

Armin Kurrle

**Durchgängige Dokumentation von verteilten Zielsystemen in der Produktentwicklung durch Verwendung semantischer Metainformationen am Beispiel Connected Car**

Continous documentation of distributed system of objectives using semantic meta-information in Connected Car development

Band 108

Systeme ■ Methoden ■ Prozesse

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

Forschungsberichte

Dipl.-Inform. Armin Kurrle

**Durchgängige Dokumentation von verteilten  
Zielsystemen in der Produktentwicklung durch  
Verwendung semantischer Metainformationen am  
Beispiel Connected Car**

Continous documentation of distributed system of  
objectives using semantic meta-information in  
Connected Car development

Band 108

Herausgeber: Univ.-Prof. Dr.-Ing. Dr. h.c. A. Albers  
Univ.-Prof. Dr.-Ing. S. Matthiesen

Copyright: IPEK ▪ Institut für Produktentwicklung, 2017  
Karlsruher Institut für Technologie (KIT)  
Die Forschungsuniversität in der Helmholtz-Gemeinschaft

Alle Rechte vorbehalten

Druck: Stolzenberger Druck und Werbung GmbH & Co. KG, Leimen  
06224-7697915

ISSN 1615-8113

# **Durchgängige Dokumentation von verteilten Zielsystemen in der Produktentwicklung durch Verwendung semantischer Metainformationen am Beispiel Connected Car**

Zur Erlangung des akademischen Grades  
**Doktor der Ingenieurwissenschaften**  
der Fakultät für Maschinenbau  
Karlsruher Institut für Technologie (KIT)

genehmigte  
**Dissertation**

von

Dipl.-Inform. Armin Kurrle  
aus Balingen

Tag der mündlichen Prüfung: 18. Dezember 2017  
Hauptreferent: Univ.-Prof. Dr.-Ing. Dr. h.c. A. Albers  
Korreferent: Univ.-Prof. Dr.-Ing. M. Abramovici



## **Vorwort der Herausgeber (Stand: Juli 2017)**

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<sup>1</sup> 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

---

<sup>1</sup> Eh.: Institut für Maschinenkonstruktionslehre und Kraftfahrzeugbau, Universität Karlsruhe (TH)

## **Vorwort zu Band 108**

Produktentwicklung wird zunehmend komplexer. Diese Feststellung trifft sicherlich den Entwicklungstrend in den letzten Jahren. Insbesondere die Mechatronisierung, das heißt die Integration von mechanischen, elektrischen und informationstechnischen Teilsystemen in ein Gesamtprodukt und die Ergänzung dieses Produktes durch Services, oder Geschäftsmodelle und Internet-Vernetzungspotenziale, führen zu Strukturen, die man nur noch sinnvoll mit einem generalistischen Ansatz beschreiben kann. Dieser generalistische Ansatz zur Beschreibung komplexer Systeme – die sogenannte Systemtheorie, oder das daraus abgeleitete Systems-Engineering – ist hier so zu verstehen, dass wir zukünftig Systems-of-Systems haben werden und diese ganzheitlich planen, entwickeln und bauen müssen. Ein sehr gutes Beispiel für diese Entwicklung ist der Fahrzeugbau. Heutzutage sind moderne Fahrzeuge komplexe mechatronische Systeme mit einer Vielzahl von Teilsystemen aus den angesprochenen Domänen, die erst durch die synergetische Verknüpfung zum Gesamtprodukt, das tatsächlich dem Kunden angebotene Nutzungsprofil darlegen. Beispiele sind hier die automatisierte Stabilitätskontrolle, Abstandswarner oder die ersten Ansätze zum automatisierten Fahren. Ein weiteres Element, das hier im Fahrzeugbau eine entscheidende Rolle spielt, ist die Vernetzung der Fahrzeuge untereinander und mit stationären Systemen. In diesem Kontext spricht man von Connected Cars. Beim Connected Car werden durch den Datenaustausch zusätzliche Services und Leistungen als Produkt oder Teil des Produktes entwickelt. Produktentwicklung ist in diesem Sinne natürlich sehr viel breiter zu verstehen, als die Entwicklung eines individuellen mechanischen Artefaktes. Dieses ganzheitliche Verständnis eines Produktes als Kombination von mechatronischer Produktlösung mit Services und zusätzlichen Vernetzungsmöglichkeiten ist Grundlage der Karlsruher Schule für Produktentwicklung – KaSPro – seit vielen Jahren. Im Kontext dieser Betrachtung kommt einer strukturierten Beschreibung und Gestaltung der Entwicklung von Systems-of-Systems eine große Bedeutung zu. Hierbei kann der gesamte Entwicklungsprozess beschrieben werden auf den grundlegenden Ansätzen von ROPOHL als Überführung und Entwicklung eines Zielsystems in ein entsprechendes Objektsystem durch ein geeignetes Handlungssystem. Das Handlungssystem ist im Kontext von Systems-of-Systems natürlich extrem komplex, da hier innerhalb des Unternehmens unterschiedlichste Bereiche zusammenarbeiten und diese dann zusätzlich verknüpft werden müssen mit Zulieferern und Dienstleistern. Gleichzeitig ist zu berücksichtigen, dass nach dem Konzept der PGE – Produktgenerationsentwicklung von ALBERS ganz besonders in der Automobilindustrie viele Generationen gleichzeitig in der Entwicklung, bzw. auch am Markt sind. Auch diese zusätzliche Komplexitätsstufe ist bei der Behandlung von Systems-of-Systems und deren Zielsystemen zu berücksichtigen. Soll dann auch noch der Aspekt der Agilität als integraler Bestandteil des Entwicklungsprozesses wo immer möglich berücksichtigt werden kommt man zum AGS -

Agil Systems Design. Dies ist der integrierte Ansatz der Karlsruher Schule für Produktentwicklung zur Zusammenführung der aktivitätsbasierten Modellierung der Produktgenerationsentwicklung im IPED – integriertes Produktentstehungsmodell mit einer zielgerichteten Integration von Methoden der agilen Entwicklung – wie SCRUM – immer da wo Sie im Entwicklungsprozess des Maschinen- und Fahrzeugbau mit seinen harten Anforderungen an die Validierung mechatronischer Produkte hilfreich sind ohne das Entwicklungsrisiko unzulässig zu erhöhen und den Innovationserfolg zu gefährden.

An dieser Stelle setzt die Arbeit von Herrn Dr.-Ing. Armin Kurrle an. Er hat sich zum Ziel gesetzt, die durchgängige Dokumentation von verteilten Zielsystemen in der Produktentwicklung durch Verwendung semantischer Metainformationen als Lösungsansatz zu erforschen und so einen wichtigen Beitrag für eine agile, vernetzte und zielorientierte Entwicklung von komplexen Produktlösungen auf der Basis des ASD – Ansatzes zu erforschen.

Dezember, 2017

Albert Albers



## Kurzfassung

Insbesondere bei der interdisziplinären Entwicklung von komplexen, technischen Systemen und Systems-of-Systems sind Ziele und daraus abgeleitete Anforderungen ein wichtiges Instrument, um ein gemeinsames Zielbild des Produktes zu definieren, die Entwicklungsprozesse zu steuern und eine notwendige Basis für die Validierung und Verifikation des Produktes zu legen. Die vielfältigen Ziele und Anforderungen sowie deren Wechselwirkungen und Randbedingungen müssen dabei durchgängig als Zielsystem aufgefasst und dokumentiert werden. Die Entwicklung dieser erfolgt häufig parallel durch eine Vielzahl von Entwicklungsteams, welche teilweise unabhängig voneinander operieren können. Dabei entstehen in der Praxis eine Vielzahl partieller Zielsysteme mit hohen Abhängigkeiten untereinander.

Änderungen an einem Ziel oder einer Anforderung in einem partiellen Zielsystem können weitreichende Auswirkungen haben. Die Herstellung von Durchgängigkeit im gesamten Zielsystem über den gesamten Entwicklungsprozess und mehrere Produktgenerationen nach der Definition von ALBERS stellt folglich eine große Herausforderung dar. Dies wird in der Praxis durch Verwendung unterschiedlicher Methoden und Tools in den verteilten Entwicklungsteams zusätzlich erschwert. Werden wesentliche Inkonsistenzen bis zur Einführung nicht identifiziert, kann es weitreichende Schwierigkeiten im Entwicklungsprozess nach sich ziehen.

Im Rahmen der vorliegenden Forschungsarbeit wird ein Ansatz vorgestellt, welcher die Herausforderungen einer durchgängigen Entwicklung und Dokumentation von Zielsystemen für komplexe Systeme und Systems-of-Systems adressiert und dabei den gesamten Verlauf des Produktentstehungsprozess über mehrere Produktgenerationen unterstützt. Dazu wird die durchgängige Dokumentation partieller Zielsysteme durch Verknüpfung von Entwicklungsartefakten mit semantischen Meta-Informationen auf Basis des Resource Description Framework (RDF) ermöglicht. Ergänzt wird der Ansatz durch eine Methode für das durchgängige Management von Änderungen auf Basis des Problemlösungsprozess SPALTEN.

Die Anforderungen und Ziele für den Ansatz werden aus vier deskriptive Studien im Kontext realer Produktentwicklungsprojekte aus dem Umfeld Connected Car als Beispiel für ein System-of-Systems abgeleitet. Hierdurch wird der Ansatz an den tatsächlichen Bedarfen von Entwicklern in der Praxis ausgerichtet.

Der entwickelte Ansatz wird durch ein Software-Tool implementiert und unter praxisnahen Bedingungen mit Projektleitern aus der Industrie und in einem Live-Lab evaluiert. Die Validierung bestätigt dabei die Tauglichkeit des Ansatzes zur Verbesserung der Durchgängigkeit im Zielsystem bei verteilten, teilweise unabhängig voneinander agierenden Entwicklungsteams in der Praxis und es werden Verbesserungspotenziale und zukünftige Forschungsbedarfe aufgezeigt.

## **Abstract**

Particularly in interdisciplinary development of complex, technical systems and system-of-system, objectives and derived requirements are an important instrument to define a common vision of the product, to control the development processes and build the necessary basis for the validation and verification of the product. The multiple objectives and requirements as well as their interactions and boundary conditions have to be considered and documented as a continuous system of objectives. The development often takes place in parallel by a plurality of developing teams that can operate partially independently. In practice, this results in a number of partial systems of objectives with high interdependencies.

Changes to a goal or requirement in a partial system of objectives can have far-reaching consequences. The production of continuity throughout the system of objectives over the entire development process and multiple product generations therefore represents a major challenge. In practice, this is further impeded because of different methods and tools in the distributed development teams. If significant inconsistencies are not identified until the introduction, it may cause the failure of the product.

In the present research work, an approach is presented which addresses the challenges of continuous development of system of objectives by distributed teams especially in a distributed system-of-systems environment. The approach allows the continuous documentation of partial systems of objectives through links between development artifacts based on semantic meta-information. The continuous management of changes is supported by a method based on the problem solving method SPALTEN.

Requirements and goals for the approach are derived from four descriptive studies in the context of real product development projects from the Connected Car area as an example of a system-of-systems. Thereby, the approach is aligned to the actual developer's needs.

The developed approach is implemented by a software tool which is used for evaluation under practical conditions with project managers from industry. The validation confirms the suitability of the approach for improvement of continuity in the system of objectives developed by distributed, partially independent acting development teams. Furthermore, potential improvements are pointed out and future research needs are defined.

## Danksagung

Die vorliegende Arbeit entstand im Rahmen einer Forschungs Kooperation des IPEK – Institut für Produktentwicklung am KIT (Karlsruher Institut für Technologie) mit der Dr. Ing. h.c. F. Porsche AG.

Meinem Doktorvater Herrn Prof. Dr.-Ing. Dr. h.c. Albert Albers gilt mein ganz besonderer Dank für die Möglichkeit, Teil des IPEK Teams zu werden und auf Basis der intensiven gemeinsamen Diskussionen einen Beitrag zur IPEK - Forschungslandschaft und zur KaSPro - Karlsruher Schule der Produktentwicklung leisten zu können.

Für die Übernahme des Koreferats bedanke ich mich sehr herzlich bei Herrn Prof. Dr.-Ing. Michael Abramovici.

Beim gesamten IPEK-Team mit der Forschungsgruppe Systemische Mobilität bedanke ich mich für die freundliche Aufnahme, die konstruktiven Diskussionen, die freundschaftliche Atmosphäre und die gelungenen Unternehmungen – beruflich wie privat. Insbesondere bedanken möchte ich mich hier bei Herrn Dr.-Ing. Matthias Behrendt und Herrn Dr.-Ing. Simon Klingler.

Ebenso danke ich Herrn Stefan Arnold und Herrn Achim Hampel für das mir entgegen gebrachte Vertrauen und die großzügigen Gestaltungsfreiheiten während meiner gesamten Zeit bei Porsche.

Herrn Andreas Schmitz, Herrn Dr. Christian Spieth und Herrn Christian Gärtner danke ich für das Vorantreiben der gemeinsamen Projekte bei Porsche, sowie für die praktische und moralische Unterstützung.

Mein persönlicher Dank gilt Herrn Dr.-Ing. Nikola Bursac und Herrn Dr.-Ing. Helmut Scherer für viele inspirierende Gespräche bei der Porsche AG und am IPEK, eine erfolgreiche Zusammenarbeit und vor allem für die entstandene Freundschaft.

Außerdem möchte ich mich beim Porsche Doktoranden Netzwerk für den gewinnbringenden Austausch und viele schöne gemeinsame Abende bedanken.

Mein größter Dank gilt meiner Familie: Aline, Birgit, Wolfgang und Miriam für eine unglaubliche Unterstützung und Geduld in all den Jahren. Deshalb widme ich Euch diese Arbeit.

Karlsruhe, den 18. Dezember 2017

Armin Kurrle

„Es ist nicht ihr Ziel, der unendlichen Weisheit eine Tür zu öffnen, sondern eine Grenze zu setzen dem unendlichen Irrtum.“

*Bertolt Brecht* über die Wissenschaft

# Inhalt

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
<b>2</b>	<b>Grundlagen und Stand der Forschung</b> .....	<b>5</b>
2.1	Systems Engineering .....	5
2.1.1	Produktentstehung.....	8
2.1.2	Systemmodell der Produktentstehung.....	8
2.1.3	Produktgenerationsentwicklung.....	13
2.1.4	Produktentstehungs- und Entwicklungsprozesse.....	17
2.1.5	Management von Änderungen .....	28
2.1.6	Zwischenfazit .....	33
2.2	Zielsysteme in der Produktentstehung .....	35
2.2.1	Ziele, Anforderungen und Randbedingungen.....	35
2.2.2	Entwicklung und Management von Zielsystemen .....	38
2.2.3	Dokumentation von Zielsystemen .....	45
2.2.4	Zwischenfazit .....	57
2.3	Semantische Modelle und Techniken.....	59
2.3.1	Tagging .....	65
2.3.2	Das Ressource Description Framework (RDF) .....	66
2.3.3	Zwischenfazit .....	73
2.4	Systems-of-Systems .....	74
2.4.1	Definition und Eigenschaften .....	74
2.4.2	Typen und Beispiele von Systems-of-Systems .....	77
2.4.3	Systems-of-Systems Engineering.....	79
2.4.4	Zwischenfazit .....	83
<b>3</b>	<b>Motivation und Zielsetzung</b> .....	<b>85</b>
3.1	Motivation .....	85
3.2	Zielsetzung .....	86
<b>4</b>	<b>Forschungsdesign</b> .....	<b>88</b>
4.1	Forschungskooperation mit der Dr. Ing. h.c. F. Porsche AG .....	88
4.2	Design Research Methodology .....	90
<b>5</b>	<b>Empirische Untersuchungen zur Ableitung eines initialen Zielsystems</b> .....	<b>94</b>
5.1	Herausforderungen bei der Connected Car Entwicklung.....	95
5.1.1	Studiendesign .....	95
5.1.2	Ergebnisse der Expertenbefragung.....	97
5.1.3	Implikationen.....	101
5.2	Verständnis über die verteilte Entwicklung von Zielsystemen.....	103
5.2.1	Studiendesign .....	103
5.2.2	Ergebnisse der teilnehmenden Beobachtung.....	104
5.2.3	Implikationen.....	111

5.3	Verständnis über die verteilte Explikation von Zielsystemen .....	114
5.3.1	Studiendesign .....	114
5.3.2	Ergebnisse der Inhaltsanalyse.....	115
5.3.3	Implikationen.....	120
5.4	Verständnis über das Management von Änderungen in einem verteilten Umfeld	121
5.4.1	Studiendesign .....	121
5.4.2	Ergebnisse der Gruppendiskussionen.....	122
5.4.3	Implikationen.....	124
<b>6</b>	<b>Durchgängige Entwicklung von Zielsystemen .....</b>	<b>126</b>
6.1	Abbildung von Relationen bei heterogenen Entwicklungsartefakten .....	128
6.1.1	Verwendung semantischer Metainformationen .....	128
6.1.2	Eindeutige Referenzierung von Entwicklungsartefakten .....	131
6.1.3	Relationen zwischen Entwicklungsartefakten.....	133
6.1.4	Relationen zu Partialmodellen der Produktentwicklung .....	134
6.1.5	Relationen zu Referenzmodellen einer Produktgenerationsentwicklung .....	138
6.2	Ableitung impliziter Relationen aus vorhandenem Wissen .....	140
6.2.1	Implizite Relationen aus der Baustruktur.....	140
6.2.2	Implizite Relationen aus der Funktionsstruktur.....	142
6.2.3	Implizite Relationen aus Stakeholder-Beziehungen.....	143
6.2.4	Implizite Relationen über Entwicklungs- oder Produktgenerationen .....	145
6.3	Technische Realisierung durch eine web-basierte Plattform .....	146
6.3.1	Anwendungsarchitektur .....	146
6.3.2	Technische Systemspezifikation.....	148
6.4	Durchgängiges Management von Änderungen mit SPALTEN.....	150
6.4.1	Situationsanalyse: Änderung beobachten und Informationen sammeln .....	152
6.4.2	Problemeingrenzung: (Ziel der) Änderung definieren .....	153
6.4.3	Alternative Lösungen generieren und Lösung auswählen .....	154
6.4.4	Tragweitenanalyse: Bewertung der Änderung .....	154
6.4.5	Änderung entscheiden und umsetzen .....	156
6.4.6	Nacharbeiten und Lernen .....	156
<b>7</b>	<b>Evaluation des Ansatzes.....</b>	<b>158</b>
7.1	Evaluation der durchgängigen Dokumentation mit semantischen Metainformationen.....	158
7.1.1	Studiendesign .....	159
7.1.2	Ergebnisse und Implikationen.....	161
7.2	Evaluation des Management von Änderungen im Live-Lab ProVIL.....	165
7.2.1	Studiendesign .....	165
7.2.2	Ergebnisse und Implikationen.....	167

<b>8</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>172</b>
8.1	Zusammenfassung .....	172
8.2	Ausblick .....	177
<b>9</b>	<b>Literaturverzeichnis.....</b>	<b>179</b>
<b>10</b>	<b>Anhang.....</b>	<b>192</b>
10.1	Abbildungsverzeichnis .....	192
10.2	RDF Schema (siehe auch Kapitel 6.1) .....	196
10.3	Auszug Inferenzregeln (siehe auch Kapitel 6.2) .....	198
10.4	Ergebnisse Befragung (siehe auch Kapitel 7.2) .....	199

# 1 Einleitung

Heutige Produktinnovationen beruhen zunehmend auf einem engen Zusammenwirken von Mechanik, Elektrik/Elektronik, Regelungstechnik und Softwaretechnik, sogenannten Mechatronischen Systemen. Entwicklungen aus der Informations- und Kommunikationstechnik eröffnen dabei noch weitere Möglichkeiten. Zukünftige Produkte werden zunehmend „smart“ – d.h. interdisziplinär, komplex, autonom, vernetzt und mit eingebetteter Intelligenz.<sup>2</sup> Die physikalische Welt, in der die Systeme agieren, verschmilzt dabei mit der virtuellen Welt, über die sie miteinander vernetzt sind und kommunizieren als sogenannte Cyber-Physische Systeme (CPS).<sup>3</sup>

Eine Vielzahl der Entwicklungsprojekte kann dabei als Produktgenerationsentwicklung beschrieben werden<sup>4</sup>, wobei die Produktgenerationslebenszyklen der vernetzten Systeme unterschiedlich sein können. Um dem Kunden individuelle Produkte anzubieten und gleichzeitig die Produktkosten mit Hilfe von Skaleneffekten zu reduzieren basieren neue Produktgenerationen auf Vorgänger- oder Wettbewerbsprodukten - sogenannten Referenzprodukten - welche eine grundsätzliche Struktur vorgeben.<sup>5</sup>

Dieser Trend lässt sich insbesondere in der Automobilindustrie im Zuge einer immer weiter voranschreitenden Digitalisierung und Modularisierung beobachten. Nach einer Prognose von Gartner werden 2020 bereits über 250 Millionen Fahrzeuge vernetzt sein.<sup>6</sup> Das Auto der Zukunft wird sich mit anderen „smartem Produkten“<sup>7</sup> (z.B. Smart Buildings, Smart Grids, Smart Factorys, Smart Logistics) vernetzen und ermöglicht damit ein neues Mobilitätserlebnis sowie neue Geschäftsmodelle für IT-Unternehmen, Einzelhandel, Versicherungen und weitere. Das Fahrzeug ist dabei ein Teil eines komplexen, vernetzten System-of-Systems (SoS), an dessen Entwicklung eine Vielzahl von Menschen aus unterschiedlichen Disziplinen beteiligt ist. Die Entwicklung einzelner Systeme kann dabei teilweise unabhängig von anderen beteiligten Systemen und Organisationseinheiten ablaufen.

---

<sup>2</sup> vgl. Albers, Kurrle & Munker 2016

<sup>3</sup> vgl. Geisberger & Broy 2012

<sup>4</sup> vgl. Albers, Bursac & Wintergerst 2015

<sup>5</sup> vgl. Albers, Bursac & Wintergerst 2015

<sup>6</sup> vgl. Gartner 2015

<sup>7</sup> vgl. Abramovici 2015



Insbesondere bei der interdisziplinären Entwicklung von komplexen, technischen Systemen sind Ziele und daraus abgeleitete Anforderungen ein wichtiges Instrument in der Produktentwicklung, um ein gemeinsames Zielbild des Produktes zu definieren, die Entwicklungsprozesse zu steuern und eine notwendige Basis für die Validierung und Verifikation des Produktes zu legen. Ziele und Anforderungen können in unterschiedlichen Bereichen und in unterschiedlichen Detaillierungs- und Reifegraden existieren. Die vielfältigen Ziele und abgeleiteten Anforderungen sowie deren Wechselwirkungen und Randbedingungen müssen daher durchgängig als Zielsystem aufgefasst und abgebildet werden<sup>8</sup>. Inkonsistenzen oder Widersprüche zwischen einzelnen Zielen werden sonst erst zu spät im Verlauf der Entwicklung identifiziert, wodurch erhebliche Änderungsaufwände in Kosten und Zeit entstehen können. Werden wesentliche Inkonsistenzen im Zielsystem nicht identifiziert, kann es zu Verzögerungen in der Entwicklung, höheren Kosten auf Grund von Nacharbeiten oder sogar zum Scheitern des Produkts kommen.

Die Entwicklung und die durchgängige Dokumentation von Zielsystemen ist bereits bei der Entwicklung eines einzelnen Systems eine große Herausforderung. Es gibt hierzu bereits viele Untersuchungen und diverse Ansätze wurden entwickelt, welche versuchen mit den Herausforderungen umzugehen (vgl. auch Kapitel 2.2). Die Entwicklung des Connected Car ist für die Automobilhersteller noch ein vergleichsweise neues Feld und es gibt noch wenige Arbeiten, die diese untersucht haben. Bereits in anderen Domänen konnte festgestellt werden, dass bei der Entwicklung von Systems-of-Systems auf Grund der Interaktion einer Vielzahl teilweise unabhängig voneinander agierender Organisationseinheiten („Managerial & Operational Independence“ nach Maier<sup>9</sup>) mit unterschiedlichen Methoden, Prozessen und Tools die Entwicklung und Abbildung von Zielen und Anforderungen für Systems-of-Systems eine noch größere Herausforderung ist. Bestehende Techniken müssen weiterentwickelt werden.<sup>10</sup>

Ziel der vorliegenden Arbeit ist die Entwicklung eines Ansatzes, welcher die Herausforderungen einer verteilten und teilweise unabhängig stattfindenden Entwicklung von Zielsystemen für komplexe Systeme und Systems-of-Systems adressiert und dabei insbesondere den gesamten Verlauf der Produktentstehungsprozess über mehrere Produktgenerationen durchgängig unterstützt.

---

<sup>8</sup> vgl. Albers, Klingler & Ebel 2013

<sup>9</sup> vgl. u.a. Maier 1998, Lewis et al. 2009 und Ncube 2011

<sup>10</sup> vgl. Hallerstedte et al. 2012

Die Forschungsarbeit ist in 8 Kapitel gegliedert, welche in Abbildung 1 dargestellt sind. Diese werden im Folgenden kurz vorgestellt.

In **Kapitel 2** wird zunächst eine strukturierte Analyse des aktuellen Forschungsstandes zur Entwicklung und Abbildung von Zielsystemen, semantischen Modellen und Techniken sowie Systems-of-Systems beschrieben. Damit werden die Grundlagen dieser Arbeit geschaffen.

**Kapitel 3** stellt die grundlegende Motivation und detaillierte Zielsetzung abgeleitet aus dem Stand der Forschung dar. Das Kapitel schließt mit drei zentralen Forschungsfragen ab. Darauf aufbauend wird in **Kapitel 4** ein an den Forschungsfragen ausgerichtetes Forschungsvorgehen definiert.

**Kapitel 5** beschreibt empirische Untersuchungen zur Ableitung von Zielen und Anforderungen an einen Ansatz. Der Forschungsschwerpunkt wird im Rahmen der ganzen Arbeit auf die Zielsystementwicklung für das Connected Car gelegt. Dazu wird zunächst durch eine Befragung von Experten bei Automobilherstellern, Zulieferern und Dienstleistern aus dem Bereich Connected Car der Bedarf für einen neuen Ansatz begründet und damit der Forschungsbedarf verifiziert (vgl. Kapitel 5.1). Im Rahmen von zwei weiteren Untersuchungen bei der Dr. Ing. h.c. F. Porsche AG im Bereich Porsche Connect wird ein vertiefendes Verständnis über die verteilte Entwicklung und Explikation von Zielsystemen geschaffen (vgl. Kapitel 5.2 und 5.3). Den Abschluss der Ist-Analyse bildet eine Untersuchung im Rahmen eines regelmäßig tagenden Arbeitskreises mit dem Ziel der Sammlung von Anforderungen an das Management von Änderungen in einem verteilten Umfeld (vgl. Kapitel 5.4).

Auf dieser Basis wird in **Kapitel 6** ein neuer Ansatz beschrieben, welcher durch Verwendung von Metainformationen bei Entwicklungsartefakten Durchgängigkeit im Zielsystem über alle Entwicklungsphasen und unterschiedlichen Produktgenerationslebenszyklen herstellt. Zur Sicherstellung der Anwendbarkeit des Ansatzes wird ein auf der SPALTEN-Methodik basiertes Vorgehensmodell für das Management von Änderungen definiert. Die Implementierung des Ansatzes erfolgt durch einen Software-Prototypen, welcher am Ende des Kapitels beschrieben wird.

Abschließend wird in **Kapitel 7** der Ansatz evaluiert. Dazu wird zum einen der entwickelte Software-Prototyp dazu verwendet, um unter praxisnahen Bedingungen mit Projektleitern der Dr. Ing. h.c. F. Porsche AG die Potentiale des Ansatzes zur Erhöhung der Durchgängigkeit durch Verwendung der bereitgestellten semantischen Metainformationen zu bewerten. Zum anderen wird die Methode für das Management von Änderungen im Live-Lab ProVIL validiert.

Die Arbeit schließt in **Kapitel 8** mit einer Zusammenfassung der Ergebnisse sowie Anknüpfungspunkten an weitere Forschungsarbeiten ab.

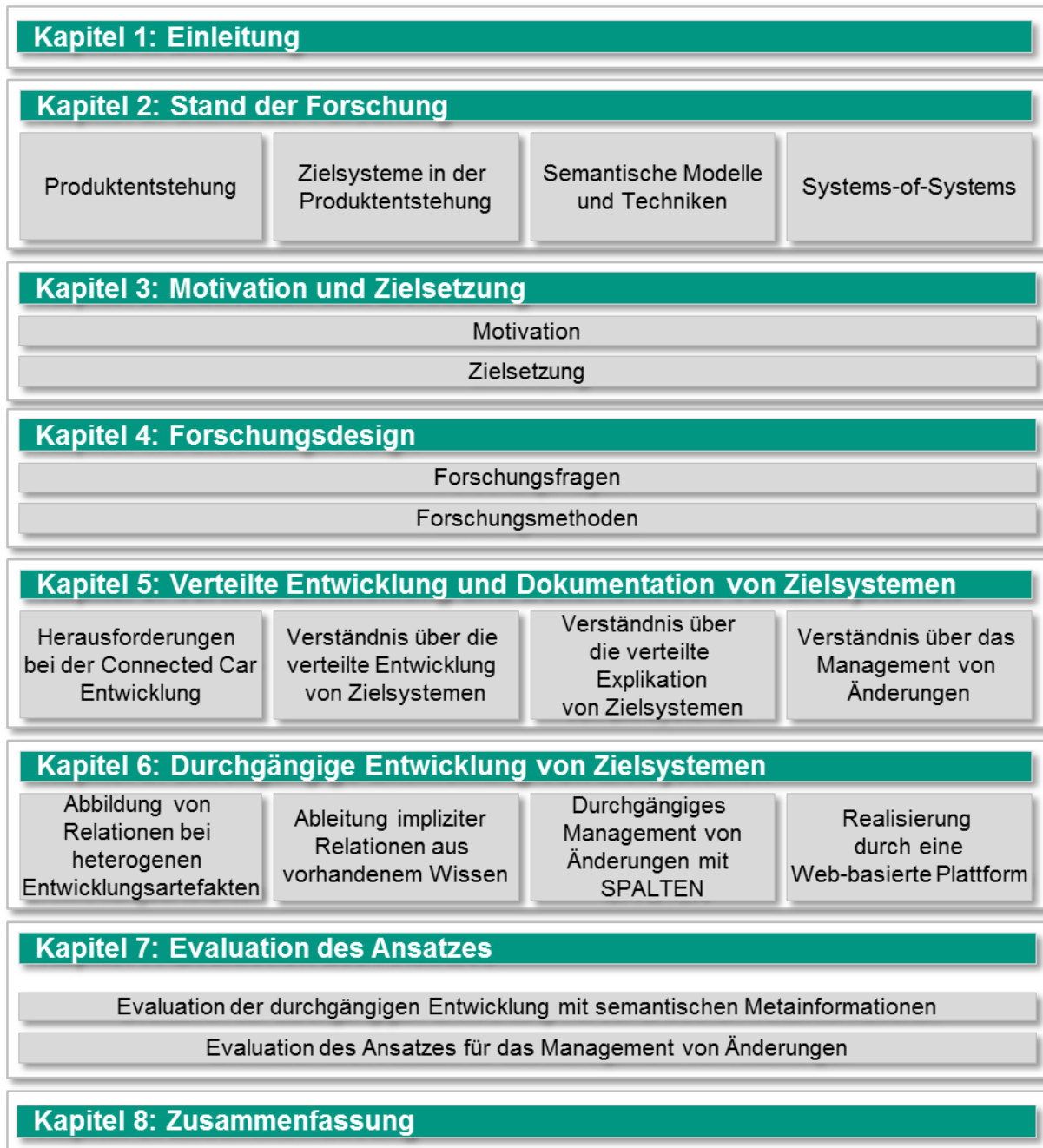


Abbildung 1: Aufbau der Arbeit

## 2 Grundlagen und Stand der Forschung

In diesem Kapitel werden die Grundlagen für die Arbeit und der relevante Stand der Forschung vorgestellt und diskutiert.

### 2.1 Systems Engineering

Bereits 1982 beschreibt PATZAK<sup>11</sup> die Entwicklung technischer Systeme als „explosionsartig“ – gekennzeichnet vor allem durch immer kürzere Innovationszeiten, größer werdende Komplexität sowie durch zunehmend fachübergreifende Problemstellungen.<sup>12</sup> Auch heute setzt sich dieser Trend weiter fort. So durchliefen die Erzeugnisse des Maschinenbaus und verwandter Branchen wie der Automobilindustrie in den letzten Jahren einen kontinuierlichen Wandel von mechanisch geprägten hin zu Mechatronischen Systemen. Dabei wurde die Informations- und Kommunikationstechnik zu einem wesentlichen Innovationstreiber<sup>13</sup>. Die Entwicklung dieser Systeme kann nicht mehr aus dem Blickwinkel einer einzelnen Fachdisziplin betrachtet werden<sup>14</sup>. Die Systemtechnik (engl. Systems Engineering) bietet durch Einnahme entsprechender Abstraktionsebenen die Möglichkeit, Erkenntnisse und Methoden eines Fachgebiets auch auf beliebige andere Anwendungsgebiete zu übertragen. Dieser Ansatz basiert, von fachlichen Restriktionen befreit, auf der Philosophie, in Systemen zu denken<sup>15</sup>.

Das Systems Engineering hat seine Wurzeln bereits in der Antike und wurde vom Biologen BERTALANFFY im 20. Jahrhundert zur allgemeinen Systemtheorie<sup>16</sup> zusammengefasst. Dies wird als ein Schlüsselwerk für das heutige Systemverständnis gesehen. Der Begriff Systemtechnik wurde in Deutschland zunächst durch ROPOHL in den 1970er Jahren geprägt. Im Jahr 1971 veröffentlichte BEITZ in den VDI-Berichten einen Beitrag mit dem Titel „Systemtechnik im Ingenieurbereich“, gefolgt von Richtungsweisenden Veröffentlichungen von DAENZER/HUBER, PATZAK, HABERFELLNER und ALBERS.

Organisationen wie die inzwischen größte internationale Systems Engineering Organisation INCOSE haben Ende der 1980er Jahre auf Grund der Vielzahl an

---

<sup>11</sup> vgl. Patzak 1982

<sup>12</sup> vgl. Patzak 1982

<sup>13</sup> vgl. Gausemeier et al. 2013

<sup>14</sup> vgl. Gausemeier et al. 2013

<sup>15</sup> vgl. Patzak 1982

<sup>16</sup> vgl. Bertalanffy 1969, übersetzt nach INCOSE 2011

aufkommenden Arbeiten im Bereich des Systems Engineering, erste Konsolidierungsbemühungen begonnen. Die INCOSE definiert SE wie folgt:

**Definition 2-1: Systems Engineering<sup>17</sup>**

Systems Engineering ist ein interdisziplinärer Ansatz und soll die Entwicklung von Systemen methodisch ermöglichen. SE fokussiert ein ganzheitliches und zusammenwirkendes Verständnis der Stakeholder Anforderungen, der Entdeckung von Lösungsmöglichkeiten und der Dokumentation von Anforderungen sowie das Synthetisieren, Verifizieren, Validieren und die Entwicklung von Lösungen. Das gesamte Problem wird währenddessen von der Konzeptentwicklung bis zur Systementwicklung betrachtet. Das Systems Engineering stellt hierfür geeignete Methoden, Prozesse und Best Practices bereit.

ROPOHL<sup>18</sup> stellt heraus, dass in der Allgemeinen Systemtheorie drei unterschiedliche Systemdeutungen vertreten sind, die jeweils einen Systemaspekt in den Vordergrund stellen, während der Systembegriff in Wirklichkeit drei Aspekte umfasst (vgl. Abbildung 2).

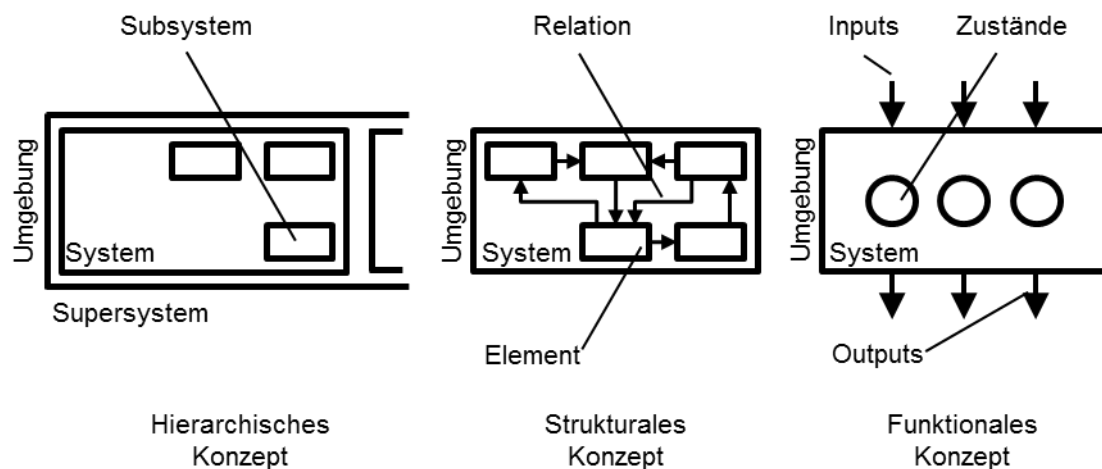


Abbildung 2: Konzepte der Systemtheorie nach ROPOHL<sup>19</sup>

Das **hierarchische Konzept** besagt, dass die Teile eines Systems selbst als Systeme gesehen werden können und das (Super-)System selbst seinerseits auch als Teil eines umfassenderen Systems angesehen werden kann – Systeme können

<sup>17</sup> vgl. Gausemeier et al. 2013

<sup>18</sup> vgl. Ropohl 2009, S. 76

<sup>19</sup> vgl. Ropohl 2009

also gleichzeitig Sub- und Supersystem sein<sup>20</sup>. Dieser hierarchische Gedanke erlaubt also mehrere Stufen von „Ganzheiten und Teilen“<sup>21</sup>, wobei man, wenn man sich in der Hierarchie abwärts bewegt, eine detailliertere Erklärung des Systems erhält. Wenn man sich in der Hierarchie aufwärts bewegt erhält man ein tieferes Verständnis über seine Bedeutung<sup>22</sup>. Das **strukturele Systemkonzept** besteht darin, ein System als Ganzheit miteinander verknüpfter Elemente zu betrachten.<sup>23</sup> Diese Auffassung kommt auch im holistischen Gesetz „das Ganze ist mehr als die Summe seiner Einzelteile“ zum Ausdruck, wobei das „mehr“ durch die Vielzahl möglicher Relationen zwischen den Elementen entsteht<sup>24</sup>. Nach Patzak<sup>25</sup> ist ein System eine Sammlung von Komponenten, zusammengebracht um eine bestimmte Funktion oder eine Reihe von Funktionen zu erfüllen. Dieser Sachverhalt wird auch im **funktionalen Konzept** nach ROPOHL dargestellt. Das System stellt hier eine sogenannte „Black Box“ dar<sup>26</sup>. Der tatsächliche innere Aufbau wird also außer Acht gelassen und die an der Systemgrenze beobachtbare Verhaltensweise in den Vordergrund gerückt. Es wird nicht gefragt: „Was ist es?“, sondern: „Was tut es?“<sup>27</sup>. Das funktionale Systemkonzept fokussiert damit auf die Eingangsgrößen (Inputs), Ausgangsgrößen (Outputs) und Zustände eines Systems<sup>28</sup>. Diese drei Systemkonzepte schließen einander nach ROPOHL nicht aus sondern stellen vielmehr verschiedene Aspekte eines Systemmodells dar und können in Kombination verwendet werden. Er integriert folglich verschiedene Definitionen vom Systembegriff<sup>29</sup>, die jeweils nur einzelne Aspekte abdecken, zu einer neuen Definition.

---

<sup>20</sup> vgl. Ropohl 2009

<sup>21</sup> vgl. Ropohl 2009

<sup>22</sup> vgl. Mesarovic & Macko 1969

<sup>23</sup> vgl. Ropohl 2009

<sup>24</sup> vgl. Ropohl 2009

<sup>25</sup> vgl. Patzak 1982

<sup>26</sup> vgl. Ropohl 2009, S. 15

<sup>27</sup> vgl. Ashby 1974 und Patzak 1982

<sup>28</sup> vgl. Ehrlenspiel 2009

<sup>29</sup> vgl. Hall & Fagen 1956, Greniewski & Kempisty 1963, Lange 1965, Wintgen 1968, Klir 1972, S. 77

**Definition 2-2: System**<sup>30</sup>

Ein System ist das Modell einer Ganzheit, die

- Beziehungen zwischen Attributen (Inputs, Outputs, Zustände) aufweist.
- aus miteinander verknüpften Teilen bzw. Subsystemen besteht.
- von ihrer Umgebung bzw. von einem Supersystem (durch eine Systemgrenze) abgegrenzt wird.

**2.1.1 Produktentstehung**

Die Zielsetzung eines Unternehmens ist die Entwicklung, Herstellung und der Vertrieb marktfähiger Produkte mit dem Zweck der Gewinnmaximierung. ALBERS versteht im Kontext der Produktentstehung unter dem Begriff Produkt eine Ware oder Dienstleistung, wobei es sich sowohl um ein physisches Erzeugnis als auch um Software handeln kann.<sup>31</sup> Ziel der Produktentstehung ist es, funktionsfähige und produzierbare Produkte zu generieren, die am Markt erfolgreich sind<sup>32</sup>. Folglich schließt der Begriff Produktentstehung die Produktentwicklung ein und beinhaltet darüber hinaus noch die Produktsystementwicklung sowie alle Tätigkeiten, die mit der Vermarktung in Zusammenhang stehen.

Bei der Entwicklung komplexer Produkte ist zunehmend eine Vielzahl interdisziplinärer Problemlösungsteams involviert. Ein Ansatz der in der Forschung sowie in einigen industriellen Branchen<sup>33</sup> bereits weit verbreitet ist und mit den daraus entstehenden Herausforderungen umzugehen versucht ist das Systems Engineering, welchem das Systemdenken zugrunde liegt. Dieser wird im Folgenden vorgestellt.

**2.1.2 Systemmodell der Produktentstehung**

Auf Basis der allgemeinen Systemtheorie leitet ROPOHL<sup>34</sup> einen systematischen Ansatz zur Beschreibung der Produktentstehung ab, mit dem Anspruch, dass das resultierende Modell der Systemtechnik "an alle Erscheinungen und Probleme herangetragen werden [kann], mit denen der Ingenieur konfrontiert wird"<sup>35</sup>.

---

<sup>30</sup> vgl. Ropohl 2009

<sup>31</sup> vgl. Albers & Braun 2011

<sup>32</sup> vgl. Ponn & Lindemann 2011

<sup>33</sup> vgl. Gausemeier et al. 2013, S.11

<sup>34</sup> vgl. Ropohl 2009

<sup>35</sup> vgl. Ropohl 1975

ROPOHL unterteilt das sozio-technische System der Produktentstehung in drei miteinander wechselwirkende Systeme. Nach ROPOHL besteht die Aufgabe von Entwicklern darin, technische Gebilde hervorzubringen. Diese fasst er unter dem Begriff der Sachsysteme zusammen. Diese Tätigkeit, aus der die Sachsysteme hervorgehen, spielt sich im Rahmen komplexer Organisationen ab. Die darin enthaltenen Maßnahmen und Einrichtungen technischer Arbeit ordnet ROPOHL der zweiten Systemklasse, den Handlungssystemen, zu. Die Entwicklungstätigkeit ist an bestimmte Zielvorgaben geknüpft, welche zum Teil aus dem Handlungssystem selbst, zum Teil aus dessen Umgebung stammen. Die Menge der Zielvorgaben bilden das Zielsystem.<sup>36</sup>

Den Zusammenhang von Ziel-, Handlungs- und Sachsystem beschreibt ROPOHL durch einen Regelkreis (vgl. Abbildung 3). Das Handlungssystem verwirklicht gemäß einem Zielsystem ein Sachsystem, wobei das Handlungssystem wiederum auf das Zielsystem und das Sachsystem auf das Handlungssystem zurückwirkt.

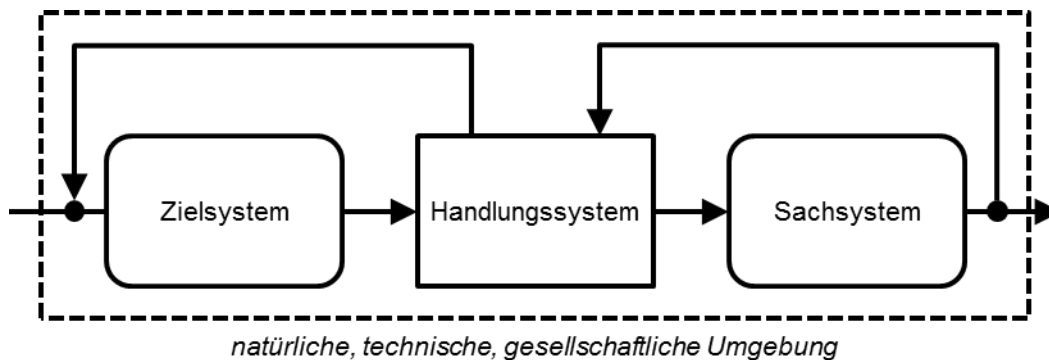


Abbildung 3: Zielsystem, Handlungssystem und Sachsystem nach ROPOHL<sup>37</sup>

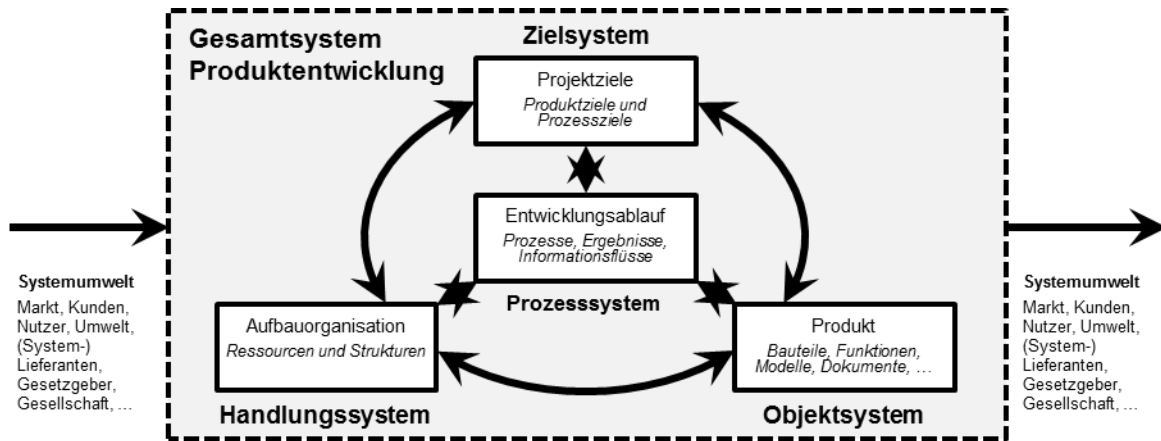
PATZAK<sup>38</sup> unterteilt das Handlungssystem weiter in ein abstraktes Programmsystem (Projekt, Vorhabensplan) und ein konkretes Wirksystem (Organisation, Handlungsträger, Sachmittel). Den Begriff Sachsystem ersetzt er durch den des Objektsystems. Die Technische Universität München greift den Ansatz von PATZAK in den 1990er Jahren auf und definiert das ZOPH-Modell, welches aus Zielsystem, Objektsystem, Prozesssystem und Handlungssystem besteht (vgl. Abbildung 4).

<sup>36</sup> vgl. Ropohl 2009, S.33

<sup>37</sup> vgl. Ropohl 1975

<sup>38</sup> vgl. Patzak 1982, S.139



Abbildung 4: ZOPH-Modell nach NEGELE<sup>39</sup>

ALBERS und MEBOLDT<sup>40</sup> entwickeln aufbauend auf dem Systemmodell nach ROPOHL<sup>41</sup> das integrierte Produktentstehungsmodell iPeM (vgl. Kapitel 2.1.4). Die Basis hierfür bildet ein Systemtripel aus Zielsystem, Handlungssystem und Objektsystem (vgl. Abbildung 5). Im Folgenden werden die Definitionen vorgestellt:

Das **Handlungssystem** ist ein sozio-technisches System und besteht aus strukturierten Aktivitäten, Methoden und Prozessen. Weiterhin enthält es alle für die Realisierung der Produktentstehung erforderlichen Ressourcen wie z.B. (Entwicklungs-)Personal, Budget, Material, Maschinen und weitere. Das Handlungssystem erstellt das Zielsystem und das Objektsystem, die beide jeweils nur über das Handlungssystem miteinander verbunden sind.

Das **Zielsystem** umfasst die mentale Vorstellung der geplanten Eigenschaften eines Produkts. Es umfasst alle relevanten Ziele, deren Wechselwirkungen und Randbedingungen, die für die Entwicklung des richtigen Produkts erforderlich sind. Das Zielsystem wird im Verlauf des Produktentstehungsprozesses fortwährend erweitert und konkretisiert.

Das **Objektsystem** enthält alle Dokumente und Artefakte, die als Teillösungen während des Entstehungsprozesses anfallen. Das eigentliche Produkt ist neben Zwischenprodukten (z.B. Zeichnungen oder Prototypen) eines der Elemente des Objektsystems.

<sup>39</sup> vgl. Negele 1998

<sup>40</sup> vgl. Albers & Meboldt 2007

<sup>41</sup> vgl. Ropohl 1975

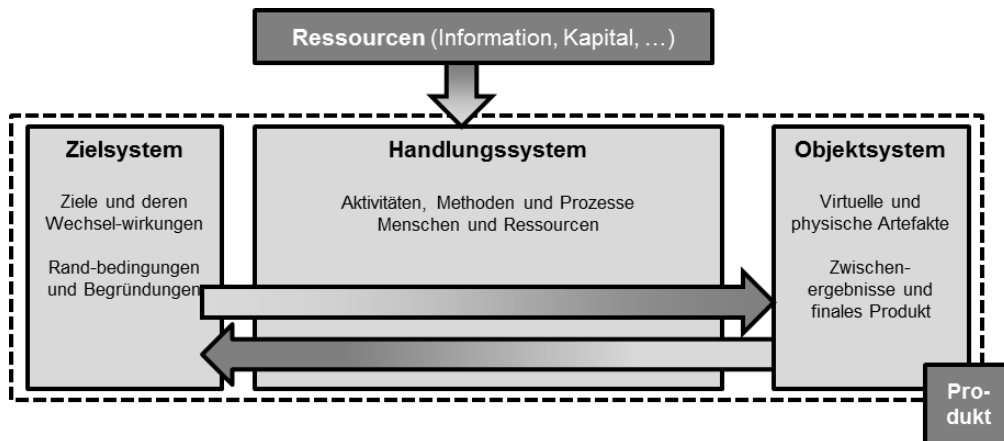


Abbildung 5: ZHO-Modell nach ALBERS<sup>42</sup>

ALBERS und LOHMEYER<sup>43</sup> liefern mit dem erweiterten ZHO-Modell ein menschenzentriertes Erklärungsmodell für die Produktentstehung, welches den Entwickler als denkenden und handelnden Menschen im Mittelpunkt eines unsicherheitsbehafteten und iterativ verlaufenden Produktentstehungsprozess beschreibt (vgl. Abbildung 6).

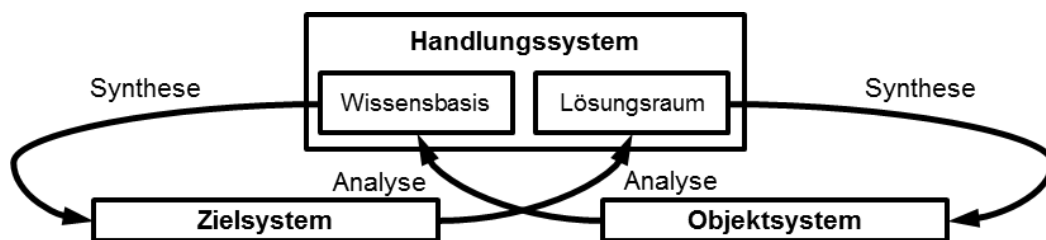


Abbildung 6: Das erweiterte ZHO-Modell<sup>44</sup>

Unsicherheiten im Produktentstehungsprozess führen zu Fehleinschätzungen, Fehlentscheidungen und schlussendlich zu Änderungen und Iterationen<sup>45</sup>. Die Konkretisierung eines zu entwickelnden Systems verläuft daher selten linear ab, sondern in Form von Iterationen<sup>46</sup>. LOHMEYER<sup>47</sup> versteht in diesem Kontext unter Iteration das vollständige Durchlaufen der folgenden vier Schritte (vgl. Abbildung 6):

1. Analyse des Zielsystems
2. Synthese des Objektsystems
3. Analyse des Objektsystems
4. Synthese des Zielsystems

<sup>42</sup> vgl. Albers & Braun 2011

<sup>43</sup> vgl. Albers & Lohmeyer 2012

<sup>44</sup> vgl. Albers, Lohmeyer & Ebel 2011

<sup>45</sup> vgl. Ebel 2015

<sup>46</sup> vgl. Lohmeyer 2013

<sup>47</sup> vgl. Lohmeyer 2013

MEBOLDT, MATTHIESEN UND LOHMEYER unterscheiden zwischen In-Stage- und Cross-Stage-Iteration (vgl. Abbildung 7).<sup>48</sup> Eine In-Stage-Iteration betrifft nur Entscheidungen, die im Rahmen eines bzw. des aktuellen Stages<sup>49</sup> liegen. Projektkosten und Projektdurchlaufzeit werden nicht negativ beeinflusst. Eine Cross-Stage-Iteration betrifft hingegen Entscheidungen, die in früheren Stages getroffen wurden. Auch sie führen zu Erkenntnisgewinn, sind jedoch mit Zeit- und Kostenaufwänden verbunden. Das Zielsystem wird hier in der Regel nicht verfeinert sondern in weiten Teilen geändert, wodurch Änderungsbedarf im Ziel- und insbesondere Objektsystem entsteht.

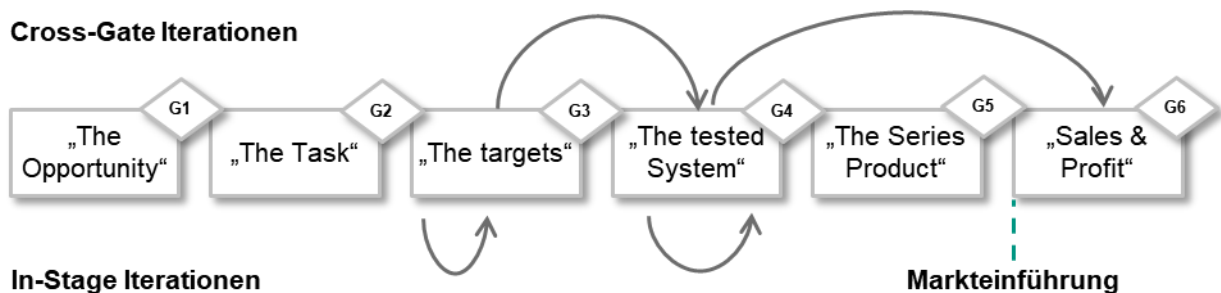


Abbildung 7: In-Stage und Cross-Stage Iterationen in der Produktentstehung<sup>50</sup>

ALBERS und LOHMEYER<sup>51</sup> fokussieren in der Betrachtung des Handlungssystems auf das Individuum Entwickler und verkürzen dieses auf seine Wissensbasis und seinen Lösungsraum. LOHMEYER<sup>52</sup> definiert eine Wissensbasis als „die Menge an Domänenwissen und fallspezifischem Wissen, welches innerhalb des Handlungssystems eines bestimmten Produktentstehungsprozesses bereitsteht.“ Der Lösungsraum „beschreibt die Menge aller zulässigen Lösungen zu einem Problem und richtet so die mentale, virtuelle und physische Modellierung von Objekten aus.“<sup>53</sup> Lohmeyer beschreibt die Wechselwirkungen zwischen den Subsystemen Wissensbasis und Lösungsraum mit dem Ziel- und Objektsystem durch Analyse und Synthese-Schritte, welche er wie folgt definiert:

<sup>48</sup> vgl. Meboldt, Matthiesen & Lohmeyer 2012

<sup>49</sup> vgl. Cooper 1990

<sup>50</sup> vgl. Meboldt, Matthiesen & Lohmeyer 2012

<sup>51</sup> vgl. Albers & Lohmeyer 2012

<sup>52</sup> vgl. Lohmeyer 2013

<sup>53</sup> vgl. Lohmeyer 2013, S. 108

**Definition 2-3: Analyse**<sup>54</sup>

Die Analyse beschreibt eine Handlung, die das Verstehen eines existierenden Systems bezweckt. Das Resultat einer Analyse ist Erkenntnis.

**Definition 2-4: Synthese**<sup>55</sup>

Die Synthese beschreibt eine Handlung, die das Erschaffen eines bis dato noch nicht existierenden Systems bezweckt. Das Resultat einer Synthese ist ein Ziel oder ein Objekt.

**2.1.3 Produktgenerationsentwicklung**

PAHL und BEITZ unterscheiden Produktentwicklungsprojekte in den folgenden Kategorien<sup>56</sup>:

- **Neukonstruktion:** Neue Aufgabenstellungen und Probleme werden mit neuen Lösungsprinzipien adressiert. Die technischen Lösungen können sowohl durch die Auswahl und Kombination bekannter Prinzipien sowie durch neuartige Lösungsprinzipien umgesetzt werden.
- **Anpassungskonstruktion:** Hier werden bekannte und bewährte Lösungsprinzipien verwendet und die Gestalt der Systeme lediglich an die geänderten Randbedingungen angepasst.
- **Variantenkonstruktion:** Die Größe und Anordnung von Systemen wird innerhalb von Grenzen variiert. Der Aufwand zur initialen Erstellung ist erhöht. Jedoch kann anschließend bei gleichbleibendem Lösungsprinzip durch Veränderung gezielter Parameter die Lösungsvielfalt erhöht werden.

Zur Differenzierung werden der Neuheitsgrad, die Unsicherheit bezüglich der vorherrschenden Randbedingungen und die Möglichkeit zur Verwendung bekannter und beherrschter Lösungsprinzipien herangezogen, wobei sich die Kategorien ihrer Ansicht nach dennoch nicht scharf abgrenzen lassen.<sup>57</sup>

In der Literatur wird weiter vermehrt darauf verwiesen, dass reine Neukonstruktionen selten vorkommen<sup>58</sup>. Stattdessen wird aus ökonomischen und risikoanalytischen

<sup>54</sup> vgl. Lohmeyer 2013, S. 108

<sup>55</sup> vgl. Lohmeyer 2013

<sup>56</sup> vgl. Pahl et al. 2007

<sup>57</sup> vgl. Pahl et al. 2013

<sup>58</sup> vgl. Albers, Bursac und Wintergerst 2015

Gründen das Ziel verfolgt, die angestrebten Funktionen und Eigenschaften eines neuen Produkts mit möglichst geringfügigen Modifikationen von bestehenden Lösungen umzusetzen<sup>59</sup>. ECKERT ET AL ergänzen somit folgerichtig, dass die Verbesserung bestehender Produkte als häufigste Art der Produktentwicklung anzusehen ist<sup>60</sup>.

Nach ALBERS<sup>61</sup> können Produkte häufig erst nach diversen Variationen über mehrere Produktgenerationen hinweg wirtschaftlich und technisch dahingehend entwickelt werden, dass sie erfolgreich auf dem Markt etabliert werden können.

KANO<sup>62</sup> hat erkannt, dass Produkte unterschiedliche Arten von Attributen aufweisen, die sogenannten Basis-, Leistungs- und Begeisterungsattribute. Über mehrere Generationen können Begeisterungsattribute zu Leistungs- und Basisattributen degradieren<sup>63</sup>. Um ein Produkt weiterhin erfolgreich am Markt zu platzieren, werden daher regelmäßig neue Begeisterungsattribute benötigt.<sup>64</sup>

Hier entsteht folglich ein Zielkonflikt, denn aus ökonomischen und risikominimierenden Gründen sollen möglichst wenige neue (Sub-)Systeme entwickelt werden. Der Anspruch an ein innovatives Produkt mit möglichst guten Leistungs- und neuen Begeisterungsattributen ist dennoch hoch.<sup>65</sup> Der Ansatz der Produktgenerationsentwicklung nach ALBERS<sup>66</sup> bietet eine Möglichkeit, um mit diesen Herausforderungen umzugehen.

---

<sup>59</sup> vgl. Deubzer & Lindemann 2009

<sup>60</sup> vgl. Eckert, Alink & Albers 2010

<sup>61</sup> vgl. Albers 1991

<sup>62</sup> vgl. Kano et al. 1984

<sup>63</sup> vgl. Albers, Bursac & Wintergerst 2015

<sup>64</sup> vgl. Bailom et al. 1996

<sup>65</sup> vgl. Albers et al. 2014a

<sup>66</sup> vgl. Albers, Bursac & Wintergerst 2015

**Definition: Produktgenerationsentwicklung<sup>67</sup>**

Die Produktgenerationsentwicklung ist die Entwicklung einer neuen Generation eines technischen Produkts, die sowohl durch die Anpassung von Teilsystemen als Übernahmevariation als auch durch eine signifikante Neuentwicklung von Teilsystemen charakterisiert ist. Neu entwickelte Anteile technischer Produkte können durch Gestalt und durch Prinzipvariation realisiert werden. Neue Produktgenerationen basieren immer auf Referenzprodukten, welche die grundsätzliche Struktur beschreiben.

In diesem Kontext kann ein technisches Produkt neben einem physischen Produkt auch eine Software oder Dienstleistung sein. Die Entwicklung einer neuen Produktgeneration baut dabei immer auf mindestens einem bereits vorhandenen Referenzprodukt auf. Dadurch können Informationen und einzelne Teilsysteme oder Entwicklungsartefakte bei der Entwicklung neuer Produktgenerationen wieder verwendet werden.

**Definition: Referenzprodukt<sup>68</sup>**

Referenzprodukte sind bestehende Produkte (z. B. Vorgänger- oder Wettbewerbsprodukte), auf deren Basis neue Produktgenerationen entwickelt werden. Dazu werden teilweise deren Struktur und Teilsysteme übernommen oder als Ausgangspunkt für Variationen zugrunde gelegt.

Im Rahmen der PGE werden Basisattribute mit möglichst geringen Veränderungen in neue Produktgenerationen übernommen um gleichzeitig die verfügbaren Ressourcen auf die Entwicklung von Leistungs- und Begeisterungsattributen als Differenzierungsmerkmale zu konzentrieren. Dadurch kann das Innovationspotential eines Produktes bei vertretbarem Risiko gesteigert werden.<sup>69</sup>

---

<sup>67</sup> vgl. Albers, Bursac & Wintergerst 2015

<sup>68</sup> vgl. Albers, Bursac & Wintergerst 2015 und Albers, Bursac & Rapp 2017

<sup>69</sup> vgl. Albers et al. 2014a

ALBERS unterscheidet folglich drei Arten von Variationen:<sup>70</sup>

**Prinzipvariation:** Die Neuentwicklung eines Subsystems durch Prinzipvariation ist dadurch gekennzeichnet, dass das Lösungsprinzip im Vergleich zum Referenzprodukt verändert wird. Diese Subsysteme werden als die Menge jener Teilsysteme definiert, die durch Prinzipvariation neu entwickelt werden.

$$(PS_{n+1} \{TS|PV_{(TS)}\})$$

**Gestaltvariation:** Die Neuentwicklung eines Subsystems durch Gestaltvariation ist dadurch gekennzeichnet, dass das Lösungsprinzip des Referenzprodukts beibehalten wird und stattdessen die Gestalt verändert wird. Diese Subsysteme werden als die Menge jener Teilsysteme definiert, die durch Gestaltvariation neu entwickelt werden.

$$(GS_{n+1} \{TS|GV_{(TS)}\})$$

**Übernahmevariation:** Die Anpassung eines Subsystems durch Übernahmevariation ist dadurch gekennzeichnet, dass die Gestalt und das Lösungsprinzip des Referenzprodukts beibehalten und die konstruktiven Anpassungen minimiert werden.

$$(\ddot{U}S_{n+1} \{TS|\ddot{U}V_{(TS)}\})$$

Demnach setzt sich eine neue Generation ( $G_{n+1}$ ) nach ALBERS aus der Vereinigung der Mengen der drei Teilsysteme zusammen.

$$G_{n+1} = \ddot{U}S_{n+1} \cup GS_{n+1} \cup PS_{n+1}$$

Der Übernahmevarianteanteil einer Produktgeneration ist damit definiert als:

$$\delta_{\ddot{U}V_{n+1}} = \frac{|\ddot{U}S_{n+1}|}{|G_{n+1}|} = \frac{|\ddot{U}S_{n+1}|}{|\ddot{U}S_{n+1} \cup GS_{n+1} \cup PS_{n+1}|} \quad [\%]$$

Analog können die Anteile neu entwickelter Teilsysteme mit Gestaltvariation und neuentwickelter Teilsysteme mit Prinzipvariation wie folgt berechnet werden:

$$\delta_{GV_{n+1}} = \frac{|GS_{n+1}|}{|\ddot{U}S_{n+1} \cup GS_{n+1} \cup PS_{n+1}|} \quad [\%]; \quad \delta_{PV_{n+1}} = \frac{|PS_{n+1}|}{|\ddot{U}S_{n+1} \cup GS_{n+1} \cup PS_{n+1}|} \quad [\%]$$

Dieses mathematische Modell der Produktgenerationsentwicklung bietet große Potentiale, diese zukünftig für die Steuerung des Entwicklungsprozesses und -risikos zu verwenden.<sup>71</sup>

<sup>70</sup> vgl. Albers, Bursac & Wintergerst 2015

### 2.1.4 Produktentstehungs- und Entwicklungsprozesse

Um mit der Komplexität und Unsicherheit in den Produktentstehungsprozessen umgehen zu können, bedarf es entsprechender Modelle der Prozesse, um die realen Abläufe verstehen, planen, steuern und verbessern zu können<sup>72</sup>. Ein Produktentstehungsprozess (PEP) beschreibt die Arbeitsabläufe von der Idee für ein neues Produkt bis zu dessen Herstellung und Verkauf.

Nach ALBERS<sup>73</sup> sind zur ganzheitlichen Modellierung von Produktentstehungsprozessen die einzelnen Modelle verschiedener Auflösungsgrade erforderlich, welche auf Basis der Systemtheorie kontinuierlich und konsistent ineinander überführt werden müssen.

LINDEMANN<sup>74</sup> unterscheidet in vier Auflösungsgrade von Produktentstehungsprozessen und ordnet diese zwischen den Polen Mikrologik und Makrologik ein (vgl. Abbildung 8). Die Grenzen zwischen diesen vier Auflösungsgraden sind dabei nicht scharf definiert und einzelne Prozessmodelle sind teilweise in der Lage, mehrere Auflösungsgrade abzudecken.

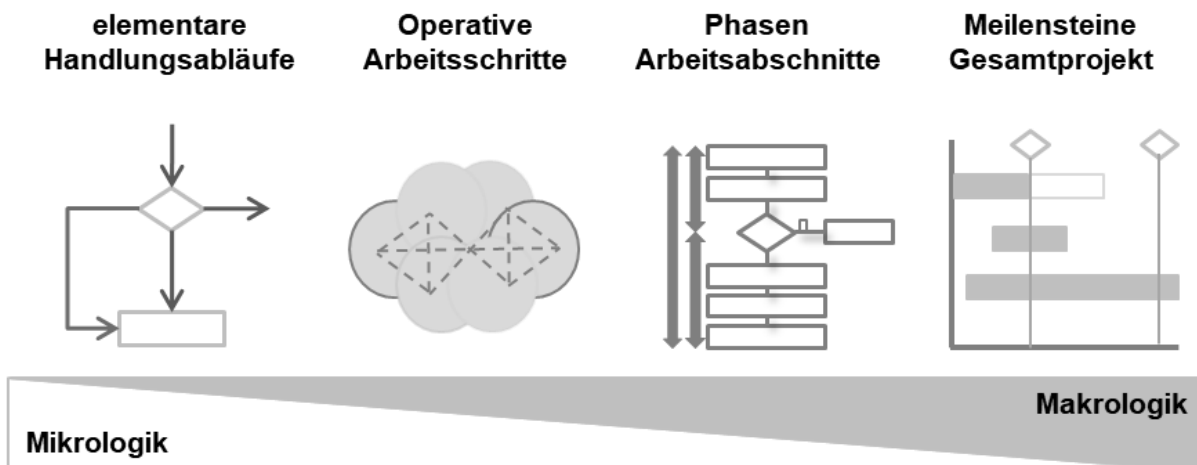


Abbildung 8: Auflösungsgrade des Produktentstehungsprozesses<sup>75</sup>

Die Prozessmodelle der Mikrologik beschreiben nach LINDEMANN<sup>76</sup> elementare Denk- und Handlungsabläufe einzelner Individuen. Hierbei handelt es sich um Zyklen der Analyse, Synthese und Bewertung, die sich im Gehirn der Individuen abspielen. Prozessmodelle der Makrologik abstrahieren die Komplexität der tatsächlichen

<sup>71</sup> vgl. Albers, Walch & Bursac 2016

<sup>72</sup> vgl. Negele et al. 1999

<sup>73</sup> vgl. Albers & Lohmeyer 2012, S. 38

<sup>74</sup> vgl. Lindemann 2009, S. 38

<sup>75</sup> vgl. Lindemann 2009, S. 38

<sup>76</sup> vgl. Lindemann 2009, S. 35f



Abläufe bei ganzen Produktentstehungsprojekten zu einem niedrigen Auflösungsgrad. Dazwischen sind Prozessmodelle angesiedelt, die einzelne operative Arbeitsschritte abbilden (Problemlösungsprozesse) und zu Phasen bzw. Arbeitsabschnitten zusammenfassen.

Basierend auf den Auflösungsgraden von LINDEMANN gliedert EBEL<sup>77</sup> die Produktentstehungsprozesse weiter in folgende vier Bereiche:

- Mentale Modelle und Modelle elementarer Problemlösung
- Modelle operativer Problemlösung
- Vorgehens- und Phasenmodelle der Produktentstehung
- Übergreifende und ganzheitliche Modelle der Produktentstehung

### Mentale Modelle und Modelle elementarer Problemlösung

Modelle elementarer Problemlösung haben zum Ziel, individuelle und implizite Problemlösungsprozesse zugänglich zu machen.<sup>78</sup> Sie fokussieren damit auf einzelne Individuen. Das TOTE-Modell<sup>79</sup> und der PDCA-Zyklus<sup>80</sup> sind Vertreter von Modellen zur Beschreibung der Problemlösungsprozesse auf elementarer Ebene (vgl. Abbildung 9).

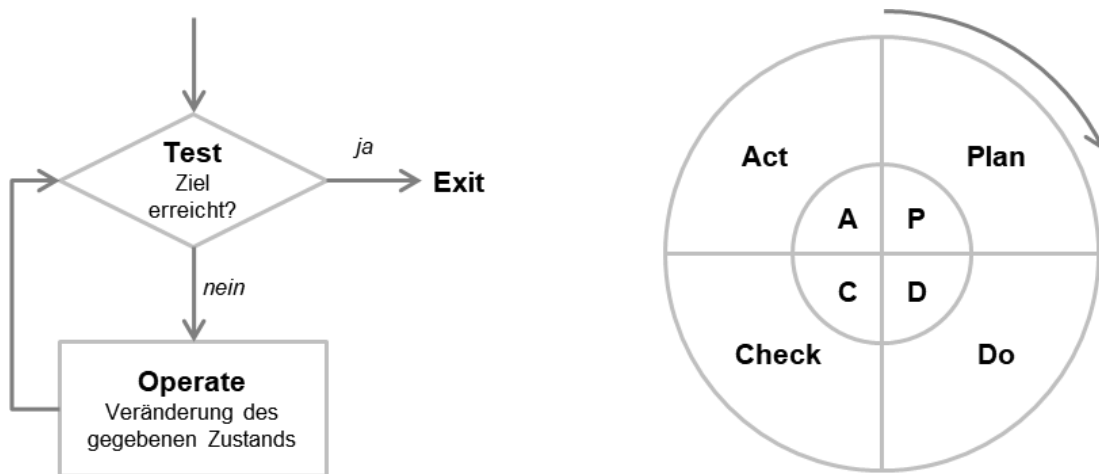


Abbildung 9: TOTE-Modell und PDCA-Zyklus

Das TOTE-Modell hat ihren Ursprung in der Kybernetik. Es beschreibt einen Problemlösungsprozess als Regelkreis aus Testen („Test“) und Handeln („Operate“) welcher solange durchlaufen wird, bis ein definierter Sollzustand (oder ein Abbruchkriterium) erreicht ist.

<sup>77</sup> vgl. Ebel 2015, S. 35

<sup>78</sup> vgl. Ebel 2015, S. 26

<sup>79</sup> vgl. Miller, Ganater & Pribram 1960, S. 88

<sup>80</sup> vgl. Deming 1982

Der PDCA-Zyklus entstammt der Qualitätssicherung und besteht aus den vier Schritten Planen („plan“), Umsetzen („do“), Überprüfen („check“) und Handeln („act“).

### **Modelle operativer Problemlösung**

Die Modelle operativer Problemlösung beschreiben ein systematisches Vorgehen zur Lösung von Problemen in jeder Projektphase durch Vorgabe einzelner Arbeitsschritte.<sup>81</sup> Viele Modelle basieren auf dem Dewey'schen Problemlösungsmodell, weshalb Ähnlichkeiten bestehen. Prinzipiell lassen sich alle Modelle operativer Problemlösung auf drei Hauptschritte abstrahieren:<sup>82</sup>

- Ziel beziehungsweise Problem klären: Wo stehen wir? Was wollen/brauchen wir? Warum?
- Lösungsalternativen generieren: Welche Möglichkeiten zur Lösung gibt es?
- Auswahl: Welche ist die beste/zweckmäßigste?

Nach HABERFELLNER<sup>83</sup> gibt es Modelle, welche deutliche Akzente hinsichtlich der Reihenfolge des Vorgehens setzen. Er gliedert die Modelle in Ist-Zustands-orientierte und Soll-Zustands-orientierte Modelle. Zu den Ist-Zustands-orientierten Vorgehensmodellen zählt vor allem die weitverbreitete „klassische Organisationsmethodik“, welche durch die drei Schritte (1) Aufnahme des Ist-Zustandes, (2) Kritik am Ist-Zustand und (3) Erarbeitung des Soll-Zustandes gekennzeichnet ist. Die Soll-Zustandsorientierten Vorgehensmodellen stellen eine radikale Abwendung von der oben beschriebenen Vorgehensweise dar, denn der Ist-Zustand ist für die Erarbeitung einer Lösung zunächst nur von untergeordneter Bedeutung<sup>84</sup>.

