interface zwischen dem Anforderungskatalog in IBM DOORS und dem Baukastenmodell in Rhapsody beleuchtet. Aufgrund der großen, im Zuge der Kompatibilitätsprüfung entstehenden Datenmengen wurden die Inhalte des Rhapsodymodells in eine, auf SQL-Basis operierende, relationale Datenbankstruktur überführt. Die Kompatibilitätsprüfungen selbst wurden mittels Java-Code automatisiert durchgeführt. Die zur Prüfung innerhalb eines FSA zu verwendenden Kompatibilitätstypen wurden im Rhapsody-Modell durch entsprechende Stereotypen an den Modulverbindungen hinterlegt. Die Ergebnisse der Kompatibilitätsprüfung werden wiederum in der Datenbank gespeichert.

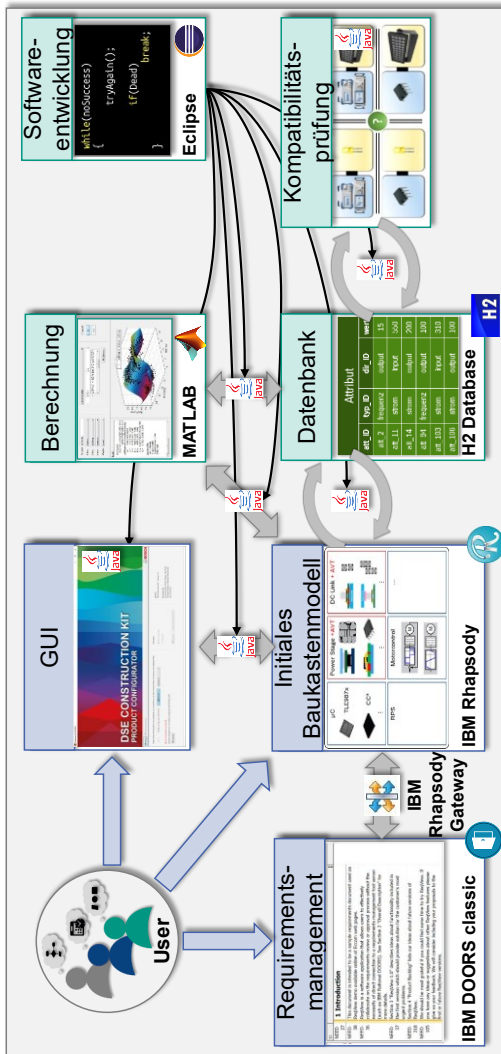


Abbildung 5.26: Überblick der entwickelten Toolkette zur Baukastenanwendung

Das vollständige Modell des Baukastens inklusive der modularen Bausteinvarianten und Produktkombinationen liegt demnach nur in der Datenbank vor. Zentrales Element der Toolkette zur Baukastenanwendung ist die Benutzeroberfläche. Von hier aus können sämtliche Interaktionen mit dem Baukastenmodell z.B. die Überführung von Änderungen am Rhapsodymodell in die Datenbank, die Ermittlung des Lösungsraums des Baukastens mittels interner Kompatibilitätsbetrachtung, oder die Use-Case spezifische Bedienung des Baukastenmodells, gesteuert werden (Abbildung 5.26). Über die Mittlerfunktion von Java können jederzeit weitere, den Experten bekannte Tools entsprechend dem föderierten Ansatz eingebunden werden.

### 5.3.2 Aufbau der zentralen Benutzeroberfläche

Die zur Steuerung der Baukastenanwendung verwendete Benutzeroberfläche ist in zwei Abschnitte gegliedert (Abbildung 5.27).

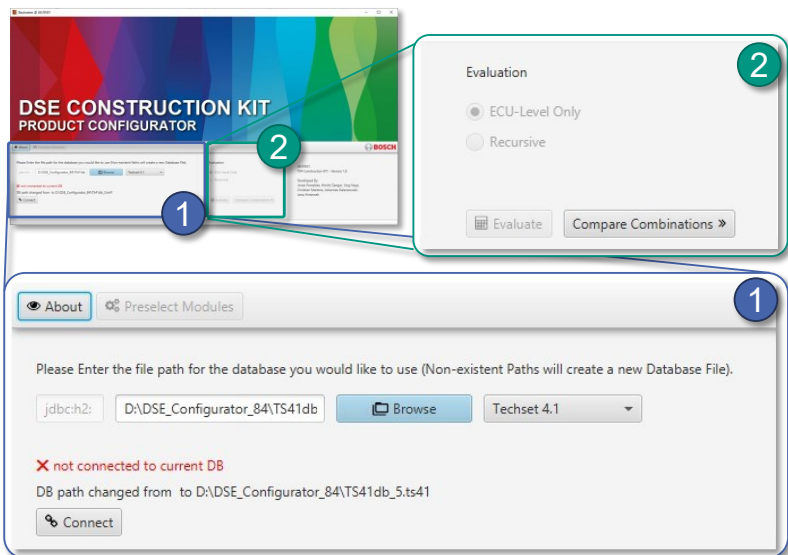


Abbildung 5.27: Aufbau der zentralen Benutzeroberfläche der Baukastenanwendung.

Im ersten Abschnitt (1) wird der Nutzer dazu aufgefordert den Pfad der hinterlegten Ergebnisdatenbank auszuwählen. Über die Schaltfläche „Connect“ kann er sich anschließend mit der Datenbank verbinden. Im vorliegenden Fall, der hybriden Baukastenarchitektur, wird jeder der vier Technologiesätze durch ein eigenes Baukastenmodell und damit verbunden eine eigene Ergebnisdatenbank realisiert. Der Nutzer trifft über die Auswahl der Ergebnisdatenbank die Entscheidung, welches Baukastenmodell er bei der Baukastenanwendung berücksichtigen möchte. Existiert noch keine Ergebnisdatenbank für den gewünschten Technologiesatz, also wurde die interne Kompatibilitätsprüfung noch nicht durchgeführt, kann der Nutzer eine neue Datenbank erzeugen. In diesem Fall wird das zugrundeliegende initiale Baukastenmodell aus Rhapsody über den Connect-Button in die Datenbank überführt. Das initiale Baukastenmodell muss dazu im Hintergrund geöffnet sein (siehe Statusfenster). Ist die Verbindung zu einer Ergebnisdatenbank hergestellt (siehe zweite Zeile Statusfenster), wird Abschnitt zwei der Benutzeroberfläche freigeschaltet. Hier kann der Nutzer in Abhängigkeit seiner Rolle die interne (mittels „Calculate“) und/oder die externe (via „Compare Combinations“) Kompatibilitätsprüfung starten.

Die Unterscheidung zwischen den Rollen der Stakeholder ist in der Toolkette folgendermaßen umgesetzt:

Da die Entwickler aktiv zur Erweiterung des Baukastens beitragen, haben sie in jedem Falle Zugriff auf das initiale Baukastenmodell in Rhapsody (Abbildung 5.28). Dieser Umstand wird dazu genutzt ihnen die volle Funktionsfähigkeit der Toolkette und damit die Durchführung der internen Kompatibilitätsprüfung zu ermöglichen. Hierfür wurde die Benutzeroberfläche insofern angepasst, dass diese automatisch erkennt, ob das initiale Baukastenmodell in Rhapsody im Hintergrund geöffnet ist. Ist dies der Fall wird in Abschnitt (2) die Neuberechnung des Baukastens freigegeben.

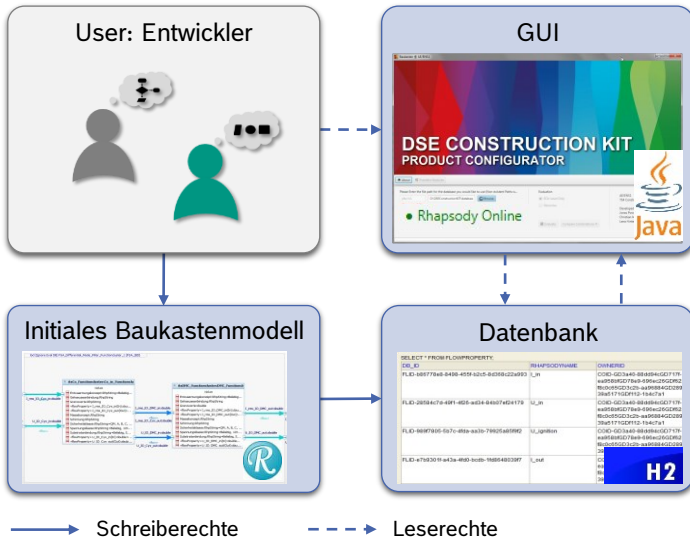


Abbildung 5.28: Toolkettenzugriff für System- und Modulentwickler. Diese ist durch den direkten Zugriff auf das initiale Baukastenmodell charakterisiert.

Bei der internen Kompatibilitätsprüfung kann sich der Entwickler anschließend entscheiden, den gesamten Lösungsbaum des Baukastens, oder nur die oberste Ebene zu berechnen. Dies ist davon abhängig, ob tiefgreifende Änderungen am initialen Baukastenmodell vorgenommen wurden. Ergebnis der internen Kompatibilitätsprüfung ist das vollständige Baukastenmodell in Form der ergänzten Ergebnisdatenbank.

Die Stakeholdergruppe der Manager wird keine aktive Veränderung am initialen Baukastenmodell vornehmen. In der Regel ist eine Nutzung von Rhapsody für diese Stakeholdergruppe nicht vorgesehen, was es dem Unternehmen ermöglicht, die an den Toolhersteller zu entrichtenden Lizenzkosten zu minimieren. Der Zugang zum Lösungsraum des Baukastens wird dementsprechend über das vollständige Baukastenmodell, die Ergebnisdatenbank realisiert.

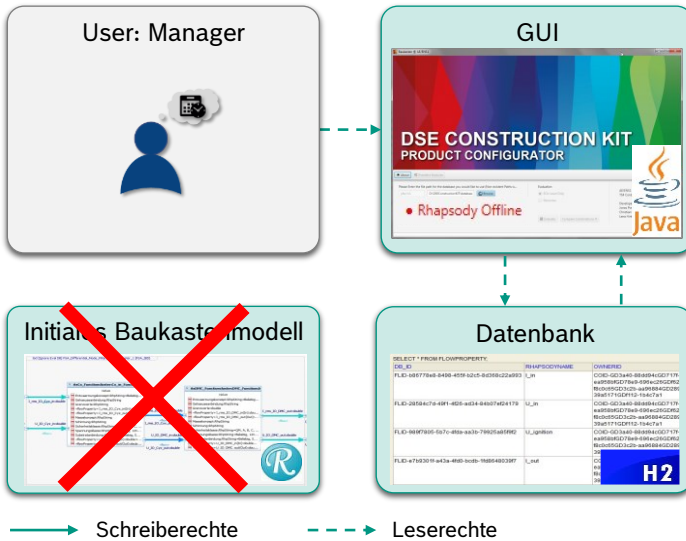


Abbildung 5.29: Toolkettenzugriff für das Management. An dieser Stelle wird eine Nutzung der Anwendung ohne Zugriff auf das initiale Baukastenmodell realisiert.

Demnach stellt ein Entwickler nach Beendigung der internen Kompatibilitätsprüfung die neueste Version der Ergebnisdatenbank für alle Nutzergruppen zur Verfügung. Diese kann in Abschnitt (1) der Benutzeroberfläche ausgewählt und verbunden werden. Da das Rhapsodymodell im Hintergrund nicht läuft, entfällt die Möglichkeit zur internen Kompatibilitätsprüfung automatisch (Abbildung 5.29). Die externe Kompatibilitätsprüfung kann von beiden Stakeholdergruppen nur dann gestartet werden, wenn die Verbindung zu einem vollständigen Baukastenmodell hergestellt ist.

### 5.3.3 Realisierung der Stakeholder Use-Cases

Im Folgenden werden die Interaktionsmöglichkeiten des Nutzers entsprechend der in Kapitel 5.1.2 ermittelten Use-Cases vorgestellt.

#### Auswahl einer Konfiguration aus dem Baukasten

Die Möglichkeit entsprechend einer Kundenanfrage passende Produktkombinationen aus dem Baukasten auswählen zu können, stellt den wohl



bedeutendsten Use-Case der Baukastenanwendung dar. Den größten Nutzen an den bereitgestellten Informationen hat dabei der Systementwickler. Dieser hat durch das Baukastenmodell Zugriff auf eine große Wissensbasis alternativer Lösungen.

Ausgangspunkt dieses Use-Cases ist ein vollständiges Baukastenmodell in Form einer Ergebnisdatenbank, welches auf externe Kompatibilität (durch Auswahl der Compare Combinations Schaltfläche in der zentralen Nutzeroberfläche) untersucht werden soll. Dem Nutzer bietet sich anschließend die Möglichkeit die Kundenanforderungen an ein Produkt, entsprechend der im Baukastenmodell erfassten Key-Parameter, auf Systemebene einzugeben. Hierzu wird ihm folgende Eingabemaske zur Verfügung gestellt (Abbildung 5.30). Die Eingabemaske gliedert sich entsprechend der Attributtypen (Flussattribute, stationäre Attribute (nicht-quantifizierbar), optionale Attribute und stationäre Attribute (quantifizierbar)) des zugrunde liegenden Baukastenmodells automatisch in vier Sektoren (I-IV). Innerhalb der Sektoren gestaltet sich die Eingabe der Kundenparameter nach einem vergleichbaren Schema (Kennzeichnung im Bild mit Zahl). Zunächst ist der Name des Key-Parameters (1) und falls vorhanden, die zugehörige Einheit (2) angegeben. Sollte für einen Key-Parameter kein Kundenwunsch vorliegen, kann dies über die Auswahl „beliebig“ (3) berücksichtigt werden. Ist ein Key-Parameter als beliebig gekennzeichnet, werden alle bei der Kompatibilitätsberechnung geprüften Kombinationen automatisch als vollständig kompatibel gegenüber dessen Erfüllung gewertet. Liegt ein konkreter Kundenwunsch in Bezug auf einen Key-Parameter vor, kann dieser entsprechend dem vorliegenden Attributtypen spezifiziert werden. Flussattribute und quantifizierbare stationäre Attribute werden über Zahlenintervalle (4a) erfasst. Für nicht-quantifizierte stationäre und optionale Attribute besteht die Möglichkeit den entsprechenden Wert aus drop-down Menüs (4b) auszuwählen. Die Auswahl von optionalen Attributen entscheidet im Nachgang darüber, ob gewisse, optionale Module bei der Kompatibilitätsbetrachtung berücksichtigt werden sollen oder nicht. Der über den Baukasten zu erfüllende Funktionsumfang kann damit innerhalb fest definierter Grenzen situativ angepasst werden. Die Berücksichtigung nicht verhandelbarer, harter Anforderungen kann über eine entsprechende Checkbox (5) erfolgen. Die Auswirkungen, die eine entsprechende Kennzeichnung auf die Kompatibilitätsprüfung mit sich bringt, sind in Kapitel 5.2.1 beschrieben. Sind alle Anforderungen entsprechend bedatet, kann der eigentliche Vergleich mit den im Baukasten enthaltenen Lösungen vollzogen werden.

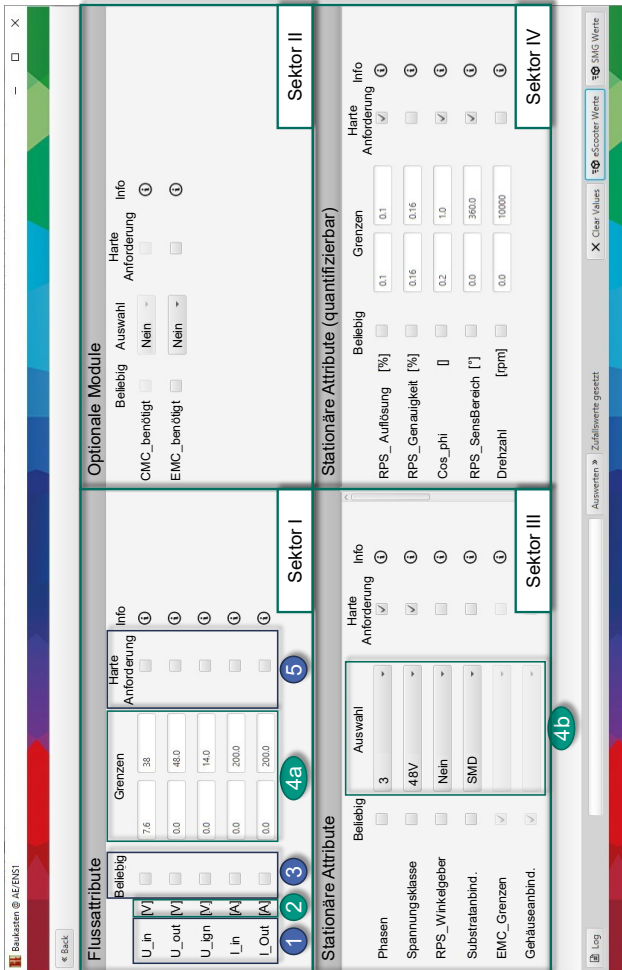


Abbildung 5.30: Übersicht der Eingabemaske der Baukastenanwendung. Diese ist entsprechend der zugrundeliegenden Attributstruktur in Vier Sektoren aufgeteilt.

Die Auswertung der externen Kompatibilitätsprüfung wird dem Nutzer im Ausgabefenster präsentiert (Abbildung 5.31).

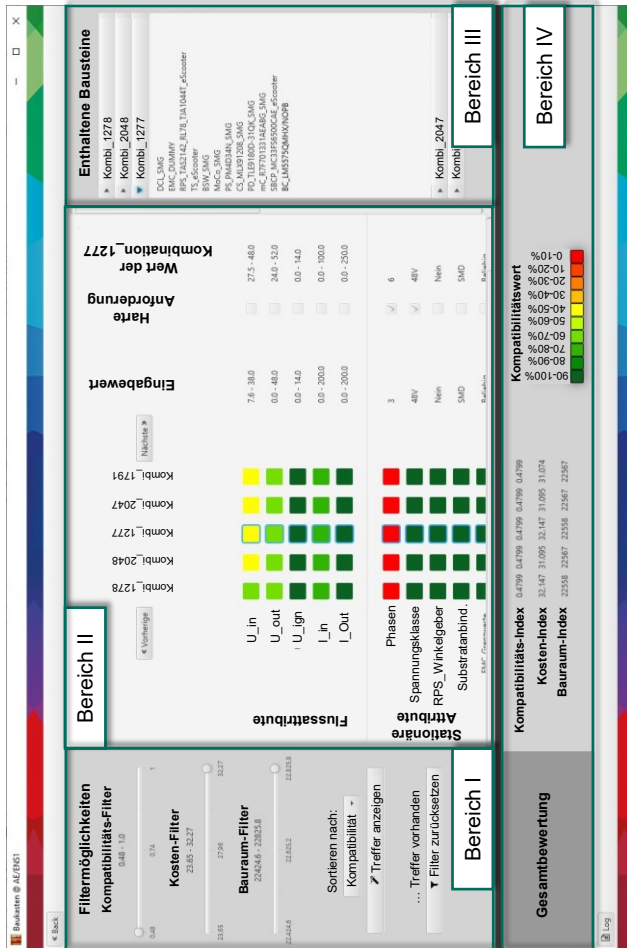


Abbildung 5.31: Ausgabemaske der externen Kompatibilitätsbewertung.

Das Ausgabefenster ist in vier Bereiche untergliedert. Bereich 1 umfasst sämtliche zur Filterung der bewerteten Kombinationen hinsichtlich KPIs angebotenen Funktionen. Die detaillierte Beschreibung der Filterfunktionen wird in Zuge des dafür

vorgesehenen Use-Cases (Kapitel 5.3.3) erfolgen. Im zweiten der vier Bereiche sind die detaillierten Kompatibilitätsinformationen der einzelnen im Baukasten enthaltenen Kombinationen hinsichtlich der eingegebenen Kundenanforderungen aufgeführt (Abbildung 5.32).

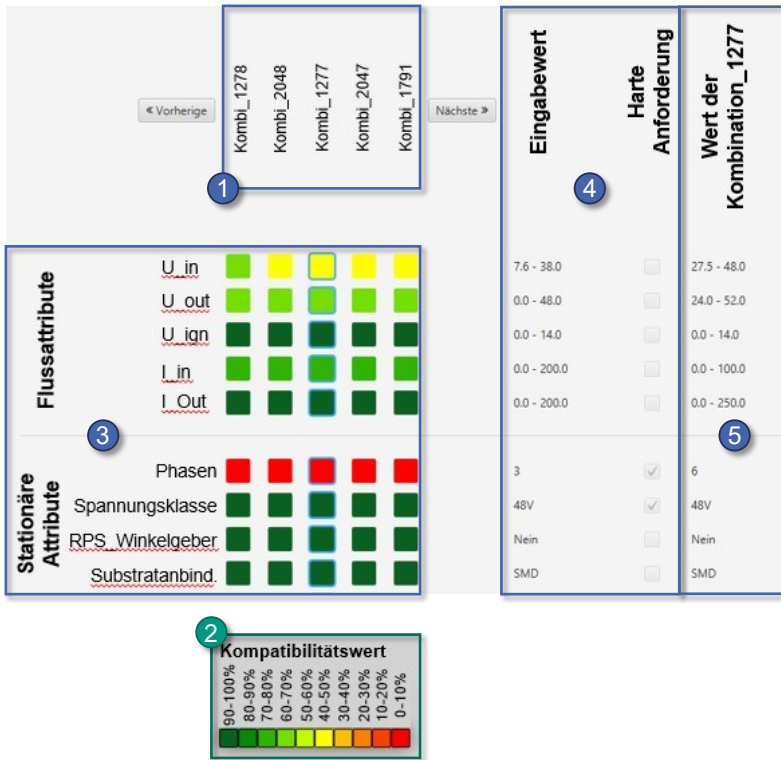


Abbildung 5.32: Bewertung der externen Kompatibilität einer Kombination durch Gegenüberstellung der Eingabewerte mit den errechneten Kombinationswerten.

Der Nutzer erfährt hierdurch eine direkte Unterstützung bei der Lösungsauswahl im Sinne der SPALTEN-Methode. Dem Nutzer werden jeweils fünf der im Baukasten enthaltenen Kombinationen zur gleichen Zeit angezeigt (1). Der für eine Kombinationen angezeigte Name entspricht im Falle, dass diese bisher noch nicht als Produkt umgesetzt wurde einem zufälligen Zahlenwert. Die dargestellten

Kompatibilitätswerte sind demnach allesamt Ergebnisse der Berechnung. Wurde eine Kombination hingegen bereits realisiert, trägt sie den Namen des realen Produktes. Die Kombinationen sind entsprechend ihrer mittels Fuzzy-Logik berechneten Kompatibilitätswerte für die einzelnen Key-Parameter aufgeführt. Um den Erfüllungsgrad der Kompatibilität anschaulich dazustellen wurde eine entsprechende Farbskala eingeführt (2). Die einzelnen Key-Parameter sind entsprechend der Sektoren des Eingabefensters von oben nach unten sortiert (3). Zum detaillierten Vergleich der Kompatibilität mit den Kundenwerten, sind diese inklusive der Kennzeichnung, ob eine harte Anforderung (4) vorliegt aus dem Eingabefenster übernommen. Durch die detaillierte Gegenüberstellung von Anforderungen und Attributen der Kombinationen kann der Nutzer jederzeit die Tragweite einer möglichen Entscheidung abwägen. Von den dargestellten fünf Kombinationen kann jeweils eine Kombination aktiv ausgewählt werden. Die berechneten Kompatibilitätswerte (5) der gewählten Kombination erscheint anschließend zur Rechten der Eingabewerte. Über die Schaltflächen „Vorherige“ und „Nächste“ kann der Nutzer nach und nach durch alle im Baukasten enthaltenen, technisch umsetzbaren Kombinationen navigieren.

Bereich drei des Ausgabefensters ermöglicht es dem Nutzer nachzuvollziehen, welche Bausteinvarianten in den dargestellten Produktkombination enthalten sind (Abbildung 5.33). Es werden jeweils die Bestandteile der als aktiv gewählten Kombination angezeigt. Der Name der einzelnen Bausteinvarianten gibt dem Nutzer jeweils Aufschluss über das Modul, für welches sie ausgewählt wurde und reale Produkte, in denen die Bausteinvariante bereits eingebaut wurde.

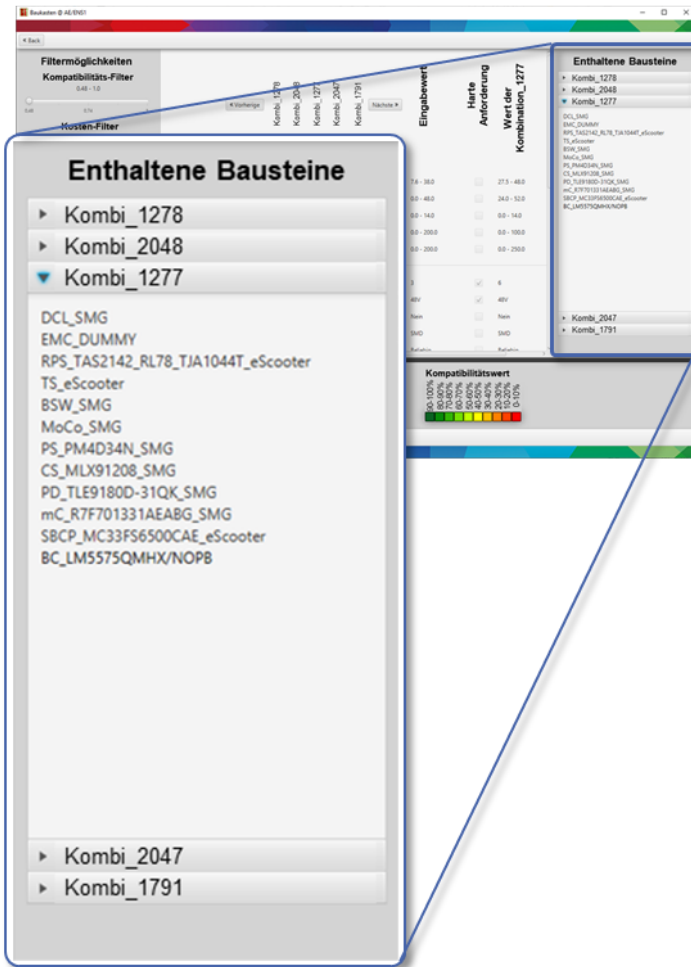


Abbildung 5.33: Übersicht der enthaltenen Bausteine einer Kombination.

Durch einen Klick auf den Namen einer Bausteinvariante erhält der Nutzer zudem Einblick in deren Bausteinsteckbrief (Abbildung 5.34):

mC_R7F701331AEABG_SMG		
Description Attributes X		
Attribute Name	Attribute Type	Attribute Value
		min
U_sensor_in_mC_VOP	Flussattribut	-0.5
U_safe_diag_mC	Flussattribut	-0.5
U_supply_in_mC	Flussattribut	3.0
control_signals_out_mC	Flussattribut	3.0
phase_feedback_in	Flussattribut	-0.5
PWM Signal	Stationäres Attribut	10.0
Kommunikationsschnittstelle	Stationäres Attribut	
Filterguete	Stationäres Attribut	0.0
Sensorinterface	Stationäres Attribut	
Phasen	Stationäres Attribut	
Sensordatenaufnahme	Stationäres Attribut	-0.5
Verfuegbarer RAM	Stationäres Attribut	
FPU	Stationäres Attribut	
Rechnerlaufzeit	Stationäres Attribut	

Abbildung 5.34: Übersicht eines Bausteinsteckbriefs der Baukastenanwendung.

Der Bausteinsteckbrief enthält neben allen Attributen und deren Attributwerte ebenfalls die in Kapitel 4.3 angesprochenen Entwicklungsinformationen aus dem Einsatz des Bausteins in realen Produkten. Dies entspricht der aktiven Nutzung der Informationen der Referenzsysteme.

Bereich 4 dient der letztendlichen Darstellung der Gesamtkompatibilität der fünf angezeigten Kombinationen, sowie deren Bewertung entsprechend der KPIs Kosten und Bauraum (Abbildung 5.35).

<b>Gesamtbewertung</b>	<b>Kompatibilitäts-Index</b>	0.4799	0.4799	0.4799	0.4799	0.4799
	<b>Kosten-Index</b>	32.147	31.095	32.147	31.095	31.074
	<b>Bauraum-Index</b>	22558	22567	22558	22567	22567

Abbildung 5.35: Darstellung der Gesamtkompatibilität inklusive Überblick der KPIs Kosten und Bauraum.

Der Gesamtkompatibilitätswert einer Kombination ergibt sich entsprechend der in Kapitel 5.2.1 vorgestellten Theorie, als Summe gewichteter Einzelkompatibilitäten wiederum als Zahl zwischen 0 und 1.

Dem Nutzer ist über die dargestellten Elemente der Baukastenanwendung die Möglichkeiten gegeben, die im Baukastenenthaltenen Kombinationen mit den Kundenanforderungen zu vergleichen.

### Vergleich möglicher Lösungen anhand von KPIs

Der zweite betrachtete Use-Case ist der Vergleich verschiedener Kombinationen anhand spezifischer KPIs. Hierbei wurde im Vorhinein mit den Stakeholdergruppen eine Beschränkung auf Einzelteilekosten und den Bauraumbedarf der einzelnen Kombinationen vereinbart. Für den Vergleich anhand der genannten KPIs wurden im initialen Baukastenmodell die Einzelteilekosten und der abgeschätzte Bauraumbedarf für alle im Baukasten enthaltenen integralen Bausteinvarianten eingefügt. Die resultierenden Kosten- und Bauraumindizes einer Kombination werden im Zuge der internen Kompatibilitätsprüfung als Summe der Bausteinwerte ermittelt. Der Vergleich der bewerteten Kombinationen findet im Ausgabefenster der Baukastenanwendung statt. Die zur Auswahl vorgesehenen Filtermöglichkeiten befinden sich in Bereich 1 der Ausgabemaske (Abbildung 5.36). Diese ermöglichen es dem Nutzer die Menge der im Baukasten enthaltenen Kombinationen entsprechend ihrer Gesamtkompatibilität, der Kosten- oder der Bauraumindizes einzuschränken. Die angegebenen Filtergrenzen werden automatisch aus den maximalen und minimalen Werten der im Baukasten enthaltenen Kombinationen ermittelt. Nach Einstellen der Filtergrenzen kann der Nutzer über das drop-down Menü „Sortieren nach“ festlegen welcher KPI zur Sortierung der Kombinationen ausschlaggebend sein soll. Nach Filterung der Kombinationen kann der Nutzer anschließend in Bereich 2 der Ausgabemaske durch die gefilterte Liste der Kombinationen navigieren. Eine präzise Auswahl der Kombinationen entsprechend festgelegter KPIs kann somit ermöglicht werden.



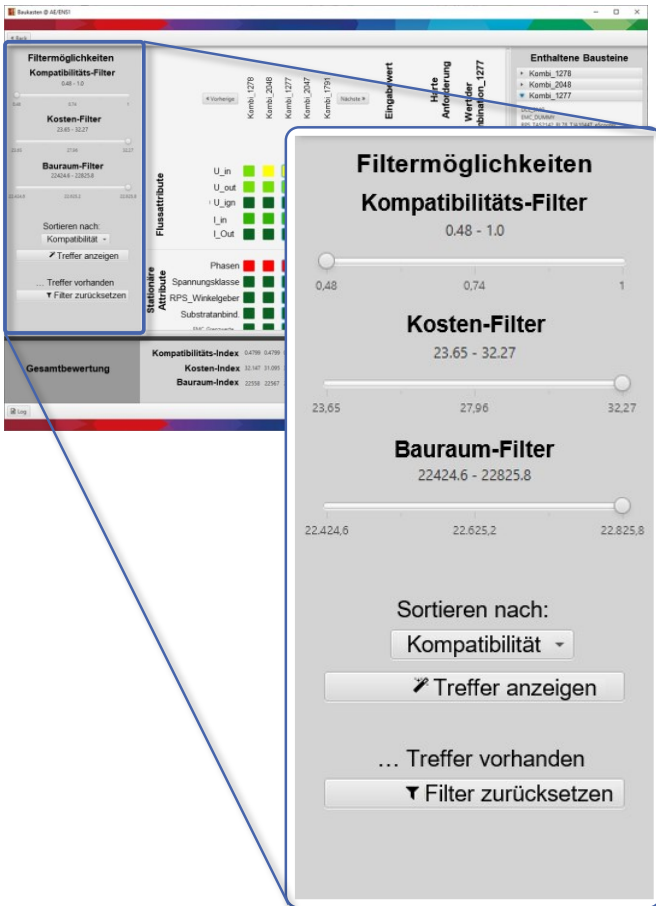


Abbildung 5.36: Filtermöglichkeiten der berechneten Kombinationen

**Ausgabe einer Spezifikation für benötigte Bausteine**

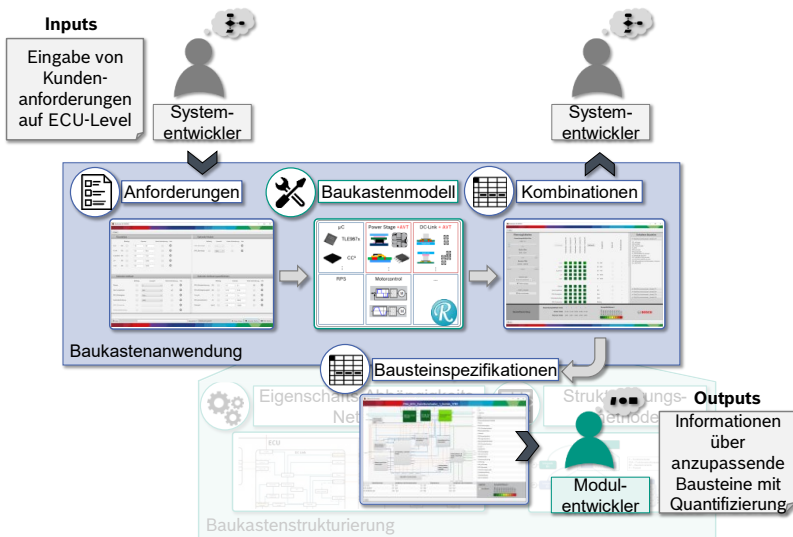


Abbildung 5.37: Ausgabe spezifizierter Bausteingrenzen entsprechend der Anforderungen in Kapitel 5.2.2.

Seitens der Modulentwickler wurde eine Unterstützung möglicher Neuentwicklungen durch die Ausgabe spezifizierter Bausteingrenzen gewünscht (Abbildung 5.37). Die theoretischen Grundlagen dieser Unterstützung wurden in Kapitel 5.2.2 bereits diskutiert. In der Ausgabemaske ist für diesen Use-Case eine entsprechende Funktion berücksichtigt. Durch Klicken auf eine der farbigen Schaltflächen der Kompatibilitätsübersicht, erhält der Nutzer eine Übersicht aller, an der Kompatibilität der jeweiligen Kombination beteiligten Bausteinvarianten. So kann er beispielsweise im Falle einer Inkompatibilität bezüglich eines Key-Parameters, die dafür verantwortlichen Bausteinvarianten identifizieren. Die Übersicht ist wie folgt aufgebaut (Abbildung 5.38):

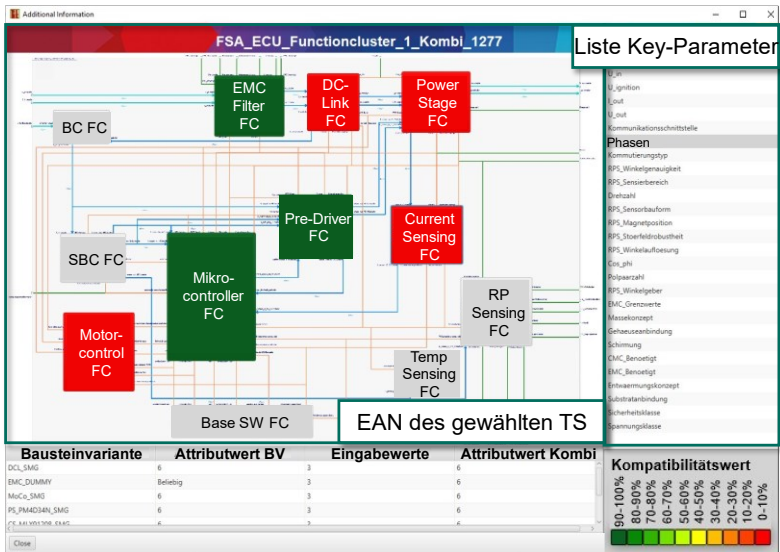


Abbildung 5.38: Auswertungsmaske zur Prüfung des Beitrags einer Bausteinvariante. Auf der linken Seite ist dargestellt, welche der Bausteinvarianten den Eingabewert erfüllen bzw. nicht erfüllen können und welche der BV nicht vom Eingabewert betroffen sind. Die rechte Seite dient der Navigation durch die einzelnen Parameter.

Anhand des obersten, im zugrunde liegenden Baukastenmodell enthaltenen Funktionscluster, sind die Kompatibilitäten der einzelnen Bausteinvarianten hinsichtlich eines konkreten Key-Parameters gekennzeichnet. Zur Kennzeichnung findet die gleiche Farbskalierung, wie zur Ausweisung der Gesamtkompatibilität genutzt, Anwendung. Nicht betroffene Bausteine werden in grau belassen. Auf der rechten Seite der Übersicht, befindet sich eine Liste aller durch den Kunden spezifizierten Key-Parameter. Durch Anwählen mit der Maus können deren Kompatibilitätsverteilungen einzeln betrachtet werden. Im unteren Teil des Übersichtsfensters sind die Attributwerte der Bausteinvarianten der ausgewählten Kombination, den Eingabewerten des Kunden und den Attributwerten der Kombination gegenübergestellt.

Im Zuge dieser Darstellung kann der Nutzer abweichende oder benötigte Bausteinvarianten identifizieren und sich eine Auskunft über die notwendigen Grenzwerte ausleiten. Eine Unterstützung bei der Spezifikation benötigter

Bausteine und der Anstoß zur aktiven Nutzung von Elementen eines Referenzsystems ist somit realisiert.

### Übersicht der verschiedenen Bausteinvarianten des Baukastens

Um dem Management entsprechend des in Kapitel 5.1 ermittelten Use-Case 4 eine Übersicht der verschiedenen Bausteinvarianten geben zu können wurde eine zusätzliche Funktion für die Baukastenanwendung realisiert.

Diese Funktion lässt sich, nachdem die gewünschte Ergebnisdatenbank verbunden ist, direkt aus der zentralen Benutzeroberfläche starten. Hierzu wählt der Nutzer die Schaltfläche „Preselect Modules“ im ersten Abschnitt der Oberfläche (Abbildung 5.39).



Abbildung 5.39: Schaltfläche zur Vorauswahl einer Bausteinvariante

Der Nutzer erhält anschließend eine Übersicht aller im Baukasten enthaltenen Module entsprechend des obersten FSA der Baukastenstrukturierung (Abbildung 5.40).

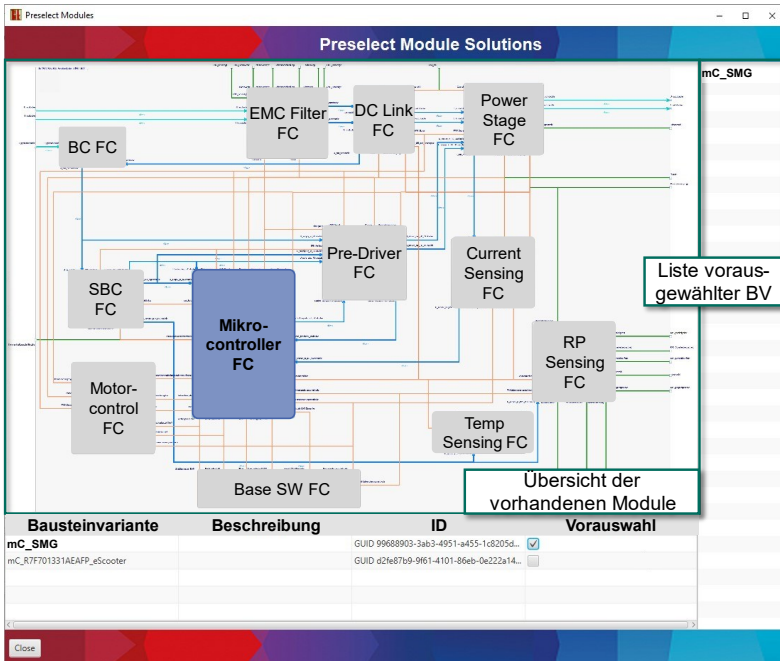


Abbildung 5.40: Übersicht der vorhandenen integralen Bausteinvarianten eines gewählten Technologiesatzes. Durch Auswahl einer Bausteinvariante erscheint diese in der Liste am rechten Bildrand.

Wählt er eines der Module aus, bekommt er eine Übersicht aller integralen und modularen Bausteinvarianten, die zu Realisierung des Moduls im Baukasten enthalten sind. Auf diese Weise kann sich der Nutzer einen detaillierten Überblick der Variantenvielfalt der Bausteine verschaffen. Gleichzeitig wird dem Nutzer eine weitere Funktionalität an die Hand gegeben. Über die Schaltfläche „Preselect“ kann der Nutzer einzelne Bausteinvarianten für eine besondere Art der externen Kompatibilitätsprüfung auswählen. Bei dieser werden statt aller im Baukasten vorhandenen Kombinationen nur diejenigen berücksichtigt in denen die vorausgewählten Bausteine enthalten sind. Abbildung 5.41 zeigt das Ergebnis der externen Kompatibilitätsbetrachtung, bei der ein spezifischer Mikrocontroller in der

Vorauswahl markiert wurde. Mittels dieser Funktionalität können beispielsweise neuartige Technologien auf deren möglichen Einsatz im Baukasten hin untersucht bzw. bereits durch den Kunden vorgegebene Bausteine berücksichtigt werden.

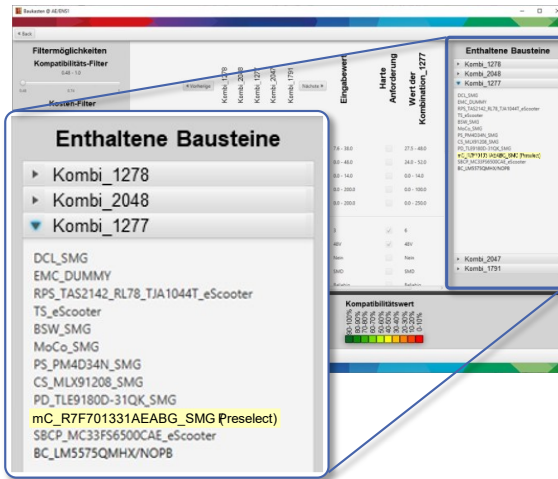


Abbildung 5.41: Ergebnis der Kompatibilitätsbewertung bei getätigter Bausteinvorauswahl. Die vorausgewählte Bausteinvariante ist farblich hervorgehoben und weist die Namensweiterung (Preselect) auf.

## Historie der Baukastenfragen hinterlegen und auswerten

Abschließend wird das Anlegen und Auswerten einer Historie der Baukastenfragen betrachtet. Dieser Punkt dient neben der Identifikation weiterer Lücken des Baukastens der Tragweitenanalyse sowie dem Nacharbeiten und Lernen. Werden bestimmte Parameterausprägungen häufiger angefragt, bzw. wird eine Bausteinvariante über mehrere Anfragen als das schwächste Glied einer Kombination identifiziert, kann über die Historie der Handlungsbedarf für die Technologieentwicklung erhärtet werden. In der vorliegenden Implementierung wurde dies durch das automatische Anlegen von Log-Dateien des Ausgabefensters realisiert. Diese können anschließend in handelsüblichen Textverarbeitungsprogrammen zusammengeführt und dem Management anschließend in kondensierter Form zur Verfügung gestellt werden.

## 5.4 Diskussion und Zwischenfazit

Aufbauend auf den Ergebnissen der Baukastenstrukturierung wurde unter den ermittelten Stakeholdergruppen, eine deskriptive Studie zur Ermittlung der Bedarfe an die Baukastenanwendung durchgeführt. Hierbei wurde das initiale Baukastenmodell als Schnittstelle der beiden, im Rahmen der vorliegenden Arbeit entwickelten MBSE-gestützten Methoden identifiziert (Abbildung 5.42).

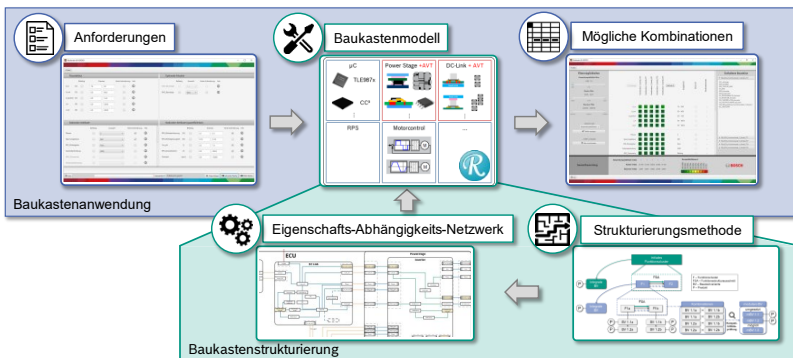


Abbildung 5.42: Übersicht der entwickelten Methoden zur Baukastenstrukturierung und -anwendung für mechatronische Steuergeräte.

Neben den allgemeinen Bedarfen an die Baukastenanwendung, konnten als zusätzliches Ergebnis der empirischen Studie, konkrete Bedarfe an die spezifische Implementierung in Form einer Toolkette ermittelt werden.

Im Anschluss der deskriptiven Studie I wurde, analog zur Baukastenstrukturierung des Kapitels 4, eine zweigeteilte präskriptive Studie durchgeführt. Im Zuge der präskriptiven Studie wurde die Methode zur Baukastenanwendung zunächst auf Basis theoretischer Überlegungen zur Kompatibilitätsprüfung (Kapitel 5.2) entwickelt. Anschließend wurde die, unter den Rahmenbedingungen der Untersuchungsumgebung geschaffene, MBSE-gestützte Toolkette zur Baukastenanwendung entsprechend der ermittelten Stakeholder Use-Cases vorgestellt. (Kapitel 5.3). Die getrennte Betrachtung von Theorie und praktischer

Umsetzung wurde dazu genutzt, die zu Grunde gelegten Teilforschungsfragen sequenziell zu beantworten:

*FF2.1: Wie werden die Baukastenstrukturierung und –anwendung mechatronischer Steuergeräte mittels MBSE-Methoden entsprechend gestaltet?*

*FF2.2: Wie werden die Baukastenstrukturierung und –anwendung mechatronischer Steuergeräte mittels MBSE-Methoden entsprechend integriert?*

Durch die entwickelte Methode zur Baukastenanwendung, die das Ergebnis der Präskriptiven Studie darstellt, konnten alle Elemente des in Kapitel 5.1 ermittelten Zielsystems entsprechend befriedigt werden.

Die entwickelten **Mechanismen zur internen und externen Kompatibilitätsprüfung** bilden das theoretische Methodengerüst der Baukastenanwendung. Die Bedarfe der einzelnen Stakeholdergruppen lassen sich bereits größtenteils durch deren Anwendung abdecken. Zur Erinnerung sind die Bedarfe der einzelnen Stakeholder in nachfolgender Abbildung erneut aufgeführt (Abbildung 5.43).

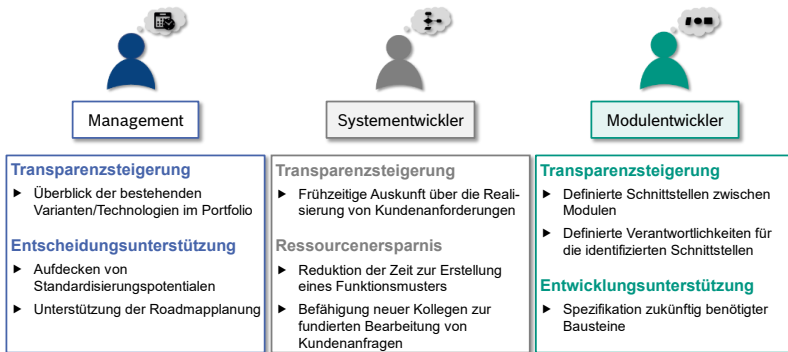


Abbildung 5.43: Initiale Bedarfe der Stakeholder hinsichtlich der entwickelten Baukastenanwendung

Mittels der **internen Kompatibilitätsprüfung** wird einerseits der Lösungsraum des Baukastens hinsichtlich der abzuleitenden Produkte aufgespannt, andererseits wird das Baukastenmodell hinsichtlich der verfügbaren modularen Bausteinvarianten vervollständigt. Auf Basis des vollständigen Baukastenmodells, kann dem Management mit einfachen Mitteln ein **Überblick der vorhandenen**



**Varianten/Technologien** gegeben werden. Die interne Kompatibilitätsprüfung bietet damit eine signifikante Unterstützung des Nutzers durch das Aufzeigen alternativer Lösungen in Sinner der SPALTEN-Methode.

Der Bedarf der Systementwicklung nach einer Möglichkeit zur **frühzeitigen Auskunft über die Realisierbarkeit von Kundenanfragen** kann mittels der **externen Kompatibilitätsbetrachtung** adressiert werden. Diese dient dem Vergleich spezifischer Kundenanforderungen mit den im Baukasten vorhandenen Produktkombinationen und trägt entscheidend zur späteren Lösungsauswahl bei. Alle weiteren Bedarfe seitens der Systementwicklung gründen ebenfalls auf den Ergebnissen der einzelnen Kompatibilitätsbetrachtungen, erweisen sich allerdings als hochgradig abhängig von der letztendlichen Implementierung der Baukasten-anwendung.

Von der Modulentwicklern geäußerte Bedarfe nach definierten Schnittstellen und Zuständigkeiten sind bereits über die Baukastenstrukturierung abgedeckt, werden durch die **Parametrierung der modularen Bausteinvarianten** allerdings sinnvoll ergänzt.

Das vom Management gewünschte **Aufdecken von Standardisierungspotentialen**<sup>18</sup>, und die von den Modulentwicklern angefragte Unterstützung bei der **Spezifikation zukünftiger Bausteinvarianten** können über eine Kombination beider Kompatibilitätsbetrachtungen realisiert werden. In diesem Zusammenhang wurde gezeigt, wie die Anwendung des Baukastens dazu genutzt werden kann **etwaige Lücken des Baukastens aufzudecken** und damit verbunden **Entwicklungspotenziale auszuweisen**. Grundlage dieser Betrachtung ist die aktive **Nutzung der Informationen aus Referenzsystemen** im Sinne der bewussten PGE. Da die Baukasten-anwendung in erster Linie von den als Key-Parameter definierten Attributen des initialen Baukastenmodells abhängig ist, lässt sich eine Übertragbarkeit auf einen großen Teil weiterer mechatronischer Produkte erahnen. An dieser Stelle bietet sich Raum für weitere Forschungsarbeit.

Im Zuge der Implementierung wurde eine **MBSE-gestützte Toolkette** zur Baukasten-anwendung entwickelt. Hierzu wurden die spezifischen Bedarfe der Stakeholdergruppen im Vorfeld als Use-Cases für die Interaktion mit der Toolkette, über dedizierte Ein- und Ausgabeprozesse definiert. Bei der Umsetzung wurden die im Unternehmen vorliegenden Randbedingungen in Bezug auf die **Tool-**

---

<sup>18</sup> Dieses wirkt sich auch direkt auf die Unterstützung der Roadmapplanung aus

**Verfügbarkeit** für die einzelnen Stakeholdergruppen durch eine **intelligente Benutzeroberfläche** berücksichtigt. Mittels MBSE-gestützter Automatisierung der Baukastenanwendung, kann das gesammelte Erfahrungswissen der Frühen Phase der PGE mechatronischer Steuergeräte einer deutlich bereiteren Masse an Nutzern zur Verfügung gestellt werden. Dies kann gerade in der Einarbeitung neuer Kollegen dazu genutzt werden, einen schnellen Überblick über die Technologien zu ermöglichen. Der vermeintlich größte Vorteil der automatisierten Baukastenanwendung liegt allerdings darin **technisch mögliche, von inkompatiblen Lösungen** im Vorfeld zu separieren. Dies kann v.a. in der Frühen Phase des Produktentstehungsprozesses dazu genutzt werden Produktkonzepte frühzeitig zu validieren und damit die Konkretisierung des vagen Zielsystems voranzutreiben. Die Gestaltung der Toolkette auf Basis eines förderierten Ansatzes ermöglicht es jederzeit weitere, den Experten bereits bekannte Tools miteinzubeziehen. Auf diese Weise kann die **Akzeptanz der automatisierten Baukastenanwendung** weiter erhöht werden.

Hinsichtlich der Kompatibilitätsprüfungen bieten sich unzählige Möglichkeiten die erzielten Ergebnisse hinsichtlich ihrer Genauigkeit zu verbessern. Vielversprechend erscheint hier beispielsweise die Berücksichtigung der von Ishii et al. (1988) entwickelten Matching Range. Weiterhin lassen sich die Performancedaten realer Produkte in die Berechnung der Kompatibilitäten miteinbeziehen, um die theoretische Prüfung von Kombinationen realitätsnäher zu gestalten. In diesem Zusammenhang bedarf es der weiteren Auseinandersetzung mit der Fragestellung, wie die Informationen aus Referenzsystemen zielgerichtet in den spezifischen Entwicklungsschritten zur Verfügung gestellt werden können. Eine weitere sinnvolle Erweiterung der Baukastenanwendung stellt die durchgängige Integration komplexer Verhaltensmodelle dar. Darüber hinaus können die technischen Informationen der Bausteine um planerische, beispielsweise in Form von Referenzprozessen des iPeM-Phasenmodells integriert werden.

## 6 Validierung der entwickelten Methoden

Das vorliegende Kapitel dient der Darlegung der Ergebnisse der als initial ausgeprägten deskriptiven Studie II<sup>1</sup>. Zu Beginn werden die, zur Evaluation der entwickelten Methoden, herangezogenen Kriterien in Form von Impact Models<sup>2</sup> dargelegt (Kapitel 6.1). Auf Basis der Impact Models findet anschließend eine Vorstellung der ausgewählten Methoden zur Evaluation und Erfolgsbewertung statt (Kapitel 6.2). Im Nachgang werden die Ergebnisse der durchgeführten Validierungsmaßnahmen entsprechend der beteiligten Stakeholdergruppen ausgewiesen (Kapitel 6.3). Zum Abschluss des Kapitels werden die mittels deskriptiver Studie II geschaffenen Ergebnisse reflektiert und diskutiert (Kapitel 6.4).

### 6.1 Impact Models

Dieser Abschnitt dient der Erläuterung, der zur Validierung der entwickelten Methoden ermittelten Impact Models. Ein Impact Model nach Blessing & Charkrabarti (2009) zeigt den gewünschten Soll-Zustand einer wissenschaftlichen Problemstellung, in Abhängigkeit, der vom Forscher realisierten, methodischen Unterstützung. Hierzu werden, die in der deskriptiven Studie I ermittelten Einflussfaktoren qualitativ mit den einzelnen Aspekten der Unterstützung in Beziehung gesetzt. Anhand ausgewählter, messbarer Erfolgskriterien kann der tatsächliche Einfluss der Maßnahmen anschließend mittels geeigneter Validierungsaktivitäten bestimmt werden. Im Zentrum der Validierungsaktivitäten steht die Beantwortung der nachfolgenden Forschungsfrage:

*FF3: Welchen Beitrag leisten die Methoden zur Baukastenstrukturierung und -anwendung in der Frühen Phase der PGE von mechatronischen Steuergeräten zur Unterstützung der beteiligten Domänen und zur Steigerung der Effizienz im Entwicklungsprozess?*

---

<sup>1</sup> Teile der Validierungsergebnisse sind über folgende Publikationen aufgeteilt veröffentlicht worden: Powelske et al. (2017), Powelske et al. (2018)

<sup>2</sup> vgl. DRM

Um die Beiträge der einzelnen Methoden unabhängig untersuchen zu können, wurde die initiale Forschungsfrage in entsprechende Teilforschungsfragen gegliedert:

*FF3.1: Welchen Beitrag leistet die Anwendung der MBSE-gestützten Strukturierungsmethode für Baukästen mechatronischer Steuergeräte in der Frühen Phase der PGE für die Entwickler?*

*FF3.2: Welcher Beitrag zur Effizienzsteigerung kann im Entwicklungsprozess mechatronischer Steuergeräte, durch den Einsatz MBSE-gestützter Methoden zur Baukasten-anwendung in der Frühen Phase der PGE geleistet werden?*

*FF3.3: Welcher Zusatznutzen entsteht durch die aktive Nutzung der Baukasten-anwendung zur Identifikation potenzieller funktionaler und technischer Lücken?*

Die Veranschaulichung der möglichen Beiträge von Baukastenstrukturierung und -anwendung erfolgt über separate Impact Models. Diese werden im Folgenden vorgestellt.

### **6.1.1 Aufbau der Impact Models**

Für die im Rahmen der vorliegenden Arbeit verwendeten Impact Models wurde die initiale Darstellung von Blessing & Chakrabarti (2009) angepasst. Der Aufbau, der hier vorgestellten Impact Models, umfasst die Zusammenhänge zwischen zwei verschiedenen Arten von Elementen: Einflussfaktoren und Unterstützungen (Abbildung 6.1). Bei den als Ellipsen dargestellten Einflussfaktoren handelt es sich, um die in der DS I ermittelten und innerhalb der Reference Models illustrierten Einflussfaktoren. Die über die PS bereitgestellten Unterstützungen werden als Sechseck dargestellt. Die Zusammenhänge zwischen zwei Elementen erfolgt durch eine gerichtete Verbindung in Form eines Pfeils. Der Grad der Beeinflussung wird abhängig von den Arten der Elemente, durch die angegebenen Symbole auf den Pfeilen verdeutlicht. Zwischen zwei Einflussfaktoren wird die Beeinflussung durch eine Paarung von Symbolen ausgedrückt. Hierbei darf jeweils nur ein Symbol pro Faktor ausgewählt werden. Beispiel: Ein „+“ bei Faktor B und ein „-“ bei Faktor A bedeutet eine Erhöhung von B führt zu einer Absenkung von A.

Auf Seiten des beeinflussten Faktors steht zusätzlich eine „0“ zur Verfügung. Dies bedeutet, dass eine mögliche Beeinflussung besteht, diese aber nicht in jedem Fall

erfolgen muss.<sup>3</sup> Beispiel: Eine erhöhte Anzahl von Änderungen an einem Modell („+“ bei Faktor B) kann zu einer Verringerung der Qualität des Modells führen. Dementsprechend würde hier eine „0“ bei Faktor A vermerkt.

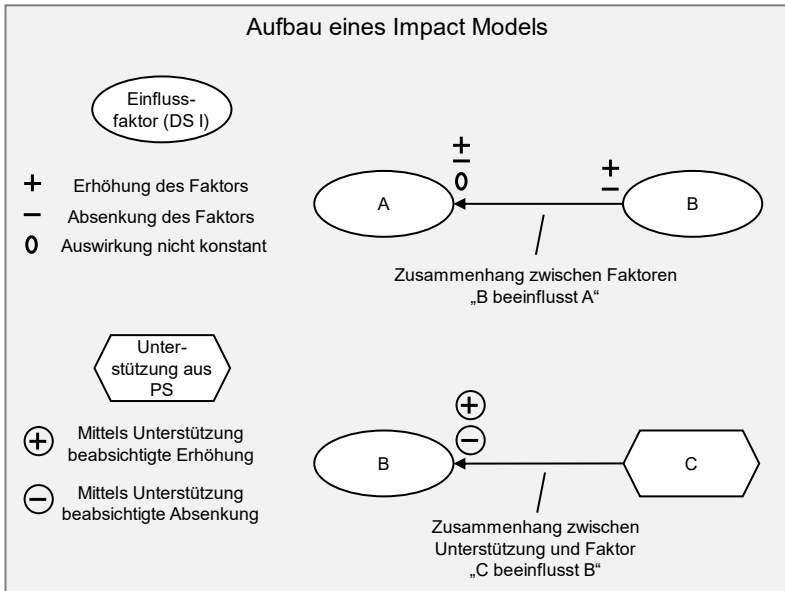


Abbildung 6.1: Aufbau eines Impact Models in Anlehnung an die DRM (Blessing & Chakrabarti, 2009).

Die innerhalb der Impact Models aufgeführten Einflussfaktoren und Zusammenhänge beschränken sich auf die als Ergebnisse der deskriptiven Studien I ermittelten. Es existieren darüber hinaus viele weitere, im Bereich der Modellerstellung und -anwendung betrachtete, häufig bereits über zahlreiche

<sup>3</sup> Dies stellt eine der angedeuteten Anpassungen gegenüber der initialen Modellierung von Blessing & Chakrabarti (2009) dar. Dort bedeutet die „0“, dass keine Beeinflussung besteht.

Studien belegte Aspekte. Eine detaillierte Übersicht jener Aspekte findet sich beispielsweise bei Scherer (2017).

Abhängig von ihrem Beitrag zur wissenschaftlichen Problemstellung werden einige der ermittelten Einflussfaktoren als Schlüsselfaktoren, Erfolgsfaktoren und messbare Erfolgsfaktoren bezeichnet. Die Schlüsselfaktoren wurden bereits in den Reference Models (Kapitel 4.1.2, Kapitel 5.1.1) dargestellt. Sie sind diejenigen Einflussfaktoren, die durch direktes Adressieren mittels unterstützender Maßnahmen, am ehesten zur Verbesserung der Problemsituation beitragen können.<sup>4</sup> Einflussfaktoren deren Beeinflussung den gewünschten Erfolg des Forschungsvorhabens darstellt werden als Erfolgsfaktoren bezeichnet. Da sich der letztendliche Erfolg eines Forschungsvorhabens häufig nur schwer erfassen lässt<sup>5</sup>, bietet es sich an zusätzliche messbare Erfolgskriterien zu definieren. Diese sind so zu wählen, dass deren Beeinflussung im Zeithorizont des Vorhabens verlässlich ermittelt werden kann. Die im Rahmen der ermittelten Impact Models verwendeten Ausprägungen der Einflussfaktoren und Unterstützungen sind in Abbildung 6.2 dargestellt.

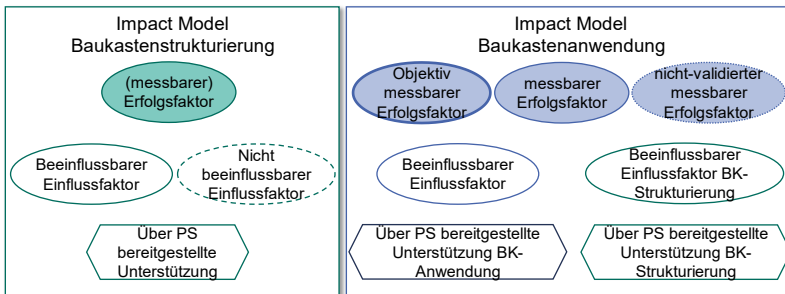


Abbildung 6.2: Ausprägungen der Einflussfaktoren und Unterstützungen der vorliegenden Arbeit in Anlehnung an die DRM (Blessing & Chakrabarti, 2009).

---

<sup>4</sup> vgl. DRM

<sup>5</sup> Dies kann beispielsweise an begrenzten Ressourcen des Forschungsprojektes liegen.

## 6.1.2 Impact Model Baukastenstrukturierung

Bei der Baukastenstrukturierung handelt es sich, um die notwendige Vorarbeit für die aktive Ableitung von Produkten im Rahmen der späteren Baukastenanwendung. Betrachtet man den gewünschten Beitrag der Strukturierungsmethode isoliert, drängt sich deren Akzeptanz in den Reihen der Stakeholder als Erfolgsfaktor auf. Die **Akzeptanz der Strukturierungsmethode** lässt sich durch geeignete Validierungsaktivitäten im Nachgang erfassen. Sie stellt daher gleichzeitig einen messbaren Erfolgsfaktor dar. Zur Übersichtlichkeit wurde das Impact Model zur Baukastenstrukturierung in zwei Teile aufgeteilt. Der erste Teil des Impact Models stellt maßgeblich die Zusammenhänge der in Kapitel 4.1 ermittelten Schlüsselfaktoren mit der realisierten Unterstützung dar (Abbildung 6.3).

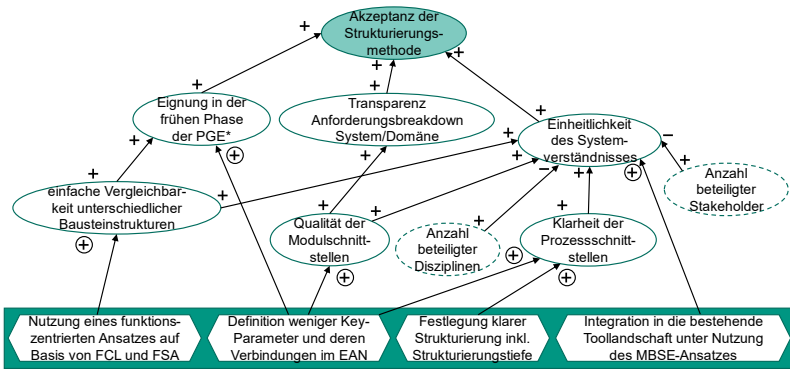


Abbildung 6.3: Rechter Teil des Impact Models der Baukastenstrukturierung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).