Die SPALTEN-Problemlösungsmethode nach ALBERS<sup>85</sup> ist eine Vorgehensweise zur operativen Problemlösung unterschiedlicher Randbedingungen und Komplexitätsgrade und ist zwischen den beiden skizzierten Extremformen anzusiedeln. SPALTEN ist einerseits als ein Akronym aufgebaut. Dadurch werden wenige Segmente verwendet. Das erleichtert den Produktentwicklern ein stringentes Vorgehen und die Konzentration auf die Problemlösung. Des weiteren wird das Nachbereiten und Lernen in den Fokus gerückt, was insbesondere bei der Entwicklung von Zielsystemen zentral ist, um abbilden zu können wie die Ziele zustanden kommen. Daher wird im Rahmen der vorliegenden Arbeit die SPALTEN

---

<sup>81</sup> vgl. Ebel 2015 und Lindemann 2009

<sup>82</sup> vgl. Haberfellner, Weck & Fricke 2012

<sup>83</sup> vgl. Haberfellner, Weck & Fricke 2012, S. 83

<sup>84</sup> vgl. Haberfellner, Weck & Fricke 2012

<sup>85</sup> vgl. Albers et al. 2005

Problemlösungsmethodik verwendet. Weitere weit verbreitete Vertreter sind der Problemlösungszyklus der Systemtechnik nach DAENZER und HUBER<sup>86</sup> und das Münchner Vorgehensmodell<sup>87</sup>.

SPALTEN besteht aus sieben Schritten, die im Problemlösungsprozess sequenziell angewendet werden können, wobei wie im Prozessmodell sichtbar einzelne Schritte übersprungen werden können (vgl. Abbildung 10).

**Situationsanalyse (SA):** Vorbereitende Informationserfassung über Soll- und Ist-Zustand sowie verknüpfte Randbedingungen.

**Problemeingrenzung (PE):** Untersuchung der gesammelten Information zur Eingrenzung des Kerns der weiteren Betrachtung. Ziel ist es, die Ursache und Wirkung der Soll-Ist-Abweichung eindeutig zu ermitteln. Darauf aufbauend werden Entscheidungskriterien definiert.

**Alternative Lösungssuche (AL):** In diesem Schritt werden mögliche Lösungen bzw. Handlungsalternativen für die Überwindung des Delta zwischen Soll- und Ist-Zustand entwickelt.

**Lösungsauswahl (LA):** Vergleich der identifizierten Handlungsalternativen und Auswahl einer umzusetzenden Lösung nach den zuvor definierten Kriterien.

**Tragweitenanalyse (TA):** Systematische Untersuchung und Abwägung von Chancen und Risiken, die mit der getroffenen Lösungsauswahl verbunden sind. Gegebenenfalls wird ein Maßnahmenplan definiert. Durch einen Perspektivenwechsel geht man davon aus, dass die Lösung bereits umgesetzt worden ist und versucht, die Konsequenzen zu analysieren.

**Entscheiden und Umsetzen (EU):** Beschluss zur Lösungsumsetzung und Implementierung der Lösung durch die Verantwortlichen.

**Nacharbeiten und Lernen (NL):** Reflektion des Problemlösungsprozesses und gegebenenfalls Festhalten von Erkenntnissen für zukünftige Prozesse im Sinne eines kontinuierlichen Verbesserungsprozess (KVP).

Durch einen **Informationscheck (IC)** wird nach jedem Schritt überprüft, ob die Informationsbasis ausreichend genutzt wird und ob diese ausreichend ist, um den nächsten Schritt durchzuführen. Des Weiteren wird nach jedem Schritt geprüft, ob das Problemlösungsteams (PLT) hinsichtlich seiner Eignung für die vorliegenden Aufgaben (noch) optimal zusammengesetzt ist.

---

<sup>86</sup> vgl. Daenzer & Huber 1996

<sup>87</sup> vgl. Lindemann 2009

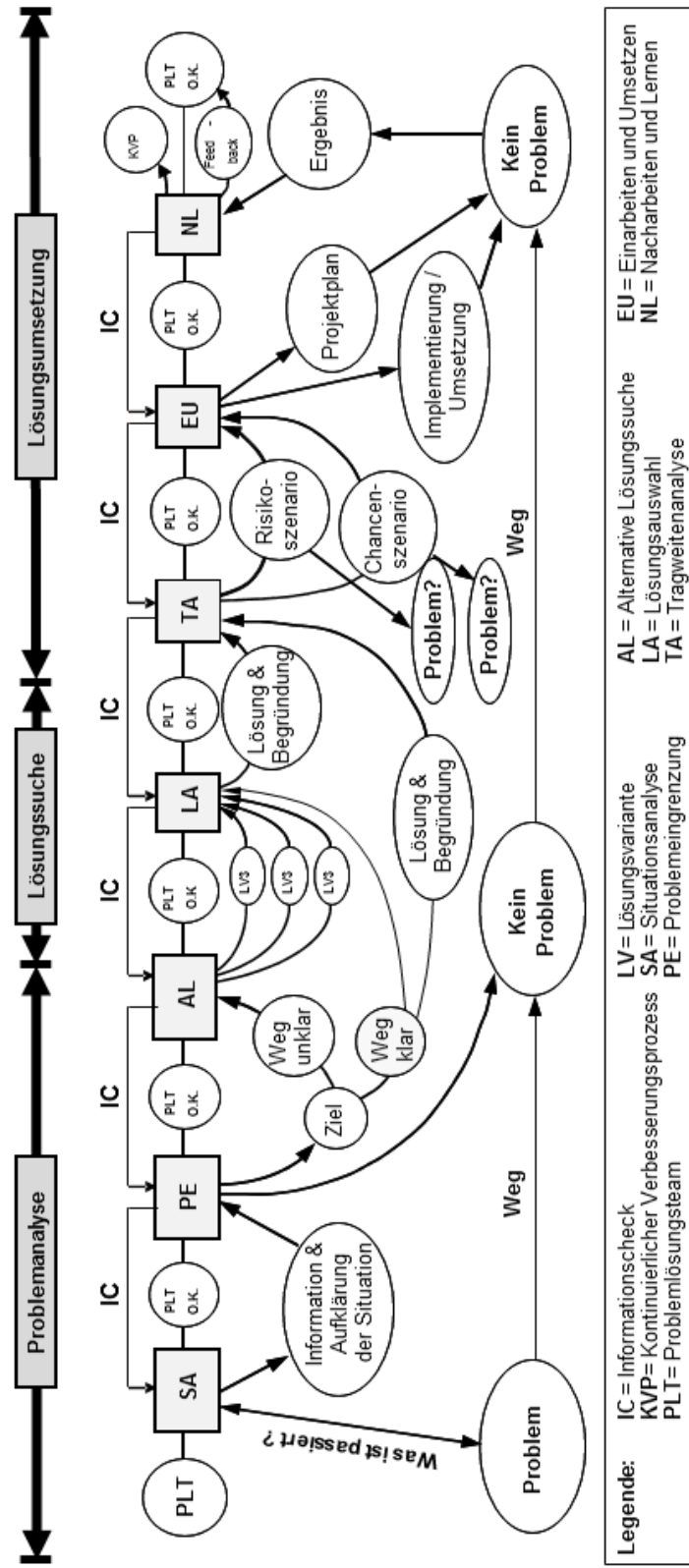


Abbildung 10: SPALTEN-Problemlösungsmethode<sup>88</sup>

<sup>88</sup> vgl. Albers et al. 2005

Die SPALTEN-Problemlösungsmethode hat fraktalen Charakter. Das bedeutet, dass jeder einzelne SPALTEN-Schritt wieder durch einen separaten SPALTEN-Prozess bearbeitet werden kann<sup>89</sup>.

### Vorgehens- und Phasenmodelle der Produktentstehung

Vorgehens- und Phasenmodelle bilden die Makrologik aus. Sie strukturieren den Produktentstehungsprozess in inhaltlich und zeitlich abtrennbare Arbeitsabschnitte bzw. Phasen.<sup>90</sup> Innerhalb der einzelnen Arbeitsabschnitte bzw. Phasen können die Modelle der elementaren und operativen Problemlösung Anwendung finden. BOEHM und TURNER<sup>91</sup> unterscheiden zwischen sogenannten „plan-driven“ und „agility-driven methods“. „Plan-driven“ Methoden geben eine klare Schrittfolge vor. Zu den bekanntesten Vertretern gehört das Wasserfallmodell, der Stage-Gate Prozess nach COOPER<sup>92</sup> und das V-Modell nach VDI Norm 2221<sup>93</sup>. Weitere Vertreter der „plan-driven methods“ sind VDI 2221, der SIMILAR-Process, IEEE-15288 und das Simultaneous (Concurrent) Engineering, welche für diese Arbeit nicht relevant sind und daher nicht weiter erläutert werden.

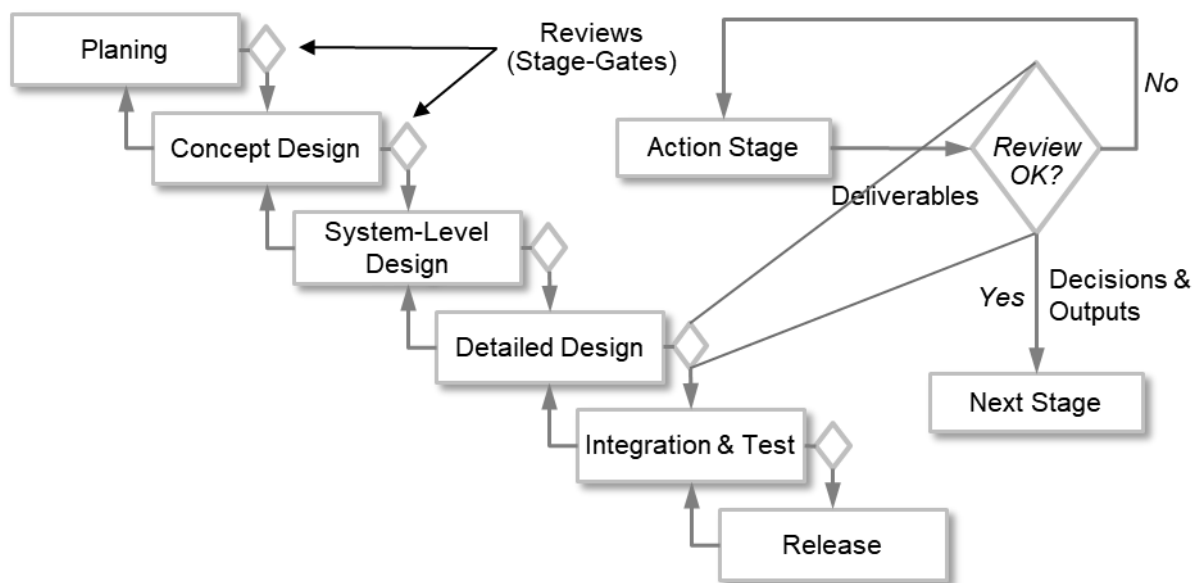


Abbildung 11: Der Stage-Gate-Prozess<sup>94</sup>

<sup>89</sup> vgl. Albers, Braun & Muschik 2010, S. 41

<sup>90</sup> vgl. Ebel 2015

<sup>91</sup> vgl. Boehm & Turner 2004

<sup>92</sup> vgl. VDI 2221, Cooper 1990

<sup>93</sup> vgl. VDI 2221

<sup>94</sup> vgl. Cooper 1990, S. 46 & S. 62

Der **Stage-Gate-Ansatz** ist ein managementorientiertes Modell von Produktentstehungsprozessen (vgl. Abbildung 11).<sup>95</sup> In einem Phasenmodell werden Aktivitäten entlang einer Zeitachse dargestellt. Phasen (engl. stages) werden durch Meilensteine (engl. gates) voneinander getrennt. Zu den Meilensteinen erfolgen Projektreviews und in einem Prozessmodell wird definiert, welche Ergebnisse zu einem Zeitpunkt vorliegen sollten, um die nächste Phase zu erreichen. Neben zeitlicher Verzögerungen bei der Durchführung einzelner Phasen (z.B. sehr spätes Testen) kann an diesem Ansatz die Konzentration auf ausschließlich technische Aspekte kritisiert werden.

Das V-Modell beschreibt ein generisches und systematisches Vorgehen zum Entwurf Mechatronischer Systeme.<sup>96</sup> Es untergliedert den Entwicklungsprozess in drei übergeordnete Phasen (vgl. Abbildung 12):

- Domänenübergreifender Systementwurf
- Domänenspezifischer Entwurf
- Systemintegration

Zunächst werden die Anforderungen an das Produkt definiert und Testfälle für die Validierung abgeleitet. Im domänenübergreifenden Systementwurf werden dann auf Basis der formulierten Anforderungen die wesentlichen physikalischen und logischen Wirkungsweisen des Produktes festgelegt und spezifiziert. Dabei wird die Gesamtfunktion in Teilfunktionen untergliedert („partitioniert“) und Wirkprinzipien zugeordnet. Im nächsten Schritt werden im Rahmen eines domänenspezifischen Entwurfs die Lösungskonzepte in den Domänen Maschinenbau, Elektrotechnik und Informationstechnik konkretisiert. In der Systemintegration werden die Ergebnisse der Teilsysteme zu einem Gesamtsystem integriert und deren Zusammenwirken anhand der Anforderungen überprüft („Eigenschaftsabsicherung“).

Zur Ermittlung und Absicherung der Systemeigenschaften werden die drei übergeordneten Phasen von Modellbildungs- und Modellanalysetätigkeiten begleitet. Für die Entstehung eines komplexen Mechatronischen Erzeugnisses sind dabei in der Regel mehrere Durchläufe erforderlich.

Die Darstellung der Entwicklung im V-Modell als einen starren und sequenziellen Prozess widerspricht vielen Forschungsarbeiten in der Produktentstehung. MUSCHIK<sup>97</sup> stellt beispielsweise fest, dass Anforderungen zu Projektbeginn aufgrund hoher Unsicherheiten nicht vollständig und in ausreichendem Reifegrad abgeleitet

---

<sup>95</sup> vgl. Cooper 1990, S. 46 & S. 62

<sup>96</sup> vgl. VDI 2221 und Unger 2003

<sup>97</sup> vgl. Muschik 2011

werden können. ALBERS<sup>98</sup> beschreibt, dass ein zunächst vages initiales Zielsystem während des Produktentstehungsprozesses kontinuierlich weiterentwickelt wird. Die Validierung sollte dabei nicht auf spätere Phasen verlagert und von der Entwicklung getrennt werden, sondern in einer iterativen Vorgehensweise berücksichtigt werden.<sup>99</sup>

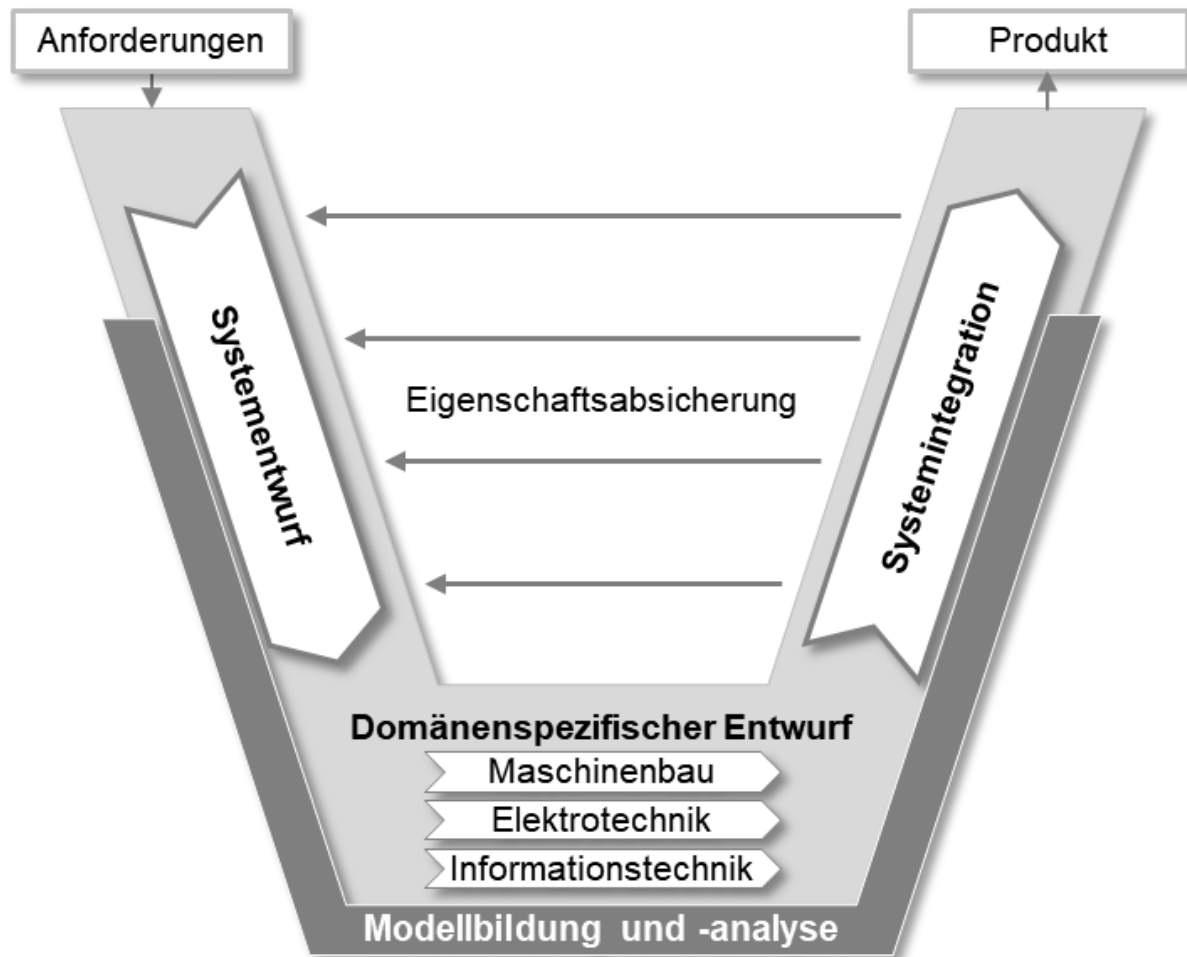


Abbildung 12: V-Modell nach VDI 2221<sup>100</sup>

„Plan-driven methods“ werden insbesondere in Zusammenhang mit großen IT-Projekten dahingehend kritisiert, dass sie den Entwicklungsprozess unnötig schwerfällig machen, lange Entwicklungszeiten erfordern und ein eher restriktives Verhalten gegenüber Änderungen der Spezifikation fördern würden<sup>101</sup>. Aus diesem Bedürfnis heraus haben sich die sogenannten agilen Methoden entwickelt. Popularität erreichten die agilen Methoden erstmals 1999 mit Kent Becks Buch zu

<sup>98</sup> vgl. Albers 2010

<sup>99</sup> vgl. VDI 2221 und Albers 2010, S. 40

<sup>100</sup> vgl. VDI 2221

<sup>101</sup> vgl. Habermann, Weck & Fricke 2012

eXtreme Programming<sup>102</sup>. Im Februar 2001 haben 17 Erstunterzeichner die sogenannten agilen Werte im Agilen Manifest<sup>103</sup> formuliert, welche bis heute als das Fundament der agilen Vorgehensmodelle gilt:

- „Individuen und Interaktionen mehr als Prozesse und Werkzeuge.“
- „Funktionierende Software mehr als umfassende Dokumentation.“
- „Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung.“
- „Reagieren auf Veränderung mehr als das Befolgen eines Plans.“

Zu den wichtigsten Vertretern der agilen Methoden gehört das Vorgehensmodell Scrum<sup>104</sup>, welches für Projekt- und Produktmanagement Anwendung findet. Weitere wichtige Vertreter der agilen Prozessmodelle sind das Spiralmodell<sup>105</sup>, eXtreme Programming<sup>106</sup>, Feature Driven Development (FDD)<sup>107</sup>, und Crystal<sup>108</sup>.

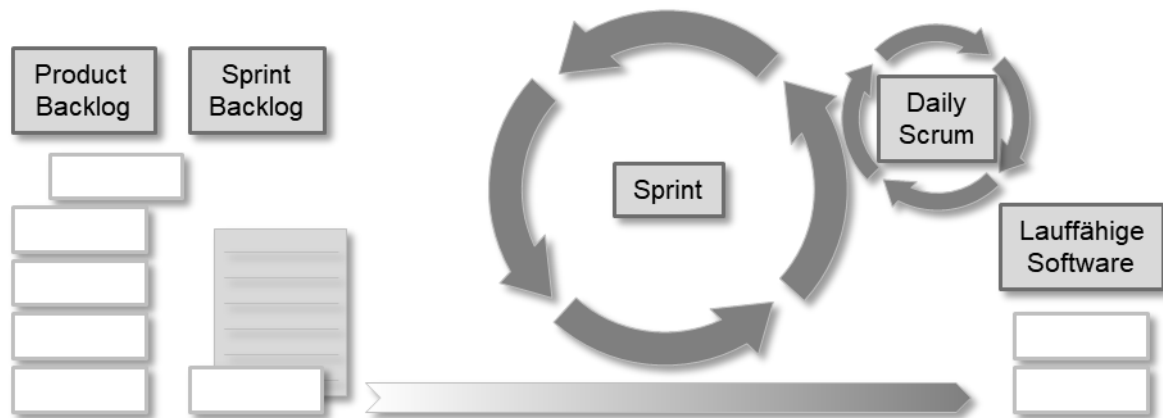


Abbildung 13: Das SCRUM-Vorgehensmodell<sup>109</sup>

Die Intention von Scrum ist es, zu Beginn noch unklare Ziele, Anforderungen und Lösungsansätze sukzessive zu konkretisieren. Im Zielsystem des Produkts (bei Scrum „Product Backlog“ genannt) werden Ziele priorisiert und einzelnen Entwicklungszyklen („Sprints“) zugeordnet. Ein Sprint dauert in der Regel zwischen 2 und 4 Wochen und hat zum Ziel, ein lauffähiges Inkrement des Produkts dem Kunden zur Verfügung zu stellen. Zur Realisierung des Product Backlog werden Anforderungen (sogenannte „User Stories“) identifiziert und einer Art Projektplan („Sprint Backlog“) eingegliedert. Scrum sieht eine kurze tägliche Besprechung

<sup>102</sup> vgl. Beck 1999

<sup>103</sup> vgl. <http://agilemanifesto.org/iso/de/>, Letzter Zugriff am 26.06.2016

<sup>104</sup> vgl. Schwaber 2004

<sup>105</sup> vgl. Boehm 1998

<sup>106</sup> vgl. Beck 1999

<sup>107</sup> vgl. Palmer 2005

<sup>108</sup> vgl. Cockburn 2005

<sup>109</sup> vgl. Schwaber 2004



(ca. 15 Minuten) im Entwicklerteam vor („Daily Scrum“). Es dient dem Team dazu, sich abzustimmen und gegenseitig über laufende Aktivitäten oder Hindernisse zu informieren.

Finden diese Vorgehensmodelle bereits in der Softwareentwicklung weite Verbreitung ist die Übertragbarkeit auf andere Domänen oder Entwicklungsszenarien (z.B. Systems-of-Systems) häufig nicht ohne Weiteres möglich.

### **Übergreifende und ganzheitliche Modelle der Produktentstehung**

Um der Komplexität realer Produktentstehungsprozesse gerecht zu werden, bilden übergreifende und ganzheitliche Modelle mehrere Auflösungsgrade des Produktentstehungsprozesses ab und berücksichtigen dabei verschiedene Aktivitäten, Phasen und Stakeholder der Produktentstehung.<sup>110</sup>

Das integrierte Produktentstehungsmodell (iPeM) nach ALBERS<sup>111</sup> ist ein ganzheitliches Rahmenwerk zur Erfassung und Unterstützung der Produktentstehung, das verschiedene Ansätze und Methoden der Produktentstehung integriert. Es basiert auf dem ZHO-Modell der Systemtechnik und der Problemlösungsmethodik SPALTEN. Das iPeM beschreibt die Produktentstehung auf Basis eines generischen Metamodells, welches ein anfangs vages Zielsystem in ein Objektsystem mit Hilfe von Aktivitäten überführt. Aktivitäten lassen sich in Aktivitäten der Produktentstehung und Aktivitäten der Problemlösung untergliedern. ALBERS und BRAUN<sup>112</sup> konnten zeigen, dass diese generischen Elemente ausreichen, um die Produktentstehung ganzheitlich zu umfassen.

Obwohl die meisten Produkte in Generationen entwickelt werden, konzentrieren sich die aktuellen Produktentwicklungsmodelle auf die Entwicklung von nur einem Produkt. ALBERS ET AL.<sup>113</sup> haben das iPeM daher im Kontext der Produktgenerationsentwicklung weiterentwickelt (vgl. Abbildung 14).

Das iPeM im Kontext der PGE umfasst zwei Hauptcluster:

**Produktentwicklungsaktivitäten:** Hierunter sind die Kernaktivitäten der Produktentwicklung zu finden. Sie können auf jeden Entwicklungsprozess angewendet werden.

---

<sup>110</sup> vgl. Ebel 2015

<sup>111</sup> vgl. Albers & Braun 2011

<sup>112</sup> vgl. Albers & Braun 2011

<sup>113</sup> vgl. Albers et al. 2016

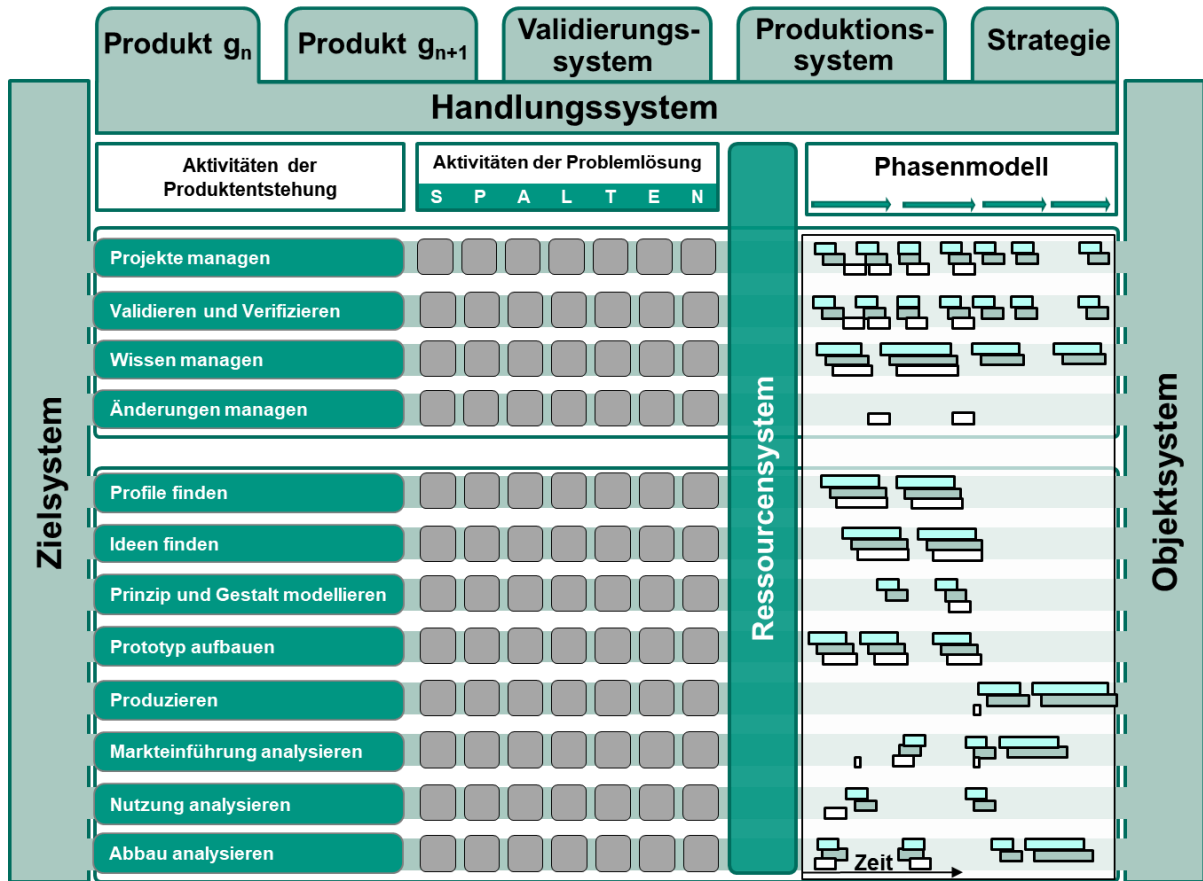


Abbildung 14: Das iPem im Kontext der PGE<sup>114</sup>

**Grundtätigkeiten:** Sie werden parallel und regelmäßig wiederkehrend zu allen anderen Aktivitäten durchgeführt, um diese zu unterstützen und zu verbessern. Dieser Cluster besteht aus den neuen Aktivitäten Wissensmanagement, Management von Änderungen sowie die bestehenden Aktivitäten Projektplanung und Controlling und Validierung und Verifikation.

Der ganzheitliche Ansatz des iPem wird weiter durch einen multidimensionalen Ansatz implementiert, die auf die unterschiedlichen Ansätze der Produktentwicklung und der Produktgenerationsentwicklung anwendbar ist. Jede dieser Methoden oder jede einzelne Produktgeneration bildet eine einzelne Schicht.

**Produkt:** Die erste Schicht beschreibt die Entwicklung des Produktes selbst. Da es sich in Generationen entwickelt, ist es möglich, eine Schicht für jede dieser einzelnen Generationen hinzuzufügen. Auf diese Weise lassen sich die Wechselbeziehungen der verschiedenen Generationen (z.B. ein Motor wird für eine Fahrzeuggeneration entwickelt und soll auf die nächste Generation übertragen werden) zuordnen. Darüber hinaus können die Ressourcen über mehrere Projekte geplant werden.

<sup>114</sup> vgl. Albers et al. 2016

**Validierungssystem:** In dieser Schicht werden Elemente entwickelt, welche die Validierung weiterer Produkte ermöglichen. Zum Beispiel muss ein Prüfstand auch geplant, konstruiert und validiert werden.

**Produktionssystem:** Hierunter sind alle Aktivitäten beinhaltet, welche eine effektive und effiziente Produktion ermöglichen sollen. Die Entwicklung eines Produktionssystems hat seinen eigenen Entwicklungsprozess.

**Strategie:** Diese Schicht beinhaltet die Entwicklung einer langfristigen Strategie, die das Unternehmen dabei unterstützen soll, eine (nachhaltig) vorteilhafte Marktposition einzunehmen.

Jede Dimension (Produkt, Strategie, Produktionssystem, Validierungssystem) des iPeM enthält ein individuelles Objektsystem. Diese individuellen Objektsysteme interagieren miteinander. Im Gegensatz dazu wird das Zielsystem und das Handlungssystem durchgehend abgebildet.

Zu weiteren bekannten Vertretern gehört das Münchner Produktkonkretisierungsmodell (MKM)<sup>115</sup> und das 3-Zyklen-Modell<sup>116</sup> der Produktentstehung. In diese Modelle sind die Erkenntnisse der Produktgenerationsentwicklung nicht eingeflossen, weshalb im Rahmen dieser Arbeit das iPeM verwendet wird.

### 2.1.5 Management von Änderungen

LINDEMANN UND REICHWALD<sup>117</sup> definieren eine Änderung als die vereinbarte Festlegung eines neuen Zustands anstelle des bisherigen Zustands und die dazugehörige Transformation. Diese Definition verbirgt jedoch die tatsächliche Komplexität und auch Gefahren, welche mit einer Änderung einhergehen. In einem laufenden Projekt haben Änderungen nicht nur Auswirkungen auf das Produkt selbst, sondern meist auch auf Projektkosten und Zeitmanagement.<sup>118</sup>

---

<sup>115</sup> vgl. Gausemeier, Plass & Wenzelmann 2009

<sup>116</sup> vgl. Ponn & Lindemann 2011

<sup>117</sup> vgl. Lindemann 1998

<sup>118</sup> vgl. DIN 6789-3

Änderung können folgende Ursachen haben:<sup>119</sup>

- Erhaltung der Aktualität eines Produktes und dessen Dokumentation
- funktionelle Verbesserungen
- Behebung von Fehlern und Ausschussursachen
- Fertigungsrationalisierung
- Änderung gesetzlicher Bestimmungen
- Sicherheitsanforderungen
- Kundenwünsche bzw. Marktbedürfnisse

Nach DIN 199-4 umfasst das Änderungswesen „die innerbetriebliche Organisation und die dazugehörigen Organisationsmittel zur Änderung von Gegenständen, zum Beispiel von Unterlagen oder Teilen“<sup>120</sup>. Es schließt weiterhin in der Regel die Folgeänderungen von spezifischen Fertigungs-, Mess- und Prüfmitteln ein. Die ITIL (Information Technology Infrastructure Library) ist eine Sammlung von „best practices“ für Entwicklung und Betrieb von IT-Systemen. Dort wird Änderungsmanagement (engl. Change Management) definiert als die Kontrolle „über alle Änderungen“, wobei sich diese Definition lediglich auf Veränderungen am Produkt bezieht. ALBERS ET AL. erweitern diese Definitionen wie folgt:

**Definition 2-5: Management von Änderungen**<sup>121</sup>

Das Management von Änderungen beinhaltet die Koordination von technischen, wirtschaftlichen und sozialen Veränderungen. Inhärente Elemente sind die Früherkennung von Fehlern und die Umsetzung der entsprechenden Maßnahmen.

Ein Management von Änderungen ist also nicht nur bei Änderungen am eigentlichen Produkt sondern beispielsweise auch bei Änderungen an der Organisation, welche das Produkt entwickelt, notwendig.

Änderungen werden immer durch die Erkennung einer Soll- und Ist-Abweichung eingeleitet. Zur Bearbeitung von Änderungen lassen sich folglich Methoden der Problemlösung verwenden.<sup>122</sup>

---

<sup>119</sup> vgl. PMI 2004

<sup>120</sup> vgl. DIN 199-4

<sup>121</sup> vgl. Albers et al. 2016

<sup>122</sup> vgl. Albers et al. 2016

Abbildung 15 illustriert den gesamten Änderungsprozess nach DIN 199-4. Der Prozess ist unterteilt in die Module Änderungsvorlauf und Änderungsdurchführung. Parallel dazu ist die Einrichtung eines Projektmanagements zur Planung, Schätzung, Monitoring und Kontrolle der einzelnen Prozesse notwendig.

Die erste Aktivität im Prozess ist das Erstellen eines Änderungsantrages. Das Ergebnisartefakt dieser Phase ist ein eindeutig identifizierbarer Änderungsantrag in schriftlicher oder elektronischer Form. Im nächsten Schritt wird der Antrag geprüft. Dabei findet eine Priorisierung und Kategorisierung der Änderung statt. Ergebnis dieser Phase ist ein genehmigter oder abgelehnter Änderungsantrag (engl. „Request for Change (RfC)“) mit Begründung der Ablehnung oder Genehmigung. Bei Genehmigung wird der Änderungsantrag zu einem Änderungsauftrag. Es beginnt das zweite Modul der DIN 199-4 - die Änderungsdurchführung. Während der Änderungsdurchführung entstehen verschiedenartige Entwicklungsartefakte und eine Beschreibung der durchgeführten Änderungen. Zuletzt wird die Änderung ausgeliefert und (vor Kunde) eingesetzt.

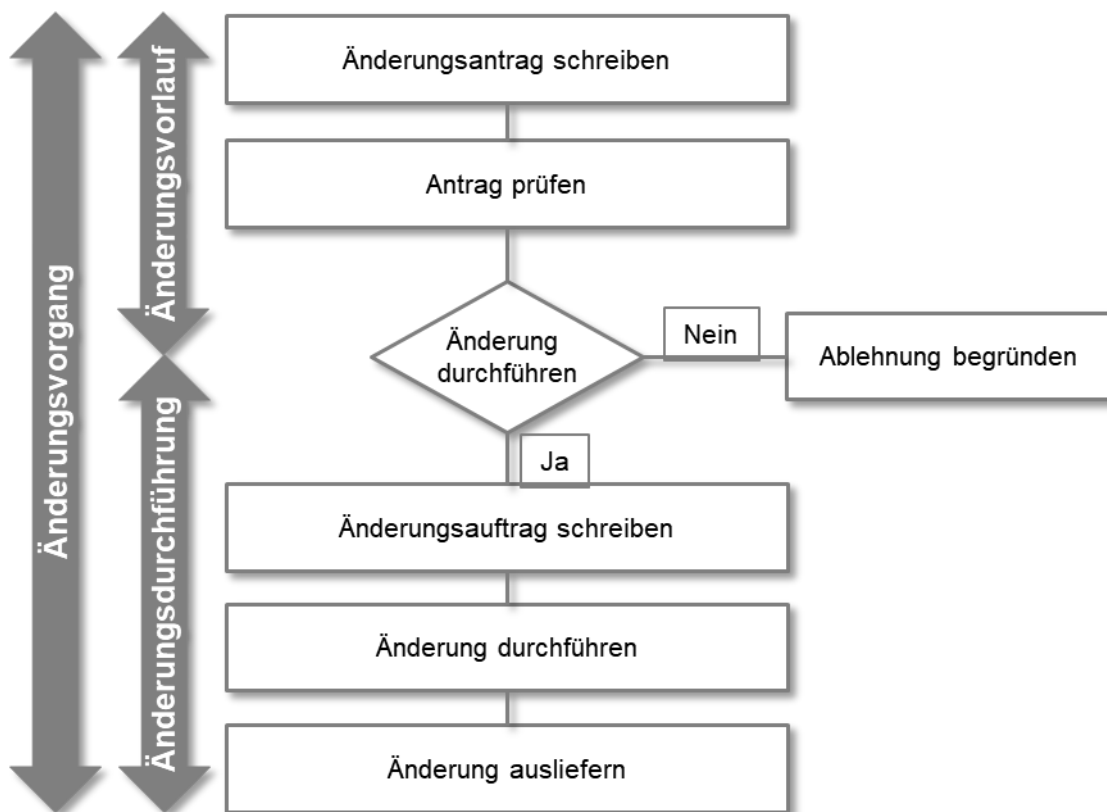


Abbildung 15: Änderungsprozess nach DIN 199-4<sup>123</sup>

<sup>123</sup> vgl. DIN 199-4

Nach LINDEMANN UND REICHWALD<sup>124</sup> entspricht dieser arbeitsteilige und bürokratisch organisierte Prozess nicht mehr dem Umfeld, in dem Unternehmen heute agieren. Sie fordern ein integriertes Änderungsmanagement, das neben technischen Aspekten auch teamorientierte Prozesse und organisatorische Maßnahmen mit einschließt, um die Effizienz und Effektivität von Änderungen im Unternehmen zu verbessern.

Sie definieren 3 Bausteine und 8 Aktionsfelder für ein integriertes Änderungsmanagement, welche nicht getrennt voneinander betrachtet werden dürfen (vgl. Abbildung 16).

- **Aktionsfeld 1: Vermeidung und Vorverlagerung von Änderungen**  
Durch die Vermeidung oder Vorverlagerung von Änderungen sollen Auswirkungen von Änderungen auf Kosten, Qualität oder Zeit eingeschränkt werden.
- **Aktionsfeld 2: Änderungserkennung**  
Änderungen müssen durch durchgängige Soll- und Ist-Analysen und Vergleiche erkannt werden. Dabei müssen die Aspekte Problemmeldung, Problemdokumentation und Priorisierung hinsichtlich Wichtigkeit und Dringlichkeit berücksichtigt werden.
- **Aktionsfeld 3: Problem- und Ursachenanalyse**  
Die Problem- und Ursachenanalyse ist insbesondere bei der Bearbeitung fehlerbedingter Änderungen notwendig. Ziel ist, ein vorhandenes Problem soweit zu strukturieren und seine Komplexität zu reduzieren, bis es als definierte Aufgabe bewältigt werden kann. Dazu gehören die Problembeschreibung, die Abgrenzung des Problembereichs sowie die Analyse der Problemursache.
- **Aktionsfeld 4: Strategien zum Entwickeln von Lösungsalternativen**  
Ein Änderungsgremium, zusammengesetzt aus den von der Änderung betroffenen Abteilungen oder Mitarbeitern, hat die Aufgabe, geeignete Lösungsalternativen aufzubauen, auf vorhandenen Randbedingungen zu entwickeln und zur weiteren Realisierung auszuwählen.
- **Aktionsfeld 5: Auswirkungserfassung und Änderungsplanung**  
Mit diesem Aktionsfeld sollen Auswirkungen von Änderungen auf Produkt, Aufbauorganisation, Prozess und Kosten erfasst und zur Entscheidung vorbereitet werden.

---

<sup>124</sup> vgl. Lindemann & Reichwald 1998

- **Aktionsfeld 6: Gesamtheitliche, wirtschaftliche Bewertung und Entscheidung**

Auf Basis der entwickelten und ausgewählten Lösung, sowie der identifizierten Auswirkungen soll mit diesem Aktionsfeld eine gesamtheitliche und wirtschaftliche Bewertung die Basis für eine Entscheidung legen.

- **Aktionsfeld 7: Effiziente Abwicklung von Änderungen**

Für eine effiziente Umsetzung der Änderung ist eine bereichsübergreifende Zusammenarbeit, die sorgfältige Planung der Änderungsabwicklung und die Koppelung mit der Projektplanung der differenzierten Umsetzungsstrategien notwendig.

- **Aktionsfeld 8: Lernorientierte Auswertung von Änderungsdaten**

Änderungsdaten bilden das Wissen und die Erfahrung der Unternehmen und ihrer Mitarbeiter ab. Dies sollte dazu genutzt werden, Fehler zu vermeiden oder die Bewertung und Planung von Änderungen zu optimieren.

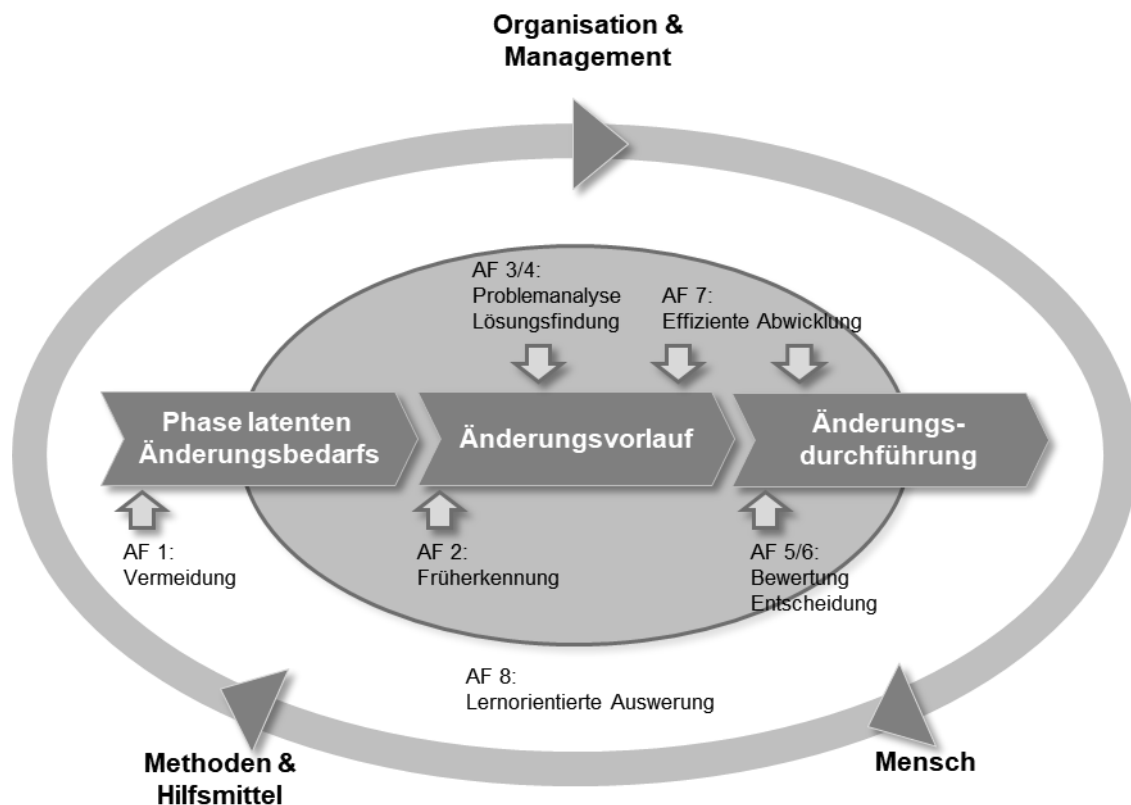


Abbildung 16: Bausteine und Aktionsfelder des Integrierten Änderungsmanagement<sup>125</sup>

Dabei ist jedes Aktionsfeld aus drei Bausteinen aufgebaut, die zur Implementierung und Ausführung des Managements von Änderungen herangezogen werden müssen.

<sup>125</sup> vgl. Lindemann & Reichwald 1998, S. 261

**Organisation & Management:** Das flexible Management von Änderungen muss durch eine geeignete Aufbau- und Ablauforganisation unterstützt werden.

**Methoden & Hilfsmittel:** Aufgaben des Änderungsmanagements sind zu komplex, um unstrukturiert bearbeitet werden zu können. Das Änderungsmanagement muss folglich durch geeignete Hilfsmittel und Methoden unterstützt werden.

**Mitarbeiter als zentrale Leistungsträger:** Der Mitarbeiter ist derjenige, der sich an den Bausteinen bedient und die Aktionsfelder tätigt. Der Mitarbeiter benötigt dafür sowohl die Motivation als auch die notwendige Qualifikation zur Durchführung der Aufgaben. Zur Durchführung von Änderungen stellt Lindemann (S. 93) neben den fachlichen Qualifikationen 4 Anforderungen an den Mitarbeiter, die es durch geeignete Maßnahmen zu motivieren gilt:

- Innovationsfähigkeit
- Basisqualifikationen
- Sozialkompetenz
- Komplexitätsbeherrschung

### 2.1.6 Zwischenfazit

Heutige Produktentstehungsprozesse sind als sozio-technische Systeme zu verstehen, die starke Wechselwirkungen zwischen Produkt und Mensch aufweisen. Um Produkte hervorzubringen, die am Markt erfolgreich sind, muss die Produktentstehung mit einer hohen Produktkomplexität, häufigen Änderungen und einem zunehmenden Grad an Interdisziplinarität umgehen können, wofür entsprechende Ansätze benötigt werden. Jeder Produktentstehungsprozess ist dabei einzigartig und individuell<sup>126</sup> und somit im Vorfeld auch nicht vollständig planbar. Nach ALBERS<sup>127</sup> können Produkte häufig sogar erst nach diversen Variationen über mehrere Produktgenerationen hinweg wirtschaftlich und technisch dahingehend entwickelt werden, dass sie erfolgreich auf dem Markt etabliert werden können.

Der Systems Engineering Ansatz versucht durch Einnahme entsprechender Abstraktionsebenen auf Basis der allgemeinen Systemtheorie, Erkenntnisse und Methoden auf andere Anwendungsgebiete übertragbar zu machen. Die Produktentstehung lässt sich dabei abstrakt als ein Tripel aus Ziel-, Handlungs- und Objektsystem darstellen.

Das iPeM greift das ZHO-Modell und die Produktgenerationsentwicklung auf und verbindet verschiedene Verständnisse von Produktentstehungsprozessen, indem es

---

<sup>126</sup> vgl. Lindemann 1998

<sup>127</sup> vgl. Albers 2010



die Sichten von Projektmanagement (Planung, Steuerung und Kontrolle) und Entwicklungsunterstützung (Problemlösung und Wissensgenerierung) in einem Modell zusammenführt. Damit stellt es aufgrund seiner integrativen Betrachtung von Produktentstehungsprozessen sowie der starken Berücksichtigung von iterativer Wissensgewinnung und adaptiver Prozessplanung und -steuerung ein geeignetes Rahmenwerk für die Erforschung menschenzentrierter Entwicklungsaspekte bereit.

Das Management von Änderungen ist ein essentieller Baustein in der Entwicklung, um Durchgängigkeit und Konsistenz über den gesamten Entwicklungsprozess herzustellen. Das iPeM berücksichtigt das Management von Änderungen grundsätzlich, jedoch müssen noch konkreten Methoden und Maßnahmen entwickelt werden, um Entwickler beim Umgang mit Änderungen, insbesondere in einem verteilten Entwicklungsumfeld zu unterstützen.

## 2.2 Zielsysteme in der Produktentstehung

Ein Zielsystem beinhaltet nach ALBERS<sup>128</sup> alle expliziten Ziele und Anforderungen eines zu entwickelnden Produktes, einschließlich derer Abhängigkeiten und Randbedingungen, innerhalb eines definierten Interessenbereichs (d.h. innerhalb eines System-of-Interest) zu einem bestimmten Zeitpunkt. BADER<sup>129</sup> betont, dass in der Entwicklung die Produktziele zwar eine zentrale Rolle einnehmen, diese jedoch mit einer Vielzahl weiterer Ziele in Zusammenhang stehen (vgl. Abbildung 17).

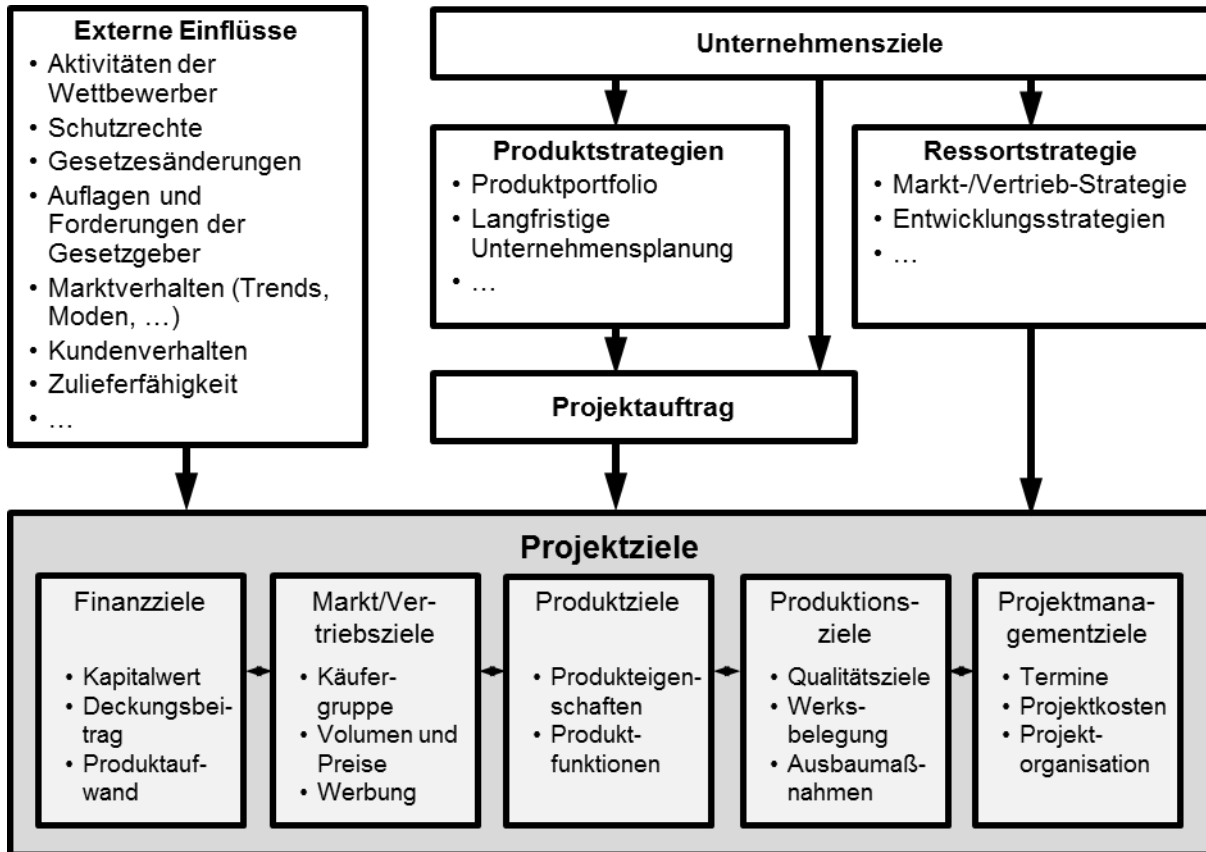


Abbildung 17: Zielsystem eines Entwicklungsprojekts nach BADER<sup>130</sup>

### 2.2.1 Ziele, Anforderungen und Randbedingungen

HABERFELLNER ET AL.<sup>131</sup> definiert Ziele als Aussagen darüber, was mit einer zu gestaltenden Lösung und auf dem Weg zu dieser Lösung erreicht bzw. vermieden werden soll. Ziele beschreiben nach CROSS<sup>132</sup> einen Bedarf und definieren damit,

<sup>128</sup> vgl. Albers 1991

<sup>129</sup> vgl. Albers 2010

<sup>130</sup> vgl. Bader 2007, S. 20

<sup>131</sup> vgl. Haberfellner et al. 2012

<sup>132</sup> vgl. Cross 2008

was durch das zu entwickelnde Produkt erreicht werden soll. Anforderungen beschreiben, was das Produkt tun soll (nicht was es sein soll). EILETZ<sup>133</sup> versteht unter Zielen einen gemeinsam vereinbarten, zukünftigen Sollzustand, der durch aktives Handeln erstrebt oder vermieden wird. Anforderungen hingegen formulieren gewünschte Eigenschaften der Lösung bzw. gewünschte Sachverhalte. Nach EHRENSPIEL<sup>134</sup> definiert Ziele eine Soll-Vorstellungen des Auftraggebers. Aus Zielen werden Anforderungen abgeleitet, die die Solleigenschaften des zukünftigen Produktes beschreiben. POHL<sup>135</sup> versteht unter einem Ziel, die intentionale Beschreibung eines charakteristischen Merkmals des zu entwickelnden Systems bzw. des zugehörigen Entwicklungsprozesses. ALBERS ET AL.<sup>136</sup> sehen 4 Zwecke von Zielen:

- Definition eines Soll-Zustands
- Mentale Führung der Entwicklung
- Handlungsregulierung
- Bewertungskriterium für die Validierung

**Definition 2-7: Ziel<sup>137</sup>**

Ein Ziel ist die bewusste gedankliche Vorwegnahme eines zukünftigen Soll-Zustands, der gemeinsam vereinbart und durch aktives Handeln erreicht oder vermieden wird.

BADER<sup>138</sup> ist der Ansicht, dass sich Anforderungen aus Zielen und weiteren Randbedingungen ableiten sollen. BADER verwendet in diesem Kontext den Begriff der Prämisse. Er definiert ihn als eine bei der Entwicklung zwingend zu beachtende Bedingung, die sich aus einem Ziel ergibt, das von Anderen verantwortet wird und nicht selbst umgesetzt werden kann. Eine Anforderung nach POHL<sup>139</sup> ist eine Bedingung oder Eigenschaft, die ein System oder eine Person benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen. Dabei kann es sich auch um eine Bedingung oder Eigenschaft handeln die ein System oder eine Systemkomponente aufweisen muss, um einen Vertrag zu erfüllen oder einem Standard, einer

---

<sup>133</sup> vgl. Eiletz 1999

<sup>134</sup> vgl. Ehrlenspiel 2009

<sup>135</sup> vgl. Pohl 2008

<sup>136</sup> vgl. Albers, Ebel & Lohmeyer 2012, S. 11

<sup>137</sup> vgl. Eiletz 1999

<sup>138</sup> vgl. Bader 2007

<sup>139</sup> vgl. Pohl 2008

Spezifikation oder einem anderen formell auferlegten Dokument zu genügen. Häufig wird eine klare Trennung zwischen Zielen und Anforderungen gefordert, oftmals jedoch nicht vorgenommen<sup>140</sup>. Nach OERDING<sup>141</sup> lässt sich dies im Kern darauf zurückführen, dass eine klare Trennung zwischen Zielen und Anforderungen nur bedingt möglich ist.

**Definition 2-8: Anforderung**<sup>142</sup>

Eine Anforderung ist

- (1) eine Eigenschaft oder Bedingung, die ein System oder eine Person benötigt, um ein Problem zu lösen oder ein Ziel zu erreichen.
- (2) 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.
- (3) eine dokumentierte Repräsentation einer Eigenschaft oder Bedingung wie in den ersten beiden Punkten beschrieben.

LOHMEYER<sup>143</sup> versteht unter einer Randbedingung eine aus dem Umfeld der Entwicklung resultierende Einschränkung, die von Anderen verantwortet wird und daher zwar ermittelt, aber nicht eigenständig definiert oder geändert werden kann. POHL<sup>144</sup> ergänzt, dass Randbedingungen nur schwer oder gar nicht veränderbar sind. Sie schränken die Art und Weise ein, wie ein System entwickelt wird. MUSCHIK<sup>145</sup> unterscheidet zwischen exogenen und endogenen Randbedingungen. Exogene Randbedingungen sind durch das Entwicklungsumfeld (z.B. Wirtschaft, Politik, Gesellschaft und Umwelt) und das Marktumfeld (z.B. Kunde und Wettbewerb) bestimmt. Endogene Randbedingungen dahingegen entstammen dem Umfeld des eigenen Unternehmens (z.B. Unternehmensstrategie und Ressourcenverfügbarkeit).

---

<sup>140</sup> vgl. Stechert 2010

<sup>141</sup> vgl. Oerding 2009

<sup>142</sup> vgl. IEEE 1990

<sup>143</sup> vgl. Lohmeyer 2013

<sup>144</sup> vgl. Pohl 2008

<sup>145</sup> vgl. Muschik 2011, S. 61

**Definition 2-9: Randbedingung**<sup>146</sup>

Eine Randbedingung ist eine aus dem Umfeld der Entwicklung resultierende Einschränkung, die von Anderen verantwortet wird und daher zwar ermittelt, aber nicht eigenständig definiert oder geändert werden kann.

EBEL<sup>147</sup> bringt Ziele, Anforderungen und Randbedingungen in Zusammenhang (vgl. Abbildung 18). Er beschreibt ein Ziel als die Definition dessen, was erreicht werden soll. Anforderungen definieren die Eigenschaften des Produkts, die realisiert werden sollen und leiten sich aus den Zielen ab. Randbedingungen haben einen direkten Einfluss auf die Anforderungen und definieren Einschränkungen, welche bei der Entwicklung berücksichtigt werden müssen.<sup>148</sup>

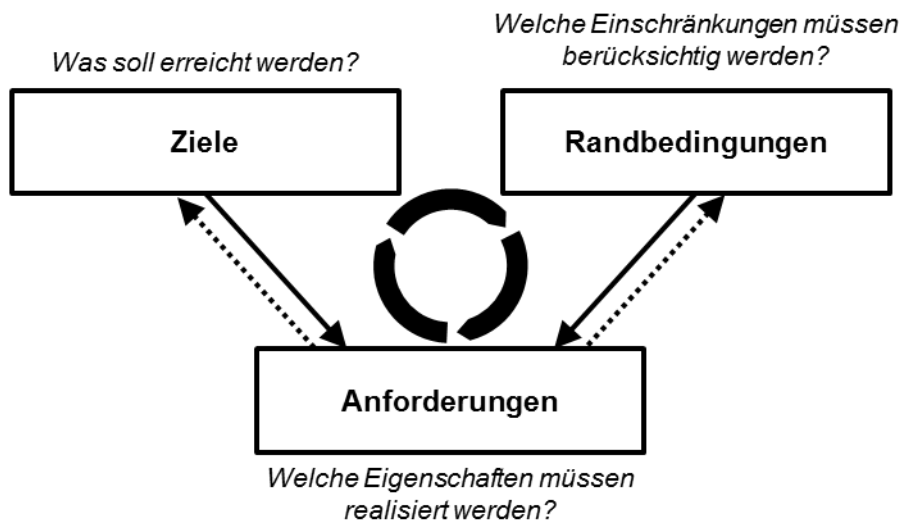


Abbildung 18: Zusammenhang zwischen Zielen, Anforderungen und Randbedingungen<sup>149</sup>

## 2.2.2 Entwicklung und Management von Zielsystemen

Das grundlegende Zielsystem eines Entwicklungsprozesses besteht anfangs aus vagen, allgemeinen Zielen und wird auch als *initiales Zielsystem*<sup>150</sup> bezeichnet.<sup>151</sup> Im Laufe der Entwicklung werden die Verknüpfungen zum Objektsystem stetig

<sup>146</sup> vgl. Lohmeyer 2013

<sup>147</sup> vgl. Ebel 2015

<sup>148</sup> vgl. Ebel 2015, S. 66

<sup>149</sup> vgl. Ebel 2015

<sup>150</sup> vgl. Albers & Muschik 2010

<sup>151</sup> vgl. Albers, Lohmeyer & Ebel 2011

präzisiert, woraus sich konkrete operative Ziele und Anforderungen ergeben.<sup>152</sup> HABERFELLNER ET AL.<sup>153</sup> betonen, dass dem Prozess der Zielbildung eine große Bedeutung in der Produktentstehung zukommt, da sie nur dann die Lösungssuche steuern können, wenn sie formuliert allen relevanten Akteuren bekannt und von Allen akzeptiert sind. Meboldt nennt drei Möglichkeiten, Veränderungen an einem Zielsystem vorzunehmen. Durch das Erweitern des Zielsystems werden neue valide Elemente, die nicht aus anderen ableitbar sind, dem Zielsystem hinzugefügt. Beim Konkretisieren wird ein neues Element von bereits bestehenden Elementen abgeleitet. Schließlich wird beim Verändern des Zielsystems Inkonsistenzen derart begegnet, dass Elemente verändert oder gar entfernt werden.<sup>154</sup>

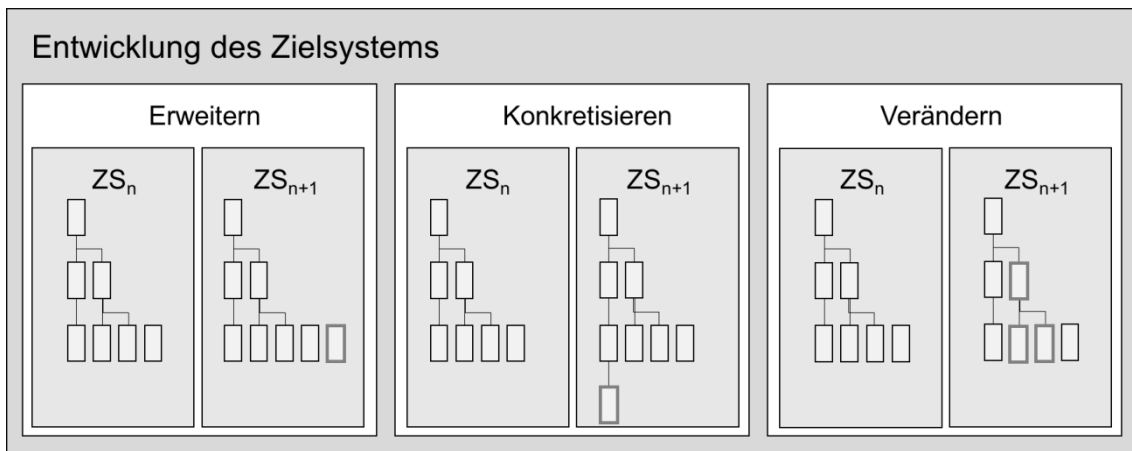


Abbildung 19: Entwicklung des Zielsystems; eigene Darstellung nach Meboldt<sup>155</sup>

ALBERS UND MEBOLDT<sup>156</sup> nennen die kontinuierliche Erweiterung und Konkretisierung des Zielsystems während des Produktentwicklungsprozesses Operationalisierung.

EHRLENSPIEL<sup>157</sup> betont das einer intensiven Kommunikation mit dem Endkunden bzw. Auftraggeber eine hohe Bedeutung bei der Entwicklung von Zielsystemen zukommt. LINDEMANN<sup>158</sup> nennt eine Vielzahl weiterer Quellen für Ziele und Anforderungen (vgl. Abbildung 20).

MUSCHIK<sup>159</sup> stellte im Rahmen von empirischen Untersuchungen bezüglich der Zielsystementwicklung bei einem deutschen Automobilhersteller fest, dass bei der Fahrzeugentwicklung eine Reihe von unterschiedlichen Fachabteilungen mit

<sup>152</sup> vgl. Meboldt 2008

<sup>153</sup> vgl. Haberfellner et al. 2012

<sup>154</sup> vgl. Meboldt 2008, S.189

<sup>155</sup> vgl. Meboldt 2008

<sup>156</sup> vgl. Albers & Meboldt 2007

<sup>157</sup> vgl. Ehrlenspiel 2009

<sup>158</sup> vgl. Lindemann 2009

<sup>159</sup> vgl. Muschik 2011

unterschiedlichen Zielgrößen involviert ist. Dabei werden Entscheidungen auf verschiedenen Hierarchieebenen gefällt, beginnen mit dem individuellen Mitarbeiter über die Fachabteilungen bis hin zu Gremiensitzungen mit Beteiligung der Vorstandsebene. Eine Schwierigkeit besteht darin, die Transparenz von Zielen zu gewährleisten und der Vernetzung von Zielen und Zielsystemen Rechnung zu tragen.<sup>160</sup> Aufbauend auf den Erkenntnissen von MUSCHIK abstrahieren ALBERS ET AL.<sup>161</sup> menschenzentrierte Aspekte zur Unterstützung der Zielsystementwicklung:

- *Transparente Darstellung der Zielhierarchie und Vernetzungen*
- *Berücksichtigung und Nutzbarmachung von Vorwissen*
- *Standardisierung und Einsatz von Vorwissen in adäquatem Verhältnis bezüglich vorliegender Unsicherheit*
- *Synchronisierung von Zwischenergebnissen*
- *Flexible, teilstandardisierte Methoden und Tools (Anpassung an Belastbarkeit/Messbarkeit der Informationen)*

Nach MUSCHIK spielen Entscheidungen eine zentrale Rolle bei der Entwicklung von Zielsystemen. Da Entscheidungen als Teil eines übergreifenden Problemlösungsprozesses angesehen werden können, schlägt Muschik die Problemlösungsmethodik SPALTEN für die Zielsystementwicklung vor.

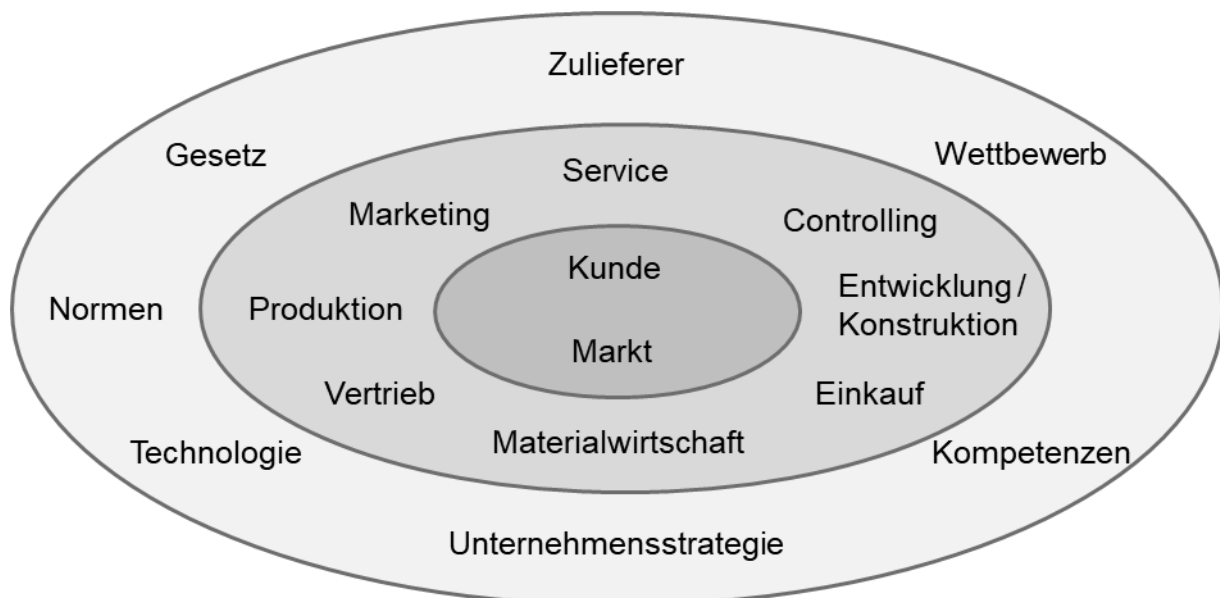


Abbildung 20: Quellen von Zielen und Anforderungen<sup>162</sup>

<sup>160</sup> vgl. Muschik 2011

<sup>161</sup> vgl. Albers, Muschik & Ebel 2010, S. 97

<sup>162</sup> vgl. Lindemann 2009

Obwohl Anforderungen aus Zielen und Randbedingungen abgeleitet werden können, kann die Entwicklung von Zielsystemen nicht sequentiell verlaufen. DARLINGTON UND CULLEY<sup>163</sup> führen dies auf die starken Wechselwirkungen zwischen Zielen, Anforderungen und Randbedingungen zurück. Nach ALBERS<sup>164</sup> wird das Zielsystem parallel zum Objektsystem entwickelt. Damit ist der Konkretisierungsgrad des Zielsystems mit dem des Objektsystems verknüpft.

Auf Grund von Unsicherheit im Entwicklungsprozess ist im ZHO-Modell nach ALBERS<sup>165</sup> die Entwicklung von Zielsystemen durch einen iterativen Prozess beschrieben. Das Handlungssystem entscheidet auf Basis einer Wissensbasis durch Synthese, welche Ziele, Anforderungen und Randbedingungen zum Teil des Zielsystems werden (vgl. auch Kapitel 2.1.2). Dieser Schritt beinhaltet das explizite Festlegen, Verfeinern und Ändern von Zielen und Randbedingungen<sup>166</sup>. Rückflüsse aus der Entwicklung in das Zielsystem werden über die Wissensbasis getätigt. LOHMEYER hat festgestellt, dass die Entwicklung von Zielsystemen häufig im Team erfolgt. Das betrachtete Handlungssystem setzt sich folglich aus mehreren Menschen zusammen, wobei jeder Mensch seine individuelle Wissensbasis mit einbringt.

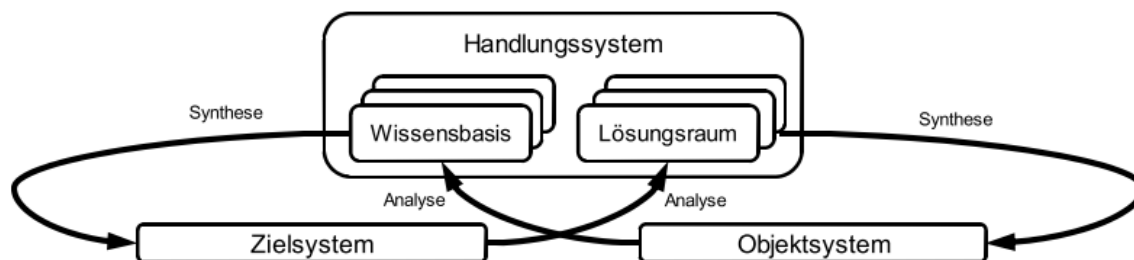


Abbildung 21: Individuelle Wissensbasis und Lösungsraum bei der Entwicklung im Team<sup>167</sup>

Entwickler interagieren in der Teamarbeit miteinander.<sup>168</sup> Dazu nutzen sie eine äußere Ich-Ebene, d.h. sie explizieren ihre vorgedachten und vorausgewählten Ziele und Anforderungen in Form von Dokumenten und Modellen (vgl. Abbildung 22).

<sup>163</sup> vgl. Darlington 2002

<sup>164</sup> vgl. Albers, Ebel & Lohmeyer 2012

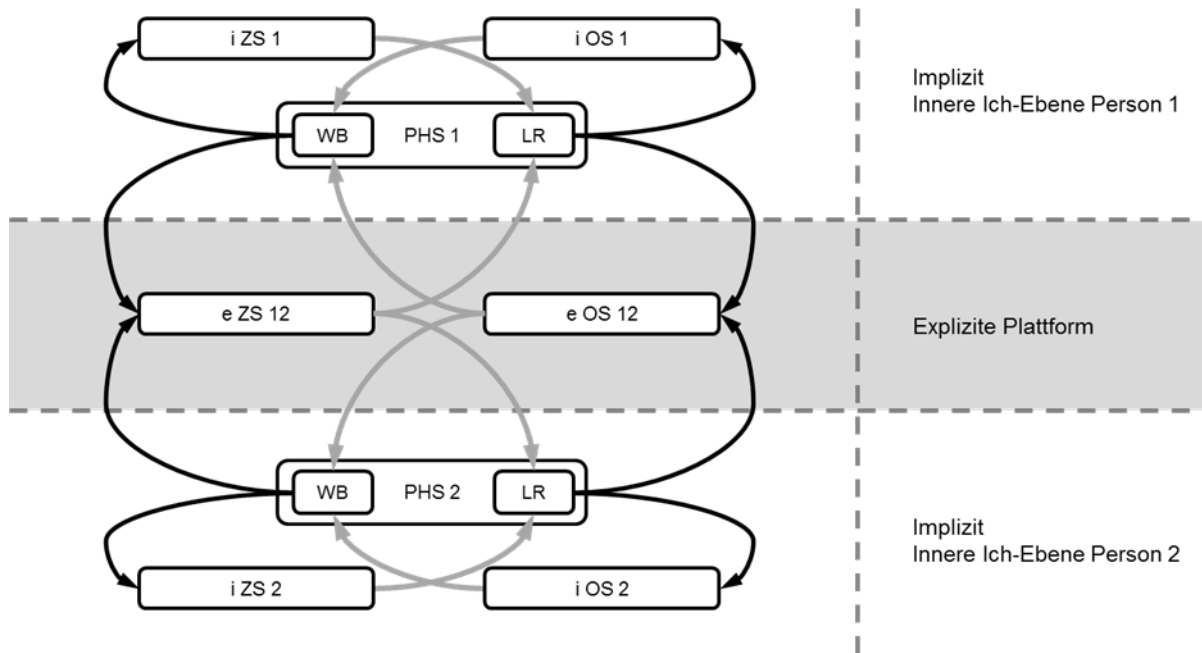
<sup>165</sup> vgl. Albers & Lohmeyer 2012

<sup>166</sup> vgl. Albers, Ebel & Lohmeyer 2012

<sup>167</sup> vgl. Albers, Ebel & Lohmeyer 2012

<sup>168</sup> vgl. Lohmeyer 2013, S. 135



Abbildung 22: Interaktionsebenen des erweiterten ZHO-Modells<sup>169</sup>

Nach LOHMEYER<sup>170</sup> ist die Synthese eines Zielsystems somit ein sozio-technischer Entscheidungsprozess, welcher nicht nur durch die Wissensbasis sondern insbesondere auch durch die unterschiedlichen Verantwortungs- und Interessenbereichen der Beteiligten bestimmt ist.

Als Teilaktivität bei der Entwicklung von Zielsystemen kann das weit verbreitete Requirements Engineering und Management Anwendung finden, welches seinen Ursprung in der Softwareentwicklung hat. Heute sind Requirements Engineering und Management auch im Systems Engineering weit verbreitet. So werden diese häufig auch in den INCOSE (International Council on Systems Engineering) Handbüchern referenziert.

Die Abgrenzung zwischen Requirements Engineering und Requirements Management variiert von Autor zu Autor. SCHIENMANN, LEFFINGWELL und HOOD<sup>171</sup> sehen das Requirements Management als einen systematischen Ansatz zur Erhebung, Organisation und Dokumentation von Anforderungen, wobei Requirements Engineering einen Teilbereich darstellt. WIEGERS, EBERT, POHL, RUPP und PARTSCH<sup>172</sup> sehen hingegen das Requirements Management als einen Teil vom Requirements Engineering an.

<sup>169</sup> vgl. Lohmeyer 2013

<sup>170</sup> vgl. Lohmeyer 2013

<sup>171</sup> vgl. Lohmeyer 2013, Schienmann 2002 und Leffingwell 2003

<sup>172</sup> vgl. Wieggers 2005, Ebert 2014, Pohl 2008, Pohl & Rupp 2011 und Partsch 2010

POHL<sup>173</sup> definiert 3 Kernaktivitäten des Requirements Engineering und ergänzt diese um die Querschnittsaktivitäten Validierung und Management (vgl. Abbildung 23). Die Kernaktivitäten liegen nach POHL zufolge in der Dokumentation, der Gewinnung und der Übereinstimmung der Anforderungen. Nach POHL stehen die Kernaktivitäten in starker Wechselwirkung zueinander. So kann beispielsweise die Gewinnung neuer Anforderungen die Dokumentation dieser beeinflussen oder durch die Übereinstimmungsaktivität identifizierte Konflikte zur Gewinnung neuer Anforderungen beitragen und als Quelle neuer Ideen dienen. Die verschiedenen Aktivitäten können demzufolge nicht sequentiell durchlaufen werden, sondern unterliegen einem iterativen Prozess.

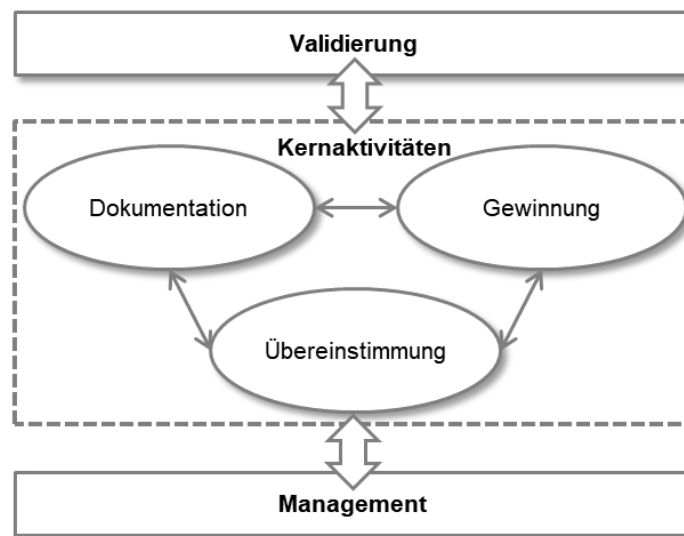


Abbildung 23: Aktivitäten des Requirements Engineering nach POHL<sup>174</sup>

Mit der Gewinnung von Anforderungen ist die Identifikation von Anforderungsquellen, die Sammlung von existierenden Anforderungen sowie die Entwicklung innovativer Anforderungen gemeint. Zur Ermittlung von Anforderungen stehen verschiedene Methoden zur Verfügung:<sup>175</sup>

- Befragungstechniken: z.B. Interviews, Workshops und Fragebögen
- Kreativtechniken: z.B. Brainstorming, Perspektiventchnik, Analogietechnik
- Dokumentenzentrierte Techniken: z.B. perspektivenbasiertes Lesen
- Beobachtungstechniken: z.B. Feldbeobachtungen

Die Dokumentation hat zum Zweck, gewonnene Informationen oder Anforderungen an das zu entwickelnde System in einer Art und Weise zu dokumentieren, sodass

<sup>173</sup> vgl. Pohl 2008

<sup>174</sup> In Anlehnung an Pohl 2008, S.39

<sup>175</sup> vgl. Pohl 2008

Zusammenhänge transparent werden und der Zugriff für alle Beteiligten erleichtert wird (vgl. auch Kapitel 2.2.3).

Die Übereinstimmung verfolgt das Ziel, Konflikte oder Inkonsistenzen zwischen einzelnen Anforderungen zu identifizieren, zu analysieren und aufzulösen.

Neben den Kernaktivitäten nennt POHL<sup>176</sup> noch zwei Querschnittsaktivitäten, die während des gesamten Prozess der Anforderungsgewinnung, Dokumentation und Übereinstimmung durchgeführt werden.

Im Rahmen der Validierung werden die identifizierten Anforderungen überprüft, um Fehler, verletzte Standards, Mehrdeutigkeit oder Inkonsistenz aufzudecken.

Das Anforderungsmanagement hat die Aufgabe, Anforderungen über den gesamten Lebenszyklus des zu entwickelnden Systems zu verwalten. Dazu gehören nach POHL alle Aktivitäten, die notwendig sind, um Anforderungen zu strukturieren, für unterschiedliche Rollen aufzubereiten sowie konsistent zu ändern und umzusetzen. Das Management von Änderungen wird folglich als Teil des Anforderungsmanagement gesehen. Nach HOOD ET AL.<sup>177</sup> adressiert das Requirements Management auch das Monitoring, also das Überwachen der entwickelten Anforderungen.

HOOD ET AL. unterteilen den Requirements Engineering Prozess weiter in zwei Teilprozesse (vgl. Abbildung 24). In einem ersten Schritt wird der Umfang festgelegt. Dazu werden zunächst Schnittstellen identifiziert und definiert, sowie relevante Stakeholder und Rollen erkannt. In einem zweiten Schritt werden die Anforderungen erhoben, spezifiziert, analysiert und validiert.

---

<sup>176</sup> vgl. Pohl & Rupp 2011

<sup>177</sup> vgl. Hood et al. 2008

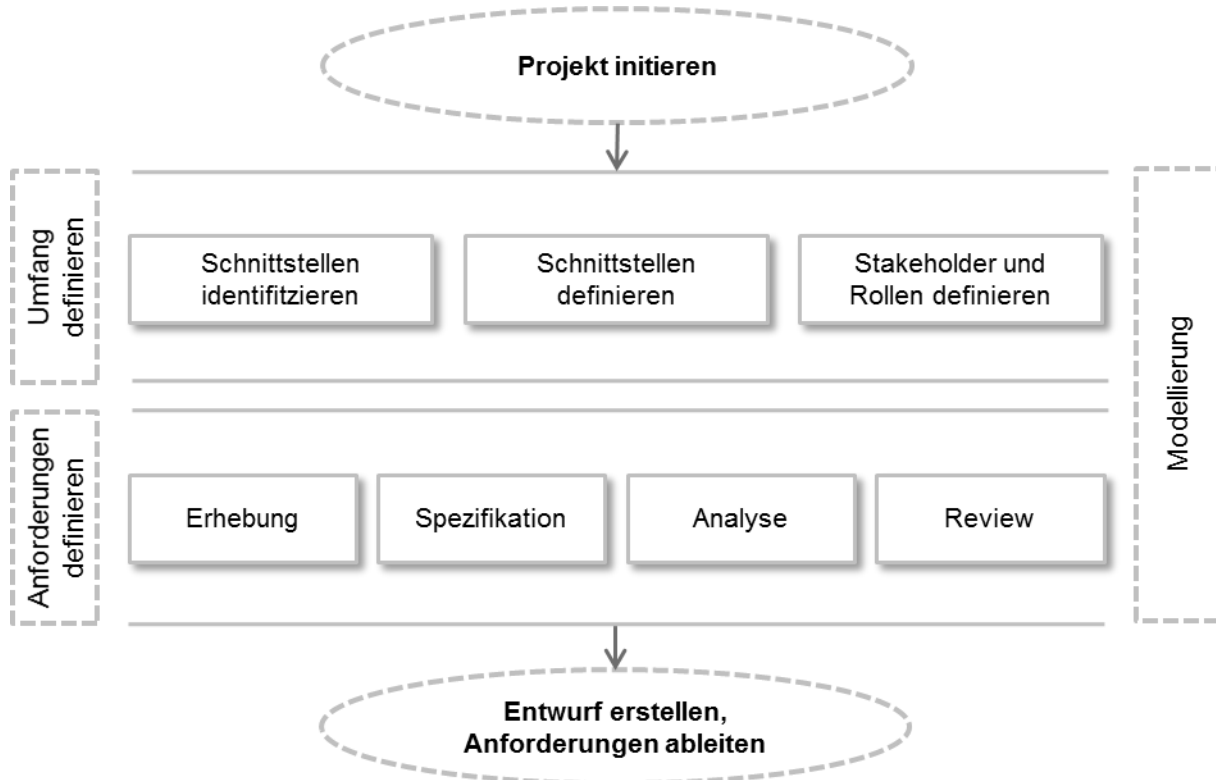


Abbildung 24: Requirements Engineering Prozess nach Hood et al.<sup>178</sup>

### 2.2.3 Dokumentation von Zielsystemen

BIRKHOFFER UND JÄNSCH<sup>179</sup> gehen davon aus, dass mentale Modelle Informationen über Objekte, Prozesse, Strategien oder Motivationen enthalten, die situativ zu strukturierten Mustern, Schemata, Konzepten oder Bildern zusammengesetzt werden. In einem Interaktionsprozess werden dazu zunächst individuelle mentale Modelle auf Basis subjektiv wahrgenommener Information gebildet, die dann intersubjektiv miteinander verglichen und schließlich situationsspezifisch integriert, geändert, erweitert oder angepasst. Um mentale Modelle vergleichen zu können ist eine externe Repräsentation notwendig.

Durch Dokumentation versucht man die gemeinsame Basis für die Produktentwicklung festzuhalten und allen Beteiligten zugänglich zu machen.<sup>180</sup> Übertragen auf die Zielsystembildung ergibt sich nach MEBOLDT<sup>181</sup> das Potential, durch ein durchgängig dokumentiertes und transparent gestaltetes Zielsystem Inkonsistenzen im Zielsystem, also Zielkonflikte, frühzeitig zu identifizieren.

<sup>178</sup> vgl. Hood et al. 2008, S.50

<sup>179</sup> vgl. Birkhofer and Jänsch 2003

<sup>180</sup> vgl. Pohl & Rupp 2011

<sup>181</sup> vgl. Meboldt 2008

Bei der Dokumentation von Zielsystemen kann zwischen drei unterschiedlichen Formalitätsgraden differenziert werden, die sich in Syntax (Definition der erlaubten Zeichen und deren möglichen Kombination) und Semantik (Bedeutung der Zeichen und deren Kombination) unterscheiden:

- Informale Dokumentation: Semantik ist nicht eindeutig festgelegt.
- Semi-formale Dokumentation: Definierte, eindeutige Syntax.
- Formale Dokumentation: Streng definierte Semantik.

### **Informale Dokumentation**

Die am weitesten verbreitete Dokumentationsform basiert auf der informalen Sprache und ist die natürlich-sprachige Dokumentation<sup>182</sup>. Diese erfolgt häufig in sogenannten Lasten- und Pflichtenheften (vgl. Abbildung 25).<sup>183</sup> Das Lastenheft<sup>184</sup> stellt eine Zusammenstellung aller Anforderungen des Auftraggebers hinsichtlich Liefer- und Leistungsumfang dar und definiert damit das Was und Wofür der Entwicklung. Das Lastenheft wird vom Auftraggeber erstellt und dient als Vertragsgrundlage zwischen Auftraggeber und Auftragnehmer. Das Pflichtenheft<sup>185</sup> enthält das Lastenheft und erweitert dieses um Realisierungsanforderungen und Umsetzungskonzepte. Das Pflichtenheft definiert damit das Wie und Womit die Anforderungen des Auftraggebers zu realisieren sind. Das Pflichtenheft wird in der Regel vom Auftragnehmer erstellt. Wesentliches Element zur Strukturierung von Ziele und Anforderungen im Lasten- und Pflichtenheft stellen Anforderungslisten dar<sup>186</sup>. LINDEMANN schlägt eine weitere Strukturierung der Anforderung als MindMaps vor<sup>187</sup>. Eine weitere Möglichkeit zur informalen Dokumentation stellen Wikis oder Software für das Aufgabenmanagement dar. Diese werden häufig im Rahmen agiler Entwicklungsmethoden verwendet, um ein Product Backlog oder User Stories (vgl. Kapitel 2.1.4) zu definieren.

---

<sup>182</sup> vgl. Pohl 2008

<sup>183</sup> vgl. Pohl 2008

<sup>184</sup> vgl. Verein Deutscher Ingenieure (VDI) 2001

<sup>185</sup> vgl. Verein Deutscher Ingenieure (VDI) 2001

<sup>186</sup> vgl. Ponn & Lindemann 2011

<sup>187</sup> vgl. Lindemann 2009, S.113

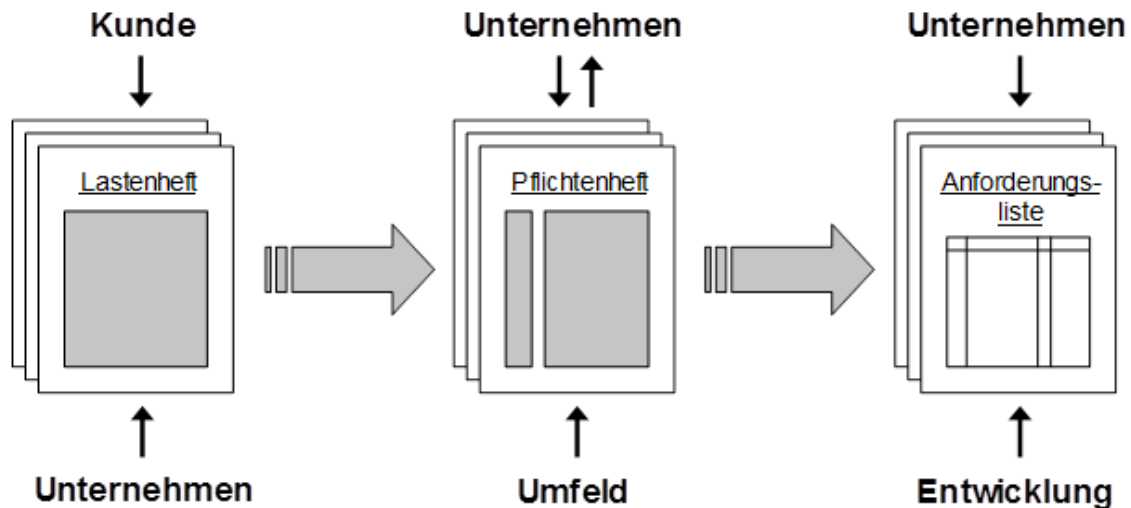


Abbildung 25: Abgrenzung von Lastenheft, Pflichtenheft und Anforderungsliste nach KRUSE<sup>188</sup>

Die natürlich-sprachige Dokumentation ist flexibel und vielseitig einsetzbar. Jeder der Stakeholder kann ohne große Einarbeitungszeit die Inhalte lesen und verstehen.<sup>189</sup> Problematisch an dieser Dokumentationsart sind allerdings die Mehrdeutigkeit der Sprache und die Vermischung der verschiedenen Perspektiven, die unterschiedliche Interpretationsmöglichkeiten und Missverständnisse zur Folge haben können.<sup>190</sup>

Des Weiteren kann die Verbindung zwischen Zielen, und Anforderungen verloren gehen, da diese auf unterschiedliche Dokumente verteilt sein können, die von unterschiedlichen Personen erstellt wurden. Aus einer mangelnden Vernetzung können erhebliche (Cross-Stage) Iterationen entstehen, da sich als realisierbar eingeschätzte Kundenwünsche als nicht (voll) umsetzbar herausstellen können oder wichtige (personengebundene) Informationen verloren gegangen sind.<sup>191</sup>

EBEL hat im Rahmen einer empirischen Untersuchung die Tauglichkeit von Wikis für die Abbildung von Zielsystemen untersucht. Dabei hat er insbesondere die prinzipiell nicht vorhandene Hierarchie der einzelnen Wiki-Seiten bemängelt. Die Untersuchungen zeigten, dass Entwickler zwar in vernetzten Zielen und Anforderungen denken, sich innerhalb dieser Strukturen jedoch klare Hierarchien ausbilden, die die Entwickler insbesondere als strukturierendes Kriterium für eine explizite Modellierung nutzen.<sup>192</sup>

<sup>188</sup> vgl. Kruse 1996

<sup>189</sup> vgl. Pohl & Rupp 2011

<sup>190</sup> vgl. Ebert 2014

<sup>191</sup> vgl. Braun, Ebel & Albers 2013

<sup>192</sup> vgl. Ebel 2015

ALBERS ET AL.<sup>193</sup> haben die Verwendung von MindMaps zur Modellierung von Zielsystemen untersucht, indem Ziele und Anforderungen aus Lastenheften extrahiert und in einer MindMap modelliert wurde. Dabei konnte festgestellt werden, dass sich eine hierarchische Struktur zwar gut mit MindMaps abbilden lässt, sich im Laufe der Entwicklung jedoch auf Grund unterschiedlicher Perspektiven keine eindeutige Hierarchie bildet, was mit MindMaps nicht abbildbar ist.

### Semi-formale Dokumentation

Wichtige Vertreter einer semi-formalen Dokumentation sind modellbasierte Ansätze. In dieser Dokumentationsform werden Anforderungen abstrahiert, um die Realität der Systemumgebung und das zu entwickelnde System geeignet abzubilden. Dafür werden Einheiten, mögliche Beziehungen zwischen diesen Einheiten und eine graphische Notation für ein Modell definiert. Zur Abbildung von Anforderungen in Modellen werden in der Praxis verschiedene Modellierungssprachen, die sich in Syntax und Semantik unterscheiden, verwendet. Beispiele für Modellierungssprachen sind unter anderem die aus dem Kontext der Softwareentwicklung stammende Unified Modeling Language (UML) und die daraus entwickelte Systems Modeling Language (SysML).

Mit Hilfe von SysML lassen sich die Struktur, das Verhalten und die Anforderungen eines Systems auf formale Art und Weise beschreiben und miteinander in Beziehung setzen<sup>194</sup>. Die SysML bietet hierfür verschiedene Diagrammtypen, auf deren Basis eine durchgängige Modellierung des Systems mittels grafischer Elemente erfolgen kann (vgl. Abbildung 26).

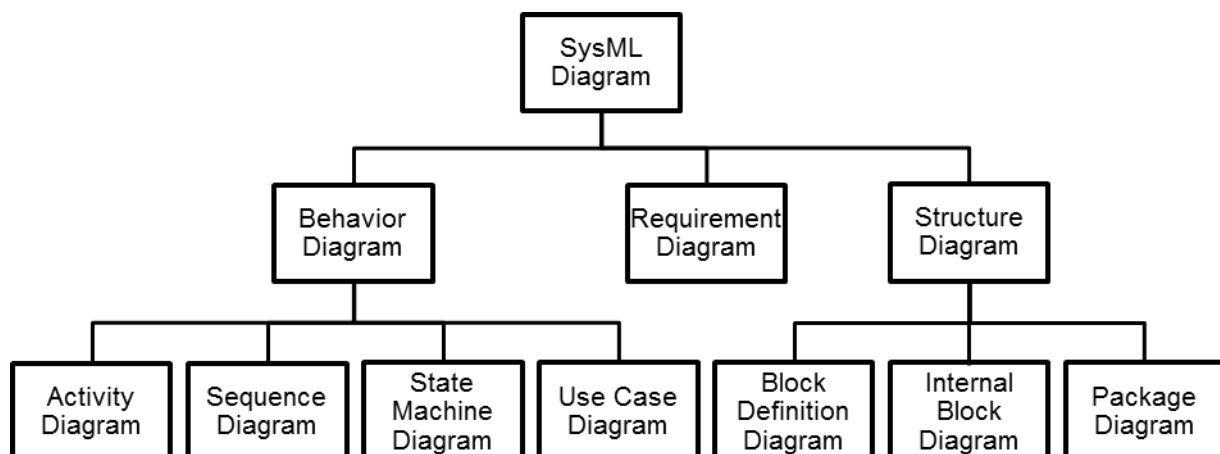


Abbildung 26: Diagrammtypen von SysML<sup>195</sup>

<sup>193</sup> vgl. Albers, Klingler & Ebel 2013

<sup>194</sup> vgl. Alt 2012

<sup>195</sup> vgl. Haskins et al. 2011

ALBERS und ZINGEL<sup>196</sup> nutzen und erweitern die SysML, um eine funktionsbasierte Modellierung Mechatronischer Systeme, die auf dem Contact & Channel Ansatz<sup>197</sup> aufbaut, realisieren zu können. GILZ und EIGNER<sup>198</sup> zeigen auf, wie SysML-Modelle in PLM integriert werden können, um deren Aktualität und Verfügbarkeit gewährleisten zu können.

SCHERER hat einen modellbasierten Ansatz zur Entwicklung von Zielsystemen im Kontext der Produktgenerationsentwicklung entwickelt. Er hat gezeigt, dass Zielsysteme dadurch deutlich effizienter und effektiver entwickelt werden können, weil bestehendes Wissen wiederverwendet werden kann.<sup>199</sup>

Durch modellbasierte Ansätze entsteht eine kompakte und perspektivenorientierte Darstellung der Anforderungen, deren Eindeutigkeit aufgrund der höheren Formalisierung größer ist.<sup>200</sup> Durchgängigkeit im Zielsystem kann direkt durch Verlinkung der Elemente hergestellt werden.

Die abstrahierten Modelle sind allerdings nur für geübte Leser leicht verständlich und können nicht universell eingesetzt werden.<sup>201</sup> IWANEK ET AL.<sup>202</sup> stellen fest, dass beispielsweise SysML mittlerweile in der Industrie als Modellierungssprache weit verbreitet ist. Gleichzeitig stellen sie fest, dass die Akzeptanz der Modellierungssprache bei Softwareentwicklern nach wie vor höher zu sein scheint, als bei Ingenieuren. Sie führen dies zum Einen darauf zurück, dass die SysML auf der UML (Unified Modeling Language) basiert und daher Konzepte aus der Software-Entwicklung in den Kontext der Systemmodellierung überträgt. Dabei erfordert insbesondere die Objektorientierung grundlegende Softwarekenntnisse und setzt ein Umdenken bei Entwicklungsingenieuren voraus. Zum Anderen fokussieren unterstützende Methoden vor allem auf die Entwicklung komplexer und softwareintensiver Systeme.

In der Praxis wird häufig eine Mischform der natürlich-sprachigen und modellbasierten Dokumentation von Anforderungen angewandt, sodass die Schwächen durch die Stärken der jeweiligen Möglichkeit kompensiert werden, z.B. Einbetten von Modellen in ein Lastenheft.<sup>203</sup> Durch diese Kombinationsform treten

---

<sup>196</sup> vgl. Albers & Zingel 2013

<sup>197</sup> vgl. Albers & Sadowski 2014

<sup>198</sup> vgl. Gilz & Eigner 2013

<sup>199</sup> vgl. Scherer 2016

<sup>200</sup> vgl. Ebert 2014

<sup>201</sup> vgl. Pohl & Rupp 2011

<sup>202</sup> vgl. Iwanek et al. 2013

<sup>203</sup> vgl. Ebel 2015



jedoch auch wieder Nachteile auf, zum Beispiel eine mangelnde Durchgängigkeit auf Grund fehlender Verlinkungen.

### **Formale Dokumentation**

Formale Semantiken werden derzeit hauptsächlich in der Informatik angewendet, um Software zu spezifizieren. Dies wird zum Beispiel für den Nachweis der Korrektheit von Computerprogrammen benötigt. Eine formale Semantik arbeitet dabei mit vollständig mathematischen Methoden.

Ohne genaue Kenntnis des Formalismus ist eine Formale Dokumentation zur Kollaboration nur wenig geeignet.

### **Tools**

Nach POHL UND RUPP müssen Werkzeuge für das Requirements Engineering folgende Eigenschaften haben:

- Verwalten verschiedener Informationen
- Verwalten von logischen Beziehungen zwischen Informationen
- Eindeutige Identifikation
- Bearbeiten der veralteten Informationen
- Bilden von unterschiedlichen Sichten auf die verwalteten Informationen je nach Einsatzzweck
- Organisieren der verwalteten Informationen
- Erstellen von Reports oder Auswertungen über die verwalteten Informationen
- Generieren von Ergebnisdokumenten unterschiedlicher Form aus den verwalteten Anforderungen

Untersuchungen von EBEL<sup>204</sup> haben gezeigt, dass die Akzeptanz für ein Werkzeug zur Modellierung von Zielsystemen insbesondere dann gegeben ist, wenn es intuitiv bedienbar ist und ein Nutzen aus der Modellierung gezogen werden kann, der über die reine Dokumentation hinausgeht und der bereits bei moderatem Modellierungsaufwand sichtbar wird. Darüber hinaus hat sich gezeigt, dass ein Werkzeug zur Zielsystemmodellierung folgenden Aspekten Rechnung tragen sollte:

- Wiederverwendbarkeit von Zielsystemen bzw. Wiederverwendbarkeit von Erkenntnissen aus Projekten mittels einer Durchgängigkeit von unternehmensweiter Wissensbasis und projektspezifischen Zielsystemen.
- Berücksichtigung spezifischer Änderungsautorisationen für jedes einzelne Element des Zielsystems.

---

<sup>204</sup> vgl. Ebel 2015

- Identifikation relevanter Änderungen bzw. Identifikation von Änderungen an relevanten Elementen des Zielsystems.

EBERT<sup>205</sup> gibt einen Überblick über in der Praxis verbreitete Werkzeuge zur Dokumentation von Anforderungen (vgl. Abbildung 27). DOORS von IBM wird hierbei als Markt- und Technologieführer im Bereich des Anforderungsmanagements angesehen, da es sich insbesondere für große Projekte eignet.

EBERT gliedert die Werkzeuge zur Dokumentation von Anforderungen weiter in 5 Kategorien:<sup>206</sup>

Mit Hilfe von **Office-Tools** können vor allem in kleinen Projekten auf Basis von selbst entworfener Vorlagen wirksame Kontrollen von Anforderungen im Anforderungsmanagement durchgeführt werden. Vorteilhaft bei diesen Tools sind die leichte und flexible Handhabung, die Anpassbarkeit und die direkte Nutzung verschiedener Anspruchsträger aufgrund einer weiten Verbreitung. Der Einsatz von Office-Tools sollte jedoch nur als Unterstützung anderer Tools dienen und nicht als alleiniges Tool eingesetzt werden.

Vertreter: Microsoft Excel, Microsoft Word

**Wiki-Technologien** werden meist in agilen und kleinen Unternehmen eingesetzt, da diese einen unmittelbaren Zugriff auf Anforderungen durch unterschiedliche Benutzer zur Organisation dieser erlauben. Allerdings wird dabei ein gewisses Maß an Formalismus benötigt, um alle Statusinformationen und Attribute uneingeschränkt nutzen zu können. ALBERS ET AL. implementieren das iPeM Vorgehensmodell in einem semantischen Wiki, um Wissensmanagement in der Produktentwicklung zu unterstützen.<sup>207</sup>

Vertreter: Fit-Nesse, OpenCollective, Atlassian Confluence

Mittels sogenannter **Workflows** lässt sich die Abfolge von Schritten im RE besonders gut beschreiben. Sie informieren automatisch die jeweiligen verantwortlichen Personen, dass Anforderungen, Fehlermeldungen, Testfälle, etc. zu bearbeiten sind. Oftmals zeigen diese Tools jedoch Schwächen bei der Verwaltung und Versionierung von Dokumenten auf, da sie aus dem Bereich des Projektmanagements kommen. Vertreter: Bugzilla, Redmine, Trac, Mylyn)

Des Weiteren werden im Anforderungsmanagement Entwicklungsumgebungen und **Modellierungswerkzeuge** (Anwendungslebenszyklus- (ALM) und **Produktlebenszyklus-(PLM)** Werkzeuge) eingesetzt. Sie ermöglichen durch ein

---

<sup>205</sup> vgl. Ebert 2014

<sup>206</sup> vgl. Ebert 2014

<sup>207</sup> vgl. Ebert 2014

Hauptnetz, welches alle Daten miteinander verknüpft, einen transparenten Austausch dieser und halten die Inhalte konsistent. Dabei stellen sie häufig noch zusätzliche Kommunikationsmechanismen wie beispielsweise Trigger zur Verfügung, um andere Anwendungen darauf hinzuweisen, dass Änderungen vorliegen. Die Anwendung von ALM-/PLM-Werkzeuge sorgen folglich für eine durchgängige Unterstützung bei der Ermittlung, Verwaltung und Pflege von Anforderungen über den gesamten Lebenszyklus hinweg.

Vertreter: PreeVision, Visual Studio, Enterprise Architect, InStep

**Spezielle RE-Tools** wurden explizit für das Anforderungsmanagement entwickelt und unterstützen vor allem die Verwaltung und Nachverfolgung von Anforderungen, Spezifikationen und weiteren Dokumenten. Sie verfügen über eine gute graphische Umgebung, sodass die Beschreibung von Anforderungen, Workflows, Abhängigkeitsbeziehungen und Modellen erleichtert wird und gewährleistet durch Texteditoren die Austauschbarkeit von Anforderungen. Die Benutzeroberfläche dieser Tools ist denen der Office-Tools ähnlich und verfügen beispielsweise über Import- und Exportmechanismen sowie vordefinierte Tabellenformate. Zudem erlauben Filter Verknüpfungen, Gruppierungen, die Weiterbearbeitungen und Verwaltung komplexer Anforderungen und deren Beziehungen. Vertreter: DOORS, Integrity, Quality Center

Immer häufiger wird im Requirement Engineering zudem auf **Agile Cloud-Lösungen** zurückgegriffen, die einen schnellen Zugriff auf Anforderungen für verschiedene Stakeholder ermöglichen. Ein gewisses Risiko liegt bei diesen Tools jedoch in der Performance, der Verfügbarkeit und der Sicherheit, welches sich nachteilig auf das Anforderungsmanagement auswirken kann. Vertreter: Contour, Jira, Polarion und Rally

Trotz dieser Ansätze und der Vielzahl an verfügbaren Tools bleibt Microsoft Office weiterhin eines der meist genutzten Werkzeuge zur Abbildung von Zielsystemen<sup>208</sup>. Dies kann auf die hohe subjektive Effizienz dieses Werkzeugs und der damit verbundenen bzw. umsetzbaren Methoden zurückgeführt werden. SAUTER<sup>209</sup> definiert die subjektive Effizienz als das Verhältnis aus subjektivem Nutzen zu subjektivem (Mehr-)Aufwand, die ein Anwender bei der Verwendung eines Systems empfindet (vgl. Abbildung 28).

---

<sup>208</sup> vgl. Hood et al. 2008

<sup>209</sup> vgl. Albers, Ebel & Sauter 2010, S. 334

Werkzeug	Hersteller	Schlüsselfunktionen	Kosten
DOORS	IBM	Dokumentation, große Projekte, Produktlinien, Nachverfolgbarkeit, XML-Unterstützung, ReqIF, Anbindung an UML-Tools, Testmanagement-Tools, PLM-Tools, MS Projekts	Hoch
Enterprise Architect	Sparx Systems	Dokumentation, UML-basierte Analyse und Modellierung, Nachverfolgbarkeit, Teststatus, XML-Unterstützung	Mittel
IRqA	Visure	Dokumentation, OO-Analyse und ERM, Nachverfolgbarkeit, Teststatus, XML-Unterstützung, ReqIF, MS-Office-Anbindung	Mittel
MKS Integrity	PTC	ALM-Umgebung mit Komponenten für RE, Dokumentation, Analyse, Nachverfolgbarkeit, Konsistenzsicherung, XML-Unterstützung, ReqIF, MS-Office-Anbindung	Hoch
OSRMT	Open-Source	Einstiegswerkzeug, Dokumentation, Projektmanagement, Konfigurationsmanagement, Modellierung	Niedrig
Polarion Requirements	Polarion	Web-2.0-Umgebung, internetbasiertes RE, Anbindung an ALM-Umgebung, Workflow-Management	Mittel
PREEvision	Vector	PLM-/ALM-Umgebung mit Data-Backbone zur werkzeugübergreifenden Datenkonsistenz, ReqIF-Import/Export, Change-, Projekt, Testdatenmanagement	Hoch
Reqtify	Geensys	Dokumentation, Nachverfolgbarkeit, MS-Office-Anbindung, Simulink-Anbindung	Mittel
Requirements Composer	IBM	Zentralisierte Plattform für die Erhebung, Analyse, Validierung und Verwaltung von Anforderungen, gemeinsame Arbeit an den Anforderungen im Team, Einbeziehung von Kunden und Fachabteilungen, nahtlose Integration in Lifecycle	Hoch
RMF	various	Eclipse-basiert, Einstiegswerkzeug, ReqIF-Datenaustausch	Niedrig
RMTrak	RBC	Einstiegswerkzeug, Dokumentation, MS-Office, SQL Interface	Niedrig
Truereq PLM	Truereq	Einstiegswerkzeug, Dokumentation, XML-Unterstützung, kostenlose Einzelplatzlizenz	Niedrig

Abbildung 27: Werkzeuge des Requirements Engineering<sup>210</sup>

ALBERS ET AL.<sup>211</sup> haben herausgefunden, dass digitale Ordnerstrukturen noch immer zu den häufigsten Ablageorten für Dokumente gehören. Am zweithäufigsten werden

<sup>210</sup> vgl. Sauter 2011

<sup>211</sup> vgl. Albers et al. 2014

Dokumentenmanagementsysteme genannt. Insbesondere bei der Verwendung von Microsoft Office zur Dokumentation von Anforderungen und Zielen erhalten Dokumentenmanagementsysteme eine hohe Bedeutung. Sie ermöglichen einen schnelleren Informationszugriff und höhere Dokumentensicherheit. Ein Beispiel für ein Dokumenten Management System ist Microsoft SharePoint oder SAP für Enterprise Resource Planning. Dies bestärkt den Stand der Forschung, wonach häufig eine dokumentenzentrierte Produktentwicklung vorliegt.<sup>212</sup>

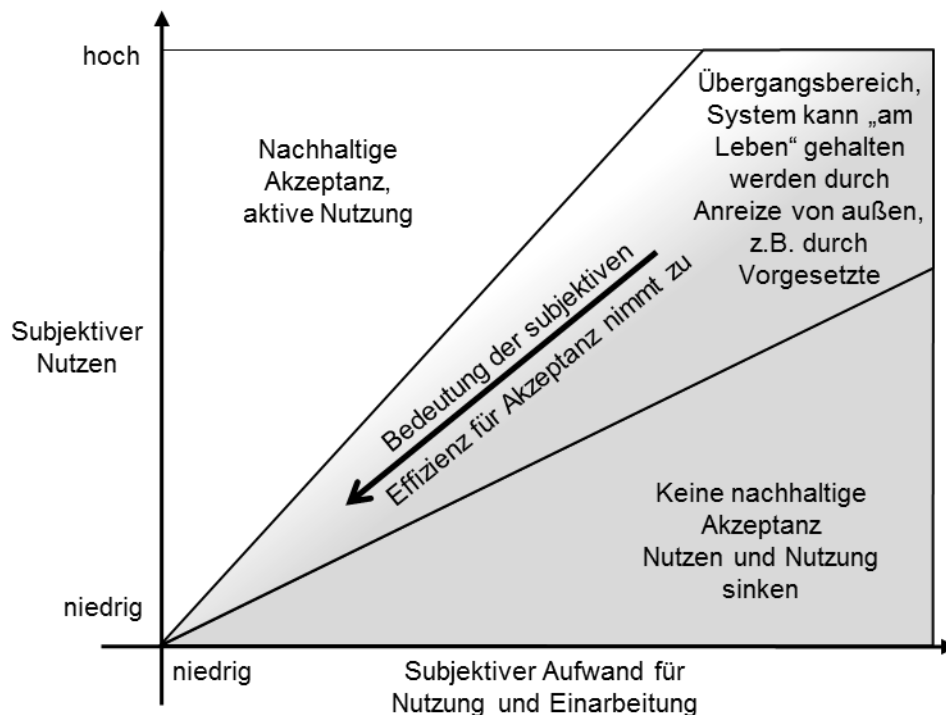


Abbildung 28: Subjektive Effizienz von Methoden<sup>213</sup>

### Abbildung von Relationen im Zielsystem

Um Durchgängigkeit im Zielsystem realisieren zu können, müssen sämtliche relevanten Relationen abgebildet werden. Die Abbildung von Relationen im gesamten Zielsystem hat folglich eine große Bedeutung. Werden wesentliche Inkonsistenzen bis zur Einführung nicht identifiziert, kann es zum Scheitern des Produkts kommen.

ALBERS UND LOHMEYER<sup>214</sup> definieren Durchgängigkeit von Modellen als die gleichzeitige Realisierung horizontaler und vertikaler Durchgängigkeit sowie Konsistenz (vgl. Abbildung 29). Horizontale Durchgängigkeit bedeutet, dass ein

<sup>212</sup> vgl. Bursac 2016, S. 197

<sup>213</sup> vgl. Sauter 2011

<sup>214</sup> vgl. Albers & Lohmeyer 2012

Modell entlang des gesamten Produktlebenszyklus verwendet werden kann. Vertikale Durchgängigkeit bezieht sich auf den Detaillierungsgrad, den ein Modell abbildet. Das bedeutet, dass ein kontinuierlicher Wechsel der Betrachtungsebene in Hinblick auf den Detaillierungsgrad möglich sein muss. Konsistenz bedeutet, dass zwei Modelle, die dasselbe Original repräsentieren, keine widersprüchlichen Aussagen beinhalten.

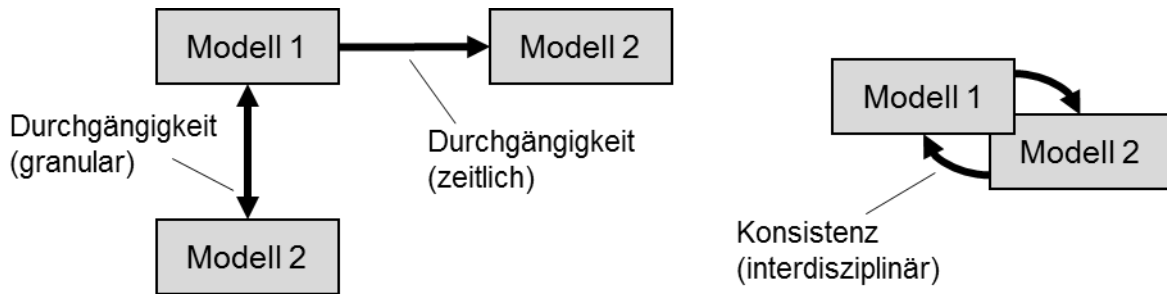


Abbildung 29: Durchgängigkeit und Konsistenz von Modellen<sup>215</sup>

Zur Operationalisierung des Zielsystems fordert EILETZ<sup>216</sup> Durchgängigkeit der Ziele von abstrakten strategischen Zielen bis hin zu konkreten Detailzielen.

KRUSE<sup>217</sup> unterscheidet dabei zwei Arten von Beziehungen im Zielsystem. Strukturbeziehungen werden verwendet, um eine Hierarchie im Zielsystem zu bilden, zum Beispiel durch Konkretisierung, Spezialisierung, Dekomposition oder Variation einzelner Elemente. Semantikbeziehungen beschreiben die Wirkungen einzelner Elemente aufeinander, zum Beispiel ausschließend, konkurrierend und unterstützend. Nach GEBAUER<sup>218</sup> können Beziehungen dabei nicht nur als Graph sondern auch als Matrix abgebildet werden (vgl. Abbildung 30).

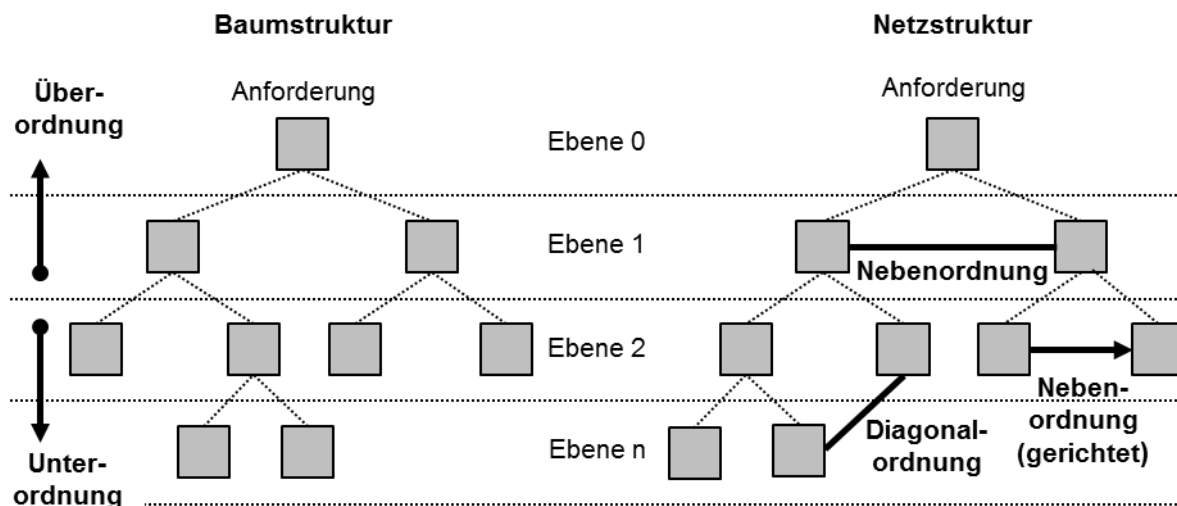
Die Untersuchungen von EBEL haben gezeigt, dass für eine ganzheitliche und durchgängige Modellierung von Zielsystemen eine Reihe angrenzender Partialmodelle der Produktentwicklung berücksichtigt werden müssen.

<sup>215</sup> vgl. Albers & Lohmeyer 2012

<sup>216</sup> vgl. Eiletz 1999

<sup>217</sup> vgl. Kruse 1996

<sup>218</sup> vgl. Gebauer 2000, S. 48

Abbildung 30: Grundsätzliche Beziehungsarten im Zielsystem<sup>219</sup>

Er definiert neun Partialmodelle, die miteinander in Wechselwirkung stehen:<sup>220</sup>

**Ziele:** Dieses Partialmodell umfasst alle Ziele und definiert damit den Soll-Zustand dessen, was durch das Entwicklungsprojekt erreicht werden soll.

**Anforderungen:** Dieses Partialmodell beinhaltet sämtliche Anforderungen, die das zu entwickelnde Produkt erfüllen muss, um die definierten Ziele zu erreichen.

**Anwendungsfälle:** Das Partialmodell der Anwendungsfälle beinhaltet sämtliche (formalen) Beschreibungen der Interaktionen zwischen dem zu entwickelnden System und den Systemnutzern (Personen und Systeme).

**Funktionen:** Partialmodell mit allen Funktionen und deren Verkettungen, die zur Realisierung der formulierten Ziele und Anforderungen erforderlich sind.

**Gestalt/Implementation:** Dieses Partialmodell beinhaltet die Gestalt bzw. Implementation der in den Anforderungen beschriebenen Eigenschaften zur Verwirklichung der vereinbarten Ziele.

**Phasen und Produktentstehungsaktivitäten:** Dieses Partialmodell umfasst die Handlungen und Entwicklungsphasen, die erforderlich sind, um die formulierten Ziele zu erreichen.

**Meilensteine und Deliverables:** Hierin sind alle Meilensteine und Deliverables und damit sämtliche Zeitpunkte und Ereignisse enthalten, zu denen der Projektfortschritt explizit zu überprüfen ist und wichtige Entscheidungen getroffen werden müssen.

<sup>219</sup> vgl. Gebauer 2000, S. 153ff

<sup>220</sup> vgl. Ebel 2015

**Stakeholder:** In diesem Partialmodell sind alle Personen, Gruppen oder Organisationen enthalten, die ein Interesse am Produktentstehungsprozess haben.

**Tests:** Um die zuvor definierten Ziele und Anforderungen hinsichtlich ihrer Erfüllung zu überprüfen beinhaltet dieses Partialmodell alle Tests, die im Rahmen eines Produktentstehungsprozesses durchgeführt werden.

Diese Partialmodelle werden nicht sequentiell und losgelöst voneinander entwickelt, sondern kontinuierlich im Verlauf des Produktentstehungsprozess in einem iterativen Wechselspiel<sup>221</sup>. Zur Herstellung von Durchgängigkeit zwischen Elementen der Partialmodelle schlägt EBEL 13 Typen von Relationen vor.<sup>222</sup>

**Relationen entlang der Hierarchie** definieren die jeweils als dominant definierte Wechselwirkung eines Elements. Vertreter: Dekomposition und Verfeinerung.

**Generische Relationen** setzen zwei Elemente in Beziehung, ohne die Art der Beziehung genauer zu benennen. Vertreter: Ungerichtete und gerichtete Relationen.

**Verträglichkeits-Relationen** definieren, wie kompatibel Elemente der Partialmodelle zueinander sind. Vertreter: Unterstützung, Konflikt und Inkonsistenz.

**Ziel-Objekt-Relationen** benennen die Wechselwirkungen zwischen Zielen, Anforderungen, Funktionen und der Gestalt bzw. Implementation dieser. Vertreter: Ableitung und Erfüllung.

**Verifizierungs- und Validierungsrelationen:** Diese Relationstypen stellen einen Zusammenhang zwischen formulierten Zielen und Anforderungen und definierter Tests dar. Vertreter: Verifizierung und Validierung.

**Handlungsrelationen** beschreiben Wechselwirkungen zu Elementen des Handlungssystems. Vertreter: Verantwortlichkeit und Treiber.

#### 2.2.4 Zwischenfazit

Ein Zielsystem beinhaltet alle expliziten Ziele, Anforderungen und Randbedingungen eines zu entwickelnden Produktes zu einem bestimmten Zeitpunkt. Ein Ziel ist dabei die Definition dessen, was erreicht werden soll. Anforderungen definieren die Eigenschaften des Produkts, die realisiert werden sollen und leiten sich aus den Zielen ab. Randbedingungen haben einen direkten Einfluss auf die Anforderungen und definieren Einschränkungen, welchen bei der Entwicklung berücksichtigt werden müssen.

---

<sup>221</sup> vgl. Ebel 2015

<sup>222</sup> vgl. Ebel 2015



Angesichts der Dynamik der Produktentstehung ändern sich Ziele, Anforderungen und Randbedingungen kontinuierlich. Dies kann sowohl auf interne als auch auf externe Faktoren zurückgeführt werden. Die Synthese des Zielsystems stellt daher einen iterativen Abstimmungs- und Einigungsprozess dar, in welchem auf Basis eines unsicherheitsbehafteten Wissensstandes entschieden wird, welche Ziele in welcher Form zu definieren sind. Bestehende Ansätze für die Entwicklung von Zielsystemen und für das Management von Änderungen bei Zielen und Anforderungen müssen insbesondere für die Unterstützung von verteilten und teilweise unabhängig voneinander agierenden Handlungssystemen weiterentwickelt werden.

Entwickler nutzen zur Interaktion im Team eine äußere Ich-Ebene, das heißt sie explizieren ihre vorgedachten und vorausgewählten Ziele und Anforderungen in Form von Dokumenten und Modellen. Um Durchgängigkeit im Zielsystem realisieren zu können, müssen dabei sämtliche relevanten Relationen verfügbar gemacht werden. Die Herstellung von Durchgängigkeit und Konsistenz im gesamten Zielsystem stellt folglich eine große Bedeutung dar. Werden wesentliche Inkonsistenzen bis zur Einführung nicht identifiziert, kann es zum Scheitern des Produkts kommen. In der Praxis werden eine Vielzahl von Methoden und Tools zur Explikation von Zielsystemen verwendet. Bei modellbasierten Ansätzen werden Anforderungen abstrahiert, um die Realität der Systemumgebung und des zu entwickelnde Systems geeignet abzubilden. Dadurch kann Durchgängigkeit im Zielsystem durch Verlinkung der Elemente hergestellt werden. Die abstrahierten Modelle sind allerdings nur für geübte Leser leicht verständlich und können nicht universell eingesetzt werden. Die am weitesten verbreitete Dokumentationsform basiert noch immer auf der informalen Sprache. Dies ist insbesondere mit einer großen Flexibilität und schnellen Verständlichkeit zu erklären. Trotz einer Vielzahl an verfügbaren Tools bleibt Microsoft Office daher weiterhin eines der meist genutzten Werkzeuge zur Dokumentation von Zielen und Anforderungen. Das Herstellen von Durchgängigkeit ist insbesondere dann eine Herausforderung und Ansätze hierfür fehlen noch, wenn Relationen zwischen Entwicklungsartefakten existieren, sich die zugrunde liegende Dokumentationsform jedoch unterscheidet. Es müssen daher noch Ansätze entwickelt werden welche dabei unterstützen, Durchgängigkeit und Konsistenz im Zielsystem herzustellen, auch wenn heterogene Dokumentationsformen in verteilten Handlungssystemen gewählt werden.

Dokumenten Management Systeme können durch Verwendung von semantischen Modellen und Techniken dabei unterstützen, einen schnelleren Informationszugriff und eine höhere Dokumentensicherheit auch bei heterogene Entwicklungsartefakten zu ermöglichen. Daher werden im nächsten Kapitel bestehende semantische Modelle und Techniken vorgestellt.

### 2.3 Semantische Modelle und Techniken

Als Semantik wird im Allgemeinen ein Teilgebiet der Linguistik bezeichnet, welche sich mit Sinn und Bedeutung von Sprache oder (sprachlichen) Zeichen befasst.<sup>223</sup>

In den Computerwissenschaften wird Semantik häufig aufgegriffen mit dem Ziel, das Wissensmanagement zu unterstützen.<sup>224</sup> Wissen wird basierend auf der Projektion eines Realitätsausschnitts in eine entsprechende semantische Repräsentation auf einem Computer formalisiert.<sup>225</sup> Je strukturierter und expliziter die Semantiken (Bedeutungen) vorliegen, desto leistungsfähiger werden die Möglichkeiten, um mittels eines Computers mit dem modellierten Zielsystem zu arbeiten<sup>226</sup>.

Prominentes Beispiel ist das sogenannten „Semantic Web“ oder auch „Web of Data“<sup>227</sup>, welches durch das World Wide Web Consortium (W3C)<sup>228</sup> getrieben wird. Das W3C entwickelt, veröffentlicht und empfiehlt in diesem Kontext einheitliche Web-Standards, zu denen mitunter das Resource Description Framework (RDF) und OWL gehören.<sup>229</sup> Das „Semantic Web“ stellt eine Erweiterung des World Wide Webs um Metadaten sowie Beziehungen dar. Das bietet die Möglichkeit, digitale Inhalte über Metadaten explizit eine Bedeutung zuzuordnen, indem die Inhalte in einen übergeordneten Kontext eingeordnet werden.

Unter Metadaten versteht man hierarchisch geordnete sowie strukturierte Informationen.<sup>230</sup> Im Allgemeinen dienen Metadaten dazu, Aussagen über einen bestimmten Datensatz zu treffen.

Metadaten können durch Maschinen interpretiert werden, um auf Basis von Regeln neue (semantische) Metadaten abzuleiten (Inferenz). Weiterhin ermöglichen sie eine Kontext-spezifische Suche und damit einen effizienteren Zugriff auf explizit vorhandenes Wissen. Somit ermöglichen die semantischen Metadaten das Erschließen neuer Informationen.<sup>231</sup>

Metadaten werden zum Beispiel im Produktlebenszyklusmanagement (PLM) verwendet. Ein PLM-Ansatz unterstützt dabei alle Phasen des Produktlebenszyklus

---

<sup>223</sup> vgl. Baier 2008, S.18 und Blumauer & Pellegrini 2006, S. 9

<sup>224</sup> vgl. Baier 2008, S.18.

<sup>225</sup> vgl. Ebel 2015, S.71

<sup>226</sup> vgl. Sauter 2012, S. 67

<sup>227</sup> vgl. Dengel 2012 – Linked Data

<sup>228</sup> vgl. <http://www.w3.org/>, Letzter Zugriff am 17.12.2016

<sup>229</sup> vgl. <http://www.w3.org/>, Letzter Zugriff am 17.12.2016

<sup>230</sup> vgl. Duden Einträge: Meta und Daten

<sup>231</sup> vgl. Hitzler et al. 2008

indem er Menschen, Prozesse und Technologien integriert, um Konsistenz und Nachvollziehbarkeit zu erhöhen.<sup>232</sup>

SCHÜTZ<sup>233</sup> grenzt die unterschiedlichen Arten von Metadaten mit der folgenden analytischen Kategorisierung voneinander ab:

Tabelle 2-1: Grundlagen der Metadatenarten im Dokumentenmanagement nach Schütz<sup>234</sup>

Metadatenarten	Definition	Umfang
Semantische	Beschreiben den Inhalt der Dokumente für einen besseren suchenden und navigierenden Zugriff auf ein Dokument. Hierzu gehören außerdem qualitative bzw. bewertende und quantitative Metadaten sowie relationale Metadaten bzgl. zweier Dokumente	Inhalterschließende Attribute (z.B. Volltextindizierungswerte, Schlagwörter und Abstracts); klassifizierende Attribute; Anzahl an Zugriffen
Strukturelle	Beschreiben den inneren Aufbau von Dokumenten (Mikrostruktur)	Titel, Betreff, Kapitel, Abschnitt
Administrative	Dienen der Verwaltung von Dokumenten	Zugriffsrechte, Bearbeitungsstatus, Copyrights
Technische	Beziehen sich auf technische Eigenschaften von Dokumenten und enthalten Verarbeitungshinweise	Formatierung, Dateiformat und so weiter

STOCK & STOCK<sup>235</sup> schlagen eine weitere Form der Differenzierung vor. Sie legen das Hauptaugenmerk auf die Relation, welche die Metadaten zum Objekt besitzen:

<sup>232</sup> vgl. Abramovici et al. 2009

<sup>233</sup> vgl. Schütz 2004, S.340

<sup>234</sup> vgl. Schütz 2004

<sup>235</sup> vgl. Stock & Stock 2008, S.123f

Tabelle 2-2: Formen von Metadaten; zusammengefasst nach *Stock & Stock*<sup>236</sup>

Metadatenformen	Bedeutung	Beispiel
Formale Relation	Identifikation des Objekts	hat Autor hat Verlag hat Titel hat ISBN usw.
Dokumenten- relationen	Zusammenhänge von Objekten	Primär: Werk, Ausdrucks- form  Sekundär: äquivalente, derivative, deskriptive
Erschließungs- relationen	Inhaltliche Auswertung	Verdichtung des Inhalts z.B. mittels Abstracts
Relationen der Intertextualität	Im Sinne von Referenzen oder Zitationen	Zitiert x Zitiert von y Verlinkt auf z
Nicht-thematische Informationsfilter	Attribute wie Zielgruppe oder Genre	Qualität Kanal Ziele des Inhalts Schwierigkeitsgrad
Verwaltungs- relationen	Angaben zu Abläufen usw.	Bearbeiter Erstellungsdatum

Eine formale oder auch logische Semantik ist erforderlich, um Inkompatibilitäten und Inkonsistenzen zu vermeiden und somit korrektes Schlussfolgern zu ermöglichen.<sup>237</sup>

Nach DENGEL<sup>238</sup> kommt eine Semantik im formalen Sinne dadurch zustande, dass ein Begriff unzweideutig gebraucht wird und damit nur eine Bedeutung hat. Ein Begriff muss also „wohlunterscheidbar“ sein von anderen Begriffen.

<sup>236</sup> vgl. Stock and Stock 2008 S.91f

<sup>237</sup> vgl. Hitzler et al. 2008

<sup>238</sup> vgl. Dengel 2012 sowie Staab 2002

### Semantische Modelle

Semantische Modelle sollen die Kommunikation unter Menschen, unter Maschinen sowie die Mensch-Maschinen Kommunikation unterstützen.<sup>239</sup> Die dabei auftretenden Herausforderungen werden im semiotischen Dreieck<sup>240</sup> verdeutlicht (vgl. Abbildung 31).

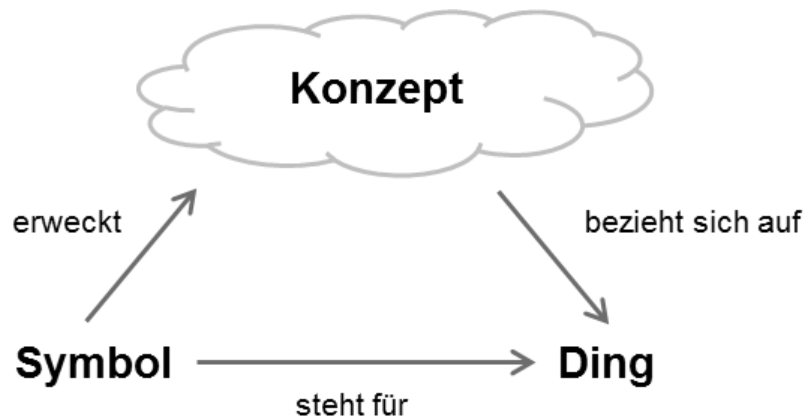


Abbildung 31: Das semiotische Dreieck – Rahmenbedingung der Ontologie nach MAYANK, KOSITSYNA & AUSTIN<sup>241</sup>

Ohne ein definiertes, einheitliches Konzept ist eine reibungslose Kommunikation zwischen Sender und Empfänger einer Information nicht möglich.<sup>242</sup> MAEDCHE<sup>243</sup> erläutert dies an einem Beispiel. Das „Symbol“ Jaguar wird vom Sender an den Empfänger übermittelt. Diesem ist es jedoch nicht möglich zu wissen, ob es sich hierbei um das Tier, die Automarke oder einem Begriff aus der Computertechnik handelt. Es muss demnach das einheitliche „Konzept“ berücksichtigt werden, um von einem „Symbol“ auf das referenzierte „Ding“ schließen zu können.

ULLRICH, MAIER & ANGELE<sup>244</sup> unterscheiden semantische Modelle bzgl. ihrer semantischen Mächtigkeit (vgl. Abbildung 32).

<sup>239</sup> vgl. auch Mayank, Kositsyna & Austin 2004, Ogden & Richards 1923 sowie Peirce 1966 und Bolzano 1810

<sup>240</sup> vgl. Mayank, Kositsyna & Austin 2004

<sup>241</sup> vgl. Mayank, Kositsyna & Austin 2004, S.64f

<sup>242</sup> vgl. Dengel 2012

<sup>243</sup> vgl. Maedche 2002

<sup>244</sup> vgl. Ulrich, Maier & Angele 2003

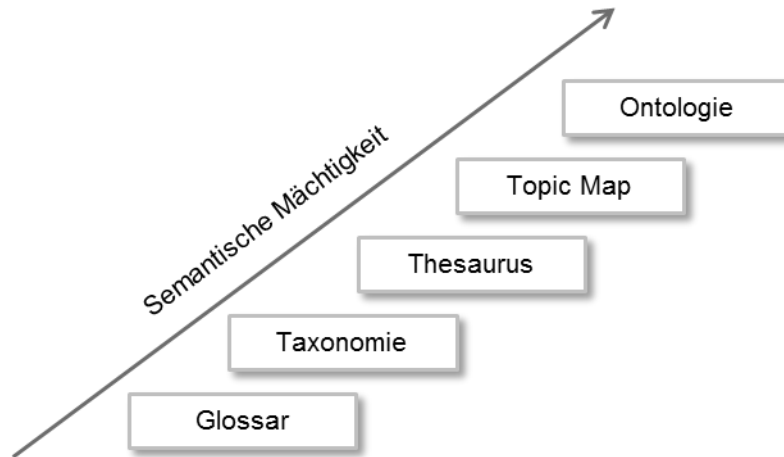


Abbildung 32: Semantische Mächtigkeit der semantischen Modelle  
nach ULLRICH, MAIER & ANGELE<sup>245</sup>

Das Glossar stellt eine Liste von Wörtern mit beigefügten Erklärungen dar. Eine Taxonomie oder auch Klassifikationsschema ist ein Modell, mit dem Objekte nach bestimmten Kriterien klassifiziert werden. Sie erleichtern den Umgang mit Einzelfällen und ermöglichen summarische Aussagen, die bis hin zu einer Erklärung von Zusammenhängen führen können. Ein Thesaurus, oder auch Wortnetz genannt, bezeichnet ein kontrolliertes Vokabular, dessen Begriffe durch Relationen miteinander verbunden sind. Eine Topic Map dient der Sammlung von Wissen über ein Subjekt. Eine Ontologie zeichnet sich hierbei durch die höchste semantische Mächtigkeit aus.

Der Begriff Ontologie wurde ursprünglich in der Philosophie geprägt, und bezeichnet dort die „Lehre vom Sein“.<sup>246</sup>

GRUBER<sup>247</sup> definiert eine Ontologie als eine explizite Spezifikation einer Begriffsbildung. STUDER, BENJAMINS UND FENSEL<sup>248</sup> erweitern diese Definition weiter. Für Sie ist eine Ontologie eine formale, explizite Spezifikation einer geteilten Begriffsbildung. Der formale Charakter einer Ontologie soll ein maschinell interpretierbares, abstraktes Modell gewährleisten.<sup>249</sup>

NOY UND MCGUINNESS verstehen unter einer Ontologie die formale explizite Beschreibung von Konzepten in einer Domäne, Beziehungen zwischen diesen

<sup>245</sup> vgl. Ulrich, Maier & Angele 2003

<sup>246</sup> vgl. Heidegger 1955

<sup>247</sup> vgl. Gruber 1993

<sup>248</sup> vgl. Studer, Benjamins & Fensel 1998

<sup>249</sup> vgl. Noy & McGuinness 2001, S. 3

Konzepten (Relationen), Eigenschaften der Konzepte (z.B. Attribute) und Einschränkungen der Attribute.<sup>250</sup>

Die Verwendung von Ontologien im Bereich des Wissensmanagements unterstützt die Integration von Mensch, Organisation und Informationstechnologie (vgl. Abbildung 33). Nach ALBERS ET AL.<sup>251</sup> ermöglichen Ontologien einen effizienten Informationsaustausch, verhindern Missverständnisse und damit resultierende Fehler.

DENGEL<sup>252</sup> beschreibt den Wesensgehalt einer Ontologie wie folgt:

- Nutzung gemeinsamer Symbole und Begriffe im Sinne einer Syntax
- das gemeinschaftliche (Ein-)Verständnis bezüglich deren Bedeutung, also der Semantik
- Klassifikation der Begriffe in Form einer Taxonomie
- Vernetzung der Begriffe mit Hilfe von assoziativen Relationen bei gleichzeitiger Festlegung von Regeln und Definitionen darüber, welche Relationen sinnvoll und erlaubt sind

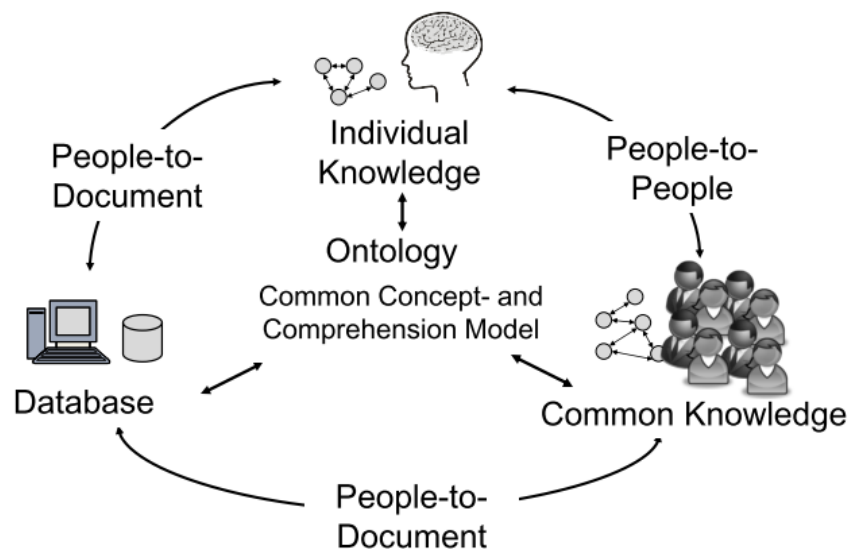


Abbildung 33: Ontologien als gemeinsames Verständnismodell<sup>253</sup>

<sup>250</sup> vgl. Dengel 2012

<sup>251</sup> vgl. Albers, Lohmyer & Schmalenbach 2010, S.65

<sup>252</sup> vgl. Dengel 2012

<sup>253</sup> vgl. Schütt 2007 und Voß 2007

### 2.3.1 Tagging

Mit Tagging bezeichnet man das freie Erweitern von digitalen Inhalten (Objekten, Ressourcen) mit Metadaten in Form von Schlagwörtern (Tags).<sup>254</sup> Der Begriff leitet sich aus dem englischen Begriff „Tag“ für Zettel, Marke oder Anhang ab.

Für das Tagging (Deutsch: Verschlagworten) durch Gruppen von Personen in Sozialen Netzwerken wird häufig auch der Begriff Social Tagging oder Collaborative Tagging verwendet. Tagging von Webseiten wird auch als Social Bookmarking bezeichnet.<sup>255</sup>

Ziel von Tags ist es, Nutzern schnell das Wichtigste zum Inhalt von Objekten und deren Relationen zu vermitteln, was insbesondere beim Auffinden von Informationen unterstützen soll. Tags können beim Tagging in der Regel frei vergeben werden und sind nicht durch ein vorgegebenes Vokabular oder eine starre Taxonomie eingeschränkt. Ein wichtiger Vorteil besteht somit darin, dass individuelle Assoziationen und Denkmodelle bei der Vergabe der Schlagwörter genutzt werden können, wodurch ein intuitiver Zugriff auf die Informationen ermöglicht wird.<sup>256</sup> Die Erweiterung vorhandener Objekte durch Tags ermöglichen somit die Inhaltsexploration und die Wissensverlinkung.<sup>257</sup>

Die Einführung von Beziehungen zwischen Benutzern, Ressourcen und Tags ergeben konzeptionelle Strukturen<sup>258</sup>, welche auch als Folksonomien bezeichnet werden. Der Begriff wurde im Jahr 2004 durch VANDER WAL<sup>259</sup> geprägt. Er definiert Folksonomie als das Ergebnis vom persönlichen, freien Verschlagworten von Informationen und Objekten für den persönlichen/individuellen Zugriff.

Der Vorgang des Verschlagwortens wird in der Regel von der Person durchgeführt, die die Information konsumieren will. KLESKE hat erkannt, dass dieses Prinzip eine radikale Bewegung weg vom bisher üblichen Weg der durch die Entwickler vorgegebenen Taxonomie hin zu einer komplett durch die einzelnen Benutzer bestimmten Taxonomie ist.<sup>260</sup>

Tagging ist im Wissens- und Dokumentenmanagement in der Praxis bereits weit verbreitet. Als Beispiel kann die IPEK-Projekterfassung (IPEK-PE) genannt werden. Sie ist ein von ALBERS definiertes und am IPEK - Institut für Produktentwicklung am

---

<sup>254</sup> vgl. Albers, Lohmeyer & Schmalenbach 2010

<sup>255</sup> vgl. Sauter 2011 sowie Held & Cress 2008, S.74f

<sup>256</sup> vgl. Sauter 2011

<sup>257</sup> vgl. Fengel, Rebstock, and Nüttgens 2008

<sup>258</sup> vgl. Mika 2007 sowie Fengel, Rebstock & Nüttgens, S.74f

<sup>259</sup> vgl. Van der Wal 2004

<sup>260</sup> vgl. Sauter 2011



Karlsruher Institut für Technologie (KIT) entwickeltes Dokumenten-, Datei- und E-Mail-Managementsystem, welches auf der freien Verschlagwortung von Inhalten durch die Benutzer basiert.<sup>261</sup> MAUL verwendet Tags, um Informationen im Kontext des Innovationsmanagement auf einer Innovationsplattform zum besseren Auffinden und zum Anreichern von Produktprofilen und Konzepten<sup>262</sup>.

Im Rahmen einer Untersuchung von SAUTER wurde festgestellt, dass das Tagging in der IPEK-PE von den Benutzern eindeutig als hilfreiches Werkzeug empfunden wird, sowohl im Institutsbetrieb als auch in den Projekten.<sup>263</sup> SAUTER hat gezeigt, wie kritisch das vom Benutzer empfundene Verhältnis von Nutzen zu Aufwand für die nachhaltige Akzeptanz ist. Tagging wurde hier von Benutzern immer sehr positiv empfunden.

Die Untersuchungen von SAUTER haben aber auch ergeben, dass etwas weniger als ein Drittel der Benutzer den Eindruck hatte, dass bei der Nutzung der IPEK-PE auch Probleme auftauchten, die typisch für Folksonomien und andere Systeme mit freier Verschlagwortung sind.<sup>264</sup> Durch das freie Vokabular werden eventuell relevante Inhalte nicht gefunden, weil jeder Benutzer eigene Assoziationen zu Inhalten hat und gleiche Begriffe mit unterschiedlichen Bedeutungen belegt werden können.

Um mit diesem Nachteil umzugehen werden sogenannte Tag Recommender eingesetzt. Diese Softwaresysteme sollen die Benutzer bei der Vergabe von Tags durch Vorschläge unterstützen. Einfache Tag Recommender arbeiten in der Regel so, dass die bereits von anderen Benutzern für das gleiche Objekt vergebenen Schlagworte vorgeschlagen werden. Weitere Ansätze nutzen eine Analyse des Textes der Objekte mittels statistischer und computer-linguistischer Verfahren, um Tags vorzuschlagen. Darüber hinaus gibt es Ontologie-gestützte Tag-Recommender, welche Vorschläge auf Basis einer von Experten erstellten Ontologie oder eines Thesaurus generieren.<sup>265</sup> Tagging-Systeme mit Ontologie-gestütztem Tag-Recommender sind Gegenstand aktueller Forschungs- und Entwicklungsarbeiten.<sup>266</sup>

### 2.3.2 Das Ressource Description Framework (RDF)

Das Ressource Description Framework (RDF) ist eine vom World Wide Web Konsortium (W3C) entwickelte Sprache zur formalen Beschreibung von Informationen über Objekte, sogenannten Ressourcen, die durch eindeutige

---

<sup>261</sup> vgl. Albers et al. 2009

<sup>262</sup> vgl. Maul 2015

<sup>263</sup> vgl. Sauter 2011

<sup>264</sup> vgl. Sauter 2011

<sup>265</sup> vgl. Blumauer & Hochmeister 2009

<sup>266</sup> vgl. u.a. Sauter 2011 sowie Albers, Braun & Schmalenbach 2012

Bezeichner (URIs) identifiziert werden. RDF wurde ursprünglich als Datenmodell für Metadaten entwickelt und findet heute breite Anwendung im Semantic Web.

DENGEL<sup>267</sup> formuliert die Herausforderungen, denen mit RDF begegnet werden kann wie folgt:

- Wissen liegt verteilt vor. Es sind geeignete Identifizierungs- und Zugriffsmechanismen notwendig.
- Wissen ist subjektiv. Aussagen müssen diskutierbar sein.
- Es gibt viele unterschiedliche Interessengruppen. Die Repräsentationsformen für Wissen müssen demnach entsprechend erweiterbar und einfach zu benutzen sein.

RDF hat Eigenschaften, die Daten zu strukturieren, selbst wenn sich die zugrunde liegenden Schemata der Daten unterscheiden. Es unterstützt insbesondere die Entwicklung des Schemas mit der Zeit, ohne dass alle Daten der Verbraucher verändert werden.<sup>268</sup> Die Verknüpfungs-Struktur bildet einen gerichteten, markierten Graphen, wo die Kanten eine benannte Verbindung zwischen zwei Ressourcen (Knoten) darstellt. Aussagen über Ressourcen werden in Form von Tripeln, sogenannten RDF-Statements, definiert.<sup>269</sup> Ein RDF-Statement besteht immer aus einem Subjekt, Prädikat und Objekt. HAFFNER<sup>270</sup> beschreibt diese wie folgt:

- Subjekt – die Ressource über die eine Aussage getroffen wird
- Prädikat – bezeichnet die Eigenschaft des Subjekts
- Objekt – das Argument des Prädikats (Property Value)

Die Menge der RDF-Statements bildet einen gerichteten Graphen (vgl. Abbildung 34).



Abbildung 34: Darstellung eines RDF-Tripels als gerichteter Graph<sup>271</sup>

RDF stellt ein Datenmodell zur Verfügung, welches einfach zu erweitern ist.<sup>272</sup> Das Subjekt, Prädikat und Objekt wird in der Regel durch sogenannten Uniform

<sup>267</sup> vgl. Dengel 2012 S.109

<sup>268</sup> <https://www.w3.org/RDF/>, Zuletzt aufgerufen am 18.03.2016

<sup>269</sup> vgl. Cyganiak, Wood & Lanthaler 2014 sowie Haffner 2012

<sup>270</sup> vgl. Haffner 2012

<sup>271</sup> vgl. Cyganiak, Wood, & Lanthaler 2014

<sup>272</sup> vgl. Hitzler et al. 2008 S.35f sowie Dengel 2012 S.109f

Ressource Identifier (URI) eindeutig identifiziert. Das Objekt kann sowohl eine Ressource als auch ein sogenanntes RDF-Literal sein, wobei Letzteres einfache Werte beschreibt (z.B. eine Ganzzahl).<sup>273</sup> Zu den Ressourcen wird alles gezählt, was eine eindeutige Identität besitzt. Dies können Bücher, Dokumente, Orte, oder auch Menschen sein.<sup>274</sup>

Mit RDF ist es möglich, Informationen aus einer Vielzahl an Quellen miteinander zu verknüpfen. Somit können mehrere Graphen zu einem großen zusammengefasst werden. Aus diesem ist es möglich, neue Informationen zu extrahieren.

An dieser Stelle werden auf Grundlage von HITZLER<sup>275</sup>, DENGEL<sup>276</sup>, GANDON & SCHREIBER<sup>277</sup>, BECKETT ET AL.<sup>278</sup>, BERNERS-LEE<sup>279</sup> sowie BERNERS-LEE ET AL.<sup>280</sup> die wichtigsten Aspekte der etablierten Syntax knapp aufgezeigt. Für detaillierte Beschreibungen sei auf die hierfür verwendeten Quellen verwiesen.

Frame Logic (F-Logic) ist ebenfalls eine formale Sprache zur Wissensrepräsentation, welche RDF in vielen Bereichen stark ähnelt. RDF gilt jedoch als mächtiger und weiter verbreitet weshalb hier der Fokus auf dieser Betrachtung gelegt wird.