Zusätzlich zu den ermittelten Schlüsselfaktoren ist die **Einheitlichkeit des Systemverständnisses** eine bedeutende Einflussgröße auf die Akzeptanz der Strukturierungsmethode. Das Systemverständnis wird durch einen hohen Grad an Interdisziplinarität und eine Vielzahl unterschiedlicher Stakeholdergruppen erwiesenermaßen erschwert (vgl. Kapitel 2.2). Über die erarbeiteten Unterstützungen lässt sich das einheitliche Systemverständnis allerdings direkt, wie im Falle der Nutzung des MBSE-Ansatzes oder indirekt, über die adressierten Schlüsselfaktoren verbessern. Über die von der Modulentwicklung geforderte **Qualität der Schnittstellendefinition** kann zudem ein bedeutender Beitrag hinsichtlich der **Transparenz bei der Anforderungsableitung** geschaffen werden.

Gerade an der Schnittstelle zwischen System und Domäne ist eine transparente Aussage über die Verlässlichkeit der Anforderungen ein wichtiger Aspekt zur Schaffung von Akzeptanz der Strukturierungsmethode. Neben der Verbesserung der Schnittstellendefinition stellt die Identifikation der relevanten Key-Parameter im EAN eine wichtige Voraussetzung für den **Einsatz der Methode in der Frühen Phase**<sup>6</sup> dar. Durch die hohe Unsicherheit in Bezug auf die Anforderungen in der Frühen Phase, müssen Produktkonzepte bereits an wenigen Parametern auf deren Eignung geprüft werden können. Bei der Erfassung der Produktkonzepte spielt die **Vergleichbarkeit der unterschiedlichen Baueinstrukturen** eine große Rolle. Je einfacher diese gelingt, desto besser können die unterschiedlichen Ansätze gegeneinander abgewogen werden.

Im zweiten Teil des Impact Models werden die Aspekte der Modellierung in Bezug auf deren Beitrag zur Akzeptanz der Strukturierungsmethode illustriert (Abbildung 6.4).

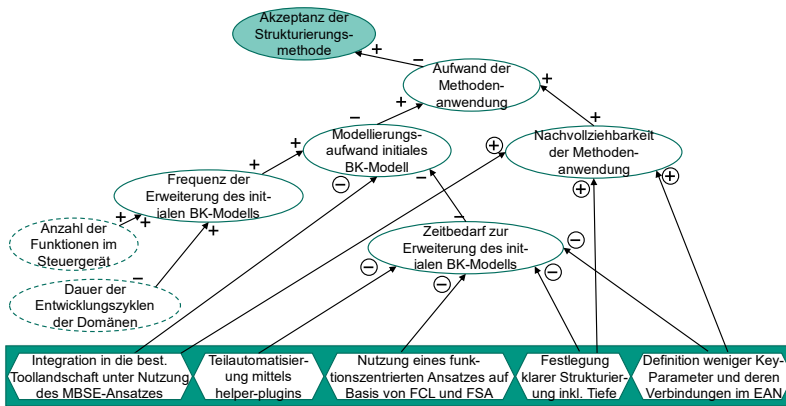


Abbildung 6.4: Linker Teil des Impact Models der Baukastenstrukturierung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).

<sup>6</sup> der PGE mechatronischer Steuergeräte im Einsatz in Kleinantrieben



Im Zentrum der Betrachtung steht der **Aufwand zur Anwendung der Strukturierungsmethode**. Der **Nutzen der Strukturierungsmethode** wird im Zuge der darauf aufbauenden Baukastenanwendung ersichtlich. Dementsprechend findet dieser im Impact Model der Baukastenanwendung Berücksichtigung. Die Methodenanwendung umfasst alle Schritte von der Analyse der bestehenden Produkte bis zur Erstellung des initialen Baukastenmodells. In Bezug auf die Ermittlung der Key-Parameter und der Bausteinsteckbriefe wird davon ausgegangen, dass die Aufwände bereits so gering wie möglich gestaltet wurden. Die Beschränkung der Anzahl Key-Parameter erfolgte beispielsweise unter der Prämisse „So wenige wie möglich, so viele wie nötig“. Eine weitere Beeinflussung dieser Parameter durch den Forscher, ist aufgrund der direkten Abhängigkeit von der Aussage der Experten, hinsichtlich des Akzeptanzgewinnes, als nicht sinnvoll einzustufen. Dementsprechend werden mit der **Nachvollziehbarkeit der Methodenanwendung** und dem **Modellierungsaufwand des initialen Baukastenmodells**, Einflussfaktoren betrachtet, die durch die Unterstützungsmaßnahmen direkt beeinflusst werden können. Der Modellierungsaufwand in Zusammenhang mit dem initialen Baukastenmodell kann dabei stellvertretend für den Modellierungsaufwand von MBSE-Methoden angesehen werden. Dieser stellt, wie in Kapitel 2.2.7 erörtert bis heute eine große Hemmschwelle für den flächendeckenden Einsatz der MBSE-Methoden dar (vgl. Kapitel 2.2.7). Durch die Abbildung des initialen Baukastenmodells in der bestehenden Toollandschaft, kann die Toolkenntnis des Nutzers vorausgesetzt werden, was die Aufwände bereits senken kann (Scherer, 2016). Die zunehmende Funktionsanzahl der Steuergeräte und die geringen Entwicklungszyklen der beteiligten Domänen erhöhen die **Frequenz**, mit der **Anpassungen am Modell** getätigt werden müssen. Seitens der methodischen Unterstützung lässt sich der **Zeitbedarf für die notwendigen Anpassungen** beeinflussen. An dieser Stelle wurde von Powelske et al. (2018) bereits nachgewiesen, dass der gezielte Einsatz von helper-plugins eine signifikante Verringerung der Zeitdauer bewirken kann. Gerade bei der Überführung von Daten aus Referenzsystemen z.B. Anforderungs- und Funktionskatalogen liegen große Potenziale. Nahezu alle Aspekte der erwirkten Unterstützung wurden unter der Prämisse der einfachen und kontinuierlichen Erweiterbarkeit entwickelt. Die **Nachvollziehbarkeit der Methodenanwendung** wird über die klaren Strukturierungskriterien, die Einbindung der Entwickler in den Definitionsprozess der Key-Parameter und die Nutzung der bekannten Toolumgebung positiv beeinflusst.

Das vorgestellte Impact Model stellt den gewünschten Zustand dar, der beim Einsatz der Strukturierungsmethode erreicht werden soll. Die dargestellten Zusammenhänge werden im Rahmen geeigneter Validierungsaktivitäten evaluiert (Kapitel 6.2).

### 6.1.3 Impact Model Baukastenentwicklung

Der Erfolg einer Baukastenentwicklung muss sich schlussendlich immer an einer langfristigen **Steigerung des Umsatzes** messen lassen. Für die Dauer des zugrundeliegenden Forschungsprojektes erweisen sich solche weitreichenden Aussagen allerdings als unmöglich. Dementsprechend werden, im Zuge einer Anwendungsvalidierung (Marxen, 2011) erfassbare, messbare Erfolgsfaktoren definiert. Aus der Ermittlung des Zielsystems für die Baukastenentwicklung hat sich die **Dauer bis zur Bearbeitung einer Kundenanfrage** als sinnvolle Messgröße herauskristallisiert. Je schneller eine valide Aussage über die potenzielle Erfüllung spezifischer Kundenanforderungen getroffen werden kann, desto früher sind Folgeaktivitäten, wie die Erstellung von Funktionsmustern möglich. Spinnt man diesen Gedanken weiter, lassen sich hieraus bedeutende Beiträge zu einer verkürzten Time-to-market und in letzter Instanz zur beabsichtigten Umsatzsteigerung konstruieren. Analog zur Baukastenstrukturierung kann ebenfalls die **Akzeptanz der entwickelten Methode** ermittelt werden. Die direkte Abhängigkeit von den Ergebnissen der zugrundeliegenden Baukastenstrukturierung ermöglicht es zudem, direkte Rückschlüsse auf deren Qualität zu ziehen. Im Impact Model der Baukastenentwicklung finden daher zusätzlich ausgewählte Einflussfaktoren aus der Betrachtung der Baukastenstrukturierung Berücksichtigung.

Entsprechend der unterschiedlichen Bedarfe der Stakeholdergruppen wurde das Impact Model in eine management- (Abbildung 6.5) und eine entwicklerorientierte Sicht (Abbildung 6.6, Abbildung 6.7) aufgeteilt.

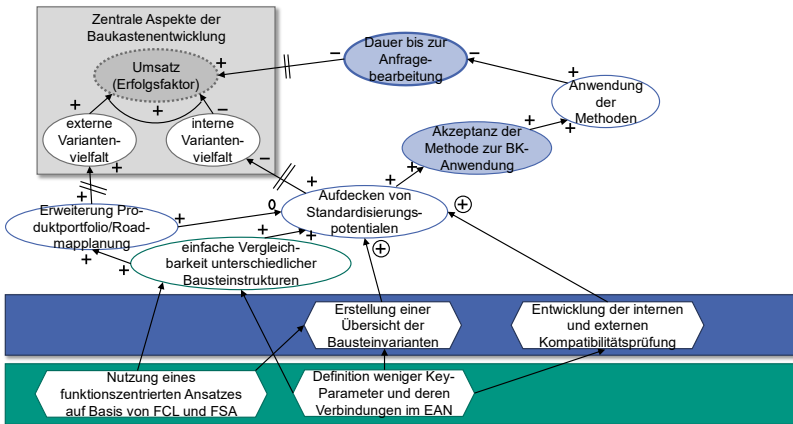


Abbildung 6.5: Impact Model der Managementsicht auf die Baukasten-anwendung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).

Für das Management zählen das **Aufdecken von Standardisierungspotentialen** und die mögliche **Erweiterung des Produktportfolios** zu den unmittelbaren Schlüsselfaktoren. Beide Aspekte stehen in direkter Relation zur Motivation hinter der Einführung von Baukästen: der Erhöhung der externen Vielfalt, bei gleichzeitiger Reduktion der internen Vielfalt. Wie in Kapitel 5.2 dargestellt wurde, kann die Potenzialermittlung hinsichtlich Standardisierung über die Variantenübersicht und die automatisierte interne und externe Kompatibilitätsbetrachtung direkt unterstützt werden. Der Grundstein für die Unterstützung ist, wie in Abbildung 6.5 dargestellt, bereits in der Baukastenstrukturierung angelegt. Nur wenn bereits dort die richtigen Key-Parameter identifiziert wurden und sich der Vergleich der Bausteinstrukturen als nachvollziehbar erweist kann der dargestellte Soll-Zustand eintreten. Mit der Erweiterung des Produktportfolios können über die Roadmapplanung neue Standardisierungspotentiale aufgedeckt werden. Das an dieser Stelle keine zwingende Beeinflussung vorliegt lässt sich bereits ohne Validierungsschritt absehen. Werden die Bedarfe des Managements wie gewünscht adressiert, ist bei einem vertretbaren Aufwand zur Methodenbereitstellung und entsprechender Qualität der Produktempfehlungen mit der **Akzeptanz der Methode** zu rechnen. Beide Punkte werden im Impact Model der Entwicklersicht betrachtet. Ist die Methode zur Baukasten-anwendung akzeptiert steigt die **Bereitschaft zu deren Nutzung**. Tritt der im Impact Model visualisierte Zustand ein, wird davon ausgegangen, dass die Anwendung der Baukastentoolkette eine signifikante

**Verringerung der Bearbeitungszeit einer Kundenanfrage** mit sich bringt. Je häufiger die Methode Anwendung findet, desto eher kann die bisherige Vorgehensweise substituier werden, was eine dauerhafte Effizienzsteigerung mit sich bringt.

Das Impact Model zur Illustration der Entwicklersicht wurde zur Übersichtlichkeit in einen linken und einen rechten Teil aufgetrennt. Der linke Teil des Impact Models weist zwei Schwerpunkte auf. Die Aufwände zur **Anpassung und Erweiterung der Toolkette** und die Beeinflussung der **Qualität der Produktempfehlungen** (Abbildung 6.6).

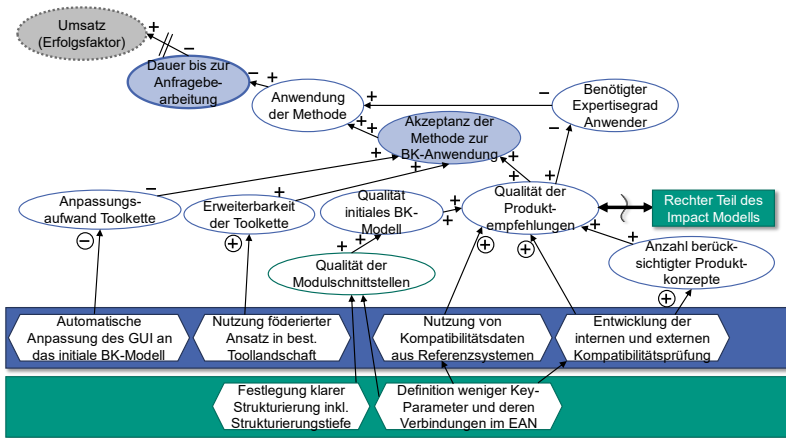


Abbildung 6.6: Linker Teil des Impact Modells der Entwicklersicht auf die Baukastenanwendung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).

Mit beiden Aspekten wird vor allem der Systementwickler bei seiner Tätigkeit unterstützt. Die Ein- und Ausgabemasken der Baukastenanwendung wurden bereits so gestaltet, dass diese sich automatisch auf Basis des zugrundeliegenden initialen Baukastenmodells zusammensetzen. Eine Übertragung des Ansatzes auf weitere Technologiesätze ist damit ohne nennenswerte Anpassungen an der Toolkette möglich.

Die Nutzung des föderierten Ansatzes ermöglicht es jederzeit weitere, in der Untersuchungsumgebung genutzte Tools in die Toolkette einzubinden. Somit wird es den Experten ermöglicht weiterhin auf die ihre bekannte Toolumgebung

zugreifen zu können und die Funktionalität der Baukastenanwendung weiterhin zu erhöhen. Die in Kapitel 5.3.1 beschriebene Integration komplexer Verhaltensmodelle in die Toolkette ermöglicht es detailliertere Kompatibilitätsaussagen treffen zu können. Eine Ausweitung der Baukastenanwendung auf die Serienentwicklung erscheint somit möglich. Die **Qualität der Produktempfehlungen** ist sicherlich der entscheidendste Faktor hinsichtlich der **Akzeptanz der Baukastenanwendung**. Diese kann gleichermaßen durch die Struktur des Baukastens als auch die Kompatibilitätsprüfungen unterstützt werden. Besonders Einfluss hat an dieser Stelle die Nutzung von Kompatibilitätsdaten aus Referenzprodukten. Durch die bereits im Feld befindlichen Produktdaten können Bausteinpaarungen mit größerer Sicherheit als kompatibel ausgewiesen werden, als dies mit bloßer Berechnung der Fall ist. Die im Baukasten realisierte Kombinatorik ermöglicht es zudem große Datenmengen zu verarbeiten. Dies ermöglicht es dem Systementwickler eine deutlich höhere **Anzahl möglicher Produktkonzepte** in seine Überlegungen mit einzubeziehen. Mit steigender Qualität der Produktempfehlung steigt die Verlässlichkeit in die Aussage der Anwendung des Baukastens. Die Prüfung von Kundenanfragen kann somit einer **größeren Anwenderschaft** verfügbar gemacht werden. Gerade neuen Kollegen, kann über die im Rahmen der vorliegenden Arbeit geschaffenen Ergebnisse der Einstieg in die Entwicklungstätigkeiten erleichtert werden. Als Hauptanwender der Toolkette werden in erster Linie die Systementwickler von einer aufwandsarmen und verlässlichen Unterstützung bei der Produktableitung profitieren.

Der rechte Teil des Impact Models (Abbildung 6.7) zeigt neben dem Beitrag der Baukastenanwendung zur **Konkretisierung des Zielsystems** in der Frühen Phase, den Zusatznutzen hinsichtlich der **Unterstützung der Baustein- bzw. Technologieentwicklung**. Das **Aufdecken potenzieller Lücken im Baukasten** öffnet die Türen für weitere Standardisierungspotenziale. An dieser Stelle kann der Bogen zum Impact Model der Managementsicht gespannt werden. Über die gleiche Maßnahme – die Gegenüberstellung der Kundenanforderung mit den Bausteindaten kann dem Modulentwickler eine gewisse Klarheit in Bezug auf die benötigten Bausteinschnittstellen gegeben werden. Mit diesen Ergebnissen ließe sich der Prozess der Technologieentwicklung beschleunigen. Aufgrund der angesprochenen Limitierung der Dauer des Forschungsvorhabens kann dieser Aspekt jedoch nicht validiert werden. Durch eine zunehmende Konkretisierung des Zielsystems der einzelnen Bausteine steigt die Anzahl an Elementen des Produktzielsystems ebenfalls an. Durch eine hohe Qualität der Produktempfehlungen aus dem Baukasten heraus steigt deren Nutzbarkeit für die Funktionsmustererstellung. Je früher ein Funktionsmuster erstellt werden kann, desto belastbarer können Produktkonzepte validiert werden. Dies resultiert

wiederm in einer Konkretisierung des vagen Zielsystems der Frühen Phase der PGE mechatronischer Steuergeräte.

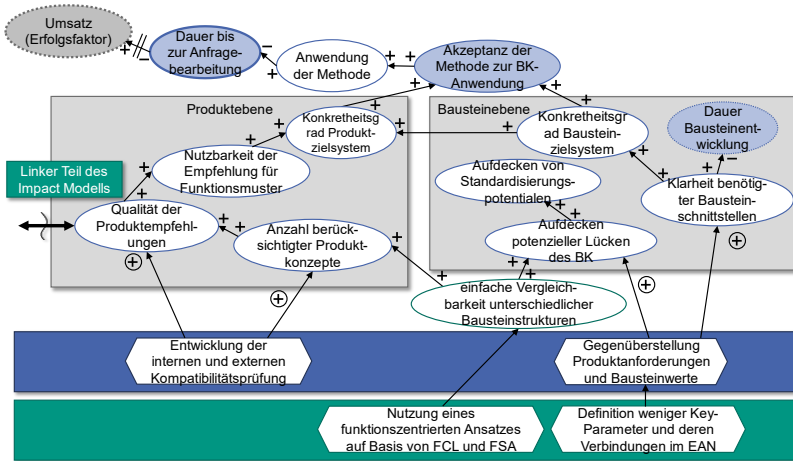


Abbildung 6.7: Rechter Teil des Impact Modells der Entwicklersicht auf die Baukastenanwendung für mechatronische Steuergeräte entsprechend der DRM (Blessing & Chakrabarti, 2009).

## 6.2 Übersicht der Validierungsschritte

In Zusammenhang mit der Erstellung der Impact Models wurde festgelegt anhand welcher Größen der Erfolg der entwickelten Methoden innerhalb der Dauer des zugrundeliegenden Forschungsprojektes ermittelt werden kann. Da sich der eigentliche Erfolgsfaktor, die langfristige Steigerung des Umsatzes einer Betrachtung innerhalb Projektzeitraums entzieht, wurden die **Akzeptanz der entwickelten Methoden** und die **reduzierte Dauer bis zur Anfragebearbeitung** als messbare Faktoren abgeleitet. Das vorliegende Unterkapitel dient der Vorstellung der Validierungsaktivitäten, mit denen die Ergebnisse des Forschungsvorhabens evaluiert wurden.

Zunächst werden einige Randbedingungen für die Validierung im Rahmen der vorliegenden Arbeit betrachtet:

Die Projektlage innerhalb der gewählten Untersuchungsumgebung macht eine umfassende Validierung der Ansätze der vorliegenden Arbeit nicht möglich. Voraussetzung hierfür wäre die parallele Entwicklung eines Produktes mit der neuen und der aktuellen Entwicklungspraxis. Wie bereits bei der Auswahl der Kategorie des Forschungsvorhabens in Kapitel 3.2.1 angeklungen ist, kann die zugrundeliegende deskriptive Studie II dementsprechend nur initial durchgeführt werden.

Zur Messung des Erfolges hinsichtlich der Dauer zur Bearbeitung einer Kundenanfrage werden die bisherige Entwicklungspraxis und die mittels Baukasten-anwendung unterstützte Vorgehensweise im Rahmen von Fallstudien gegenübergestellt. Um eine umfassende Vergleichbarkeit zwischen beiden Ansätzen zu gewährleisten, müssten sämtliche Aufwände der Vorbereitung und Durchführung gegenübergestellt werden. Die Aufwände zur Strukturierung und Anwendung des Baukastens lassen sich sehr genau erfassen. Die Aufwände der aktuellen Entwicklungspraxis hingegen, können lediglich in puncto tatsächliche Bearbeitungsdauer ermittelt werden. Sämtliche vorbereitende Tätigkeiten, wie beispielsweise der Erfahrungsaufbau beteiligter Kollegen, oder die Strukturierung der bisherigen Entwicklungsumgebung liegen bereits weit in der Vergangenheit zurück. Sie lassen sich daher nicht widerspruchsfrei ermitteln. Aus diesem Grund wurde für die Validierung folgende Einschränkung getroffen: Zur Evaluation der Zeitersparnis bei der Bearbeitung von Kundenanfragen werden ausschließlich die tatsächlich zur Aussage über ein geeignetes Produktkonzept benötigten Zeiten gegenübergestellt. Der Schwerpunkt der Validierungsaktivitäten wird daher auf die **Anwendungsvalidierung** in Anlehnung an Marxen (2011) gelegt. Um die Übertragbarkeit des Ansatzes auf Antriebe höherer Komplexität untersuchen zu können, wurden die Methoden der Baukastenstrukturierung und -anwendung ebenfalls auf ein Portfolio des TS4 angewendet. Die Anwendungen des TS 4 weisen gegenüber des TS 1 Beispielsportfolios trotz vergleichbarem Einsatzgebiet<sup>7</sup> eine höhere Komplexität auf. Die höhere Komplexität lässt sich dabei auf höhere Anforderungen hinsichtlich Performance und Sicherheit zurückführen. Abbildung 6.8 zeigt eine Übersicht der unterschiedlichen Anforderungen und Einsatzzwecke der untersuchten Portfolios:

---

<sup>7</sup> Im Rahmen von Kleinantrieben im Automobilbereich

### ANWENDUNG AN 12V ANTRIEBEN (AUTOMOTIVE)

- ▶ Geringe Leistungen: 50-800W
- ▶ Geringe Anforderungen an die Regelung/  
Prozessorstärke ~0 .. 40 DMlps
- ▶ Niedrige Sicherheitsklasse



### ANWENDUNG AN 48V ANTRIEBEN (MULTIMARKET)

- ▶ Hohe Leistungen: 4-11kW
- ▶ Hochleistungsregelung/  
Prozessorstärke ~184 .. 800 DMlps
- ▶ Höchste Sicherheitsklasse



Abbildung 6.8: Gegenüberstellung der Anforderungen der untersuchten 12 V und 48V Antriebe.

### Übersicht der Validierungsmaßnahmen:

Eine Übersicht der im Rahmen der vorliegenden Arbeit zur Validierung der entwickelten Methoden durchgeführten Aktivitäten zeigt Abbildung 6.9.

Die Validierungsaktivitäten sind entsprechend von vier Kriterien geordnet:

- Welche **Methode** kann im vorliegenden Schritt validiert werden?
- Welche **Validierungsmaßnahme** wird für welches Produktportfolio verwendet?
- Welche **Stakeholdergruppe** ist an der Aktivität beteiligt?
- Welche **Teilforschungsfrage** soll durch die Aktivität beantwortet werden?



Bei der Frage nach der zu validierenden Methode wurde berücksichtigt, dass die Baukastenanwendung ebenfalls dazu genutzt werden kann einzelne Aspekte der Baukastenstrukturierung zu evaluieren. Der Erfolg der entwickelten Methoden wurde durchgängig in Form von teilstrukturierten Interviews (Marxen, 2011) mit den beteiligten Stakeholdern ermittelt. Um eine Durchgängigkeit des Entwicklungs- vorhabens auch in der Validierung zu gewährleisten, wurden die gleichen 29 Personen in den Validierungsaktivitäten involviert, die bereits bei der Erhebung der Zielsysteme eingebunden waren. Die Interviews<sup>8</sup> stellen eine ideale Möglichkeit dar, die Akzeptanz der menschenzentrierten Methodenentwicklung zu ermitteln. Auf diese Weise können die individuellen Meinungen der Befragten in Gänze erfasst und deren unterschiedliche Schwerpunkte berücksichtigt werden.

Validierte Methode	Geplante Validierungsmaßnahme		Stakeholder	Beantwortung Teilforschungsfrage
	12V-Antriebe	48V-Antriebe		
BK-A BK-S	Anwendung der Strukturierungsmethode	Anwendung der Strukturierungsmethode	Systementwickler Modulentwickler	Teilforschungsfrage FF 3.1 <b>Beitrag BK- Strukturierung</b>
BK-A BK-S	Evaluation der Schnittstellendefinition/ Moduleingrenzung			
BK-A BK-S	Back-Testing	Back-Testing	Systementwickler	Teilforschungsfrage FF 3.2
BK-A BK-S	Parallel-Testing	Parallel-Testing		<b>Effizienzsteigerung BK-Anwendung</b>
BK-A BK-S	Evaluation der Bedienerfreundlichkeit der implementierten Toolkette		Systementwickler Modulentwickler Management	Teilforschungsfrage FF 3.2 <b>Effizienzsteigerung BK-Anwendung</b>
BK-A BK-S	Evaluation der Identifikation potentieller Lücken des Baukastens		Modulentwickler Management	Teilforschungsfrage FF 3.3 <b>Zusatznutzen BK-Anwendung</b>

**Legende:** BK-A Methode zur BK-Anwendung      BK-S Methode zur BK-Strukturierung

Abbildung 6.9: Übersicht der Validierungsaktivitäten entsprechend deren Verwendung für die entwickelten Methoden und der involvierten Stakeholder.

<sup>8</sup> Die jeweiligen Interviewleitfäden können dem Anhang der vorliegenden Arbeit entnommen werden

Die einzelnen Maßnahmen zur Validierung werden im folgenden Abschnitt chronologisch anhand ihres Beitrages zur Beantwortung der Teilforschungsfragen vorgestellt. Dazu wird insbesondere auf deren Beitrag zu den Erfolgsfaktoren im Impact Model und die Vorgehensweise zur Datenerhebung in Form von Studiensteckbriefen eingegangen:

- **Anwendung der Strukturierungsmethode:**

Abbildung 6.10 zeigt die zur gezielten Evaluation der Strukturierungsmethode durchgeführten Schritte und die angewendeten Forschungsmethoden:

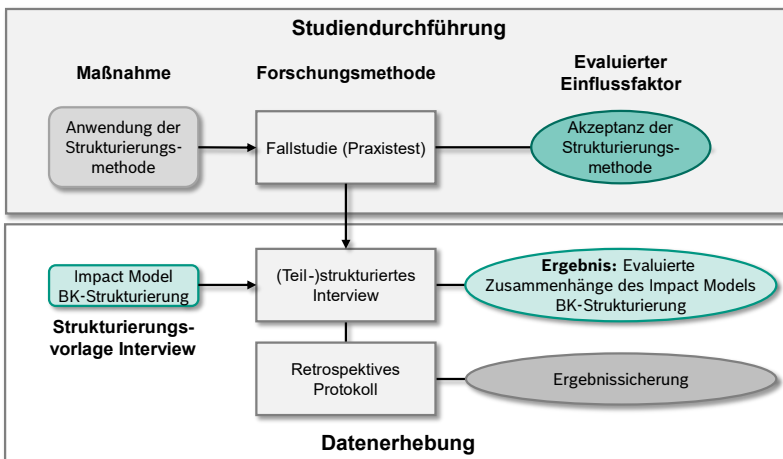


Abbildung 6.10: Studiensteckbrief: Validierung der Anwendung der entwickelten Strukturierungsmethode.

Mit der Validierungsmaßnahme wird das Ziel verfolgt, die Akzeptanz der Strukturierungsmethode im Kreis der späteren Zielanwender zu evaluieren. Die Methode wurde daher zwei ausgewählten Systementwicklern<sup>9</sup> in einem Praxistest

<sup>9</sup> Diese sind Teil, der zur Durchführung der Validierungsaktivitäten ausgesuchten 29 Personen.

zur Verfügung gestellt. Diese sind als spätere Zielanwender verantwortlich für die Umsetzung der Methode anhand des Portfolios der Untersuchungsumgebung.

### **Aufbau des Praxistests:**

Um eine belastbare Aussage hinsichtlich Akzeptanz der Methode zu erreichen, wurden gezielt Entwickler ausgesucht, deren Expertisegrad im Umgang mit MBSE-Methoden sich stark unterscheidet. Während einer der Probanden bereits eingängige Erfahrungen mit der Systemmodellierung in Rhapsody gesammelt hatte, war der andere hauptsächlich mit der dokumentenzentrierten Arbeit vertraut. Der Praxistest wurde weiterhin in zwei Teile aufgeteilt: die Anwendung der Strukturierungsmethode zur Identifikation der Modulstrukturen und die Implementierung des initialen Baukastenmodells in Rhapsody.

Zunächst wurden beide Entwickler gebeten die ersten beiden Schritte zur Baukastenstrukturierung (Kapitel 4.3) eigenständig am Beispielportfolio des TS1 umzusetzen. Dazu wurden den Probanden dieselben Informationen in Form der schematischen Blockschaltbilder und des Funktionskataloges zur Verfügung gestellt. Als Ergebnis der Strukturierungsmethode lagen zwei Baukastenreferenzmodelle inklusive abgeleiteter Modulstrukturen vor. Da der Schritt zur Ermittlung der Key-Parameter als sehr aufwändig gilt, wurde den Entwicklern für den nächsten Teil des Praxistests bereits der ermittelte EAN als Vorlage bereitgestellt.<sup>10</sup> Auf Basis des EAN galt es das initiale Baukastenmodell in Rhapsody zu erstellen. Bei der Modellierung sollten die einzelnen Elemente zunächst ohne den Einsatz der helper-plugins und anschließend mit deren Unterstützung durchgeführt werden. Auf diese Weise können, die im Impact Model dargestellten Zusammenhänge zwischen der Aufwandsreduktion der Modellierung und dem daraus resultierenden Akzeptanzzuwachs zusätzlich geprüft werden. Im Anschluss an den Praxistest wurden die Entwickler getrennt voneinander in teilstrukturierten Interviews hinsichtlich der Akzeptanz gegenüber der Anwendung der Strukturierungsmethode befragt. Dabei wurde vom Autor der vorliegenden Arbeit gezielt auf die in den Impact Models dargestellten Zusammenhänge zwischen den Faktoren („+, -, 0“) eingegangen. Die Interviewergebnisse wurden anschließend in Form retrospektiver Protokolle fixiert und einem finalen Review durch die Probanden unterzogen.

---

<sup>10</sup> Da beide Probanden bereits bei der Ermittlung des EAN beteiligt waren wurde dessen Modellierung innerhalb des Workshops ausgespart.

- **Evaluation der Schnittstellendefinition und Moduleingrenzung:**

Mit den Modulentwicklern wurde in Bezug auf die Baukastenstrukturierung maßgeblich der Schritt zur Festlegung der Modulschnittstellen und deren Verbindung im Rahmen der Eigenschaftsabhängigkeitsnetzwerke untersucht (Abbildung 6.11).

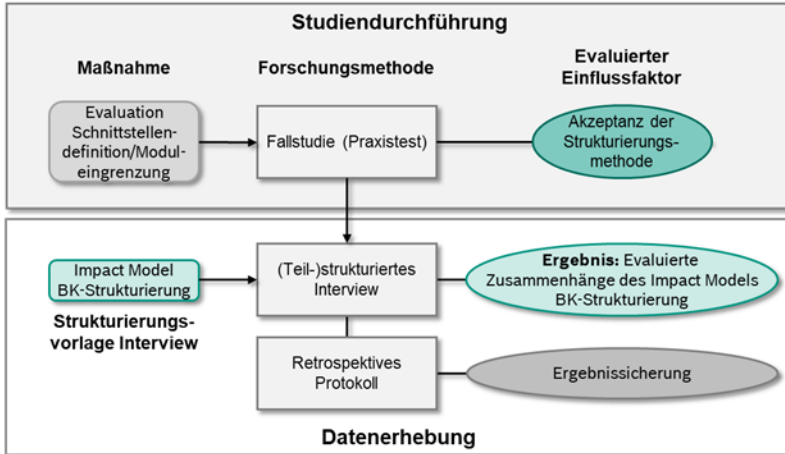


Abbildung 6.11: Studiensteckbrief: Evaluation der Schnittstellendefinition /Moduleingrenzung der entwickelten Baukastenstrukturierungsmethode.