Neben der Darstellung als Graph lässt sich RDF auch in serialisierter Form durch Zeichenketten darstellen, welche maschinenlesbar sind. Hierfür gibt es drei Möglichkeiten:

- **RDF/XML**<sup>281</sup>

Die Darstellung von RDF mittels der eXtensible Markup Language (XML)<sup>282</sup> gehört zur gebräuchlichsten Methode. Mit Hilfe von XML werden Dokumente anhand einer formalen Struktur beschrieben. XML ist ein weltweit standardisiertes Format und erlaubt die Definition von Daten in einer hierarchischen, natürlich-sprachlichen Struktur (vgl. Abbildung 35).<sup>283</sup>

---

<sup>273</sup> vgl. Hitzler et al. 2008 S.37ff, Dengel 2012 S.118ff, Cyganiak, Wood & Lanthaler 2014

<sup>274</sup> vgl. Hitzler et al. 2008 S.38

<sup>275</sup> vgl. Hitzler et al. 2008 S.39ff

<sup>276</sup> vgl. Dengel 2012 S.114ff

<sup>277</sup> vgl. Gandon & Schreiber 2014

<sup>278</sup> vgl. Beckett et al. 2014

<sup>279</sup> vgl. Berners-Lee 2005

<sup>280</sup> vgl. Berners-Lee et al. 2007

<sup>281</sup> <https://www.w3.org/RDF/>, Letzter Zugriff am 13.11.2016

<sup>282</sup> <http://www.w3schools.com/xml/>, Letzter Zugriff am 13.11.2016

<sup>283</sup> vgl. Dengel 2012

```

1: <?xml version="1.0"?>
2: <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3:   xmlns:dc="http://purl.org/dc/elements/1.1/">
4:   <rdf:Description rdf:about="http://www.w3.org/">
5:     <dc:title>World Wide Web Consortium</dc:title>
6:   </rdf:Description>
7: </rdf:RDF>

```

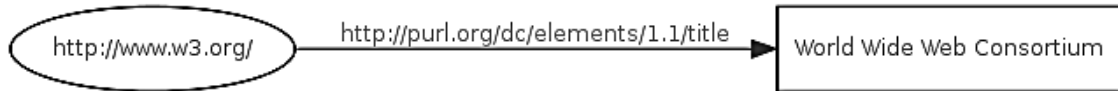


Abbildung 35: Beispiel Darstellung von RDF mit XML und zugehöriger RDF Graph

- **N3**<sup>284</sup>

Die Notation 3 (N3) ist eine formale Sprache mit dem Ziel, die Lesbarkeit für Menschen gegenüber RDF/XML zu erhöhen. N3 geht aber über die Anwendbarkeit von RDF hinaus und beinhaltet komplexere Elemente, wie z.B. Regeln und Pfade. Beispiel:

```

@prefix dc: <http://purl.org/dc/elements/1.1/>.
{ [ dc:creator "Ora" ] dc:wrote [ dc:title "Moby Dick" ] }

```

- **N-Triple und Turtle**<sup>285</sup>

Turtle gilt als Erweiterung des minimalen N-Triple. Die Ausdrucksmächtigkeit von Turtle beschränkt sich auf RDF-Graphen. Damit wurde die Komplexität weiter gesenkt und die Lesbarkeit für Menschen beibehalten. Beispiel:

```

<http://www.w3.org/> <http://purl.org/dc/elements/1.1/title> „World Wide Web Consortium“

```

### RDF-Schema (RDFS)

Mit RDF Schema (RDFS) lässt sich die einfache Datenbeschreibung von RDF um die Möglichkeit, Datensätze auch mit einfachem, terminologischem Wissen bzw. Schemawissen zu versehen, erweitern.<sup>286</sup> Damit können Typen als auch Klassen von Ressourcen entwickelt werden, sowie Properties bestimmter Ressourcenklassen

<sup>284</sup> <https://www.w3.org/TeamSubmission/n3>, Letzter Zugriff am 13.11.2016

<sup>285</sup> <https://www.w3.org/TR/n-triples/> und <https://www.w3.org/TR/turtle/>, Letzter Zugriff am 13.11.2016

<sup>286</sup> vgl. Dengel 2012, S.124

eingeschränkt werden.<sup>287</sup> Die Ontologie wird über die Definition von Regeln in Form von Syntax und Semantik definiert. Damit kann festgelegt werden, welche Properties (Prädikate) zur Charakterisierung von Ressourcen (Subjekten) zulässig sind. Außerdem können Wertebereiche, also Property-Values, bestimmt werden.<sup>288</sup>

Einige markante Beispiele des zur Verfügung stehenden Vokabulars sind:

*rdf: type*

Spezifizierung einer Ressource als Instanz einer Klasse. Die Ressource wird somit als Teil dieser Klasse kategorisiert und besitzt folglich deren Eigenschaften.<sup>289</sup>

Subjekt	Prädikat	Objekt
ex: Max	rdf: type	ex: Person

*rdfs: subclassOf, rdfs: subPropertyOf*

Dies beschreibt die taxonomische Beziehung zwischen den Klassen. Eine Unterklasse übernimmt die Eigenschaften der übergeordneten. Damit ermöglicht dieses Property transitiv Schlüsse.

ex: Person	rdfs: subclassOf	ex: Säugetier
ex: friendOf	rdfs: subPropertyOf	ex: knows

Property Restriktionen mittels *domain* und *range*

Domain und Ränge sind selbst Properties und dienen der Verknüpfung von Properties mit Klassen. Sie schränken eine Klasse hinsichtlich der mit dieser Klasse verknüpften Properties ein

ex: author	rdfs: domain	ex: Blog
ex: author	rdfs: range	ex: Person

*RDF-Container und Blank Nodes*

RDF-Container dienen dazu, geordnete Listen (rdf: Seq), ungeordnete Mengen (rdf: bag) oder alternative Möglichkeiten (rdf: Alt) in RDF zu realisieren. Ein Blank Node dient als Hilfsknoten und besitzt keine URI sondern eine rdf:nodeID (hier beispielhaft id1 und wird als `_:id1` für die Serialisierung abgekürzt).

<sup>287</sup> vgl. Dengel 2012, S.124f

<sup>288</sup> vgl. Haffner 2012

<sup>289</sup> „ex:“ dient hier als verkürzte Schreibweise für eine Präfix URI.

http://.../BlogKim	ex:author	_:id1
_:id1	rdf: type	ex: Person
_:id1	ex: Person	„Kim“

*Literale*

Wie bereits erwähnt, werden URIs verwendet, um Ressourcen zu referenzieren. Wird jedoch mit konkreten Datenwerten gearbeitet, werden diese durch Literale dargestellt. Hierbei handelt es sich um „reservierte Bezeichner für RDF-Ressourcen eines bestimmten Datentyps“<sup>290</sup>.

http://.../BlogKim	dc: title	„Kim’s Blog“
--------------------	-----------	--------------

Neben RDFS existieren weitere Ontologie-Beschreibungssprachen wie die Web Ontology Language (OWL). OWL basiert teilweise auf Beschreibungslogik und wurde durch mengentheoretische Konstrukte angereichert. Im Gegensatz zu RDFS ermöglicht die Web Ontology Language (OWL) die Umsetzung einer vollumfänglichen Prädikatenlogik. OWL besitzt somit eine höhere Mächtigkeit bei der Umsetzung einer Semantik.<sup>291</sup>

Abramovici et al.<sup>292</sup> nutzen OWL zur Beschreibung von semantischen Metadaten im Produktlebenszyklusmanagement.

**Querying / Reasoning**

Die Simple Protocol and RDF Query Language (SPARQL) ist eine Abfragesprache für RDF zur Extraktion von Wissen aus Metainformationen in einem RDF Graph. Beispiel: Folgende Anfrage listet die Namen aller afrikanischen Hauptstädte und das Land, in dem sich die jeweilige Hauptstadt befindet. Variablen werden in SPARQL mit vorangestelltem „?“ gekennzeichnet.

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
  abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
  abc:isInContinent abc:Africa .
}
```

<sup>290</sup> vgl. Hitzler et al. 2008

<sup>291</sup> vgl. Haffner 2012

<sup>292</sup> vgl. Abramovici et al. 2016

Ein semantischer Reasoner ist ein Stück Software das in der Lage ist, logische Konsequenzen aus einer Reihe von geltend gemachten Tatsachen oder Axiomen herzuleiten. Damit lässt sich aus abgeleiteten Bedingungen neues Wissen ermitteln.

Bei Reasonern kommen in Abhängigkeit der notwendigen Expressivität einer Ontologie vor allem Untermengen der Prädikatenlogik (z.B. regelbasierte Reasoner auf Basis Horn Logik) oder Beschreibungslogiken (im Folgenden auch „Description Logic“ oder kurz DL) zum Einsatz. Description Logics sind eine Familie von Sprachen zur Wissensrepräsentation, die zur Beschreibung von Zusammenhängen zwischen Konzepten einer Domäne genutzt werden können.

Das folgende Beispiel eines regelbasierten RDF Reasoners definiert die Regel: wenn ein Entwicklungsartefakt ein System beschreibt und ein Stakeholder verantwortlich für ein System ist, dann ist der Stakeholder auch interessiert am Entwicklungsartefakt.

```
@prefix po: <http://porsche/rdf/>.
@include <RDFS>.

[rule1: (?artefact po:beschreibtInhaltVon ?system) (?system
po:verantwortlich ?stakeholder) -> (?artefact po:interessiert
?stakeholder)]
```

### Verwendung von Ontologien in der Produktentwicklung

KIM<sup>293</sup> und LIANG<sup>294</sup> verwenden Ontologien um Produktmodelle mit semantischen Beschreibungen zu erweitern, um die Zusammensetzung von Systemen und deren Begründungen zu dokumentieren. PATIL, DUTTA UND SRIRAM<sup>295</sup> haben ein plattformunabhängiges, semantisches Austauschformat für Produktdaten entwickelt, welches die bestehenden Lösungen (z.B. STEP – Standard for Exchange of Product Model Data) um semantische Bedeutungen erweitert. STIEFEL ET AL. haben ein Konzept entwickelt, wie sich verteilte und dezentrale Produktentwicklungstätigkeiten mit RDF formal beschreiben lassen.<sup>296</sup> ALBERS, BRAUN UND SCHMALENBACH<sup>297</sup> haben eine formale Ontologie des iPeM auf Basis F-Logic entwickelt.

---

<sup>293</sup> vgl. Kim 2006

<sup>294</sup> vgl. Liang & Paredis 2003

<sup>295</sup> vgl. Patil, Dutta & Sriram 2005

<sup>296</sup> vgl. Stiefel, Hausknecht & Müller 2010

<sup>297</sup> vgl. Albers, Braun & Schmalenbach 2012

Tabelle 3: Ontologie von ALBERS, BRAUN UND SCHMALENBACH<sup>298</sup>

Concepts	Relations	Attributes
iPem_BasicElement	ActivityhasInputObject	ActivityBestpracticeInformation
Activity	ActivityhasInputObjective	ActivityDateEnd
GenericActivity	ActivityhasOutputObject	ActivitydateStart
MacroActivity	ActivityhasOutputObjective	ActivitySuitableMethod
MicroActivity	ActivityhasprecedentActivity	DigitalDocumentWikipediaLink
Object	ActivityhasResources	DigitalModelPDMLink
DigitalDocument	ActivityisofTypeMacroactivity	DigitalDocumentFileSystemLink
DigitalModel	ActivityisofTypeMicroActivity	iPem_BasicElementDescription
PhysicalDocument	ActivityisofTypeSubMicroActivity	MicroActivityAbbreviation
PhysicalModel	xActivityhasconsecutiveActivity	ObjectiveContent
Objective	xObjectiscreatedbyRessource	xActivityDuration
Resource	xObjectisinfluencedbyObjective	
	xObjectisInputofActivity	
	xObjectisOutputofActivity	
	xObjectiveInfluencesObject	
	xObjectiveisInputofActivity	
	xObjectiveisOutputofActivity	
	xObjectiverequiresResource	
	xResourcecontributesoActivity	
	xResourcecontributesoObject	
	xResourcecontributesoObjective	

### 2.3.3 Zwischenfazit

Metadaten können dazu verwendet werden, Aussagen über ein bestimmtes Objekt oder einen Datensatz zu treffen. Semantische Metadaten sind auf Grund Ihrer formalen Definition maschinenlesbar (und maschinenverstehbar). In den Computerwissenschaften wird Semantik daher häufig aufgegriffen mit dem Ziel, das Wissensmanagement zu unterstützen. Mit Ihnen können komplexe Informationszusammenhänge beschrieben und darüber hinaus auch Rückschlüsse auf weitere vorhandene Informationszusammenhänge in Modellen, die bisher nicht explizit modelliert wurden, gezogen werden. Mit semantischen Metadaten kann man Dokumente verlinken, auch wenn sich deren innere Struktur und Form unterscheidet.

Mit Tags kann man einem Nutzer schnell das Wichtigste zum Inhalt von Objekten und deren Relationen vermitteln. Eine formale oder auch logische Semantik ist jedoch erforderlich, um Inkompatibilitäten und Inkonsistenzen zu vermeiden und somit korrektes Schlussfolgern zu ermöglichen. Durch das freie Vokabular werden daher eventuell relevante Inhalte nicht gefunden, weil jeder Nutzer eigene Assoziationen zu Inhalten hat und gleiche Begriffe mit unterschiedlichen Bedeutungen belegt werden können. Um mit diesem Nachteil umzugehen können

<sup>298</sup> vgl. Albers, Braun & Schmalenbach 2012



Tag Recommender eingesetzt werden, welche zwar dem Nutzer Vorschläge für das Tagging bereitstellen, die Nachteile jedoch auch nicht vollständig eliminieren.

Das Resource Description Framework gilt als grundlegender Baustein des Semantischen Webs. Mit ihm lässt sich eine Verknüpfungs-Struktur in Form eines gerichteten, markierten Graphen bilden, wobei die Kanten eine benannte Verbindung zwischen zwei Ressourcen (Knoten) darstellen. RDF stellt dabei ein Datenmodell zur Verfügung, welches einfach zu erweitern ist. Mit RDF Schema lässt sich die einfache Datenbeschreibung von RDF um die Möglichkeit, Datensätze und Relationen auch mit einfachem, terminologischem Wissen bzw. Schemawissen zu versehen, erweitern. Aus diesem Grund wird erwartet, dass sich RDF besonders zur Beschreibung von heterogenen Entwicklungsartefakten und deren Relationen eignet. Bestehende Ontologien bieten bereits grundlegende formalisierte Repräsentationen in der Produktentwicklung. Jedoch fehlen noch wichtige Elemente zur Beschreibung von Entwicklungsartefakten und deren Relationen, insbesondere zur Herstellung von Durchgängigkeit und Konsistenz im Zielsystem.

## **2.4 Systems-of-Systems**

Der Begriff "Systems-of-Systems" (SoS) findet bereits seit den 1950er Jahren Verwendung. In der Literatur gibt es eine Vielzahl von Definitionen und Charakterisierungen. BOULDING<sup>299</sup> verwendet den Begriff in seiner Arbeit über die allgemeine Systemtheorie um die Organisation eines theoretischen Gebildes zu beschreiben. Er beschreibt ein SoS als die Anordnung von theoretischen Systemen und Gebilden in einer hierarchischen Komplexität.

Der United States Strategic Defense Initiative (SDI) wird eine wesentliche Rolle bei der Etablierung von SoS als Entwicklungskonzept in den späten 1980ern zugesprochen. Sie beschreiben ein SoS als Zusammenschluss von unabhängig voneinander agierenden Systemen. Später folgen zunehmende Aktivitäten in Forschung und Industrie und das Feld bekam zunehmend Interesse. 15 Jahre später etablierte sich SoS Engineering als eine angesehene Disziplin<sup>300</sup>.

### **2.4.1 Definition und Eigenschaften**

Als Antwort auf die immer größere Verbreitung des Begriffs SoS charakterisierte MAIER SoS mit 5 grundlegenden Eigenschaften, auf welche heute in zahlreichen Arbeiten referenziert wird. Nach MAIER<sup>301</sup> ist ein System-of-Systems eine

---

<sup>299</sup> vgl. Boulding 1956

<sup>300</sup> vgl. Keating et al. 2003

<sup>301</sup> vgl. Maier 1998

Ansammlung von Komponentensystemen, welche selbst als Systeme gesehen werden können, jedoch zusätzliche Eigenschaften besitzen:

**Operative Unabhängigkeit der Elemente:** Die Komponentensysteme, aus denen ein System-of-Systems besteht sind ihrerseits vollständig autark nutzbare Systeme. Sie können somit auch einen System-of-Systems-Verbund verlassen und dennoch sinnvolle Arbeit verrichten.

**Unabhängiges Management der Elemente:** Die Komponentensysteme eines SoS werden in der Regel separat entwickelt und beschafft. Ihr Lebenszyklus ist unabhängig von dem Lebenszyklus des System-of-Systems.

**Geographische Verteilung:** Die geografische Ausdehnung und Verteilung der Komponentensysteme eines System-of-Systems ist häufig groß. Bezogen auf das SoS bedeutet das, dass seine Teilsysteme auf Grund ihrer räumlichen Distanz zueinander im Wesentlichen nur Informationen (Daten) und nicht erhebliche Mengen an Masse, Stoffen oder Energie austauschen können.

**Evolutionäre Entwicklung:** Die Entwicklung eines System-of-Systems ist niemals abgeschlossen. Seine Entwicklung und Existenz ist evolutionär, das heißt über den gesamten Lebenszyklus können Komponentensysteme hinzugefügt, entfernt oder geändert werden.

**Emergentes Verhalten:** Emergenz (vom lateinischen *emergere* für „das Auftauchen“, „das Herauskommen“ oder „das Emporsteigen“) ist die spontane Herausbildung von neuen Eigenschaften oder Funktionen eines Systems infolge des Zusammenspiels seiner Elemente. Übertragen auf ein System-of-Systems bedeutet das, dass durch den Zusammenschluss der Teilsysteme bestimmte Eigenschaften und Dienstleistungen für den Nutzer herausgebildet werden, die nicht mehr eindeutig bestimmten Teilsystemen zugeordnet werden können. Nach KRYGIEL<sup>302</sup> sind bei einem SoS Systeme in einer Weise verwandt oder verbunden, sodass Funktionen erzielt werden, welche von den einzelnen Systemen nicht alleine erfüllt werden können.

ROTH<sup>303</sup> merkt an, das nicht jedes System-of-Systems immer alle der vorgenannten Merkmale in gleicher Weise hat. Bei einigen SoS ist beispielsweise die geographische Verteilung eher gering oder nicht vorhanden.

In der Literatur gibt es eine Vielzahl weiterer unterschiedlicher Definitionen eines SoS, wobei sich diese in vielen Eigenschaften, insbesondere bei der Unabhängigkeit

---

<sup>302</sup> vgl. Krygiel 1999

<sup>303</sup> vgl. Roth 2015

und bei der Erfüllung übergeordneter Funktionen durch Zusammenschluss der Systeme gleichen. JAMSHIDI<sup>304</sup> definiert ein SoS als die Integration einer bestimmten Anzahl von Komponentensystemen (engl. Constituent Systems), welche unabhängig voneinander und trotzdem für einen gewissen Zeitraum vernetzt sind, um ein höheres Ziel zu erreichen. Die INCOSE<sup>305</sup> definiert ein SoS als Systeme, deren Systemelemente wiederum selbst Systeme sind, welche typischerweise umfangreiche, interdisziplinäre Aufgaben mit vielen heterogenen und verteilten Systemen erfüllen. Diese integrierende Menge von Systemelementen erzeugt eine Funktionalität, die ein einzelnes System niemals erreichen könnte. In der internationalen Norm ISO/IEC 15288<sup>306</sup> der IEEE über den Lebenszyklus von Systemen wird ein SoS als System-of-Interest beschrieben, wessen Elemente selbst auch wieder als Systeme angesehen werden können. Ein SoS bringt eine Menge von Systemen zusammen, um eine Aufgabe zu erfüllen, welche keines der beteiligten Systeme alleine erfüllen könnte. Jedes Komponentensystem wird dabei selbstständig verwaltet, hat eigene Ziele und Ressourcen. Diese müssen jedoch angepasst werden, um die Ziele des SoS zu erfüllen. ABBOTT<sup>307</sup> argumentiert, dass der SoS Begriff für Systeme reserviert werden muss, welche qualitativ und strukturell unterschiedlich zu traditionellen Systemen sind. Ein SoS soll nicht als Hierarchie von Komponenten gesehen werden, sondern als eine Umgebung, der Systeme beitreten, in ihr operieren und interagieren können. LUZEAUX<sup>308</sup> zieht das Ziel der Maximierung der Wertschöpfungskette mit heran. Nach ihm ist ein System-of-Systems ein Zusammenschluss von Systemen, welche unabhängig voneinander erworben und verwendet werden können und dessen globale Wertschöpfungskette der Entwickler, Käufer und/oder Nutzer versucht, zu einem gewissen Zeitpunkt über eine gewisse Menge von denkbaren Zusammenschlüssen zu maximieren.

Das Department of Defence (DoD) hat ein Handbuch für die Entwicklung von Systems-of-Systems veröffentlicht<sup>309</sup>. Darin wird ein SoS als eine Menge oder Anordnung von Systemen beschrieben, die daraus resultiert, dass unabhängige und eigenständig nützliche Systeme zu einem größeren System integriert werden, welches einzigartige Fähigkeiten bereitstellt.

---

<sup>304</sup> vgl. Jamshidi 2009

<sup>305</sup> vgl. GfSE 2012

<sup>306</sup> vgl. ISO/IEC 15288

<sup>307</sup> vgl. Abott 2006

<sup>308</sup> vgl. Luzeaux & Ruault 2010

<sup>309</sup> vgl. Systems Engineering Guide for Systems of Systems des US Verteidigungsministeriums 2008

## 2.4.2 Typen und Beispiele von Systems-of-Systems

SoS existieren in unterschiedlichen Größen und Ausprägungen. Nach MAIER<sup>310</sup> gibt es drei unterschiedliche Typen von Systems-of-Systems, welche sich im Wesentlichen über die Beziehung der Organisationseinheiten, welchen die Komponentensysteme und das SoS zugehörig sind, unterscheiden lassen. DAHMANN ET. AL.<sup>311</sup> erweitern die von Maier identifizierten Typen noch um einen weiteren Typ. Diese vier Typen und Ihre Charakteristiken sind in Abbildung 36 dargestellt und werden im Folgenden erläutert. Tatsächlich sind die meisten SoS eine Kombination von diesen Typen. Eine Klassifikation hilft jedoch dabei, die unterschiedlichen Arten von SoS genauer zu verstehen.<sup>312</sup>

**Gelenkt (engl. directed):** Die Komponentensysteme werden zentral von einer Organisationseinheit oder einem Unternehmen verwaltet, jedoch sind die Komponentensysteme selbst fähig, unabhängig voneinander zu operieren. Der primäre Zweck der Systeme ist die Erfüllung der SoS-Ziele und ist diesen untergeordnet.<sup>313</sup>

*Beispiel „Luftverteidigungssystem“:* Ein integriertes Luftverteidigungssystem besteht aus einer Vielzahl von Systemen, welche zwar auch unabhängig vom SoS operieren können, jedoch alle unter direkter Kontrolle des Militärs stehen.

**Kollaborativ (engl. collaborative):** Kollaborative Systems-of-Systems unterscheiden sich von einem gelenkten SoS dadurch, dass die zentrale SoS Organisation nicht zwingend auch die Befugnis hat, die Systeme zu betreiben. Das Komponentensystem muss mehr oder weniger freiwillig kollaborieren, um die Ziele des SoS zu erfüllen.

*Beispiel „Internet“:* Das IETF<sup>314</sup> erarbeitet Standards für das Internet, hat aber keine Macht, diese bei den beteiligten Akteuren im Internet durchzusetzen. Akteure nehmen am Internet freiwillig teil.

---

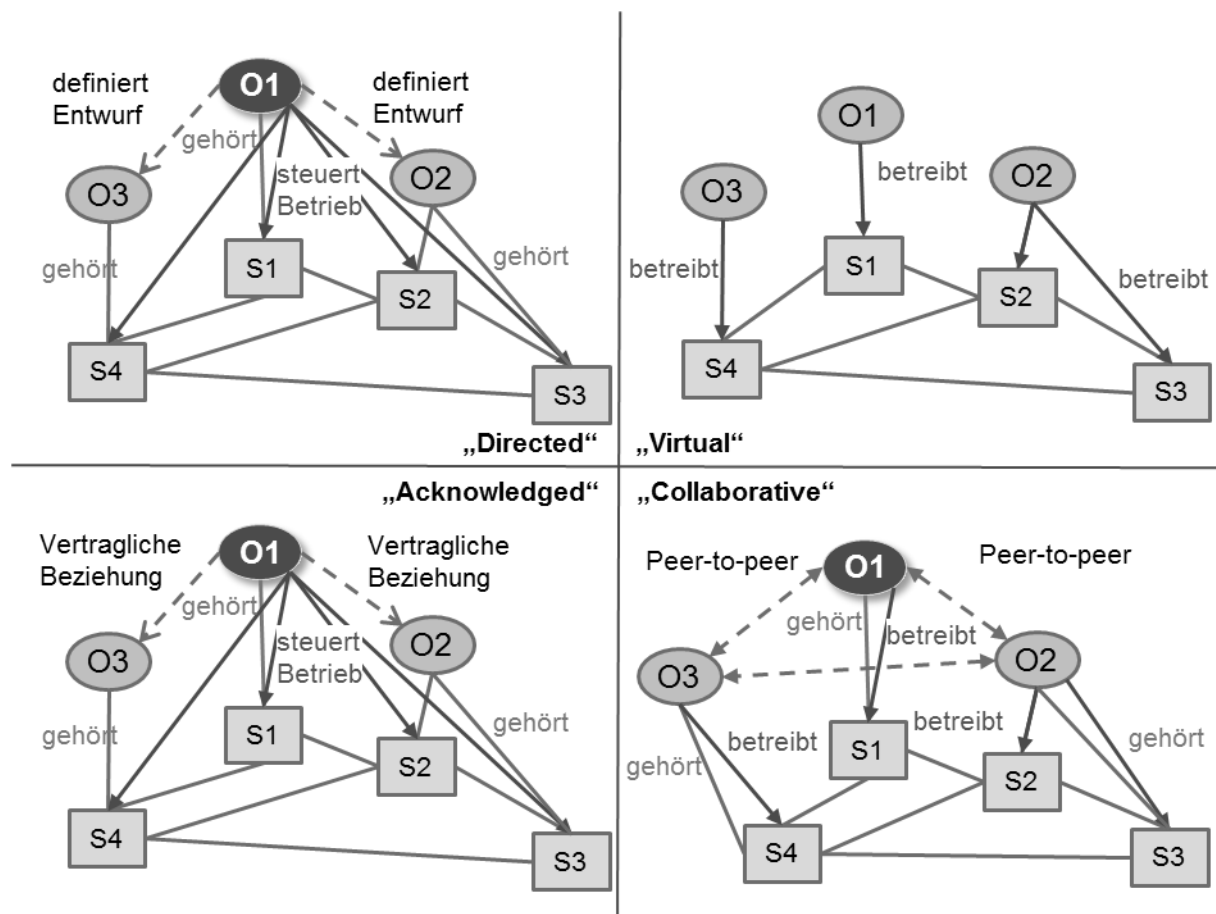
<sup>310</sup> vgl. Maier 1998

<sup>311</sup> vgl. Dahmann and Baldwin 2008, S. 2

<sup>312</sup> vgl. Dahmann 2015

<sup>313</sup> vgl. Maier 1998, S. 278

<sup>314</sup> <https://www.ietf.org/>, letzter Zugriff am 17.12.2016

Abbildung 36: Typen von Systems-of-Systems<sup>315</sup>

**Virtuell (engl. virtual):** Virtuelle SoS haben weder eine zentrale Organisationseinheit, welche das SoS verwaltet, noch gemeinsam verfolgte Ziele. Ein virtuelles SoS kann absichtlich oder zufällig entstehen.

*Beispiel „World Wide Web“:* Ersteller von Webseiten wählen selbst, ob sie den Standards folgen oder nicht. Die Standards entwickeln sie nicht auf kontrollierte Art und Weise, sondern gehen vom Markterfolg oder von verschiedenen Innovatoren aus.

**Anerkannt (engl. acknowledged):** Bei diesem Typ gibt es von allen beteiligten Organisationseinheiten anerkannte Ziele, eine zentrale Verwaltungsinstanz und dedizierte Ressourcen für das SoS. Die Komponentensysteme behalten jedoch ihren unabhängigen Besitz, eigene Ziele und Entwicklungsressourcen. Änderungen der Systeme basieren auf Vereinbarungen zwischen der Organisationseinheit des SoS und der Komponentensysteme.

<sup>315</sup> vgl. Henshaw 2013, S. 6

*Beispiel „Connected Car“:* Das vernetzte Fahrzeug (engl. Connected Car) ist ein Fahrzeug, welches mit einem oder mehreren Steuergeräten ausgestattet ist, um sich drahtlos mit anderen Systemen außerhalb des Fahrzeugs zu verbinden (z.B. andere Fahrzeuge, Verkehrsinfrastruktur, Server im Internet). Der Fahrzeughersteller bildet hier häufig die zentrale Verwaltungsinstanz. Komponentensysteme (z.B. das Fahrzeug selbst oder Verkehrsinfrastruktur) können jedoch auch unabhängig vom Systemverbund sinnvoll operieren. Zwischen Fahrzeughersteller und den Betreibern anderer Systeme werden in der Regel vertragliche Vereinbarungen geschlossen.

### 2.4.3 Systems-of-Systems Engineering

Obwohl SoS in sehr unterschiedlichen Domänen entstehen und ihre Gestalt und Anwendung sich unterscheidet, gibt es Ähnlichkeiten bei den Herausforderungen, die bei deren Entwicklung vorhanden sind<sup>316</sup>. Aus den SoS-spezifischen Eigenschaften ergeben sich neue Herausforderungen in der Entwicklung. Das System-of-Systems Engineering (SoSE) ist eine Menge von Entwicklungsprozessen, Werkzeugen und Methoden für die Konzeption und Entwicklung von Lösungen für den Umgang mit Systems-of-Systems Herausforderungen<sup>317</sup>. SoSE befasst sich dabei mit der Planung, Analyse, Organisation und Integration der Funktionen von alten und neuen Systemen in einer SoS-Funktion, die größer ist als die Summe der Funktionen der Komponentensysteme<sup>318</sup>. ALBERS ET. AL<sup>319</sup> fassen die Herausforderungen bei der Entwicklung von SoS wie folgt zusammen:

**Keine zentrale Kontrolle:** Jedes Komponentensystem wird in der Regel in einer eigenen Projektorganisation, häufig sogar in unterschiedlichen Unternehmen unabhängig voneinander entwickelt und betrieben. Eine alles übergreifende, koordinierende Organisation für das System-of-Systems ist nicht immer gegeben. Eine Vielzahl von Stakeholdern ist involviert.

**Unterschiedlicher Produktlebenszyklus:** Die Komponentensysteme befinden sich in unterschiedlichen Phasen ihres Produktlebenszyklus. Einige der Komponentensysteme sind möglicherweise noch im Entwicklungsstadium während andere bereits kurz vor dem Abbau stehen. Die Entwicklung der einzelnen Komponentensysteme läuft nicht zwangsläufig synchron und koordiniert ab.

**Häufige Änderungen:** Die Entwicklung des SoS ist niemals beendet. Es werden ständig neue Systeme zum SoS hinzugefügt oder entfernt. Änderungen an

---

<sup>316</sup> vgl. Nielsen et al. 2015

<sup>317</sup> vgl. Systems Engineering Guide for Systems of Systems des US Verteidigungsministeriums 2008

<sup>318</sup> vgl. Systems Engineering Guide for Systems of Systems des US Verteidigungsministeriums 2008

<sup>319</sup> vgl. Albers, Kurrle & Klingler 2016

Komponentensystemen treten häufig auf und können andere Komponentensysteme betreffen.

**Interessenkonflikte:** SoS bestehen aus vielen Komponentensystemen, welche auch unabhängig vom SoS sinnvolle Aufgaben durchführen können. Jedes Komponentensystem hat entsprechend auch eigene Ziele. Die beteiligten Stakeholder können wenig Interesse daran haben, die übergeordneten Ziele des SoS mit ihren Systemen umzusetzen und können diese sogar ablehnen. Ziele der Komponentensysteme können mit den Zielen des SoS in Konflikt stehen.

**Komplexität:** Kein Akteur überblickt alles vollständig. Die Komponentensysteme weisen einen hohen Grad an Heterogenität auf. Komponentensysteme stellen für andere Projektorganisationen eine „Black Box“ dar.

NCUBE betont, dass die Lösung dieser Probleme es notwendig macht, eine Vielzahl von unterschiedlichen Disziplinen wie Maschinenbau, Sozialwissenschaften, Systems Engineering, Informatik, Mathematik, Psychologie und viele weitere zu involvieren<sup>320</sup>. Dies erfordert Methoden und Tools, welche ein kollaboratives Arbeiten von der Anforderungserhebung bis zum Testen und Warten unterstützen<sup>321</sup>.

HALLERSTEDE ET AL.<sup>322</sup> ergänzen, dass viele Herausforderungen bei der Entwicklung von SoS im Allgemeinen vom Systems Engineering behandelt werden, jedoch im Rahmen von SoS eine noch größere Bedeutung erlangen (zum Beispiel „Evolution“).

ALBERS UND LOHMEYER heben hervor, dass insbesondere die Spezifikation der Schnittstellen und die Berücksichtigung des übergeordneten System-Verhaltens zentrale Elemente des Systems-of-Systems Engineering sind<sup>323</sup>.

KEATING ET AL.<sup>324</sup> beschreiben die Unterschiede zwischen Systems Engineering und SoS Engineering wie folgt:

**Fokus:** Üblicherweise fokussiert sich Systems Engineering auf die Entwicklung einzelner, komplexer Systeme. Die Definition, Analyse und Entwicklung ist darauf ausgerichtet, um ein spezifisches Problem oder Ziel mit der Umsetzung oder Anpassung eines Systems zu adressieren. SoSE fokussiert darauf, mehrere komplexe Systeme zu integrieren. Dabei können sowohl neu zu schaffende Systeme als auch bestehende Systeme - oder eine Mischung aus beiden - integriert werden.

---

<sup>320</sup> vgl. Ncube 2011, S. 2

<sup>321</sup> vgl. Nielsen et al. 2015, S. 2

<sup>322</sup> vgl. Hallerstede et al. 2012, S. 573

<sup>323</sup> vgl. Albers & Lohmeyer 2012

<sup>324</sup> vgl. Keating et al. 2003 und Keating, Padilla & Adams 2008

**Zielsetzung:** Bei SoSE muss im Problemlösungsprozess die Entwicklung einer zufriedenstellenden, nicht unbedingt optimalen Problemlösung angestrebt werden. Eine optimale Lösung existiert im Kontext von SoS nach KEATING nicht.

**Ansatz:** Akzeptierte Prozesse für die Ausführung von Systems Engineering gibt es viele. Diese können jedoch zu restriktiv sein. SoSE muss flexibel genug sein, um mit wechselnden Problemstellungen, Kontext und Randbedingungen umzugehen. Die Methoden des Systems Engineering müssen folglich kompatibel mit den sich potentiell schnell ändernden Randbedingungen eines SoS gemacht werden.

**Erwartungshaltung:** Beim Systems Engineering besteht die Erwartungshaltung, dass eine Lösung erzielt wird. SoSE muss mit einer Klasse von komplexen Problemen umgehen. Die Erwartungshaltung ist daher zunächst, dass eine erste Reaktion auf eine Menge von Problemen erzielt wird.

**Problem:** Traditionelle Systems Engineering Ansätze adressieren erfolgreich gut abgegrenzte Probleme. Diese Randbedingungen sind bei schnell wachsenden SoS eher die Ausnahme.

**Analyse:** Beim SoSE steht besonders der Einfluss aus dem Kontext im Vordergrund, weniger die rein technische Sichtweise auf das Problem.

**Ziele:** SoSE Umgebungen und Ansätze müssen für pluralistische, dynamische Systemziele ausgelegt sein.

**Grenzen:** Die Grenzen für SoSE sind willkürlich, durchlässig und wahrscheinlich in ständigem Fluss. Was inbegriffen und bei der Analyse/Synthese ausgeschlossen ist unterliegt einem ständigen Wandel.

Das US Verteidigungsministerium (Department of Defense) hat einen Leitfaden zur Entwicklung von Systems-of-Systems erstellt (Systems Engineering Guide for Systems-of-Systems). Dort werden 7 Kernelemente vorgestellt, welche benötigt werden, um SoS Fähigkeiten zu entwickeln, welche auf dem Systems Engineering basieren:

1. Übersetzen von SoS Zielen in High-Level SoS Anforderungen
2. Verständnis über Komponentensysteme und deren Beziehungen aufbauen
3. Beurteilen, inwieweit das SoS die definierten Ziele über die Zeit erfüllt
4. Entwickeln und Aufrechterhalten einer Architektur für das SoS
5. Überwachung und Bewertung möglicher Auswirkungen von Änderungen auf das SoS.
6. SoS Anforderungen und Lösungsmöglichkeiten an die Verantwortlichen (z.B. Systembesitzer) adressieren
7. Verbesserungen koordiniert in das SoS mit einfließen lassen



Requirements Engineering ist bereits für einzeln betrachtete Systeme eine große Herausforderung. Eine Vielzahl von Techniken zur Unterstützung der Entwicklung existiert. Allerdings bieten diese Techniken auf Grund der besonderen Herausforderungen bei Systems-of-Systems nicht ausreichend Unterstützung.<sup>325</sup>

Der von LEWIS ET AL.<sup>326</sup> vorgestellte SoS Requirements Engineering Ansatz beinhaltet eine Kombination aus top-down und bottom-up Vorgehen. Zum Top-Down Vorgehen gehören das Verstehen des SoS-Kontexts, die Identifikation von SoS-Zielen und Interaktionen. Beim Bottom-Up Vorgehen geht es darum zu verstehen, welche Fähigkeiten potenzielle Komponentensysteme für das SoS mitbringen. Er beschreibt das Vorgehen in 5 Schritten:

1. Identifikation des SoS Kontext
  - a. Identifikation des SoS Typ
  - b. Analyse der SoS und System Umgebung
2. Identifikation der Ziele für das SoS und der individuellen Systeme
  - a. Gemeinsame Ziele der teilnehmenden Systeme identifizieren und Ziele hierarchisch abbilden (Baum)
  - b. In Konflikt stehende Ziele des SoS und der teilnehmenden Systeme identifizieren
3. Verstehen der SoS Interaktionen
  - a. Jede SoS Interaktion sollte auf die Ziele des SoS abgebildet werden um zu erklären, wie SoS Ziele durch SoS Interaktionen erfüllt werden
  - b. Szenario-basierte Techniken und Anwendungsfälle können verwendet werden, um SoS Interaktionen zu extrahieren
4. Identifikation von Randbedingungen im Bezug auf Fähigkeiten der individuellen beteiligten Systeme
5. Analyse der Lücke zwischen Zielen des SoS und möglichen Fähigkeiten der individuellen Systeme

### **Modellbasierte Ansätze vom Forschungsprojekt COMPASS**

Im Rahmen des Forschungsprojektes Comprehensive Modelling for Advanced System-of-Systems (COMPASS) wurde ein modell-basierter Ansatz basierend auf der Modellierungssprache SysML für das Requirements Engineering von Systems-of-Systems vorgestellt.<sup>327</sup> Dieser Ansatz berücksichtigt das Vorgehen von

---

<sup>325</sup> vgl. Lewis et al. 2009

<sup>326</sup> vgl. Lewis et al. 2009

<sup>327</sup> vgl. Holt, Perry & Hansen 2012

LEWIS ET AL., den kontext-basierten Requirements Engineering Ansatz ACRE<sup>328</sup> und den SoS Guide des US Verteidigungsministeriums. Er besteht aus 3 Komponenten:

- Ein Rahmenwerk für ein modellbasiertes Requirements Engineering (MBRE)
- Erweiterung des Rahmenwerks zur Abbildung von SoS Projekten in SysML
- Prozesse zur Anwendungsanleitung

ALBERS ET AL. haben einen Ansatz zur modellbasierten Entwicklung von SoS auf Basis von SysML entwickelt<sup>329</sup>.

Sie gliedern den Entwicklungsprozess in sechs Schritte:

1. Anwendungsfälle des SoS mit SysML Anwendungsfalldiagramm definieren
2. Verfeinerung der Anwendungsfälle und Darstellung der Kommunikationsabläufe durch SysML Sequenzdiagramme
3. Ableitung von Anforderungen an das SoS
4. Definition des System-Kontexts mit allen beteiligten Systemen und Darstellung in SysML Blockdiagrammen. Verknüpfung der Anforderungen (Requirements Diagramm) mit den Blockdiagrammen
5. Konkretisierung des Sequenzdiagramms durch Ergänzung der Informationen über beteiligte Systeme und Schnittstellen
6. Ableitung von Anforderungen an die beteiligten Komponenten (engl. „constituent systems“)

#### 2.4.4 Zwischenfazit

Systems-of-Systems existieren in unterschiedlichen Größen und Ausprägungen und sind bereits heute in den unterschiedlichsten Domänen wie Verteidigungsindustrie, Luft- und Raumfahrt oder Automobilindustrie zu finden. Ihre Entwicklung ist insbesondere dadurch gekennzeichnet, dass die beteiligten Komponentensysteme unabhängig voneinander entwickelt und betrieben werden. Dadurch entstehen neue Herausforderungen bei deren Entwicklung.

Das Connected Car ist bereits heute Teil eines vernetzten SoS. Die Entwicklung von SoS ist bereits in anderen Domänen wie der Verteidigungsindustrie untersucht, es besteht jedoch noch ein Forschungsbedarf zur Untersuchung der spezifischen Herausforderungen in der Entwicklung des Connected Car, welche sich aus den Eigenschaften von SoS ableiten können.

---

<sup>328</sup> vgl. Holt, Perry & Brownsword 2012

<sup>329</sup> vgl. Albers, Kurrle & Moeser 2014

Auf Grund der verteilten und teilweise unabhängigen Entwicklung von Komponentensystemen, hohen Abhängigkeiten und häufigen Änderungen erlangt das Management von Änderungen im Kontext von SoS eine noch größere Bedeutung. Es fehlen noch konkrete Ansätze, welche Entwickler beim durchgängigen Management von Änderungen bei SoS unterstützen.

Basis eines funktionierenden Managements von Änderungen ist eine durchgängige und nachvollziehbare Dokumentation des Zielsystems. Insbesondere bei SoS werden Ziele und Anforderungen sehr heterogen dokumentiert. Es wird vermutet, dass sich eine Homogenisierung der Dokumentationsform (z.B. durch einen modellbasierten Ansatz) über alle Entwicklungsphasen, Produktgenerationen und beteiligten Organisationen in der Praxis nur schwer durchsetzen lässt. Es werden noch Untersuchungen benötigt, welche diese Hypothese bestätigen. Weiter werden Ansätze benötigt, welche Durchgängigkeit und Konsistenz im Zielsystem herstellen, auch wenn sich die zugrundeliegende Dokumentationsform unterscheidet.

### 3 Motivation und Zielsetzung

In diesem Kapitel wird zunächst die Motivation der Arbeit dargestellt und anschließend die Zielsetzung abgeleitet.

#### 3.1 Motivation

Im Rahmen einer Tagung der Wissenschaftlichen Gesellschaft für Produktentwicklung (WiGeP) im Jahr 2015 bewerteten Vertreter aus Industrie und Forschung Thesen zur zukünftigen Produktentwicklung. Dabei war sich der Großteil der Befragten einig, dass Produkte zunehmend „smart“ werden.<sup>330</sup> Smarte Produkte vernetzen sich mit ihren smarten Nutzungsumgebungen (z.B. Smart Buildings, Smart Grids oder der Smart Factory).<sup>331</sup> 22 von 25 Vertretern gaben an, dass dabei auch transdisziplinäre Systeme und Stakeholder wie Finanzdienstleister oder Versicherungen in der Entwicklung berücksichtigt werden müssen. Die Teilnehmer vertreten hingegen nicht die Meinung, dass autonom agierende Fahrzeuge den Fahrzeugmarkt schnell durchdringen und damit verändern werden. Die Vernetzung von Fahrzeugen und Infrastruktur ist für autonom fahrende Fahrzeuge unabdingbar, laut Meinung der Teilnehmer bietet diese jedoch bereits heute neue Potenziale, zum Beispiel bei der Elektromobilität.

Die Entwicklung des Connected Car ist ein noch vergleichsweise neues Entwicklungsfeld für die Automobilhersteller. Dabei steht nicht mehr nur die Entwicklung des Fahrzeugs allein im Fokus, sondern dieses wird Teil eines komplexen System-of-Systems. Bestehende Methoden, Prozesse und Tools der Produktentstehung müssen daher für die Entwicklung des Connected Car weiterentwickelt werden.

Die Entwicklung von Systems-of-Systems ist insbesondere dadurch gekennzeichnet, dass die beteiligten Komponentensysteme teilweise unabhängig voneinander durch eine Vielzahl von Stakeholdern und Organisationseinheiten entwickelt und betrieben werden. Produktgenerationslebenszyklen können dabei unterschiedlich sein. Viele Experten sind sich einig, dass insbesondere in der Definition und Dokumentation von Zielen und Anforderungen, sowie im Management von Änderungen bei Systems-of-Systems neue Herausforderungen entstehen. Um diesen Herausforderungen begegnen zu können, muss zunächst ein grundlegendes

---

<sup>330</sup> vgl. Albers, Kurrle & Munker 2016

<sup>331</sup> vgl. Abramovici 2015

Verständnis über die Entwicklung von Zielen und Anforderungen bei einem SoS geschaffen werden.

Die vielfältigen Ziele und Anforderungen sowie deren Wechselwirkungen und Randbedingungen müssen dabei durchgängig als Zielsystem über alle Entwicklungsphasen, Produktgenerationen und beteiligten Organisationseinheiten aufgefasst und dokumentiert werden. Da die am weitesten verbreitete Dokumentationsform für Ziele und Anforderungen noch immer auf der informalen Sprache basiert und Microsoft Office weiterhin eines der meist genutzten Werkzeuge ist, benötigen Entwickler einen Ansatz, der mit diesen Randbedingungen umgehen kann. Semantische Metainformationen können Entwicklungsartefakte verlinken, auch wenn sich deren darunterliegende Struktur unterscheidet. Es wird daher erwartet, dass semantische Technologien Potentiale zur Herstellung von Durchgängigkeit und Konsistenz in einem verteilten Zielsystem besitzen.

Die Entwicklung des Zielsystems ist niemals abgeschlossen. Änderungen treten bei SoS häufig und in den unterschiedlichsten Bereichen auf und können weitreichende Auswirkungen haben. Werden wesentliche Inkonsistenzen bis zur Einführung nicht identifiziert, kann es zum Scheitern des Produkts kommen. Bestehende Ansätze für das Management von Änderungen würdigen die Herausforderungen in einem verteilten Umfeld bisher nicht ausreichend. Entwickler benötigen daher einen Ansatz, welcher beim Umgang mit häufigen Änderungen in einem verteilten SoS Entwicklungsumfeld unterstützt.

### **3.2 Zielsetzung**

Ziel dieser Arbeit ist die Entwicklung eines Ansatzes für die Entwicklung und Dokumentation von Zielsystemen, welcher den gesamten Verlauf des Produktentstehungsprozess über Produktgenerationen hinweg unterstützt und dabei insbesondere die Herausforderungen in einem verteilten, teilweise unabhängig voneinander agierenden Umfeld berücksichtigt. Der Ansatz soll dazu im Kern dabei unterstützen, Durchgängigkeit und Konsistenz im Zielsystem sowie in Richtung angrenzender Partialmodelle der Entwicklung herzustellen. Dies soll zum einen durch eine Methode für die durchgängige Dokumentation von Zielen und Anforderungen in einem verteilten Umfeld erreicht werden. Zum anderen soll der Ansatz durch eine Methode für das Management von Änderungen in der verteilten Produktentwicklung erweitert werden.

Im Folgenden werden drei zentrale, aufeinander aufbauende Forschungsfragen formuliert, welche zur Erreichung der Zielsetzung der Arbeit beantwortet werden sollen. Ihre Beantwortung stellt den wesentlichen wissenschaftlichen Beitrag dieser Arbeit dar.

Die erste Forschungsfrage zielt darauf ab, den Bedarf sowie Ziele, Anforderungen und Randbedingungen für eine neue Methode zur Entwicklung von Zielsystemen im Kontext des Connected Car zu beleuchten. Dazu sollen insbesondere Herausforderungen in der Entwicklung aus der Perspektive der Entwickler identifiziert werden und ein tieferes Verständnis davon geschaffen werden, wie Zielsysteme für Systems-of-Systems entwickelt werden und wie Entwickler mit Änderungen umgehen müssen. Damit wird der grundlegende Forschungsschwerpunkt für die weiteren Untersuchungen gelegt.

**Forschungsfrage 1: Initiales Zielsystem des Ansatzes**

Welche Anforderungen, Ziele und Randbedingungen werden an einen Ansatz zur Entwicklung von Zielsystemen für SoS am Beispiel des Connected Car aus der Perspektive der Entwickler gestellt?

Basierend auf den Untersuchungen zur Beantwortung der zweiten Forschungsfrage soll mit der Forschungsfrage 2 untersucht werden, wie ein Ansatz gestaltet sein muss, der die definierten Ziele, Anforderungen und Randbedingungen für die Entwicklung und das Management von Zielsystemen für SoS am Beispiel des Connected Car erfüllt.

**Forschungsfrage 2: Gestaltung eines Ansatzes**

Wie muss ein Ansatz zur Unterstützung der durchgängigen Entwicklung von Zielsystemen bei verteilten und (teilweise) unabhängig voneinander agierenden Handlungssystemen über Produktgenerationen hinweg gestaltet sein?

Die letzte Forschungsfrage zielt dahin zu beantworten, ob der entwickelte Ansatz die durchgängige Entwicklung und das Management von Zielsystemen unterstützt und ob eine Praxistauglichkeit gegeben ist.

**Forschungsfrage 3: Evaluation des Ansatzes**

Welchen Beitrag kann der Ansatz zur Verbesserung der Durchgängigkeit und des Management von Änderungen bei verteilten und (teilweise) unabhängigen, interdisziplinären Handlungssystemen in der Praxis liefern?

## 4 Forschungsdesign

In diesem Kapitel wird das Forschungsvorgehen vorgestellt, welches zur Beantwortung der definierten Forschungsfragen gewählt wurde. Der Rahmen für die durchgeführten Untersuchungen wird durch eine dreijährige Forschungs Kooperation zwischen dem IPEK – Institut für Produktentwicklung und dem Bereich Porsche Connect der Dr. Ing. h.c. F. Porsche AG gesetzt. Die Entwicklung des Connected Car dient dabei als Fallbeispiel für die Entwicklung von Zielsystemen durch verteilte, teilweise unabhängig voneinander agierende Handlungssysteme.

Um eine Übertragbarkeit der Ergebnisse sicherzustellen, beschränken sich die Untersuchungen nicht nur auf das Umfeld der Dr. Ing. h.c. F. Porsche AG, sondern beziehen auch weitere Unternehmen und Forschungsaktivitäten im Bereich Connected Car mit ein.

### 4.1 Forschungskoooperation mit der Dr. Ing. h.c. F. Porsche AG

Die Dr. Ing. h.c. F. Porsche AG (Porsche AG) ist ein Sportwagenhersteller im Premiumsegment mit knapp 25.000 Mitarbeitern und Sitz in Stuttgart-Zuffenhausen. Die Produktpalette umfasst derzeit die Modelle Macan, Cayenne, 918 Spyder, 911 Carrera, 718 und Panamera (vgl. Abbildung 37).



Abbildung 37: Aktuelle Modellpalette der Dr. Ing. h.c. F. Porsche AG<sup>332</sup>

<sup>332</sup> <https://vmmedia.porsche.de/>, letzter Zugriff am 14.11.2016

Digitale Innovationen sind ein Schwerpunkt der strategischen Ausrichtung der Dr. Ing. h.c. F. Porsche AG.<sup>333</sup> Wesentliche Entwicklungen laufen dabei heute in 8 unterschiedlichen Bereichen (vgl. Abbildung 38). Das aktuelle Dienstportfolio reicht von Echtzeitverkehrsinformationen (Navigation Services) über die Anzeige und Fernsteuerung von Fahrzeugfunktionen (z.B. Standheizung und Klimatisierung) bis zur Anzeige von personalisierten Inhalten (z.B. Nachrichten) im Fahrzeugen.

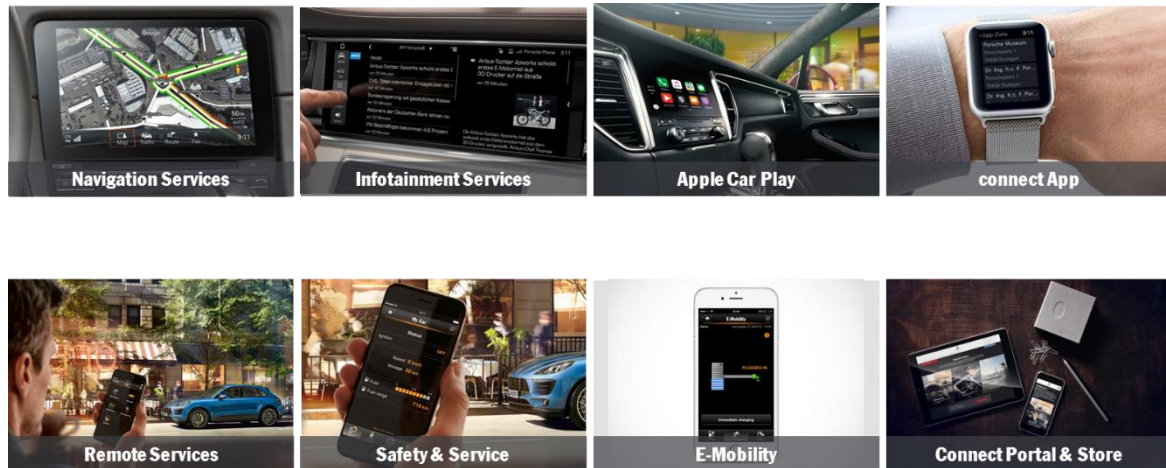


Abbildung 38: Dienstportfolio Porsche Connect<sup>334</sup>

An der Entwicklung des Connected Car bei der Porsche AG sind Mitarbeiter aus insgesamt über 32 Abteilungen der Bereiche Fahrzeugentwicklung, IT, Vertrieb, Strategie, Beschaffung und Finanz beteiligt. Die Mitarbeiter, welche an der Entwicklung des Zielsystems für das Connected Car beteiligt sind haben dabei teilweise einen sehr unterschiedlichen fachlichen Hintergrund. Darüber hinaus sind zahlreiche Zulieferer, Dienstleister, Geschäftspartner und Drittanbieter involviert. Die Entwicklungsaktivitäten für Porsche Connect werden in einem zentralen Projekthaus mit Vertretern aus den unterschiedlichen Ressorts gebündelt. Abbildung 39 gibt einen Überblick über die beim Connected Car involvierten Systeme und Schnittstellen.

Auf Grund der Vielzahl involvierter, teilweise unabhängig voneinander operierender Organisationseinheiten und der starken Ausprägung der Produktgenerationsentwicklung bei der Porsche AG eignet sich dieses Untersuchungsumfeld besonders für die Erreichung der definierten Forschungsziele.

<sup>333</sup> Dr. Oliver Blume, Vorstandsvorsitzender der Porsche AG in einem Interview am 14.04.16 in der Frankfurter Allgemeinen Zeitung

<sup>334</sup> <http://www.porsche.com>, Letzter Zugriff am 14.11.2016



**Fahrzeug / On-Board-Unit**

- Fahrzeugdiagnose / -status
- Echtzeitverkehrsinformation
- Content-bezogene Dienste (z.B. Wetter, Nachrichten, Tankstellenpreise)
- Intelligente Parkplatzsuche

**Drittanbieter / Infrastruktur**

- Navigationsinformationen
- Vernetzte Infrastruktur, z.B. Ampeln, Gefahreninfos
- Social Media Content
- Multimedia Content



Drahtlose Kommunikation,  
Fahrzeugbackend

**Web-/Tablet-Portal**

- Fahrzeugdiagnose / -status
- Fernsteuerung z.B. Sonnendach, Türen, Licht etc.
- Personalisierung

**Smartphone / Wearables**

- Fahrzeugdiagnose / -status
- Fernsteuerung z.B. Türen, Licht, Sonnendach etc.
- Auffinden des Fahrzeugs



Abbildung 39: Das Connected Car – ein System-of-Systems<sup>335</sup>

## 4.2 Design Research Methodology

Das Forschungsvorgehen orientiert sich an der Design Research Methodology (DRM) von BLESSING UND CHAKRABARTI<sup>336</sup>. Die DRM schlägt eine Untergliederung in vier Forschungsphasen vor (vgl. Abbildung 40). Die vorliegende Forschungsarbeit ist zwar linear im Text dargestellt, das Forschungsvorhaben selbst wurde jedoch stark iterativ durchgeführt. So fanden auch noch in späten Phasen auf Grund von Erkenntnisgewinnen Rücksprünge statt. Beispiele hierfür sind eine fortlaufende Anpassung der Zielsetzung, die kontinuierliche Analyse des Forschungsstandes und die Erkenntnis über die Notwendigkeit einer weiteren empirischen Studie im Rahmen der präskriptiven Studie.

<sup>335</sup> In Anlehnung an Johanning & Mildner 2015, S.23, Bilder: <https://vmmedia.porsche.de/>

<sup>336</sup> vgl. Blessing & Chakrabarti 2009

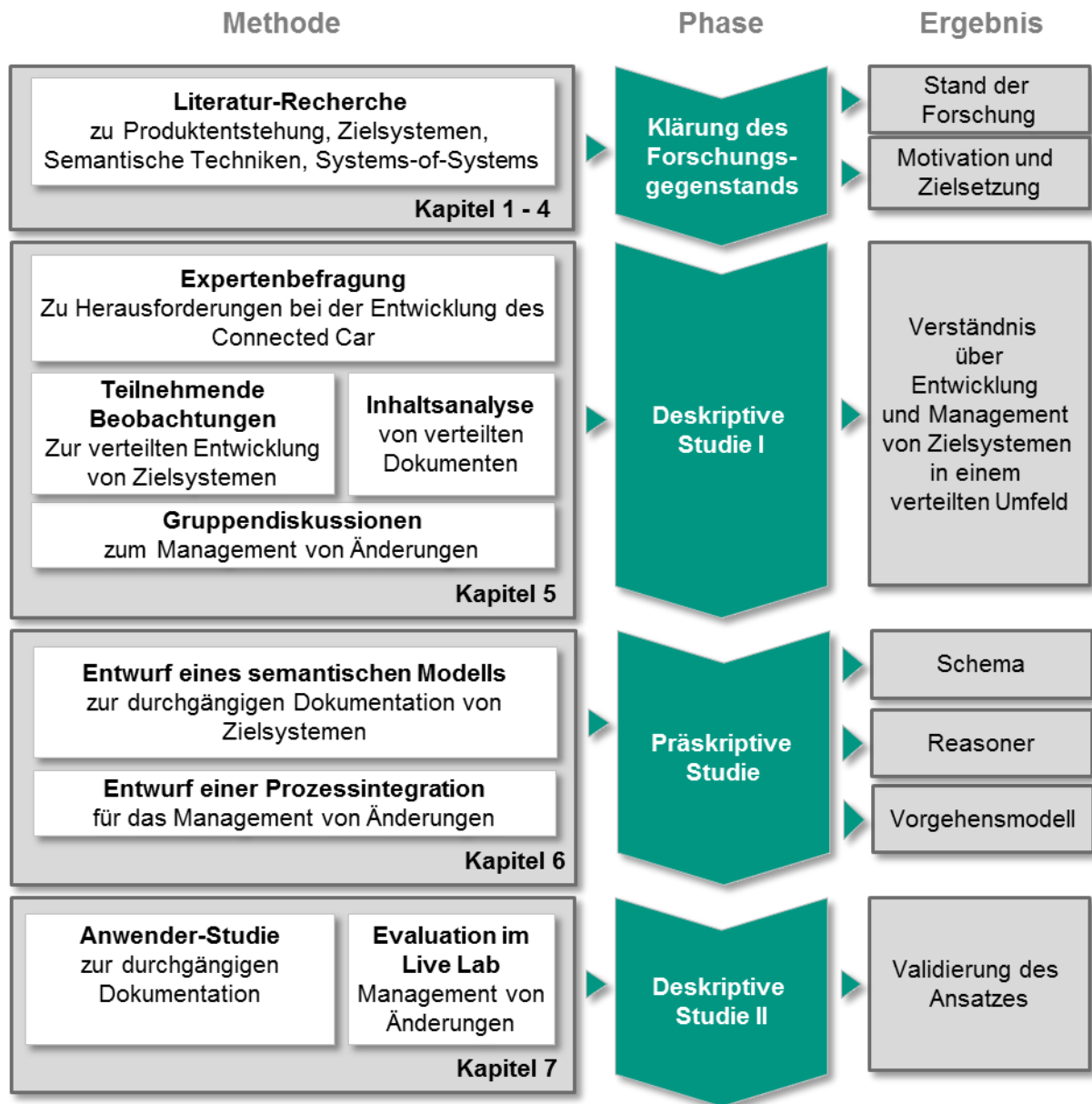


Abbildung 40: Forschungsdesign der Arbeit basierend auf der DRM<sup>337</sup>

In der ersten Phase wird durch eine Literaturrecherche und Eingrenzung der relevanten Forschungsfelder die Basis für die erarbeiteten Forschungsfragen geschaffen. Die Ergebnisse dieser Phase sind in den Kapiteln 1-4 dargestellt.

Im Rahmen der deskriptiven Studie I wird das initiale Zielsystem für einen Ansatz aufgestellt. Dazu wurde als empirischer Untersuchungsgegenstand das Connected Car als Fallbeispiel gewählt. Abbildung 41 gibt einen Überblick über verfügbare und verwendete Methoden zur Datenerhebung. Zur Beantwortung der definierten Forschungsfragen werden im Rahmen der Deskriptiven Studie I sowohl

<sup>337</sup> vgl. Blessing & Chakrabarti 2009

Befragungen, eine teilnehmende Beobachtung sowie eine Dokumentenanalyse durchgeführt. Die Auswahl der Datenerhebungsmethoden orientiert sich jeweils an der Zielsetzung der Untersuchung und wird in Kapitel 5 erläutert.

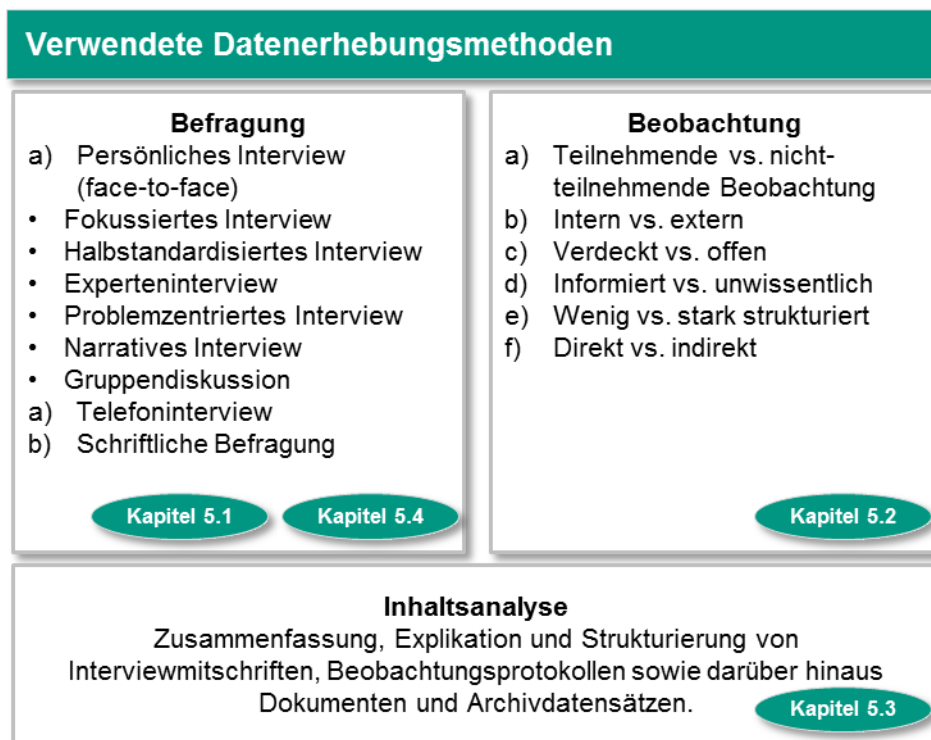


Abbildung 41: Datenerhebungsmethoden und ihre Verwendung im Rahmen der ersten deskriptiven Studie<sup>338</sup>

Im ersten Teil der Untersuchungen der Deskriptiven Studie I wird zunächst der Forschungsbedarf abgesichert und wichtige Randbedingungen an einen Ansatz abgeleitet (siehe auch Kapitel 5.1). Dazu werden semi-strukturierte Leitfadengespräche sowie eine schriftliche Expertenbefragung über einen Online-Fragebogen mit Mitarbeitern von verschiedenen Fahrzeugherstellern, Dienstleistern und Zulieferern, welche im Bereich Connected Car tätig sind, durchgeführt.

Im zweiten Teil wird die Kollaboration bei der Entwicklung von Zielsystemen in den verteilten, heterogenen Handlungssystemen im Umfeld Connected Car untersucht (siehe auch Kapitel 5.2). Dazu wird basierend auf Erkenntnissen einer dreijährigen teilnehmenden Beobachtung aus einem realen Entwicklungsprojekt bei der Dr. Ing. h.c. F. Porsche der grundlegende Entwicklungsprozess abstrakt als Tripel aus Zielsystem, Handlungssystem und Objektsystem dargestellt. Mit den Untersuchungsergebnissen soll insbesondere die Anwendbarkeit des Ansatzes in der Praxis sichergestellt werden.

<sup>338</sup> In Anlehnung an S. Albers et al. 2007, S. 38 und Marxen 2014

Der dritte Teil der deskriptiven Studie fokussiert auf die Struktur und Dokumentation des Zielsystems (siehe auch Kapitel 5.3). Dazu wird als Forschungsmethode eine Inhaltsanalyse auf Entwicklungsartefakten im Umfeld Connected Car bei der Dr. Ing. h.c. F. Porsche AG angewendet.

Im letzten Teil der deskriptiven Studie I soll ein Verständnis über das Management von Änderungen geschaffen werden (siehe auch Kapitel 5.4). Die Erkenntnisse wurden im Rahmen eines halbjährigen regelmäßig tagenden Arbeitskreises im Rahmen von Gruppendiskussionen mit Vertretern aus der Industrie erhoben.

Im Rahmen der Präskriptiven Studie wird die Forschungsfrage 3 („Wie muss ein Ansatz zur Unterstützung der durchgängigen Entwicklung von Zielsystemen gestaltet werden?“) beantwortet (siehe auch Kapitel 6). Dabei werden die gewonnenen Erkenntnisse der Deskriptiven Studie I zur Synthese eines Ansatzes für das Management von Änderungen und zur durchgängigen Verlinkung von Entwicklungsartefakten genutzt.

Die Validierung des entwickelten Ansatzes erfolgt im Rahmen der Deskriptiven Studie II (siehe auch Kapitel 7). Dabei wird insbesondere ein Augenmerk auf die Praxistauglichkeit des entwickelten Ansatzes gelegt. Im Rahmen eines Live-Lab wird die Tauglichkeit des entwickelten Ansatzes für das Management von Änderungen validiert. Durch retrospektive Abbildung von Dokumenten und Änderungen aus einem realen Entwicklungsprojekt wird abschließend der Ansatz zur durchgängigen Abbildung der Entwicklungsartefakte validiert.

## 5 Empirische Untersuchungen zur Ableitung eines initialen Zielsystems

In diesem Kapitel wird zunächst die Forschungsfrage 1 beantwortet, welche Anforderungen, Ziele und Randbedingungen an einen Ansatz zur Entwicklung von Zielsystemen für SoS am Beispiel des Connected Car aus der Perspektive der Entwickler gestellt werden. Aus dieser Forschungsfrage werden vier weitere Forschungsfragen abgeleitet (vgl. Abbildung 42).

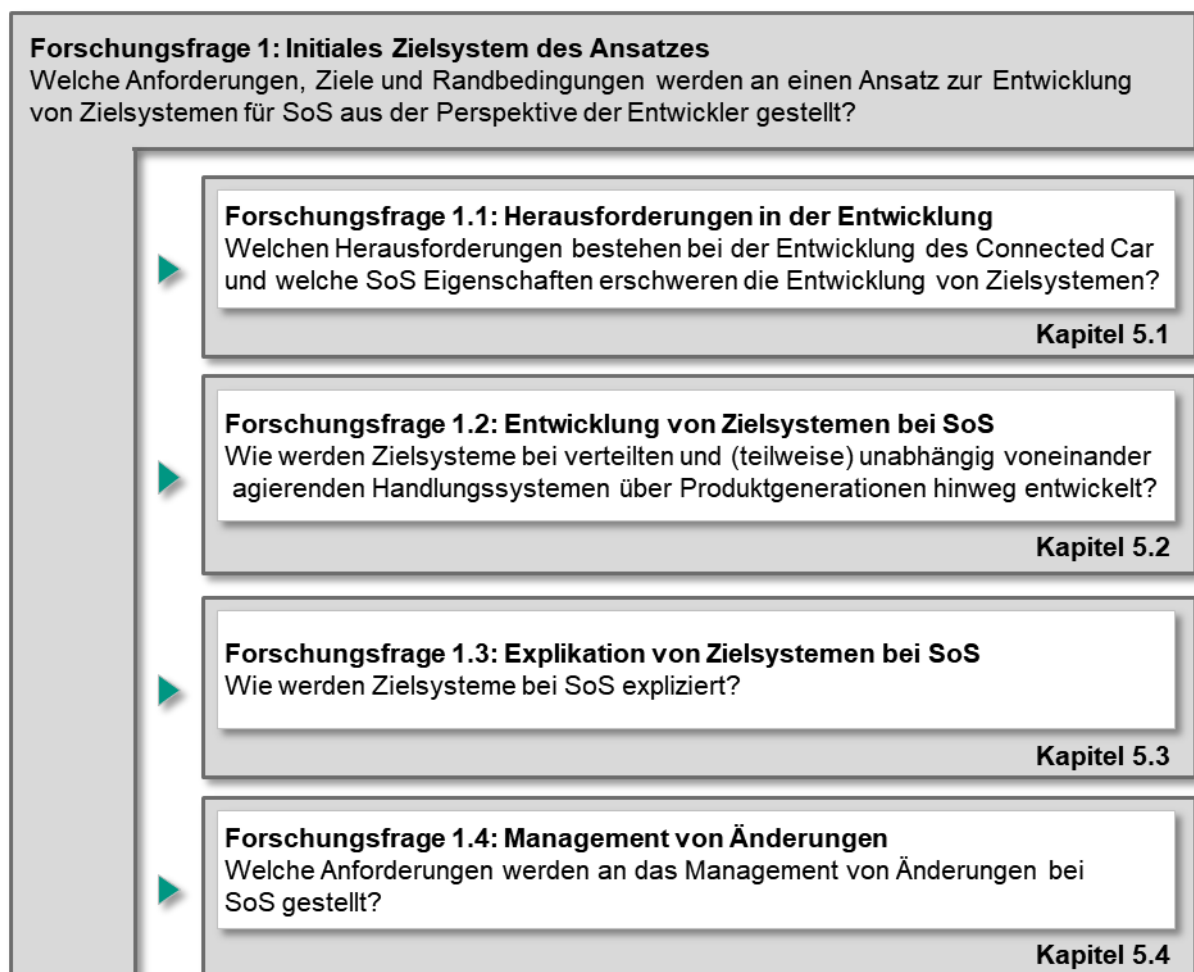


Abbildung 42: Forschungsfragen der ersten deskriptiven Studie

Im Folgenden werden die Ergebnisse von Untersuchungen beschrieben, welche die Herausforderungen bei der Entwicklung und Abbildung von Zielsystemen für das Connected Car beleuchten und ein Verständnis über den Prozess der Zielsystementwicklung und Struktur des Zielsystems anhand von Fallbeispielen aus realen Entwicklungsprojekten im Umfeld des Connected Car darstellen. Diese Untersuchungen sollen neben einem grundlegenden Verständnis der Problematik auch die Basis für die Anwendbarkeit des Ansatzes in der Praxis legen.

## 5.1 Herausforderungen bei der Connected Car Entwicklung

Im ersten Teil der Untersuchungen wird durch eine Vorstudie der Forschungsbedarf durch Experten aus der Entwicklungspraxis abgesichert und wichtige Randbedingungen für einen Ansatz abgeleitet. Mit den Untersuchungen soll insbesondere die Forschungsfrage 1.1 beantwortet werden.<sup>339</sup>

### **Forschungsfrage 1.1: Herausforderungen in der Entwicklung**

Welche Herausforderungen bestehen bei der Entwicklung des Connected Car und welche SoS Eigenschaften erschweren die Entwicklung von Zielsystemen?

#### 5.1.1 Studiendesign

Zur Beantwortung der definierten Forschungsfrage wird eine Online-Befragung von Experten aus dem Bereich Connected Car durchgeführt.<sup>340</sup> Experten sind ein Medium, um Sachverhalte zu erschließen. Diese fungieren dabei als „Zeugen“ für den Untersuchungsgegenstand.<sup>341</sup> Eine Online-Befragung erlaubt eine direkte Datenerfassung und ermöglicht einen einfachen Zugang zu großen Auswahlgruppen.<sup>342</sup> Der Link zur Umfrage wird an Mitarbeiter von fünf Automobilherstellern, 25 Dienstleistern und 12 Beratungsunternehmen mit Erfahrung im Bereich Connected Car gesendet. Dazu werden verschiedene Kommunikationskanäle verwendet, darunter die persönliche Weiterreichung durch private Kontakte aus der Industrie als auch der direkte Kontakt mit Entwicklern oder Kollegen über soziale Netzwerke (wie z.B. XING) und per E-Mail. Im Fragebogen werden persönliche Einstellungen zu Sachverhalten mit Hilfe einer Likert-Skala erfasst. Herausforderungen und Handlungsbedarfe können die Experten darüber hinaus über ein Freitextfeld nennen.

Zur weiteren Exploration von Expertenwissen wurden zusätzlich Interviews mit Mitarbeitern der Dr. Ing. h.c. F. Porsche AG sowie dessen Zulieferern und Dienstleistern durchgeführt. Die Experteninterviews werden als semi-strukturierte Leitfadengespräche<sup>343</sup> aufgebaut. Nur wenige Fragen sind zu Beginn des Gesprächs

---

<sup>339</sup> Die in Kapitel 5.1 dargestellten Untersuchungen sind im Rahmen der Publikation Albers, Kurrle & Klingler 2016 veröffentlicht worden.

<sup>340</sup> Im Rahmen einer betreuten Abschlussarbeit (Schaupp 2015)

<sup>341</sup> vgl. Gläser and Laudel 2006

<sup>342</sup> vgl. Marxen 2014

<sup>343</sup> vgl. Marxen 2014

vorbereitet und viele Fragen können als Reaktion auf die Antworten der Interviewpartner formuliert werden, was eine hohe Flexibilität bei der Datenerhebung erlaubt<sup>344</sup> und besser auf spezifisches Fachwissen der Experten eingegangen werden kann. Das Design und die Durchführung der Expertenbefragung basiert auf dem Modell der Phasen der Datenerhebung von NIESCHLAG<sup>345</sup>. Das Modell wurde in dieser Studie so angepasst und angewandt, damit die Forschungsfragen beantwortet werden. Die Gespräche wurden, nach Zustimmung des jeweiligen Interviewpartners aufgezeichnet, anschließend transkribiert und die Aussagen kategorisiert.

Auf die Anfrage zur Teilnahme an der Online-Umfrage haben insgesamt 113 Mitarbeiter geantwortet, wobei lediglich 25 Teilnehmer Erfahrung im Bereich Connected Car vorweisen konnten. An den Experteninterviews haben insgesamt 20 Experten aus dem Bereich Connected Car teilgenommen. Alle Befragten sind aus dem deutschsprachigen Raum. Einen Überblick über die Teilnehmer der Studie ist in Abbildung 43 dargestellt.

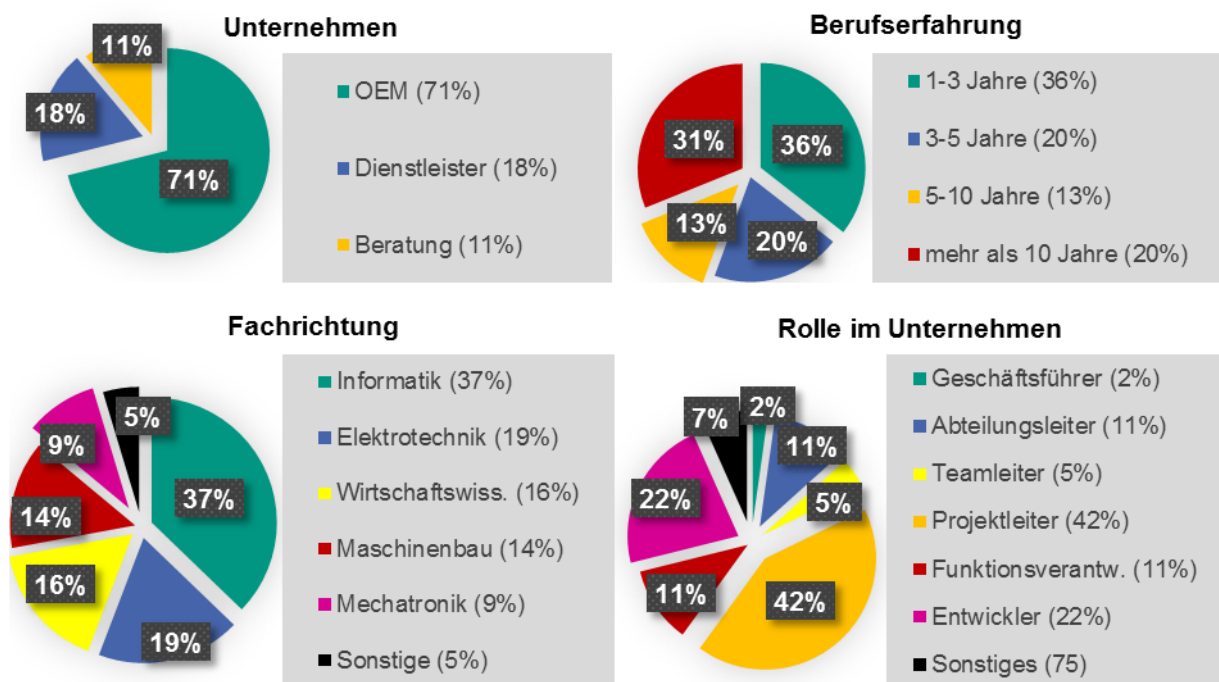


Abbildung 43: Informationen über Teilnehmer der Studie

<sup>344</sup> vgl. Hamann & Erichson 2000

<sup>345</sup> vgl. Nieschlag, Dichtl & Hörschgen 2002

### 5.1.2 Ergebnisse der Expertenbefragung<sup>346</sup>

Herausforderungen und Handlungsbedarfe wurden von den Experten im Rahmen der Experteninterviews und im Rahmen der Online-Befragung (Nennungen über Freitextfeld) genannt.<sup>347</sup> Diese wurden in sechs Kategorien unterteilt.

#### **Vielzahl heterogener Handlungssysteme**

Die Entwicklung des Connected Car erfolgt in einer Vielzahl von heterogenen Handlungssystemen mit unterschiedlichen Entwicklern, Methoden, Prozessen, Tools und Budget. Die Mehrheit der Experten ist sich einig, dass daraus Herausforderungen entstehen, was folgende Einzelaussagen belegen.

- *Vielzahl involvierter Systeme: „Beim Connected Car wirkt eine Vielzahl von Systemen zusammen, wobei jeweils hohe Abhängigkeiten zwischen den Systemen bestehen.“ „Sehr viele Systeme entstehen dabei parallel.“ „Unterschiedliche Entwicklungszyklen sind möglich.“*
- *Vielzahl von Ansprechpartnern: „An der Entwicklung ist eine Vielzahl von verteilten Ansprechpartnern beteiligt.“ „Fast die gesamte Wirkkette der Automobilunternehmen ist betroffen und fast alle Bereiche müssen ihre Prozesse anpassen.“ „Die Sprachen der Beteiligten können unterschiedlich sein.“*
- *Vielzahl von externen Partnern: „Neben den Automobilherstellern ist eine Vielzahl von externen Partnern, überwiegend aus dem IT-Bereich, beteiligt. Diese Welten müssen synchronisiert werden. Beide Seiten müssen sich dabei anpassen - die Autoindustrie muss schneller werden, wobei die IT-Industrie lernen muss, dass ein Auto auch mal 15 Jahre im Markt sein kann.“*

#### **Änderungen in der Entwicklung**

Die Entwicklung des Connected Car bewegt sich in einem sehr volatilen Umfeld. Viele Experten sehen eine große Herausforderung im Management der Änderungen während der Entwicklung:

- *Häufige Änderungen: „Es gibt häufige Änderungen auf Grund der parallelen Entwicklung der vielen involvierten Systeme.“ „50% der Anforderungen haben sich im Laufe des Projektes geändert.“ „Der aktuelle Fahrzeugentwicklungsprozess ist dafür zu träge.“ „Im Fahrzeuglastenheft werden Anforderungen definiert, die bereits beim Verfassen veraltet sein können.“*

---

<sup>346</sup> vgl. auch Albers, Kurrle & Klingler 2016



- Kombination aus zentralem und dezentralem Änderungsmanagement: *„Änderungen sind auf mehreren Detaillierungsebenen möglich und müssen an betroffene, ggf. detailliertere Projekte kommuniziert werden.“* *„Die Kombination von zentralem und dezentralem Anforderungsmanagement ist erforderlich.“*
- Bewertung der Auswirkungen von Änderungen: *„Änderungen können sehr viele Parteien betreffen. Welche Änderungen für wen interessant sind und durch wen entschieden werden müssen, ist nicht trivial und muss in einem übergreifenden Prozess definiert werden.“*
- Verteilte Dokumentation des Entscheidungsstandes: *„Auf Grund der häufigen Änderungen kann die Aktualität der Dokumente nicht ausreichend sein, da Informationen nicht nachgezogen werden bzw. sehr verteilt vorliegen. Die Folge ist ein Delta zwischen tatsächlicher Umsetzung und Lastenheft.“*

### **Informationsaustausch**

Auf Grund der unabhängigen Entwicklung der beteiligten Systeme überblickt beim Connected Car kein Akteur alles vollständig. Eine wesentliche Herausforderung liegt daher im Informationsaustausch:

- *„Eine Hauptschwierigkeit ist ein Informationsdefizit und die Intransparenz der beteiligten Systeme (Black Box).“*
- *„Eine große Herausforderung liegt im Informationsaustausch und der Identifikation des Ist-Zustandes der Systeme.“*

### **Verteilte, heterogene Dokumentation**

Auf Grund der Vielzahl involvierter Systeme und Akteure ergeben sich Herausforderungen bei der Definition und Dokumentation von Zielen und Anforderungen:

- Dokumentation: *„Die verwendeten Anforderungsdokumente sind nicht einheitlich bzw. nicht in einem einheitlichen Tool dokumentiert.“* *„Man muss theoretisch alle betroffenen Lastenhefte von A-Z lesen um zu schauen, ob man etwas vergessen hat.“* *„Alle Anforderungen in einem Dokument oder Tool zu dokumentieren ist auf Grund der vielen involvierten, heterogenen Projekte und Partner nicht möglich.“*
- Unabhängige Entwicklung: *„Die Systeme besitzen auch Funktionalität, welche über das Connected Car hinausgeht. Diese Ziele und Anforderungen können in Konflikt mit den Zielen des Connected Car stehen.“* *„Viele parallele Anforderungen an das Connected Car müssen aufgeteilt werden und auf verschiedene Entwicklungsprojekte aufgeteilt, jedoch am Ende wieder zusammengefasst werden.“* *„Auf Grund der unabhängigen Entwicklung ist häufig nicht klar, bis zu welchem Zeitpunkt in der Entwicklung Anforderungen an welches System definiert werden*

können.“ *„Anforderungen an das Fahrzeug müssen frühzeitig in den Automobil-Entwicklungsprozess integriert werden.“*

- Durchgängigkeit und Nachverfolgbarkeit: *„Bestehende Prozesse sind häufig nur auf die einzelnen Systeme ausgelegt.“* *„Auf Grund der vielen Projektgrenzen gibt es einen sehr hohen Abstimmungsbedarf bei der Zieldefinition und die Nachverfolgbarkeit wird erschwert.“*
- Anforderungen auf mehreren Ebenen: *„Manche Anforderungen betreffen die gesamte System-Kette, andere hingegen betreffen nur einzelne Systeme.“* *„Anforderungen waren nicht in ausreichendem Detaillierungsgrad vorhanden.“*
- Geographische Verteilung: *„Rechtliche Unterschiede oder verschiedene Drittanbieter und Partner auf Grund der geographischen Verteilung müssen bei der Anforderungsdefinition berücksichtigt werden.“* *„Dabei geht es auch um Verhinderung des Missbrauchs personenbezogener Daten ohne Kenntnis des Fahrzeugeigentümers bzw. Fahrers.“*

### **Verteilte Verifizierung & Validierung**

Im Rahmen der Entwicklung des Connected Car wird durch eine Verifizierung auf mehreren Ebenen beurteilt, ob ein System oder das gesamte SoS richtig entwickelt wurde. Die Eigenschaften der Systeme werden dazu formal mit den spezifischen Anforderungen und Zielen abgeglichen. Durch eine Validierung der Systemkette wird zusätzlich an kundenorientierten Kriterien hinsichtlich Einsatzzweck oder Benutzererwartung beurteilt, ob das richtige Produkt entwickelt wurde (z.B. durch kundennahe Erprobung). Die Experten sehen hier folgende Herausforderungen:

- Testen und Definition von Testgrenzen: *„Im Testen liegt eine große Herausforderung. Jede Abteilung testet ihr eigenes System, im Gesamtverbund muss dies allerdings auch abgesichert werden.“* *„Verknüpfung von Anforderungen mit Testfällen zur Ermittlung der Testabdeckung ist eine Herausforderung.“*
- Fehleridentifikation und Fehlermanagement: *„Die Fehlersuche wird auf Grund der vielen involvierten Systeme erschwert.“* *„Früher gab es nur ein Fehlermanagement-System, heute gibt es verschiedene [...]“*

### **Mangelnde Ausschöpfung der Potentiale einer Produktgenerationsentwicklung**

Viele Experten wünschen sich eine zunehmende Modularisierung und Standardisierung der Connected Car Systeme und Dienste:

- *„Eine einheitliche Systemumgebung für alle OEMs fehlt.“* *„Viele Insellösungen existieren.“*
- *„Standardisierung und Modularisierung der Services muss angestrebt werden.“*

Aus Sicht der Mehrheit aller Befragten sind daher Anpassungen an bestehenden Prozessen, Methoden und Tools notwendig, um die Entwicklung des Connected Car zu unterstützen (vgl. Abbildung 44).

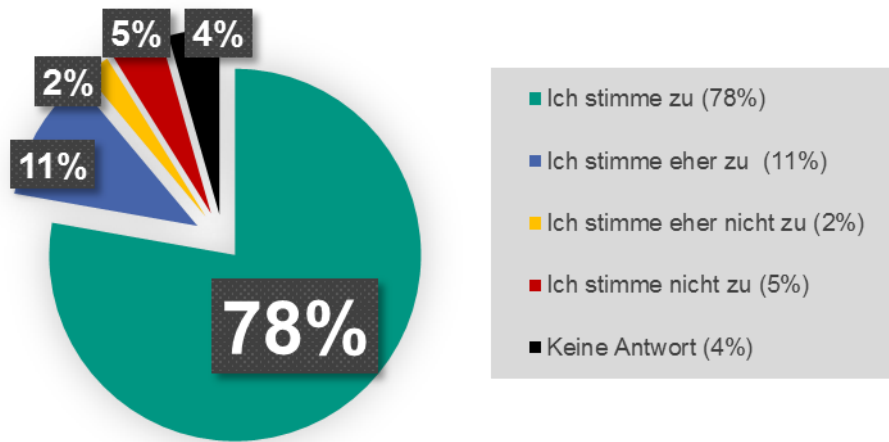


Abbildung 44: Notwendigkeit für Änderungen an bestehenden Prozessen, Methoden und Tools (Konsolidiertes Ergebnis der Online-Umfrage und Experteninterviews)

Im Rahmen der Experteninterviews wurden die Teilnehmer weiter dazu aufgefordert zu bewerten, welche der SoS-Typischen Eigenschaften<sup>348</sup> die Definition des Zielsystems erschweren. Die Ergebnisse sind in Abbildung 45 dargestellt.

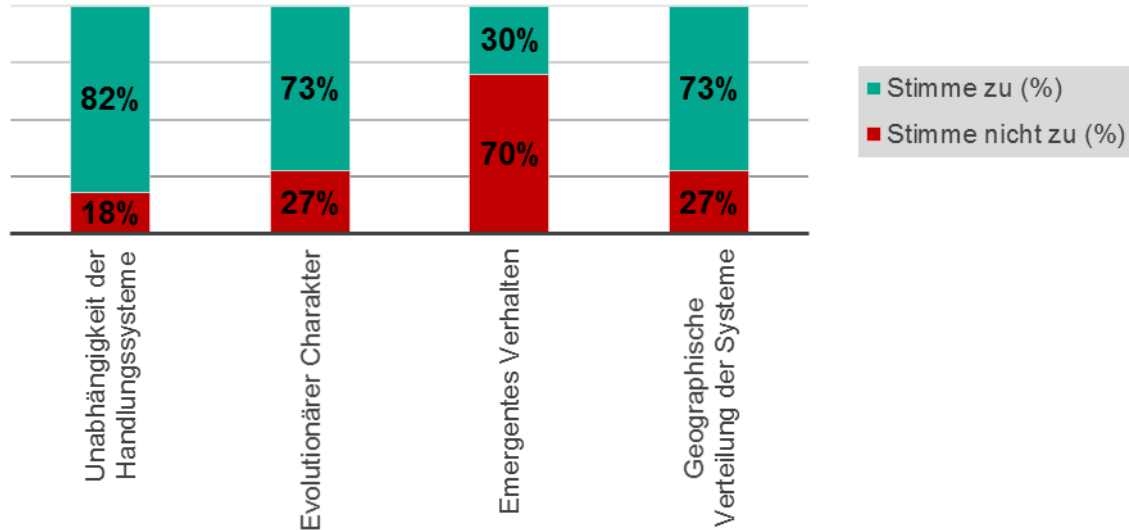


Abbildung 45: Bewertung der SoS Eigenschaften, welche die Definition des Zielsystems erschwert haben

<sup>348</sup> vgl. Kapitel 2.4

Die Ergebnisse zeigen, dass die große Mehrheit der Befragten der Meinung ist, dass die Eigenschaften von SoS neue Herausforderungen bei der Entwicklung von Zielsystemen für das Connected Car darstellen. Die Hauptschwierigkeit sehen die Teilnehmer in der Unabhängigkeit der Handlungssysteme. Weniger Einfluss auf die Zielsystementwicklung wird durch das emergente Verhalten des SoS gesehen.

### 5.1.3 Implikationen

Die Ergebnisse der Studie zeigen, dass die Herausforderungen aus Sicht der Experten sehr vielfältig sind. Dabei wird offengelegt, dass viele bereits in der Produktentwicklung bekannten Herausforderungen im Kontext Connected Car eine noch stärkere Bedeutung erlangen. Insbesondere die Unabhängigkeit der beteiligten Handlungssysteme sehen die Teilnehmer als größten Einflussfaktor. In Tabelle 4 werden sechs Ziele abgeleitet, welche an eine Entwicklungsunterstützung für das Connected Car gestellt werden.

Ziele		Begründung
1	Finden eines gemeinsamen Ansatzes bei Methoden, Prozessen und Tools für eine Vielzahl heterogener Handlungssysteme.	Unterschiedliche Methoden, Prozessen und Tools existieren in der Entwicklung und müssen zusammen gebracht werden. Beteiligte Stakeholder können dabei andere Ziele verfolgen, wie das SoS selbst. Dabei gilt es auch, unterschiedliche Entwicklungszyklen zu synchronisieren.
2	Unterstützung des Informationsaustausch und Offenlegung, welche Informationen für wen interessant sind.	Auf Grund einer Vielzahl von involvierten Stakeholdern liegt ein wesentlicher Erfolgsfaktor in der Schaffung eines funktionierenden Informationsaustauschs. Stakeholder werden schnell von der Informationsflut überrollt, denn nicht alle Informationen sind für jeden Relevant.
3	Kombination aus zentralem und dezentralem Management von Änderungen.	Häufige Änderungen erfordern ein gut funktionierendes Management von Änderungen, welches insbesondere bei der Bewertung von Änderungen über alle beteiligten Systeme und Stakeholder unterstützt und unabhängig vom SoS agierende Handlungssysteme berücksichtigt.
4	Herstellung von Durchgängigkeit und Konsistenz bei der verteilten, heterogenen Dokumentation.	Eine heterogene Dokumentation von Entwicklungsartefakten erschwert das Herstellen von Durchgängigkeit und Konsistenz.
5	Nahtlose Validierung und Fehlermanagement.	Auf Grund einer Vielzahl beteiligter Systeme und Stakeholder sind Testgrenzen häufig nicht klar. Es besteht die Gefahr einer unvollständigen Validierung und Verifizierung im Gesamtverbund.
6	Produktgenerationen übergreifende Standards und Referenzproduktmodelle.	Derzeit existieren noch viele „Insellösungen“. Eine Standardisierung und Modularisierung im Sinne einer Produktgenerationsentwicklung kann die bekannten Potentiale heben.

Tabelle 4: Herausforderungen bei der Entwicklung des Connected Car

Die identifizierten Herausforderungen und abgeleiteten Ziele betreffen in vielen Bereichen die Zielsystementwicklung. Beim Connected Car wirkt eine Vielzahl von Systemen zusammen, wobei hohe Abhängigkeiten zwischen den Zielen und Anforderungen an die beteiligten Systeme existieren. Die bestehenden Prozesse sind häufig nur für das zu entwickelnde System und nicht übergreifend ausgelegt. Die Systeme können in unterschiedlichen Phasen ihres Produktlebenszyklus sein, was zu einem unterschiedlichen Reife- und Härtegrad<sup>349</sup> des Zielsystems führt. Informationen werden sehr heterogen dokumentiert, was zu einem Verlust von Durchgängigkeit und Konsistenz führen kann. Es wird jedoch vermutet, dass noch weitere Faktoren zu einem Verlust von Durchgängigkeit und Konsistenz im Zielsystem eines SoS führen können, was weitere Untersuchungen notwendig macht. Dazu kommt ein hoher Abstimmungsaufwand auf Grund einer Vielzahl von beteiligten Stakeholdern. Neben den Automobilherstellern ist eine Vielzahl von externen Partnern, überwiegend aus dem IT-Bereich, beteiligt. Diese „Welten“ müssen nach Aussage der Experten synchronisiert werden.

Die Entwicklung des Connected Car bewegt sich des Weiteren in einem sehr volatilen Umfeld, wobei Änderungen an einer Vielzahl von Systemen möglich sind. Häufige Änderungen der Ziele und Anforderungen sind die Folge. Änderungen können dabei sehr viele Parteien betreffen. Mehrere Experten sehen eine große Herausforderung darin zu bewerten, welche Änderungen für wen interessant sind und durch wen sie entschieden werden müssen. Zur Herstellung von Durchgängigkeit und Konsistenz im Zielsystem ist ein gut funktionierendes Management von Änderungen, welches auch unabhängig vom SoS agierende Handlungssysteme berücksichtigt, daher essentiell. Welche Anforderungen an das Management von Änderungen gestellt werden, muss jedoch noch untersucht werden.

Die von den Experten genannten Herausforderungen zeigen auch, dass der Wunsch besteht, Potentiale einer Produktgenerationsentwicklung noch weiter zu nutzen. Ein Ansatz für die durchgängige Entwicklung und Dokumentation des Zielsystems muss daher auch frühere Produktgenerationen unterstützen.

Mit den identifizierten Herausforderungen und extrahierten Anforderungen und Randbedingungen an einen Ansatz wird kein Anspruch auf Vollständigkeit und universelle Gültigkeit gelegt werden. Die Relevanz der genannten Aspekte wird jedoch über die Zusammenstellung der Teilnehmer und die methodische Vorgehensweise im Rahmen der Expertenbefragung legitimiert und folglich als Gültig betrachtet.

---

<sup>349</sup> vgl. auch Kapitel 2.2.2

## 5.2 Verständnis über die verteilte Entwicklung von Zielsystemen

Ziel dieser Untersuchungen ist zunächst die Schaffung eines grundlegenden Verständnisses über die Kollaboration bei der Entwicklung von Zielsystemen in den verteilten, heterogenen Handlungssystemen im Umfeld Connected Car. Dabei sollen insbesondere folgende Forschungsfragen untersucht werden:

### **Forschungsfrage 1.2: Entwicklung von Zielsystemen bei SoS**

Wie werden Zielsysteme bei SoS am Beispiel Connected Car entwickelt?

Dazu gehören folgende Teilfragen:

- Wie lässt sich die Entwicklung abstrakt als Tripel aus Zielsystem, Handlungssystem und Objektsystem darstellen?
- Welche Ursachen führen zu einem Verlust von Durchgängigkeit und Konsistenz im Zielsystem?
- Welche Randbedingungen für eine Toolauswahl zur Explikation von Zielsystemen gibt es?

### 5.2.1 Studiendesign

Zur Beantwortung der oben genannten Forschungsfragen werden im Rahmen einer dreijährigen Forschungs Kooperation mit der Dr. Ing. h.c. F. Porsche AG Erkenntnisse aus teilnehmenden Beobachtungen als Fallstudienuntersuchung im Bereich Connected Car gewonnen. Die Erkenntnisse aus den Beobachtungen werden ergänzt durch Gruppendiskussionen mit verantwortlichen Entwicklern und leitenden Angestellten der Dr. Ing. h.c. F. Porsche AG.

Eine teilnehmende Beobachtung erlaubt eine direkte Datensammlung, da der Beobachter selbst zur Datenquelle wird.<sup>350</sup> Teilnehmende Beobachtungen zeichnen sich durch die Authentizität der gewonnenen Daten aus. Sie ermöglichen durch Ihre Offenheit und die Problemorientierung die Erforschung komplexer sozialer Systeme, über die bisher noch wenig bekannt ist und sie zwingt Forscher ins Feld zur direkten Kontaktaufnahme mit dem Untersuchungsgegenstand.<sup>351</sup> Gleichzeitig besteht jedoch die Gefahr, dass Einfluss auf die Untersuchungsumgebung genommen wird.<sup>352</sup> Die Methode ist dabei nicht auf die Auskunftsbereitschaft und die verbalen Fähigkeiten

---

<sup>350</sup> vgl. Marxen 2014

<sup>351</sup> vgl. Atteslander 2010

<sup>352</sup> vgl. Marxen 2014

von Versuchspersonen angewiesen.<sup>353</sup> Die teilnehmende Beobachtung eignet sich daher besonders als Datenerhebungsmethode zur Erarbeitung der definierten Forschungsziele.