### Aufbau des Praxistests:

Hierzu wurden die ermittelten EANs für die Technologiesätze 1 und 4 insgesamt 15 Modulentwicklern zur Verfügung gestellt. Nach ca. 3 Monaten wurden teilstrukturierte Interviews mit den Entwicklern durchgeführt, bei denen die in den Impact Models dargestellten Zusammenhänge in Bezug auf den Beitrag der EANs diskutiert wurden. Darüber hinaus wurde das Vorgehen zur Strukturierung gemeinsam mit drei Vertretern der Modulentwicklung an einem Beispiel durchexerziert.

Die Vorgehensweise der Methodenapplikation wurde dabei analog zum Praxistest der Systementwickler erlernt. Im Anschluss der Methodenapplikation fanden ebenfalls teilstrukturierte Interviews statt, deren Ergebnisse in Form retrospektiver Protokolle gesichert wurden.

- **Back-Testing:**

Zur Validierung von Baukastenanwendung und -strukturierung gleichermaßen wird unter anderem die Methode des Back-Testings herangezogen (Abbildung 6.12). Beim Back-Testing oder dem historischen Rückvergleich (Walch, 2017) handelt es sich um eine Validierungsmethode bei der, historische Daten zur Überprüfung einer neuen Methode herangezogen werden. Im Falle der vorliegenden Arbeit wird das Back-Testing innerhalb eines Praxistests mit zwei Vertretern der Systementwicklung durchgeführt.

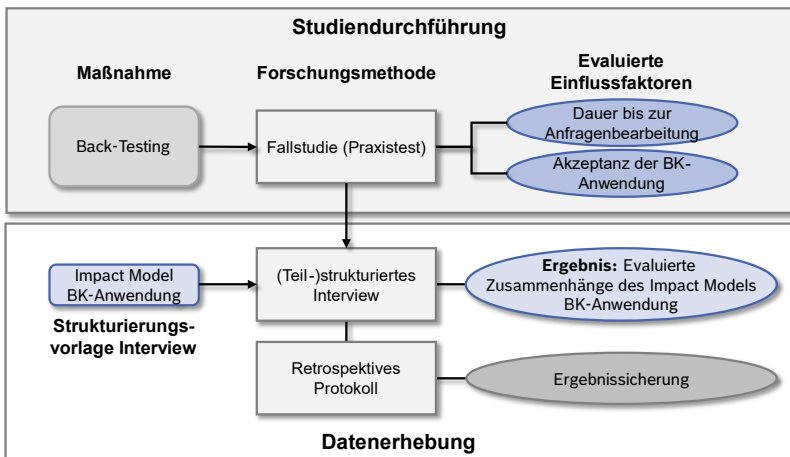


Abbildung 6.12: Studiensteckbrief: Der verwendeten Validierungsmaßnahme Back-Testing.

### Aufbau des Praxistests:

Als Grundlage des Back-Testings werden die Ausgangsdaten einer bereits realisierten Kundenanfrage als Input für die Baukastenanwendung genutzt. Hierzu werden die Kundenanforderungen von den Systementwicklern in die Toolkette zur Baukastenanwendung eingetragen. Anschließend werden die von der Toolkette empfohlenen Kombinationen von den Probanden auf Plausibilität geprüft. Da dieser Schritt in hohem Maße von der Erfahrung der Systementwickler abhängig ist, wurden zum Praxistest zwei Entwickler mit jeweils über 10 Jahren Berufserfahrung

auf dem Gebiet der Entwicklung elektrischer Antriebe eingeladen. Im nächsten Schritt werden die Entwickler gebeten die vom Baukasten angebotenen Lösungen mit den, bei der erstmaligen Bearbeitung der Kundenanfrage berücksichtigten Produktkonzepten zu vergleichen. Hierfür liegen den Entwicklern neben dem, bereits vollständig entwickelten Produkt, zusätzlich detaillierte Informationen zur getroffenen Entscheidung bezüglich des gewählten Produktkonzeptes vor. Im vorliegenden Praxistest handelte es sich bei den Entwicklern jeweils, um die Personen, die die initiale Kundenanfrage betreut hatten. Dementsprechend aussagekräftig ist deren spätere Meinung gegenüber der Plausibilität der vom Baukasten angebotenen Lösungen. Mittels Back-Testing können, neben den Aussagen über die Akzeptanz der Methode detaillierte Erkenntnisse, über den Effizienzgewinn durch die Anwendung der Methode gewonnen werden. Um erste Indizien auf die Übertragbarkeit des Ansatzes auf andere mechatronische Steuergeräte zu bekommen, wurde das Back-Testing jeweils an zwei Steuergeräten aus dem 12V-Segment (TS1) und zwei Steuergeräten aus dem 48V-Segment (TS4) durchgeführt. Im Anschluss an das Back-Testing wurden mit den Entwicklern wiederum teilstrukturierte Interviews auf Basis der Impact-Models zur Baukasten-anwendung durchgeführt. In Abhängigkeit der Plausibilität der Produktempfehlungen konnten somit einzelne Zusammenhänge der Impact Models evaluiert werden.

- **Parallel-Testing:**

Als Parallel-Testing wird im Rahmen dieser Arbeit das parallele Bearbeiten einer Kundenanfrage auf Basis der bestehenden Entwicklungspraxis und der neuentwickelten Methode verstanden (Abbildung 6.13).

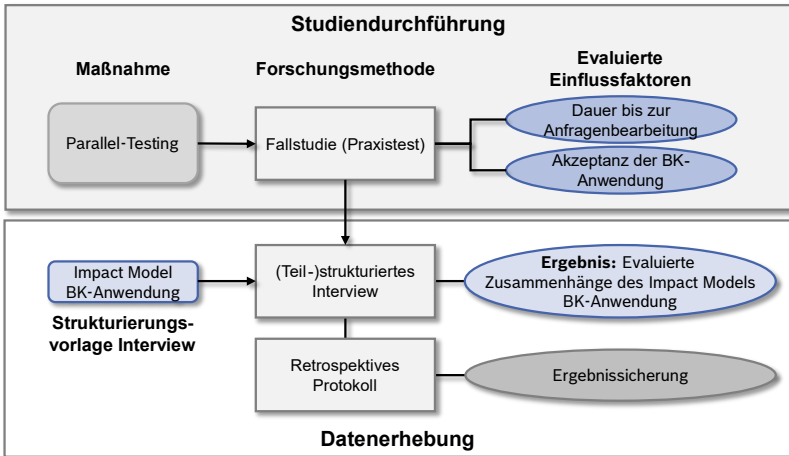


Abbildung 6.13: Studiensteckbrief: Der verwendeten Validierungsmaßnahme Parallel-Testing.

### Aufbau des Praxistests:

Neben der konventionellen Bearbeitung der Kundenanfrage wurde der dazu beauftragte Systementwickler gebeten, die Kundenanfrage über die Toolkette zur Baukasten-anwendung zu bearbeiten. Im Anschluss der Bearbeitung wurden die Ergebnisse beider Prozesse gegenübergestellt. Im Gegensatz zum Back-Testing kann die letztendliche Entscheidung für das zu wählende Produktkonzept aktiv durch die Baukasten-anwendung beeinflusst werden. Mit beiden Testing-Methoden können sowohl die Baukastenstrukturierung als auch die einzelnen Aspekte der Baukasten-anwendung geprüft werden. Voraussetzung für die Akzeptanz der Methoden ist die Korrektheit der vom Baukasten empfohlenen Produktkombinationen. Über das Parallel-Testing kann gleichzeitig die zur Bearbeitung der Kundenanfrage benötigte Zeit ermittelt und der bisherigen Praxis gegenübergestellt werden. Aufgrund der Projektlage innerhalb der Untersuchungsumgebung konnte das Parallel-Testing nur an einem Produkt der 12V-Klasse und der 48V-Klasse durchgeführt werden. Im Anschluss der bearbeiteten Anfragen erfolgte eine zum Back-Testing analoge Sicherung der Forschungsergebnisse.

- **Evaluation der Bedienerfreundlichkeit der implementierten Toolkette:**

Zur Ermittlung der Bedienerfreundlichkeit der implementierten Toolkette wurde diese den Stakeholdern (n=29), die bereits innerhalb der deskriptiven Studie I bei der Bedarfserhebung involviert waren, zur Verfügung gestellt (Abbildung 6.14).

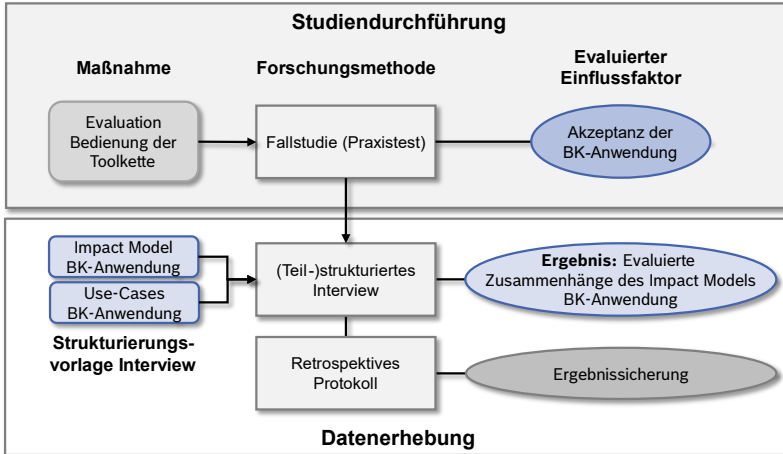


Abbildung 6.14: Studiensteckbrief: Evaluation der Bedienerfreundlichkeit der entwickelten Toolkette.

### Aufbau des Praxistests:

Den Stakeholdern wurde eine persönliche Kopie der Baukasten-anwendung für mehrere Wochen zur Verfügung gestellt. Sie wurden gebeten die Funktionen der Toolkette hinsichtlich der initial ermittelten Use-Cases zu prüfen. Zusätzlich sollten generelle Verbesserungsvorschläge über die Bedienung und Darstellung der Ergebnisse gesammelt werden. Nach Ablauf des Testzeitraums wurden die einzelnen Stakeholder bezüglich der Bedienerfreundlichkeit der Anwendung, der Umsetzung der Use-Cases und generellen Verbesserungsvorschlägen interviewt. Die Ergebnisse der Befragung wurden anschließend in Form retrospektiver Protokolle konsolidiert. Als Vorlage der Befragung wurden die Impact Models zur Baukasten-anwendung und die erarbeiteten User-Stories und Use-Cases genutzt.



• **Evaluation der Identifikation potenzieller Lücken des Baukastens:**

Die Evaluation der Identifikation potenzieller Lücken des Baukastens fand zusammen mit Vertretern des Managements (n=3) und der Modulentwicklung (n=5) im Rahmen eines Praxistests statt (Abbildung 6.15).

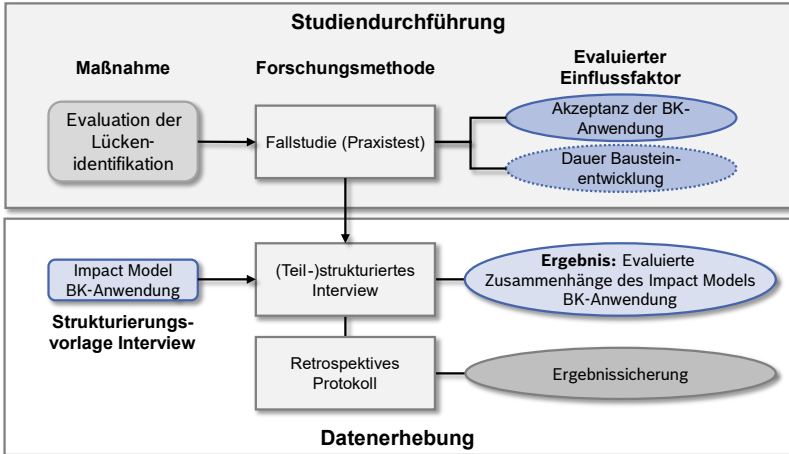


Abbildung 6.15: Studiensteckbrief: Evaluation der Lücken des Baukastens.

**Aufbau des Praxistests:**

Der Praxistest wurde im Rahmen eines Workshops umgesetzt. Innerhalb des Workshops wurden zusammen mit den Stakeholdern diverse Szenarien untersucht, in denen der Baukasten keine völlig kompatible Lösung anbieten kann. Hierzu wurden sowohl historische Daten als auch zukünftig absehbare Parametersätze<sup>11</sup> in die Baukasten-anwendung eingegeben. Anhand der Ergebnisse der

<sup>11</sup> Etwa auf der Basis der angenommenen Weiterentwicklung von Kundenanforderungen z.B. hinsichtlich der Erhöhung der Leistungsdichte eines Antriebes

Baukastenentwicklung wurden diejenigen Bausteine identifiziert, die zur Verletzung der Eingabeparameter führen. Auf Basis der aufgedeckten Lücken wurde die Nutzbarkeit der Informationen über benötigte Bausteine und damit verbunden die Potenziale zur Standardisierung ermittelt. Weiterhin wurde über die Preselect-Funktion dargestellt, wie zukünftige Technologien bereits vor deren Entwicklung auf die Kompatibilität zu den im Baukasten enthaltenen Bausteinen geprüft werden können. Im Zuge dieser Untersuchungen wurden die bisherigen Prozesse zur Aufnahme und Bearbeitung einer Kundenanfrage sowie zur Planung von Neuentwicklungen, um die Nutzung der Baukastenentwicklung ergänzt. Im Anschluss an den Praxistest wurden die Teilnehmer einzeln unter Zuhilfenahme der Impact-Models bezüglich der o.g. Potentiale befragt.

### 6.3 Ergebnisse der Validierungsaktivitäten

Die Ergebnisse der Validierungsaktivitäten werden im Folgenden anhand der abgeleiteten Teilforschungsfragen vorgestellt. Die von den Stakeholdern, im Rahmen der teilstrukturierten Interviews ermittelten Aussagen<sup>12</sup>, werden dazu als Belege der in den Impact Models dargestellten Zusammenhänge präsentiert<sup>13</sup>.

*FF3.1: Welchen Beitrag leistet die Anwendung der MBSE-gestützten Strukturierungsmethode für Baukästen mechatronischer Steuergeräte in der Frühen Phase der PGE für die Entwickler?*

Aus Sicht der am Praxistest beteiligten Systementwickler wurde die Strukturierungsmethode in ihrer Aufbereitung als „nachvollziehbar“ empfunden. Die **Nutzung klarer Strukturierungskriterien** wurde als „sehr positiv und hilfreich für den Anwender“ beschrieben. Die **Ermittlung der Key-Parameter** und die Darstellung der Modulverbindungen im EAN bieten den Systementwicklern ein „mächtiges Werkzeug in der Kommunikation mit den Domänenentwicklern und dem Management“. Das **Eigenschafts-Abhängigkeits-Netzwerk** bietet weiterhin die Möglichkeit „außerhalb der Baukastenentwicklung zur Illustration der Zusammenhänge zwischen externen und internen Parametern“ genutzt zu werden. Die damit erzeugte **Transparenz zwischen System- und**

---

<sup>12</sup> Die Kennzeichnung der Aussagen erfolgt nach demselben Schema wie in Kapitel 4.1.2, wobei die ID der Interviewpartner beibehalten wird

<sup>13</sup> Die innerhalb der Impact Models ermittelten Faktoren sind dazu über die **Textformatierung** hervorgehoben

**Domänenanforderungen** „sorgt für Klarheit in den Zuständigkeiten“ und vermeidet „unnötige Iterationen“. Die Anzahl der ermittelten Key-Parameter wird als „genau richtig für die **Frühe Phase [der PGE mechatronischer Steuergeräte]**“ empfunden. Durch die direkte Einbindung der Modulexperten bei der Definition der Key-Parameter wird die **Qualität der modellierten Schnittstellen** als „sehr hoch“ eingeschätzt. Die **Vergleichbarkeit der modularen und integralen Bausteinvarianten** gilt als „unverzichtbare Grundlage für die Baukastenanwendung“. Der **Modellierungsaufwand des initialen Baukastenmodells** wurde von den Probanden im Vorhinein als „sehr hoch“ eingeschätzt, erwies sich allerdings durch den Einsatz der helper-plugins als „überraschend gering“. Die über die Zeit nötigen **Erweiterungen am initialen Baukastenmodell** wurden aus diesem Grund ebenfalls als „zeitlich unkritisch“ bezeichnet. Für die Erfassung der Bausteinsteckbriefe müsse „im Nachgang ein verbindlicher Prozess geschaffen werden“, um die Aufwände gering zu halten. Vor allem in der Nutzung realer Bausteindaten und Informationen aus bereits abgeschlossenen Projekten sehen die Systementwickler „eine große Chance das Entwicklungsrisiko für Folgeprojekte drastisch zu reduzieren“. Die Frage nach der **Akzeptanz der Strukturierungsmethode** wurde von den Zielanwendern zusammenfassend als „hoch“ bezeichnet, wobei der Einsatz der helper-plugins als „besonderer Potenzialträger hinsichtlich der **Reduktion der Modellierungsaufwände**“ hervorgehoben wurde<sup>14</sup>.

Seitens der Modulentwicklung wurde vor allem der EAN für seine „übersichtliche und auf das Wesentliche konzentrierte Darstellungsweise“<sup>15</sup> als „bedeutende Unterstützung in der Frühen Phase“<sup>16</sup> empfunden. Gleichzeitig kann dieser „als Diskussionsgrundlage in Entwicklungsprojekten“<sup>17</sup> herangezogen werden. Die darin vermerkten Key-Parameter und deren Verbindungen tragen dazu bei, einen „Umgang mit der hohen Komplexität in der Steuergeräteentwicklung“<sup>18</sup> zu ermöglichen. Gleichzeitig werden „die Verantwortlichkeiten und die Herkunft von Schnittstelleninformationen widerspruchsfrei dargestellt“<sup>19</sup>. Die Nutzung eines durchgängigen MBSE-gestützten Modells zur Strukturierung des Baukastens wird

---

<sup>14</sup> Aufgrund der Beteiligung von lediglich zwei Probanden wurde auf die explizite Kennzeichnung der Aussagen in diesem Abschnitt verzichtet. Es handelte sich jeweils um Aussagen der Interviewpartner 20 und 21 (Entwicklung, System)

<sup>15</sup> Interviewpartner 5 (Entwicklung, Hardware)

<sup>16</sup> Interviewpartner 8 (Entwicklung, Hardware), 11 (Entwicklung, Software)

<sup>17</sup> Interviewpartner 2 (Entwicklung, Mechanik)

<sup>18</sup> Interviewpartner 9 (Entwicklung, Hardware), 3 (Entwicklung, Mechanik)

<sup>19</sup> Interviewpartner 7 (Entwicklung, Hardware)

von einem Großteil der Kollegen „positiv“<sup>20</sup> hinsichtlich eines **einheitlichen Systemverständnisses** gesehen. Die mittels Strukturierungsmethode identifizierten Modulgrenzen „erscheinen durchaus sinnvoll“<sup>21</sup> und lassen „keine Fragen bezüglich der Zuständigkeiten offen“<sup>22</sup>. Durch die direkte Einbindung der Experten in die Definition der Key-Parameter und die „nachvollziehbare Vorgehensweise bei der Strukturierung“ wurde ein „echter Mehrwert für die Organisation“<sup>23</sup> geschaffen. Der im Impact Model der Baukastenstrukturierung dargestellte Soll-Zustand wurde von den Modulentwicklern als „erfüllt“<sup>24</sup> bezeichnet.

*FF3.2: Wie kann der Entwicklungsprozess mechatronischer Steuergeräte mittels MBSE-gestützter Methoden zur Baukastenanwendung in der Frühen Phase der PGE effizient unterstützt werden?*

Die mittels Baukastenanwendung erzielte **Effizienz bei der Bearbeitung** von Kundenanfragen wurde im Rahmen des Parallel-Testings ermittelt. In beiden Fällen der Anwendung (TS1 und TS4) sind im Schnitt 60 Minuten bis zur endgültigen Aussage über die Verwendung eines Produktkonzeptes vergangen. Hierin sind sowohl die Eingabe der Anforderungen, die Berechnung der Kombinatorik im Baukasten, sowie die Analyse und Auswahl der Ziel-Kombinationen enthalten. Als Vergleichswert erstreckte sich die Betrachtung mit der bisherigen Entwicklungspraxis, gemittelt auf eine effektive Arbeitszeit von 21 Stunden. An dieser Stelle gilt es zu vermerken, dass es sich bei dem bearbeitenden Entwickler um den Experten handelte, der die bisherige Entwicklungspraxis maßgeblich mitdefiniert hat. Die benötigte Zeit **weniger erfahrener Kollegen** nach der „alten“ Vorgehensweise wurde von dem Experten auf mindestens 1,5 – 2 Wochen geschätzt. Die **Qualität der Produktempfehlungen** wurde weiterhin als „sehr hoch für die Frühe Entwicklungsphase“ eingeordnet und explizit auch für „die Nutzung durch weniger erfahrene Kollegen“ empfohlen. Alle der angebotenen Empfehlungen waren „durchaus plausibel“, wenn auch viele der Kombinationen als „nahezu identisch“ bezeichnet werden konnten. Dies konnte im Nachgang auf den Umstand zurückgeführt werden, dass alle Module im bisherigen Zustand der Baukastenanwendung als gleichwertig für die Konzeptauswahl berücksichtigt sind. Beispielsweise wird durch bloßes Ersetzen eines Temperatursensors bereits eine

---

<sup>20</sup> Interviewpartner 4 (Entwicklung, Hardware), 10 (Entwicklung, Software)

<sup>21</sup> Interviewpartner 3 (Entwicklung, Mechanik)

<sup>22</sup> Interviewpartner 8 (Entwicklung, Hardware)

<sup>23</sup> Interviewpartner 9 (Entwicklung, Hardware)

<sup>24</sup> Interviewpartner 15 (Entwicklung, Software), 1 (Entwicklung, Mechanik)

neue Kombination errechnet. Die Auswirkungen des Sensors auf das Gesamtkonzept sind im Vergleich zum Wechsel eines Mikrocontrollers oder der Endstufe deutlich weniger konzeptentscheidend. Durch eine Gewichtung der einzelnen Elemente hinsichtlich deren Rolle im Steuergerät könnte der Effizienzzuwachs noch deutlicher ausgestaltet werden. Beispielsweise könnten die angebotenen Kombinationen in Haupt- und Nebenvarianten gegliedert werden. Die eingebaute Filtermöglichkeit wurde bereits als eine sinnvolle Option, die Lösungsvielfalt einzuschränken, gesehen. Bereits auf Basis der geringen Anzahl der bei der Baukastenstrukturierung und -anwendung berücksichtigten Produkte war eine deutliche Effizienzsteigerung erkennbar. Durch die Erweiterung des Ansatzes um alle Produkte und die eine Ausweitung auf die noch fehlenden Technologiesätze kann zukünftig ein „enormes Potenzial“ frei werden. Da die Ergebnisse der Produktempfehlungen sowohl für TS1 als auch TS4 Antriebe als plausibel erklärt wurde, lässt sich bereits eine Übertragbarkeit der Methoden auf weitere Arten mechatronischer Antriebe vermuten. Die Untersuchung dieser Annahme bietet eine ideale Grundlage für weitere Forschungsarbeiten.<sup>25</sup>

Die detaillierte Plausibilisierung der vom Baukasten empfohlenen Kombinationen wurde im Back-Testing durchgeführt. Hier wurde analog zum Parallel-Testing ermittelt, dass sich unter der hohen **Anzahl der Kombinationen** viele „sehr ähnliche“ befinden. Durch die geringe Anzahl im Baukastenmodell eingepflegter Bausteinvarianten wurden nur wenige, bei der zurückliegenden Bearbeitung der Anfrage, nicht berücksichtigte und in Bezug auf das Produktkonzept neue Kombinationen ausgegeben. Anhand dieser wurde allerdings deutlich, dass die Baukasten-anwendung in ihrem Zielzustand eine „immense Unterstützung“ bei der Betrachtung des Lösungsraums darstellt. Im Back-Testing wurden weiterhin die Ergebnisse der Kompatibilitätsberechnungen validiert. Hierzu wurden einzelne Kombinationen in der Kompatibilitätsübersicht betrachtet. Abbildung 6.16 zeigt die Kompatibilitätsübersicht am Beispiel des Portfolios der 48V-Antriebe. Die dort angegebenen Kompatibilitätswerte wurden anschließend den Erfahrungswerten des Entwicklers gegenübergestellt. Bei den betrachteten Kombinationen handelte es sich sowohl, um bereits umgesetzte als auch theoretisch mögliche, vom Baukasten empfohlene Lösungen. In allen Fällen wurden die Ergebnisse der Kompatibilitätsprüfung als „plausibel und nachvollziehbar“ eingeschätzt. Eine vom Baukasten empfohlene Kombination eignet sich daher als „Grundlage zum Funktionsmustersaufbau“. Die schnelle Aussagefähigkeit über die mögliche

---

<sup>25</sup> Die Aussagen des vorliegenden Abschnittes zum Parallel-Testing wurden ausnahmslos von Interviewpartner 21 (Entwicklung, System) getätigt.

Umsetzung einer Kundenanfrage in Form von passenden Kombinationen kann aktiv dazu genutzt werden **das vage Zielsystem zu konkretisieren und Produktkonzepte frühzeitig zu validieren**<sup>26</sup>.

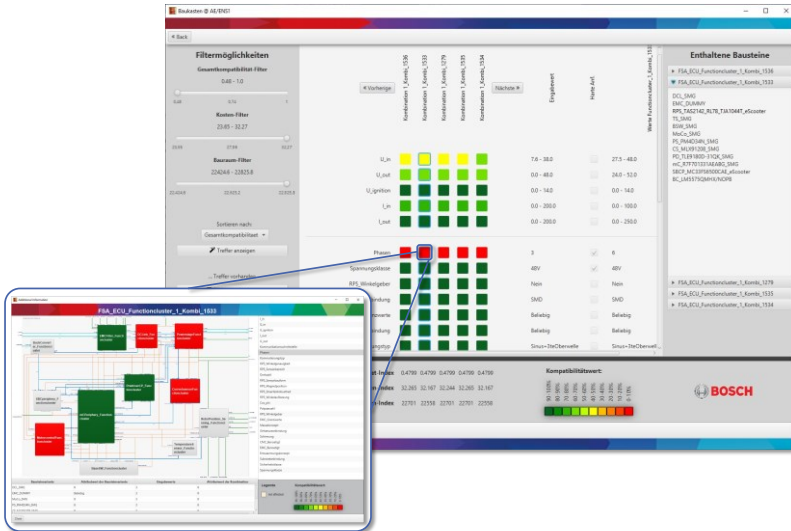


Abbildung 6.16: Kompatibilitätsübersicht der implementierten Bausteinanwendung.

Hinsichtlich der Bedienerfreundlichkeit der Toolkette erwies sich das Feedback aller Befragten als durchweg positiv. So ist das Potenzial der Bausteinanwendung nicht nur „deutlich erkennbar“<sup>27</sup>, sondern durch die Implementierung „sehr zugänglich gestaltet“<sup>28</sup>. Die Umsetzung der einzelnen initial ermittelten Use-Cases sei „durchaus sehr gelungen“<sup>29</sup>. Großen Anteil daran hat die Berücksichtigung der

<sup>26</sup> Die Aussagen zum Back-Testing wurden von Interviewpartner 18 und 21 (Entwicklung, System) getätigt.

<sup>27</sup> Interviewpartner 7 (Entwicklung, Hardware), 16 (Entwicklung, System)

<sup>28</sup> Interviewpartner 25 (Management, Linie), 17 (Entwicklung, System)

<sup>29</sup> Interviewpartner 19 (Entwicklung, System), 12 (Entwicklung, Software)

verschiedenen Rollen beim Zugriff auf das Baukastenmodell in der Benutzeroberfläche. Diese wurde zudem als „wichtiger Enabler für die durchgängige Nutzung der Toolkette“<sup>30</sup> gesehen. Die **Erweiterbarkeit der Toolkette** auf Basis des förderierten Ansatzes wird als „zusätzliche Stärke der Baukastenanwendung“<sup>31</sup> gesehen und eine Erweiterung des Einsatzgebietes des Baukastens über die Frühe Phase hinaus „erscheint sinnvoll“<sup>32</sup>. Die **Aufwände zur Erweiterung und Pflege** der Toolkette wurden bei der Betrachtung ausgeklammert. Diese würde im Falle eines flächendeckenden Einsatzes der Baukastenanwendung professionalisiert und durch die entsprechenden Kompetenzträger im Unternehmen übernommen. Als weitere sinnvolle Ergänzungen wurde die Ausweitung der Betrachtung auf den Antrieb besonders hervorgehoben. Durch die Berücksichtigung aktueller Motortypen könnten die Anforderungen an die Steuergeräte noch detaillierter betrachtet werden. Gleichzeitig sehen die Befragten große Potenziale bezüglich der Berücksichtigung von Maßnahmen der funktionalen Sicherheit in Abhängigkeit der Sicherheitsziele und deren ASIL Einstufung. Die **Akzeptanz der Baukastenanwendung** unter den Stakeholdern konnte durch die Bedienerfreundlichkeit resümierend als „sehr hoch“<sup>33</sup> ermittelt werden.

*FF3.3: Welcher Zusatznutzen entsteht durch die aktive Nutzung der Baukastenanwendung zur Identifikation potenzieller funktionaler und technischer Lücken?*

Im Rahmen der Baukastenanwendung konnte ein zusätzlicher Nutzen durch die aktive Anwendung, der im Baukasten hinterlegten Referenzprodukte und deren Bausteine geschaffen werden. Neben den Aussagen über die potenzielle Erfüllung von Kundenanfragen, kann die Kompatibilitätsprüfung aktiv dazu genutzt werden, **potenzielle Schwachstellen im Baukasten aufzudecken**. Dies kann sowohl auf Basis bestehender als auch zukünftig angenommener Kundenanforderungen geprüft werden. Für das Management ergibt sich hierdurch eine „ideale Möglichkeit **Standardisierungspotenziale aufzudecken**“<sup>34</sup> und das bestehende Portfolio „bedarfsgerecht [zu] erweitern“<sup>35</sup>. Gerade die Identifikation technischer Lücken stellt durch die Nähe zur Technologieentwicklung innerhalb der

---

<sup>30</sup> Interviewpartner 26 (Management, Linie), 22 (Entwicklung, System)

<sup>31</sup> Interviewpartner 21 (Entwicklung, System), 18 (Entwicklung, System)

<sup>32</sup> Interviewpartner 24 (Management, Linie), 16 (Entwicklung, System)

<sup>33</sup> Interviewpartner 4 (Entwicklung, Hardware), 12 (Entwicklung, Software), 20 (Entwicklung, System), 23 (Management, Linie), 28 (Management, Programm/Projekt)

<sup>34</sup> Interviewpartner 24 (Management, Linie), 29 (Management, Programm/Projekt)

<sup>35</sup> Interviewpartner 23 (Management, Linie), 27 (Management, Programm/Projekt)

Untersuchungsumgebung einen „echten Mehrwert für Management und Entwicklung“<sup>36</sup> dar. Hierbei wird im Sinne der liegenden Acht nach Albers, Ebel & Lohmeyer (2012) die Wissensbasis genutzt um eine **Konkretisierung Zielsystems**, in diesem Falle hinsichtlich benötigter Bausteine, zu erreichen. Aus Sicht der Modulentwicklung bietet die Übersicht der einzelnen Bausteingrenzen und deren Auswirkung auf die Parametrierung einer Produktkombination eine „bedienerfreundliche Unterstützung“<sup>37</sup> die als „brauchbare Indikation benötigter Bausteine“<sup>38</sup> genutzt werden kann. Zusätzlich könnten über die bereitgestellte Vorauswahl von Bausteinen zukünftig „**neue Technologien** auf deren Eignung im Baukasten geprüft werden“<sup>39</sup>. Hiermit kann direkt auf die Vorteile der PGE im Falle einer durchgängigen aktiven Nutzung entlang der Phasen der Forschung und Entwicklung geschlossen werden.

Neben der Identifikation technischer Lücken kann die Baukastenanwendung auch dazu genutzt werden **Aussagen über nicht enthaltene Funktionen** zu generieren. Dies bietet laut der befragten Stakeholder ebenfalls eine „gelungene Möglichkeit die Roadmap- und Technologieplanung zu unterstützen“<sup>40</sup>. Bereits über die, automatisch auf das zugrundeliegende Baukastenmodell angepasste Eingabemaske können die funktionalen Abweichungen ausgemacht werden.

## 6.4 Diskussion der Ergebnisse

Im Rahmen der Validierung wurde geprüft inwiefern die ermittelten Einflussfaktoren der deskriptiven Studie I durch die, in der präskriptiven Studie realisierten unterstützenden Maßnahmen adressiert werden konnten. Hierzu wurden die Zusammenhänge zunächst in spezifischen auf das Forschungsvorhaben konkretisierten Impact Models dargestellt (Kapitel 6.1). Mit den Impact Models konnte ein detaillierter Überblick der komplexen Einflüsse innerhalb der Untersuchungsumgebung der vorliegenden Arbeit gezeichnet werden. Gleichzeitig wurde durch die Einführung messbarer Erfolgsfaktoren eine, auf den Zeithorizont des Forschungsprojektes ausgerichtete Validierungsgrößen hergeleitet. Auf Basis der Impact Models wurden geeignete, im Kontext der menschzentrierten

---

<sup>36</sup> Interviewpartner 24 (Management, Linie)

<sup>37</sup> Interviewpartner 8 (Entwicklung, Hardware)

<sup>38</sup> Interviewpartner 7 (Entwicklung, Hardware), 9 (Entwicklung, Hardware)

<sup>39</sup> Interviewpartner 8 (Entwicklung, Hardware)

<sup>40</sup> Interviewpartner 23 (Management, Linie), 24 (Management, Linie)



Methodenentwicklung, nutzbare Validierungsmaßnahmen abgeleitet (Kapitel 6.2). Die Ergebnisse der unterschiedlichen Validierungsmaßnahmen wurden entsprechend der zugrundeliegenden Teilforschungsfragen dargelegt (Kapitel 6.3):

*FF3.1: Welchen Beitrag leistet die Anwendung der MBSE-gestützten Strukturierungsmethode für Baukästen mechatronischer Steuergeräte in der Frühen Phase der PGE für die Entwickler?*

*FF3.2: Wie kann der Entwicklungsprozess mechatronischer Steuergeräte mittels MBSE-gestützter Methoden zur Baukastenentwicklung in der Frühen Phase der PGE effizient unterstützt werden?*

*FF3.3: Welcher Zusatznutzen entsteht durch die aktive Nutzung der Baukastenentwicklung zur Identifikation potenzieller funktionaler und technischer Lücken?*

Aus den Validierungsergebnissen lässt sich schließen, dass mit den im Rahmen der vorliegenden Arbeit entwickelten Methoden zur Baukastenstrukturierung und -anwendung eine **akzeptierte und effiziente Unterstützung von Management und Entwicklung** umgesetzt wurde. Der mittels Baukastenentwicklung **geschaffene Effizienzzuwachs** konnte als bedeutend ermittelt werden und kann durch überschaubare Anpassungen noch einen weiteren Ausbau erfahren. Über die vielfältigen Validierungsmaßnahmen, die in Form von Workshops durchgeführt wurden, konnten die innerhalb der Impact Models dargestellten Zusammenhänge detailliert betrachtet werden. So konnte die Eignung der Methoden für **die Frühe Phase der mechatronischen Steuergeräte** widerspruchsfrei nachgewiesen werden. Von zentraler Bedeutung ist an dieser Stelle, das zusammen mit den Experten entwickelte **Eigenschafts-Abhängigkeits-Netzwerk**. Dieses bietet über die Nutzung innerhalb der Baukastenentwicklung hinaus einen großen Mehrwert hinsichtlich eines **einheitlichen Systemverständnisses**. Durch eine Vielzahl an Praxistests ist gleichzeitig **die Nachvollziehbarkeit der Methode**, sowie deren **Akzeptanz in Abhängigkeit der Modellierungsaufwände** nachgewiesen. Die hohe Qualität der Produktempfehlungen aus dem Baukasten macht eine Bearbeitung von Kundenanfragen durch verschiedene Stakeholder unterschiedlicher Expertisegrade möglich. Die Modellierungsaufwände werden durch den gezielten Einsatz von helper-plugins auf ein erträgliches Maß reduziert. Die **bedienerfreundliche Implementierung** der Baukastenentwicklung leistet zusätzlich einen großen Beitrag zur Akzeptanz der entwickelten Methoden. Die zusätzlich ermöglichte Identifikation potenzieller Lücken im Baukasten, lässt einen bedeutenden Beitrag für die Technologieplanung erahnen. Durch die erfolgreiche Anwendung der entwickelten Methoden auf Antriebe außerhalb der elektrischen

Kleinantriebe im Automobilbereich, konnten darüber hinaus erste Indizien auf die **Übertragbarkeit des Ansatzes** gesammelt werden.

## 7 Zusammenfassung und Ausblick

In diesem Kapitel werden zunächst, die im Zuge der vorliegenden Arbeit generierten Ergebnisse und die daraus abgeleiteten Erkenntnisse, rekapituliert und aggregiert. Anschließend wird ein Ausblick auf mögliche, an die Ergebnisse der Arbeit anknüpfende Forschungsvorhaben, gewährt.

### 7.1 Zusammenfassung

In der vorliegenden Arbeit wurden menschenzentrierte, MBSE-gestützte Methoden zur Strukturierung und Anwendung von Baukästen in der Frühen Phase der PGE mechatronischer Steuergeräte entwickelt. Die entwickelten Methoden wurden im Rahmen eines Begleitprojektes, in einer Vorausentwicklungsabteilung für mechatronische Steuergeräte implementiert und anschließend, entsprechend ihrem Einsatz in der Untersuchungsumgebung validiert. Im Rahmen einer Literaturrecherche zur Klärung des Forschungsgegenstandes wurde zunächst Aufschluss über den relevanten Stand der Forschung gegeben. Auf Basis der identifizierten Forschungslücken wurde der Forschungsbedarf abgeleitet.

Aus der Betrachtung der wissenschaftlichen Grundlagen sind die individuellen Herausforderungen im Entwicklungsprozess mechatronischer Steuergeräte identifiziert worden: Diese sind entsprechend der Entwicklungsschwerpunkte geprägt von funktionsdominierten Denkweisen der Software- und Hardwareentwicklung. Das führt unter anderem zu einer nachgelagerten Betrachtung der Aufbau- und Verbindungstechnik. Aufgrund der hohen, häufig negativ emergenten, interdisziplinären Wechselwirkungen ergeben sich langwierige Iterationsschleifen und eine deutliche Zunahme der Komplexität in der Entwicklung. Die Komplexität der Produkte selbst, wird durch technologische Trends wie etwa den der Hochintegration elektronischer Funktionsträger und die vom Markt geforderte Flexibilität weiter verstärkt. Ergebnis ist die Ausprägung sogenannter Mischformen modularer und integraler Produktstrukturanteile, die sich in Abhängigkeit der Anforderungslage zu unterschiedlichsten technischen Realisierungen ergeben. Weiterhin führen die kurzen technologischen Entwicklungszyklen der beteiligten Domänen, allen voran in den Bereichen der Halbleitertechnik und der Softwareentwicklung, zu einer anhaltenden Dynamisierung. Der Entwicklungsprozess selbst, orientiert sich fast ausschließlich an sequenziellen Prozessmodellen, die im Bereich der funktionalen Sicherheit durch

zusätzliche Anforderungen an die Nachverfolgbarkeit und Durchgängigkeit der Dokumentation ergänzt wurden.

Da die Baukastenentwicklung im Kontext der PGE-Produktgenerationsentwicklung als kontinuierliche, den Produktentwicklungszyklen begleitende Aktivität zu betrachten ist, gilt es Ansätze zu finden, die den spezifischen Herausforderungen mechatronischer Steuergeräte Rechnung tragen. Aus dem Kanon der KaSPro – Karlsruher Schule für Produktentwicklung konnte ein, im Bereich der Frühen Phase geeigneter, MBSE-gestützter Ansatz zur Baukastenentwicklung identifiziert werden. Dieser stellt die Baukastenentwicklung in Form eines Makroprozesses als Kreislauf aus Induktion und Deduktion von (Referenz-) Modellen dar. Um die Spezifika der mechatronischen Steuergeräte allerdings zielgerichtet adressieren zu können, bedarf es einer detaillierteren Auseinandersetzung mit den Prozessen der Induktion und Deduktion. Beispielsweise stellen die kurzen Entwicklungszyklen neuer Technologien deutlich tiefgreifendere Anpassungen des Baukastens in einer deutlich höheren Frequenz dar, als dies sonst im Automobilsektor der Fall ist. Gleichzeitig müssen Wege gefunden werden die Vielzahl negativ-emergenter Wechselwirkungen in einer für, den Einsatz innerhalb der Frühen Phase verträglichen Dosis, bei der Erstellung ganzheitlicher Modelle zu berücksichtigen. Die hohen Modellierungsaufwände bei der Erstellung konsistenter Systemmodelle sind nach wie vor ein großes Hemmnis für die flächendeckende Akzeptanz des MBSE-Ansatzes in der industriellen Praxis. Potenziale ergeben sich an dieser Stelle aus dem maßvollen Einsatz von (Teil-)Automatisierung. Der ermittelte Forschungsbedarf wurde anschließend kondensiert und in die folgende Zielformulierung des Forschungsvorhabens überführt:

### **Ziel der Arbeit**

Im Rahmen dieser Arbeit wird das Ziel verfolgt, die Strukturierung und Anwendung von Baukästen in der Frühen Phase der PGE – Produktgenerationsentwicklung von mechatronischen Steuergeräten zu unterstützen, die Akzeptanz der Methodenanwendung des MBSE zu steigern und eine Zeitreduktion bis zur initialen Konzeptauswahl in der Frühen Phase der PGE zu erwirken.

Im Nachgang wurde das dargelegte Forschungsziel anhand dreier Forschungsfragen operationalisiert. Diese wurden im Zuge der vorliegenden Arbeit beantwortet:

**Forschungsfrage 1:**

*FF1: Welche Spezifika und Bedarfe im Bereich der Strukturierung und Anwendung von Baukästen existieren in der Frühen Phase der PGE von mechatronischen Steuergeräten und welches Zielsystem an entsprechende Methoden lässt sich daraus ableiten?*

**Forschungsfrage 2:**

*FF2: Wie sind die Strukturierung und Anwendung eines Baukastens mechatronischer Steuergeräte auf Basis des ermittelten Zielsystems, unter Verwendung der Methoden des MBSE zu gestalten und zu integrieren?*

**Forschungsfrage 3:**

*FF3: Welchen Beitrag leisten die Methoden zur Baukastenstrukturierung und -anwendung in der Frühen Phase der PGE von mechatronischen Steuergeräten zur Unterstützung der beteiligten Domänen und zur Steigerung der Effizienz im Entwicklungsprozess?*

Die wissenschaftliche Vorgehensweise zur Beantwortung der Forschungsfragen wurde unter Verwendung eines, an die Forschungsmethode der DRM – Design Research Methodology von Blessing & Charkrabarti (2009), angelehnten Ansatzes realisiert (Abbildung 7.1). Entsprechend des aufeinander aufbauenden Charakters der Methoden zur Baukastenstrukturierung und -anwendung wurden diese sequenziell in den Kapiteln 4 und 5 abgearbeitet. Hierzu wurde für jede der Methoden zunächst eine umfassende deskriptive Studie I in Form von teilnehmenden Beobachtungen und teilstrukturierten Interviews durchgeführt. Die Ergebnisse der Studien wurden in sogenannten Reference Models fixiert. Anschließend wurden auf Basis der ermittelten Elemente des Zielsystems zweigeteilte, präskriptive Studien zur Entwicklung und Implementierung der entsprechenden Methoden durchgeführt. Die im Rahmen, des als Entwicklung bezeichneten Teils der präskriptiven Studie, vorgestellten theoretischen Betrachtungen wurden dabei weitestgehend unabhängig von den Randbedingungen der Untersuchungsumgebung gestaltet. Die in den Implementierungskapiteln umgesetzte praktische Anwendung der Methoden hingegen, wurde auf die Bedürfnisse und die tooltechnischen Vorgaben des Begleitprojektes optimiert.

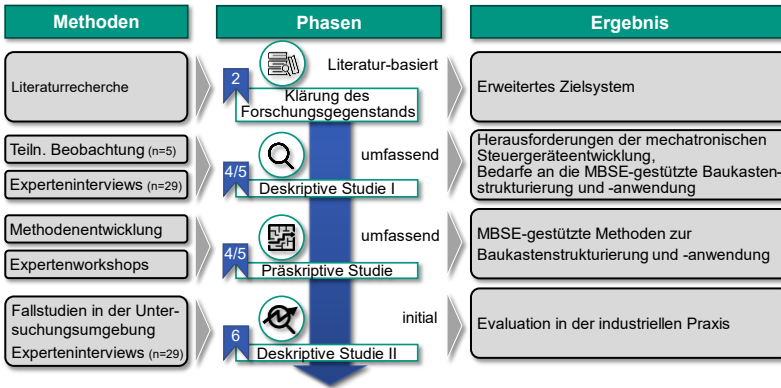


Abbildung 7.1: Vorgehensweise zur Beantwortung der Forschungsfragen in Anlehnung an die DRM (Blessing & Chakrabarti, 2009).

Für die Methode der Baukastenstrukturierung ist ein funktionszentrierter Ansatz gewählt worden, um die Vergleichbarkeit der unterschiedlichen integralen und modularen Bauweisen der zugrundeliegenden Funktionsträger zu ermöglichen. Über die Einführung von Funktionsclustern und Funktionsstrukturausschnitten konnte ein Schulderschluss zwischen der im V-Modell geforderten lösungsneutralen Funktionsbeschreibung und deren konkreten Umsetzung in Produkten hergestellt werden. Dies entspricht eindeutig dem von Albers & Wintergerst (2014) vertretenen praxisnahen Ansatz des Funktions-Gestalt-Übergangs. Zur Unterstützung des Entwicklers bei der vielfach diskutierten Frage nach einer geeigneten Strukturierungstiefe von Baukästen wurden klare Strukturierungskriterien geschaffen. Für den Einsatz in der Frühen Phase der Produktentwicklung sind relevante Key-Parameter zusammen mit den verantwortlichen Experten für die Module ermittelt worden. Diese wurden anschließend in einem Eigenschafts-Abhängigkeits-Netzwerk, welches einer abstrakten Darstellung eines von Albers im C&C<sup>2</sup>-A beschriebenen Wirk-Netztes gleicht, dargestellt (Albers & Wintergerst, 2014). Die über die Methode vergleichbaren modularen und integralen Bausteinvarianten wurden durch Informationen aus den realen Entwicklungsprozessen ergänzt. Somit kann während der späteren Anwendung jederzeit auf wertvolle Informationen aus Referenzsystemen zugegriffen werden. Im Rahmen der Implementierung konnte durch den Einsatz spezifischer helper-plugins, der Modellierungsaufwand nicht nur signifikant verringert werden, es wurde darüber

hinaus auch eine geeignete Methode zur Überführung von Inhalten aus dem dokumentenzentrierten Arbeiten in ein durchgängiges Systemmodell aufgezeigt.

Die Schwerpunkte bei der Entwicklung der Methode der Baukastenanwendung lagen auf der Gestaltung der Kompatibilitätsprüfung, sowie der benutzerfreundlichen Umsetzung einer Toolkette zur automatisierten Ausleitung von Produktempfehlungen. Bei der Kompatibilitätsprüfung wurden zunächst die interne und externe Kompatibilität unterschieden. Die interne Kompatibilität behandelt den Abgleich der Schnittstellen zwischen zwei Modulen und deren dahinterliegenden Bausteinvarianten. Auf Basis der internen Kompatibilitätsbetrachtung werden, aus den im Baukasten enthaltenen integralen Bausteinvarianten, zunächst modulare Bausteinvarianten und später Bausteinkombinationen für den Funktionsumfang eines Steuergerätes erstellt und evaluiert. Ausschlaggebend für die Betrachtung der Modulschnittstellen sind dabei die im EAN identifizierten Key-Parameter und deren Verbindungen. Mittels externer Kompatibilitätsbetrachtung können im Nachgang die von der Baukastenanwendung als kompatibel ermittelten Kombinationen, mit den vom Kunden an ein gewünschtes Produkt gestellten Anforderungen, verglichen werden. Zur Betrachtung der unterschiedlichen Attribute hinsichtlich deren Kompatibilität wurden verschiedene Kompatibilitätstypen und Auswahlkriterien entwickelt. Im Rahmen der Implementierung wurde eine, auf dem föderierten Ansatz beruhende Toolkette zur Erfüllung der Stakeholder Use-Cases bereitgestellt. Diese ermöglicht es, basierend auf Kundenanforderungen, passende Bausteinkombinationen zu identifizieren und hinsichtlich gewisser KPI's zu filtern. Darüber hinaus kann die Baukastenanwendung aktiv dazu genutzt werden, Bausteine auszumachen, die für die Nicht-Erfüllung von Kundenanforderungen sind. Auf diesem Wege können neue Standardisierungspotenziale aufgedeckt und die Entwicklung neuer Module zielgerichtet unterstützt werden.

Die Validierung der entwickelten Methoden wurde im Rahmen einer initialen deskriptiven Studie II durchgeführt. Hierzu wurden die ermittelten Bedarfe an die einzelnen Methoden der realisierten Unterstützung gegenübergestellt und deren tatsächlicher Beitrag zur Verbesserung der ingenieurstechnischen Entwicklungspraxis anhand messbarer Erfolgskriterien evaluiert. Über eine Vielzahl, vom Autor dieser Arbeit durchgeführter Interviews und Praxistests konnte die Akzeptanz der Methoden und die herbeigeführte Effizienzsteigerung durch das Management und die Entwickler innerhalb der Untersuchungsumgebung eindeutig belegt werden. Durch die Anwendung des Baukastens waren gerade die, in der Frühen Phase der Produktentwicklung mechatronischer Steuergeräte herrschenden Herausforderungen zielgerichtet zu adressieren. Mit dem Eigenschafts-Abhängigkeits-Netzwerk wurde über die Baukastenentwicklung hinaus ein

Kommunikationsinstrument geschaffen, welches die Einheitlichkeit des Systemverständnisses in der interdisziplinären Produktentwicklung nachweislich erhöht. An dieser Stelle ergeben sich vielzählige Möglichkeiten simulative Ansätze mit der Systemarchitektur zu koppeln<sup>1</sup>. Durch die teilautomatisierte Anwendung des Baukastenmodells und die hohe Qualität der resultierenden Produktempfehlungen konnte ein größerer Anwenderkreis dazu befähigt werden, fundierte Aussagen über die Eignung von Produktkonzepten zu treffen. Die ausgewählten Produktkonzepte können somit, durch die beschleunigte Aussage in Form von Funktionsmustern schneller der frühzeitigen Validierung zugeführt werden. Der gezielte Einsatz von helper-plugins wurde als wichtiger Schritt zur Schaffung von Akzeptanz von MBSE-Ansätzen in der industriellen Praxis identifiziert. Die mittels Baukastenanwendung ermöglichte Identifikation potenzieller Lücken im Baukasten, lässt erahnen, welche Potenziale in der aktiven, teilautomatisierten Nutzung von Referenzsystemen im Sinne des Modells der PGE (Albers, Rapp et al., 2019), liegen. Dies ist besonders hinsichtlich der zukünftigen Entwicklung von Advanced Systems ein bedeutender Faktor.

Innerhalb der Validierung sind weiterhin erste Indizien hinsichtlich der Übertragbarkeit des Ansatzes auf eine zusätzliche Klasse elektrischer Antriebe aufgezeigt worden. Viele weitere mechatronische Systeme weisen ähnliche Spezifika im Entwicklungsprozess, wie beispielsweise eine Pfadabhängigkeit hinsichtlich der Softwareentwicklung, dynamische Entwicklungszyklen der beteiligten Domänen und einen eindeutigen Trend zur Hochintegration, auf. Es ist daher naheliegend, dass sich die entwickelten Methoden für ein deutlich größeres Spektrum mechatronischer Produkte zum Einsatz eignen. Dieses und weitere im Rahmen dieser Arbeit adressierte Themenfelder bergen große Potenziale für den Aufsatz weiterer Forschungsprojekte. Der folgende Ausblick soll die Aufmerksamkeit auf jene Stellen richten.

## 7.2 Ausblick

Im vorliegenden Abschnitt werden, auf Basis der erzielten Erkenntnisse sinnvolle Anknüpfungspunkte für weitere Forschungsvorhaben beleuchtet. Wie bereits in der Validierung der entwickelten Methoden angeklungen ist, bietet es sich an weitere Untersuchungen hinsichtlich der Übertragbarkeit der Ansätze auf mechatronische Systeme mit verwandten Herausforderungen, anzustellen. Neben der Steigerung

---

<sup>1</sup> Die daraus entstehenden Forschungsbedarfe werden im Ausblick erörtert.



des einheitlichen Systemverständnisses konnten bereits Potentiale in der Verkürzung der Zeit bis zur physischen Validierung von Produktkonzepten und dem damit verbundenen Effizienzgewinn in der Frühen Phase aufgedeckt werden. An dieser Stelle kann beispielsweise eine Einbindung der Ergebnisse zukunftssträchtiger Forschungsfelder wie der AR-gestützten Validierung betrachtet werden (Albers, Reinemann et al., 2019). Darüber hinaus erscheint eine Erweiterung des Ansatzes auf die Ebene des Gesamtantriebs sinnvoll. Hierzu wären die bisherigen Untersuchungen, um die Betrachtung verschiedener Typen von Elektromotoren zu ergänzen (vgl. Kapitel 6.3). Als Ergebnis dieser Erweiterung ist bereits jetzt eine drastische Reduktion der Anzahl der unternehmensinternen Schnittstellen für die Erstbewertung der Kundenanfragen abzusehen.

Die im Kern der Baukastenanwendung befindlichen Kompatibilitätsbetrachtungen bieten ebenfalls breitgefächerte Möglichkeiten zur Weiterführung der entwickelten Ansätze. Das an dieser Stelle ein hohes wissenschaftliches Interesse besteht, konnte unter anderem durch den Gewinn des GfSE-Preises<sup>2</sup> durch die betreute Abschlussarbeit von Zanger (2018) bestätigt werden. Die innerhalb der vorliegenden Arbeit präsentierte Kompatibilitätsbetrachtung kann beispielsweise um eine Priorisierung von Modulen und deren Bausteine bei der Kombinationsbildung erweitert werden. Zentrale Komponenten wie etwa der Mikrocontroller haben in Bezug auf das resultierende Produktkonzept deutlich höhere Auswirkungen wie beispielsweise der funktionsgleiche Austausch eines Sensorbausteins (vgl. Kapitel 6.3). Die Einführung einer Matching Range nach Ishii et al. (1988) birgt ebenfalls große Potenziale (vgl. Kapitel 5.4). Die automatisierte Kompatibilitätsbetrachtung scheint gerade im Zuge der zunehmenden Vernetzung der Produktwelt zu komplexen Systems-of-Systems, eine mögliche Schlüsselrolle in der Entwicklung von Advanced Systems einnehmen zu können. Auch hier bedarf es Ansätzen, die auf dem grundlegenden Modell der PGE aufbauen, wie bereits durch Albers, Peglow et al. (2018) gezeigt werden konnte. Bei der Betrachtung der hochvernetzten Systeme stellt die Abbildung des Systemverhaltens eine große Herausforderung dar. Zukünftige Forschungsarbeiten können an dieser Stelle auf den Ergebnissen zur Integration komplexer Verhaltensmodelle in ein initiales Baukastenmodell, auf Basis des Eigenschafts-Abhängigkeits-Netzwerkes, aufbauen.

---

<sup>2</sup> Dies bezieht sich auf die Endrunde des GfSE-Studentenpreises am 10.11.2017 die Zanger mit dem Thema „Entwicklung einer Methode zur Kompatibilitätsbewertung von Bausteinen eines Baukastens, am Beispiel mechatronischer Systeme“ für sich entscheiden konnte

Das Eigenschafts-Abhängigkeits-Netzwerk und die damit verbundene Identifikation von Key-Parametern, wurde als vermeintlicher Potenzialträger hinsichtlich der Erweiterbarkeit der präsentierten Ansätze über die Frühe Phase hinaus ausgewiesen. Zusammen mit den Vorarbeiten von Scherer (2016) in Bezug auf die Unterstützung der Serienentwicklung von Baukästen bieten sich Ansatzpunkte für die Forschung im Bereich menschenzentrierter Methoden.

Infolge der zunehmenden Bedeutung der funktionalen Sicherheit im Rahmen der Entwicklung mechatronischer Steuergeräte bietet sich ein weiterer Forschungsbedarf für die Erweiterung des Ansatzes. Dementsprechend müssen Wege gefunden werden, wie die Produktkonzepte entsprechend der festgelegten ASIL-Stufen in puncto funktionaler Redundanz ergänzt werden können. Durch die Diskussion verschiedener Baukastenarchitekturen und deren kontinuierlicher Weiterentwicklung in Generationen wurde zusätzlicher Nährboden für die wissenschaftliche Auseinandersetzung geschaffen. An dieser Stelle wird es durch verschiedene Ausprägungen der Baukastenarchitektur möglich, dass Module gleichzeitig in verschiedenen Baukastenmodellen berücksichtigt werden. Für die Weiterentwicklung jener Module entsteht daraus eine deutlich höhere Komplexität. Ein möglicher Erkenntnisgewinn über den zielgerichteten Umgang mit dieser Situation kann über ein aktuelles Forschungsprojekt in Kooperation mit einem Werkzeugmaschinenhersteller erwartet werden (Powelske et al., 2017; Albers, Peglow et al., 2018; Peglow et al., 2017).

Wird die Baukastenentwicklung konsequent weitergedacht, darf diese nicht bei der Betrachtung der Produktportfolien enden, um die vollen wirtschaftlichen Potenziale zu entfalten. Grundlage hierfür liefern die Ergebnisse der aktuellen Forschung des IPEK zum Produkt-Produktionssystem-Co-Design (Albers, Stürmlinger et al., 2021; Mandel et al., 2020; Stürmlinger et al., 2020). Diese beschreiben die enge Verzahnung bei der Entwicklung von Produkten und den zugehörigen Produktionssystemen und zeigen Wege auf, wie eine integrierte Betrachtung beider Systeme erfolgreich erfolgen kann. In der vorliegenden Arbeit wurde eine Klasse mechatronischer Steuergeräte behandelt, die sich über eine starke Verknüpfung zu den Prozessen der Fertigung auszeichnet (Kapitel 2.1.2). In diesem Zusammenhang stellen sich im Bereich der Baukastenentwicklung z.B. folgende Fragen, die eine Erweiterung der entwickelten Methoden hinsichtlich des Produktionssystems erlauben: Welche Synergien lassen sich über verschiedene Elemente/Produktvarianten hinsichtlich der Nutzung vergleichbarer Produktionsanlagen nutzen? Welche Änderung im Produkt führt zu welcher Änderung im Produktionssystem und umgekehrt?

Im Bereich des MBSE ist durch die vorliegende Arbeit ein Beitrag zur Steigerung der Akzeptanz in der industriellen Praxis geleistet worden. Mit der Nutzung von helper-plugins konnten die Modellierungsaufwände signifikant gesenkt werden. Gerade Unternehmen, die hinsichtlich der Einführung von MBSE noch am Anfang stehen, kann der Umstieg von der dokumentenzentrierten Denkweise auf konsistente Systemmodelle durch die helper-plugins erheblich erleichtert werden (Powelske et al., 2018). Trotz der bereits bekannten Potenziale steckt die Forschung auf diesem Gebiet noch immer in den Kinderschuhen. In Bezug auf die Entwicklung von Advanced Systems ist abzusehen, dass auch die Modellerstellung zukünftig einen deutlich höheren Anteil an Automatisierung, beispielsweise durch den Einsatz von KI erfahren wird (Dumitrescu et al., 2021). Für die Baukastenentwicklung liegt an dieser Stelle eine große Chance, in der gezielten Nutzung von Machine learning zur Bereitstellung von Informationen aus Referenzsystemen entsprechend dem Modell der PGE. Als Resultat könnten die Ergebnisse der Produktempfehlungen immer weiter verbessert und zusätzlich die Vision einer gewissen Selbstoptimierung des Baukastens vorangetrieben werden. Generell zeigen die Ergebnisse der vorliegenden Arbeit auf, dass die aktive Nutzung von Entwicklungsdaten aus Referenzprojekten einen großen Einfluss hinsichtlich der Aufwands- und Risikominimierung in der Produktentwicklung hat. Vor dem Hintergrund der zunehmenden Digitalisierung und Vernetzung bedarf es zielführender Methoden zur Nutzung des Wissens aus Referenzsystemen, gerade in Fällen, in denen dieses Wissen nicht explizit vorliegt. Erste Ansätze zur weiteren Forschung auf diesem Gebiet finden sich beispielsweise bei Klippert et al. (2020). Da sich die, in der vorliegenden Arbeit erfolgte Implementierung der Methoden eng an den Rahmenbedingungen der Untersuchungsumgebung allen voran der verwendeten Toollandschaft orientiert, muss eine Übertragung der Ansätze auf die Toolumgebung anderer Hersteller geprüft werden. In Zusammenhang mit den in diesem Abschnitt genannten Anknüpfungspunkten wissenschaftlicher Arbeiten, sei auf das vom IPEK mitgestaltete BMBF-Verbundprojekt MoSyS - Mensorientierte Gestaltung komplexer Systems of Systems verwiesen (IPEK, 2020; Frauenhofer IEM, 2020). Dieses umfasst beispielsweise Ansätze des musterbasierten Architekturmanagements, des KI-unterstützten Engineerings und des Einsatzes von digitalen Zwillingen in der Produktentwicklung vor dem Hintergrund des soziotechnischen Systems.

Über den direkten Ergebnistransfer und die Weiterentwicklung der Systematik hinaus sollte die Übertragbarkeit der entwickelten Methoden auch auf andere Domänen evaluiert werden. Im Rahmen eines durch die Deutsche Forschungsgesellschaft (DFG) geförderten Teilprojekts des Schwerpunktprogramms SSP2187, wird die intelligente Modularisierung für den skalierbaren Betonbau durch Adaption der Methoden zur Baukastenentwicklung

untersucht. Bereits in der Antragsstellung sind die bisherigen Kenntnisse der MBSE-gestützten Baukastenentwicklung der KaSPro – Karlsruher Schule der Produktentwicklung miteingeflossen. Auf diesem Gebiet bieten sich demnach große Potenziale zur Integration der im Rahmen dieser Arbeit entwickelten Ansätze.

# Literaturverzeichnis

- Abouda, K., Besse, P., Clairet, M., El Sherif, A., Hemon, E., Lopez, D., ... & Turpin, P. (2021). How System Basis Chips rise to the power and reliability challenges for future-proof electrification and autonomous vehicle electronic control unit designs. In *33rd International Symposium on Power Semiconductor Devices and ICs (ISPSD)* (S. 17–22). Nagoya, Japan. <https://doi.org/10.23919/ISPSD50666.2021.9452239>
- Albers, A. (1994). Simultaneous Engineering, Projektmanagement und Konstruktionsmethodik Werkzeuge zur Effizienzsteigerung. In VDI (Hrsg.), *VDI Berichte: Bd. 1120. Entwicklung und Konstruktion im Strukturwandel: Deutscher Konstrukteurtag* (S. 73-106). Düsseldorf: VDI-Verlag.
- Albers, A. (2003). Produktentwicklung - Heute und Morgen. *Konstruktion*, 55(11/12), 3–5.
- Albers, A. (2008). Einführung in Antriebssysteme. In W. Steinhilper & B. Sauer (Hrsg.), *Springer Lehrbuch. Konstruktionselemente des Maschinenbaus 2* (S. 245-278). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-540-76654-4\\_4](https://doi.org/10.1007/978-3-540-76654-4_4)
- Albers, A. (2010). Five Hypotheses about Engineering Processes and their Consequences. In *Proceedings of the 8th International Symposium on Tools and Methods of Competitive Engineering (TMCE)* (S. 343–356). Ancona, Italien.
- Albers, A. (2011). *Der Entwickler im Zentrum des Systems der Produktentstehung: Integriertes Produktentstehungsmodell & X-in-the-Loop-Framework* [Konferenzbeitrag]. Stuttgarter Symposium für Produktentwicklung (SSP), Stuttgart, Deutschland.
- Albers, A., Behrendt, M., Klingler, S. & Matros, K. (2016). Verifikation und Validierung im Produktentstehungsprozess. In U. Lindemann (Hrsg.), *Handbuch Produktentwicklung* (S. 541–569). München: Carl Hanser Verlag.