Die Untersuchungen wurden im Zeitraum von 2013 bis 2016 durchgeführt.

### **5.2.2 Ergebnisse der teilnehmenden Beobachtung**

Im Folgenden werden die Ergebnisse der teilnehmenden Beobachtung vorgestellt.

Die Entwicklung des Zielsystems für das Connected Car erfolgt durch eine Vielzahl verteilter, teilweise parallel arbeitender Entwicklungsteams. Das Zielsystem unterteilt sich dabei in eine Menge partieller Zielsysteme, welche in mehreren Dokumenten und Software-Tools expliziert werden (vgl. Abbildung 46). Diese Entwicklungsartefakte haben nicht nur den Zweck, den Wissensaustausch zwischen den Entwicklern oder Entwicklungsteams und anderen beteiligten Stakeholdern zu unterstützen, sondern bilden in der Regel die Informationsbasis für eine Entscheidung zu einem definierten Meilenstein in der Entwicklung. Die Entwicklung der partiellen Zielsysteme verläuft iterativ auf mehreren Detailebenen, wobei In-Stage und Cross-Gate Iterationen<sup>354</sup> beobachtet werden konnten. Durchgängigkeit und Konsistenz war dabei nicht immer gegeben. Dabei konnten mehrere Ursachen für einen Verlust von Durchgängigkeit und Konsistenz identifiziert werden.

---

<sup>353</sup> vgl. Kinnear und Taylor 1996

<sup>354</sup> vgl. Meboldt, Matthiesen & Lohmeyer 2012

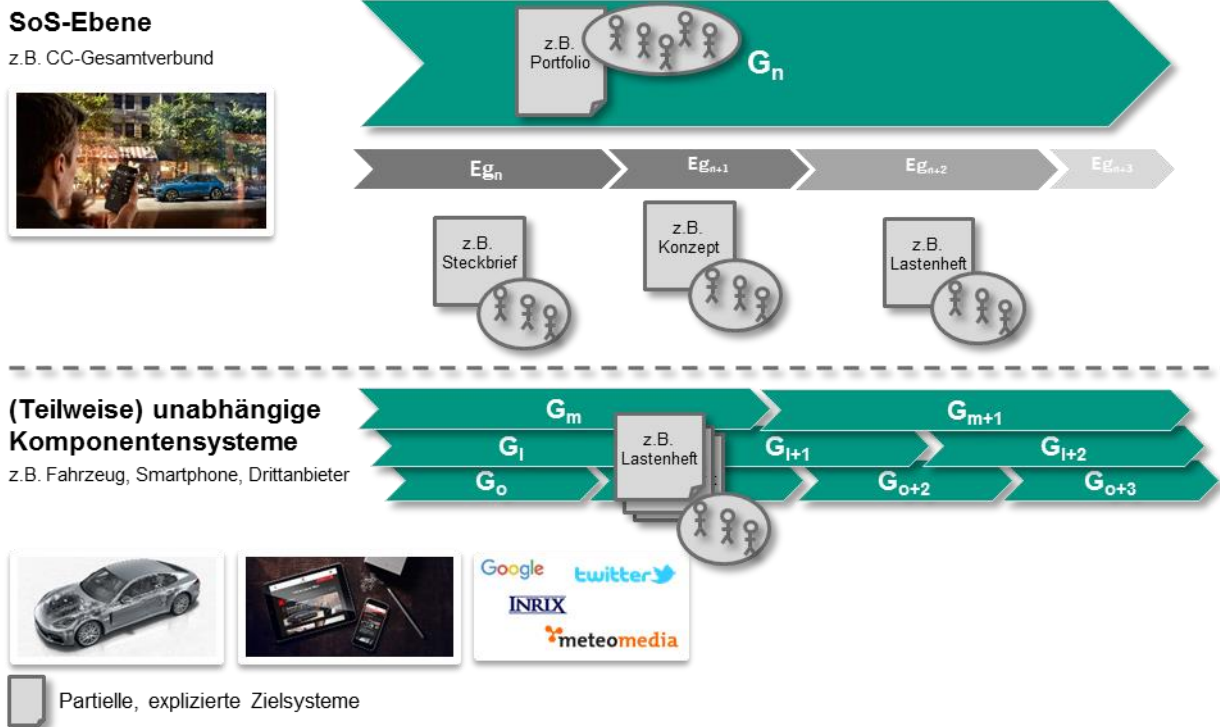


Abbildung 46: Partielle explizierte Zielsysteme in der Entwicklung im Bereich Porsche Connect

Die Entwicklung des Connected Car bei der Porsche AG erfolgt in mehreren Stages, wobei am Ende jeder Stage ein Meilenstein mit fest definierten Entwicklungsartefakten definiert wurde. Verantwortlichkeiten und Handlungssysteme (z.B. Methoden der Anforderungserhebung oder verwendete Tools) für die Entwicklung des Zielsystems können sich dabei über Stages hinweg ändern.

**Beispiel:** In einer frühen Phase<sup>355</sup> wird für das Connected Car ein Dienstesteckbrief mit Microsoft PowerPoint erstellt, welcher die wesentlichen Ziele auf einem sehr groben Detaillierungslevel beinhaltet. Dieser baut häufig auf einem bestehenden Referenzprodukt (z.B. ein Online-Dienst einer früheren Fahrzeuggeneration oder eine bestehende Smartphone App) oder auf externen Informationen (z.B. bestehende Funktionen eines möglichen Entwicklungspartners) auf. Auf Basis des analysierten Referenzproduktes sowie weiterer Bewertungen durch die Fachbereiche (z.B. Entwicklungskosten, Marktpotential) werden Produktprofile definiert und Chancen und Risiken identifiziert, welche eine Grundlage einer Umsetzungsentscheidung durch ein befähigtes Gremium (z.B. Vorstand) darstellen. Anforderungen und Ziele werden dann im Laufe des Entwicklungsprozess weiter konkretisiert oder durch Informationen aus bestehenden Referenzproduktmodellen

<sup>355</sup> Hierbei wird die Frühe Phase im Sinne der PGE verstanden, vgl. A Albers, Bursac, and Wintergerst 2015 und Bursac 2016



übernommen und in einem Lastenheft in IBM Rational Doors oder Microsoft Word definiert. Hierin sind Anforderungen an den Ende-zu-Ende Connected Car Systemverbund definiert. Das Connected Car Dienstportfolio der Porsche AG wird in einer zentralen Portfolioliste mit Microsoft Excel verwaltet.

Die Handlungssysteme haben dabei nur teilweise Ressourcen geteilt, was zu heterogenen Methoden, Prozessen und Tools in der Entwicklung führt. Im untersuchten Fallbeispiel konnten dabei drei Typen von Beziehungen zwischen den Handlungssystemen auf den unterschiedlichen Entwicklungs-Ebenen (vgl. auch Abbildung 46) beobachtet werden:

1. Vollständig geteilte Ressourcen
2. Teilweise geteilte Ressourcen
3. Vollständig unabhängig agierende Handlungssysteme

Der erste Typ stellt eine vollständige Inkludierung der Handlungssysteme dar. Dieser Typ kann innerhalb der Entwicklung zum Beispiel in einer Abteilung betrachtet werden. Die Ressourcen des Handlungssystems werden vollständig geteilt. Dies bezieht sich sowohl auf das Budget, als auch auf Entwickler, Methoden, Prozesse und Tools.

Der zweite Typ stellt eine teilweise Überdeckung der Handlungssysteme dar. Ressourcen werden hier geteilt, jedoch verfügt jedes Handlungssystem auch über eigene Ressourcen. Dieser Typ einer Beziehung findet z.B. in der parallelen Entwicklung in zwei unterschiedlichen Bereichen (z.B. Fahrzeugentwicklung und IT) statt. Eine Synchronisation findet häufig in Gremienterminen statt. Die Zielsysteme können einen unterschiedlichen Reifegrad besitzen.

Der dritte Typ beschreibt eine vollständige Unabhängigkeit des Handlungssystems auf Ebene  $n$  und  $n-1$ . Insbesondere im letzten beschriebenen Fall verlaufen die Entwicklungsprozesse nicht zwangsläufig synchronisiert.

Im untersuchten Fallbeispiel konnte ein Verlust von Durchgängigkeit und Konsistenz beim Übergang zu neuen Entwicklungs- oder Produktgenerationen beobachtet werden. Das zugrundeliegende Referenzzielsystem entwickelt sich auch nach Übernahme weiter.

**Beispiel:** Für welche Länder ein bestimmter Online-Dienst angeboten werden soll wird in einer frühen Phase in einer Portfolioliste durch den Connected Car Portfolioverantwortlichen der Porsche AG definiert und von allen Betroffenen in späteren Entwicklungsphasen für die Entwicklung berücksichtigt. Bei Änderung der Länderabdeckung (z.B. auf Grund von rechtlichen Rahmenbedingungen) muss diese Information auch im Dienstesteckbrief oder in einem Lastenheft angepasst werden.

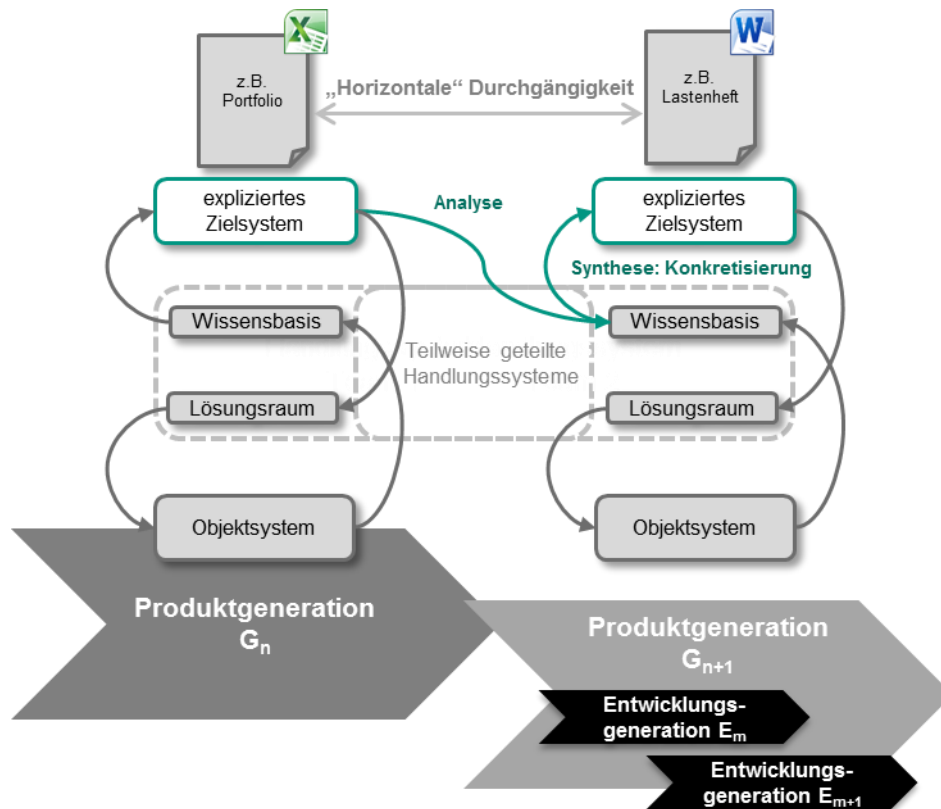


Abbildung 47: Verlust „horizontaler Durchgängigkeit“ bei neuen Entwicklungs- und Produktgenerationen

Auf Grund der hohen technischen Komplexität des Connected Car ist die Entwicklung in verschiedene technische Entwicklungsbereiche (z.B. Fahrzeugentwicklung, IT) unterteilt. Die Entwicklung erfolgt somit auf mehreren Entwicklungsebenen, welche zu einem System-of-Systems integriert werden müssen. Anforderungen an die Ende-zu-Ende Kette müssen an die einzelnen Entwicklungsbereiche heruntergebrochen werden.

Bei diesem Top-Down Entwicklungsvorgehen werden zunächst Ziele und Anforderungen auf Ebene  $n$  durch ein Handlungssystem definiert. Durch eine Funktionszuordnung sowie Dekomposition von Anforderungen und Zielen wird eine Menge von (Sub-)Systemen festgelegt, welche zur Erfüllung der Anforderungen auf Ebene  $n$  benötigt werden (vgl. Abbildung 48).

Dies ist im betrachteten Fallbeispiel bereits in einer frühen Phase erfolgt und konkretisiert sich im Laufe der Entwicklung. Auf Ebene  $n-1$  erfolgt schließlich eine Verfeinerung der auf Ebene  $n$  definierten Ziele und Anforderungen, welche Relevant für das (Sub-)System sind. Je nach Komplexität des zu entwickelnden Systems kann sich der Vorgang auf beliebigen, ggf. darunter liegenden Ebenen wiederholen. Anforderungen und Ziele werden folglich auf mehreren Ebenen definiert, wobei das explizierte, partielle Zielsystem auf Ebene  $n$  beim Top-Down Vorgehen jeweils einen wichtigen Input für das Zielsystem auf Ebene  $n-1$  liefert.

**Beispiel:** Ein Anwendungsfall aus dem Dienstesteckbrief „Fernbedienung Standheizung“ lautet:

„Der Kunde kann die Standheizung seines Fahrzeugs über eine Smartphone App fernsteuern:

- Aktiviert oder deaktiviert die Standheizung adhoc,
- Kann 3 Einmaltimer programmieren, wobei immer nur einer aktiv sein kann
- Die Standheizung unterscheidet abhängig von der Temperatur, ob Heizung oder nur Innenraumlüftung aktiviert wird“

Ein zugehöriges Beispiel für eine Anforderung auf Ebene n-1 in einem Lastenheft für eine Anwendung („Business Service“) auf einem IT Backend lautet:

„Der Business Service soll sicherstellen, dass nur ein Timer aktiv ist.“

Da die Entwicklung auf Ebene n nach Dekomposition und Verfeinerung der Anforderungen auf Ebene n-1 weiter iterativ verläuft, müssen Änderungen an darunter liegende Ebenen kommuniziert werden, um Durchgängigkeit im gesamten Zielsystem zu gewährleisten. Dies war im beobachteten Fallbeispiel nicht immer gegeben, wodurch es zu Inkonsistenzen im Zielsystem kam (vgl. Abbildung 48).

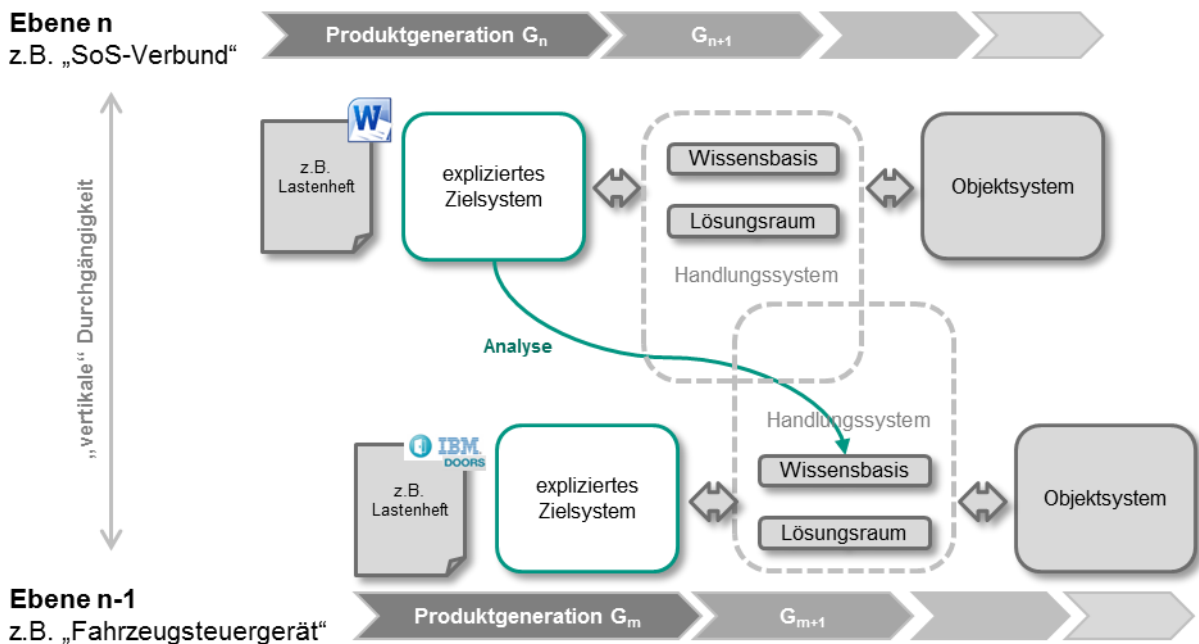


Abbildung 48: Verlust „vertikaler Durchgängigkeit“ bei einem Top-Down Vorgehen

Neben einem Top-Down Vorgehen konnte auch ein Bottom-Up Vorgehen beobachtet werden. Beim Bottom-Up Vorgehen analysieren die Entwickler auf Ebene n Entwicklungsartefakte aus dem Objektsystem von Ebene n-1 und lassen Erkenntnisse aus der Analyse in die Wissensbasis mit einfließen, welcher für die Definition des Zielsystems verwendet wird. Ein Beispiel für ein solches Vorgehen ist die Integration einer bestehenden Funktionalität (z.B. durch den Drittanbieter Google)

in den Connected Car Verbund zu Bereitstellung einer neuen Connected Car Funktion. Das Referenzprodukt ist also außerhalb der Systemgrenze des OEMs, wodurch Informationen nicht so leicht zugänglich sind, was wiederum das Entwicklungsrisiko steigert. In diesem Fall kennen die Verantwortlichen auf *Ebene n-1* den Zweck des Verbundes auf *Ebene n* nicht unbedingt. Bei einer Weiterentwicklung oder Ablösung einer bestehenden Funktion auf *Ebene n-1* kann Durchgängigkeit und Konsistenz verloren gehen. Änderungen auf *Ebene n-1* müssen folglich an *Ebene n-1* kommuniziert oder durch Entwickler auf *Ebene n* beobachtet werden, insbesondere dann wenn die Entwicklung auf *Ebene n-1* (teilweise) unabhängig von der Entwicklung auf *Ebene n* abläuft – was im untersuchten Fallbeispiel beobachtet werden konnte.

Anpassungen am Zielsystem konnten auch auf Grund von Änderungen an anderen Partialmodellen als notwendig erachtet werden. Dies lässt sich damit begründet, dass die Entwicklung der einzelnen Partialmodelle iterativ und in starker Wechselwirkung zueinander vorgenommen wird.<sup>356</sup>

**Beispiel:** Eine Verschiebung im Projektplan der Entwicklung der Smartphone-App zur Bereitstellung der oben genannten Anforderung kann die Anpassung eines Terminziels im Dienstesteckbrief oder der Portfolioliste notwendig machen.

Auf Grund mangelnder Verlinkung der Entwicklungsartefakte war ein sehr hoher Koordinationsaufwand notwendig, um Durchgängigkeit und Konsistenz im Zielsystem sicherzustellen. Trotz der mangelnden Eigenschaften für die durchgängige Abbildung von Anforderungen und Zielen werden Tools von Microsoft Office auch im untersuchten Fallbeispiel häufig verwendet, um Ziele und Anforderungen zu dokumentieren und zu verwalten. Trotz der sehr guten Eigenschaften im Hinblick auf Durchgängigkeit wurden modellbasierte Methoden und Tools nur sehr selektiv und nur in Verbindung mit anderen Tools wie Microsoft Word oder einem web-basierten Wiki eingesetzt.

Eine Begründung, wieso die Entwickler Zielsystemelemente nicht in einem zentralen und integrierten Tool (z.B. Modellbasierter Ansatz mit Cameo Systems Modeler oder RE Tool IBM Rational Doors) explizieren und verlinken, sondern nach wie vor auf sehr heterogene Tools setzen kann auf eine Vielzahl von identifizierten Randbedingungen zurückgeführt werden.

---

<sup>356</sup> vgl. auch Ebel 2015, S. 114

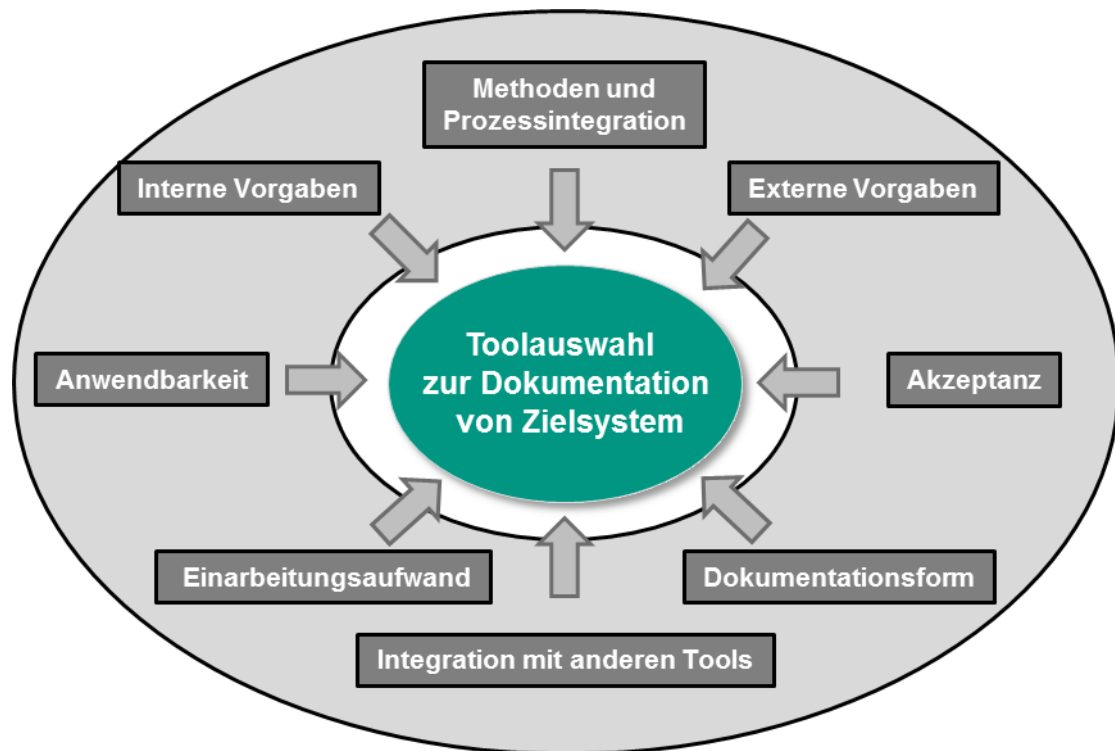


Abbildung 49: Randbedingungen für eine Toolauswahl zur Dokumentation von Zielsystemen

**Interne Vorgaben:** Tools zur Dokumentation von Anforderungen und Zielen können vom Projektteam (z.B. von einer Projektqualitätssicherung), einer Abteilung (z.B. „weil es alle in der Abteilung verwenden“) oder vom Unternehmen (z.B. weil das Unternehmen bereits einen Lizenzvertrag mit einem Anbieter verhandelt hat) für ein Vorhaben gesetzt werden können.

**Externe Vorgaben:** Im untersuchten Fallbeispiel waren Tools (z.B. IBM Rational Doors) oder spezielle Anforderungsformate teilweise von Lieferanten gefordert.

**Integration mit anderen Tools:** Die Elektrik/Elektronikentwicklung der Porsche AG wird durch über 40 Softwaretools unterstützt. Die Anzahl der Softwaretools steigt weiter, wenn zusätzliche Tools aus dem IT-Ressort oder von Dritten miteinbezogen werden. Diese Tools sind häufig miteinander vernetzt (z.B. Anforderungsmanagement und Testmanagement, Aufgabenverwaltung, Projektmanagement), was beim Ablösen einzelner Tools häufig zum Ersatz einer Reihe weiterer Tools führen kann.

**Methoden und Prozessintegration:** In den verwendeten Tools waren häufig Vorlagen oder Workflows definiert, welche die verwendeten Methoden und Prozesse in der Entwicklung im Tool abbilden.

**Dokumentationsform:** Ein Dokument in Microsoft PowerPoint ist auf eine Präsentation ausgelegt. Wenn Anforderungen und Ziele auf einer hohen Detailebene oder mit Skizzen häufig präsentiert werden müssen eignen sich Office Tools gut. Bei

einer Dokumentation in IBM Rational Doors wird weniger Wert auf eine entsprechende Präsentation gelegt.

**Anwendbarkeit / Einarbeitungsaufwand:** Insbesondere wenn Entwicklungsartefakte einem breiten Stakeholderkreis (z.B. Dienstesteckbrief) zur Verfügung gestellt werden muss, spielt der Aufwand zur Einarbeitung eine große Rolle. Tools wie Microsoft PowerPoint werden dann Tools wie IBM Rational Doors oder Cameo Systems Modeler vorgezogen.

**Verfügbarkeit und Kosten:** Nicht alle verfügbaren Tools können zwangsläufig auch in einem Unternehmenskontext eingesetzt werden (z.B. weil bei einem Open Source Projekt keine Support und Wartungsverträge angeboten werden). Weiter haben Unternehmen ein Interesse daran, die Kosten für Softwaretools so gering wie möglich zu halten und dafür ggf. Einbußen bei den Funktionen eines Tools in Kauf zu nehmen.

**Generelle Akzeptanz der Nutzer:** In Diskussion mit den Entwicklern kristallisierte sich häufig heraus, dass gewisse Vorlieben der Nutzer die Entscheidung in Richtung eines Tools beeinflussen, welche nicht zwangsläufig rational begründbar sondern auf dem subjektives Empfinden bei der Nutzung der Tools zurückzuführen sind.

### 5.2.3 Implikationen

Die Ergebnisse der empirischen Untersuchungen bei der Dr. Ing. h.c. F. Porsche AG liefern zunächst ein grundsätzliches Verständnis über die Kollaboration bei der Entwicklung von Zielsystemen in einem verteilten, teilweise unabhängig voneinander agierendem Umfeld. Weiter konnten die definierten Forschungsfragen beantwortet werden.

Entwickler nutzen Dokumente und Tools, um ihr implizites Wissen zu teilen<sup>357</sup>. Entwicklungsartefakte bilden eine wichtige Basis für Entscheidungen im Produktentwicklungsprozess. Das Zielsystem wird dabei nicht in einem einzigen Dokument oder Tool über den gesamten Verlauf dokumentiert, sondern es gibt eine Vielzahl partieller, explizierter Zielsysteme, welche durch unterschiedliche Entwickler expliziert werden. Das Herstellen von Durchgängigkeit und Konsistenz war durch eine mangelnde Transparenz bezüglich vorhandener Abhängigkeiten zwischen Entwicklungsartefakten sowie einem fehlenden durchgängigen Management von Änderungen deutlich erschwert. Insgesamt ist ein hoher Koordinationsaufwand notwendig, um Konsistenz herzustellen. Inkonsistenzen und Verlust von Informationen können daher häufig beobachtet werden.

---

<sup>357</sup> vgl. auch Kapitel 2.2.3

Es wurde weiter festgestellt, dass Durchgängigkeit nicht nur über den zeitlichen Verlauf und über unterschiedliche Detaillierungsgrade (z.B. zwischen Anforderungen an den gesamten Systemverbund und Schnittstellen einer Komponente) sondern auch über mehrere Produktgenerationen hergestellt werden muss. Des Weiteren sollen Relationen zu angrenzenden Partialmodellen abgebildet werden können. Anforderungen und Ziele werden zeitlich in unterschiedlichen Produkt- und Entwicklungsgenerationen entwickelt, wobei die Handlungssysteme sich über die Zeit verändern können. Produktgenerationslebenszyklen können dabei unterschiedlich sein. Im untersuchten Fallbeispiel wurde beobachtet, dass Durchgängigkeit zwischen Produktgenerationen teilweise nicht vorhanden waren. Die Potentiale einer Produktgenerationsentwicklung wurden folglich nicht genutzt. Die Explikation des Zielsystems kann über die Zeit variieren. Damit sind in der Regel auch Anpassungen an den explizierten Zielsystemen früherer Phasen notwendig. Entsprechend muss „horizontale“ Durchgängigkeit auch über mehrere Produktgenerationslebenszyklen hergestellt werden. Dieser Zusammenhang ist in Abbildung 50 dargestellt.

Anforderungen und Ziele werden auf mehreren Ebenen definiert, wobei Anforderungen und Ziele durch eine Dekomposition bzw. Funktionszerteilung einzelnen Bauteilen und Komponenten zugeordnet und dort weiter verfeinert werden. Handlungssysteme können dabei vollständig unabhängig voneinander agieren. Unterschiedliche Entwicklungszyklen, Methoden und Tools sind die Folge. Ein Ansatz muss folglich bei der Herstellung „vertikaler Durchgängigkeit“ unterstützen.

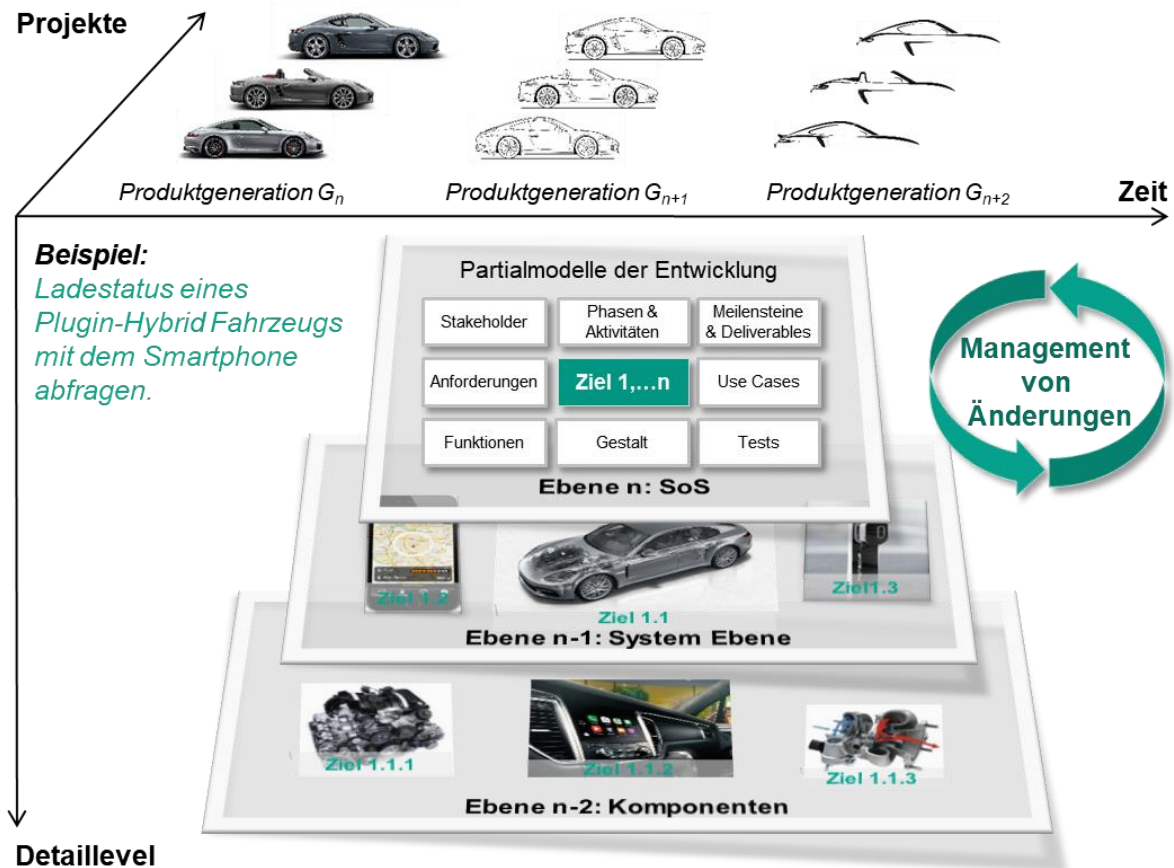


Abbildung 50: Dimensionen der Durchgängigkeit bei der Dokumentation von Zielsystemen<sup>358</sup>

Dabei steht die Toolauswahl unter einer Menge von Randbedingungen, welche die Einführung eines zentralen Ansatzes (z.B. modellbasiertes Anforderungsmanagement) über alle Entwicklungsphasen und Detailebenen, insbesondere in einem SoS-Umfeld mit (teilweise) unabhängig voneinander agierenden Organisationseinheiten, nicht möglich macht.

Um die Akzeptanz des Ansatzes in der Praxis sicherzustellen, müssen die Kosten für die Verwendung eines neuen Ansatzes in positivem Verhältnis zum Nutzen stehen. Der Einarbeitungsaufwand sollte folglich gering sein und die Notwendigkeit für Schulungen vermieden werden<sup>359</sup>. Eine Verringerung des Abstimmungsaufwandes und eine Verringerung des Dokumentationsaufwandes bei gleichzeitiger Verbesserung der Durchgängigkeit und Konsistenz muss durch einen Ansatz angestrebt werden.

<sup>358</sup> vgl. auch Albers et al. 2017

<sup>359</sup> vgl. Subjektive Effizienz nach Sauter 2011



### 5.3 Verständnis über die verteilte Explikation von Zielsystemen

Ergänzend zu den Untersuchungen in Kapitel 5.2 soll mit dieser Untersuchung ein Verständnis davon geschaffen werden, wie verteilte Zielsysteme dokumentiert werden. Folgende Forschungsfragen sollen dabei untersucht werden:

#### **Forschungsfrage 1.3: Explikation von Zielsystemen bei SoS**

Wie werden Zielsysteme bei SoS expliziert? Weitere Teilfragen sind:

- Wie können Entwicklungsartefakte referenziert und beschrieben werden?
- Welche Typen von Relationen gibt es zwischen Entwicklungsartefakten und deren Elementen?
- Wie werden implizit Relationen zwischen Entwicklungsartefakten „im Kopf der Entwickler“ hergestellt?

#### 5.3.1 Studiendesign

Als Forschungsmethode wird eine Inhaltsanalyse von Entwicklungsartefakten im Umfeld Connected Car bei der Dr. Ing. h.c. F. Porsche AG durchgeführt. Die Inhaltsanalyse ist ein empirisches Datenerhebungsverfahren womit sich Kommunikationsinhalte wie Texte, Bilder und Filme untersuchen lassen.<sup>360</sup> Damit sollen neben der Beschreibung und Auswertung der eigentlichen Entwicklungsartefakte auch Zusammenhänge seiner Entstehung und Verwendung erkannt werden.<sup>361</sup> HARDER<sup>362</sup> unterscheidet dabei drei Funktionen der Inhaltsanalyse:

- Die diagnostische Funktion: Die Bedingungen, aus denen Texte hervorgegangen sind
- Die prognostische Funktion: Das zukünftige Verhalten der Textquelle
- Die kommunikationstheoretische Funktion: Der Wirkzusammenhang zwischen Sender und Empfänger von Inhalten

Mit den Untersuchungen soll im Wesentlichen der Fokus auf die kommunikationstheoretischen Funktionen gelegt werden. Durch die Analyse soll das Material soweit reduziert werden, dass die wesentlichen Inhalte erhalten bleiben.<sup>363</sup>

---

<sup>360</sup> vgl. Atteslander 2010

<sup>361</sup> vgl. Atteslander 2010, S. 196

<sup>362</sup> vgl. Harder 1974, S. 226

<sup>363</sup> vgl. Mayring 1990, S. 54

Als Basis für die Untersuchungen wurden alle Entwicklungsartefakte bei der Porsche AG ausgewählt, welche Anforderungen oder Ziele an den Online-Dienst „Nachrichten“ für das Fahrzeug Panamera (Generation 2) beschreiben.

Mit dem Online-Dienst „Nachrichten“ kann sich der Kunde aktuelle Nachrichten im Fahrzeug anzeigen oder vorlesen lassen. Im Auslieferungszustand sind dazu mehrere Nachrichtenlieferanten (z.B. Tagesschau) standardmäßig vorkonfiguriert. Der Kunde hat darüber hinaus die Möglichkeit, über ein Kundenportal im Internet zusätzliche Nachrichten Feeds zu seinem Account im Web Portal hinzuzufügen (siehe auch Abbildung 51).



Abbildung 51: Anzeigen und Personalisierung von Nachrichten im Porsche Panamera<sup>364</sup>

Obwohl der Dienst vor Kunde nur einfache Funktionen erfüllt, sind an seiner Entwicklung eine Vielzahl von Mitarbeitern aus unterschiedlichen Bereichen beteiligt, weshalb dieses Fallbeispiel für die Inhaltsanalyse als besonders geeignet angesehen wird.

Insgesamt wurden unterschiedliche Versionen von 25 Dokumenten aus 10 Entwicklungsprojekten betrachtet, welche Ziele und Anforderungen an den Online-Dienst Nachrichten für das Connected Car bei der Dr. Ing. h.c. Porsche AG beschreiben oder auf diese verweisen. Die Dokumente wurden im Zeitraum zwischen 2013 und 2016 erstellt oder aktualisiert.

### 5.3.2 Ergebnisse der Inhaltsanalyse

Die Ergebnisse der Untersuchungen bestätigen den Stand der Forschung<sup>365</sup> hinsichtlich verwendeter Tools für die Dokumentation von Anforderungen und Zielen. Im untersuchten Fallbeispiel gehören Microsoft Office Tools zu den am häufigsten verwendeten Software-Werkzeugen (vgl. Abbildung 52). In agilen Software-Entwicklungsprojekten werden häufig auch Web-basierte Wiki und Aufgabenverwaltungstools wie Atlassian Confluence und Jira verwendet, da diese

<sup>364</sup> Quelle: porsche.com, Zugriff zuletzt am 17.09.16

<sup>365</sup> vgl. Kapitel 2.2.3

entsprechende Plugins für eine agile Entwicklung (z.B. Abbildung eines Product Backlog) bieten.

Anforderungen und Ziele werden in der Regel in natürlicher Sprache auf Deutsch und Englisch formuliert. Eine semi-formale Dokumentation zum Beispiel mit modellbasierten Ansätzen erfolgt selten und wenn dann nur in Kombination mit natürlicher Sprache. Eine formale Dokumentation mit streng definierter Semantik wird nur in einem Entwicklungsartefakt verwendet. In 16 Entwicklungsartefakten werden Skizzen oder Bilder zur Erklärung von Sachverhalten als Ergänzung zur natürlichen Sprache verwendet. Direkte Verweise auf andere Dokumente oder Tools werden nur in drei Entwicklungsartefakten durch Verwendung von Dokumenten-Links verwendet.

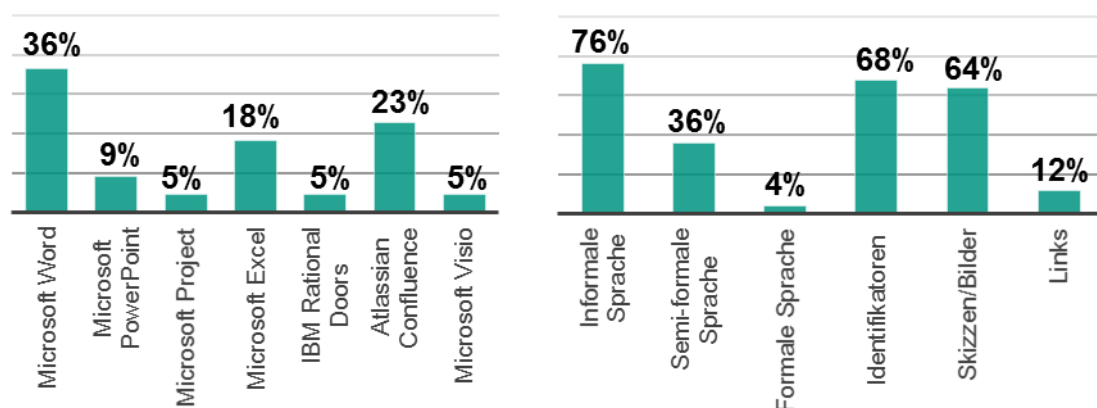


Abbildung 52: Verwendete Tools zur Dokumentation von Anforderungen und Zielen (links) / Verteilung der Detailebene der Dokumente (rechts)

Die Dokumentation von Begründungen zu Herkunft oder Definition von Anforderungen und Zielen wird in Ausnahmefällen nur vereinzelt dokumentiert. Häufig werden hierfür dann die Kommentarfunktion bei Microsoft Word oder eine Änderungshistorie im Entwicklungsartefakt selbst verwendet.

### Eindeutige Referenzierung von Entwicklungsartefakten

Zur eindeutigen Referenzierung von Entwicklungsartefakten werden sechs Merkmale identifiziert, wobei nicht alle Merkmale bei allen Projekten verwendet wurden:

- Name des Dokuments oder Dokumenttyp
- Status
- Version
- Projekt
- Projektphase
- Speicherort/Pfad

Im untersuchten Fallbeispiel wird in der Regel entweder ein eigener Name für das Dokument vergeben oder der Dokumenttyp ausgewiesen (z.B. Lastenheft, Projektplan oder Steckbrief). Zusätzlich zu Angaben des Status und der Version

wurden häufig noch Projektangaben gemacht (z.B. „Lastenheft Infotainmentsystem Panamera“). In einigen Fällen wurde zusätzlich noch eine Projektphase (z.B. Vorstudie oder Umsetzung) zur Referenzierung verwendet, da Verantwortlichkeiten für Dokumente sich über Projektphasen hinweg ändern können.

### Status und Version von Entwicklungsartefakten

Status oder Version eines Entwicklungsartefakts wurden entweder im Dateinamen, im Dokument oder Tool selbst oder gar nicht dokumentiert. Im untersuchten Fallbeispiel wurden für unterschiedliche Entwicklungsartefakte unterschiedliche Bezeichner für Status und Versionierung verwendet. Der Status eines Dokuments kann dabei im Wesentlichen in fünf Phasen unterteilt werden (vgl. Abbildung 53).

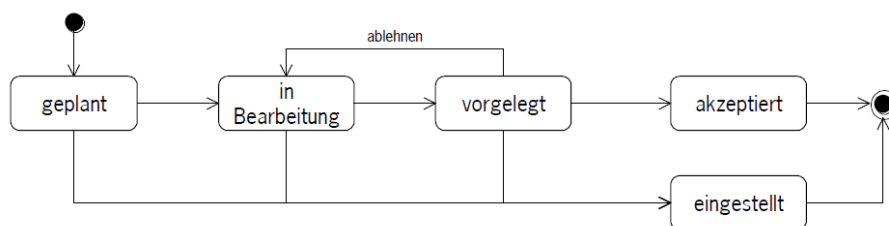


Abbildung 53: Beispiel für einen Status-Workflow bei Entwicklungsdokumenten

Im untersuchten Fallbeispiel waren mehrere Stufen eines Freigabeprozesses möglich, wobei abhängig vom Entwicklungsartefakt der Freigabeprozess individuell sein kann (z. B. in der Reihenfolge Autor, Projektteam, Projektlenkungskreis, Vorstand). Ein Dokument kann folglich zwar von einem Projektteam akzeptiert sein, das Dokument wurde aber noch nicht einem befähigten Entscheidungsgremium (z.B. Vorstand) final vorgelegt und freigegeben.

Alle untersuchten Dokumente wurden versioniert, wobei keine einheitliche Versionierung vorhanden war. Am häufigsten wurde eine einfache Versionierung (Beispiel: „v1“) oder eine Unterscheidung zwischen Hauptversion, Nebenversion und Revision (Beispiel: v1.00.01) verwendet. Um das geänderte Dokument wieder in den Zustand „akzeptiert“ zu überführen, ist eine explizite Freigabe erforderlich, wobei unterschiedliche Stakeholder aus dem System-Verbund involviert werden müssen.

Bei einer neuen Revision sind gegenüber der letzten akzeptierten Version lediglich kleine Fehlerbeseitigungen vorgenommen worden. Zur Überführung in den Zustand „akzeptiert“ ist keine explizite Freigabe erforderlich. Bei der letzten Versionierungsart ist es schwierig festzustellen, wann von einer „wesentlichen“ oder „unwesentlichen“ Änderung gesprochen wird. Eine direkte Verbindung zum Kreis der Betroffenen wird nicht hergestellt.

Im untersuchten Fallbeispiel wird Version und Status häufig händisch gepflegt, da entsprechende Toolunterstützung nicht verfügbar war. Dieses Vorgehen wird jedoch

als sehr fehleranfällig angesehen. Im untersuchten Fallbeispiel konnte bei 18 von 25 Dokumenten eine unvollständige Pflege der Status und Versionsinformation bei Änderungen festgestellt werden. Insgesamt liegen pro Dokument durchschnittlich über 5 dokumentierte Versionen vor. Ebenso wird nur selten vermerkt, welche Änderungen am Dokument vorgenommen wurden, was eine Nachverfolgbarkeit über die Zeit stark erschwert. Auf Grund der hohen Abhängigkeiten zwischen den Entwicklungsartefakten führt dies zu einem inkonsistenten Zielsystem des SoS.

In 90 Prozent aller Fälle werden Anforderungen und Ziele gemeinsam in einem Dokument expliziert. Weiter konnte festgestellt werden, dass Ziele und Anforderungen in der Regel nicht getrennt von anderen Partialmodellen der Entwicklung dokumentiert werden<sup>366</sup>.

### **Verweis auf Funktionen und Baustruktur**

Bei allen Dokumenten, in denen Ziele, Anforderungen oder Anwendungsfälle definiert werden gab es ebenfalls eine (grobe) Funktions- oder Baustruktur. Die Entwickler stellten folglich häufig einen direkten Bezug zur geplanten Gestalt her (als Prämisse der definierten Anforderungen und Ziele). Umgekehrt wird in allen untersuchten Dokumenten, welche eine Gestalt (Baustruktur) definiert haben auf Anforderungen verwiesen. Dokumente, welche eine Teststrategie oder ein Testvorgehen definieren referenzierten in 5 von 6 Fällen direkt auf Ziele und Anforderungen.

### **Verweis auf Stakeholder und Meilensteine**

In 20 von 25 Dokumenten werden Personen, Gruppen oder Organisationen, die ein berechtigtes Interesse am Verlauf oder Ergebnis des Produktentstehungsprozesses haben (Stakeholder) dokumentiert. Dazu gehörten im Wesentlichen allgemeine Angaben wie Name, Firmen- oder Abteilungszugehörigkeit und Kontaktdaten. Relevante Stakeholder wurden in der Regel direkt im Dokument gepflegt. Dabei kann unterschieden werden zwischen einem (verantwortlichen) Autor und einem reinen Interessenten.

In 13 von 25 Dokumenten wird ein direkter Bezug zu Meilensteinen oder Phasen und Aktivitäten der Produktentwicklung hergestellt.

### **Explizite Relationen zwischen Artefakten**

Neben der Referenzierung von Elementen einzelner Partialmodelle werden auch Referenzen zwischen Artefakten abgebildet. Dabei konnten drei Typen identifiziert werden:

---

<sup>366</sup> vgl. auch Kapitel 2.2.3

- Generische Relationen
- Hierarchische Relationen
- Benannte Relationen

Generische Relationen wurden verwendet, um Entwicklungsartefakte miteinander in Beziehung zu setzen, die Art der Beziehung wurde nicht weiter spezifiziert.

Eine hierarchische Relation wird dazu verwendet um auszudrücken, ob ein Entwicklungsartefakt als Teil/Untermenge eines anderen Entwicklungsartefakts gesehen werden kann.

**Beispiel:** Bei Anfertigung eines Lastenhefts in mehreren Teilprojekten wurde ein „Zentraldokument“ erstellt, welches alle übergreifenden Inhalte definiert und für alle „Filialdokumente“ gültig ist.

Benannte Relationen stellen einen definierten Zusammenhang zwischen Entwicklungsartefakten her, wobei die Art der Relation individuell benannt wird (Beispiel siehe Abbildung 54).

#### 1.4 Mitgeltende Dokumente

- Länderübersicht der Dienste (2014-03-03\_Masterdiensteliste\_G2Länderübersicht.xlsx)
- Bedienspezifikation (MIB\_07\_OnlineDienste\_v006.pdf)

Abbildung 54: Ausschnitt aus einem Lastenheft – Kapitel „Mitgeltende Dokumente“

#### Implizite Relationen zwischen Entwicklungsartefakten

Neben einer Menge von explizit abgebildeten Relationen in den Artefakten konnte eine Menge von impliziten Relationen zwischen Entwicklungsartefakten identifiziert werden, welche die Entwickler aus vorhandenen Informationen ableiten und die nicht expliziert sind. Implizite Relationen werden folglich im Kopf der Entwickler gebildet und stellen ein wichtiges Instrument beim Auffinden von Informationen dar. Folgende Typen von impliziten Relationen zwischen Entwicklungsartefakten konnten identifiziert werden:

- Implizite Relationen über Baustruktur/Systemarchitektur
- Implizite Relationen über Funktionsstruktur
- Implizite Relationen über Stakeholder-Beziehungen
- Implizite Relationen über den Dokumententyp

Es konnte beobachtet werden, dass implizit eine Relation zwischen Entwicklungsartefakten hergestellt wird, wenn eine Baustruktur oder Systemarchitektur bekannt ist und die Dokumente Informationen darüber enthalten, für welche Systeme die Inhalte gelten.

Weiter konnte beobachtet werden das implizite Relationen hergestellt werden, sofern sich Inhalte auf gleiche Funktionen oder auf Funktionen, die in Abhängigkeit zueinander stehen, beziehen (Beispiel: Nennung der Funktion „Nachrichten“ in zwei unterschiedlichen Entwicklungsartefakten lässt auf einen Zusammenhang schließen).

Implizite Relationen sind auch möglich, wenn sich Autoren oder Interessierte an einem Dokument gleichen oder zwischen diesen Abhängigkeiten bestehen.

Neben impliziten Relationen aus Inhalten des Dokuments kann auch aus dem Typ des Entwicklungsartefakts abgeleitet werden, welche Inhalte zu erwarten sind. Folgende Typen von Entwicklungsartefakten wurden identifiziert:

- Funktionsliste: enthält Funktionen
- Anforderungsdefinition: enthält Ziele, Anforderungen und/oder Anwendungsfälle
- Spezifikation: enthält Gestalt/Implementation
- Projektplan: enthält Meilensteine, Phasen und Aktivitäten
- Testspezifikation: enthält Testfälle, Testdokumentationen
- Systemdokumentation: enthält Gestalt/Implementation

### 5.3.3 Implikationen

Der Dokumentationsform der Entwicklungsartefakte, welche Ziele und Anforderungen beinhaltet konnte ein nutzerspezifischer Zweck zugeordnet werden. Die im Stand der Forschung bereits genannte weite Verbreitung von Microsoft Office (vgl. Kapitel 2.2.3) konnte mit den Untersuchungen bestätigt werden. Bei dieser Dokumentationsform fehlt die Explikation der für ein vollständiges Zielsystem notwendigen Relationen. Eine Ablösung eines Tools (z.B. durch einen Modellbasierten Ansatz) kann sich jedoch in Einzelfällen als aufwändig oder auf Grund von Rahmenbedingungen (vgl. Kapitel 5.2) sogar als unmöglich darstellen.

Obwohl die untersuchten Entwicklungsartefakte an viele Stakeholder verteilt werden müssen, werden Begründungen zur Herkunft oder Aufbau von Zielen und Anforderungen oder Änderungen nur sehr selten dokumentiert. Dadurch ist eine Nachvollziehbarkeit im verteilten Entwicklungsumfeld nur schwer möglich.

Häufig werden mehrere Partialmodelle in einem Dokument expliziert. Eine mögliche Erklärung ist, dass sich Aufbau und Inhalt von Entwicklungsartefakten häufig daran orientieren, welche Informationen die Basis für eine Entscheidung darstellen. Dabei werden selten nur Elemente aus einem Partialmodell herangezogen, sondern immer auch der aktuelle Wissensstand oder Abhängigkeiten zu anderen Partialmodellen. Entscheidungen werden häufig dezentral in den Dokumenten festgehalten, wobei Inkonsistenzen festgestellt wurden, welche zu vermeiden sind.

Es stellte sich weiter heraus, dass die Dokumente unterschiedliche Detaillierungsebenen umfassen, woraus Relationen zwischen Entwicklungsartefakten resultieren.

Entwickler nutzen eine Vielzahl impliziter und expliziter Relationen zur Verlinkung von Entwicklungsartefakten. Die System-Struktur ist dabei wesentlich beim Auffinden von Informationen. Artefakte enthalten Elemente. Elemente können mit anderen Elementen in Beziehung stehen (vgl. z.B. Ebel). Ganze Artefakte können aber auch mit anderen Artefakten in Beziehung stehen.

Eine manuelle Historisierung von Anforderungsdokumenten wird von den Nutzern häufig nicht gepflegt, was eine Nachverfolgbarkeit über die Zeit erschwert.

Die im Rahmen der Untersuchungen gesammelten Erkenntnisse beruhen zwar auf Dokumenten, welche aus nur einem Umfeld stammen. Eine Übertragbarkeit der Erkenntnis wird auf Grund der Größe und Verteilung des Projekts und der Vielzahl an beteiligten Stakeholdern aus unterschiedlichen Bereichen und Disziplinen dennoch angenommen.

#### **5.4 Verständnis über das Management von Änderungen in einem verteilten Umfeld**

Änderungen treten in den betrachteten Entwicklungsszenarien sehr häufig auf. Wann von einer Änderung gesprochen und wie mit diesen umgegangen wird ist jedoch in der Praxis häufig unklar. Daher soll mit den folgenden Untersuchungen ein Verständnis vom Management von Änderungen in einem verteilten Umfeld geschaffen werden. Ziel der Untersuchung ist insbesondere die Beantwortung der folgenden Forschungsfragen:

##### **Forschungsfrage 1.4: Management von Änderungen bei SoS**

Welche Anforderungen werden an das Management von Änderungen bei SoS gestellt?

##### **5.4.1 Studiendesign**

Zur Beantwortung der definierten Forschungsfragen wurde eine Studie im Rahmen eines regelmäßig tagenden Arbeitskreises durchgeführt, an dem 6 Industrievertreter eines Automobilherstellers, zwei Dienstleistern und einer Unternehmensberatung teilgenommen haben, die sich mit dem Themengebiet des Management von Änderungen im Umfeld Connected Car beschäftigten. Die Untersuchungen erfolgten im Sommer 2015. Daten wurden im Rahmen von Gruppendiskussionen zu vorgegebenen Fragestellungen erhoben.



Eine Gruppendiskussion erlaubt die freie Interaktion der Gruppenmitglieder zu einem gestellten Thema. Die Diskussion wird höchstens Ausnahmsweise vom Forscher durch Fragen beeinflusst. Indem die Teilnehmer untereinander interagieren, werden möglicherweise Auffassungen und Normen manifest, die nicht als Reaktion auf Stimuli von Außen entstehen.<sup>367</sup>

Ein wesentliches Ziel der Methode ist es, die spezifischen Sichtweisen und den Sprachgebrauch der Teilnehmer zu explorieren.<sup>368</sup> Es wird davon ausgegangen, dass die Methode besonders gut zur Beantwortung der definierten Fragestellungen nutzbar ist, weil sich hier Deutungsmuster und Handlungsorientierungen von Akteuren in realen sozialen Prozessen interaktiv entfalten können.<sup>369</sup>

#### **5.4.2 Ergebnisse der Gruppendiskussionen**

Im Folgenden werden die Ergebnisse der Gruppendiskussionen erläutert und interpretiert.

##### **Abgrenzung des Management von Änderungen**

Im Rahmen der durchgeführten Gruppendiskussionen wurde zunächst ein gemeinsames Verständnis davon geschaffen, wann von Änderungsmanagement gesprochen wird. Änderungen sind nach Ansicht der Experten immer dann möglich, sobald sich eine Gruppe von Mitarbeitern oder ein Gremium bereits auf einen bestimmten Soll-Zustand festgelegt hat. Dieser abgestimmte Soll-Zustand wurde in der Gruppe als „Nulllinie“ bezeichnet (vgl. Abbildung 55). Häufig ist dieser Zeitpunkt mit gewissen Meilensteinen im Entwicklungsprozess verknüpft (vgl. In-Stage und Cross-Gate Iterationen). Jedoch müssen Entscheidungen häufig auch abseits der festgelegten Entwicklungsmeilensteine getroffen werden, was ein Management von Änderungen erforderlich macht. Ein Entscheidungsgremium kann sich dabei aus Vertretern unterschiedlicher Ressorts oder Unternehmen zusammensetzen. Von einer Änderung wird folglich nicht gesprochen, wenn noch keine abgestimmte „Nulllinie“ existiert, sondern an dieser im Rahmen von Iterationen gearbeitet wird (z.B. Status „in Arbeit“ bei einem Lastenheft). Änderungen sind folglich in allen Phasen der Entwicklung möglich.

Nicht von Änderungsmanagement wird nach Ansicht der Experten gesprochen, wenn eine Änderung auf Grund eines Fehlers durchgeführt werden muss (z.B. wenn ein

---

<sup>367</sup> vgl. Atteslander 2010

<sup>368</sup> vgl. Stewart & Shamdasani 1998

<sup>369</sup> vgl. Kelle, Tobor & Metje 2007

Lieferant eine Funktion nicht wie spezifiziert umgesetzt hat). Fehler müssen im Rahmen eines Fehlermanagements verwaltet und gelöst werden.

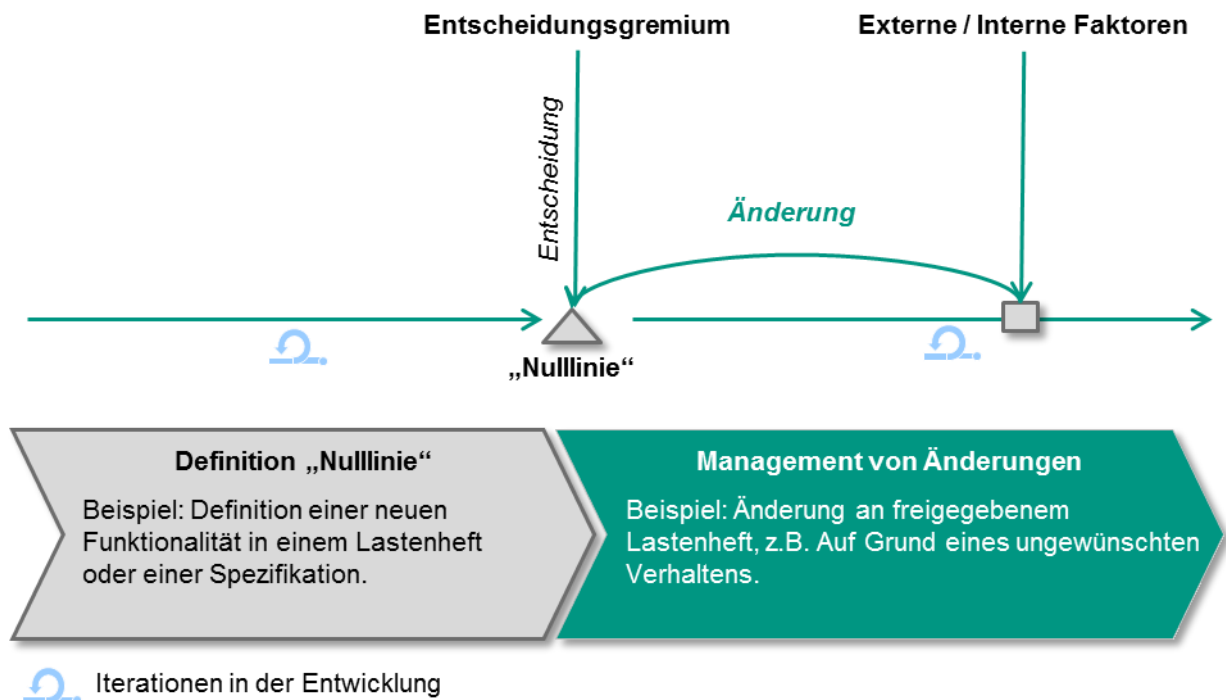


Abbildung 55: Abgrenzung des Managements von Änderungen

## Anforderungen an das Management von Änderungen in einem verteilten Umfeld

Folgende Anforderungen und Voraussetzungen wurden in der Gruppe an ein Management von Änderungen in einem verteilten Connected Car Umfeld gestellt

- Integration bestehender Prozesse, Tools und Gremien:**  
Die rein für Fahrzeug oder IT-Komponenten genutzten Prozesse, Tools und Gremien im Kontext Änderungsmanagement sollen bestehen bleiben. Die Freigabekriterien beteiligter Komponenten sind einzuhalten.
- Dynamisch zusammengesetztes Entscheidungsgremium:**  
Eine Änderung sollte prinzipiell durch Jeden beantragt werden können („Antragssteller“). Wer über eine Änderung entscheiden muss, ist abhängig davon, wer von der Änderung betroffen ist. Ein entsprechendes Entscheidungsgremium muss sich daher dynamisch und abhängig von der Änderung zusammensetzen. Der Änderungsantrag muss durch einen Ende-zu-Ende Funktionsverantwortlichen (mit Unterstützung des Änderungsantragssteller) vorangetrieben werden.
- Identifikation betroffener Systeme:**  
Jeder Änderungsantrag muss eine Funktionszerteilung auf betroffene Komponenten beinhalten. Entsprechend der Funktionszerteilung erfolgt die Bewertung der Änderungen durch die beteiligten Stakeholder.

- **Identifikation betroffener Module und Baukästen:**  
Änderungen im Rahmen einer Produktgenerationsentwicklung können Baukästen oder Modulen betreffen. Das Management von Änderungen muss daher über Produktgenerationen hinweg konsistent gehalten werden. Abhängigkeiten und Auswirkungen zu Nutzern des Baukastens müssen transparent gemacht werden.
- **Identifikation betroffener Märkte:**  
Änderungen können abhängig vom jeweiligen Markt sein. Betroffene Märkte müssen daher im Rahmen der Bewertung der Änderung kenntlich gemacht werden.
- **Verlinkung der betroffenen Entwicklungsartefakte („Nulllinie“):**  
Wesentlich für die Bewertung einer Änderung ist die Möglichkeit, betroffene Systeme und deren Entwicklungsartefakte zu identifizieren. Notwendige Voraussetzung für das Management von Änderungen ist folglich eine Verlinkung der (bestehenden) Entwicklungsartefakte beziehungsweise der Elemente der Partialsysteme.
- **Kommunikation und Konsolidierung der Bewertung der Änderung:**  
Änderungen müssen an alle Betroffenen kommuniziert und durch diese bewertet werden. Bei einer getrennten Bewertung durch unterschiedliche Verantwortliche muss die Bewertung konsolidiert und übergreifend bewertet werden, um Konflikte auszuschließen. Kosten, Aufwand und benötigte Ressourcen für die Umsetzung müssen übergreifend transparent gemacht werden.
- **Nachverfolgbarkeit von Änderungen:**  
Die Nachverfolgbarkeit von Änderungen muss über die Zeit über alle involvierten Parteien gewährleistet sein.
- **Zeit der Bearbeitung einer Änderung:**  
Die Bearbeitung einer Änderung muss in einem von allen Betroffenen akzeptierten zeitlichen Rahmen erfolgen. Hierfür muss im Vorfeld eine Zeitschiene zur Bewertung, Entscheidung und Umsetzung definiert werden.
- **Vermeidung von Duplikaten:**  
(Semantische) Duplikate müssen ausgeschlossen bzw. aufgezeigt werden.

### 5.4.3 Implikationen

Mit den durchgeführten Untersuchungen konnte zunächst ein Verständnis davon geschaffen werden, wann man von einem Management von Änderungen spricht. Für die Definition einer Änderung ist eine Verlinkung von Entwicklungsartefakten notwendig, um eine „Nulllinie“ über alle involvierten Parteien zu definieren. Die Notwendigkeit zur Verlinkung von Entwicklungsartefakten als wichtige Grundlage für

das Management von Änderungen konnte durch die Untersuchungen damit bestätigt werden.

Um Auswirkungen einer Änderung in einem verteilten Entwicklungsteam bewerten zu können ist Wissen über die Bau- oder Funktionsstruktur sowie entsprechender Verantwortlichkeiten erforderlich, welches den Beteiligten verfügbar gemacht werden muss. Eine Verlinkung von Bau- oder Funktionsstruktur mit Entwicklungsartefakten, welche Zielsystemelemente explizieren, ist folglich erforderlich.

Um auch Prozessual Durchgängigkeit herzustellen müssen Methoden und Prozesse in den verteilten Entwicklungsteams in einem übergreifenden und koordinierenden Management von Änderungen integriert werden.

Im Kontext von SoS werden Änderungen nicht zwangsläufig an alle Betroffenen kommuniziert, sondern müssen durch die Verantwortlichen beobachtet werden. Die Bewertung von Änderungen kann dabei nicht starr durch eine Organisationseinheit erfolgen, sondern erfordert vielmehr ein fallspezifisches Problemlösungsteam. Wer in einem Entscheidungsgremium teilnehmen muss, ergibt sich aus den Verantwortlichkeiten. Aus den Aussagen der Experten konnten vier generische Rollen für die Teilnahme an einem Entscheidungsgremium für das Management von Änderungen identifiziert werden (vgl. Abbildung 56), welche abhängig von der Änderung besetzt werden müssen.

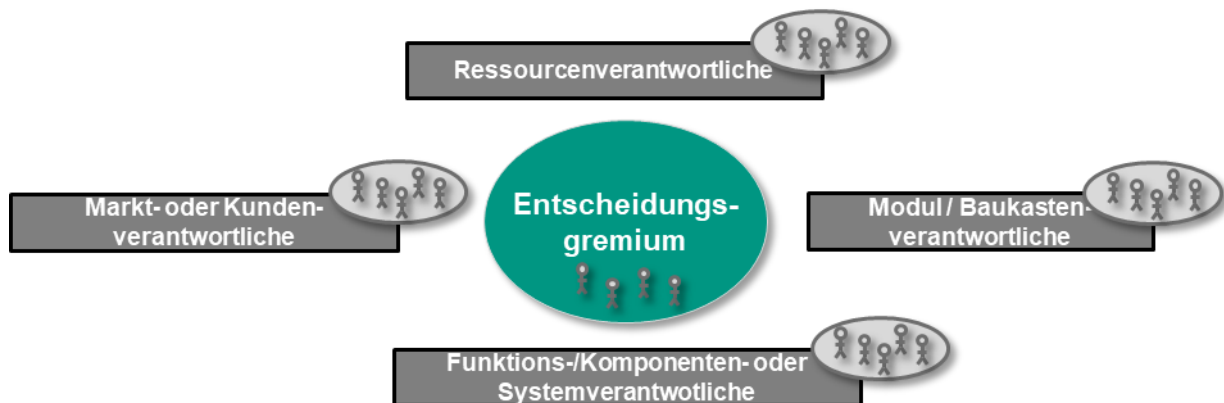


Abbildung 56: Dynamische Zusammensetzung eines Entscheidungsgremiums für das Management von Änderungen

## 6 Durchgängige Entwicklung von Zielsystemen

In diesem Kapitel soll die Forschungsfrage beantwortet werden, wie ein Ansatz gestaltet werden muss, um die durchgängige Entwicklung von Zielsystemen, insbesondere in einem verteilten Umfeld, zu unterstützen. Dabei werden die Erkenntnisse aus dem Stand der Forschung und der ersten deskriptiven Studie aufgegriffen, um einen Ansatz zu entwickeln.

Die Ergebnisse der deskriptiven Studie I haben gezeigt, dass in einem verteilten Umfeld eine Vielzahl partieller explizierter Zielsysteme entsteht. Relationen werden häufig dezentral in den Entwicklungsartefakten dokumentiert, wodurch Abhängigkeiten nicht allen Beteiligten transparent gemacht werden. Es wurde weiter festgestellt, dass Entwickler auch Relationen zwischen Entwicklungsartefakten implizit aus vorhandenem Wissen bilden, dieses Wissen aber Anderen nicht immer zugänglich machen. Änderungen können dabei sehr häufig auftreten, wodurch es zu hohem Abstimmungsaufwand und einem Verlust von Durchgängigkeit und Konsistenz kommen kann. Bei der Dokumentation und dem Management von Änderungen können unterschiedliche Methoden und Tools verwendet werden, wobei eine Toolauswahl immer unter einer Menge von Randbedingungen getroffen wird.<sup>370</sup> Bei einem SoS steht die Vereinheitlichung von Methoden und Tools zur Dokumentation von Anforderungen und Zielen zusätzlich unter erschwerten Bedingungen, da eine Vielzahl von Organisationseinheiten, häufig auch aus unterschiedlichen Branchen, beteiligt sind. Abgeleitet aus den Erkenntnissen der ersten deskriptiven Studie werden zusammenfassend vier Ziele an einen Ansatz zur Unterstützung der durchgängigen Entwicklung und Dokumentation von Zielsystemen abgeleitet:

**Z-1: *Integration und nicht Ablösung bestehender Methoden, Prozesse und Tools in den verteilten Entwicklungsteams***

**Z-2: *Durchgängigkeit im Zielsystem muss über den Detaillierungsgrad, Produkt- und Entwicklungsgenerationen und unterschiedliche Projekte auch bei unterschiedlichen Produktgenerations- und Entwicklungszyklen hergestellt werden***

---

<sup>370</sup> vgl. Kapitel 5.3.2

**Z-3:** Verwendung von **Metainformationen** bei Entwicklungsartefakten:

- a. zur Abbildung von Relationen bei heterogenen Entwicklungsartefakten
- b. als Basis für eine Ableitung von impliziten Relationen zwischen Entwicklungsartefakten

**Z-4:** Kombination aus **zentralem und dezentralem Management von Änderungen** mit dynamischer Zusammensetzung der Problemlösungsteams und Entscheidungsgremien

Aus diesen Zielen lassen sich drei weitere Forschungsfragen ableiten (vgl. Abbildung 57), welche in den folgenden Kapiteln beantwortet werden.

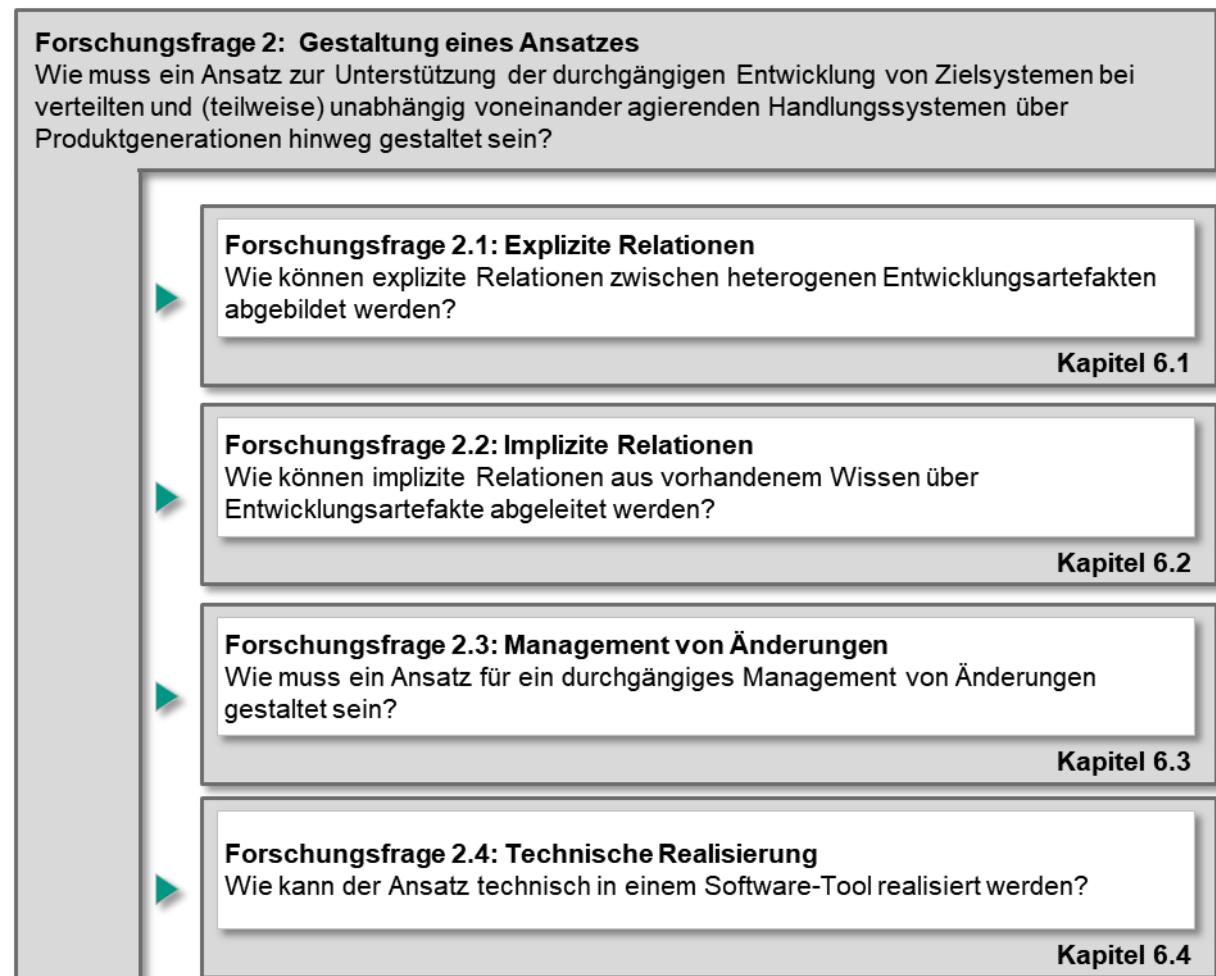


Abbildung 57: Forschungsfragen der Präskriptiven Studie

## 6.1 Abbildung von Relationen bei heterogenen Entwicklungsartefakten

Zunächst soll die Forschungsfrage beantwortet werden, wie explizite Relationen zwischen heterogenen und verteilten Entwicklungsartefakten mit semantischen Metainformationen abgebildet werden können.<sup>371</sup>

### 6.1.1 Verwendung semantischer Metainformationen

Modellbasierte Vorgehensweisen wurden insbesondere dafür entwickelt, eine durchgängige Entwicklung von der Anforderung bis zur Validierung zu ermöglichen. Diese bieten daher hervorragende Eigenschaften für eine „zoomende“ Arbeitsweise.<sup>372</sup> Modellbasierte Methoden eignen sich auch sehr gut, um in einem Entwicklungsteam oder in einer bestimmten Entwicklungsphase Zielsysteme durchgängig über mehrere Produktgenerationen abzubilden.<sup>373</sup> Diese geben jedoch bereits eine Dokumentationsform vor, welche abhängig vom nutzerspezifischen Zweck in einer Entwicklungsphase nicht immer geeignet ist. Die teilweise noch immer umstrittene Akzeptanz und vergleichsweise geringe Verbreitung in der Praxis im Vergleich zu Microsoft Office Tools<sup>374</sup> scheinen dies zu unterstreichen. Ein Ansatz zur durchgängigen Abbildung von verteilten partiellen Zielsystemen über alle Entwicklungsphasen muss folglich von der Dokumentationsform (z.B. SysML-Modell, Word-Dokument, Doors) abstrahieren.

Aus diesem Grund wird für die Abbildung des Zielsystems ein Artefakt-orientierter Ansatz gewählt, welcher von Inhalt, Struktur und Repräsentationsform abstrahiert. Die Herausforderungen bei der Herstellung von Durchgängigkeit bei einem Artefakt-orientierten Ansatz liegen in der Abbildung von Abhängigkeiten zwischen den heterogenen Entwicklungsartefakten. Für die Verlinkung der Dokumente gibt es unterschiedliche Möglichkeiten. Insbesondere moderne Dokumenten Management Systeme wie Microsoft Sharepoint oder Alfresco bieten bereits Funktionen, um Dokumente zu verwalten und miteinander in Beziehung zu setzen<sup>375</sup>. Diese bieten jedoch keine Möglichkeit, Relationen zwischen Entwicklungsartefakten mit zusätzlichem Wissen aus der Produktentwicklung anzureichern, wodurch die Möglichkeiten zur Herstellung von Durchgängigkeit in den geforderten Dimensionen unzureichend sind. Durch Verwendung semantischer Metainformationen kann ein

---

<sup>371</sup> Die in Kapitel 6.1 dargestellten Untersuchungen sind im Rahmen der Publikation Albers et al. 2017 veröffentlicht worden.

<sup>372</sup> vgl. auch Kapitel 2.2.3

<sup>373</sup> vgl. u.a. Zingel 2013, Ebel 2015, Bursac 2016 und Scherer 2016

<sup>374</sup> vgl. auch Kapitel 2.2.3

<sup>375</sup> vgl. auch Kapitel 2.2.3

inhaltlicher Zusammenhang zwischen Entwicklungsartefakten transparent und damit Durchgängigkeit und Konsistenz im Zielsystem über die folgenden Dimensionen hergestellt werden (vgl. Abbildung 58):

- Detaillierungsgrad (insbesondere Verlinkung von SoS- und Komponentensystemebene)
- Produkt- und Entwicklungsgenerationen ( $G_n, E_{gn}$ )
- unterschiedliche Projekte

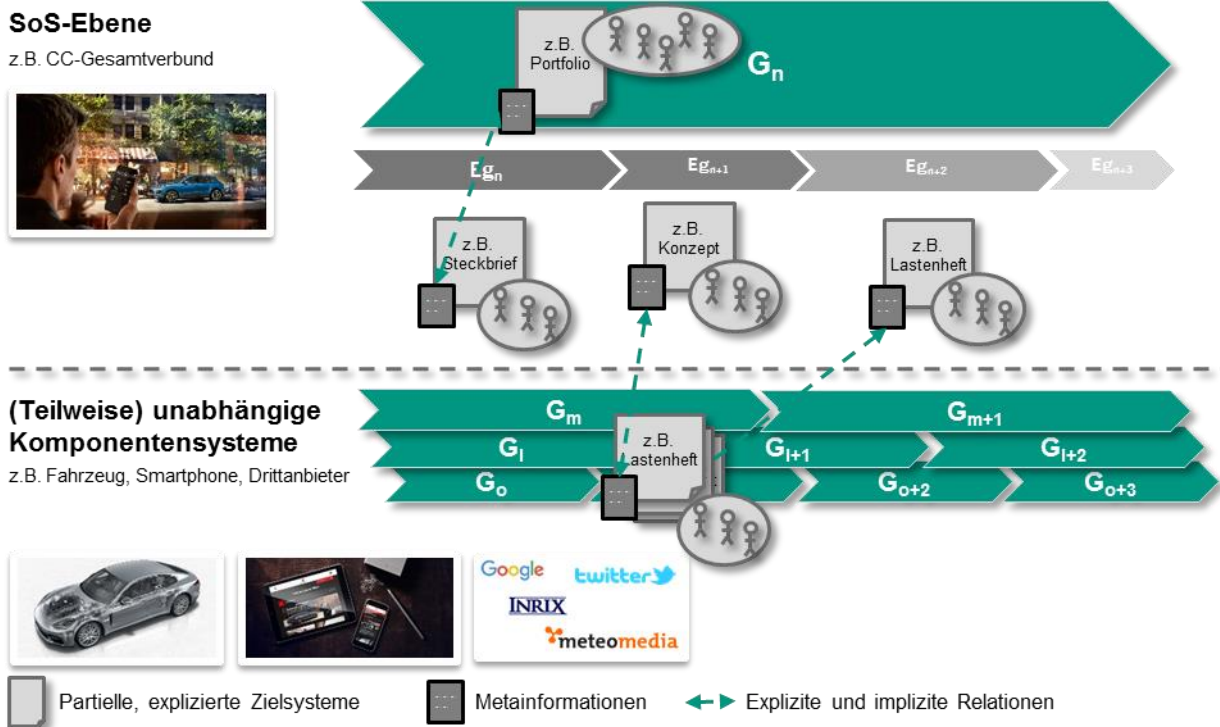


Abbildung 58: Herstellung von Durchgängigkeit durch Verwendung von Metainformationen

Semantische Technologien bieten Eigenschaften, Daten zu strukturieren, selbst wenn sich die zugrunde liegenden Schemata der Daten unterscheiden (vgl. auch Kapitel 2.3). Dadurch ist dieser Ansatz für den betrachteten Entwicklungskontext sehr gut geeignet. Mit Metainformationen kann Wissen aus Objekten exploriert und verlinkt werden<sup>376</sup>. Es wird daher erwartet, dass durch Verwendung von semantischen Technologien ein entscheidender Mehrwert bei der Herstellung von Durchgängigkeit in einem verteilten Entwicklungsumfeld erreicht wird, der den Mehraufwand in der Pflege zu rechtfertigen vermag. Verschiedene Arbeiten haben die potenzielle semantischer Funktionalitäten für die Abbildung von Zielsystemen bereits grundsätzlich bestätigt<sup>377</sup>.

<sup>376</sup> vgl. z.B. auch Schütt 2007, S.11

<sup>377</sup> vgl. auch Kapitel 2.3



Um implizite Relationen aus Metainformationen ableiten zu können ist eine grundsätzliche Struktur notwendig. Ontologien ermöglichen die semantisch reichhaltige Strukturierung von Metainformationen. Das Resource Description Framework (RDF) bietet bereits eine gute Basis zur Beschreibung von Ressourcen mit Metainformationen. Die Verknüpfungs-Struktur von RDF bildet einen gerichteten, markierten Graphen, wo die Kanten eine benannte Verbindung zwischen zwei Ressourcen (Knoten) darstellt, wobei in unserem Anwendungsfall eine Ressource sowohl ein Entwicklungsartefakt als auch ein System oder ein Stakeholder sein kann. Neben der Darstellung als Graph lässt sich RDF auch in serialisierter Form durch Zeichenketten darstellen, welche maschinenlesbar sind. Damit sind logische Ableitungen (sogenannte Inferenzen) durch Software möglich. Mit RDF Schema (RDFS) bietet RDF zusätzlich die Möglichkeit, Datensätze auch mit einfachem, terminologischem Wissen bzw. Schemawissen zu versehen und zu erweitern.<sup>378</sup>

Nach EBEL darf eine zugrundeliegende Ontologie dem Entwickler nicht in einer abstrakten Form präsentiert werden, sondern muss auf Basis einer geschlossenen Liste relevanter Attribute und Relationen zu Verfügung gestellt werden.<sup>379</sup> Im Folgenden wird eine Ontologie durch ein RDF Schema definiert. Die verwendeten Attribute und Relationen beruhen hierbei zum einen auf dem aktuellen Stand der Forschung, zum anderen auf Erkenntnissen der deskriptiven Studie I.

Eine vollständige Liste aller im RDF Schema beinhalteten Relationen und Klassen ist im Anhang dargestellt.

---

<sup>378</sup> vgl. Dengel 2012 S.124

<sup>379</sup> vgl. Ebel 2015

### 6.1.2 Eindeutige Referenzierung von Entwicklungsartefakten

In diesem Kapitel werden Attribute beschrieben, um ein Artefakt mit Metainformationen eindeutig zu beschreiben. Dazu werden 7 Attribute vorgeschlagen (vgl. Abbildung 59).

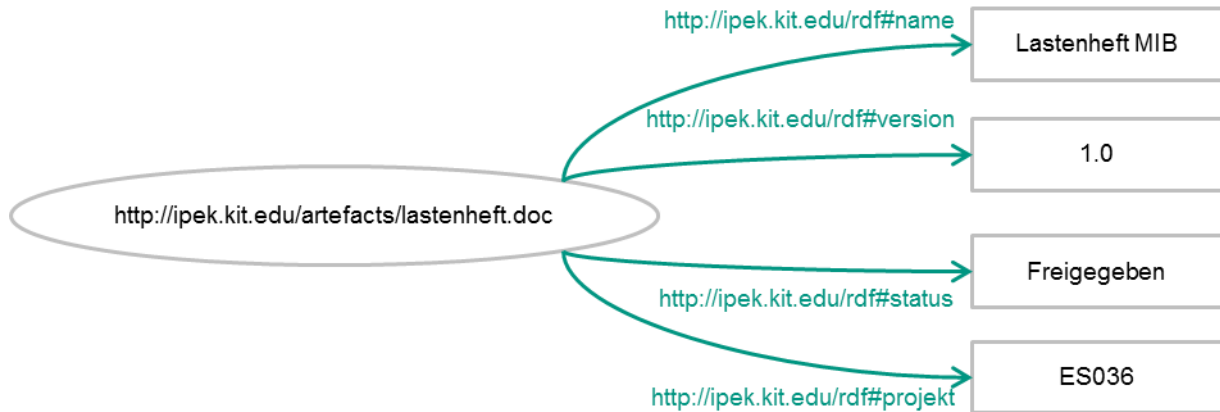


Abbildung 59: Beispiel für RDFS Klasse Artefakt

#### ***URI : Literal***

Der Uniform Resource Identifier ist ein eindeutiger Bezeichner, um Ressourcen zu identifizieren.

#### ***Name des Artefakts : Literal***

Die Untersuchungen im Rahmen der deskriptiven Studie haben ergeben, dass der Name eines Dokuments noch immer am häufigsten benutzt wird, um Informationen aufzufinden. Daher soll dem Benutzer hiermit die Möglichkeit gegeben werden, seinem Artefakt einen Namen zu geben.

#### ***Lokation: Literal***

Die Lokation bildet den Speicherort des Artefakts ab (z.B. Datei auf einem Netzlaufwerk).

#### ***Versionsnummer : Literal***

Versionen sind notwendig, um einen Arbeitsstand eines Artefakts eindeutig zu referenzieren. Zur Pflege von Versionen bei einem Artefakt empfiehlt sich ein zweistufiges Vorgehen, basierend auf Best Practices von der Versionierung von Dokumenten<sup>380</sup>.

<sup>380</sup> vgl. auch Rochfort 2015

### **Status : List <Literal>**

Der Status eines Artefakts stellt eine zweckdienliche Vereinigung des Reife- und Härtegrades als Beurteilungsdimensionen analog Zielen und Anforderungen dar<sup>381</sup>. Der hier vorgestellte Status orientiert sich an in der Praxis etablierten Freigabestatus, die im Rahmen des Freigabemanagements von PDM-Systemen zum Einsatz kommen<sup>382</sup>. Hierbei orientieren sich die jeweiligen Status-Ausprägungen am Lebenszyklus des bezeichnenden Artefakts<sup>383</sup>:

- **Geplant:** Der Lebenszyklus und damit Status eines jeden Elements beginnt mit der Planung seiner Anfertigung. Hiermit wird lediglich die Intention zum Ausdruck gebracht, dass ein neues Artefakt erstellt werden soll.
- **In Arbeit:** In diesem Status nimmt ein Element noch keinen Einfluss auf andere Elemente bzw. den Produktentstehungsprozess und kann damit beliebig (oft) geändert werden.
- **Vorgelegt:** Elemente des Ziel- und Objektsystems werden in einem nächsten Schritt eingefroren. Hierdurch wird einerseits signalisiert, dass (vorerst) keine weiteren Änderungen am betreffenden Element mehr vorgenommen werden und andererseits wird hierdurch der Freigabeprozess des Elements angestoßen.
- **Freigegeben:** Durch eine entsprechende Freigabe erlangt ein Element den Status Freigegeben, wodurch es auf Basis einer (begründeten) Entscheidung fortan zu berücksichtigen ist und nicht mehr geändert werden darf. Abhängig vom Freigabeprozess kann eine Freigabe auf mehreren Ebenen erfolgen.
- **Verworfen:** Elemente, die keine Gültigkeit (mehr) besitzen und deren Inhalt und Status sich voraussichtlich nicht mehr ändern wird, erhalten den Status Zurückgezogen und sind nur noch zum Zwecke der Nachvollziehbarkeit zu berücksichtigen.

### **Erstellt / Geändert : Literal**

Das Erstellungs- bzw. Änderungsdatum soll dem Benutzer dabei unterstützen, einen zeitlichen Bezug zu Aktivitäten an diesem herzustellen.

---

<sup>381</sup> vgl. auch Ebel 2015

<sup>382</sup> vgl. Stockinger 2008, S. 15

<sup>383</sup> vgl. Stockinger 2008, S. 15

### 6.1.3 Relationen zwischen Entwicklungsartefakten

Neben der Referenzierung von Elementen einzelner Partialmodelle wurden auch Referenzen zwischen Artefakten abgebildet. Dabei konnten drei Typen identifiziert werden:

#### ***istTeilVon: Artefakt (Hierarchische Relation)***

Eine hierarchische Relation kann dazu verwendet werden um auszudrücken, ob ein Entwicklungsartefakt als Teil/Untermenge eines anderen Entwicklungsartefakts gesehen werden kann.

*Beispiel: Anfertigung eines übergreifenden „Zentraldokuments“, dessen Inhalte auch für alle „Filialdokumente“ gültig sind.*



Abbildung 60: Darstellung der Relation „istTeilVon“ im RDF Graph

#### ***istVerwandtMit: Artefakt (Generische Relation)***

Bei einer generischen Relation werden Entwicklungsartefakte miteinander in Beziehung gesetzt, ohne die Art der Beziehung weiter zu spezifizieren.

*Beispiel: Es ist bekannt oder wird vermutet, dass zwischen zwei Entwicklungsartefakten eine Relation besteht, die Art der Beziehung ist jedoch noch nicht bekannt.*



Abbildung 61: Darstellung der Relation „istVerwandtMit“ im RDF Graph

### 6.1.4 Relationen zu Partialmodellen der Produktentwicklung

Durch Relationen zu Elementen von Partialmodellen der Produktentwicklung soll durch die Metainformationen ein Rückschluss gezogen werden können, auf welche Partialmodelle der Produktentwicklung sich die Inhalte des Entwicklungsartefakts beziehen. Die Berücksichtigten Partialmodelle sind in Abbildung 62 dargestellt.

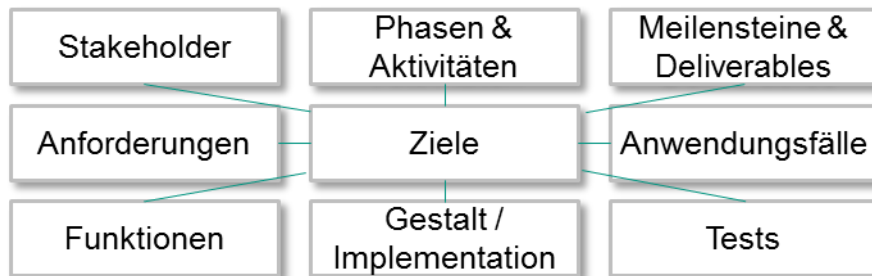


Abbildung 62: Partialmodelle der Produktentwicklung<sup>384</sup>

#### Relationen zu Stakeholdern

Personen, Gruppen oder Organisationen, die ein Interesse an Entwicklungsartefakten haben (Stakeholder) sollen explizit abgebildet werden können. Dabei kann unterschieden werden zwischen einem (verantwortlichen) Autor und einem reinen Interessenten.

Zur leichteren Zuordnung und Unterstützung der Kontaktaufnahme soll eine Person mit Name, Firmen- oder Abteilungszugehörigkeit und Kontaktdaten in den Metainformationen hinterlegt werden können.

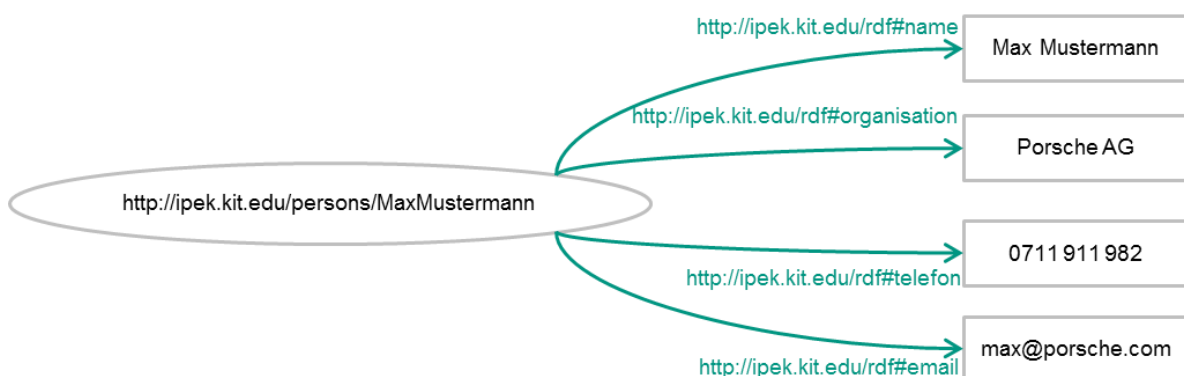


Abbildung 63: Beispiel für RDFS Klasse Person

<sup>384</sup> vgl. Ebel 2015, S. 152

**Verantwortlicher/Mitwirkende Autoren: Person**

Über das Attribut Verantwortlicher wird einem Artefakt der verantwortlichen Akteur (bzw. Stakeholder) zugeordnet. Der Verantwortliche gilt als Hauptansprechpartner für das Entwicklungsartefakt. Neben einem Verantwortlichen ist an der Erstellung eines Dokuments häufig noch eine Vielzahl von anderen Personen beteiligt. Diese Beziehung soll über das Attribut Mitwirkende abgebildet werden. Diese Information ist wichtig, um z.B. einen Person über Änderungen informieren zu können.

**Interessierte Personen: Person**

Neben den verantwortlichen Mitarbeitern wurde in den Dokumenten häufig ein Verteilerkreis gepflegt, welcher Mitarbeiter darstellt, die zwar nicht unmittelbar an der Erstellung beteiligt sind, jedoch an den Inhalten des Artefakts sowie an Änderungen dieser interessiert sind. Über das Attribut Interessierte Personen soll eine Liste mit Personen gepflegt werden können, welche sich für Änderungen an diesem Dokument interessieren.



Abbildung 64: Abbildung der Relation „interessiert“ im RDF Graph

### **Relationen zu Phasen, Aktivitäten & Meilensteinen**

Durch Relationen zu einer Entwicklungsphase, Aktivität oder einem Meilenstein soll das Entwicklungsartefakt zeitlich eingeordnet werden können.

#### ***entwicklungsphase: Entwicklungsphase / Aktivität***

Mit diesem Attribut kann ein Entwicklungsartefakt einer Entwicklungsphase oder einer Aktivität zugeordnet werden. Eine Entwicklungsphase oder Aktivität muss dafür als Ressource angelegt werden.

#### ***meilenstein: Meilenstein***

Meilensteine sind wichtige Anlaufpunkte in der Produktentwicklung um parallel laufende Entwicklungen zu synchronisieren.

#### ***folgtAuf: Entwicklungsphase / Aktivität / Meilenstein***

Mit der folgtAuf-Beziehung können Entwicklungsaktivitäten zueinander in Beziehung gesetzt werden.

### **Relationen zu Zielen, Anforderungen und Anwendungsfällen**

Durch diese Typen von Relationen kann deklariert werden, welche Ziele, Anforderungen oder Anwendungsfälle in Entwicklungsartefakten vorhanden sind und wie diese in Beziehung zueinander stehen.

#### ***beschreibtInhalteVon: Ziel, Anforderung, Anwendungsfall***

Mit der beschreibtInhalteVon-Relation kann ein Ziel, Anforderung oder Anwendungsfall mit einem Entwicklungsartefakt in Beziehung gesetzt werden.

#### ***istTeilVon: Ziel, Anforderung, Anwendungsfall***

Mit dieser Relation können Ziele, Anforderungen und Anwendungsfälle zueinander hierarchisch in Beziehung gesetzt werden.

### **Relationen zu Funktionen, Gestalt / Implementation und Tests**

Im Rahmen der deskriptiven Studie konnte beobachtet werden, dass die Struktur der Systemlandschaft häufig eine wichtige Information darüber liefert, welche Entwicklungsartefakte in Relation zueinander stehen. Daher soll eine Zuordnung von Entwicklungsartefakten zu einer Funktions- und/oder Baustruktur ermöglicht werden.

## System / Funktion

Ein System oder eine Funktion soll in den Metainformationen hinterlegt werden können. Dazu werden im Wesentlichen nur zwei Attribute hinterlegt:

- **Name:** Der eindeutige Name des Systems
- **Verantwortlich:** Die verantwortliche Person, welche das System entwickelt oder betreibt

## beschreibtInhalteVon : System / Funktion

Mit dieser Beziehung kann ein Zusammenhang von einem Entwicklungsartefakt zu einer Funktion oder einem System hergestellt werden.

*Beispiel: Ein Lastenheft beschreibt Anforderungen an ein bestimmtes System.*

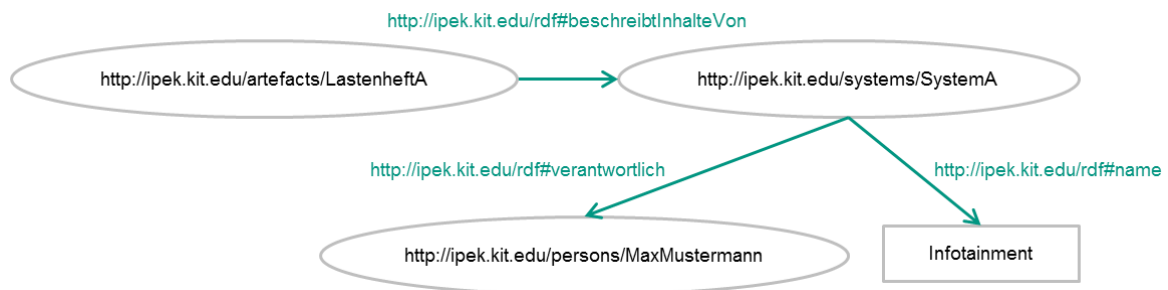


Abbildung 65: Darstellung der „beschreibtInhalteVon“ Relation im RDF Graph

## istTeilVon : System / Funktion (Hierarchische Relation)

Zur Abbildung einer Funktions- oder Baustruktur mit Metainformationen wird die hierarchische Relation um den Typ System/Funktion erweitert.

*Beispiel: System Infotainmentsystem ist ein Subsystem des Fahrzeugs.*



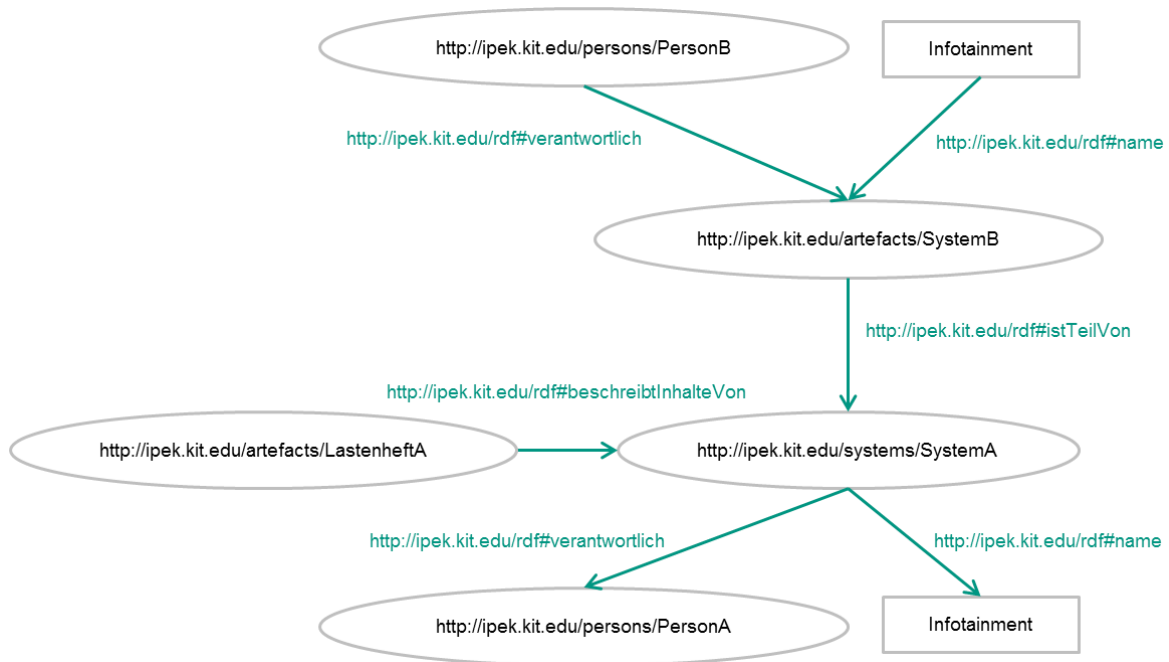


Abbildung 66: Darstellung der „istTeilVon“ Beziehung im RDF Graph

**verwendet: System / Funktion**

Neben einer hierarchischen Relation soll eine gerichtete Relation zwischen Funktionen und Systemen in den Metainformationen hergestellt werden können.

*Beispiel: Infotainmentsystem verwendet Schnittstelle vom System Fahrzeugbackend.*

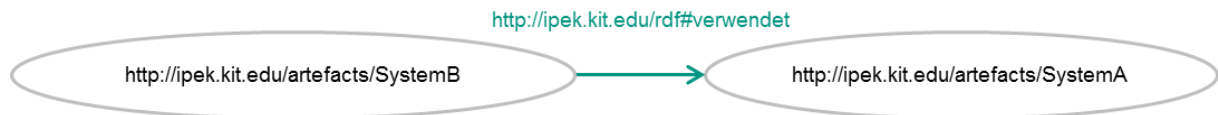


Abbildung 67: Darstellung der „verwendet“ Relation im RDF Graph

**6.1.5 Relationen zu Referenzmodellen einer Produktgenerationsentwicklung**

Entwicklungsartefakte oder Elemente eines Partialmodells basieren häufig auf bestehenden Referenzmodellen im Sinne der Produktgenerationsentwicklung<sup>385</sup>. Durch Verwendung von Metainformationen können diese Relationen transparent und für die Entwicklung nutzbar gemacht werden.

<sup>385</sup> vgl. Albers, Bursac & Wintergerst 2016

**entwicklungsphase: Produkt- oder Entwicklungsgeneration**

Durch das Attribut Entwicklungsphase soll ein Artefakte eindeutig einer oder mehreren Entwicklungs- oder Produktgenerationen zugeordnet werden können.

**basiertAuf: Partialmodell, Entwicklungsartefakt**

Durch die basiertAuf-Beziehung kann spezifiziert werden, ob ein Element aus einem Partialmodell oder einem anderen Entwicklungsartefakt im Sinne eines Referenzmodells basiert.

**variationsanteil: Literal**

Die basiertAuf-Beziehung kann durch Angabe des Variationsanteils des Referenzmodells erweitert werden.

**variationstyp: Literal**

Die basiertAuf-Beziehung kann durch Angabe des Variationstyps des Referenzmodells erweitert werden.

**herkunft: Literal**

Die basiertAuf-Beziehung kann durch Angabe der Herkunft des Referenzmodells erweitert werden.

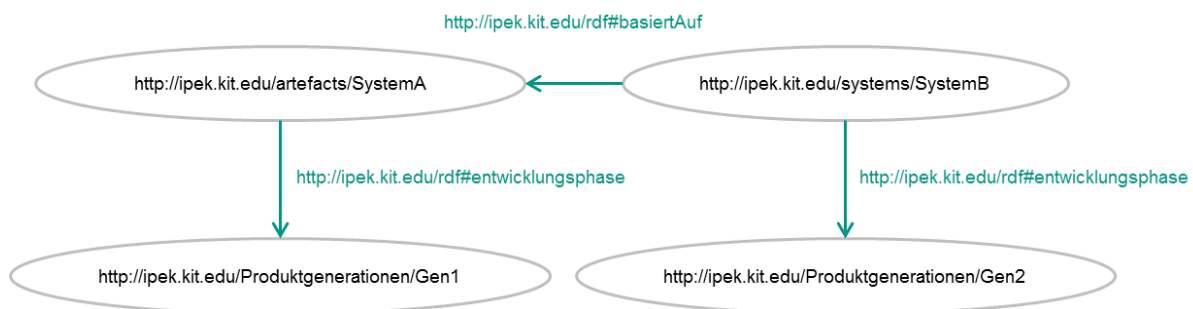


Abbildung 68: Darstellung der Relationen der Produktgenerationsentwicklung im RDF Graph

## 6.2 Ableitung impliziter Relationen aus vorhandenem Wissen

Aus den definierten RDF Attributen und Relationen lassen sich eine Menge von impliziten Relationen logisch ableiten (vgl. Abbildung 69). Hierfür werden im Folgenden Ableitungsregeln definiert, welche auf den Erkenntnissen der deskriptiven Studie I aufbauen.<sup>386</sup>

Die im Folgenden dargestellten Beispiele sind in der Sprache OWL formuliert.

*Beispiel:*