- Albers, A., Behrendt, M., Klingler, S., Reiß, N. & Bursac, N. (2017). Agile product engineering through continuous validation in PGE - Product Generation Engineering. *Design Science*, 3(5), 19. <https://doi.org/10.1017/dsj.2017.5>
- Albers, A., Behrendt, M., Schroeter, J., Ott, S. & Klingler, S. (2013). X-in-the-Loop: A framework for supporting central engineering activities and contracting complexity in product engineering processes. In U. Lindemann, S. Venkataraman, Y. Kim, S. W. Lee, J. Clarkson & G. Cascini (Hrsg.), *Proceedings of the 19th International Conference on Engineering Design (ICED13)* (S. 391-400). Seoul, Korea.
- Albers, A. & Braun, A. (2011a). A generalised framework to compass and to support complex product engineering processes. *International Journal of Product Development*, 15(1/2/3), 6. <https://doi.org/10.1504/IJPD.2011.043659>
- Albers, A. & Braun, A. (2011b). Der Prozess der Produktentstehung. In F. Henning & E. Moeller (Hrsg.), *Handbuch Leichtbau. Methoden, Werkstoffe, Fertigung* (S. 3–30). München: Carl Hanser Verlag.
- Albers, A., Braun, A., Enkler, H.-G., Wynn, D. & Clarkson, P. J. (2009). Contact and Channel Modelling to support the design of technical systems. In M. Norell Bergendahl, M. Grimheden, L. Leifer, P. Skogstad & U. Lindemann (Hrsg.), *Proceedings of the 17th International Conference on Engineering Design (ICED09)* (S. 61-72). Palo Alto, USA.
- Albers, A., Braun, A. & Muschik, S. (2010). Uniqueness and the Multiple Fractal Character of Product Engineering Processes. In P. J. Clarkson, P. Heisig & S. Vajna (Hrsg.), *Modelling and Management of Engineering Processes* (S. 15–26). London: Springer. [https://doi.org/10.1007/978-1-84996-199-8\\_2](https://doi.org/10.1007/978-1-84996-199-8_2)
- Albers, A., Burkardt, N., Meboldt, M. & Saak, M. (2005). SPALTEN Problem Solving Methodology in the Product Development. In A. Samuel & W. Lewis (Hrsg.), *Proceedings of the 15th International Conference on Engineering Design (ICED05)* (S. 1–12). Melbourne, Australien.
- Albers, A., Bursac, N. & Rapp, S. (2017). PGE – Produktgenerationsentwicklung am Beispiel des Zweimassenschwungrads. *Forschung im Ingenieurwesen*, 81(1), 13–31. <https://doi.org/10.1007/s10010-016-0210-0>

- Albers, A., Bursac, N. & Scherer H. (2014). Modelbased Systems Engineering zur Unterstützung der Baukastenentwicklung. In *Newsletter Wissenschaftliche Gesellschaft für Produktentwicklung WiGeP*, 7-8, 20–21.  
<https://doi.org/10.3139/104.014070>
- Albers, A., Bursac, N., Scherer, H., Birk, C., Powelske, J. & Muschik, S. (2019). Model-based systems engineering in modular design. *Design Science*, 5, Artikel e17. <https://doi.org/10.1017/dsj.2019.15>
- Albers, A., Bursac, N. & Wintergerst, E. (2015). Produktgenerationsentwicklung. Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In *Beiträge zum 3. Stuttgarter Symposium für Produktentwicklung (SSP)* (S. 1–10). Stuttgart, Deutschland.
- Albers, A. & Düser, T. (2010). A New Process for Configuration and Application of Complex Validation Environments Using the Example of Vehicle-in-the-Loop at the Roller Test Bench. In *Proceedings of the ASME 2010 International Mechanical Engineering Congress and Exposition. Volume 11: New Developments in Simulation Methods and Software for Engineering Applications; Safety Engineering, Risk Analysis and Reliability Methods; Transportation Systems* (S. 807–816). Vancouver, British Columbia, Canada.  
<https://doi.org/10.1115/IMECE2010-39959>
- Albers, A., Dumitrescu, R., Marthaler, F., Albers, A., Kühfuss, D., Strauch, M., Siebe, A. & Bursac, N. (2018). PGE – Produktgenerationsentwicklung und Zukunftsvorausschau. Eine systematische Betrachtung zur Ermittlung der Zusammenhänge. In *Beiträge zum 14. Symposium für Vorausschau und Technologieplanung (SVT)* (S. 27–51). Berlin, Deutschland.
- Albers, A., Ebel, B. & Lohmeyer, Q. (2012). Systems of objectives in complex product development. In *Proceedings of TMCE 2012* (S. 267–278). Karlsruhe, Deutschland.
- Albers, A., Fischer, J., Klingler, S. & Behrendt, M. (2014). *Durchgängige Validierung und Verifizierung am Beispiel der akustischen Eigenschaften eines Elektrofahrzeugs*. [Konferenzbeitrag]. Grazer Symposium Virtuelles Fahrzeug (GSVF), Graz, Österreich.
- Albers, A. & Gausemeier, J. (2012). Von der fachdisziplinorientierten Produktentwicklung zur Vorausschauenden und Systemorientierten Produktentstehung. In R. Anderl, M. Eigner, U. Sendler & R. Stark (Hrsg.),

*Smart Engineering. acatech DISKUSSION* (S. 17-29). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-29372-6\\_3](https://doi.org/10.1007/978-3-642-29372-6_3)

- Albers, A., Haug, F., Heitger, N., Arslan, M., Rapp, S. & Bursac, N. (2016). Produktgenerationsentwicklung. Praxisbedarf und Fallbeispiel in der automobilen Produktentwicklung. In *Beiträge zum 12. Symposium für Vorausschau und Technologieplanung (SVT)* (S. 1–15). Berlin, Deutschland.
- Albers, A., Haug, F., Heitger, N., Fahl, J. & Hirschter, T. (2019). Entwicklungsgenerationen zur Steuerung der PGE – Produktgenerationsentwicklung: Von der Bauteil- zur Funktionsorientierung in der Automobilentwicklung. In *Beiträge zum 5. Stuttgarter Symposium für Produktentwicklung (SSP)* (S. 253–262). Stuttgart, Deutschland.
- Albers, A., Heitger, N., Haug, F., Fahl, J., Hirschter, T. & Bursac, N. (2018). Supporting potential innovation in the Early Phase of PGE–Product Generation Engineering: Structuring the development of the Initial System of Objectives. In *R&D Management Conference, 13*. Mailand, Italien.
- Albers, A., Kürten, C., Rapp, S., Birk, C., Hünemeyer, S. & Kempf, C. (2022). *KIT Scientific Working Papers: Bd. 199. SGE – Systemgenerationsentwicklung: Analyse und Zusammenhänge von Entwicklungspfaden in der Produktentstehung*. Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000151151>
- Albers, A. & Lohmeyer, Q. (2012). Advanced Systems Engineering. Towards a Model-Based and Human-Centered Methodology. In *Proceedings of TMCE 2012* (S. 407–416). Karlsruhe, Deutschland.
- Albers, A., Lohmeyer, Q. & Ebel, B. (2011). Dimensions of objectives in interdisciplinary product development projects. In S.J. Culley, B.J. Hicks, T.C. McAloone, T.J. Howard & Y. Reich (Hrsg.), *Proceedings of the 18th International Conference on Engineering Design (ICED11)* (S. 256–265). Kopenhagen, Dänemark.
- Albers, A., Lohmeyer, Q. & Radimersky, A. (2012). Individuelle und organisatorische Akzeptanz von Methoden des Systems Engineering. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering. TdSE 2012* (S. 351-360). München: Carl Hanser Verlag.



- Albers, A., Mandel, C., Yan, S. & Behrendt, M. (2018). System of systems approach for the description and characterization of validation environments. In *Design 2018: proceedings of the 15th International Design Conference, Volume 6* (S. 2799–2810). Dubrovnik, Kroatien.  
<https://doi.org/10.21278/idc.2018.0460>
- Albers, A., Matthiesen, S., Bursac, N., Moeser, G., Klingler, S., Schmidt, S., Munker, F. et al. (2016). Model-Based Systems Engineering (MBSE) in der Karlsruher Schule: Fünf Jahre Forschung für die Anwendung. *develop^3 systems engineering*, 3(2), 38–41.
- Albers, A., Matthiesen, S., Bursac, N., Moeser, G., Schmidt, S. & Lüdcke, R. (2014). Abstraktionsgrade der Systemmodellierung - von der Sprache zur Anwendung. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering. TdSE 2014* (S. 183–192). München: Carl Hanser Verlag.  
<https://doi.org/10.13140/2.1.4350.8007>
- Albers, A., Munker, F., Zingel, C. & Behrendt, M. (2013). Integrative Systemmodellierung von Hardware- und Softwarekomponenten in SysML am Beispiel eines innovativen Datengateways. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering. TdSE 2013* (S. 347-356). München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446439467.033>
- Albers, A., Peglow, N., Powelske, J., Birk, C. & Bursac, N. (2018). Coping with Complex Systems-of-Systems in the Context of PGE–Product Generation Engineering. *Procedia CIRP*, 70, 457-462.  
<https://doi.org/10.1016/j.procir.2018.01.004>
- Albers, A., Rapp, S., Birk, C. & Bursac, N. (2017). Die „Frühe Phase“ der PGE - Produktgenerationsentwicklung. In *Beiträge zum 4. Stuttgarter Symposium für Produktentwicklung (SSP)* (S. 1–10). Stuttgart, Deutschland.
- Albers, A., Rapp, S., Fahl, J., Hirschter, T., Revfi, S., Schulz, M., Stürmlinger, T. & Spadinger, M. (2020). Proposing a generalized description of variations in different types of systems by the model of PGE – Product Generation Engineering. In *Proceedings of the Design Society: DESIGN Conference, Volume 1* (S. 2235-2244). Cavtat, Kroatien.  
<https://doi.org/10.1017/dsd.2020.315>
- Albers, A., Rapp, S., Heitger, N., Wattenberg, F. & Bursac, N. (2018). Reference Products in PGE – Product Generation Engineering: analyzing challenges

based on the system hierarchy. *Procedia CIRP*, 70, 469-474.  
<https://doi.org/10.1016/j.procir.2018.02.046>

- Albers, A., Rapp, S., Spadinger, M., Richter, T., Birk, C., Marthaler, F., Heimicke, J., Kurtz, V. et al. (2019). The Reference System in the Model of PGE. Proposing a Generalized Description of Reference Products and their Interrelations. In S. Wartzack & B. Schleich (Hrsg.), *Proceedings of the 22nd International Conference on Engineering Design (ICED19)* (S. 1693–1702). Delft, Niederlande. <https://doi.org/10.1017/dsi.2019.175>
- Albers, A., Reinemann, J., Fahl, J. & Hirschter, T. (2019). Augmented Reality for Product Validation. Supporting the Configuration of AR-Based Validation Environments. In *Proceedings of 21st HCI International Conference. Part II* (S.429–448). Orlando, FL, USA. [https://doi.org/10.1007/978-3-030-21565-1\\_29](https://doi.org/10.1007/978-3-030-21565-1_29)
- Albers, A., Reiß, N., Bursac, N. & Breitschuh, J. (2016). 15 Years of SPALTEN Problem Solving Methodology in Product Development. In *Proceedings of the 2016 NordDesign Conference* (S. 411–420). Trondheim, Norwegen.
- Albers, A., Reiß, N., Bursac, N. & Richter, T. (2016). iPeM – Integrated Product Engineering Model in Context of Product Generation Engineering. *Procedia CIRP*, 50, 100–105. <https://doi.org/10.1016/j.procir.2016.04.168>
- Albers, A., Reussner, R., Kurrle, A., Burger, E., Moeser, G., Bursac, N., Klingler, S. & Behrendt, M. (2017). Continuity in the development of seamless mobility: An approach for a System-of-Systems environment. *International Research Journal of engineering and Technology (IRJET)*, 4(2), 668-678.
- Albers, A., Scherer, H., Bursac, N. & Rachenkova, G. (2015). Model based systems engineering in construction kit development—two case studies. *Procedia CIRP*, 36, 129-134. <https://doi.org/10.1016/j.procir.2015.01.044>
- Albers, A., Stürmlinger, T., Revfi, S. & Behdinan, K. (2021). Extended target weighing approach (ETWA): Impact and risk analysis of lightweight concepts in the product-production system-co-design. In *Proceedings of the 23rd International Conference on Engineering Design (ICED21)* (S. 1537–1546). Goteborg, Schweden. <https://doi.org/10.1017/pds.2021.415>

- Albers, A., Walch, M. & Bursac, N. (2016). Entscheidungsunterstützung durch die Variationsanteile der Produktgenerationsentwicklung. *Konstruktion*, 68(7-8), 67-74. <https://doi.org/10.37544/0720-5953-2016-07-08-67>
- Albers, A. & Wintergerst, E. (2014). The Contact and Channel Approach (C&C<sup>2</sup>-A). Relating a System's Physical Structure to Its Functionality. In A. Chakrabarti & L. T. M. Blessing (Hrsg.), *An anthology of theories and models of design. Philosophy, approaches and empirical explorations* (S. 151–171). London: Springer. [https://doi.org/10.1007/978-1-4471-6338-1\\_8](https://doi.org/10.1007/978-1-4471-6338-1_8)
- Albers, A. & Zingel, C. (2013). Extending SysML for Engineering Designers by Integration of the Contact & Channel-Approach (C&C<sup>2</sup>-A) for Function-Based Modeling of Technical Systems. In *Proceedings of the Conference on Systems Engineering Research (CSER)* (S. 353-362). Atlanta, USA. <https://doi.org/10.1016/j.procs.2013.01.037>
- Albers, A. & Zingel, C. (2013a). Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML. In M. Abramovici & R. Stark (Hrsg.), *Lecture Notes in Product Engineering. Smart Product Engineering* (S. 83-92). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-30817-8\\_9](https://doi.org/10.1007/978-3-642-30817-8_9)
- Alexander, C., Ishikawa, S. & Silverstein, M. (1977). *A pattern language: towns, buildings, construction*. New York: Oxford university press.
- Alink, T. (2010). Bedeutung, Darstellung und Formulierung von Funktion für das Lösen von Gestaltungsproblemen mit dem C&C-Ansatz. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 48). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000023234>
- Alt, O. (2012). *Modellbasierte Systementwicklung mit SysML*. München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446431270>
- Arslan, M., Bursac, N., Killer, B. & Albers, A. (2018). Developing R&D-Strategies for Future Innovations: Activities in the Context of Technological Transformation and PGE – Product Generation Engineering. In *R&D Management Conference*, 19. Mailand, Italien.

- ASPICE (2017). *Automotive SPICE. Process Reference Model. Process Assessment Model. Version 3.1*. Zugriff am 30.10.2022. Verfügbar unter [http://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE\\_PAM\\_31.pdf](http://www.automotivespice.com/fileadmin/software-download/AutomotiveSPICE_PAM_31.pdf)
- Automobil-Elektronik (2011). *Elektronik im BMW von morgen*. Zugriff am 30.10.2022. Verfügbar unter <https://www.all-electronics.de/wp-content/uploads/migrated/article-pdf/42286/cover-interview-bmw.pdf>
- AUTOSAR (2020). *Classic Platform. AUTOSAR Classic Release R20-11*. Zugriff am 30.10.2022. Verfügbar unter [https://www.autosar.org/fileadmin/user\\_upload/standards/classic/20-11/AUTOSAR\\_TPS\\_SoftwareComponentTemplate.pdf](https://www.autosar.org/fileadmin/user_upload/standards/classic/20-11/AUTOSAR_TPS_SoftwareComponentTemplate.pdf)
- Badke-Schaub, P., Daalhuizen, J., Roozenburg, N. (2011). Towards a Designer-Centred Methodology: Descriptive Considerations and Prescriptive Reflections. In H. Birkhofer (Hrsg.), *The Future of Design Methodology* (S. 181-197). London: Springer.
- Bahill, A. T. & Gissing, B. (1998). Re-evaluating systems engineering concepts using systems thinking, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(4), 516-527. <https://doi.org/10.1109/5326.725338>.
- Bailom, F., Hinterhuber, H. H., Matzler, K. & Sauerwein, E. (1996). Das Kano-Modell der Kundenzufriedenheit. *Marketing: Zeitschrift für Forschung und Praxis*, 18(2), 117-126.
- Baker, N., James, S. & Rubenstein, A. (1967). The Effects of perceived needs on the generation of ideas in R&D labs. *IEEE Transactions on Engineering Management, EM-14*(4), 156-163. <https://doi.org/10.1109/TEM.1967.6446983>
- Bartelt, C., Ternité, T. & Ziegler, M. (2005). Modellbasierte Entwicklung mit dem V-Modell XT. *OBJEKTSpektrum*, 12(5), 54-63.
- Becker, H. (2007). *Auf Crashkurs. Automobilindustrie im globalen Verdrängungswettbewerb*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-540-70982-4>

- Bertalanffy, L. von. (1949). General System Theory. *Biologia Generalis*, 19(1), 114–129.
- Bertalanffy, L. von. (1968). *General System Theory. Foundations, development, applications*. New York: Braziller.
- Binder, A. (2012). *Elektrische Maschinen und Antriebe: Grundlagen, Betriebsverhalten*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-540-71850-5>
- Blackenfelt, M. & Sellgren, U. (2000). Design of robust interfaces in modular products. In *Proceedings of the ASME 2000 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2: 26th Design Automation Conference* (S. 483-491). Baltimore, Maryland, USA. <https://doi.org/10.1115/DETC2000/DAC-14486>
- Blees, C. (2011). *Eine Methode zur Entwicklung modularer Produktfamilien*. Dissertation. Hamburg: Technische Universität Hamburg.
- Blessing, L. T. M. & Chakrabarti, A. (2009). *DRM, a Design Research Methodology*. London: Springer. <https://doi.org/10.1007/978-1-84882-587-1>
- Böhme, G. (1993). Fuzzy-Mengen. In *Fuzzy-Logik* (S. 1-25). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-86785-9\\_1](https://doi.org/10.1007/978-3-642-86785-9_1)
- Bolton, W. (2003). *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*. London: Pearson education.
- Bone, M. & Cloutier, R. (2010). The current state of model-based systems engineering: Results from the OMG™ SysML request for information 2009. In *Proceedings of the 8th conference on systems engineering research* (S. 225-232). Hoboken, NJ, USA.
- Braun, A. (2013). Modellbasierte Unterstützung der Produktentwicklung - Potentiale der Modellierung von Produktentstehungsprozessen am Beispiel des integrierten Produktentstehungsmodells (iPeM). Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 72). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000040221>

- Brown, M. (2001). *Power supply cookbook (2nd ed)*. Boston: Newnes.
- Broy, M. H. B. (2013). Software-Architekturen der Zukunft. *ATZagenda*, 2(1), 46–49. <https://doi.org/10.1007/s40357-013-0014-2>
- Bursac, N. (2016). Model Based Systems Engineering zur Unterstützung der Baukastenentwicklung im Kontext der Frühen Phase der Produktgenerationsentwicklung. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 93). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000054484>
- Bursac, N., Albers, A. & Öschlärer, M. (2016). Baukastenentwicklung durch MBSE am Beispiel einer modularen Fertigungsanlage im Kontext der Industrie 4.0. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering. TdSE 2016* (S. 247-256). München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446451414.024>
- Butz, H. (2011). Systemkomplexität methodisch erkennen und vermeiden. In R. Jochem & K. Landgraf (Hrsg.), *Anforderungsmanagement in der Produktentwicklung: Komplexität reduzieren, Prozesse optimieren, Qualität sichern* (S. 183-218). Düsseldorf: Symposion.
- Cadet, M., Sinnwell, C., Fischer, J., Rosen, R., Stephan, N. & Meissner, H. (2017). Referenzentwicklungsprozess für cybertronische Produkte und Produktionssysteme. In M. Eigner, W. Koch, C. Muggeo (Hrsg.), *Modellbasierter Entwicklungsprozess cybertronischer Systeme. Der PLM-unterstützte Referenz-entwicklungsprozess für Produkte und Produktionssysteme* (S. 45–62). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-662-55124-0\\_7](https://doi.org/10.1007/978-3-662-55124-0_7)
- Chamas, M., Hopfensitz, W., Oberfell, P., Oszwald, F. & Traub, M. (2019). Entwicklung eines Informationsmodells für eine Hochintegrationsplattform im Automotive-Umfeld. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering. TdSE 2019*. München: GfSE Verlag.
- Cooper, R. G. (1990). Stage-gate systems: a new tool for managing new products. *Business Horizons*, 33(3), 44–54. [https://doi.org/10.1016/0007-6813\(90\)90040-I](https://doi.org/10.1016/0007-6813(90)90040-I)

- Cooper, R. G. & Kleinschmidt, E. J. (1993). Screening new products for potential winners. *Long Range Planning*, 26(6), 74–81. [https://doi.org/10.1016/0024-6301\(93\)90208-W](https://doi.org/10.1016/0024-6301(93)90208-W)
- Czichos, H. (2015). *Mechatronik: Grundlagen und Anwendungen technischer Systeme (3., überarbeitete und erweiterte Auflage)*. Wiesbaden: Springer Vieweg.
- Daenzer, W. F. & Huber, F. (1994). *Systems Engineering. Methodik und Praxis (8. Aufl.)*. Zürich: Verl. Industrielle Organisation.
- Deicke, M. (2018). Virtuelle Absicherung von Steuergeräte-Software mit hardwareabhängigen Komponenten. Dissertation. In W. Hardt (Hrsg.), *Wissenschaftliche Schriftenreihe "Eingebettete, selbstorganisierende Systeme"* (Bd. 16). Chemnitz: Technische Universität Chemnitz, Fakultät für Informatik.
- Delligatti, L. (2014). *SysML distilled: a brief guide to the systems modeling language*. Upper Saddle River, NJ: Addison-Wesley.
- Deubzer, F. & Lindemann, U. (2009). Networked product modeling—use and interaction of product models and methods during analysis and synthesis. In M. Norell Bergendahl, M. Grimheden, L. Leifer, P. Skogstad & U. Lindemann (Hrsg.), *Proceedings of the 17th International Conference on Engineering Design (ICED09)* (S. 371-380). Palo Alto, CA, USA.
- Diehl, H. (2009). *Systemorientierte Visualisierung disziplinübergreifender Entwicklungsabhängigkeiten mechatronischer Automobilsysteme*. Dissertation. München: Technische Universität München.
- Dombrowski, U., Schmidt, S. & Wrehde, J. (2006). Herausforderungen in der Automobilproduktion. *Zeitschrift Für Wirtschaftlichen Fabrikbetrieb*, 101(5), 254–259. <https://doi.org/10.3139/104.101013>
- Dröschel, W. & Wiemers, M. (1999). *Das V-Modell 97: der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz*. Berlin: deGruyter. <https://doi.org/10.1515/9783486800265>
- Dumitrescu, R., Albers, A., Riedel, O., Stark, R. & Gausemeier, J. (Hrsg.). (2021). *Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft, Ein Beitrag zum Advanced Systems Engineering*, Paderborn.

- Ebel, B. (2015). Modellierung von Zielsystemen in der interdisziplinären Produktentstehung. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 85). Karlsruhe: Karlsruher Institut für Technologie (KIT).  
<https://doi.org/10.5445/IR/1000048334>
- Eckert, C. M., Alink, T. & Albers, A. (2010). Issue driven analysis of an existing product at different levels of abstraction. In *Proceedings of DESIGN 2010, the 11th International Design Conference* (S. 673-682). Dubrovnik, Kroatien.
- Eckert, C., Alink, T., Ruckpaul, A. & Albers, A. (2011). Different notions of function: results from an experiment on the analysis of an existing product. In *Journal of Engineering Design*, 22(11-12), 811-837.  
<https://doi.org/10.1080/09544828.2011.603297>
- eNOVA (2016). eNOVA-Positionspapier: *Halbleiterinnovation – Eine Enabling Technology für das Automobil der Zukunft*. Zugriff am 30.10.2022. Verfügbar unter <https://www.strategiekreis-automobile-zukunft.de/publikationen/enova-positionspapier-halbleiterinnovation-eine-enabling-technology-fuer-das-automobil-der-zukunft>
- Ehrlenspiel, K. (2009). *Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit (4. Aufl.)*. München: Carl Hanser Verlag.
- Ehrlenspiel, K. & Meerkamm, H. (2013). *Integrierte Produktentwicklung. Denkabläufe, Methodeneinsatz, Zusammenarbeit (5. Aufl.)*. München: Carl Hanser Verlag.
- Eigner, M. (2014). Product Lifecycle Management (PLM). In M. Eigner, D. Roubanov & R. Zafirov (Hrsg.), *Modellbasierte virtuelle Produktentwicklung* (Bd. 1, S. 267–300). Berlin: Springer. [https://doi.org/10.1007/978-3-662-43816-9\\_2](https://doi.org/10.1007/978-3-662-43816-9_2)
- Eigner, M., Gilz, T. & Zafirov, R. (2012). Proposal for Functional Product Description as Part of a PLM Solution in Interdisciplinary Product Development. In *Proceedings of DESIGN 2012, the 12th International Design Conference* (S.1667-1676). Dubrovnik, Kroatien.
- Eiletz, R. (1999). Zielkonfliktmanagement bei der Entwicklung komplexer Produkte - am Beispiel PKW-Entwicklung. Dissertation. In U. Lindemann (Hrsg.),



*Konstruktionstechnik München* (Bd. 32). Technische Universität München, Aachen: Shaker.

Eilmus, S., Gebhardt, N., Rettberg, R. & Krause, D. (2012). Evaluating a methodical approach for developing modular product families in industrial projects. In *Proceedings of DESIGN 2012, the 12th International Design Conference*. (S.837-846). Dubrovnik, Kroatien.

Elektroniknet (2021). *Auf dem Weg zum Multifunktionstalent*. Zugriff am 30.10.2022. Verfügbar unter <https://www.elektroniknet.de/halbleiter/analog-mixed-signal/auf-dem-weg-zum-multifunktionstalent.189621.html>

Eppinger, S. D. & Browning, T. R. (2012). *Design structure matrix methods and applications*. Cambridge, Mass: MIT Press.

Estefan, J. A. (2008). Survey of Candidate Model-Based Systems Engineering (MBSE) Methodologies, rev. B. In *International Council on Systems Engineering (INCOSE)*. *INCOSE-TD-2007-003-02*. Zugriff am 30.10.2022. Verfügbar unter [http://www.omgsysml.org/MBSE\\_Methodology\\_Survey\\_RevB.pdf](http://www.omgsysml.org/MBSE_Methodology_Survey_RevB.pdf)

Erden, M. S., Komoto, H., Van Beek, T. J., D'Amelio, V., Echavarria, E. & Tomiyama, T. (2008). A review of function modeling: Approaches and applications. In *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AI EDAM*, 22(2), 147-169.

Erixon, G. (1998). *Modular Function Deployment: A Method for Product Modularisation*. Dissertation. The Royal Institute of Technology, Department of Manufacturing Systems, Assembly Systems Division, Stockholm.

Feldhusen, J., Grote, K.-H., Göpfert, J. & Tretow, G. (2013). Technische Systeme. In J. Feldhusen & K.-H. Grote (Hrsg.), *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung (8. Aufl.)* (S. 237–279). Berlin: Springer.

Feldhusen, J., Grote, K.-H., Nagarajah, A., Pahl, G., Beitz, W. & Wartzack, S. (2013). Vorgehen bei einzelnen Schritten des Produktentstehungsprozesses. In J. Feldhusen & K.-H. Grote (Hrsg.), *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung (8. Aufl.)* (S. 291–409). Berlin: Springer.

- Felsch, A. (2010). *Organisationsdynamik: zur Konstitution organisationaler Handlungssysteme als kollektive Akteure (1. Aufl.)*. Wiesbaden: VS Verlag für Sozialwissenschaften.
- Föllinger, O. & Dörrscheidt, F. (1994). *Regelungstechnik: Einführung in die Methoden und ihre Anwendung (8., überarb. Aufl.)*. Heidelberg: Hüthig.
- Forschungsvereinigung Räumliche Elektronische Baugruppen 3-D MID (Hrsg.). (2004). *Technologie 3D-MID: räumliche elektronische Baugruppen: Herstellungsverfahren, Gebrauchsanforderungen, Materialkennwerte*. München: Carl Hanser Verlag.
- Franke, H.-J., Hesselbach, J., Huch, B. & Firchau, N. L. (2002). *Variantenmanagement in der Einzel- und Kleinserienfertigung*. München: Carl Hanser Verlag.
- Fraunhofer IEM (2020). MoSyS gestaltet die Zukunft der deutschen Ingenieursarbeit. Zugriff am 30.10.2022. Verfügbar unter <https://www.iem.fraunhofer.de/de/newsroom/presse-und-news/mosys-zukunft-deutscher-ingenieursarbeit.html>
- Friedenthal, S., Griego, R. & Sampson, M. (2007). *INCOSE model based systems engineering (MBSE) initiative*. [Konferenzbeitrag]. INCOSE 2007 symposium (Vol. 11), San Diego, USA.
- Friedenthal, S., Moore, A & Steiner, R. (2008). *A practical guide to SysML: the systems modeling language*. Burlington, Mass: Elsevier/Morgan Kaufmann.
- Galbraith, J. (1973). *Designing complex organizations*. Reading, Mass: Addison-Wesley Pub. Co.
- Gausemeier, J., Anacker, H., Czaja, A. M., Wassmann, H. & Dumitrescu, R. (2013). Auf dem Weg zu intelligenten technischen Systemen. In J. Gausemeier, R. Dumitrescu, F.-J. Rammig, W. Schäfer & A. Trächtler (Hrsg.), *9. Paderborner Workshop: „Entwurf mechatronischer Systeme“* (S. 11-47). Paderborn: Verlagsschriftenreihe des Heinz-Nixdorf-Instituts.
- Gausemeier, J., Czaja, A. M., Wiederkehr, O., Dumitrescu, R., Tschirner, C. & Steffen, D. (2013). Studie: Systems Engineering in der industriellen Praxis. In J. Gausemeier, R. Dumitrescu, F.-J. Rammig, W. Schäfer & A. Trächtler

- (Hrsg.), 9. *Paderborner Workshop: „Entwurf mechatronischer Systeme“* (S. 113–122). Paderborn: Verlagsschriftenreihe des Heinz-Nixdorf-Instituts.  
<https://doi.org/10.3139/9783446439467.012>
- Gausemeier, J., Czaja, A. M. & Dülme, C. (2015). Innovationspotentiale auf dem Weg zu Industrie 4.0. In J. Gausemeier, R. Dumitrescu, F.-J. Rammig, W. Schäfer & A. Trächtler (Hrsg.), 10. *Paderborner Workshop: „Entwurf mechatronischer Systeme“* (S.11-49). Paderborn: Verlagsschriftenreihe des Heinz-Nixdorf-Instituts.
- Gausemeier, J., Dumitrescu, R., Echterfeld, J., Pfänder, T., Steffen, D. & Thielemann, F. (2019). *Innovationen für die Märkte von morgen: strategische Planung von Produkten, Dienstleistungen und Geschäftsmodellen*. München: Carl Hanser Verlag.
- Gausemeier, J., Plass, C. & Wenzelmann, C. (2009). *Zukunftsorientierte Unternehmensgestaltung: Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen*. München: Carl Hanser Verlag.
- Gausemeier, J., Trächtler, A. & Schäfer, W. (2014). *Semantische Technologien im Entwurf mechatronischer Systeme: Effektiver Austausch von Lösungswissen in Branchenwertschöpfungsketten*. München: Carl Hanser Verlag.
- Genduso, F., Miceli, R., Rando, C. & Galluzzo, G. R. (2010). Back EMF Sensorless-Control Algorithm for High-Dynamic Performance PMSM. *IEEE Transactions on Industrial Electronics*, 57(6), 2092–2100.  
<https://doi.org/10.1109/TIE.2009.2034182>
- Gladysz, B. (2019). Gestalt- und wirkzusammenhangsbasierte Beschreibung von Fehlermechanismen für eine effektivere und effizientere Identifikation, Analyse sowie Nachvollziehbarkeit von Fehlerfolgen und -ursachen. Dissertation. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 119). Karlsruhe: Karlsruher Institut für Technologie (KIT).  
<https://doi.org/10.5445/IR/1000098444>
- Göpfert, J. (1998). *Modulare Produktentwicklung*. Wiesbaden: Deutscher Universitätsverlag. <https://doi.org/10.1007/978-3-663-08152-4>

- Götz, A. & Donges, C. (2016). Automatisierter Übergang vom dokumenten- zum modell-zentrierten Requirements Engineering als Ausgangsbasis für MBSE. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering. TdSE 2016* (S. 301-310). München: Carl Hanser Verlag.
- Grabowski, H., Geiger, K. (1997). *Neue Wege zur Produktentwicklung*. Stuttgart & Heidelberg: Raabe Stuttgart.
- Grundel, M., Abulawi, J., Moeser, G., Weilkiens, T., Scheithauer, A., Kleiner, S., ... Albers, A. (2014). FAS4M – No more: “Please mind the gap!”. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering. TdSE 2014* (S. 65-74). München: Carl Hanser Verlag.
- Haberfellner, R., Weck, O. L. de, Fricke, E. & Vössner, S. (2012). *Systems Engineering. Grundlagen und Anwendung (12. Aufl.)*. Zürich: Orell Füssli.
- Hainzl, H. & Nicolics, J. (2001). Miniaturisierung — der Motor der Entwicklung in der Elektronik. *e&i Elektrotechnik und Informationstechnik*, 118(2), 69–80. <https://doi.org/10.1007/BF03157755>
- Harashima, F., Tomizuka, M. & Fukuda, T. (1996). Mechatronics - „What Is It, Why, and How?“ An editorial. *IEEE/ASME Transactions on Mechatronics*, 1(1), 1–4. <https://doi.org/10.1109/TMECH.1996.7827930>
- Heismann, R. & Maul, L. (2012). Mit systematischem Innovationsmanagement zum Erfolg. In S. Ili (Hrsg.), *Innovation Excellence. Wie Unternehmen ihre Innovationsfähigkeit systematisch steigern* (S. 39–62). Düsseldorf: Symposion.
- Heitger, N. (2019). Methodische Unterstützung der initialen Zielsystembildung in der Automobilentwicklung im Modell der PGE - Produktgenerationsentwicklung. Dissertation. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 120). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000098206>
- Henderson, R. M. & Clark, K. B. (1990). Architectural innovation: The reconfiguration of existing product technologies and the failure of established firms. *Administrative science quarterly*, 35(1) 9-30. <https://doi.org/10.2307/2393549>

- Heumann, K. (1996). *Grundlagen der Leistungselektronik*. Wiesbaden: Vieweg+Teubner Verlag. <https://doi.org/10.1007/978-3-322-89227-0>
- Hillenbrand, M. (2012). Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen. Dissertation. In *Steinbuch Series on Advances in Information Technology* (Bd. 4). Karlsruhe: Karlsruher Institut für Technologie (KIT).
- Hoffmann, C. A. & Vietor, T. (2014). Erklärungsmodell der modularen Baukastensystematik in der Automobilindustrie. *Konstruktion*, 5(5), 79-82.
- Holt, J. & Perry, S. (2014). *Professional applications of computing: Bd. 10. SysML for systems engineering: a model-based approach (2. ed)*. London: Institution of Engineering and Technology.
- Hölttä-Otto, K. & deWeck, O. (2007a). Degree of modularity in engineering systems and products with technical and business constraint. *Concurrent Engineering*, 15(2), 113-126. <https://doi.org/10.1177/1063293X07078931>
- Hölttä-Otto, K. & deWeck, O. (2007b). Metrics for assessing coupling density and modularity in complex products and systems. In *Proceedings of the ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 3: 19th International Conference on Design Theory and Methodology* (S.343-352). Las Vegas, USA. <https://doi.org/10.1115/DETC2007-34871>
- Hossain, N. U. I. & Jaradat, R. M. (2018). A synthesis of definitions for systems engineering. In *Proceedings of the International Annual Conference of the American Society for Engineering Management* (S.1-10). Coeur d'Alene, Idaho, USA.
- Hüning, F. (2019). *Embedded Systems für IoT*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-662-57901-5>
- IEEE (1990). *IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology*. <https://doi.org/10.1109/IEEESTD.1990.101064>.
- INCOSE (2007). *System Engineering Vision 2020, Version 2.03*. Zugriff am 30.10.2022. Verfügbar unter [http://www.ccose.org/media/upload/SEVision2020\\_20071003\\_v2\\_03.pdf](http://www.ccose.org/media/upload/SEVision2020_20071003_v2_03.pdf)

- INCOSE (Hrsg.). (2011). *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Hoboken, USA: Wiley.
- IPEK (2020). *Start des BMBF-Verbundprojekts „MoSys-Menschorientierte Gestaltung komplexer System of Systems“*. Zugriff am 30.10.2022. Verfügbar unter <https://www.ipek.kit.edu/8469.php>
- Isermann, R. (1999). *Mechatronische Systeme: Grundlagen*. Berlin, Heidelberg: Springer.
- Ishii, K., Adler, R. & Barkan, P. (1988). Application of design compatibility analysis to simultaneous engineering. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 2(1), 53-65.  
<https://doi.org/10.1017/S0890060400000548>
- ISO (2018). ISO 26262. *Road vehicles -- Functional safety (Second Edition)*. ISO, Geneva, Switzerland.
- Iwanek, P., Kaiser, L., Dumitrescu, R. & Nyßen A. (2013). Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering. TdSE 2013* (S. 337-346). München: Carl Hanser Verlag.
- Jetter, A. (2005). *Produktplanung im Fuzzy Front End: Handlungsunterstützungssystem auf der Basis von Fuzzy Cognitive Maps (1. Aufl.)*. Dissertation. Wiesbaden : Dt. Univ.-Verl.
- Jiao, J., Simpson, T. W. & Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1), 5–29. <https://doi.org/10.1007/s10845-007-0003-2>
- Johanning, V. & Mildner, R. (2015). *Car IT kompakt*. Wiesbaden: Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-09968-8>
- Kano, N., Seraku, N., Takahashi, F. & Tsuji, S. (1984). Attractive Quality and Must-be Quality. *Journal of the Japanese Society for Quality Control*, 14(2), 147–156.
- Kaffenberger, R., Schulze, S.-O., Weber, H. (Hrsg.) (2012). *INCOSE Systems Engineering Handbuch*. München: GfSE Verlag.