```
[rule:  
(?systemA ipek:istTeilVon ?systemB)  
(?artefaktA ipek:beschreibtInhalteVon ?systemA)  
->  
(?artefaktA ipek:beschreibtInhalteVon ?systemB)]
```

Dieses Beispiel kann wie folgt übersetzt werden:

„**WENN** System A ist ein Teil von System B **UND** Artefakt A beschreibt Inhalte von System A **DANN** Beschreibt Artefakt A auch System B.“

### 6.2.1 Implizite Relationen aus der Baustruktur

Eine (teilweise) bekannte Baustruktur (zum Beispiel auch aus vorherigen Produktgenerationen) kann dazu verwendet werden, um Relationen zwischen Entwicklungsartefakten abzuleiten. Dazu können sowohl eine hierarchische Struktur als auch (un-)gerichtete Relationen herangezogen werden.

#### Ableitungen aus hierarchischen Relationen der Baustruktur

Ist bekannt, dass ein System B ein Subsystem eines anderen Systems A ist und System A wird in einem Artefakt beschrieben, dann kann daraus abgeleitet werden das Änderungen an Artefakt A auch System B betreffen. Die zugehörige Regel in RDF sieht wie folgt aus:

*Beispiel 1:*

```
[rule1:  
(?systemA ipek:istTeilVon ?systemB)  
(?artefaktA ipek:beschreibtInhalteVon ?systemA)  
->  
(?artefaktA ipek:beschreibtInhalteVon ?systemB)]
```

Aus der systemischen Beziehung kann weiter eine Relation zwischen zwei Entwicklungsartefakten abgeleitet werden.

---

<sup>386</sup> Die in Kapitel 6.2 dargestellten Untersuchungen sind im Rahmen der Publikation Albers et al. 2017 veröffentlicht worden.

**Beispiel 2:**

```
[rule2:
(?systemB ipek:istTeilVon ?systemA)
(?artefaktA ipek:beschreibtInhalteVon ?systemA)
(?artefaktB ipek:beschreibtInhalteVon ?systemB)]
->
(?artefaktA ipek:istVerwandtMit ?artefaktB)]
```

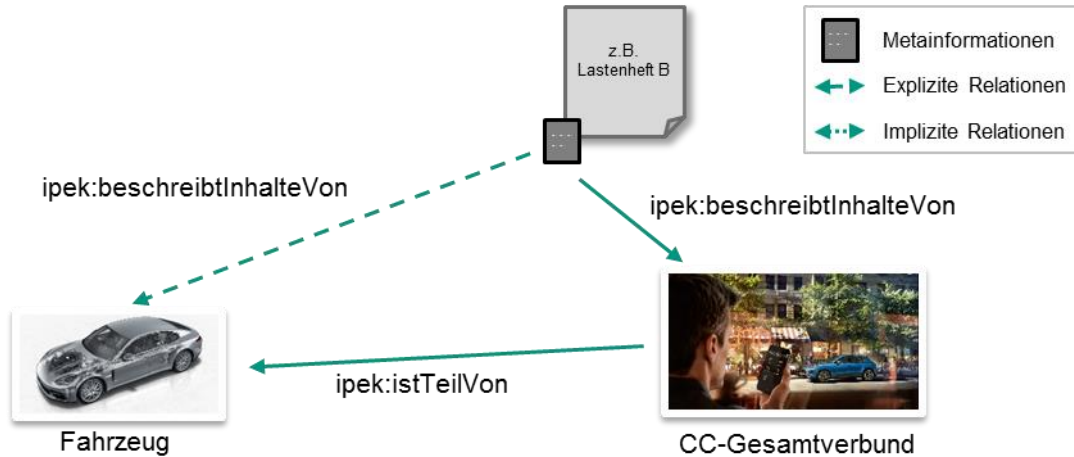


Abbildung 69: Ableitung impliziter Relationen aus hierarchischen Relationen

**Implizite Relationen aus generischen Relationen**

Diese Ableitung lässt sich ebenfalls bei einer generischen Relation treffen wenn lediglich bekannt ist, dass ein System ein anderes in irgendeiner Form verwendet.

**Beispiel 1:**

```
[rule3:
(?systemA ipek:verwendet ?systemB)
(?artefaktA ipek:beschreibtInhaltVon ?systemA)
(?artefaktB ipek:beschreibtInhaltVon ?systemB)]
->
(?artefaktA ipek:istVerwandtMit ?artefaktB)]
```

Werden in zwei unterschiedlichen Dokumenten auf die gleichen Systeme verwiesen, dann kann angenommen werden, dass ein Zusammenhang zwischen diesen Dokumenten besteht.

**Beispiel 2:**

```
[rule4:
(?artefaktA ipek:beschreibtInhaltVon ?systemA)
(?artefaktB ipek:beschreibtInhaltVon ?systemA)]
->
(?artefaktA ipek:istVerwandtMit ?artefaktB)]
```

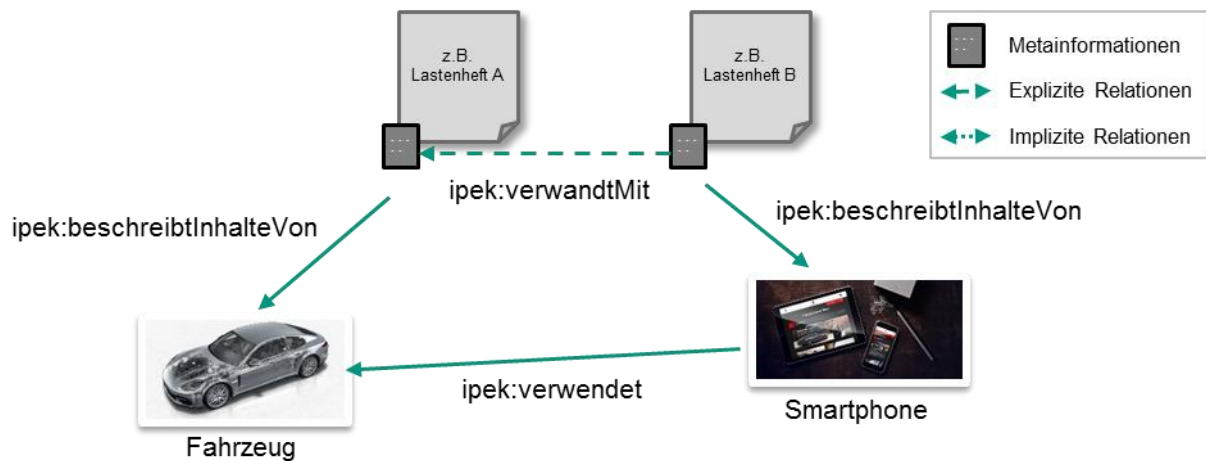


Abbildung 70: Ableitung impliziter Relationen aus generischen Relationen

### 6.2.2 Implizite Relationen aus der Funktionsstruktur

Implizite Relationen können auch hergestellt werden, sofern sich Inhalte auf gleiche Funktionen oder auf Funktionen, die in Abhängigkeit zueinander stehen, beziehen.

Zum Beispiel lässt die Nennung der Funktion „Nachrichten“ (oder einer Subfunktion) in zwei unterschiedlichen Entwicklungsartefakten auf einen Zusammenhang der Entwicklungsartefakte schließen.

Analog der Inferenz am Beispiel der Baustruktur lässt sich eine Ableitung bei Kenntnis einer hierarchischen oder (un-)gerichteten Funktionsstruktur durchführen.

#### Beispiel 1:

```
[rule1:
(?funktionB ipek:istTeilVon ?funktionA)
(?artefaktA ipek:beschreibtInhalteVon ?funktionA)
->
(?artefaktA ipek:beschreibtInhalteVon ?funktionB)]
```

#### Beispiel 2:

```
[rule2:
(?funktionA ipek:verwendet ?funktionB)
(?artefactA ipek:beschreibtInhaltVon ?funktionA)
(?artefactB ipek:beschreibtInhaltVon ?funktionB)]
->
(?artefactA ipek:istVerwandtMit ?artefactB)]
```

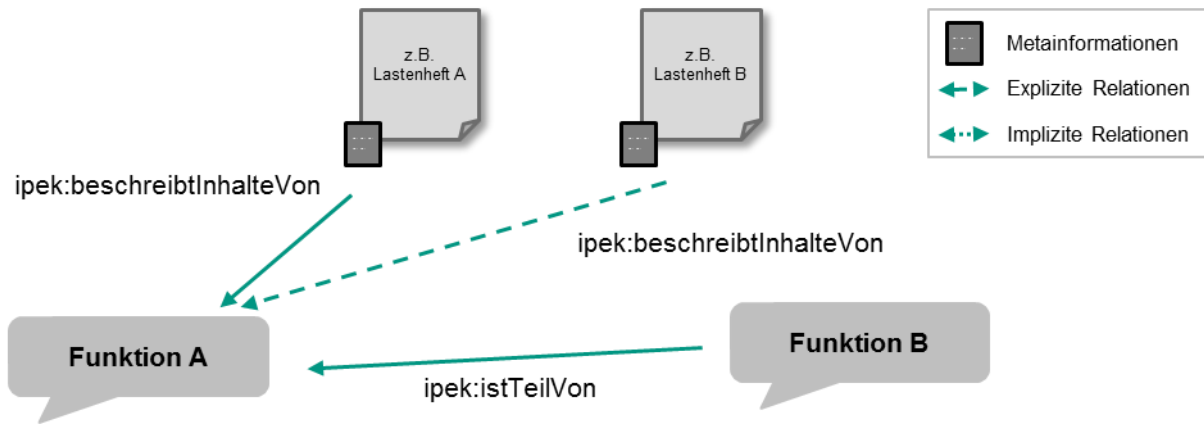


Abbildung 71: Ableitung impliziter Relationen aus einer Funktionsstruktur

Werden in zwei unterschiedlichen Entwicklungsartefakten auf die gleichen Funktionen verwiesen, kann ebenfalls eine Schlussfolgerung getroffen werden, dass eine Relation besteht.

*Beispiel 3:*

```
[rule3:
(?artefactA ipek:beschreibtInhaltVon ?funktionA)
(?artefactB ipek:beschreibtInhaltVon ?funktionA)]
->
(?artefactA ipek:istVerwandtMit ?artefactB)]
```

**6.2.3 Implizite Relationen aus Stakeholder-Beziehungen**

Relationen zu Stakeholdern sind wichtig, um ableiten zu können, wer an Änderungen an einem Dokument interessiert sein könnte. Die im Folgenden genannten Regeln können auf den bereits definierten Ableitungen aufsetzen und damit eine wichtige Basis dafür liefern, wer bei Änderungen informiert werden muss.

**Implizite Relationen über System-/Funktions-Verantwortlichkeiten:**

Aus einer System- und/oder Funktionsverantwortung kann ein Interesse an einem Dokument geschlussfolgert werden, wenn die entsprechende Funktion oder das System im Dokument genannt werden.

*Beispiel 1:*

```
[rule1:
(?artefactA po:beschreibtInhaltVon ?systemA)
(?systemA po:verantwortlich ?stakeholderA)
->
(?stakeholderA po:interessiert ?artefaktA)]
```

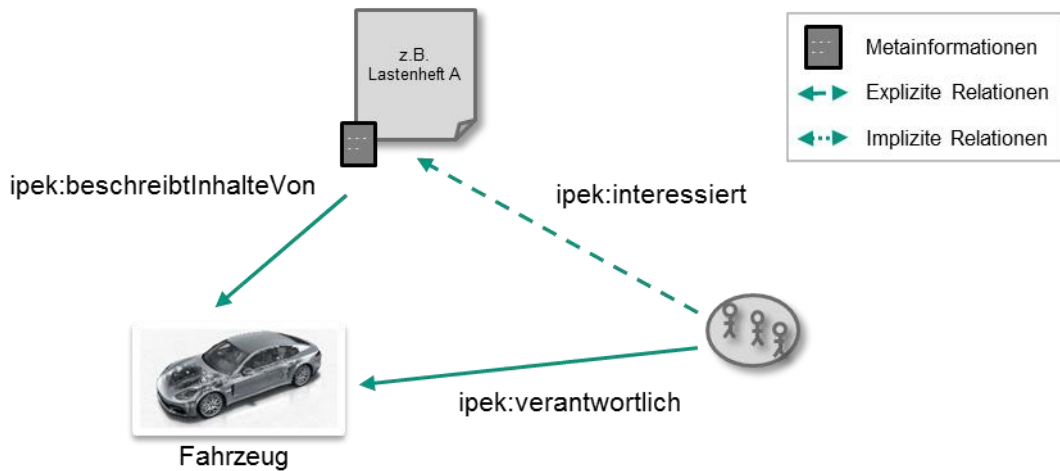


Abbildung 72: Ableitung impliziter Relationen aus Stakeholder-Beziehungen

### Implizite Relationen über Artefakt-Relationen:

Ist bekannt, dass ein Stakeholder interessiert oder verantwortlich für ein Entwicklungsartefakt ist, kann aus einer bekannten Relation zwischen zwei Entwicklungsartefakten eine Ableitung zu einer weiteren Stakeholder-Relation gebildet werden.

#### Beispiel 2:

```
[rule2:
(?artefact1 po:istVerwandtMit ?artefact2)
(?artefact1 po:verantwortlich ?stakeholder)
->
(?artefact2 po:interessiert ?stakeholder)]
```

#### Beispiel 3:

```
[rule3:
(?artefact1 po:istTeilVon ?artefact2)
(?artefact1 po:verantwortlich ?stakeholder)
->
(?artefact2 po:interessiert ?stakeholder)]
```

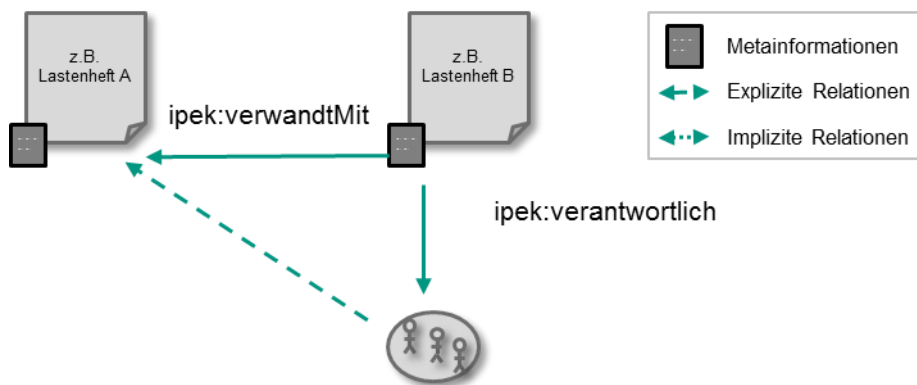


Abbildung 73: Implizite Relationen über Artefakt-Beziehungen

### 6.2.4 Implizite Relationen über Entwicklungs- oder Produktgenerationen

Neben impliziten Relationen aus Inhalten des Dokuments kann auch aus Informationen über die Entwicklungs- oder Produktgeneration Rückschlüssen darüber abgeleitet werden, ob Entwicklungsartefakte in Relation zueinander stehen.

*Beispiel:*