- Kappos, E. & Kinniment, D. J. (1996). Application-specific processor architectures for embedded control: case studies. *Microprocessors and Microsystems*, 20(4), 225–232. [https://doi.org/10.1016/0141-9331\(96\)01084-8](https://doi.org/10.1016/0141-9331(96)01084-8)
- Kaufmann, U. & Schuler, R. (2016). Systems Re-Engineering-ein Beitrag zur Integration von MBSE und PLM. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering. TdSE 2016* (S. 343-352). München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446451414.033>
- Kersten, W. (1999). Wirksames Variantenmanagement durch Einbindung in den Controlling- und Führungsprozeß im Unternehmen. In VDI (Hrsg.), *VDI Berichte: Bd. 1510. Plattformkonzepte auch für Kleinserien und Anlagen?* (S. 155-176). Düsseldorf: VDI-Verlag.
- Khurana, A. & Rosenthal, S.R. (1997). Integrating the Fuzzy Front End of New Product Development. *Sloan Management Review*, 38(2), 103-120.
- Klare, H., Kramer, M., Langhammer, M., Werle, D., Burger, E. & Reussner, R. (2021). Enabling consistency in view-based system development — The Vitruvius approach. *Journal of Systems and Software*, 171, Artikel 110815. <https://doi.org/10.1016/j.jss.2020.110815>
- Kleiner, S. (2003). Föderatives Informationsmodell zur Systemintegration für die Entwicklung mechatronischer Produkte. Dissertation. In R. Anderl (Hrsg.), *Forschungsberichte aus dem Fachgebiet Datenverarbeitung in der Konstruktion* (Bd.15). Aachen: Shaker.
- Kleiner, S. & Kramer, C. (2013). Model based design with systems engineering based on RFLP using V6. In M. Abramovici & R. Stark (Hrsg.), *Lecture Notes in Production Engineering. Smart Product Engineering* (S. 93-102). Berlin, Heidelberg: Springer.
- Klingler, S. (2016). Eine Methode zur effizienten und effektiven Unterstützung der kontinuierlichen Validierung im Kontext der PGE – Produktgenerationsentwicklung. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 101). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000073864>

- Klippert, M., Marthaler, F., Spadinger, M. & Albers, A. (2020). Industrie 4.0 – An empirical and literature-based study how product development is influenced by the digital transformation. *Procedia CIRP*, 91, 80–86.  
<https://doi.org/10.1016/j.procir.2020.02.152>
- Koen, P., Ajamian, G., Burkart, R., Clamen, A., Davidson, J., D'Amore, R., ... & Wagner, K. (2001). Providing clarity and a common language to the “fuzzy front end”. *Research-Technology Management*, 44(2), 46-55.
- Kohlhase, N. (1997). *Strukturieren und Beurteilen von Baukastensystemen: Strategien, Methoden, Instrumente*. Düsseldorf: VDI-Verlag.
- Koller, R. (1998). *Konstruktionslehre für den Maschinenbau*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-80417-5>
- Kopenhagen, F. (2004). *Systematische Ableitung modularer Produktarchitekturen*. Dissertation, Technische Universität Hamburg-Harburg, Aachen: Shaker.
- Korff, A. (2008). *Modellierung von eingebetteten Systemen mit UML und SysML*. Heidelberg: Spektrum, Akad. Verl.
- Kremser, A. (2004). *Elektrische Maschinen und Antriebe*. Wiesbaden: Vieweg+Teubner Verlag. <https://doi.org/10.1007/978-3-663-01252-8>
- Kyura, N. & Oho, H. (1996). Mechatronics-an industrial perspective. *IEEE/ASME transactions on mechatronics*, 1(1), 10–15.
- Lackes, R. & Siepermann, M. (2018). *Definition von Kompatibilität*. Zugriff am 30.10.2022. Verfügbar unter <https://wirtschaftslexikon.gabler.de/definition/kompatibilitaet-39149/version-262565>
- Lamm, J. G. & Weilkiens, T. (2010). Funktionale Architekturen in SysML. In M. Maurer & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering. TdSE 2010* (S.109–118). München: Carl Hanser Verlag.
- Lamm, J. G. & Weilkiens, T. (2014). Method for deriving functional architectures from use cases. *Systems Engineering*, 17(2), 225-236.



- Lamm, J. G. & Weikiens, T. (2015). Happy Birthday! 5 Jahre Funktionale Architekturen nach FAS. In C. Muggeo & S.-O. Schulze (Hrsg.), *Tag des Systems Engineering. TdSE 2015* (S. 57-68). München: Carl Hanser Verlag.
- Lewes, G. H. (1875). *Problems of life and mind. First Series: the foundations of a creed*. Cambridge: University Press.
- Lindemann, U. (2009). *Methodische Entwicklung technischer Produkte (3. Aufl.)*. Berlin: Springer.
- Lohmeyer, Q. (2013). Menschzentrierte Modellierung von Produktentstehungssystemen unter besonderer Berücksichtigung der Synthese und Analyse dynamischer Zielsysteme. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 59). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000035102>
- Löw, P., Pabst, R. & Petry, E. (2010). *Funktionale Sicherheit in der Praxis: Anwendung von DIN EN 61508 und ISO/DIS 26262 bei der Entwicklung von Serienprodukten (1. Aufl.)*. Heidelberg: dpunkt-Verlag.
- Luhmann, N. (1991). *Soziale Systeme: Grundriß einer allgemeinen Theorie (4. Aufl.)*. Frankfurt am Main: Suhrkamp.
- Lykins, H., Friedenthal, S. & Meilich, A. (2000). Adapting UML for an object-oriented systems engineering method (OOSEM). In *Proceedings of the Tenth Annual INCOSE Symposium, International Council on Systems Engineering*. (S. 490-497). Minneapolis, USA. <https://doi.org/10.1002/j.2334-5837.2000.tb00416.x>
- Maier, M. W. (1998). Architecting principles for systems-of-systems. In *Systems Engineering*, 1(4), 267–284. [https://doi.org/10.1002/\(SICI\)1520-6858\(1998\)1:4<267::AID-SYS3>3.0.CO;2-D](https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D)
- Mandel, C., Stürmlinger, T., Yue, C., Behrendt, M. & Albers, A. (2020). Model-Based Systems Engineering Approaches for the integrated development of product and production systems in the context of Industry 4.0. In *IEEE International Systems Conference (SysCon)* (S. 1–7). Online Veranstaltung.

- Mantwill, F. (2014). Es muss uns gelingen!. *Konstruktion Zeitschrift für Produktentwicklung und Ingenieur-Werkstoffe*, 66(1/2), 3.
- Marthaler, F., Finster, F., Siebe, A., Bursac, N. & Albers A. (2019). PGE-Product Generation Engineering and Foresight: Developing Future-robust, Product Architectures. In *ISPIM Conference Proceedings* (S. 1-27). Manchester, England.
- Marxen, L. (2011). A Framework for Design Support Development based on the integrated Product Engineering Model iPeM. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 74). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000045823>
- Matheis, J. (2010). *Abstraktionsebenenübergreifende Darstellung von Elektrik/Elektronik-Architekturen in Kraftfahrzeugen zur Ableitung von Sicherheitszielen nach ISO 26262*. Dissertation. Aachen: Shaker.
- Matthiesen, S. (2002). Ein Beitrag zur Basisdefinition des Elementmodells "Wirkflächenpaare & Leitstützstrukturen" zum Zusammenhang von Funktion und Gestalt technischer Systeme. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des Instituts für Maschinenkonstruktionslehre und Kraftfahrzeugbau* (Bd. 6). Karlsruhe: Universität Karlsruhe (TH). <https://doi.org/10.5445/IR/12092002>
- Matthiesen, S., Schmidt, S., Moeser, G. & Munker, F. (2014). The Karlsruhe SysKIT Approach – A Three-step SysML Teaching Approach for Mechatronic Students. *Procedia CIRP*, 21, 385–390. <https://doi.org/10.1016/j.procir.2014.03.136>
- Matthiesen, S. (2021). Gestaltung - Prozess und Methoden. In B. Bender & K. Gericke (Hrsg.), *Pahl/Beitz Konstruktionslehre (9. Aufl.)* (S. 397-465). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-662-57303-7\\_13](https://doi.org/10.1007/978-3-662-57303-7_13)
- McManus, H. & Hastings, D. (2005, July). A framework for understanding uncertainty and its mitigation and exploitation in complex systems. In *Proceedings of the 15th Annual INCOSE Symposium, International Council on Systems Engineering* (S.484-503). Rochester, USA.

- Meboldt, M. (2008). Mentale und formale Modellbildung in der Produktentstehung. Als Beitrag zum integrierten Produktentstehungs-Modell (iPeM). Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 29). Karlsruhe: Karlsruher Institut für Technologie (KIT).  
<https://doi.org/10.5445/IR/1000028850>
- Meerkamm, H. (1995). Integrierte Produktentwicklung im Spannungsfeld von Kosten-, Zeit- und Qualitätsmanagement. In VDI (Hrsg.), *VDI Berichte: Bd. 1136. Wettbewerbsvorteile durch integrierte Produktentwicklung* (S. 1-14). Düsseldorf: VDI-Verlag.
- Meier, R. (2002). *Strategien für einen produktorientierten Einsatz räumlicher spritzgegossener Schaltungsträger (3-D MID)*. Dissertation. Friedrich-Alexander-Universität Erlangen-Nürnberg. Bamberg: Meisenbach Verlag.
- Mistree, F., Allen, J. K. & Attia, F. (1993). Designing at a high level of abstraction. *Behavioral Science*, 38(2), 124-138.
- Moeser, G., Albers, A. & Klingler, S. (2015). Modellbasierte Funktions-Gestalt-Synthese: Unterstützung der Entwickler bei der Definition der Produktgestalt. In *Design for X: Beiträge zum 26. DfX Symposium* (S.221-232). Herrsching, Deutschland.
- Moeser, G., Grundel, M., Weilkiens, T., Kümpel, S., Kramer, C. & Albers, A. (2016). Modellbasierter mechanischer Konzeptentwurf: Ergebnisse des FAS4M-Projektes. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering. TdSE 2016* (S. 417-428). München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446451414.040>
- Müller, M., Schiffbänker, P., Albers, A., Braun, A. & Bursac, N. (2017). Efficient application of MBSE using reference models: a PGE case study. In A. Maier, S. Škec, H. Kim, M. Kokkolaras, J. Oehmen, G. Fadel, F. Salustri & M. Van der Loos (Hrsg.) *Proceedings of the 21st International Conference on Engineering Design (ICED17)* (S. 271-280). Vancouver, Canada.
- Muschik, S. (2011). Development of Systems of Objectives in Early Product Engineering. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 50). Karlsruhe: Karlsruher Institut für Technologie (KIT).  
<https://doi.org/10.5445/IR/1000023768>

- Nikowitz, M., Hofer, M. & Schrödl, M. (2019). Untersuchung der Reduktion der harmonischen elektrischen Größen beim Betrieb einer Synchron-Reluktanzmaschine. *e&i Elektrotechnik und Informationstechnik*, 136(2), 120–128. <https://doi.org/10.1007/s00502-019-0715-y>
- Oerding, J. (2009). Ein Beitrag zum Modellverständnis der Produktentstehung - Strukturierung von Zielsystemen mittels C&CM. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 37). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000013689>
- Object Management Group (Hrsg.) (2007). *OMG Systems Modelling Language V1.0*. Zugriff am 30.10.2022. Verfügbar unter <https://doc.omg.org/formal/2007-09-01.pdf>.
- Pahl, G., Beitz, W., Grote, K.-H. & Feldhusen, J. (2007). *Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung, Methoden und Anwendung* (7. Aufl.). Berlin: Springer.
- Peglow, N., Powelske, J., Birk, C., Albers, A. & Bursac, N. (2017). Systematik zur Differenzierung von Varianten im Kontext der PGE–Produktgenerationsentwicklung. In *Tagungsband 15. Gemeinsames Kolloquium Konstruktionstechnik* (S. 9-18). Düsseldorf, Deutschland.
- Piller, F. T. & Waringer, D. (1999). *Modularisierung in der Automobilindustrie: neue Formen und Prinzipien, Modular Sourcing, Plattformkonzept und Fertigungssegmentierung als Mittel des Komplexitätsmanagements*. Aachen: Shaker.
- Pimpler, T. U. & Eppinger, S. D. (1994). Integration analysis of product decompositions. In *Proceedings of the ASME 1994 Design Technical Conferences collocated with the ASME 1994 International Computers in Engineering Conference and Exhibition and the ASME 1994 8th Annual Database Symposium. 6th International Conference on Design Theory and Methodology*. (S. 343-351). Minneapolis, USA. <https://doi.org/10.1115/DETC1994-0034>
- Pohl, K., Hönninger, H., Achatz, R. & Broy, M. (Hrsg.). (2012). *Model-Based Engineering of Embedded Systems*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-34614-9>

- Ponn, J. & Lindemann, U. (2008). *Konzeptentwicklung und Gestaltung technischer Produkte*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-540-68563-0>
- Powelske, J., Birk, C., Albers, A. & Bursac, N. (2017). MBSE-gestützte Strukturierung von Baukästen mechatronischer Produkte im Kontext der PGE – Produktgenerationsentwicklung, am Beispiel zweier Fallstudien. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering. TdSE 2017* (S. 203-212). München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446455467.023>
- Powelske, J., Mandel, C., Behrendt, M. & Albers, A. (2018). The role of helper plug-ins to efficiently support modelling and improve the acceptance of MBSE. In *Proceedings of Tag Des Systems Engineering (TdSE) / EMEASEC 2018* (S.1-13). Berlin, Deutschland.
- Qamar, A., During, C. & Wikander, J. (2009). Designing mechatronic systems, a model-based perspective, an attempt to achieve SysML-Matlab/Simulink model integration. In *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (S. 1306–1311). Singapur. <https://doi.org/10.1109/AIM.2009.5229869>
- Rapp, S., Moeser, G., Eichhorn, P. & Albers, A. (2018). Identifying Expedient Variations in PGE–Product Generation Engineering. In *Proceedings of the 2018 NordDesign Conference* (Artikel 141730). Linköping, Schweden.
- Renner, I. (2007). *Methodische Unterstützung funktionsorientierter Baukastenentwicklung am Beispiel Automobil*. Dissertation. München: TU München.
- Reichwein, A. & Paredis, C. J. (2011). Overview of architecture frameworks and modeling languages for model-based systems engineering. In *Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE* (S. 1341–1349). Washington, USA. <https://doi.org/10.1115/DETC2011-48028>
- Reif, K. (2011). *Bosch Autoelektrik und Autoelektronik: Bordnetze, Sensoren und elektronische Systeme* (6., überarb. und erw. Aufl). Wiesbaden: Vieweg + Teubner. <https://doi.org/10.1007/978-3-8348-9902-6>

- Reif, K. & Dietsche, K.-H. (2018). *Kraftfahrtechnisches Taschenbuch (29., überarbeitete und erweiterte Auflage)*. Wiesbaden: Springer Fachmedien Wiesbaden GmbH.
- Reinemann, J. (2021). Entwicklung einer Systematik zur Validierung interaktiver Produkte in Augmented-Reality-Umgebungen in der Frühen Phase im Modell der PGE – Produktgenerationsentwicklung. Dissertation. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 133). Karlsruhe: Karlsruher Institut für Technologie (KIT).  
<https://doi.org/10.5445/IR/1000128685>
- Reiß, N. (2018). Ansätze zur Steigerung der Methodenakzeptanz in agilen Prozessen der PGE - Produktgenerationsentwicklung. Dissertation. In A. Albers & S. Matthiesen (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 112). Karlsruhe: Karlsruher Institut für Technologie (KIT).  
<https://doi.org/10.5445/IR/1000084762>
- Reuss, H.-C., Diesner, S. & Thoben, M. (2004). Alternative concepts of power distribution for comfort electronics in passenger vehicles. In Proceedings *AUTOREG 2004: Steuerung und Regelung von Fahrzeugen und Motoren* (S. 605–616). Wiesloch, Deutschland.
- Robertson, D. & Ulrich, K. T. (1998). Planning for product platforms. *Sloan Management Review*, 39(4), 19-31.
- Ropohl, G. (1975). Einleitung in die Systemtechnik. In G. Ropohl (Hrsg.), *Systemtechnik. Grundlagen und Anwendung* (S. 1–77). München: Carl Hanser Verlag.
- Ropohl, G. (2009). *Allgemeine Technologie. Eine Systemtheorie der Technik (3. Aufl.)*. Karlsruhe: Univ.-Verl. Karlsruhe.
- Roth, K. (2000). *Konstruieren mit Konstruktionskatalogen*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-17466-7>
- Salvador, F. (2007). Toward a product system modularity construct: literature review and reconceptualization. *IEEE Transactions on Engineering Management*, 54(2), 219-240. <https://doi.org/10.1109/TEM.2007.893996>

- Sander, M. (2001). Entwicklungslinien und aktuelle Trends im Marketing. In S.K. Berninghaus & M. Brulke (Hrsg.), *Beiträge zur Mikro- und zur Makroökonomik* (S. 395-407). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-56606-6\\_32](https://doi.org/10.1007/978-3-642-56606-6_32)
- Saunders, S. (2011). *Does a Model Based Systems Engineering Approach Provide Real Program Savings? – Lessons Learnt*. [Konferenzbeitrag] Informational Symposium on Model-Based Systems Engineering DSTO, Edinburgh, Australia.
- Schäuffele, J. & Zurawka, T. (2010). *Automotive Software Engineering*. Wiesbaden: Vieweg+Teubner. <https://doi.org/10.1007/978-3-8348-9368-0>
- Scherer, H. (2016). Modellbasierte Methoden zur Modellierung des Zielsystems und des Funktions-Gestalt-Zusammenhangs zur Unterstützung der Serienentwicklung von Baukästen am Beispiel von Hybrid-Triebstrangsystemen. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 97). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000057850>
- Scherer, H., Albers, A. & Bursac, N. (2017). Model based requirements engineering for the development of modular kits. *Procedia CIRP*, 60, 145-150. <https://doi.org/10.1016/j.procir.2017.01.032>
- Schlager, C., Messnarz, R., Sporer, H., Riess, A., Mayer, R. & Bernhardt, S. (2018). Hardware SPICE Extension for Automotive SPICE 3.1. In X. Larrucea, I. Santamaria, R. V. O'Connor & R. Messnarz (Hrsg.), *Systems, Software and Services Process Improvement: Bd. 896. EuroSPI 2018. Communications in Computer and Information Science* (S. 480–491). Cham: Springer International Publishing. [https://doi.org/10.1007/978-3-319-97925-0\\_41](https://doi.org/10.1007/978-3-319-97925-0_41)
- Schlick, C., Bruder, R. & Luczak, H. (2018). Betriebs- und Arbeitsorganisation. In C. Schlick, R. Bruder & H. Luczak (Hrsg.), *Arbeitswissenschaft* (S. 635–679). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-662-56037-2\\_9](https://doi.org/10.1007/978-3-662-56037-2_9)
- Schmidt, K., Buhlmann, M., Ficek, C. & Richter, K. (2012). Entwurfsaspekte für hochintegrierte Steuergeräte mit unterschiedlichen ASIL-Stufen. *ATZelektronik*, 7(1), 34-39. <https://doi.org/10.1365/s35658-012-0117-y>

- Schröder, D. (2013). *Elektrische Antriebe - Grundlagen*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-30471-2>
- Schröder, D. & Böcker, J. (2021). *Elektrische Antriebe – Regelung von Antriebssystemen*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-662-62700-6>
- Schuh, G. (Hrsg.). (2015). *Leitfaden zur Baukastengestaltung: Ergebnisse des Forschungsprojekts Gestaltung innovativer Baukasten- und Wertschöpfungsstrukturen (GiBWert)*. Stuttgart: VDMA Verlag.
- Schuh, G. & Riesener, M. (2018). *Produktkomplexität managen: Strategien - Methoden - Tools* (3., vollständig überarbeitete Auflage). München: Carl Hanser Verlag.
- Schuh, G., Rudolf, S., Thomas, C., Schmitz, S., Vogels, T. & Schrey, E. (2014). *Gestaltung innovativer Baukasten- und Wertschöpfungsstrukturen*. Zugriff am 30.10.2022. Verfügbar unter [http://www.glinnotec.de/fileadmin/user\\_upload/Aufsatz\\_GiBWert.pdf](http://www.glinnotec.de/fileadmin/user_upload/Aufsatz_GiBWert.pdf)
- Schuh, G., Rudolf, S. & Vogels, T. (2014). Development of modular product architectures. *Procedia CIRP*, 20, 120-125. <https://doi.org/10.1016/j.procir.2014.05.042>
- Schulze, M. (2008). *Elektrische Servoantriebe: Baugruppen mechatronischer Systeme*. München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446416772.001>
- Shrutika, C., Matani, S., Chaudhuri, S., Gupta, A., Gupta, S. & Singh, N. (2021). Back-EMF estimation based sensorless control of Brushless DC motor. In *1st International Conference on Power Electronics and Energy (ICPEE)* (S. 1–6). Bhubaneswar, Indien. <https://doi.org/10.1109/ICPEE50452.2021.9358657>
- Sillitto, H., Martin, J., McKinney, D., Griego, R., Dori, D., Krob, D., ... & Jackson, S. (2019). *Systems engineering and system definitions*. Zugriff am 30.10.2022. Verfügbar unter [https://www.incose.org/docs/default-source/default-document-library/final\\_-se-definition.pdf](https://www.incose.org/docs/default-source/default-document-library/final_-se-definition.pdf)
- Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society*, 106(6), 467-482.



- Sosa, M., Eppinger, S. & Rowles, C. (2000). Designing modular and integrative systems. In *Proceedings of DETC '00, ASME 2000 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (S.303-312). Baltimore, Maryland, USA.  
<https://doi.org/10.1115/DETC2000/DTM-14571>
- Sosa, M., Eppinger, S. & Rowles, C. (2003). Identifying modular and integrative systems and their impact on design team interactions. *Journal of Mechanical Design*, 125(2), 240-252. <https://doi.org/10.1115/1.1564074>
- Sosa, M., Eppinger, S. & Rowles, C. (2004). The misalignment of product architecture and organizational structure in complex product development. *Management Science*, 50(12), 1674-1689.
- Specht, G., Beckmann, C. & Amelingmeyer, J. (2002). *F&E-Management: Kompetenz im Innovationsmanagement (2., überarb. und erw. Aufl)*. Stuttgart: Schäffer-Poeschel.
- Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Wien: Springer.
- Stark, R., Beier, G., Figge, A. & Wöhler, T. (2010). Domänenübergreifende Abhängigkeiten – Mehrwert durch Modellverknüpfungen in der Systementwicklung. In *Proceedings of Software and Systems Engineering Essentials SEE 2010* (S.139-154). München, Deutschland.
- Steffen, D. (2007). Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme. Dissertation. In J. Gausemeier (Hrsg.), *Verlagsschriftenreihe des Heinz-Nixdorf-Instituts* (Bd. 207). Paderborn: Universität Paderborn, Heinz-Nixdorf-Institut.
- Stetter, R., Rauchenberger, J., Spiegelberger, B., Müller, T., Pfeifer, G., Witzel, O. & Gideon, F. (2009). *Betriebliche Einführungsstrategie für ein anwendungsorientiertes Vorgehensmodell für die Entwicklung zuverlässigerer mechatronischer Systeme im Maschinen- und Anlagenbau*. Projektabschlussbericht des Forschungsprojekts BESTVOR. Zugriff am 30.10.2022. Verfügbar unter <http://publica.fraunhofer.de/dokumente/N-157223.html>

- Steward, D. V. (1981). Design structure system: a method for managing the design of complex systems. *IEEE Transactions on Engineering Management, EM* 28(3), 71-74.
- Störling, H.-D., Kallenbach, E. & Amrhein, W. (Hrsg.). (2011). *Handbuch elektrische Kleinantriebe (4., neu bearb. Aufl)*. München: Carl Hanser Verlag.
- Stone, R. B. (1997). *Towards a theory of modular design*. Dissertation. Austin: The University of Texas, Faculty of the Graduate School.
- Streichert, T. & Traub, M. (2012). *Elektrik/Elektronik-Architekturen im Kraftfahrzeug*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-642-25478-9>
- Stürmlinger, T., Jost, D., Mandel, C., Behrendt, M. & Albers, A. (2020). Impact and risk analysis in the integrated development of product and production system. *Procedia CIRP, 91*, 627–633. <https://doi.org/10.1016/j.procir.2020.02.221>
- Svendsen, K.-H. & Thorp Hansen, C. (1994). Decomposition of Mechanical Systems and Breakdown of Specifications. In *Proceedings of ICED93* (S.119–126). Den Haag, Niederlande.
- Taylor, L. E. & Henderson, M. R. (1994). The roles of features and abstraction in mechanical design. In *Proceedings of the ASME 1994 Design Technical Conferences collocated with the ASME 1994 International Computers in Engineering Conference and Exhibition and the ASME 1994 8th Annual Database Symposium. 6th International Conference on Design Theory and Methodology*. (S. 131-140). Minneapolis, USA. <https://doi.org/10.1115/DETC1994-0014>
- Tazir, N. (2011). Viel diskutiert, selten wirklich angewendet - Systems Engineering. *Produkt-DatenJournal, 18*(2).
- Tschöke, H., Gutzmer, P. & Pfund, T. (2019). *Elektrifizierung des Antriebsstrangs: Grundlagen - vom Mikro-Hybrid zum vollelektrischen Antrieb*. Berlin, Heidelberg: Springer. <https://doi.org/10.1007/978-3-662-60356-7>
- Tröster, P. M., Klotz, T., Rapp, S., Wessels, H., Ott, S. & Albers, A. (2021). Modellierung einer Einscheibentrockenkupplung mithilfe des C&C2-Ansatzes zur Identifikation von kritischen Gestalt-Funktion-Zusammenhängen bezüglich

- des Schwingungsphänomens zwangserregtes Kupplungsrupfen. *Forschung im Ingenieurwesen*, 85(1), 881–894. <https://doi.org/10.1007/s10010-021-00560-0>
- Ulrich, K. T. (1994). Fundamentals of product modularity. In S. Dasu, C. Eastman (Hrsg.), *Management of Design* (S. 219-231). New York: Springer.
- Ulrich, K. T. (1995). The role of product architecture in the manufacturing firm. In *Research Policy*, 24(3), 419–440. [https://doi.org/10.1016/0048-7333\(94\)00775-3](https://doi.org/10.1016/0048-7333(94)00775-3)
- Van Beek, T. J., Erden, M. S. & Tomiyama, T. (2010). Modular design of mechatronic systems with function modeling. *Mechatronics*, 20(8), 850-863. <https://doi.org/10.1016/j.mechatronics.2010.02.002>
- Van Wie, M. J., Rajan, P., Campbell, M. I., Stone, R. B. & Wood, K. L. (2003). Representing product architecture. In *Proceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 3b: 15th International Conference on Design Theory and Methodology*. (S. 731-746). Chicago, USA. <https://doi.org/10.1115/DETC2003/DTM-48668>
- Verworn B. (2005) *Die frühen Phasen der Produktentwicklung. Forschungs-/Entwicklungs-/Innovations-Management*. Wiesbaden: Deutscher Universitätsverlag. [https://doi.org/10.1007/978-3-663-09708-2\\_2](https://doi.org/10.1007/978-3-663-09708-2_2)
- VDI Verein Deutscher Ingenieure e.V. (1993). *VDI 2221. Entwicklung technischer Produkte und Systeme. Modell der Produktentwicklung*. Berlin: Beuth.
- VDI Verein Deutscher Ingenieure e.V. (2004). *VDI 2206. Entwicklungsmethodik für mechatronische Systeme*. Berlin: Beuth.
- VDI Verein Deutscher Ingenieure e.V. (2018). *VDI 2221 Blatt 1. Entwicklung technischer Produkte und Systeme. Modell der Produktentwicklung*. Berlin: Beuth.
- VDI Verein Deutscher Ingenieure e.V. (2018). *VDI 2221 Blatt 2. Entwicklung technischer Produkte und Systeme. Gestaltung individueller Produktentwicklungsprozesse*. Berlin: Beuth.

- VDI Verein Deutscher Ingenieure e.V. (2021). *VDI 2206. Entwicklung mechatronischer und cyber-physischer Systeme*. Berlin: Beuth.
- Volpato, G. & Stocchetti, A. (2008). Managing product life cycle in the auto industry: evaluating carmakers effectiveness. *International Journal of Automotive Technology and Management*, 8(1), 22–41.  
<https://doi.org/10.1504/IJATM.2008.018766>
- Walch, M. (2017). Variantenentwicklung im ZHO-Modell im Kontext der PGE - Produktgenerationsentwicklung – Methode zur Entscheidungsunterstützung bei der Konzeptentwicklung in der Angebotsphase auf Basis abgeleiteter Varianten. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 103). Karlsruhe: Karlsruher Institut für Technologie (KIT).  
<https://doi.org/10.5445/IR/1000073747>
- Walden, D. D., Roedler, G. J., Forsberg, K. J., Hamelin, R. D. & Shortell, T. M. (2015). *Systems Engineering Handbook. Fourth Edition*. Hoboken, USA: John Wiley & Sons.
- Wallaschek, J. (1995). Modellierung und Simulation als Beitrag zur Verkürzung der Entwicklungszeiten mechatronischer Produkte. In VDI (Hrsg.), *VDI Berichte: Bd. 1215. Simulation in der Praxis: neue Produkte effizienter entwickeln* (S.35–50). Düsseldorf: VDI-Verlag.
- Wallentowitz, H. & Reif, K. (Hrsg.). (2011). *Handbuch Kraftfahrzeugelektronik: Grundlagen - Komponenten - Systeme - Anwendungen* (2., und aktualisierte Aufl). Wiesbaden: Vieweg + Teubner.
- Weber, C. (2005). What is “complexity”? In A. Samuel & W. Lewis (Hrsg.), *Proceedings of the 15th International Conference on Engineering Design (ICED05)* (S. 292-293, executive summary, full paper DS35\_485.49). Melbourne, Australien.
- Weik, M. H. (2001). *Computer Science and Communications Dictionary*. Boston, MA: Springer US. <https://doi.org/10.1007/1-4020-0613-6>
- Weilkiens, T. (2007). *Systems engineering with SysML/UML: modeling, analysis, design*. Amsterdam, Boston: Morgan Kaufmann OMG Press/Elsevier.

- Weilkiens, T. (2016). *SYSMOD - the systems modeling toolbox: pragmatic MBSE with SysML (2nd edition, version 4.1)*. Fredesdorf: MBSE4U.
- Wilmsen, M., Dühr, K., Heimicke, J. & Albers, A. (2019). The First Steps Towards Innovation. A Reference Process for Developing Product Profiles. In S. Wartzack & B. Schleich (Hrsg.), *Proceedings of the 22nd International Conference on Engineering Design (ICED19)* (S. 1673–1682). Delft, Niederlande. <https://doi.org/10.1017/dsi.2019.173>
- Wintergerst, E. (2015). Leitfaden zur deduktiven Gestaltvariation durch Ermittlung der funktionsbestimmenden Stellgrößen in der Produktgenerationsentwicklung. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 86). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000048477>
- Winzer, P. (2017). *Steigerung von Drehmoment und Wirkungsgrad bei Synchronmaschinen durch Nutzung der magnetischen Asymmetrie*. Dissertation. Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000071097>
- Wuketits, F. M. (1979). Die Bedeutung des Systemdenkens in der Biologie. *Biologie in unserer Zeit*, 9(3), 73–79. <https://doi.org/10.1002/biuz.19790090305>
- Wyatt, D. F., Eckert, C. M. & Clarkson, P. J. (2009). Design of product architectures in incrementally developed complex products. In M. Norell Bergendahl, M. Grimheden, L. Leifer, P. Skogstad & U. Lindemann (Hrsg.), *Proceedings of the 17th International Conference on Engineering Design (ICED09)* (S. 167-178). Palo Alto, CA, USA.
- Wynn, D., Eckert, C. & Clarkson, J. (2007). Modelling iteration in engineering design. In J.-C. Bocquet (Hrsg.), *Proceedings of the 16th International Conference on Engineering Design (ICED07)* (S. 693-694), Paris, Frankreich.
- Yu, J. S., Gonzalez-Zugasti, J. P. & Otto, K. N. (1999). Product Architecture Definition Based Upon Customer Demands. In *Journal of Mechanical Design*, 121(3), 329–335. <https://doi.org/10.1115/1.2829464>
- Zamirowski, E. J. & Otto, K. N. (1999). Identifying product portfolio architecture modularity using function and variety heuristics. In *Proceedings of the ASME*

1999 *Design Engineering Technical Conferences. Volume 3: 11th International Conference on Design Theory and Methodology*. (S. 187-197). Las Vegas, USA. <https://doi.org/10.1115/DETC99/DTM-8760>

Zingel, J. C. (2013). Basisdefinition einer gemeinsamen Sprache der Produktentwicklung im Kontext der Modellbildung technischer Systeme und einer Modellierungstechnik für Zielsystem und Objektsystem technischer Systeme in SysML auf Grundlage des ZHO-Prinzips. Dissertation. In A. Albers (Hrsg.), *Forschungsberichte des IPEK - Institut für Produktentwicklung. Systeme, Methoden, Prozesse* (Bd. 70). Karlsruhe: Karlsruher Institut für Technologie (KIT). <https://doi.org/10.5445/IR/1000037421>

**Studentische Abschlussarbeiten die im Kontext dieser Dissertation am IPEK – Institut für Produktentwicklung am Karlsruher Institut für Technologie (KIT) vom Autor Co-betreut wurden:**

Zanger, M. A. (2018). *Entwicklung einer Methode zur Kompatibilitätsbewertung von Bausteinen eines Baukastens, am Beispiel mechatronischer Systeme*. Bachelorarbeit. Karlsruhe: Karlsruher Institut für Technologie (KIT).

**Literatur der Robert Bosch GmbH, die im Kontext dieser Dissertation von dem Autor verwendet wurde und die nicht öffentlich zugänglich ist:**

Robert Bosch GmbH. (2022). *Unternehmensinterne Mediathek*.

# Glossar

<b>Mechatronik</b>	[Mechatronik ist] „das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse, sowie bei der Prozessgestaltung“ (Kapitel 2.1.1)
<b>Elektronisches Steuergerät/Electronic Control Unit (ECU)</b>	Das Steuergerät empfängt die elektrischen Signale der Sensoren, wertet sie aus und berechnet die Ansteuersignale für die Stellglieder (Aktoren). Das Steuerungsprogramm – die „Software“ – ist in einem Speicher abgelegt. Die Ausführung des Programms übernimmt ein Mikrocontroller. Die Bauteile eines Steuergerätes werden als „Hardware“ bezeichnet. (Kapitel 2.1.2)
<b>Model Based Systems Engineering</b>	Model Based Systems Engineering ist die formalisierte Anwendung der Modellbildung zur Unterstützung der Aktivitäten im Rahmen der Anforderungsdefinition, des Systemdesigns, der Systemanalyse sowie der Verifikation und Validierung von Systemen, beginnend mit der Konzeptfindung über die Entwicklung und spätere Phasen des Lebenszyklus hinaus. (Übersetzt nach Friedenthal, Griego & Sampson, 2007) (Kapitel 2.2)
<b>Systems Engineering</b>	Systems Engineering ist ein interdisziplinärer Ansatz mit dem Ziel, die Entwicklung erfolgreicher Systeme zu ermöglichen. Systems Engineering konzentriert sich dabei auf die Definition und Dokumentation der Systemanforderungen in der frühen Entwicklungsphase, die Erarbeitung eines Systemdesigns und die Überprüfung des Systems auf Einhaltung der gestellten Anforderungen über alle Phasen des Lebenszyklus hinweg. Hierfür stellt das

<b>System</b>	<p>Systems Engineering geeignete Methoden und Prozesse bereit. (Kapitel 2.2)</p> <p>Ein System ist das Modell einer Ganzheit, die (a) Beziehungen zwischen Attributen (Inputs, Outputs, Zustände etc.) aufweist, die (b) aus miteinander verknüpften Teilen bzw. Subsystemen besteht, und die (c) von ihrer Umgebung bzw. von einem Supersystem abgegrenzt wird. (Kapitel 2.2.1)</p>
<b>Funktion</b>	<p>Der Begriff Funktion ist für den Produktentwickler nur im Umgang mit Modellen von technischen Systemen anwendbar. Sie ist die Umwandlung von Input zu Output, die in dem modellierten System abgebildet wird. Sie entsteht aus der Interaktion von Gestaltelementen und kann nicht direkt beeinflusst werden. (Kapitel 2.2.1)</p>
<b>Beabsichtigtes Verhalten</b>	<p>Beabsichtigtes Verhalten beschreibt die gewünschten Reaktionen eines Produkts oder seiner Teilsysteme auf Einflüsse, die durch Interaktionen mit Umgebungssystemen induziert werden. (Wintergerst, 2015, dort nach Albers &amp; Düser, 2010) (Kapitel 2.2.1)</p>
<b>Tatsächliches Verhalten</b>	<p>Tatsächliches Verhalten beschreibt die in der Erprobung oder Anwendung festgestellten erwünschten oder unerwünschten Wirkungen. (Alink, 2010, dort nach Albers &amp; Düser, 2010) (Kapitel 2.2.1)</p>
<b>Funktionale Sicherheit</b>	<p>Die Abwesenheit unzumutbarer Risiken, welche aus dem Fehlverhalten von E/E-Systemen hervorgerufen werden können. (Kapitel 2.2.3)</p>
<b>Ziel</b>	<p>Ein Ziel ist die bewusste gedankliche Vorwegnahme eines zukünftigen Soll-Zustands, der gemeinsam vereinbart und durch aktives Handeln angestrebt oder vermieden wird. (Lohmeyer, 2012; dort nach Eiletz, 1999) (Kapitel 2.2.3)</p>
<b>Anforderung</b>	<p>Eine Anforderung ist...</p> <ul style="list-style-type: none"><li>– eine Eigenschaft oder Bedingung, die ein System oder eine Person benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen,</li></ul>



---

	<ul style="list-style-type: none"><li>- eine Eigenschaft oder Bedingung, die ein System oder eine Systemkomponente aufweisen muss, um einen Vertrag, eine Norm, eine Spezifikation oder ein anderes formell vorgegebenes Dokument zu erfüllen,</li><li>- eine dokumentierte Repräsentation einer Eigenschaft oder Bedingung, wie in den ersten beiden Punkten beschrieben (Ebel, 2015; dort nach IEEE, 1990) (Kapitel 2.2.3)</li></ul>
<b>Randbedingung</b>	Eine Randbedingung ist ein aus dem Umfeld der Entwicklung resultierender äußerer Einfluss, der von anderen verantwortet wird und daher zwar ermittelt, aber nicht eigenständig definiert oder geändert werden kann. (Albers, Ebel & Lohmeyer, 2012; Lohmeyer 2013) (Kapitel 2.2.3)
<b>Produktarchitektur</b>	Die Produktarchitektur umfasst die Funktionsstruktur, die Produktstruktur (bzw. Baustruktur), sowie sämtliche Beziehungen zur Transformation zwischen den beiden Strukturen. (in Anlehnung an Albers & Wintergerst, 2014) (Kapitel 2.2.4)
<b>Referenzsystem</b>	Das Referenzsystem für die Entwicklung einer neuen Produktgeneration ist ein System, dessen Elemente aus bereits bestehenden oder bereits geplanten sozio-technischen Systemen und der zugehörigen Dokumentation stammen und die Basis und den Ausgangspunkt für die Entwicklung der neuen Produktgeneration bilden. (übersetzt nach Albers, Rapp, et al., 2019) (Kapitel 2.3.1)
<b>Frühe Phase der PGE – Produktgenerationsentwicklung</b>	Die „Frühe Phase“ der Produktgenerationsentwicklung ist eine Phase im Entwicklungsprozess einer neuen Produktgeneration, die mit der Initiierung eines Projektes beginnt und mit einer bewerteten technischen Lösung endet, die schließlich das initiale Zielsystem hinsichtlich seiner wesentlichen Elemente abdeckt. Die zur technischen Lösung gehörende Produktspezifikation als Teil des Zielsystems enthält u.a. Informationen bzgl. der verwendeten Technologien und Subsysteme sowie deren Übernahme- und

	<p>Neuentwicklungsanteile. Sie ermöglicht eine valide Bewertung des zu entwickelnden technischen Systems hinsichtlich der relevanten Parameter wie beispielsweise der Produzierbarkeit, der notwendigen Ressourcen oder des technischen und ökonomischen Risikos. (Albers, Rapp et al., 2017) (Kapitel 2.3.2)</p>
<b>Modul</b>	<p>Ein Modul ist ein technisches Subsystem, das sich durch andere technische Subsysteme ersetzen lässt, so, dass die Menge aller Funktionen oder Attribute des technischen Systems variiert wird. (Albers, Scherer et al., 2015) (Kapitel 2.4.1)</p>
<b>Plattform und Hut</b>	<p>Eine Plattform ist die Menge jener Subsysteme, die in unterschiedlichen technischen Systemen unverändert zum Einsatz kommt. Der Hut hingegen umfasst die restlichen Subsysteme, die in den unterschiedlichen technischen Systemen die Menge aller Funktionen oder Attribute variieren. (Albers, Scherer et al., 2015) (Kapitel 2.4.1)</p>
<b>Baureihe</b>	<p>Eine Baureihe bezeichnet mehrere technische Systeme, die eine ähnliche Produktarchitektur aufweisen. Die jeweiligen technischen Systeme unterscheiden sich dabei durch die Ausprägung einzelner Attribute, die durch Skalieren variiert werden. (Albers, Scherer et al., 2015) (Kapitel 2.4.1)</p>
<b>Baukasten</b>	<p>Ein Baukasten ist die Menge aller technischen Subsysteme, die dem zugehörigen Baukasten-Regelwerk folgen, mit dem Ziel, aus diesen Subsystemen technische Systeme mit jeweils unterschiedlicher Menge aller Funktionen konfigurieren zu können. (Albers, Scherer et al., 2015) (Kapitel 2.4.1)</p>

# Anhang A

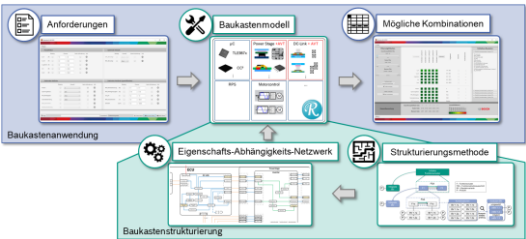
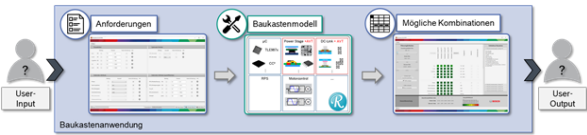
## Interviewleitfaden DS I – Baukastenstrukturierung

<p><b>Gesprächs- einleitung</b></p>	<p><b>Begrüßung</b></p> <p><b>Darstellung des Themas:</b> Entwicklung von MBSE-gestützten Methoden zur <b>Strukturierung</b> und Anwendung von Baukästen</p> <p><b>Verwendung:</b> Nutzung innerhalb eines Forschungsvorhabens zur Erlangung des Doktorgrades (deskriptive Studie I - Bedarfe)</p> <p><b>Ablauf:</b></p> <p>Aufbau: zwei Themenblöcke (offene Fragen) Themenblock I: Verifikation der Ergebnisse der teilnehmenden Beobachtung durch den Forscher Themenblock II: Bedarfsermittlung aus Sicht des Interviewpartners</p> <p>Dauer: 60 Minuten 5 Minuten Einleitung + optionale Vorbefragung 30 Minuten Themenblock I 20 Minuten Themenblock II 5 Minuten Wrap-up</p> <p><b>Allgemeines:</b> Zusicherung der Anonymität: die Interviewergebnisse werden anonymisiert, firmeninterne Informationen werden nicht veröffentlicht</p>
<p><b>(optional) allgemeine Vor- befragung</b></p>	<p><b>Wie viele Jahre Berufserfahrung haben Sie?</b></p> <p><input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/></p> <p>0-3 Jahre    3-5 Jahre    5-10 Jahre    &gt;10 Jahre</p> <p><b>Welche Funktion haben Sie im Unternehmen inne?</b></p> <p>Management:      <input type="checkbox"/>      <input type="checkbox"/></p> <p>Linie      Projekt-/Programm</p> <p><input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/></p> <p>Entwicklung: Mechanik    Hardware    Software    System</p>

<p><b>Themenblock I -</b> Strukturierter Teil des Interviews</p>	<p><b>Verifikation der Ergebnisse der teilnehmenden Beobachtung</b></p> <p><b>Teilergebnis 1:</b> Herausforderungen des PEP</p> <ul style="list-style-type: none"><li>- Pfadabhängigkeit:</li></ul> <p><i>Welche Herausforderungen für die Baukastenstrukturierung sehen Sie durch die hohe Interdisziplinarität der Steuergeräteentwicklung?</i></p> <p><i>Wie würden Sie die inhaltliche Zusammenarbeit der einzelnen Domänen bei der Entwicklung mechatronischer Steuergeräte beschreiben?</i></p> <p><i>Wie hoch ist der wahrgenommene Grad an Iterationen im Entwicklungsprozess bedingt durch die Beteiligung verschiedener Domänen?</i></p> <ul style="list-style-type: none"><li>- Sequenzielles Vorgehen nach dem V-Modell</li></ul> <p><i>Welche Rolle spielt das V-Modell im Entwicklungsalltag mechatronischer Steuergeräte?</i></p> <p><i>Welche Herausforderungen ergeben sich aus der sequenziellen Vorgehensweise des V-Modells für Ihren Arbeitsalltag?</i></p> <p><i>Welche weiteren Normen sind für die Entwicklung mechatronischer Steuergeräte bestimmend?</i></p> <ul style="list-style-type: none"><li>- Frühe Phase der PGE</li></ul> <p><i>Welche Herausforderungen sehen Sie durch die Verortung in der Vorausentwicklung bzw. der frühen Phase einer Produktentwicklung?</i></p> <p><i>Welche Rolle spielt die aktive Nutzung von Erfahrungswissen in dieser Phase?</i></p> <p>Optional: <i>Wie wird dies in ihrem Arbeitsumfeld realisiert?</i></p> <p><b>Verifikation der Ergebnisse der teilnehmenden Beobachtung</b></p> <p><b>Teilergebnis 2:</b> Herausforderungen der Architektur</p>
--	---

	<ul style="list-style-type: none"> <li>- Trend der Hochintegration</li> </ul> <p><b>Welche Herausforderungen sehen Sie in den unterschiedlichen Integrationsgraden der Bausteine eines mechatronischen Steuergerätes?</b></p> <p><b>Wie hoch ist die Anzahl an unterschiedlichen Varianten von Architekturtypen in der Steuergeräteentwicklung?</b></p> <ul style="list-style-type: none"> <li>- Unterschiedliche Technologiezyklen der beteiligten Domänen</li> </ul> <p><b>Welche Herausforderungen ergeben sich aus den unterschiedlichen Technologiezyklen der beteiligten Domänen?</b></p> <p><b>Wie wirken sich diese auf die bestehende Baukastenentwicklung aus?</b></p>
<p><b>Themenblock II - Freier Teil des Interviews</b></p>	<p><b>Bedarfe Baukastenstrukturierung Entwicklung und Management</b></p> <p>Leitfrage: <b>Welche Bedarfe sehen Sie hinsichtlich der Strukturierung eines Baukastens für das Handlungsfeld mechatronischer Kleinantriebe im Automobilbereich?</b></p> <p>Weitere Fragen werden spontan in Bezug auf die Antwort des Interviewpartners gestellt.</p> <p>Themen: Anknüpfung an die Fragen des Themenblocks I</p>
<p><b>Wrap-Up</b></p>	<p><b>Ausstiegsfrage:</b></p> <p><b>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</b></p> <p><b>Zusammenfassung</b></p> <p><b>Weiteres Vorgehen:</b></p> <ul style="list-style-type: none"> <li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li> <li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li> </ul>



	<p><b>Thematische Heranführung:</b> Zusammenspiel aus Baukastenstrukturierung und Baukastenanwendung</p>  <p>Leitfrage: <b>Welche Bedarfe sehen Sie hinsichtlich der Anwendung eines Baukastens für das Handlungsfeld mechatronischer Kleinantriebe im Automobilbereich?</b></p> <p>Weitere Fragen werden spontan in Bezug auf die Antwort des Interviewpartners gestellt.</p> <p>Themen: Anknüpfung an die Interaktion mit der Baukastenstrukturierung</p>
<p><b>Themenblock II</b> Ermittlung Interaktionskonzept (Toollösung)</p>	<p><b>Ermittlung der User-Stories zur Interaktion mit der Toollösung</b></p> <p>Leitfrage: <b>Welchen Interaktionswunsch mit einer Toolkette zur Baukastenanwendung haben Sie aus Sicht Ihrer Rolle?</b></p> 
<p><b>Wrap-Up</b></p>	<p><b>Ausstiegsfrage:</b></p> <p><b>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</b></p> <p><b>Zusammenfassung</b></p> <p><b>Weiteres Vorgehen:</b></p> <ul style="list-style-type: none"> <li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li> </ul>

	<ul style="list-style-type: none"><li>- Aufbereitung der User-Story und Objektivierung in Form von Use-Cases (gemeinsames Review)</li><li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li></ul>
--	--





	<ul style="list-style-type: none"><li>- Implementierung des initialen Baukastenmodells in Rhapsody</li></ul> <p>In beiden Fällen werden die Impact Models der Baukastenstrukturierung herangezogen und mit den Probanden offen über die Themen diskutiert.</p> <p>Die Fragen werden dazu situativ in Bezug auf die dargestellten Zusammenhänge formuliert.</p>
<b>Wrap-Up</b>	<b>Ausstiegsfrage:</b>  <b><i>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</i></b>  <b>Zusammenfassung</b>  <b>Weiteres Vorgehen:</b> <ul style="list-style-type: none"><li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li><li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li></ul>



	<p>In beiden Fällen werden die Impact Models der Baukastenstrukturierung herangezogen und mit den Probanden offen über die Themen diskutiert.</p> <p>Die Fragen werden dazu situativ in Bezug auf die dargestellten Zusammenhänge formuliert.</p>
<b>Wrap-Up</b>	<p><b>Ausstiegsfrage:</b></p> <p><b><i>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</i></b></p> <p><b>Zusammenfassung</b></p> <p><b>Weiteres Vorgehen:</b></p> <ul style="list-style-type: none"><li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li><li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li></ul>

## Interviewleitfaden des Validierungsschrittes Back-Testing

<p><b>Gesprächs- einleitung</b></p>	<p><b>Begrüßung</b></p> <p><b>Darstellung des Themas:</b> Entwicklung von MBSE-gestützten Methoden zur Strukturierung und Anwendung von Baukästen</p> <p><b>Verwendung:</b> Nutzung innerhalb eines Forschungsvorhabens zur Erlangung des Doktorgrades (deskriptive Studie II - Validierung)</p> <p><b>Ablauf:</b></p> <p>Aufbau: ein Themenblock - Themenblock: Back-Testing</p> <p>Dauer: 30 Minuten - 5 Minuten Einleitung + optionale Vorbefragung - 20 Minuten Themenblock - 5 Minuten Wrap-up</p> <p><b>Allgemeines:</b> Zusicherung der Anonymität: die Interviewergebnisse werden anonymisiert, firmeninterne Informationen werden nicht veröffentlicht</p>
<p><b>(optional) allgemeine Vor- befragung</b></p>	<p><b>Wie viele Jahre Berufserfahrung haben Sie?</b></p> <p><input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/></p> <p>0-3 Jahre 3-5 Jahre 5-10 Jahre &gt;10 Jahre</p> <p><b>Welche Funktion haben Sie im Unternehmen inne?</b></p> <p>Management: <input type="checkbox"/> <input type="checkbox"/></p> <p>Linie Projekt-/Programm</p> <p>Entwicklung: <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/></p> <p>Mechanik Hardware Software System</p>
<p><b>Themenblock – Back-Testing</b></p>	<p><b>Fragen zum Praxistest:</b></p> <p><b>Wie lange haben Sie für die Lösungsauswahl mit der Baukastenanwendung gebraucht?</b></p> <p><b>Wie lange hat dies bei der Erstbearbeitung der Projektanfrage gedauert?</b></p> <p><b>Wie würden Sie die Qualität der Lösungsempfehlungen auf Basis der Baukastenanwendung einschätzen?</b></p>

	<p><b>Wie groß war die Zahl an neuen Lösungen, die bei der initialen Bearbeitung nicht berücksichtigt wurden?</b></p> <p><b>Halten Sie die Nutzung der Methode durch weniger erfahrene Kollegen für geeignet bzw. für sinnvoll?</b></p> <p><b>Was ist Ihnen in Bezug auf die Beschaffenheit der Lösungen aufgefallen?</b></p> <p><b>Gab es Unterschiede in der Qualität der Lösungsempfehlungen zwischen den 12V und den 48V Projekten??</b></p> <p>Weitere Fragen werden fallspezifisch unter Bezugnahme auf, die im Impact Model dargestellten Zusammenhänge formuliert.</p>
<b>Wrap-Up</b>	<p><b>Ausstiegsfrage:</b></p> <p><b>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</b></p> <p><b>Zusammenfassung</b></p> <p><b>Weiteres Vorgehen:</b></p> <ul style="list-style-type: none"><li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li><li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li></ul>

## Interviewleitfaden des Validierungsschrittes Parallel-Testing

<b>Gesprächs- einleitung</b>	<p><b>Begrüßung</b></p> <p><b>Darstellung des Themas:</b> Entwicklung von MBSE-gestützten Methoden zur Strukturierung und Anwendung von Baukästen</p> <p><b>Verwendung:</b> Nutzung innerhalb eines Forschungsvorhabens zur Erlangung des Doktorgrades (deskriptive Studie II - Validierung)</p> <p><b>Ablauf:</b></p> <p>Aufbau: ein Themenblock</p> <ul style="list-style-type: none"> <li>- Themenblock: Parallel-Testing</li> </ul> <p>Dauer: 30 Minuten</p> <ul style="list-style-type: none"> <li>- 5 Minuten Einleitung + optionale Vorbefragung</li> <li>- 20 Minuten Themenblock</li> <li>- 5 Minuten Wrap-up</li> </ul> <p><b>Allgemeines:</b> Zusicherung der Anonymität: die Interviewergebnisse werden anonymisiert, firmeninterne Informationen werden nicht veröffentlicht</p>
<b>(optional) allgemeine Vor- befragung</b>	<p><b>Wie viele Jahre Berufserfahrung haben Sie?</b></p> <p><input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/></p> <p>0-3 Jahre   3-5 Jahre   5-10 Jahre   &gt;10 Jahre</p> <p><b>Welche Funktion haben Sie im Unternehmen inne?</b></p> <p>Management:      <input type="checkbox"/>      <input type="checkbox"/></p> <p>                          Linie      Projekt-/Programm</p> <p>                          <input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/></p> <p>Entwicklung: Mechanik   Hardware   Software   System</p>
<b>Themenblock – Parallel-Testing</b>	<p><b>Fragen zum Praxistest:</b></p> <p><b>Wie lange haben Sie für die Lösungsauswahl mit der Baukastenanwendung gebraucht?</b></p> <p><b>Wie lange hat dies bei der konventionellen Bearbeitung der Projektanfrage gedauert?</b></p> <p><b>Wie würden Sie die Qualität der Lösungsempfehlungen auf Basis der Baukastenanwendung einschätzen?</b></p>

	<p><b>Halten Sie die Nutzung der Methode durch weniger erfahrene Kollegen für geeignet bzw. für sinnvoll?</b></p> <p><b>Was ist Ihnen in Bezug auf die Beschaffenheit der Lösungen aufgefallen?</b></p> <p><b>Gab es Unterschiede in der Qualität der Lösungsempfehlungen zwischen den 12V und den 48V Projekten??</b></p> <p>Weitere Fragen werden fallspezifisch unter Bezugnahme auf, die im Impact Model dargestellten Zusammenhänge formuliert.</p>
<b>Wrap-Up</b>	<p><b>Ausstiegsfrage:</b></p> <p><b>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</b></p> <p><b>Zusammenfassung</b></p> <p><b>Weiteres Vorgehen:</b></p> <ul style="list-style-type: none"><li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li><li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li></ul>



## Interviewleitfaden des Validierungsschrittes Evaluation der Bedienerfreundlichkeit der implementierten Toolkette

<p><b>Gesprächseinleitung</b></p>	<p><b>Begrüßung</b></p> <p><b>Darstellung des Themas:</b> Entwicklung von MBSE-gestützten Methoden zur Strukturierung und Anwendung von Baukästen</p> <p><b>Verwendung:</b> Nutzung innerhalb eines Forschungsvorhabens zur Erlangung des Doktorgrades (deskriptive Studie II - Validierung)</p> <p><b>Ablauf:</b></p> <p>Aufbau: ein Themenblock</p> <ul style="list-style-type: none"> <li>- Themenblock: Bedienerfreundlichkeit der Toolkette</li> </ul> <p>Dauer: 15 Minuten</p> <ul style="list-style-type: none"> <li>- 2 Minuten Einleitung + optionale Vorbefragung</li> <li>- 10 Minuten Themenblock</li> <li>- 3 Minuten Wrap-up</li> </ul> <p><b>Allgemeines:</b> Zusicherung der Anonymität: die Interviewergebnisse werden anonymisiert, firmeninterne Informationen werden nicht veröffentlicht</p>
<p><b>(optional) allgemeine Vorbefragung</b></p>	<p><b>Wie viele Jahre Berufserfahrung haben Sie?</b></p> <p><input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/>      <input type="checkbox"/></p> <p>0-3 Jahre   3-5 Jahre   5-10 Jahre   &gt;10 Jahre</p> <p><b>Welche Funktion haben Sie im Unternehmen inne?</b></p> <p style="text-align: center;"><input type="checkbox"/>                      <input type="checkbox"/></p> <p>Management:    Linie            Projekt-/Programm</p> <p style="text-align: center;"><input type="checkbox"/>                      <input type="checkbox"/>                      <input type="checkbox"/>                      <input type="checkbox"/></p> <p>Entwicklung: Mechanik   Hardware   Software   System</p>
<p><b>Themenblock</b> – Bedienerfreundlichkeit der Toolkette</p>	<p><b>Fragen zum Praxistest:</b></p> <p><b>Halten Sie die Anwendung für bedienerfreundlich?</b></p>

	<p><b>Welche Verbesserungen zur Bedienerfreundlichkeit würden Sie sich wünschen?</b></p> <p><b>Würden Sie die Anwendung im Tagesgeschäft nutzen?</b></p>
<b>Wrap-Up</b>	<p><b>Ausstiegsfrage:</b></p> <p><b>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</b></p> <p><b>Zusammenfassung</b></p> <p><b>Weiteres Vorgehen:</b></p> <ul style="list-style-type: none"><li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li><li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li></ul>



	Die Fragen werden dazu situativ in Bezug auf die dargestellten Zusammenhänge formuliert
<b>Wrap-Up</b>	<b>Ausstiegsfrage:</b>  <b><i>Gibt es weitere Fragen/Anmerkungen zum Thema/Interview?</i></b>  <b>Zusammenfassung</b>  <b>Weiteres Vorgehen:</b> <ul style="list-style-type: none"><li>- Dokumentation der Interviewergebnisse in Form eines retrospektiven Protokolls (gemeinsames Review)</li><li>- Sammlung und Auswertung der Interviewergebnisse aller Interviewpartner</li></ul>

## Liste der wissenschaftlichen Veröffentlichungen des Autoren

- Albers, A., Bursac, N., Scherer, H., Birk, C., Powelske, J. & Muschik, S. (2019). Model-based systems engineering in modular design. *Design Science*, 5, Artikel e17. <https://doi.org/10.1017/dsj.2019.15>
- Albers, A., Peglow, N., Powelske, J., Birk, C. & Bursac, N. (2018). Coping with Complex Systems-of-Systems in the Context of PGE–Product Generation Engineering. *Procedia CIRP*, 70, 457-462. <https://doi.org/10.1016/j.procir.2018.01.004>
- Peglow, N., Powelske, J., Birk, C., Albers, A. & Bursac, N. (2017). Systematik zur Differenzierung von Varianten im Kontext der PGE–Produktgenerationsentwicklung. In *Tagungsband 15. Gemeinsames Kolloquium Konstruktionstechnik* (S. 9-18). Düsseldorf, Deutschland.
- Powelske, J., Birk, C., Albers, A. & Bursac, N. (2017). MBSE-gestützte Strukturierung von Baukästen mechatronischer Produkte im Kontext der PGE – Produktgenerationsentwicklung, am Beispiel zweier Fallstudien. In S.-O. Schulze, C. Tschirner, R. Kaffenberger & S. Ackva (Hrsg.), *Tag des Systems Engineering. TdSE 2017* (S. 203-212). München: Carl Hanser Verlag. <https://doi.org/10.3139/9783446455467.023>
- Powelske, J., Mandel, C., Behrendt, M. & Albers, A. (2018). The role of helper plug-ins to efficiently support modelling and improve the acceptance of MBSE. In *Proceedings of Tag Des Systems Engineering (TdSE) / EMEASEC 2018* (S.1-13). Berlin, Deutschland.