```
[rule:
(?artefact1 po:entwicklungsphase Gen1)
(?artefact1 po:entwicklungsphase Gen2)
->
(?artefact2 po:verwandtMit artefact1)]
```

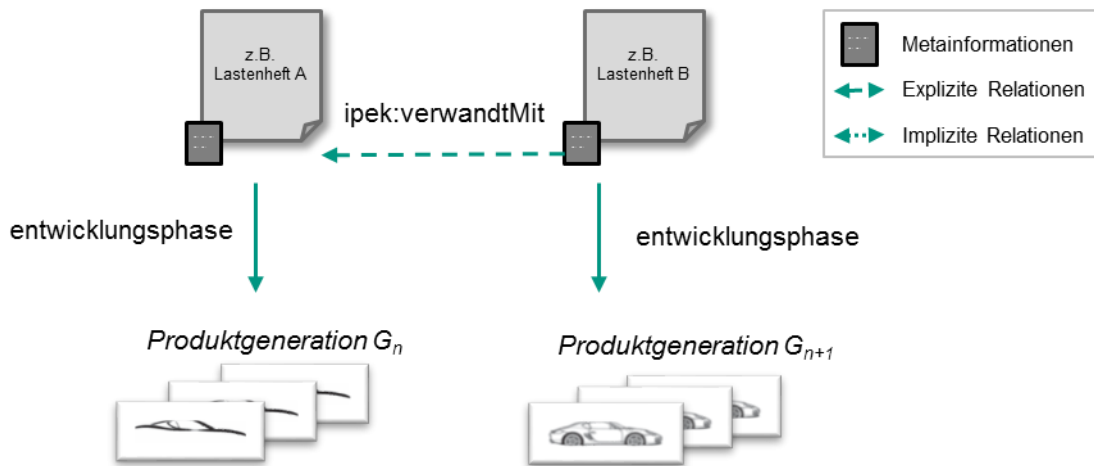


Abbildung 74: Implizite Relationen bei Produktgenerationen



### 6.3 Technische Realisierung durch eine web-basierte Plattform

Zur Herstellung von Durchgängigkeit im Zielsystem soll eine Web-basierte Plattform entwickelt werden. Eine Web-basierte Lösung bietet den Vorteil, dass neben einem Internet-Browser keine Software auf dem Endgerät (z.B. Laptop) installiert werden muss. Die Plattform erlaubt sowohl den Upload als auch die Verlinkung von Entwicklungsartefakten. Darüber hinaus können die Entwicklungsartefakte mit dem entwickelten RDF Schema beschrieben werden. Metainformationen sollen dabei prinzipiell durch jeden erstellt werden können. Eine Rechte- oder Zugriffsverwaltung auf Entwicklungsartefakte ist nicht im Umfang dieser Arbeit.

#### 6.3.1 Anwendungsarchitektur

Zur Bereitstellung der gewünschten Funktionen werden sieben Komponenten benötigt, welche integriert werden müssen. Der Aufbau der Anwendung ist in Abbildung 75 dargestellt.

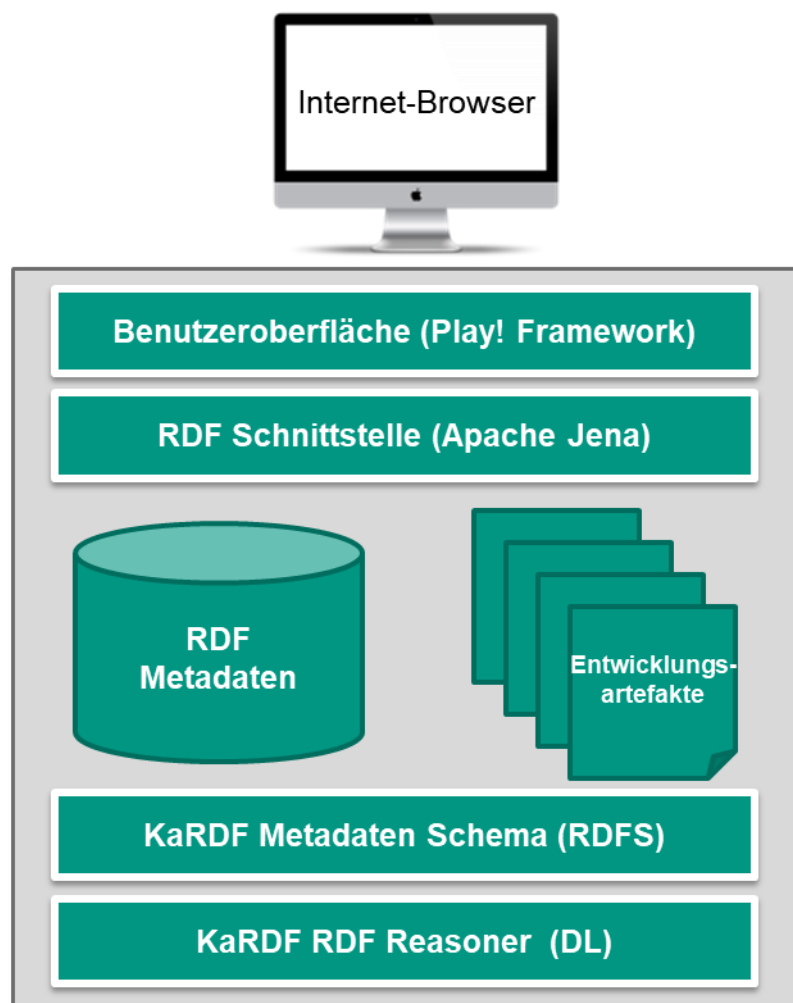


Abbildung 75: Logische Architektur der web-basierten Plattform

## Benutzeroberfläche

Aus den Untersuchungen der deskriptiven Studie I werden sieben Sichten abgeleitet:

- Entwicklungsartefakte: Übersicht über alle Artefakte und deren Metainformationen, sowie bekannter Relationen inklusive berechneter Inferenzen.
- Stakeholder: Übersicht über alle bekannten Stakeholder sowie deren Relationen
- Phasen / Aktivitäten: Übersicht über alle Phasen und Aktivitäten.
- Produktgenerationen: Darstellung von Produktgenerationen.
- Ziele / Anforderungen / Anwendungsfälle: Darstellung aller Ziele, Anforderungen und Anwendungsfälle
- Systeme: Übersicht über alle Systemen und deren Relationen
- Funktionen: Übersicht über alle bekannten Funktionen und Relationen

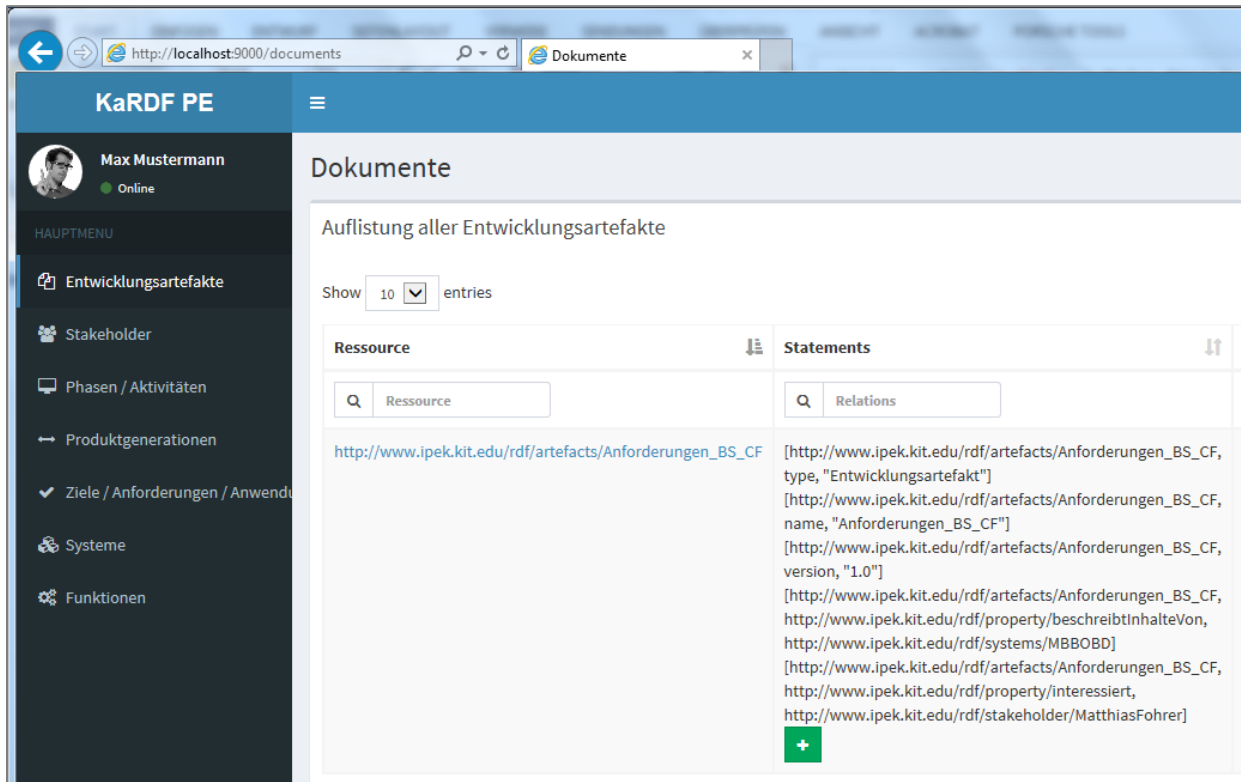


Abbildung 76: Benutzeroberfläche des Software-Prototyp

## IPEK RDF Schema

Das IPEK RDF Schema stellt die Basis für das RDF Modell dar und ermöglicht die Pflege von Metainformationen für Entwicklungsartefakte. Diese Komponente ist so ausgelegt, dass sie sich durch Erweiterung des RDF Schema auf Basis weiterer Forschungsarbeiten einfach erweitern lässt.

## IPEK RDF Reasoner

Der IPEK RDF Reasoner greift das entwickelte Regelwerk auf, um implizite Relationen aus einem bekannten RDF Modell abzuleiten. Diese Komponenten ist so ausgelegt, dass sie sich durch Erweiterung des RDF Schema auf Basis weiterer Forschungsarbeiten einfach erweitern lässt.

Der Prototyp erlaubt das einfache Hinzufügen von Relationen im Rahmen des entwickelten RDF Schema.

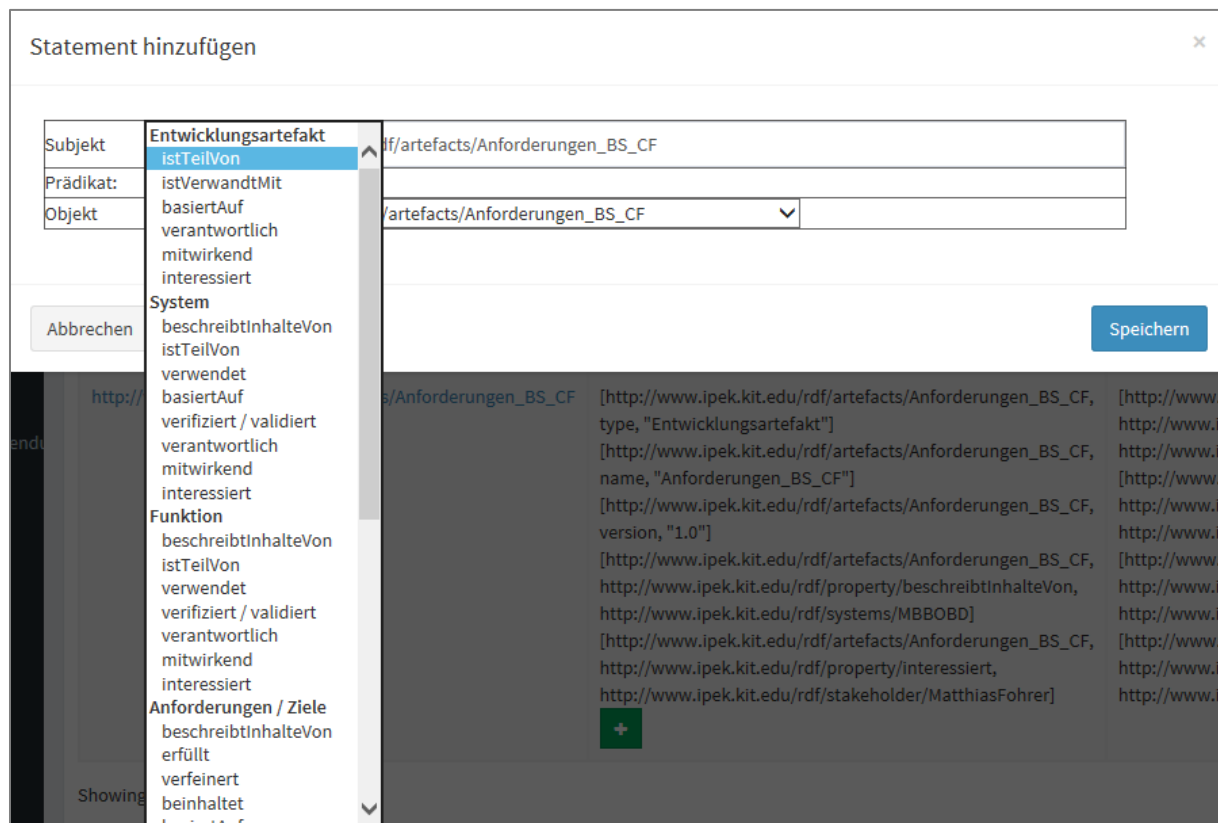


Abbildung 77: Hinzufügen von RDF Statements im entwickelten Prototyp

### 6.3.2 Technische Systemspezifikation

Der Kern der Anwendung wird mit dem Play! Framework umgesetzt. Play! ist ein Open Source Framework für Webanwendungen, das im Jahr 2007 von dem Software-Entwickler Guillaume Bort entwickelt wurde. Play setzt auf den Programmiersprachen Scala und Java auf und implementiert das Architekturmuster Model-View-Controller (MVC). Die Präsentationsschicht der Anwendung wird mit HTML5 sowie dem CSS Frameworks Bootstrap umgesetzt. Um HTML-Inhalte dynamisch anzupassen oder ereignisgesteuerte Events auszuführen wird das Javascript-Framework jQuery verwendet.

Zur Umsetzung des RDF Interface und des Reasoners wird das Open-Source Framework Apache Jena verwendet. Jena ist ein in Java geschriebenes Open Source-Framework, welches die Implementierung von Anwendung mit RDF

unterstützt. Jena bietet eine Programmierschnittstelle zum Laden und Speichern von Daten in RDF und gehört zu den populärsten RDF-Frameworks. Neben RDF bietet es auch Unterstützung für:

- OWL
- SPARQL
- RDQL

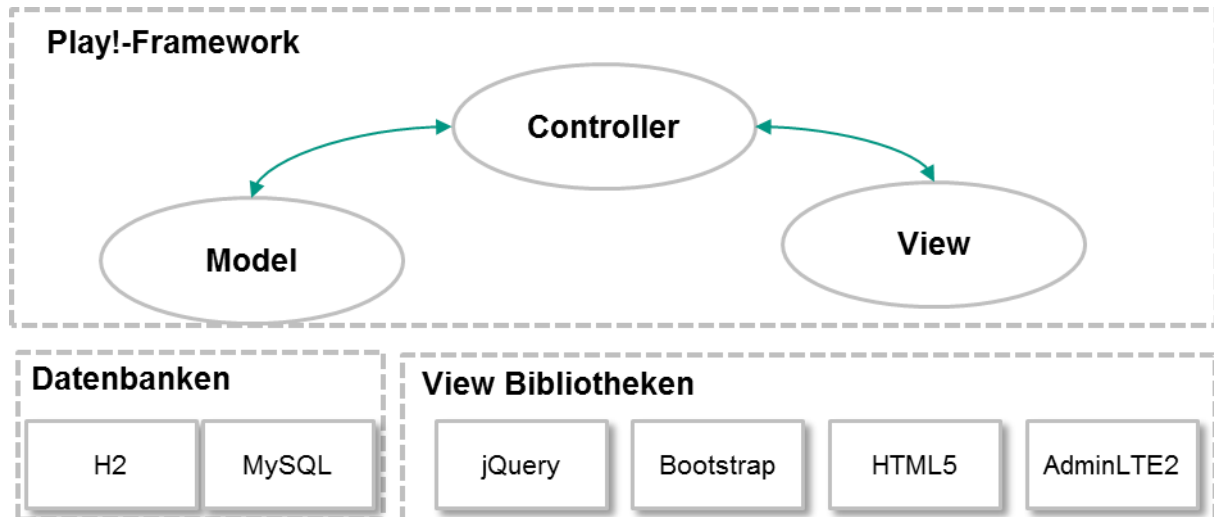


Abbildung 78: Verwendete Technologien zur Umsetzung der web-basierten Plattform

Die Implementierung von Reasonern/Inferenzmaschinen auf Basis RDF ist ebenfalls möglich. Jena unterstützt die Arbeit auf RDF Modelle im Arbeitsspeicher sowie in einer persistenten Datenbank.

Die Anwendung wird in einem Jetty-Web-Server ausgeführt. Jetty ist ein ebenfalls in Java geschriebener Open-Source Webserver und Java Servlet-Container. Das RDF Modell wird in einem MySQL Server persistiert. MySQL gehört zu den am weitesten verbreiteten relationalen Datenbanksystemen und gehört ebenfalls zum Typ der Open-Source Software. Die Anbindung der Java-Anwendung an die MySQL Datenbank erfolgt über die Java Database Connectivity (JDBC) mit einem MySQL JDBC-Treiber. Die Entwicklungsartefakte selbst werden auf einem Dateisystem gespeichert. Zur Entwicklung der IPEK RDF Ontologie wird Protégé verwendet. Bei Protégé handelt es sich um einen an der Universität Stanford entwickelten Editor für die Ontologieentwicklung. Die Anwendung ist ebenfalls in Java geschrieben und damit plattformunabhängig. Neben Protégé selbst sind auch diverse Plugins für Protégé als Open-Source-Projekte im Internet verfügbar. Protégé bietet mit dem OWL-Plugin eine OWL-Unterstützung an, womit es möglich ist, OWL basierte Ontologien zu erstellen und zu editieren. Das OWL-Plugin benutzt als Grundlage die Jena-API. Weitere Plugins wie beispielsweise OntoViz und Jambalaya für eine grafische Visualisierung von RDF Graphen können als Plugin in Protégé eingebunden werden.

## 6.4 Durchgängiges Management von Änderungen mit SPALTEN

Neben den klassischen Produktentwicklungsaktivitäten, welche die Kernaktivitäten der Produktentwicklung darstellen, gibt es noch allgemeine Aktivitäten, welche parallel und in einem regelmäßig wiederkehrenden Modus durchgeführt werden.<sup>387</sup> Damit sollen die Kernaktivitäten unterstützt und verbessert sowie der Produktentwicklungsprozess abgesichert werden. Zu diesem Cluster gehört das "Management von Änderungen".

Mit dem durchgängigen Management von Änderungen soll der entwickelte Ansatz zur durchgängigen Abbildung von Zielsystemen mit RDF prozessual in der Produktentwicklung verankert werden. Ziel ist es, Entwicklern eine Anleitung an die Hand zu geben, welche sie beim Management von Änderungen in einem verteilten Umfeld unterstützt.

Gemäß der Definition in Kapitel 5.4.2 kann die Umsetzung einer Änderung auch als Problem<sup>388</sup> aufgefasst werden, wobei das Management von Änderungen in einem verteilten Umfeld fraktalen Charakter hat. Das bedeutet, dass der gleiche Prozess auf unterschiedlichen Ebenen und in verschiedenen Phasen durch unterschiedliche Problemlösungsteams bearbeitet werden muss. Daher wurde für das Management von Änderungen ein Ansatz entwickelt, welcher auf der Problemlösungsmethodik SPALTEN<sup>389</sup> basiert (vgl. Abbildung 79). Die im Folgenden vorgestellte Methodik resultiert aus den Untersuchungen im Rahmen der deskriptiven Studie I (vgl. insbesondere Kapitel 5.4) und soll Entwicklern als eine Stütze beim Management von Änderungen in der Produktentwicklung dienen.

---

<sup>387</sup> vgl. Albers et al. 2016

<sup>388</sup> vgl. Kapitel 5.4.2

<sup>389</sup> vgl. Albers et al. 2005

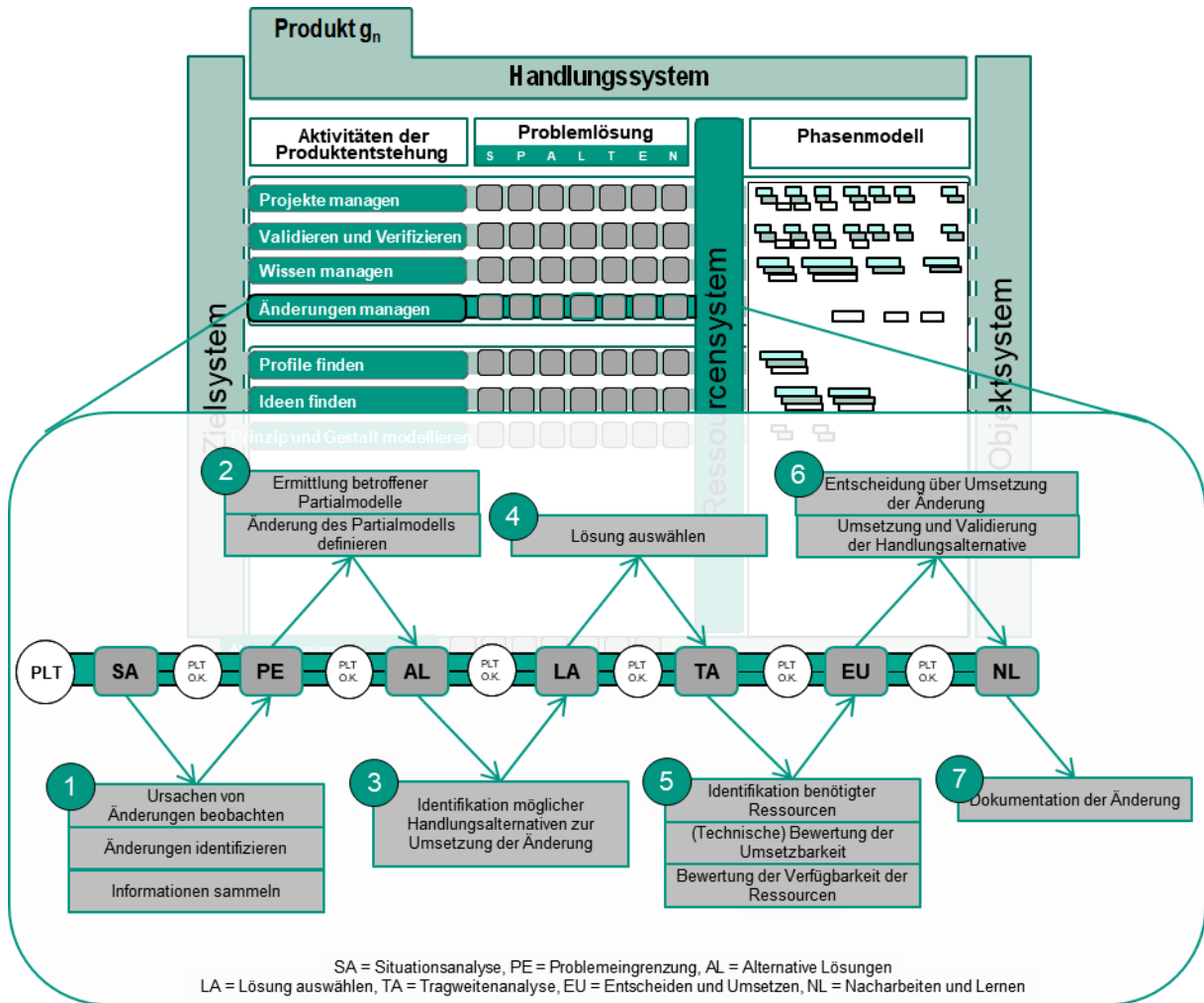


Abbildung 79: Eingliederung des Management von Änderungen in das iPeM<sup>390</sup>

Eine Änderung kann prinzipiell in allen Phasen der Entwicklung auftreten. Das Vorgehen kann in 6 ständig durchlaufenden Aktivitäten beschrieben werden:

1. Situationsanalyse (SA): Änderungen beobachten, Informationen sammeln und Änderung dokumentieren
  - a. Änderungen beobachten
  - b. Informationen sammeln
2. Problemeingrenzung (PE): Ursache und Wirkung der Soll-Ist Abweichung durch die Änderung wird ermittelt. Insbesondere werden die betroffenen Stakeholder und Systeme ermittelt.
  - a. Änderung definieren
  - b. Initiale Ermittlung betroffener Partialmodelle

<sup>390</sup> vgl. auch Albers, Reussner, Kurrle, Burger, Moeser, Klingler & Bursac 2017

3. Alternative Lösungen (AL) und Lösung auswählen (LA): Entwicklung möglicher Handlungsalternativen für die Durchführung der Änderung.
  - a. Entwicklung möglicher Handlungsalternativen
  - b. Lösung auswählen
4. Tragweitenanalyse (TA): Bewertung der Änderung, z.B. aus finanzieller oder technischer Sicht.
  - a. Bewertung der Änderung, z.B. aus finanzieller oder technischer Sicht
5. Entscheidung und Implementierung (EU): Entscheidung und Umsetzung durch alle relevanten Stakeholder.
  - a. Entscheidung
  - b. Einplanung
  - c. Umsetzung
6. Nacharbeiten & Lernen (NL): Dokumentation der Änderung. Erfahrungen bei der Änderungsdurchführung werden festgehalten.
  - a. Dokumentation der Änderung und Erfahrungen

Für die Durchführung des Managements von Änderungen wird situationsspezifisch ein Problemlösungsteam benötigt. Ergänzend zu den oben genannten Schritten muss in jedem Schritt noch einmal geprüft werden, ob das Problemlösungsteam dahingehend ausreichend besetzt ist, dass alle betroffenen Parteien eine Bewertung der Änderung abgeben.

#### **6.4.1 Situationsanalyse: Änderung beobachten und Informationen sammeln**

Änderungen können sowohl auf Grund von internen als auch auf Grund von externen, Faktoren auftreten. Bei unabhängig voneinander agierenden Handlungssystemen, wie Sie zum Beispiel bei einigen SoS zu finden sind Änderungen im System-Verbund möglich, ohne dass diese anderen im Verbund kommuniziert werden (z.B. weil der Teilnehmer des SoS auch gar keine Kenntnis von anderen hat). Um Änderungen beobachten zu können, müssen daher mögliche Einflussfaktoren in der Entwicklung identifiziert und fortlaufend während der Entwicklung beobachtet werden.

#### **Beispiel:**

- Einflussfaktor: Kunde
- Methode: Regelmäßige Akzeptanztests mit Kunden am Produkt

Nachdem eine Änderung identifiziert wurde müssen Informationen über eine identifizierte Änderung gesammelt, geordnet und Nachvollziehbar dokumentiert. Die Situationsanalyse stellt somit den Einsprungpunkt für das Management von Änderungen dar und bildet die Informationsgrundlage für die weitere Bearbeitung der Änderung. Eine genaue Interpretation der Änderung erfolgt noch nicht.

Änderungen können dabei auf mehreren Ebenen und in unterschiedlichen Phasen der Entwicklung möglich sein, wobei eine Unterscheidung notwendig ist zwischen einer Iteration, welche eine spezifizierte „Nulllinie“ nicht verändert. Zur Identifikation, ob der Prozess für das Management Änderung notwendig ist kann basierend auf den Erkenntnissen von Kapitel 5.4 eine Checkliste aufgebaut werden, welche die betroffenen Partialmodelle und zugehörige Verantwortlichkeiten nennt.

#### **Beispiel zur Verwendung der Metainformation in der Situationsanalyse:**

Die gepflegten Metainformationen liefern einen sehr guten Überblick über Entwicklungsartefakte, involvierte Systeme oder Funktionen.

#### **6.4.2 Problemeingrenzung: (Ziel der) Änderung definieren**

Im PE-Schritt wird die Informationsfülle auf die problemlösungsrelevanten Daten eingegrenzt. Hier wird die Ursache und Wirkung der Soll-Ist-Abweichung ermittelt. Dabei sollen insbesondere alle betroffenen technischen Systeme und deren Stakeholder ermittelt werden. Ursache und Wirkung der Soll-Ist Abweichung durch die Änderung wird ermittelt, insbesondere werden die betroffenen Stakeholder und Systeme ermittelt. Als Ergebnis bestimmt die PE die Ziele der Änderung und liefert eine detailliertere Beschreibung des Problems, die mit Hilfe einer Anforderungsliste oder anderen Methoden gelöst werden. Des Weiteren muss eine Priorisierung der Änderung erfolgen.

Um eine Problemeingrenzung durchführen zu können, müssen die Wirkzusammenhänge bekannt sein. Änderungen an einem technischen System können weitere Änderungen nach sich ziehen. Im Rahmen der Problemeingrenzung wird daher auch untersucht, wer von einer Änderung betroffen ist und welche „Nulllinie“ verändert werden muss. Die Identifikation der Nulllinie erfolgt auf Basis bereits bekannter Metainformationen.

Die gesammelten Erkenntnisse können beispielsweise in einem CR-Formular oder in einem entsprechenden Workflow-Tool dokumentiert werden. Dabei müssen insbesondere folgende Informationen dokumentiert werden:

- Informationen über den Anforderer der Änderung
- Betroffene Systeme und Funktionen
- Betroffene „Nulllinie“



- Zieltermin
- Priorität
- Betroffene Kunden/Märkte
- Beschreibung der Änderung

#### **6.4.3 Alternative Lösungen generieren und Lösung auswählen**

Ziel dieser Aktivität ist die Entwicklung von möglichen Lösungen und Handlungsalternativen, die analysiert, konkretisiert und dokumentiert werden. Nach ausführlicher Darlegung der Handlungsalternativen werden im Rahmen einer Lösungsauswahl die einzelnen Lösungen verglichen und hinsichtlich der Zielerfüllung aus der Problemeingrenzung bewertet. Am Ende des Schritts wird eine unter den vorgegeben Zielen am besten geeignete Lösung ausgewählt.

#### **6.4.4 Tragweitenanalyse: Bewertung der Änderung**

In der Tragweitenanalyse wird auf Basis der ausgewählten Lösungsalternative eine Bewertung der Änderung durchgeführt. Diese muss durch die von der Änderung betroffenen Stakeholdern durchgeführt und am Ende des Schrittes zusammengeführt und übergreifend bewertet werden. Neben einer finanziellen Bewertung muss hierbei insbesondere die Machbarkeit einer Änderung bestätigt werden.

Die Bewertung der Änderungen kann dabei durch unterschiedliche Parteien erforderlich sein. Die Bewertung der Änderung wird gestoppt, sofern von einer betroffenen Partei die Rückmeldung erfolgt, dass die Änderung nicht durchführbar ist.

#### **Beispiel zur Verwendung der Metainformationen in der Tragweitenanalyse:**

Bei der Tragweitenanalyse können aus den gepflegten Metainformationen über Rückschlüsse Zusammenhänge in der Produktentwicklung geschlossen werden. In Abbildung 80 ist ein Beispiel im Software-Prototyp dargestellt, wie aus vorhanden Metainformation über Inferenz automatisch Rückschlüsse darüber gezogen werden, wer an einem Entwicklungsartefakt interessiert ist. Im Falle einer Änderung des Entwicklungsartefakts können diese Informationen genutzt werden, um das Problemlösungsteam zu erweitern oder den Stakeholder zumindest über die Änderung zu informieren.

Home > Dashboard
+ Entwicklungsartefakt hinzufügen

**Dokumente**

Auflistung aller Entwicklungsartefakte

Show  entries

**Statements**

Relations

[http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, type, "Entwicklungsartefakt"]  
 [http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, name, "Anforderungen\_BS\_CF"]  
 [http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, version, "1.0"]  
 [http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, http://www.ipek.kit.edu/rdf/property/beschreibtInhalteVon, http://www.ipek.kit.edu/rdf/systems/MBBODB]  
 [http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, http://www.ipek.kit.edu/rdf/property/interessiert, http://www.ipek.kit.edu/rdf/stakeholder/MatthiasFohrer]

**Inferences**

Lukas

[http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, http://www.ipek.kit.edu/rdf/property/interessiert, http://www.ipek.kit.edu/rdf/stakeholder/LukasWerner]  
 [http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, http://www.ipek.kit.edu/rdf/property/interessiert, http://www.ipek.kit.edu/rdf/stakeholder/MatthiasFohrer]  
 [http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, http://www.ipek.kit.edu/rdf/property/beschreibtInhalteVon, http://www.ipek.kit.edu/rdf/systems/MBB]  
 [http://www.ipek.kit.edu/rdf/artefacts/Anforderungen\_BS\_CF, http://www.ipek.kit.edu/rdf/property/beschreibtInhalteVon, http://www.ipek.kit.edu/rdf/systems/MBBODB]

Showing 1 to 1 of 1 entries

Previous **1** Next

Abbildung 80: Verwendung der Suchfunktion im RDF Prototyp

Der fraktale Charakter des Ansatzes unterstützt den Anwender bei der zielgerichteten Vertiefung relevanter Aspekte. Während der Bearbeitung kann zur Bearbeitung des Teilschrittes ein neuer Prozess initiiert werden, beispielweise weil die Auswirkungen bei einem Subsystem geprüft werden müssen. Auf diese Weise kann der Ansatz strukturiert an das Abstraktionsniveau der Problembetrachtung angepasst werden.<sup>391</sup>

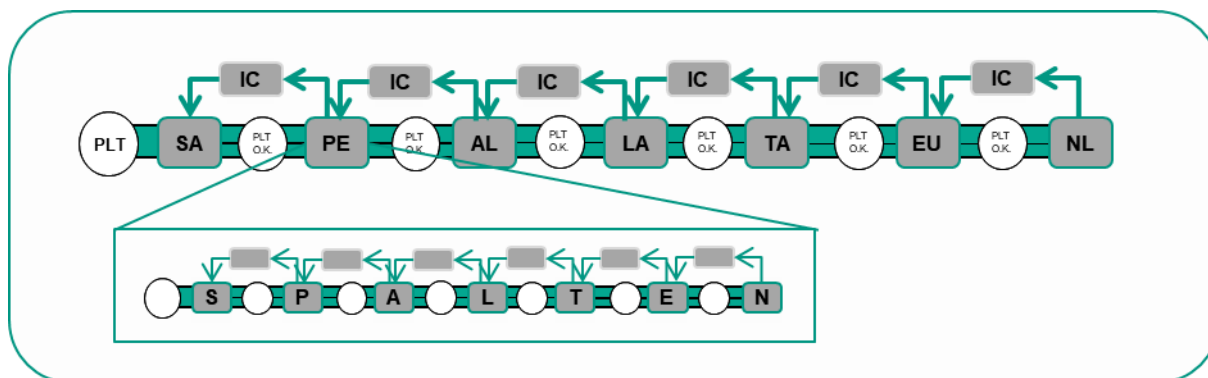


Abbildung 81: Fraktaler Charakter der SPALTEN-Methodik<sup>392</sup>

#### 6.4.5 Änderung entscheiden und umsetzen

Die Tragweitenanalyse bildet die Basis für eine Entscheidung der Umsetzung. Dabei müssen alle betroffenen Stakeholder in den Entscheidungsprozess mit einbezogen werden. Nach erfolgter Entscheidung werden die ausgewählten Maßnahmen umgesetzt. Ziel der Umsetzung ist die vollständige Bearbeitung und Realisierung der ausgewählten Lösung(en) in Verbindung mit der Integration der ermittelten Maßnahmen zur Risikominimierung und Chancenoptimierung. Ziele werden in Elemente des Objektsystems transformiert. Im Anschluss an die Umsetzung muss die Änderung validiert werden.

Nach der Umsetzung muss die Änderung dokumentiert werden. Dies muss bei allen betroffenen Parteien dezentral erfolgen. Relevante Entwicklungsartefakte müssen aktualisiert werden.

#### 6.4.6 Nacharbeiten und Lernen

Der abschließende Schritt dient dem Nacharbeiten und Lernen. Hier sollen Erfahrungen bei der Änderungsdurchführung festgehalten werden, so dass die Informationsbasis für zukünftige Problemlösungen wieder verbreitert wird. Die gesammelten neuen Erkenntnisse können durch neue Metainformationen auch anderen in der Entwicklung zugänglich gemacht werden.

<sup>391</sup> vgl. auch Schneider 2012

<sup>392</sup> vgl. A Albers, Braun, and Muschik 2010

**Beispiel zur Verwendung der Metainformation beim Nacharbeiten:**

Wird bei einer Änderung eine neue Beziehung zwischen zwei Systemen aufgebaut, so kann dies durch das bereitgestellte RDF Schema gepflegt werden. Diese Information steht dann für zukünftige Situations- oder Tragweitenanalysen zur Verfügung und wird bei der Berechnung von Inferenzen berücksichtigt.

## 7 Evaluation des Ansatzes

In diesem Kapitel wird eine zweite deskriptive Studie durchgeführt, um den entwickelten Ansatz in der Praxis zu evaluieren. Dazu werden zwei weitere Forschungsfragen aus der Forschungsfrage 3 abgeleitet (vgl. Abbildung 82).

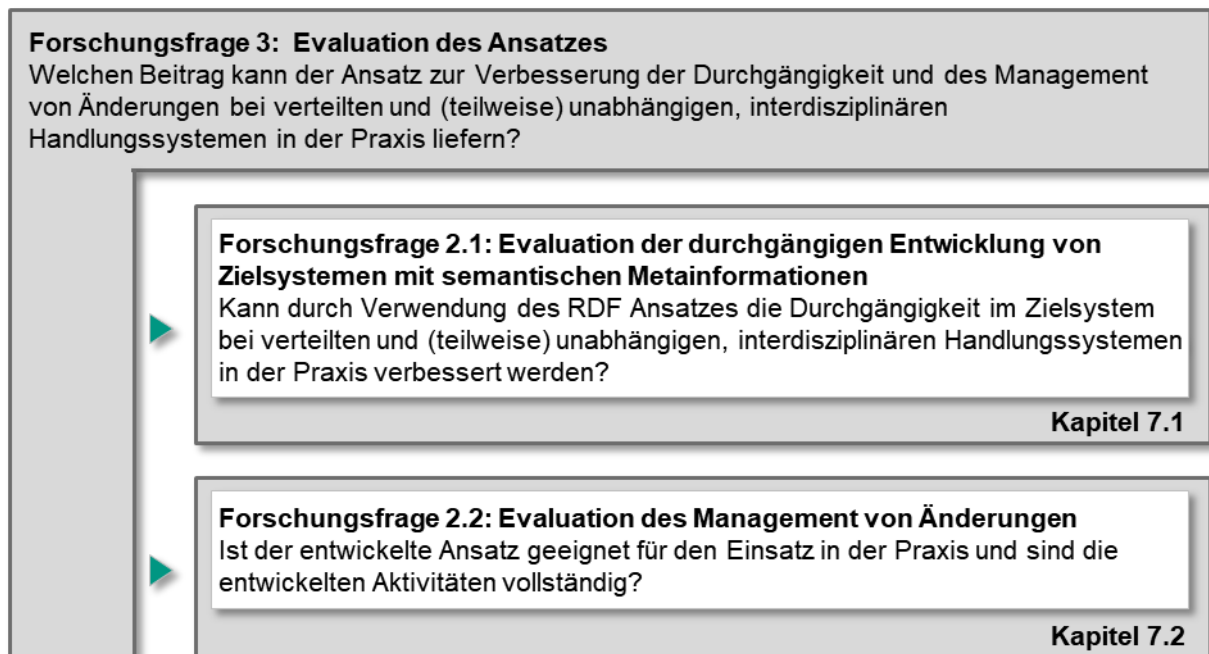


Abbildung 82: Forschungsfragen der Validierung

Im ersten Teil wird der Ansatz zur Verlinkung von Entwicklungsartefakten mit semantischen Metainformationen dazu genutzt, das Zielsystem von Entwicklungsprojekten bei der Dr. Ing. h.c. F. Porsche AG im Bereich Connected Car retrospektiv zu modellieren und die Potentiale zur Herstellung von Durchgängigkeit und die generelle Anwendbarkeit des Ansatzes zu evaluieren.

Im zweiten Teil wird die Methode für das Management von Änderungen im Rahmen eines Live-Lab bewertet. Hierbei soll insbesondere die Vollständigkeit und Tauglichkeit der Methode für den Einsatz in der Praxis bestätigt werden.

### 7.1 Evaluation der durchgängigen Dokumentation mit semantischen Metainformationen

Im ersten Teil der Validierung soll insbesondere die Forschungsfrage beantwortet werden, ob mit dem entwickelten Ansatz die Durchgängigkeit bei der Entwicklung von Zielsystemen durch verteilte und teilweise unabhängig voneinander arbeitende, interdisziplinäre Handlungssystemen verbessert werden kann. Dabei sollen folgende Punkte bewertet werden:

- Ist die Integration des Ansatzes in bestehende Methoden, Prozesse und Tools in den verteilten Entwicklungsteams möglich?
- Ist das entwickelte Metainformationen-Schema dazu geeignet, um Relationen bei heterogenen Entwicklungsartefakten abzubilden, um die Durchgängigkeit zu erhöhen?
- Können aus den definierten Metainformationen implizit über die definierten Ableitungsregeln weitere Relationen abgeleitet werden, um damit Durchgängigkeit und Konsistenz zu erhöhen?
- Übersteigt der Nutzen (Verbesserung der Durchgängigkeit in der Produktentwicklung) den Aufwand zur Pflege der Metainformationen, sodass grundsätzlich eine Motivation der Entwickler zur Nutzung des Ansatzes in der Praxis bestehen kann?

Insgesamt soll damit untersucht werden, ob der entwickelte Ansatz dabei Unterstützt, Durchgängigkeit über den Detaillierungsgrad, Produkt- und Entwicklungsgenerationen und unterschiedliche Projekte in verteilten und interdisziplinären Entwicklungsumgebungen herzustellen.

### 7.1.1 Studiendesign

Die Validierung des auf RDF basierten Ansatzes erfolgt durch retrospektive Definition semantischer Metainformationen durch Projektleiter der Dr. Ing. h.c. F. Porsche AG aus dem Bereich Porsche Connect unter Praxisbedingungen.

Die Teilnehmer der Studie werden dazu aufgefordert, Metainformationen für Entwicklungsartefakte aus einem Entwicklungsprojekt, an dem eine Vielzahl von Entwicklungsteams aus unterschiedlichen Bereichen beteiligt sind, zu definieren. Dazu verwenden die Teilnehmer den entwickelten Software-Prototypen (vgl. Abbildung 83) zunächst unter Anleitung des Versuchsleiters und später eigenständig.

Nach Pflege der Metainformationen im Software-Prototyp werden die Teilnehmer durch einen strukturierten Fragebogen zur Bewertung folgender Aspekte aufgefordert:

- Bewertung der Fähigkeit zur Integration der Methode mit bestehenden Methoden, Prozesse und Tools
- Bewertung der Erhöhung von Durchgängigkeit und Konsistenz durch Abbildung expliziter Relationen mit semantischen Metainformation
- Bewertung der Erhöhung von Durchgängigkeit und Konsistenz durch Ableitung impliziter Relationen aus den definierten Ableitungsregeln
- Bewertung der allgemeinen Anwendbarkeit des Ansatzes

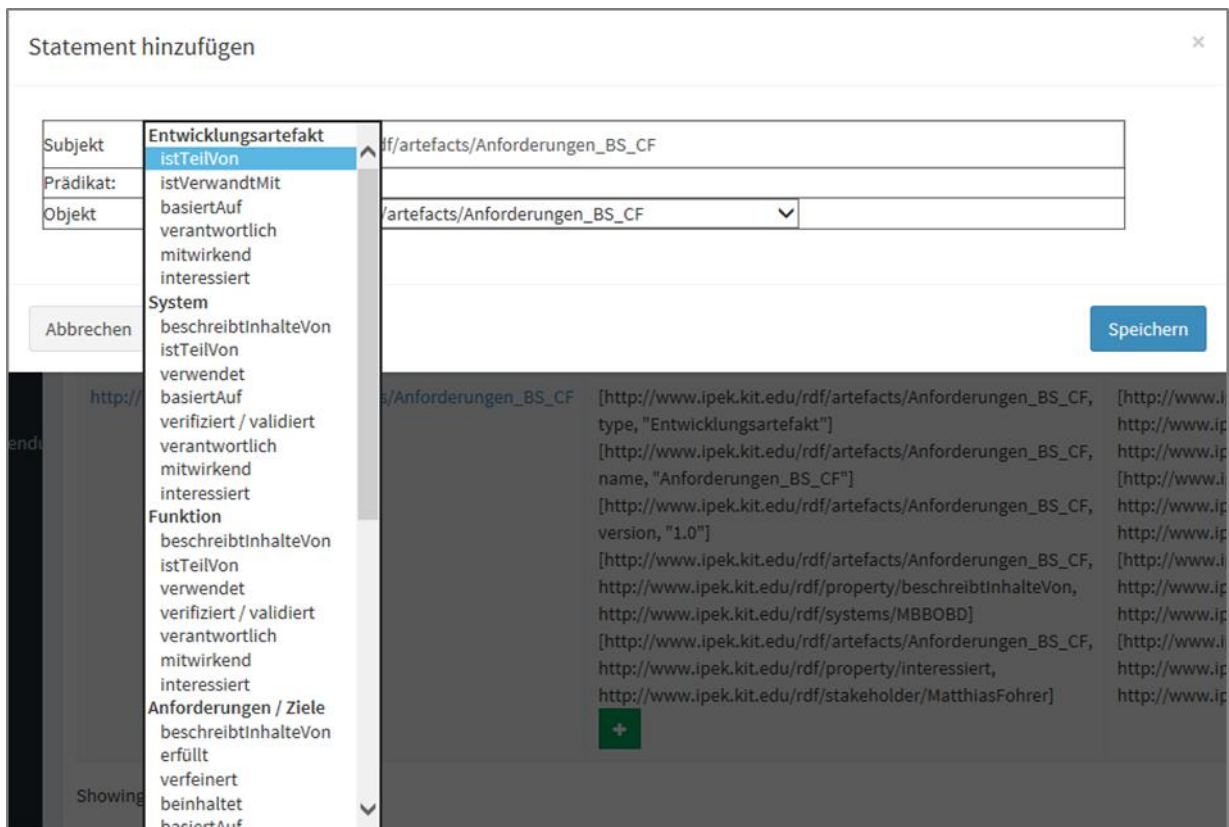


Abbildung 83: Pflege von Metainformationen im entwickelten Software-Prototyp

Neben einer quantitativen Bewertung vorgegebener Aussagen auf einer Likert- und Ordinal-Skala wird den Teilnehmern die Möglichkeit gegeben, Ihre Bewertungen zu kommentieren. Eine Bewertung der Usability des Prototyps wird explizit nicht berücksichtigt, da für eine auf Usability ausgelegte Gestaltung des Prototyps weitere Forschungen erforderlich sind, welche nicht im Fokus dieser Arbeit liegen.

Die Studie wurde im Winter 2016 durchgeführt. Insgesamt haben 5 Porsche Connect Projektleiter aus unterschiedlichen Fachbereichen der Dr. Ing. h.c. F. Porsche AG teilgenommen.

### 7.1.2 Ergebnisse und Implikationen

Insgesamt wurden 237 semantische RDF Statements (Subjekt, Prädikat, Objekt) durch die Teilnehmer im Software-Prototyp definiert. Es kann daher davon ausgegangen werden, dass für die Bewertung die prinzipiellen Möglichkeiten des Ansatzes und die Art und Weise, wie und in welchem Umfang Metainformationen gepflegt werden können durch die Teilnehmer ausreichend Verstanden wurden. Die Ergebnisse der abschließenden Bewertung werden im Folgenden vorgestellt.

#### **Integrationsfähigkeit des Ansatzes mit bestehende Methoden, Prozessen und Tools**

Mit der Integrationsfähigkeit soll insbesondere bewertet werden, ob das in Kapitel 6 definierte Ziel Z-1 durch den Ansatz erfüllt wird:

***„Z-1: Integration und nicht Ablösung bestehender Methoden, Prozesse und Tools in den verteilten Entwicklungsteams.“***

Alle Teilnehmer stimmen der Aussage zu, dass sich der Ansatz bei verteilten und teilweise unabhängig agierenden, interdisziplinären Handlungssystemen anwenden lässt. Laut Meinung der Teilnehmer werden dabei unterschiedliche Repräsentationsformen und Speicherorte, Methoden sowie Prozesse unterstützt (vgl. Abbildung 84). Teilnehmer 5 hat dabei insbesondere gelobt, dass durch den Ansatz eine „übersichtliche, zentrale Anlaufstelle für alle möglichen (technischen) Informationen zum Einsehen und Pflegen gegeben wird“, wodurch Abhängigkeit transparent gemacht werden. Dabei wird nach Teilnehmer 3 durch die Metainformationen eine Informationsstruktur bereitgestellt, die weit über einfache Dokument-Verlinkungen, wie in bestehenden Ansätzen und Tools zu finden (z.B. Microsoft Word, Atlassian Confluence, IBM Rational Doors), hinausgeht. Teilnehmer 5 hat hervorgehoben, dass dadurch nicht mehr „alle Entwicklungsartefakte durchgelesen werden müssen“. Ein weiterer Teilnehmer hat vorgeschlagen, den Ansatz auf Grund seiner „Generik“ auch in anderen Bereichen (z.B. Schulungen, On-Boarding von neuen Mitarbeitern) zu verwenden. Teilnehmer 2 kommentiert, dass durch den Ansatz Projekte und Projekteinhalte „ohne explizite Kenntnis der anderen Projekte“ synchronisiert werden können.

Um den Ansatz im Projekt oder Unternehmen zu etablieren merkt Teilnehmer 1 an, das zunächst das „MindSet“ sowie die „Vorteile der Nutzung“ allen transparent gemacht werden müssen. Für eine Prozess und Methodenintegration „muss definiert werden, wer welche Informationen wann anlegt“. Dabei muss nach Teilnehmer 4 den Nutzern erläutert werden, „wie man das System bedient“. Teilnehmer 4 und 5 sehen weitere Einschränkungen bei der Verteilung von „Zugriffsrechten“, da diese nicht zentral vergeben werden können.



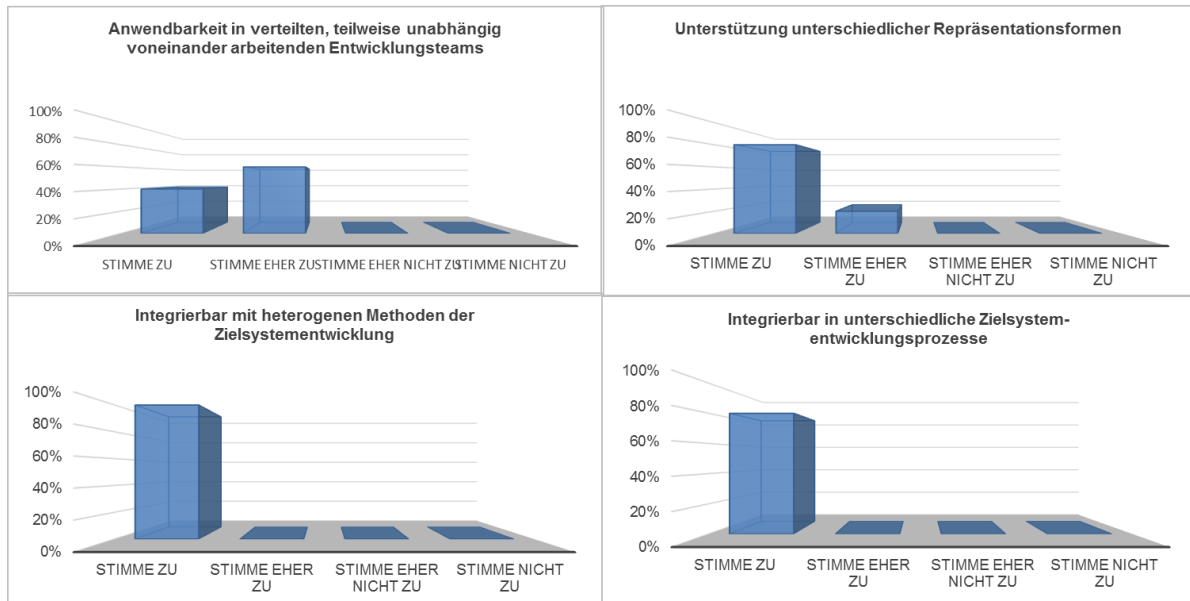


Abbildung 84: Ergebnis der Bewertung zur Anwendbarkeit in verteilten, teilweise unabhängig voneinander arbeitenden Teams

### **Bewertung der Erhöhung von Durchgängigkeit und Konsistenz durch Verwendung semantischer Metainformation in der Produktentwicklung**

Mit dem zweiten Teil der Bewertung soll abgefragt werden, inwieweit das zweite und dritte Ziel (vgl. Kapitel 6) an den Ansatz erfüllt wird:

*Z-2: Durchgängigkeit im Zielsystem muss über den Detaillierungsgrad, Produkt- und Entwicklungsgenerationen und unterschiedliche Projekte auch bei unterschiedlichen Produktgenerations- und Entwicklungszyklen hergestellt werden.*

und

*Z-3: Verwendung von Metainformationen bei Entwicklungsartefakten zur Abbildung von Relationen bei heterogenen Entwicklungsartefakten und als Basis für eine Ableitung von impliziten Relationen zwischen Entwicklungsartefakten.*

Die Teilnehmer sind der Ansicht, dass sich durch den Ansatz die Durchgängigkeit sowohl zeitlich, über den Detaillierungsgrad, über unterschiedliche Projekte und auch Partialmodellübergreifend verbessern lässt (vgl. Abbildung 85). Alle Teilnehmer sind der Ansicht, dass durch die Inferenzregeln die Durchgängigkeit weiter erhöht werden kann (Stimme zu: 60% / Stimme eher zu: 40 %).

Teilnehmer 1 lobt, dass durch den Ansatz „Medienbrüche nicht aufgelöst, aber handhabbar“ gemacht werden. Teilnehmer 2 begrüßt insbesondere die Möglichkeit, Entwicklungsartefakte von existierenden Referenzprodukten in Beziehung zu setzen. Teilnehmer 3 lobt, dass man „sehr flexibel“ ist und sich Metainformationen „dynamisch pflegen“ lassen.

Einige Teilnehmer merken jedoch an, dass durch die Verwendung des Ansatzes ein zusätzlicher Aufwand zur Pflege entsteht. Teilnehmer 3 ist jedoch der Ansicht, dass man Durchgängigkeit nie „Geschenkt kriegt“. Teilnehmer 2 schlägt vor den „Zusatzaufwand der Pflege durch manuelle Eingabe der Metadaten z.B. durch Textmining zu erweitern“. Teilnehmer 1 merkt an, dass die Metainformationen „sehr schnell sehr komplex“ werden. Er schlägt daher vor, dass Information so verlinkt und dargestellt werden müssen, dass sie „im Kontext [...] auch plausibel sind“. Teilnehmer 1 schlägt vor, die Qualität der Daten durch „Plausibilitätsprüfung“ bei der Eingabe hoch zu halten. Teilnehmer 3 sagt, dass der Erfolg nur dann sichergestellt ist, wenn die Metainformationen „gleichermaßen von allen gepflegt werden.“ Teilnehmer 4 fordert, dass das RDF Schema einfach im Tool erweitert werden kann.

Die bereitgestellten Metainformation werden von allen Teilnehmern als nahezu vollständig erachtet. Von einigen Teilnehmern wurden weitere Vorschläge für Metainformationen gemacht. Teilnehmer 1 schlägt eine Erweiterung des RDF Schemas um Rollen und Verantwortlichkeiten von Stakeholdern vor (z.B. „Person A erstellt Angebotskonzept für Funktion X“). Teilnehmer 5 fordert eine zusätzliche Gruppierung über verwendete Technologie sowie Entwicklungstypen. Teilnehmer 2 fordert eine Gruppierungsmöglichkeit von Entwicklungsartefakten über Inhaltstypen (z.B. Bedienkonzept, Übersetzungstexte). Teilnehmer 3 fordert eine Integration des Fehlermanagements.

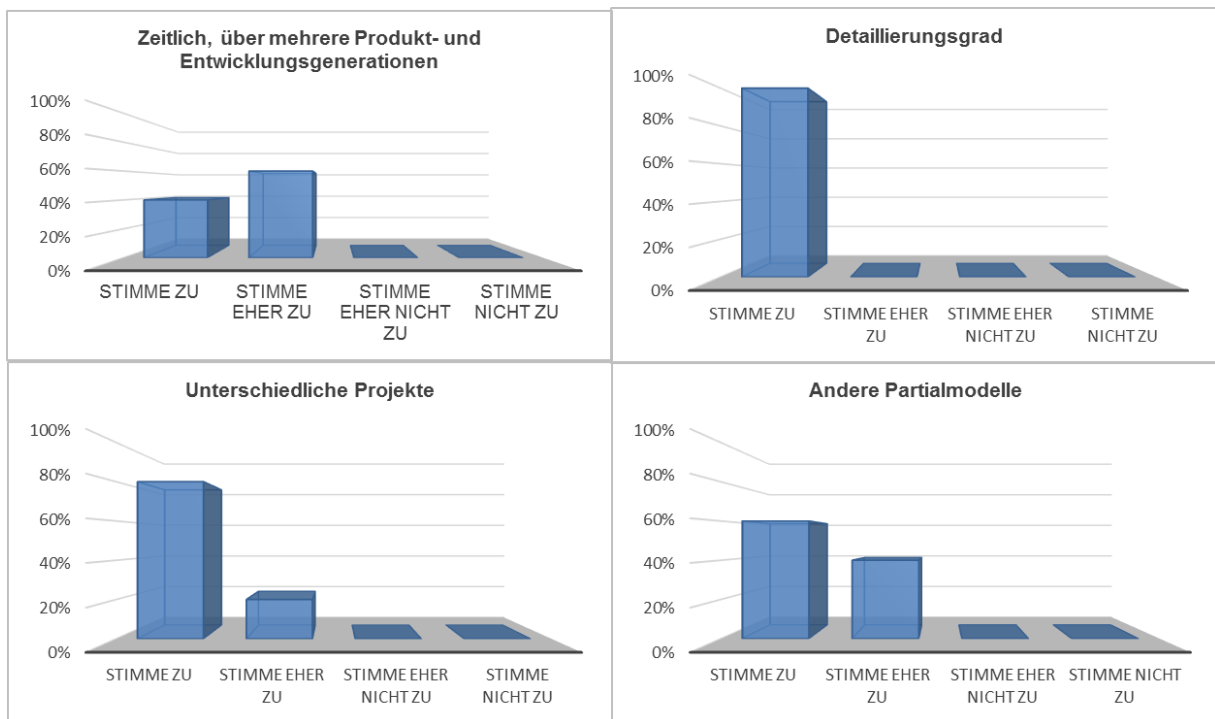


Abbildung 85: Bewertung der Erhöhung von Durchgängigkeit und Konsistenz durch den Ansatz

### Bewertung von Aufwand, Nutzen und Anwendbarkeit

Im Mittel wird der Aufwand zur Pflege der Metainformationen von den Teilnehmern als „hoch“ eingestuft. Alle Teilnehmer sind jedoch der Ansicht, dass der Nutzen ebenfalls mindestens hoch ist.

Nach Meinung von 80 Prozent der Teilnehmer übersteigt der Nutzen den Aufwand bei der Herstellung von Durchgängigkeit. Diese Teilnehmer würden den Ansatz in der Produktentwicklung gerne verwenden. Die restlichen 20 Prozent würden den Ansatz „vielleicht“ nutzen wenn sichergestellt ist, dass eine „kritische Masse“ von Metainformationen in ausreichender Qualität auch von anderen Mitarbeitern gepflegt werden.

Alle Teilnehmer sehen das Potential, dass durch entsprechende Pflegemasken und Automatismen der Aufwand zur Pflege noch deutlich gesenkt werden kann, wodurch der Ansatz zusätzlich an Attraktivität gewinnen würde.

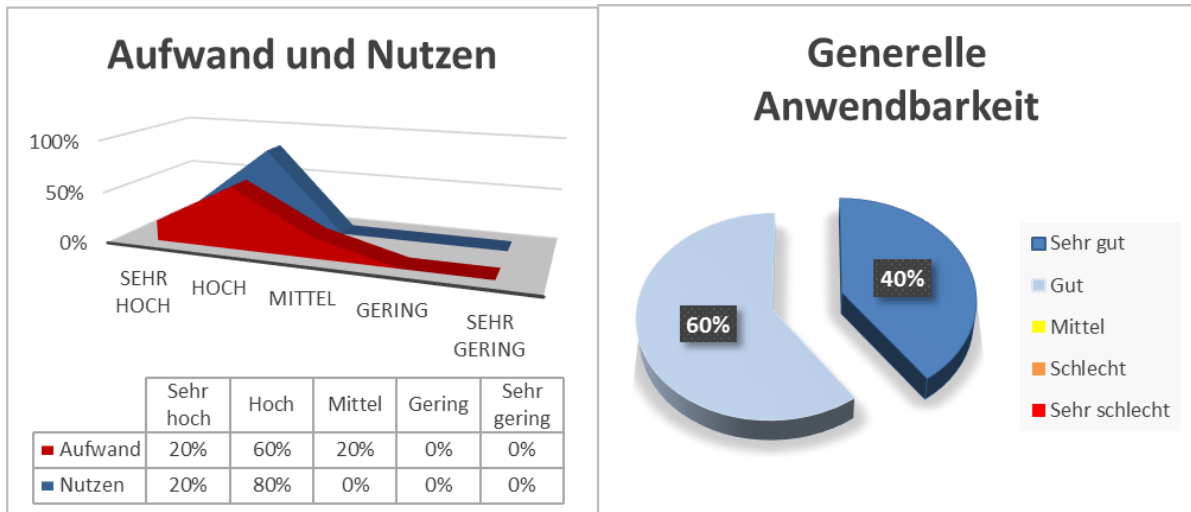


Abbildung 86: Bewertung von Aufwand und Nutzen sowie der generellen Anwendbarkeit des Ansatzes in der Produktentwicklung

## 7.2 Evaluation des Management von Änderungen im Live-Lab ProVIL

Durch die Evaluation des auf SPALTEN basierten Ansatzes für das Management von Änderungen soll die generelle Eignung des Ansatzes in verteilten Teams validiert werden und die Vollständigkeit und Tauglichkeit der definierten Aktivitäten sichergestellt werden. Ein weiteres Ziel ist die Überprüfung der Anwendbarkeit des Ansatzes in der Praxis und damit die Sicherstellung der Übertragbarkeit in andere Entwicklungsszenarien.

### 7.2.1 Studiendesign

Die Untersuchungen werden im Rahmen des Live-Lab ProVIL<sup>393</sup> durchgeführt. Bei ProVIL haben 32 Studierende des Studiengangs Maschinenbau aus höheren Fachsemestern von April bis Juli 2016 eine reale Entwicklungsaufgabe vom Projektpartner Dr. Ing. h.c. F. Porsche AG in einem gemeinsamen Entwicklungsprojekt bearbeitet. Zielsetzung des Projektes ist die Entwicklung innovativer digitaler Dienste für den Sportwagen der Zukunft. Das Live-Lab ProVIL ist für die Evaluation besonders geeignet, weil die studentischen Teams verteilt an einer Entwicklungsaufgabe aus dem Bereich Connected Car arbeiten.

Ein Live-Lab bietet eine Untersuchungsumgebung zur Erforschung von Methoden und Prozessen der Produktentwicklung in einem möglichst realen Entwicklungsprozess, bietet aber gleichzeitig die Möglichkeit, Randbedingungen in hohem Maße gestalten zu können<sup>394</sup>. Das Live-Lab-Konzept ist zwischen einer Labor- und einer Feldstudie einzuordnen<sup>395</sup>. Der Vorteil des Live-Lab Konzept gegenüber einer Feldstudie besteht unter anderem in einem weniger stark eingeschränkten, unternehmensspezifischen Geltungsbereich.

Der Entwicklungsprozess im Live-Lab ProVIL orientiert sich an vier Entwicklungsphasen (vgl. Abbildung 87). Zur Erarbeitung der Arbeitsergebnisse wenden die Studierenden verschiedene Kreativitäts-, Entwicklungs- und Bewertungsmethoden unter Nutzung von vorgegeben Werkzeugen (z.B. Softwaretools) an. Theoretische Produktentwicklungskennnisse werden den Studenten im Rahmen einer parallel stattfindenden Vorlesung vermittelt.

In der Recherchephase verschaffen sich die Teilnehmer zunächst einen Überblick über die Thematik entlang definierter Recherchefelder. Anschließend werden in der Profilphase 64 Profile auf Basis einer Marktanalyse definiert, um die Problemstellung

---

<sup>393</sup> vgl. Albers et al. 2015

<sup>394</sup> vgl. Walter, Albers & Haupt 2016

<sup>395</sup> vgl. Walter, Albers & Haupt 2016

systematisch einzugrenzen und die relevanten Randbedingungen kennenzulernen. Basierend auf den Produktprofilen entwickeln die Teilnehmer anschließend 16 Ideen in Tandems und holen sich im Rahmen eines Sounding Boards<sup>396</sup> Feedback von Experten. Von den 16 Ideen werden durch den Projektpartner 8 ausgewählt, für welche in Teams bestehend aus 4 Teilnehmern eine Konzept- und Funktionsprototypenentwicklung durchgeführt wurde. Die von den Teilnehmern erarbeiteten Ergebnisse werden im Rahmen von Meilensteinen am Ende jeder Phase anderen Teilnehmern, den Projektbetreuern und dem Projektpartner vorgestellt.

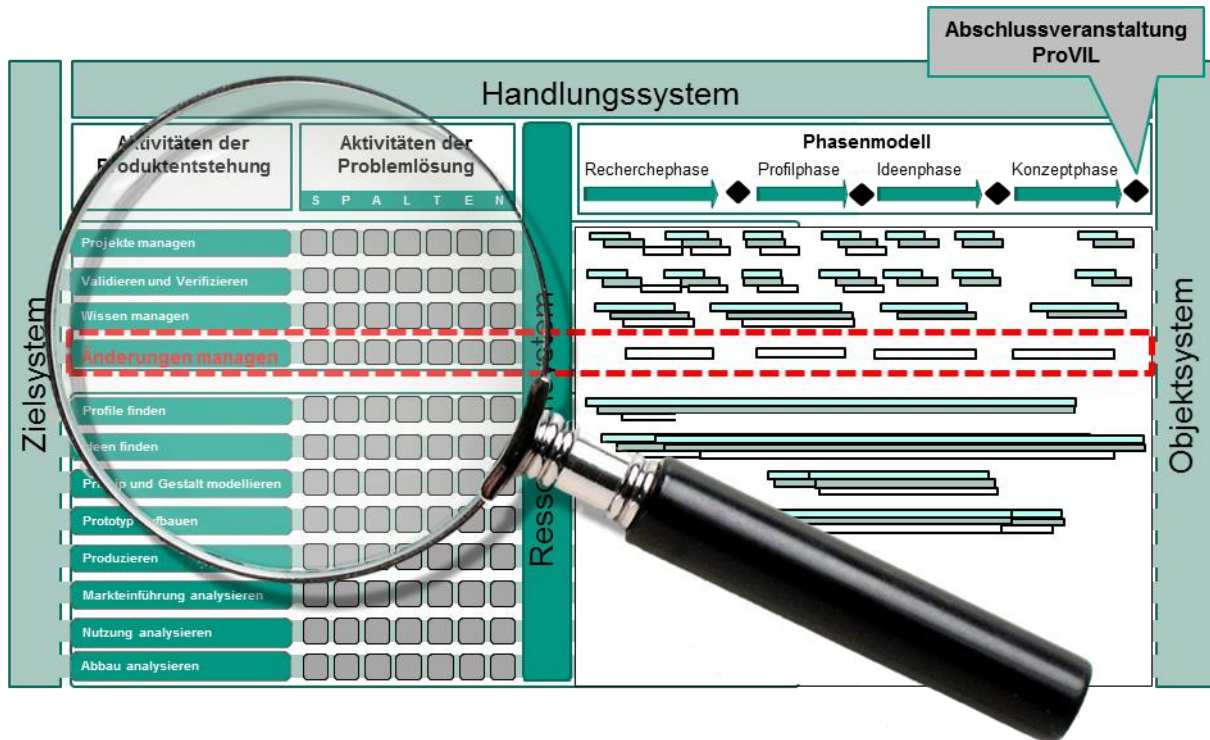


Abbildung 87: Phasen und Aktivitäten im Rahmen von ProVIL dargestellt im iPeM – Untersuchungsgegenstand im Rahmen von ProVIL

Die Arbeit der Studierenden wird in jeder der genannten Phasen durch eine Web-basierte Innovationsplattform (vgl. Abbildung 88) unterstützt. Dort haben die Studenten die Möglichkeit, ihre Produktprofile, Ideen und Konzepte im Team zu teilen und iterativ weiterzuentwickeln.

Zur Evaluation des auf SPALTEN basierten Ansatzes für das Management von Änderungen wurden die Teilnehmer von ProVIL durch einen Fragebogen bestehend aus 12 Fragen<sup>397</sup> zu Auftreten und Umgang mit Änderungen befragt. Der Fragebogen wurde im Tandem beantwortet.

<sup>396</sup> vgl. Walter, Albers, Schelleis und Kurrle 2016

<sup>397</sup> Siehe Anhang

Insgesamt haben 30 der 32 Teilnehmer von ProVIL an der Befragung teilgenommen. Die Studenten befinden sich durchschnittlich im 10. Semester. Alle Befragten haben bereits Erfahrungen in der Industrie zum Beispiel im Rahmen eines Praktikums, einer Abschlussarbeit oder im Rahmen der Lehrveranstaltung Integrierte Produktentwicklung des IPEK - Institut für Produktentwicklung sammeln können. Der Fragebogen wurde zu Projektabschluss den Studenten in Papierform ausgehändigt, wobei die Fragen im Projektteam beantwortet wurden.

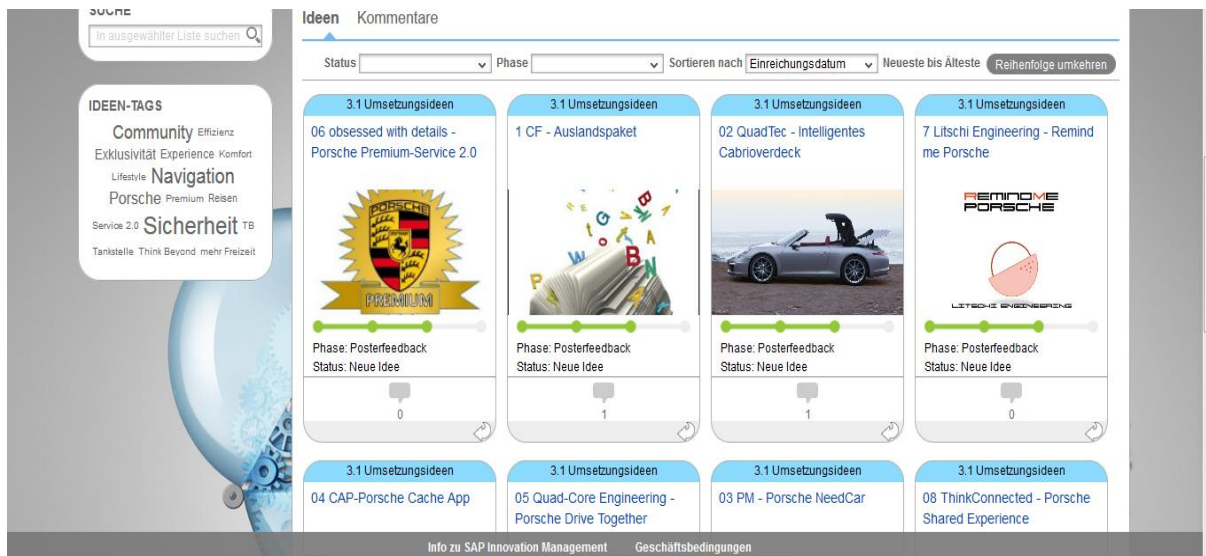


Abbildung 88: Die SAP Innovationsplattform von ProVIL

### 7.2.2 Ergebnisse und Implikationen<sup>398</sup>

Zunächst wurden die Teilnehmer befragt, in welchen Projektphasen Verantwortungsbereiche definiert wurden, um die Arbeitspakete einzelner Teilnehmer voneinander abzugrenzen. Dabei kann beobachtet werden, dass mit zunehmendem Fortschreiten der Entwicklung auch mehr Teams Verantwortungsbereiche definiert haben (vgl. Abbildung 89). Dies ist ein Anzeichen dafür, dass mit zunehmendem Fortschreiten der Entwicklung und damit auch steigender Komplexität in der Entwicklung der Bedarf nach einer klar definierten Arbeitsteilung vorhanden ist. Dies impliziert, dass Kollaboration in späteren Phasen der Entwicklung eine noch größere Bedeutung zuwächst.

<sup>398</sup> Vollständige Ergebnisse im Anhang

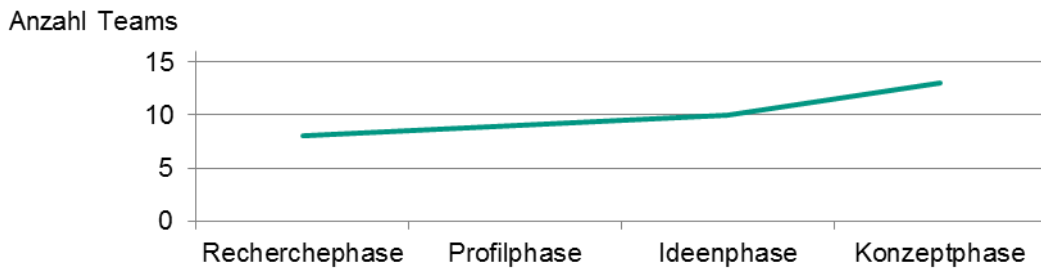


Abbildung 89: In welchen Projektphasen wurden Verantwortungsbereiche vergeben?

Weiter wurden die Teilnehmer aufgefordert, die Projektphasen nach Häufigkeit auftretender Änderungen zu sortieren, welche einen in einer früheren Phase definierten Soll-Zustand betroffen haben. Dabei wurde die Ideenphase von der Mehrzahl der Teilnehmer als die Phase genannt, in der Änderungen aufgetreten sind. In der Konzeptphase konnte folglich ein Rückgang von Änderungen beobachtet werden. Die Teilnehmer haben angegeben, dass die IPEK Betreuer und das eigene Team als größte Ursache von Änderungen gesehen wurde. Am seltensten wurden Änderungen auf Grund von Einflüssen anderer Teams verursacht.

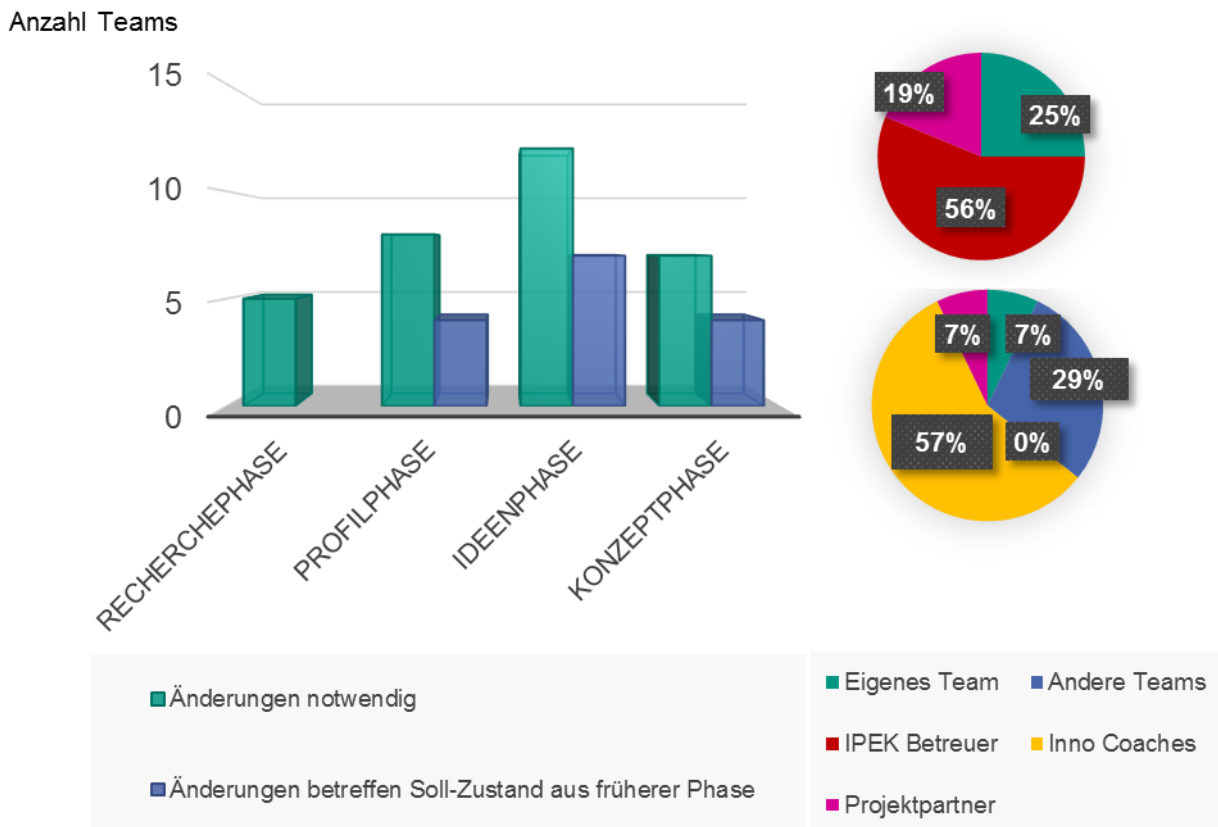


Abbildung 90: Bewertung der Häufigkeit von Änderungen in den Projektphasen

### **Bewertung der Aktivitäten des Managements von Änderungen**

Abbildung 91 zeigt die Angaben der Teilnehmer zur Verwendung der Aktivitäten des Managements von Änderungen durch die Tandems. Jede durch die Methodik bereitgestellte Aktivität wurde mindestens von zwei Drittel der befragten Tandems verwendet. Daraus kann gefolgert werden, dass keine Aktivität als überflüssig gesehen werden kann.

Im Rahmen der Situationsanalyse kann beobachtet werden, dass nur rund zwei Drittel nach Identifikation einer Änderung noch weitere Informationen gesammelt haben. Dies lässt sich auf die überschaubare Projektgröße zurückführen.

14 von 15 Tandems haben während der Problemeingrenzung betroffene Partialmodelle identifiziert, jedoch nur 12 der 15 Tandems haben die Änderung auch definiert. Am häufigsten wurden Änderungen dabei mündlich (11 von 15) oder per Skype/E-Mail (6 von 15) diskutiert. Die Innovationsplattform wurde hierfür nur selten genutzt (3 von 15). Am häufigsten wurden Änderungen am Produktprofil (9 von 15) und an Screeninhalten (8 von 15) vorgenommen.

Alternative Lösungen wurden nur von 12 der 15 Tandems gesucht. Bei einigen Teams war die Lösung zur Durchführung der Änderung auf Grund der Projektgröße häufig klar. Handlungsalternativen wurden am häufigsten mündlich definiert (11 von 15).

Vor Umsetzung einer Änderungen wurden von den meisten Teams benötigte Ressourcen identifiziert (14 von 15). Dabei wurde insbesondere geprüft, welches Teammitglied die Änderung umsetzen muss (12 von 15). Die technische Umsetzbarkeit und Bewertung der Verfügbarkeit von Ressourcen wurde weniger häufig geprüft (10 von 15 bzw. 11 von 15). Dies lässt sich darauf zurückführen, dass eine Umsetzbarkeit im Prototyp häufig bereits bei der Identifikation von Handlungsalternativen berücksichtigt werden konnte.



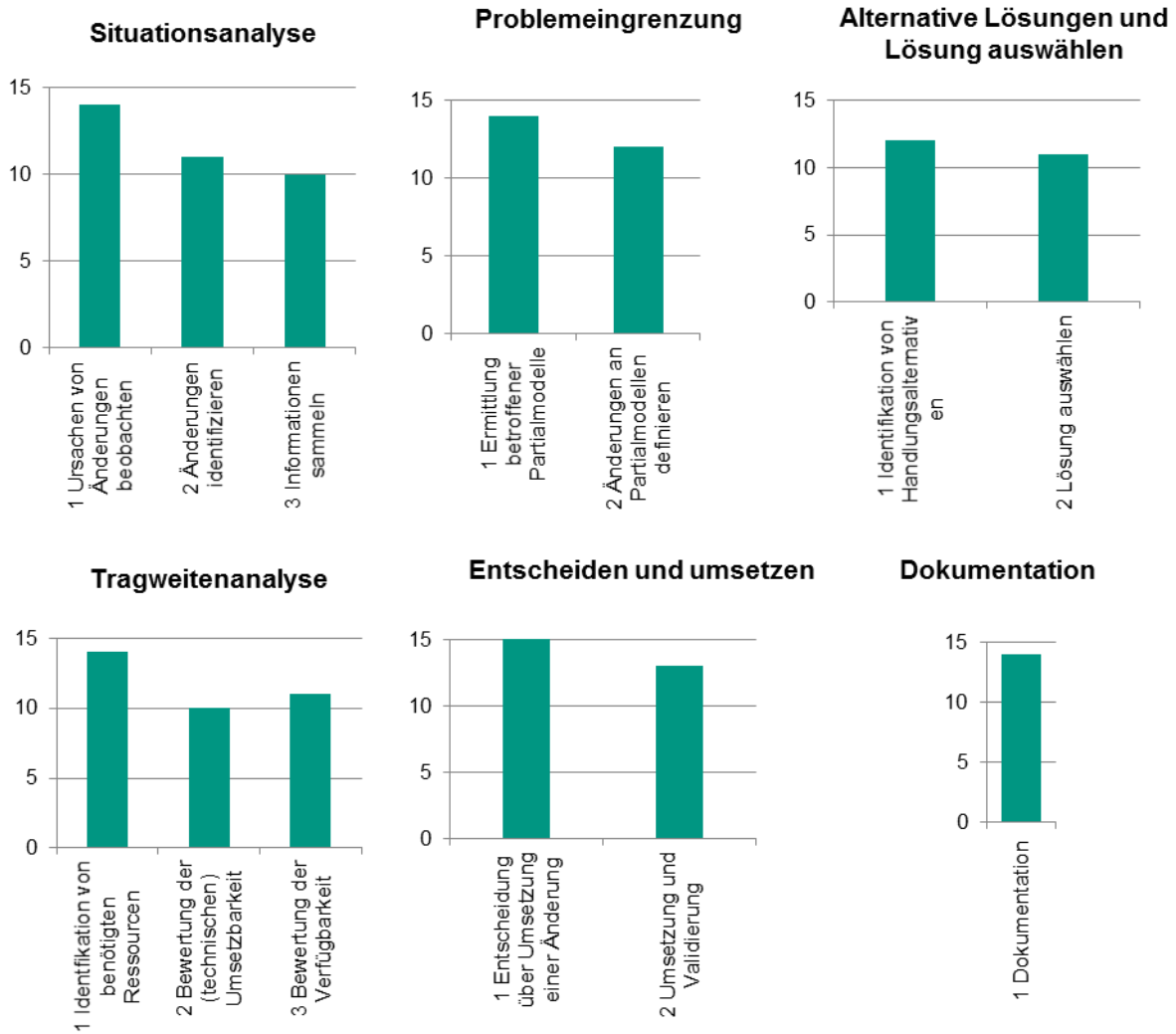


Abbildung 91: Bewertung der Verwendung von Aktivitäten des Managements von Änderungen

Alle Teams haben angegeben, dass über die Umsetzung einer Änderung entschieden wurde. Dabei wurde am häufigsten im Team selbst oder mit den IPEK Betreuern entschieden. Nur 7 von 15 Tandems haben angegeben, dass die Umsetzung einer Änderung auch validiert wurde.

Änderungen wurden dabei grundsätzlich nach Umsetzung dokumentiert (14 von 15), jedoch haben die Teilnehmer angegeben, dass nie eine Begründung für eine Änderung mitdokumentiert wurde.

### Gesamtbewertung des Ansatzes

14 der 15 befragten Tandems erachten den auf SPALTEN basierten Ansatz für das Management von Änderungen als Grundsätzlich geeignet (vgl. Abbildung 92).

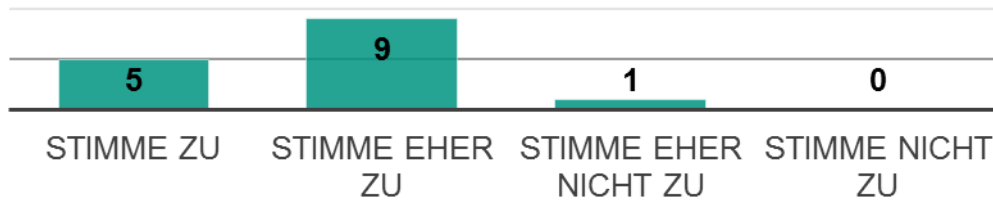


Abbildung 92: Bewertung der Eignung der Methodik für das Management von Änderungen

Teilnehmer haben dabei die vorgegebene Struktur (von 7 der Insgesamt 15 Tandems genannt) und eine Zeiteinsparung (5 Nennungen) positiv hervorgehoben. Ein einzelner Teilnehmer hat angemerkt, dass mit der Methodik „alle Handlungsschritte abgedeckt“ sind und das „analytische Vorgehen“ beim Management von Änderungen unterstützt wird. Dabei heben einige Teilnehmer hervor, dass durch die Vorgabe von Aktivitäten an die Durchführung von bestimmten Schritten „erinnert“ wird (4 Nennungen).

Verbesserungspotentiale sehen die Teilnehmer insbesondere darin, die Methodik in ein workflowbasiertes Tool zu überführen, welches die Schritte teilweise „automatisch“ durchführt (2 Nennungen). Zwei Teilnehmer sehen die Methode nicht für alle Projekttypen und Teamgrößen geeignet. Ein Teilnehmer hat dabei angemerkt, dass die Methode „zu umfangreich für ein kleines Projekt und kleines Team wie unseres“ sei. Dies unterstreicht die Forderung, dass die bereitgestellten Aktivitäten lediglich als eine Handlungsempfehlung und nicht als eine strikte Vorgabe gesehen werden darf.

## 8 Zusammenfassung und Ausblick

Dieses Kapitel schließt die Arbeit mit einer Zusammenfassung der wesentlichen Ergebnisse ab und legt mit einem Ausblick Anknüpfungspunkte für weitere Forschungsaktivitäten offen.

### 8.1 Zusammenfassung

In der vorliegenden Arbeit wird ein neuer Ansatz für die durchgängige Entwicklung und Dokumentation von Zielsystemen mit semantischen Metainformationen in verteilten und teilweise unabhängig voneinander agierenden Entwicklungsumgebungen vorgestellt. Die Arbeit knüpft damit an bestehende Arbeiten der IPEK - Forschungslandschaft und der KaSPro - Karlsruher Schule der Produktentwicklung an und erweitert diese um Erkenntnisse und Ansätze bei der verteilten Entwicklung und Abbildung von Zielsystemen über mehrere Produktgenerationen in einem Systems-of-Systems Kontext.

Die Entwicklung des Connected Car ist ein noch vergleichsweise neues Entwicklungsfeld für die Automobilhersteller. Dabei steht nicht mehr nur die Entwicklung des Fahrzeugs allein im Fokus, sondern dieses wird Teil eines komplexen System-of-Systems. Dadurch entstehen neue Herausforderungen in der Entwicklung, weshalb bestehende Methoden, Prozesse und Tools der Produktentstehung (siehe Kapitel 2) für die Entwicklung des Connected Car weiterentwickelt werden müssen.

Abgeleitet aus dem Stand der Forschung werden drei zentrale Forschungsfragen formuliert, welche das Forschungsziel definieren (siehe Kapitel 3):

- Welche Anforderungen, Ziele und Randbedingungen werden an einen Ansatz zur Entwicklung von Zielsystemen für SoS aus der Perspektive der Entwickler gestellt?
- Wie muss ein Ansatz zur Unterstützung der durchgängigen Entwicklung von Zielsystemen bei verteilten und teilweise unabhängig voneinander agierenden Handlungssystemen über Produktgenerationen hinweg gestaltet sein?
- Welchen Beitrag kann der Ansatz zur Herstellung von Durchgängigkeit für die Abbildung von Zielsystemen in der Praxis liefern?

Die Arbeit ist im Rahmen einer dreijährigen Forschungsk Kooperation zwischen dem IPEK – Institut für Produktentwicklung und dem Bereich Porsche Connect der Dr. Ing. h.c. F. Porsche AG entstanden. Auf Grund einer Vielzahl involvierter, teilweise unabhängig voneinander operierender Organisationseinheiten und der

starken Ausprägung der Produktgenerationsentwicklung bei der Porsche AG eignet sich dieses Fallbeispiel für die wissenschaftlichen Untersuchungen besonders. Das Forschungsvorgehen orientiert sich an der Design Research Methodology und ist in drei Teile untergliedert (siehe auch Kapitel 4)<sup>399</sup>.

- Deskriptive Studie I: Verständnis über die verteilte Entwicklung und Dokumentation von Zielsystemen
- Präskriptive Studie: Gestaltung eines Ansatzes zur Verbesserung der Durchgängigkeit und Konsistenz
- Deskriptive Studie II: Evaluation des Ansatzes

Anforderungen, Ziele und Randbedingungen an einen Ansatz werden in Kapitel 5 im Rahmen der ersten deskriptiven Studie aus vier empirischen Untersuchungen abgeleitet. Eine Befragung von Experten unterschiedlicher Automobilhersteller, Dienstleister und Zulieferer aus dem Bereich Connected Car zeigt, dass die Herausforderungen bei der Entwicklung von Zielsystemen sehr vielfältig sind (siehe Kapitel 5.1). Dabei wird offengelegt, dass viele bereits in der Produktentwicklung bekannte Herausforderungen im Kontext Connected Car eine noch stärkere Bedeutung erlangen. Die Experten sehen insbesondere darin eine Herausforderung, die Vielzahl von involvierten und teilweise unabhängig arbeitenden Entwicklungsteams mit unterschiedlichen Methoden, Prozessen und Tools zu integrieren. Häufige Änderungen mit weitreichenden Auswirkungen sind die Regel und bedingen eine Kombination aus zentralem und dezentralem Management von Änderungen. Auf Grund der verteilten Dokumentation und Validierung werden übergreifende Ansätze benötigt, welche Durchgängigkeit herstellen und den Informationsaustausch unterstützen. Des weiteren wird gezeigt, dass sich Entwickler eine stärkere Ausnutzung der Potentiale einer Produktgenerationsentwicklung wünschen.

Um die Anwendbarkeit des Ansatzes in der industriellen Praxis sicherzustellen, werden zwei weitere Untersuchungen bei der Dr. Ing. h.c. F. Porsche AG im Bereich Porsche Connect durchgeführt. Im Rahmen von teilnehmenden Beobachtungen wird zunächst ein grundlegendes Verständnis über die Entwicklung von Zielsystemen bei Systems-of-Systems geschaffen und Ursachen für den Verlust von Durchgängigkeit aufgezeigt (siehe Kapitel 5.2). Als wesentliche Bausteine zur Herstellung von Konsistenz im Zielsystem wird neben einer durchgängigen Dokumentation ein integriertes Management von Änderungen gefordert. Ein Ansatz muss dabei Durchgängigkeit über alle Entwicklungsphasen, unterschiedliche Produktgenerationen, Projekte und Detaillierungsgrade und zur Dokumentation

---

<sup>399</sup> vgl. Blessing & Chakrabarti 2009

angrenzender Partialmodelle (z.B. Testfälle) schaffen. Weiter wurden Randbedingungen identifiziert, welche eine Homogenisierung von Tools zur durchgängigen Abbildung von Zielsystemen (z.B. Gesamthaft durch einen modellbasierten Ansatz) bei allen beteiligten Entwicklungsteams in einem SoS Umfeld deutlich erschweren, weshalb ein integrierender Ansatz auf Basis semantischer Metainformationen gefordert wird.

Im Rahmen einer weiteren Untersuchung bei der Dr. Ing. h.c. F. Porsche AG wurde eine Inhaltsanalyse von Entwicklungsartefakten für die Entwicklung eines mobilen Online-Dienstes durchgeführt. Dabei werden Erkenntnisse darüber gesammelt, wie Ziele und Anforderungen in der Praxis dokumentiert und in einem verteilten Umfeld verlinkt werden (siehe Kapitel 5.3). Damit wird die Grundlage für den Aufbau einer Struktur von semantischen Metainformationen für Entwicklungsartefakte geschaffen. Dazu wird dargestellt, welche Relationen zwischen Entwicklungsartefakten und deren Elementen existieren können. Es wird weiter festgestellt, dass Entwickler nicht alle Relationen explizit dokumentieren, sondern viele Relationen implizit in den Köpfen der Entwickler sind und nicht allen Beteiligten transparent gemacht werden. Dabei konnten vier Typen von impliziten Relationen identifiziert werden.

Im Rahmen von Gruppendiskussionen während eines regelmäßig tagenden Arbeitskreises mit Industrievertretern von Automobilherstellern und Unternehmensberatungen wird das Verständnis über die Definition einer Änderung in der Produktentwicklung geschärft und es werden 10 Anforderungen an das Management von Änderungen in einem verteilten und teilweise unabhängig agierenden Umfeld definiert (siehe Kapitel 5.4).

Aus den Erkenntnissen der ersten deskriptiven Studie werden vier Ziele an den Ansatz abgeleitet (vgl. Kapitel 6):

**Z-1: *Integration und nicht Ablösung bestehender Methoden, Prozesse und Tools in den verteilten Entwicklungsteams***

**Z-2: *Durchgängigkeit im Zielsystem muss über den Detaillierungsgrad, Produkt- und Entwicklungsgenerationen und unterschiedliche Projekte auch bei unterschiedlichen Produktgenerations- und Entwicklungszyklen hergestellt werden***

**Z-3: *Verwendung von Metainformationen bei Entwicklungsartefakten:***

- a. *zur Abbildung von Relationen bei heterogenen Entwicklungsartefakten*
- b. *als Basis für eine Ableitung von impliziten Relationen zwischen Entwicklungsartefakten*

**Z-4: Kombination aus *zentralem und dezentralem Management von Änderungen* mit dynamischer Zusammensetzung der Problemlösungsteams und Entscheidungsgremien**

In Kapitel 6 wird ein Ansatz zur durchgängigen Entwicklung von Zielsystemen bei verteilten und teilweise unabhängig voneinander agierenden Handlungssystemen vorgestellt, welcher die im Rahmen der ersten deskriptiven Studie ermittelten Anforderungen und Ziele erfüllen soll.

Der Ansatz ermöglicht die durchgängige Dokumentation partieller Zielsysteme durch Verknüpfung von Entwicklungsartefakten mit semantischen Metainformationen. Hierzu wurde ein Metadaten-Schema auf Basis der semantischen Technologie RDF entwickelt (siehe Kapitel 6.1). Das RDF Schema beinhaltet Attribute, welche eine eindeutige Referenzierung von Entwicklungsartefakten und Relationen zu anderen Entwicklungsartefakten, einer Funktions- oder Baustruktur, Stakeholdern, Entwicklungsphasen oder Produktgenerationen erlaubt. Aus den definierten RDF Attributen lassen sich eine Menge von impliziten Relationen logisch ableiten. Dazu wurden Regeln definiert, welche Ableitungen aus der Funktions- oder Baustruktur von Systemen, Stakeholder-Beziehungen, Relationen zwischen Entwicklungsartefakten oder aus Referenzprodukten anderer Produktgenerationen ermöglichen (siehe Kapitel 6.2).

Zur prozessualen Verankerung des Ansatzes in der Produktentwicklung wird ein auf der SPALTEN-Problemlösungsmethodik basierender und in das iPeM integrierter Ansatz für das Management von Änderungen in verteilten Entwicklungsteams vorgestellt (siehe Kapitel 6.3). Damit wird Entwicklern eine Anleitung an die Hand gegeben, welche sie beim Management von Änderungen in einem verteilten Umfeld unterstützt.

Abschließend wird der Ansatz prototypisch durch ein web-basiertes Software-Tool implementiert (siehe Kapitel 6.4). Die Software erlaubt die Anreicherung von Entwicklungsartefakten mit semantischen Metainformationen auf Basis RDF und implementiert die definierten Ableitungsregeln zur Verlinkung der Entwicklungsartefakte.

In Kapitel 7 werden die Ergebnisse von zwei weiteren, deskriptiven Studien vorgestellt, welche das Forschungsziel verfolgen, den entwickelten Ansatz anhand des aufgestellten Zielsystems zu evaluieren. Die Zielerfüllung ist zusammenfassend in Abbildung 93 dargestellt. Im ersten Teil werden die durch den Ansatz geschaffenen Potentiale zur Verbesserung von Durchgängigkeit und Konsistenz im Zielsystem sowie die Anwendbarkeit in der Praxis bewertet (siehe Kapitel 7.1). Dazu wird das Software-Tool durch Projektleiter bei der Dr. Ing. h.c. F. Porsche AG genutzt, um semantische Metainformationen für Entwicklungsartefakte von realen

Entwicklungsprojekten im Bereich Connected Car zu definieren. Die Projektleiter bewerten den Ansatz über einen strukturierten Fragebogen in vier Kategorien:

- Bewertung der Fähigkeit zur Integration der Methode mit bestehenden Methoden, Prozesse und Tools
- Bewertung der Erhöhung von Durchgängigkeit und Konsistenz durch Abbildung expliziter Relationen mit semantischen Metainformation
- Bewertung der Erhöhung von Durchgängigkeit und Konsistenz durch Ableitung impliziter Relationen aus den definierten Ableitungsregeln
- Bewertung der allgemeinen Anwendbarkeit des Ansatzes

Die Ergebnisse zeigen, dass durch Verwendung der definierten semantischen Metainformationen und Ableitungsregeln die Durchgängigkeit und Konsistenz durch Verlinkung der Entwicklungsartefakte und Ableitung von Relationen gesteigert werden kann. Die Teilnehmer der Studie bestätigen außerdem die Integrationsfähigkeit des Ansatzes mit heterogenen Methoden, Prozessen und Tools sowie die generelle Anwendbarkeit in der Praxis. Kritisch wird hierbei insbesondere der Pflegeaufwand gesehen. Es wird jedoch vermutet, dass durch zusätzliche Mechanismen in weiteren Ausbaustufen des Software-Tools (z.B. Autovervollständigung, Plausibilitätsprüfungen, weitere Inferenzregeln) dieser noch deutlich verringert werden kann. Trotz des zusätzlichen Aufwandes zur Pflege von Metainformationen bewerten alle Teilnehmer den Nutzen des Ansatzes höher als den Aufwand zur Pflege, weshalb die Teilnehmer die Nutzung des Ansatzes in ihrem Bereich befürworten.

Im zweiten Teil wird die Methode für das Management von Änderungen im Rahmen des Live-Lab ProVIL evaluiert (siehe Kapitel 7.2). Dabei haben 32 Studierende des Studiengangs Maschinenbau aus höheren Fachsemestern über einen Zeitraum von drei Monaten eine reale Entwicklungsaufgabe vom Projektpartner Dr. Ing. h.c. F. Porsche AG in einem gemeinsamen Entwicklungsprojekt bearbeitet. Zum Projektabschluss wurde durch einen Fragebogen erfasst, wie häufig Änderungen aufgetreten sind und wie mit diesen umgegangen wurde. Die definierten Aktivitäten der SPALTEN Methodik wurden dabei häufig von den Teilnehmern der Studie genutzt. Insgesamt bewerten über 90 Prozent der Teilnehmer der Studie den Ansatz für das Management von Änderungen als grundsätzlich geeignet.

Ziele		Erfüllung
Z1	Integration und nicht Ablösung bestehender Methoden, Prozesse und Tools in den verteilten Entwicklungsteams.	✓
Z2	Durchgängigkeit im Zielsystem muss über den Detaillierungsgrad, Produkt- und Entwicklungsgenerationen und unterschiedliche Projekte auch bei unterschiedlichen Produktgenerations- und Entwicklungszyklen hergestellt werden.	✓
Z3	Verwendung von Metainformationen bei Entwicklungsartefakten: a. zur Abbildung von Relationen bei heterogenen Entwicklungsartefakten b. als Basis für eine Ableitung von impliziten Relationen zwischen Entwicklungsartefakten	✓
Z4	Kombination aus zentralem und dezentralem Management von Änderungen mit dynamischer Zusammensetzung der Problemlösungsteams und Entscheidungsgremien.	✓

Abbildung 93: Erfüllung der definierten Ziele durch den entwickelten Ansatz

## 8.2 Ausblick

Die mit dieser Arbeit transportierten Erkenntnisse und Ansätze sind Ausgangsbasis für weiterführende Forschungsarbeiten.

Die Potentiale **semantischer Metainformationen** in der Produktentwicklung können durch weitere Forschung auch auf andere Bereiche übertragen werden.

Dabei muss insbesondere untersucht werden, wie die Hürde zur Pflege von Metainformationen in der Produktentwicklung weiter reduziert werden kann. Hierbei können folgende Forschungsfelder Anknüpfungspunkte für weitere Arbeiten liefern:

- Identifikation von Anreizen zur Pflege von Metainformationen in der Produktentwicklung
- Gestaltung einer an den Bedarfen der Nutzer ausgerichteten Benutzerschnittstelle (z.B. mit Autovervollständigung, Plausibilitätsprüfung)
- Automatisierte Extraktion von Metainformation aus Inhalten von Entwicklungsartefakten in die definierte semantische Struktur

Es sind weiter noch Untersuchungen notwendig, welche das Verhalten einer breiten Masse von Nutzern bei der Pflege untersucht. Hierfür könnte beispielweise die IPEK Projekterfassung erweitert werden.

Weiter sind noch Untersuchungen notwendig, wie Schnittstellen zu bestehenden Methoden Tools zur Abbildung von Zielsystemen (z.B. SysML, Rational Doors, Microsoft Word) geschaffen werden können, um eine bessere Verlinkung zu Inhalten der Entwicklungsartefakte und damit einer Anreicherung der Informationen geschaffen werden können.



Es wird erwartet, dass die entwickelten Ableitungsregeln zur Erkennung von impliziten Relationen sich auch auf modellbasierte Ansätze aus der Produktentwicklung<sup>400</sup> übertragen lassen. Die generelle Übertragbarkeit muss dabei jedoch noch untersucht werden.

Die definierten Ableitungsregeln können dabei durch Erkenntnisse aus weiteren empirischen Untersuchungen in anderen Bereichen erweitert werden.

Im Bereich **Systems-of-Systems** besteht zusätzlicher Forschungsbedarf. So muss untersucht werden, wie sich Anforderungen und Ziele der Gesellschaft auf ein SoS übertragen lassen. Dabei sollten nicht nur technische Systeme wie das Fahrzeug sondern auch die Infrastruktur berücksichtigt werden.

Auf Grund unterschiedlicher Produktgenerationslebenszyklen entstehen nicht nur Herausforderungen bei der Entwicklung von Zielsystemen. Einflüsse auf andere Entwicklungsbereiche müssen noch untersucht werden. Weiter fehlt es noch an Unterstützung, wie Entwicklungsgenerationen aufgeplant werden müssen (z.B. Dauer der Entwicklung von Generationen).

Die gewonnenen Erkenntnisse müssen für die weitere Forschung im Bereich der **Validierung von Systems-of-Systems** genutzt und weiter ausgebaut werden. Dabei muss insbesondere noch untersucht werden, in welchen Bereichen die bestehenden Ansätze weiterentwickelt werden müssen.

Die weitere Übertragung von Ansätzen einer **Produktgenerationsentwicklung** auf den Bereich **Connected Car**, zum Beispiel durch Entwicklung neuer oder Übertragung bestehender Entwicklungsmethoden, sowie Definition von Modulen und Baukästen muss untersucht werden.

---

<sup>400</sup> vgl. u.a. Zingel 2013, Ebel 2015, Bursac 2016, Scherer 2016

## 9 Literaturverzeichnis

### **Abramovici et al. 2009**

Abramovici, M., Neubach, M., Schulze M. und Spura, C. (2009) Metadata Reference Model for IPS 2 Lifecycle Management. In Proceedings of the 1<sup>st</sup> CIRP Industrial Product-Service Systems (IPS2) Conference, Cranfield University.

### **Abramovici 2015**

Abramovici, M. (2015) Smart Products. In CIRP Encyclopedia of Production Engineering.

### **Abramovici et al. 2016**

Abramovici, M., Göbel, C. und Dang, H. B. (2016) Semantic data management for the development and continuous reconfiguration of smart products and systems. In CIRP Encyclopedia of Production Engineering.

### **Albers 1991**

Albers, A. (1991) Das Zweimassenschwungrad Der Dritten Generation-Optimierung Der Kom-Forteigenschaften von PKW-Antriebssträngen. In Antriebstechnisches Kolloquium Bd. 91.

### **Albers 2010**

Albers, A. (2010) Five Hypotheses and a Meta Model of Engineering Design Processes. In 8th International Symposium on Tools and Methods of Competitive Engineering TMCE 2010. Ancona.

### **Albers, Behrendt & Ott 2006**

Albers, A., Behrendt, M. und Ott, S. (2006) Systemtribometer zur Untersuchung hochbelasteter nasslaufender Frikionskontakte unter mehrachsigen Gleitbewegungen. In Tribologie Fachtagung 2005 – Reibung, Schmierung und Verschleiß, Gesellschaft für Tribologie (GfT). Göttingen, 2005

### **Albers, Braun & Muschik 2010**

Albers, A., Braun, A. und Muschik, S. (2010) Uniqueness and the Multiple Fractal Character of Product Engineering Processes. In Heisig, P., Clarkson, P. J., Vajna, S. (Hrsg.) Modelling and Management of Engineering Processes. Springer, London, S. 14-26.

### **Albers & Braun. 2011**

Albers, A. und Braun, A. (2011) A Generalized Framework to Compass and to Support Complex Product Engineering Processes. In International Journal of Product Development, Bd. 15, Nr. 1/2/3, S. 6-25.

### **Albers, Braun & Schmalenbach 2012**

Albers, A., Braun, A. und Schmalenbach, H (2012) An ontology representation of the integrated product engineering model. In 14th International dependency and structure modelling conference, Kyoto.

### **Albers & Bursac 2015**

Albers, A. und Bursac, N. (2015) Produktgenerationsentwicklung – Bedeutung Und Herausforderungen Aus Einer Entwicklungsmethodischen Perspektive. In Stuttgarter Symposium der Produktentwicklung (SSP) 2015.

### **Albers, Bursac & Wintergerst 2015**

Albers, A., Bursac, N. und Wintergerst, E. (2015) Produktgenerationsentwicklung – Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. In Stuttgarter Symposium für Produktentwicklung SSP 2015, Stuttgart.

**Albers, Ebel & Sauter 2010**

Albers, A., Ebel, B. und Sauter, C. (2010) Combining Process Model and Semantic Wiki. In 11th International Design Conference DESIGN2010, Dubrovnik.

**Albers, Ebel & Lohmeyer 2012**

Albers, A., Ebel, B. und Lohmeyer, Q. (2012) Systems of Objectives in Complex Product Development. In 9th International Symposium on Tools and Methods of Competitive Engineering TMCE 2012, Karlsruhe.

**Albers et al. 2005**

Albers, A., Burkardt, N., Meboldt, M. und Saak, M. (2005) SPALTEN Problem Solving in the Product Development. In 15th International Conference on Engineering Design ICED'05, Melbourne, Australia.

**Albers et al. 2007**

Albers, S., Klapper, D., Konradt, U., Walter, A. und Wolf, J. (2007) Methodik der empirischen Forschung. 2. Auflage. Wiesbaden. Gabler Verlag.

**Albers et al. 2014a**

Albers, A., N. Bursac, J. Urbanec, R. Lüdcke, and G. Rachenkova. 2014. "Knowledge Management in Product Generation Development: An Empirical Study." In Proceedings of the 24th Symposium Design for X.

**Albers et al. 2014b**

Albers, A., Lüdcke, R., Bursac, N. und Reiß, N. (2014) Connecting Knowledge-Management-Systems to Improve a continuous flow of Knowledge in Engineering Design Processes. In 10th International Symposium on Tools and Methods of Competitive Engineering.

**Albers et al. 2015**

Albers, A., Bursac, N., Walter, B., Hahn, C. und Schröder, J. (2015) ProVIL – Produktentwicklung Im Virtuellen Ideenlabor. In Entwerfen, Entwickeln, Erleben : Werkzeuge und Methoden in Produktionsentwicklung und Design, Dresden.

**Albers et al. 2016**

Albers, A., Reiss, N., Bursac, N. und Richter, T. (2016) The Integrated Product Engineering Model (iPeM) in Context of the Product Generation Engineering. In 26th CIRP Design Conference.

**Albers et al. 2017**

Albers, A., Reussner, R., Kurrle, A., Burger, E., Moeser, G., Bursac, N., Klingler, S. und Behrendt, M. (2017) Continuity in the development of seamless mobility: An approach for a system-of-systems environment. In International Research Journal of Engineering and Technology (IRJET).

**Albers, Klingler & Ebel 2013**

Albers, A., Klingler, S. und Ebel, B. (2013) Modeling Systems of Objectives in Engineering Design Practice. In Proceedings of the 19th International Conference on Engineering Design ICED'13, Seoul.

**Albers, Kurrle & Moeser 2014**

Albers, A., Kurrle, A. und Moeser, G. (2014) Modellbasiertes Anforderungsmanagement von Systems- of-Systems Am Beispiel Des Vernetzten Fahrzeugs. In Tag des Systems Engineering (TdSE), Bremen.

**Albers, Kurrle & Klingler 2016**

Albers, A., Kurrle, A. und Klingler, S. (2016) The Connected Car – A system-of-systems : Exploration of challenges in development from experts view. In Internationales Stuttgarter Symposium (FKFS).

**Albers, Kurrle & Munker 2016**

Albers, A., Kurrle, A. und Munker, F. (2016) Das Auto Und Die Cloud - Ein System-of-Systems. In WiGeP News 1/2016.

**Albers, Lohmeyer & Schmalenbach 2010**

Albers, A., Q Lohmeyer, and H Schmalenbach (2010) Integration of Validation Activities in Ontology Development Processes. In 8th International Symposium on Tools and Methods of Competitive Engineering TMCE 2010, Ancona.

**Albers & Lohmeyer 2012**

Albers, A. und Lohmeyer, Q. (2012) Advanced Systems Engineering – Towards a Model-Based and Human-Centered Methodology. In 9th International Symposium on Tools and Methods of Competitive Engineering TMCE 2012, Karlsruhe.

**Albers, Lohmeyer & Ebel 2011**

Albers, A., Lohmeyer, Q. und Ebel, B. (2011) Dimensions of Objectives in Interdisciplinary Product Development Projects. In Proceedings of the 18th International Conference on Engineering Design ICED'11, Copenhagen, Denmark, Vol.2, S. 256-265.

**Albers, Muschik & Ebel 2010**

Albers, A., Muschik, S. und Ebel, B. (2010) Einflüsse auf Entscheidungsprozesse in frühen Aktivitäten der Produktentstehung. In Symposium für Vorausschau und Technologieplanung, Berlin.

**Albers & Matthiesen 2002**

Albers, A., Matthiesen, S. (2002) Konstruktionsmethodisches Grundmodell zum Zusammenhang von Gestalt und Funktion technischer Systeme - Das Elementmodell "Wirkflächenpaare & Leitstützstrukturen" zur Analyse und Synthese technischer Systeme. In Konstruktion, Zeitschrift für Produktentwicklung Band 54, Heft 7/8.

**Albers & Sadowski 2014**

Albers, A. und Sadowski, E. (2014) The Contact and Channel Approach (C&C<sup>2</sup>-A) – relating a system's physical structure to its functionality. In Chakrabarti, A. und Blessing, L. (Hrsg.) An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. Springer.

**Albers, Walch & Bursac 2016**

Albers, A., Walch, M. und Bursac, N. (2016) Entscheidungsunterstützung durch die Variationsanteile der Produktgenerationsentwicklung. In Konstruktion - Zeitschrift für Produktentwicklung und Ingenieur-Werkstoffe 1.

**Albers & Meboldt 2007**

Albers, A. und Meboldt, M. (2007) IPEMM – Integrated Product Development Process Management Model Based on Systems Engineering and Systematic Problem Solving. In 16th International Conference on Engineering Design ICED'07, Paris, France, No. 537.

**Albers & Zingel 2013**

Albers, A. und 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. Atlanta, Georgia, USA.

**Alt 2012**

Alt, O. (2012) Modellbasierte Systementwicklung mit SysML. Carl Hanser, München.

**Ashby 1974**

Ashby, W. R. (1974) Einführung in die Kybernetik. Suhrkamp, Frankfurt am Main.

**Atteslander 2010**

Atteslander, P. (2010) Methoden Der Empirischen Sozialforschung. Erich Schmidt Verlag, 13. Auflage.

**Bader 2007**

Bader, J. (2007) Die Hauptstellgrößen des Produktzielmanagements am Automobilentwicklung. Dissertation, TU Dresden.

**Bailom et al. 1996**

Bailom, F., Hinterhuber, H., Matzler, K. und Sauerwein, E. (1996) Das Kano-Modell der Kundenzufriedenheit. In Marketing ZFP 18:117–126.

**Beck 1999**

Beck, K. (1999) Extreme Programming. Das Manifest. Addison-Wesley.

**Beckett et al. 2014**

Beckett, D., Berners-Lee, T., Prud'hommeaux, E. und Carothers, G. (2014) RDF 1.1 Turtle.

**Berners-Lee 2005**

Berners-Lee, T. (2005) Notation 3 Logic.

**Berners-Lee et al. 2007**

Berners-Lee, T., Connolly, D., Kagal, L., Scharf, Y. und Hendler, J. (2007) N3Logic: A Logical Framework For the World Wide Web. In Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology.

**Bertalanffy 1969**

Bertalanffy, L. (1969) General System Theory – Foundations, Development, Applications. George Braziller, New York.

**Birkhofer & Jänsch 2003**

Birkhofer, H. und Jänsch, J. (2003) Interaction between Individuals. In Lindemann, U. (Hrsg.) Human Behaviour in Design – Individuals, Teams, Tools, Springer, Berlin, S. 105-110.

**Blessing & Chakrabarti 2009**

Blessing, L. T. und Chakrabarti, A. (2009) DRM, a Design Research Methodology. Springer, London.

**Boehm & Turner 2004**

Boehm, B. und Turner, R. (2004) Balancing Agility and Discipline. Addison-Wesley, Boston.

**Bolzano 1810**

Bolzano, B. (1810) Beiträge Zu einer begründeten Darstellung der Mathematik, Widtmann.

**Boulding 1956**

Boulding, K. E. (1956) General Systems Theory--The Skeleton of Science. In Management Science, 2:197–208. doi:10.1287/mnsc.2.3.197.

**Braun, Ebel, Albers 2013**

Albers, A., Braun, A., Ebel, B. (2013) Aktivitätenbasierte Analyse von Produktentstehungsprozessen. In Maurer, M.; Schulze, S.-O. (Hrsg.) Tag des Systems Engineerings 2013. Carl Hanser Verlag, München, S. 415-424.

**Bursac 2016**

Bursac, N. (2016) Model Bases Systems Engineering Zur Unterstützung Der Baukastenentwicklung Im Kontext Der Frühen Phase Der Produktgenerationsentwicklung. Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Cockburn 2005**

Cockburn, A. (2005) Crystal Clear. Addison-Wesley.

**Cooper 1990**

Cooper, R. G. (1990) Stage-Gate Systems: A New Tool for Managing New Products. In Business Horizons, Vol. 33, No. 3, S.44-54.

**Cyganiak, Wood & Lanthaler 2014**

Cyganiak, R., Wood, D. und Lanthaler, M. (2014) RDF 1.1 Concepts and Abstract Syntax.

**Daenzer & Huber 1996**

Daenzer, W. F. und Huber, F. (1996) Systems Engineering. Orell Füssli Verlag, 10. Auflage.

**Dahmann 2015**

Dahmann, J. S. (2015) Systems of Systems Characterization and Types. In Proceedings of 4th International Conference on Complex Systems Design & Management, Paris.

**Dahmann, Kristen & Baldwin 2008**

Dahmann, J. S., Kristen J. und Baldwin (2008) Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering. In Proceedings of 2nd Annual IEEE Systems Conference.

**Darlington & Culley 2002**

Darlington, M.J. and Culley, S.J. (2002) Current Research in the Engineering Design Requirement. In Journal of Engineering Manufacture, Vol.216, No.3 pp.375-388.

**Deming 1982**

Deming, W. E. (1982) Out of the Crisis. In MIT Press, Cambridge, MA.

**Dengel 2012**

Dengel, A. (2012) Semantische Technologien. SpringerLink Bücher. Kaiserslautern: Spektrum Akademischer Verlag Heidelberg.

**Deubzer & Lindemann 2009**

Deubzer, F. und Lindemann, U. (2009) Networked Product Modelling - Use and Interaction of Product Models and Methods during Analysis and Synthesis. In Proceedings of the 17th International Conference on Engineering Design.

**Ebel 2015**

Ebel, B. (2014) Modellierung von Zielsystemen in Der Interdisziplinären Produktentstehung. Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Ebert 2014**

Ebert, C. (2014) Systematisches Requirements Engineering: Anforderungen Ermitteln, Dokumentieren, Analysieren und Verwalten. Dpunkt Verlag, Heidelberg.

**Eckert, Alink & Albers 2010**

Eckert, C. M., Alink, T. und Albers, A. (2010) Issue Driven Analysis of an Existing Product at Different Levels of Abstraction. In Proceedings of The International DESIGN Conference, Dubrovnik.

**Ehrlenspiel 2009**

Ehrlenspiel, K. (2009) Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit. 4. Auflage, Hanser, München.

**Eiletz 1999**

Eiletz, R. (1999) Zielkonfliktmanagement bei der Entwicklung komplexer Produkte – am Beispiel PKW- Entwicklung. Shaker, Aachen. Dissertation, TU München.

**Fengel, Rebstock & Nüttgens 2008**

Fengel, J., Rebstock, M. und Nüttgens, M. (2008) Modell-Tagging Zur Semantischen Verlinkung Heterogener Modelle. Emisa 2008.

**Gandon & Schreiber 2014**

Gandon, F. und Schreiber, G. (2014) RDF 1.1 XML Syntax.

**Gartner 2015**

Meulen, R. (2020) Gartner Says By 2020, a Quarter Billion Connected Vehicles Will Enable New In-Vehicle Services and Automated Driving Capabilities. Internet Artikel: <http://www.gartner.com/newsroom/id/2970017>, letzter Zugriff am 17.12.2016

**Gausemeier et al. 2013**

Gausemeier, J., Dumitrescu, R., Steffen, D., Czaja, A., Wiederkehr, O. und Tschirner, C. (2013) Systems Engineering in Der Industriellen Praxis. In Journal des 9. Paderborner Workshop: Entwurf mechatronischer Systeme.

**Gebauer 2000**

Gebauer, M. (2001) Kooperative Produktentwicklung auf Basis verteilter Anforderungen. Dissertation, Universität Karlsruhe (TH), Shaker Verlag, Aachen.

**Geisberger & Broy 2012**

Geisberger, E. und Broy, M. (2012) agendaCPS - Integrierte Forschungsagenda Cyber-Physical Systems. Acatech STUDIE. Springer.

**GfSE 2012**

GfSE. 2012. INCOSE Systems Engineering Handbuch. Version 3.

**Gilz & Eigner 2013**

Gilz, T und Eigner, M. (2013) Ansatz zur integrierten Verwendung von SysML Modellen in PLM zur Beschreibung der funktionalen Produktarchitektur In Maurer, M., Schulze, S.-O. (Hrsg.) Tag des Systems Engineerings 2013. Carl Hanser, München, S. 293-302.

**Gläser & Laudel 2006**

Gläser, J. und Gri, L. (2006) Experteninterviews Und Qualitative Inhaltsanalyse: Als Instrumente Rekonstruierender Untersuchungen, VS Verlag für Sozialwissenschaften, S. 39ff.

**Greniewski & Kempisty 1963**

Greniewski, H. und Kempisty, M. (1963) Kybernetische Systemtheorie ohne Mathematik. Dietz, Berlin.

**Gruber 1993**

Gruber, T. R (1993) A Translation Approach to Portable Ontology Specifications. In Knowledge Acquisition, Vol. 5, S. 199-220.

**Haberfeller et al. 2012**

Haberfeller, R., Weck, Fricke, E. und Vössner, S. (2012) Systems Engineering – Grundlagen und Anwendung. Orell Füssli, 12. Auflage, Zürich.

**Haffner 2012**

Haffner, A. (2012) Internationalisierung Der GND Durch Das Semantic Web. DINI-AG-KIM Workshop 2012, Mannheim.

**Hall & Fagen 1956**

Hall, A. D und Fagen, R. E. (1956) Definition of systems. General Systems 1.

**Hallerstede et al. 2012**

Hallerstede, S., Hansen, F., Holt, J. Lauritsen, R., Lorenzen, L. und Peleska, J. (2012) Technical Challenges of SoS Requirements Engineering. In Proceedings of 7th International Conference on System of Systems, S. 573–78.

**Haskins et al. 2011**

Haskins, C., Forsberg, K., Krueger, M., Walden, D. und Hamelin, R. (2011) INCOSE Systems Engineering Handbook. International Council on Systems Engineering.

**Held & Cress 2008**

Held, C. und Cress, U. (2008) Social Tagging Aus Kognitionspsychologischer Sicht. In Good Tags - Bad Tags. Social Tagging in Der Wissensorganisation. Waxmann Verlag GmbH.

**Henshaw 2013**

Henshaw, M.J.D. (2013) The Systems of Systems Strategic Research Agenda. In TAREA-PU-WP5-R-LU-26, Nr. 2.

**Hitzler et al. 2008**

Hitzler, P., Krötzsch, M., Rudolph, S., Sure, Y. (2008) Semantic Web – Grundlagen. Springer, Berlin, Heidelberg.

**Holt et al. 2012a**

Holt, J., Perry, S., Brownsword, M. und Ita, L. (2012) Model-Based Requirements Engineering for System of Systems. In Proceedings of 7th International Conference on System of Systems Engineering, S.561–566.

**Holt et al. 2012b**

Holt, J., Perry, S., Overgaard, F. und Hallerstede, S. (2012) Report on Guidelines for SoS Requirements, Compass Research.

**Hood et al. 2008**

Hood, C., Wiedemann, S., Fichtinger, S. und Pautz, U. (2008) Requirements Management. Springer Verlag, Berlin.

**IEEE 1990**

IEEE. (1990) IEEE Standard Glossary of Software Engineering Terminology.

**Jamshidi 2009**

Jamshidi, M. (2009) Systems of Systems Engineering. CRC Press.

**Johanning & Mildner 2015**

Johanning, V. und Mildner, R. (2015) Car IT Kompakt. Springer.

**Kano et al. 1984**

Kano, N., Seraku, N., Takahashi, F. und Tsuji, S. (1984) Attractive Quality and Must-Be Quality. In Journal of the Japanese Society for Quality Control 14.2.

**Kelle, Tobor & Metje 2007**

Kelle, U., Tobor, A. und Metje, B. (2007) Qualitative Evaluationsforschung Im Internet – Online - Foren Als Werkzeuge Interpretativer Sozialforschung, 178–92.

**Kim 2006**

Kim, K.-Y. (2006) Ontology-Based Assembly Design and Information Sharing for Collaborative Product Development. In Computer Aided Design, Volume 38, Issue 12.



**Klir 1972**

Klir, J. G. (1972) Trends in general systems theory. John Wiley & Sons Limited, Canada.

**Kruse 1996**

Kruse, P. J. (1996) Anforderungen in der Systementwicklung – Erfassung, Aufbereitung und Bereitstellung von Anforderungen in interdisziplinären Entwicklungsprojekten. Dissertation, TU Clausthal, VDI-Verlag, Düsseldorf.

**Krygiel 1999**

Krygiel, A. J. (1999) Behind the Wizard's Curtain: An Integration Environment for a System of Systems. In Technical Report, DTIC Document.

**Lange 1965**

Lange, O. (1965) Wholes and parts: A general theory of system behaviour. Pergamon, Oxford/London/Warschau.

**Leffingwell 2003**

Leffingwell, D. (2003) Managing Software Requirements: A Use Case Approach. Addison-Wesley.

**Lewis et al. 2009**

Lewis, G., Morris, E., Place, P., Simanta, S. und Smith, D. B. (2009) Requirements Engineering for Systems of Systems. In 3rd Annual IEEE Systems Conference.

**Liang & Paredis 2003**

Liang, V.-C. und Paredis, C. J. J. (2003) A Port Ontology for Automated Model Composition. In Proceedings of the 2003 Winter Simulation Conference.

**Lindemann & Reichwald 1998**

Lindemann, U. und Reichwald, R. (1998) Integriertes Änderungsmanagement. Springer, Berlin.

**Lindemann 2009**

Lindemann, U. (2009) Methodische Entwicklung technischer Produkte. 3. Auflage, Springer, Berlin, Heidelberg.

**Lohmeyer 2012**

Lohmeyer, Q. (2013) Menschzentrierte Modellierung von Produktentstehungssystemen unter besonderer Berücksichtigung der Synthese und Analyse dynamischer Zielsysteme. Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Luzeaux & Ruault 2010**

Luzeaux, D. und Ruault, J.-R. (2010) Systems of Systems. John Wiley & Sons, London.

**Maedche 2002**

Maedche, A. (2002) Ontology Learning for the Semantic Web. In The Springer International Series in Engineering and Computer Science.

**Maier 1998**

Maier, M. W. (1998) Architecting Principles for Systems-of-Systems. In Systems Engineering Volume 1, Issue 4, Willey.

**Marxen 2014**

Marxen, L. (2014) A Framework for Design Support Development Based on the Integrated Product Engineering Model iPeM. Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Maul 2015**

Maul, L. (2015) Vernetzte Kreativität - Menschzentrierte Gestaltung Und Integration Einer Community-Plattform Für Innovationsimpulse. Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Mayank et al. 2004**

Mayank, B., Vimal, N., Kositsyna und Austin, M. (2004) Requirements Engineering and the Semantic Web: Part II. Representation, Management, and Validation of Requirements and System-Level Architectures. In ISR Technical Report - University of Maryland.

**Meboldt, Matthiesen & Lohmeyer 2012**

Meboldt, M., Matthiesen, S. und Lohmeyer, Q. (2012) The Dilemma of Managing Iterations in Time-to-Market Development Processes. In 2nd International Conference on Modelling and Management of Engineering Processes MMEP 2012, Cambridge, UK.

**Meboldt 2008**

Meboldt, M. (2008) Modellbildung in Der Produktentstehung - Als Beitrag Zum Integrierten Produktentstehungs-Modell (iPeM). Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Mesarovic & Macko 1969**

Mesarovic, M. D., Macko, D., (1969) Foundations for a Scientific Theory of Technical Systems. In Hierarchical Structures, Whyte, L. L.; Wilson, A. G.; Wilson, D., American Elsevier Publishing Company, New York.

**Mika 2007**

Mika, P. (2007) Ontologies Are Us: A Unified Model of Social Networks and Semantics. In Web Semantics Journal, Volume 5.

**Miller, Galanter & Pribram 1960**

Miller, G. A., Galanter, E. und Pribram, K. A. (1960) Plans and the Structure of Behavior. Holt, Rhinehart, & Winston, New York.

**Muschik 2011**

Muschik, S. (2011) Development of Systems of Objectives in Early Product Engineering. Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Ncube 2011**

Ncube, C. (2011) On the Engineering of Systems of Systems: Key Challenges for the Requirements Engineering Community. In Workshop on Requirements Engineering for Systems, Services and Systems-of-Systems.

**Negele 1998**

Negele, H. (1998) Systemtechnische Methodik zur ganzheitlichen Modellierung am Beispiel der integrierten Produktentwicklung. Dissertation, TU München.

**Negele et al. 1999**

Negele, H., Fricke, E., Schrepfer, L. und Härtlein, N. (1999) Modelling of Integrated Product Development Processes. In 9th Annual Symposium of INCOSE, UK.

**Nielsen et al. 2015**

Nielsen, C., Ballegaard, P., Larsen, G., Fitzgerald, J., Woodcock, J. und Peleska, J. (2015) Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. In ACM Computing Survey Volume 48, Issue 2.

**Oerding 2009**

Oerding, J. (2009) Ein Beitrag zum Modellverständnis der Produktentstehung - Strukturierung von Zielsystemen mittels C&CM. Dissertation, IPEK, Universität Karlsruhe (TH).

**U.S. Verteidigungsministerium 2008**

Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering (2008) Systems Engineering Guide for Systems of Systems. ODUSD(A&T)SSE, Version 1, Washington.

**Ogden & Richards 1923**

Ogden, C.K. und Richards, I.A. (1923) The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism. Harcourt, Brace & World, Inc. New York.

**Pahl et al. 2007**

Pahl, G., Beitz, W., Feldhusen, J. und Grote, K.-H. (2007) Pahl/Beitz Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung - Methoden und Anwendung. 7. Auflage, Springer, Berlin, Heidelberg.

**Pahl et al. 2013**

Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., Heusel, J., Bronnhuber, T. und Hufenbach, W. (2013) Konstruktionslehre. Berlin, Heidelberg: Springer Berlin Heidelberg.

**Patil, Dutta & Sriram 2005**

Patil, L., Dutta, D. und Sriram, R. (2005) Ontology-Based Exchange of Product Data Semantics. In IEEE Transactions on Automation Science and Engineering, Volume 2, Issue 3.

**Patzak 1982**

Patzak, G. (1982) Systemtechnik, Planung komplexer innovativer Systeme. Grundlagen, Methoden, Techniken. Springer, Berlin.

**Partsch 2010**

Partsch, H. (2010). Requirements-Engineering systematisch – Modellbildung für softwaregestützte Systeme. Springer-Verlag, Heidelberg.

**Perry & Holt 2013**

Perry, S. und Holt, J. (2013) Guidelines for Systems of Systems Requirements. In COMPASS White Paper WP01.

**PMI 2004**

Project Management Institute (2004) A Guide to the Project Management Body of Knowledge. Pmbok#174, 5. Edition.

**Pohl 2008**

Pohl, K. (2008) Requirements Engineering: Grundlagen, Prinzipien, Techniken. dpunkt.verlag, Heidelberg.

**Pohl & Rupp 2011**

Pohl, K. und Rupp, C. (2011) Basiswissen Requirements Engineering: Aus- Und Weiterbildung Nach IREB-Standard Zum Certified Professional for Requirements Engineering Foundation Level. 3., dpunkt.verlag, Heidelberg.

**Ponn & Lindemann 2011**

Ponn, J. und Lindemann, U. (2011) Konzeptentwicklung und Gestaltung technischer Produkte - Systematisch von Anforderungen zu Konzepten und Gestaltungsformen. 2. Auflage, Springer, Berlin, Heidelberg.

**Rochfort 2015**

Rochfort, J. (2015) Version Control. Guidance Document. University of Nottingham.

**Ropohl 1975**

Ropohl, G. (1975) Einleitung in die Systemtechnik. In Systemtechnik – Grundlagen und Anwendung, Hanser, München, S. 1-77.

**Ropohl 2009**

Ropohl, G. (2009) Allgemeine Technologie – Eine Systemtheorie der Technik. Universitätsverlag Karlsruhe.

**Roth 2015**

Roth, S. (2015) Was Ist Eigentlich Ein System of Systems? Internet: <http://roth-soft.de/systems-engineering/was-ist-ein-system-of-systems/>, letzter Zugriff am 13.12.2016.

**Sanders 1966**

Sanders, C. (1966) Collected Papers of Charles Sanders Peirce. Edited by Charles Hartshorne and Paul Weiss. Harvard University Press, Cambridge.

**Sauter 2011**

Sauter, C. (2012) Ein Beitrag zur Integration von Wikis und Social Tagging in die Produktentstehung. Dissertation, IPEK, Karlsruher Institut für Technologie (KIT).

**Scherer 2016**

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, IPEK, Karlsruher Institut für Technologie (KIT).

**Schienmann 2002**

Schienmann, B. (2002) Kontinuierliches Anforderungsmanagement: Prozesse - Techniken - Werkzeuge. Pearson Deutschland GmbH.

**Schmitz et al. 2006**

Schmitz, C., Hotho, A., Jäschke, R. und Stumme, G. (2006) Kollaboratives Wissensmanagement. In Semantic Web. Springer Berlin Heidelberg.

**Schneider 2012**

Schneider, T. (2012) Methode Zur Erstellung von Problemlösungs- Modellen Auf Basis Des SPALTEN-Prozesses in Komplexen Entwicklungsprozessen. Dissertation, Karlsruher Institut für Technologie (KIT).

**Schütz 2004**

Schütz, T. (2004) Dokumentenmanagement. In Grundlagen Der Praktischen Information Und Dokumentation: Band 1: Handbuch Zur Einführung in Die Informationswissenschaft Und - Praxis - Band 2.

**Schwaber 2004**

Schwaber, K. (2004) Agile Project Management with Scrum. Microsoft Press, Redmond.

**Scrum Alliance 2013**

Scrum Alliance (2013) The State of Scrum: Benchmarks and Guidelines. Report. [www.scrumalliance.org](http://www.scrumalliance.org).

**Staab 2002**

Staab, S. (2002) Wissensmanagement Mit Ontologien Und Metadaten. In Informatik-Spektrum, Volume 25, Issue 3.

**Stechert 2010**

Stechert, C. (2010) Modellierung komplexer Anforderungen. Dissertation, TU Braunschweig.

**Stiefel, Hausknecht & Müller 2010**

Stiefel, P., Hausknecht, C. und Müller, J. (2010) Using Ontologies to Support Decentral Product Development Processes. In Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems.

**Stock & Stock 2008**

Stock, W. und Stock, M. (2008) Wissensrepräsentation: Informationen Auswerten Und Bereitstellen. Einführung in Die Informationswissenschaft. Oldenbourg.

**Studer, Benjamins & Fensel 1998**

Studer, R., Benjamins, V. und Fensel, D. (1998) Knowledge Engineering: Principles and Methods. In Data & Knowledge Engineering Volume 25. Elsevier Science Publishers.

**Ullrich, Maier & Angele 2003**

Ullrich, M., Maier, A. und Angele, J. (2003) Taxonomie, Thesaurus, Topic Map, Ontologie - Ein Vergleich. Ontoprise, Karlsruhe.

**Unger 2003**

Unger, D. W. (2003) Product Development Process Design: Improving Development Response to Market, Technical and Regulatory Risks. Dissertation, MIT.

**VDI 2001**

Verein Deutscher Ingenieure (VDI) (2001) Richtlinie VDI 2519 Blatt 1 - Vorgehensweise Bei Der Erstellung von Lasten/Pflichtenheften. Beuth Verlag.

**Walter et al. 2016**

Walter, B., Albers, A., Haupt, M. und Bursac, N. (2016) Produktentwicklung Im Virtuellen Ideenlabor – Konzipierung Und Implementierung Eines Live-Lab. In 27. DfX-Symposium 2016.

**Walter, Albers, Schelleis & Kurrle 2016**

Walter, B., Albers, A., Schelleis, A. und Kurrle, A. (2017) Efficient use of Sounding Board Method at project milestones and its potentials for virtualization. In 27th CIRP Design 2017, Cranfield.

**Wieggers 2006**

Wieggers, K. (2006) Software requirement. Microsoft Press, Redmond.

**Wintgen 1968**

Wintgen, G. (1968) Zur Mengentheoretischen Definition Und Klassifizierung Kybernetischer Systeme. In Wissenschaftliche Zeitschrift Der Humboldt-Universität Zu Berlin, Gesellschafts-Und Sprachwissenschaftliche Reihe 17.

**Zingel 2013**

Zingel, 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, IPEK, Karlsruher Institut für Technologie.

---

## Betreute Bachelor- und Masterarbeiten

### Starz 2014

Starz, J. „Dynamische Bewertung von Zielsystemen in der Produktentwicklung von komplexen technischen Systemen“. In: IPEK - Abschlussarbeiten: Bachelorarbeit. Bd. 2645. Betreuer: Albers, A; Co-Betreuer: Kurrle, A.; Bursac, N. Karlsruher Institut für Technologie (KIT): IPEK - Institut für Produktentwicklung, 2014.

### Johmann 2014

Johmann, Y. „Validierung von Systems-of-Systems am Beispiel eines vernetzten Fahrzeugs“. In: IPEK - Abschlussarbeiten: Bachelorarbeit. Bd. 2673. Betreuer: Albers, A; Co-Betreuer: Kurrle, A. Karlsruher Institut für Technologie (KIT): IPEK - Institut für Produktentwicklung, 2014.

### Lin 2015

Lin, T.-H. „Development of a systemic model of system-of-systems engineering based on the advanced system triple approach for the Connected Car“. In: IPEK - Abschlussarbeiten: Masterarbeit. Bd. 2588. Betreuer: Albers, A; Co-Betreuer: Kurrle, A. Karlsruher Institut für Technologie (KIT): IPEK - Institut für Produktentwicklung, 2015.

### Schaupp 2015

Schaupp, L. „Analyse von Veränderungstreibern in der Produktentwicklung der Automobilindustrie“. In: IPEK - Abschlussarbeiten: Bachelorarbeit. Bd. 2868. Betreuer: Albers, A; Co-Betreuer: Kurrle, A. Karlsruher Institut für Technologie (KIT): IPEK - Institut für Produktentwicklung, 2015.

### Knapp 2015

Knapp, T. „Konzeption eines zentralen Modells für das Anforderungsmanagement von Systems-of-Systems zur Unterstützung der E2E Absicherung am Beispiel des Connected Car“. Fakultät für Informatik und Wirtschaftsinformatik Fachgebiet Wirtschaftsinformatik. Betreuer: Prof. Dr. Rainer Neumann; Zweitbetreuer: Prof. Dr. Andreas Heberle; Hochschule Karlsruhe; Co-Betreuer: Kurrle, A., 2015

### Rutkowski 2015

Rutkowski, L. „Benchmark von Anforderungsmanagement-Tools in der frühen Phase des Produktentstehungsprozesses“. In: IPEK - Abschlussarbeiten: Bachelorarbeit. Bd. 2907. Betreuer: Albers, A; Co-Betreuer: Heitger, N.; Kurrle, A. Karlsruher Institut für Technologie (KIT): IPEK - Institut für Produktentwicklung, 2015.

### Barwanietz 2016

Barwanietz, D. „Ein Konzept zur durchgängigen Abbildung von mobilen Online Diensten für das Connected Car“. In: IPEK - Abschlussarbeiten: Bachelorarbeit. Vorgelegt an der Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt in der Fakultät Informatik und Wirtschaftsinformatik. Erstprüfer: John, I.; Zweitprüfer: Huffstadt, C. Betreuer: Kurrle, A, 2016.

### Schreiber 2016

Schreiber, M. „Ein semantischer Ansatz für das Management von Zielsystemen auf Basis des Resource Description Frameworks“. In: IPEK - Abschlussarbeiten: Masterarbeit. Bd. 2967. Betreuer: Albers, A; Co-Betreuer: Kurrle, A. Karlsruher Institut für Technologie (KIT): IPEK - Institut für Produktentwicklung, 2016.

## 10 Anhang

### 10.1 Abbildungsverzeichnis

Abbildung 1: Aufbau der Arbeit.....	4
Abbildung 2: Konzepte der Systemtheorie nach ROPOHL .....	6
Abbildung 3: Zielsystem, Handlungssystem und Sachsystem nach ROPOHL.....	9
Abbildung 4: ZOPH-Modell nach NEGELE .....	10
Abbildung 5: ZHO-Modell nach ALBERS .....	11
Abbildung 6: Das erweiterte ZHO-Modell .....	11
Abbildung 7: In-Stage und Cross-Stage Iterationen in der Produktentstehung .....	12
Abbildung 8: Auflösungsgrade des Produktentstehungsprozesses .....	17
Abbildung 9: TOTE-Modell und PDCA-Zyklus .....	18
Abbildung 10: SPALTEN-Problemlösungsmethode.....	21
Abbildung 11: Der Stage-Gate-Prozess .....	22
Abbildung 12: V-Modell nach VDI 2221 .....	24
Abbildung 13: Das SCRUM-Vorgehensmodell .....	25
Abbildung 14: Das iPeM im Kontext der PGE.....	27
Abbildung 15: Änderungsprozess nach DIN 199-4.....	30
Abbildung 16: Bausteine und Aktionsfelder des Integrierten Änderungsmanag .....	32
Abbildung 17: Zielsystem eines Entwicklungsprojekts nach BADER.....	35
Abbildung 18: Zusammenhang zwischen Zielen, Anforderungen, Randbedingungen.....	38
Abbildung 19: Entwicklung des Zielsystems; eigene Darstellung nach <i>Meboldt</i> .....	39
Abbildung 20: Quellen von Zielen und Anforderungen .....	40
Abbildung 21: Individuelle Wissensbasis und Lösungsraum bei der Entwicklung ....	41
Abbildung 22: Interaktionsebenen des erweiterten ZHO-Modells.....	42
Abbildung 23: Aktivitäten des Requirements Engineering nach POHL .....	43
Abbildung 24: Requirements Engineering Prozess nach Hood et al. ....	45
Abbildung 25: Abgrenzung von Lastenheft, Pflichtenheft und Anforderungsliste.....	47
Abbildung 26: Diagrammtypen von SysML.....	48

---

Abbildung 27: Werkzeuge des Requirements Engineering .....	53
Abbildung 28: Subjektive Effizienz von Methoden .....	54
Abbildung 29: Durchgängigkeit und Konsistenz von Modellen .....	55
Abbildung 30: Grundsätzliche Beziehungsarten im Zielsystem .....	56
Abbildung 31: Das semiotische Dreieck – Rahmenbedingung der Ontologie.....	62
Abbildung 32: Semantische Mächtigkeit der semantischen Modelle .....	63
Abbildung 33: Ontologien als gemeinsames Verständnismodell .....	64
Abbildung 34: Darstellung eines RDF-Tripels als gerichteter Graph .....	67
Abbildung 35: Beispiel Darstellung von RDF mit XML und zugehöriger RDF Graph	69
Abbildung 36: Typen von Systems-of-Systems .....	78
Abbildung 37: Aktuelle Modellpalette der Dr. Ing. h.c. F. Porsche AG.....	88
Abbildung 38: Dienstportfolio Porsche Connect .....	89
Abbildung 39: Das Connected Car – ein System-of-Systems.....	90
Abbildung 40: Forschungsdesign der Arbeit basierend auf der DRM .....	91
Abbildung 41: Datenerhebungsmethoden und ihre Verwendung .....	92
Abbildung 42: Forschungsfragen der ersten deskriptiven Studie.....	94
Abbildung 43: Informationen über Teilnehmer der Studie .....	96
Abbildung 44: Notwendigkeit für Änderungen.....	100
Abbildung 45: Bewertung der SoS Eigenschaften .....	100
Abbildung 46: Partielle explizierte Zielsysteme in der Entwicklung.....	105
Abbildung 47: Verlust „horizontaler Durchgängigkeit“ .....	107
Abbildung 48: Verlust „vertikaler Durchgängigkeit“ .....	108
Abbildung 49: Randbedingungen für eine Toolauswahl .....	110
Abbildung 50: Dimensionen der Durchgängigkeit .....	113
Abbildung 51: Anzeigen und Personalisierung von Nachrichten .....	115
Abbildung 52: Verwendete Tools zur Dokumentation von Anforderungen .....	116
Abbildung 53: Beispiel für einen Status-Workflow bei Entwicklungsdokumenten ...	117
Abbildung 54: Ausschnitt aus einem Lastenheft.....	119
Abbildung 55: Abgrenzung des Managements von Änderungen.....	123
Abbildung 56: Dynamische Zusammensetzung eines Entscheidungsgremiums ....	125



---

Abbildung 57: Forschungsfragen der Präskriptiven Studie .....	127
Abbildung 58: Herstellung von Durchgängigkeit .....	129
Abbildung 59: Beispiel für RDFS Klasse Artefakt .....	131
Abbildung 60: Darstellung der Relation „istTeilVon“ im RDF Graph .....	133
Abbildung 61: Darstellung der Relation „istVerwandtMit“ im RDF Graph .....	133
Abbildung 62: Partialmodelle der Produktentwicklung .....	134
Abbildung 63: Beispiel für RDFS Klasse Person .....	134
Abbildung 64: Abbildung der Relation „interessiert“ im RDF Graph.....	135
Abbildung 65: Darstellung der „beschreibtInhalteVon“ Relation im RDF Graph .....	137
Abbildung 66: Darstellung der „istTeilVon“ Beziehung im RDF Graph .....	138
Abbildung 67: Darstellung der „verwendet“ Relation im RDF Graph .....	138
Abbildung 68: Darstellung der Relationen im RDF Graph .....	139
Abbildung 69: Ableitung impliziter Relationen aus hierarchischen Relationen .....	141
Abbildung 70: Ableitung impliziter Relationen aus generischen Relationen .....	142
Abbildung 71: Ableitung impliziter Relationen aus einer Funktionsstruktur .....	143
Abbildung 72: Ableitung impliziter Relationen aus Stakeholder-Beziehungen.....	144
Abbildung 73: Implizite Relationen über Artefakt-Beziehungen .....	144
Abbildung 74: Implizite Relationen bei Produktgenerationen .....	145
Abbildung 75: Logische Architektur der web-basierten Plattform .....	146
Abbildung 76: Benutzeroberfläche des Software-Prototyp .....	147
Abbildung 77: Hinzufügen von RDF Statements im entwickelten Prototyp.....	148
Abbildung 78: Verwendete Technologien .....	149
Abbildung 79: Eingliederung des Management von Änderungen in das iPeM .....	151
Abbildung 80: Verwendung der Suchfunktion im RDF Prototyp .....	155
Abbildung 81: Fraktaler Charakter der SPALTEN-Methodik.....	156
Abbildung 82: Forschungsfragen der Validierung.....	158
Abbildung 83: Pflege von Metainformationen im entwickelten Software-Prototyp ..	160
Abbildung 84: Ergebnis der Bewertung zur Anwendbarkeit.....	162
Abbildung 85: Bewertung der Erhöhung von Durchgängigkeit .....	163
Abbildung 86: Bewertung von Aufwand und Nutzen.....	164

---

Abbildung 87: Phasen und Aktivitäten im Rahmen von ProVIL .....	166
Abbildung 88: Die SAP Innovationsplattform von ProVIL .....	167
Abbildung 89: In welchen Projektphasen Verantwortungsbereiche vergeben? .....	168
Abbildung 90: Bewertung der Häufigkeit von Änderungen in den Projektphasen...	168
Abbildung 91: Bewertung der Verwendung von Aktivitäten .....	170
Abbildung 92: Bewertung der Eignung der Methodik .....	171
Abbildung 93: Erfüllung der definierten Ziele durch den entwickelten Ansatz .....	177
Abbildung 94: Berufserfahrung der Teilnehmer .....	199
Abbildung 95: Häufigkeit von Änderungen in Projektphasen .....	199
Abbildung 96: Häufigkeit der Vergabe von Verantwortungsbereichen.....	200
Abbildung 97: Ergebnisse der Untersuchungen bei der Situationsanalyse .....	201
Abbildung 98: Ergebnisse der Untersuchungen bei der Problemeingrenzung.....	201
Abbildung 99: Ergebnisse der Untersuchungen bei der Lösungsauswahl .....	201
Abbildung 100: Ergebnisse der Untersuchungen bei der Tragweitenanalyse .....	202
Abbildung 101: Ergebnisse der Untersuchungen Entscheiden und Umsetzen.....	202
Abbildung 102: Ergebnisse der Untersuchungen bei der Dokumentation .....	202

## 10.2 RDF Schema (siehe auch Kapitel 6.1)

Subjekt	Prädikat	Objekt
Entwicklungsartefakt <i>rdf:type = „artefact“</i>	<ul style="list-style-type: none"> <li>• IPEK: istTeilVon</li> <li>• IPEK: istVerwandtMit</li> <li>• IPEK: basiertAuf</li> <li>• IPEK: status</li> <li>• IPEK: entwicklungsphase</li> <li>• IPEK: fertigstellungBis</li> <li>• IPEK: beschreibtInhalteVon</li> </ul>	<ul style="list-style-type: none"> <li>• Entwicklungsartefakt</li> <li>• Entwicklungsartefakt</li> <li>• Entwicklungsartefakt</li> <li>• Status</li> <li>• Phase / Aktivität</li> <li>• Meilenstein</li> <li>• System, Funktion, Ziele, Anforderungen, Anwendungsfälle, Tests, Phasen und Aktivitäten, Meilensteine</li> </ul>
System <i>rdf:type = „system“</i>	<ul style="list-style-type: none"> <li>• IPEK: istTeilVon</li> <li>• IPEK: verwendet</li> <li>• IPEK: basiertAuf</li> <li>• IPEK: entwicklungsphase</li> <li>• IPEK: fertigstellungBis</li> <li>• IPEK: erfüllt</li> <li>• IPEK: entwicklungsphase</li> <li>• IPEK: basiertAuf</li> </ul>	<ul style="list-style-type: none"> <li>• System</li> <li>• System</li> <li>• System</li> <li>• Phase / Aktivität</li> <li>• Meilenstein</li> <li>• Ziele / Anforderungen / Anwendungsfälle</li> <li>• Produktgeneration / Entwicklungsphase / Aktivität</li> <li>• System (Referenz)</li> </ul>
Funktion <i>rdf:type = „function“</i>	<ul style="list-style-type: none"> <li>• IPEK: istTeilVon</li> <li>• IPEK: verwendet</li> <li>• IPEK: basiertAuf</li> <li>• IPEK: Erfüllt</li> </ul>	<ul style="list-style-type: none"> <li>• Funktion</li> <li>• Funktion</li> <li>• System</li> <li>• Ziele / Anforderungen / Anwendungsfälle</li> </ul>
Stakeholder <i>rdf:type = „stakeholder“</i>	<ul style="list-style-type: none"> <li>• IPEK: interessiert</li> <li>• IPEK: verantwortlich</li> <li>• IPEK: Mitwirkendend</li> </ul>	<ul style="list-style-type: none"> <li>• Alles</li> <li>• Entwicklungsartefakt, System, Funktion, Stakeholder, Ziele / Anforderungen / Anwendungsfälle</li> <li>• Entwicklungsartefakt, System, Funktion, Stakeholder, Ziele / Anforderungen / Anwendungsfälle</li> </ul>

Subjekt	Prädikat	Objekt
Ziele / Anforderungen / Anwendungsfälle <i>rdf:type = „goal“</i>	<ul style="list-style-type: none"> <li>• IPEK: verfeinert</li> <li>• IPEK: beinhaltet</li> <li>• IPEK: basiertAuf</li> </ul>	<ul style="list-style-type: none"> <li>• Ziele / Anf / Anwendungsfälle</li> <li>• Ziele / Anf / Anwendungsfälle</li> <li>• Ziele / Anf / Anwendungsfälle</li> </ul>
Entwicklungsphasen und Aktivitäten <i>rdf:type = „project“</i>	<ul style="list-style-type: none"> <li>• IPEK: folgtAuf</li> </ul>	<ul style="list-style-type: none"> <li>• Phase/Aktivität</li> </ul>
Meilensteine <i>rdf:type = „milestone“</i>	<ul style="list-style-type: none"> <li>• IPEK: folgtAuf</li> </ul>	<ul style="list-style-type: none"> <li>• Phase/Aktivität</li> </ul>
Tests <i>rdf:type = „test“</i>	<ul style="list-style-type: none"> <li>• IPEK: verifiziert</li> <li>• IPEK: validiert</li> </ul>	<ul style="list-style-type: none"> <li>• System</li> <li>• Funktion</li> <li>• Ziele / Anforderungen / Anwendungsfälle</li> </ul>
Produktgeneration <i>rdf:type = „productgeneration“</i>	<ul style="list-style-type: none"> <li>• IPEK: entwicklungsphase</li> <li>• IPEK: folgtAuf</li> </ul>	<ul style="list-style-type: none"> <li>• Alle Partialmodellelemente</li> <li>• Produktgeneration</li> </ul>

### 10.3 Auszug Inferenzregeln (siehe auch Kapitel 6.2)

Bedingung	Inferenz
(?artefact po:beschreibinhaltVon ?system) (?system po:verantwortlich ?stakeholder)	(?artefact po:interessiert ?stakeholder)
(?artefact1 po:istVerwandtMit ?artefact2) (?artefact1 po:verantwortlich ?stakeholder)	(?artefact2 po:interessiert ?stakeholder)
(?artefact1 po:beinhaltetVerfeinerungVon ?artefact2) (?artefact1 po:verantwortlich ?stakeholder)	(?artefact2 po:interessiert ?stakeholder)
(?artefact1 po:istAbleitungAus ?artefact2) (?artefact1 po:verantwortlich ?stakeholder)	(?artefact2 po:interessiert ?stakeholder)
(?artefact1 po:beinhaltetDekompositionVon ?artefact2) (?artefact1 po:verantwortlich ?stakeholder)	(?artefact2 po:interessiert ?stakeholder)
(?system1 po:subSystemVon ?system2) (?artefact po:beschreibinhaltVon ?system1)	(?artefact po:beschreibinhaltVon ?system2)
(?system1 po:verwendetSchnittstelleVon ?system2) (?artefact1 po:beschreibinhaltVon ?system1) (?artefact2 po:beschreibinhaltVon ?system2)	(?artefact1 po:istVerwandtMit ?artefact2)
(?system1 po:subSystemVon ?system2) (?artefact1 po:beschreibinhaltVon ?system1) (?artefact2 po:beschreibinhaltVon ?system2)	(?artefact1 po:istVerwandtMit ?artefact2)
(?artefact1 po:beschreibinhaltAus ?partialmodel1) (?artefact1 po:beschreibinhaltVon ?system2)	(?artefact po:beschreibinhaltVon ?system2)

## 10.4 Ergebnisse Befragung (siehe auch Kapitel 7.2)

**Berufserfahrung der Teilnehmer (Insgesamt 30 Teilnehmer):**

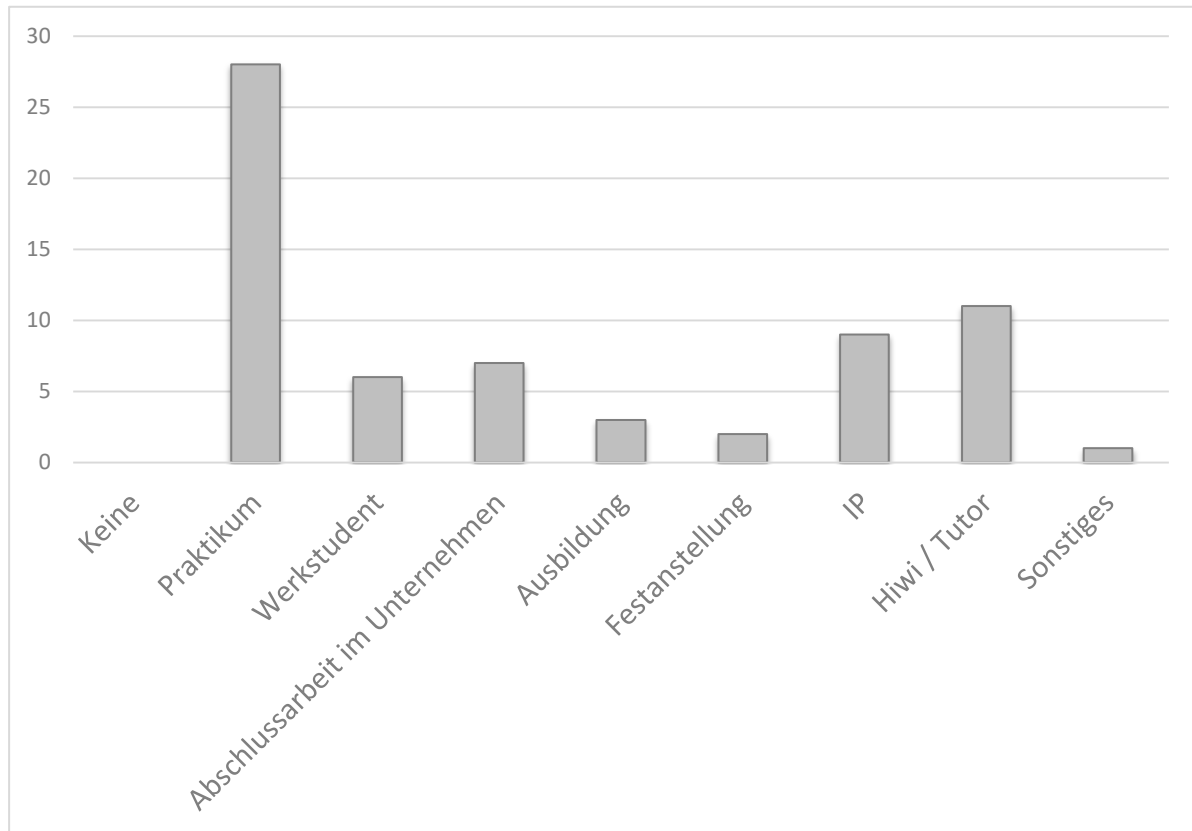


Abbildung 94: Berufserfahrung der Teilnehmer

**Frage 1: In welchen Projektphasen gab es Änderungen (hell)?**

**Frage 2: In welchen Projektphasen gab es Änderungen, welche einen in einer früheren Phase definierten Soll-Zustand betreffen (dunkel)?**

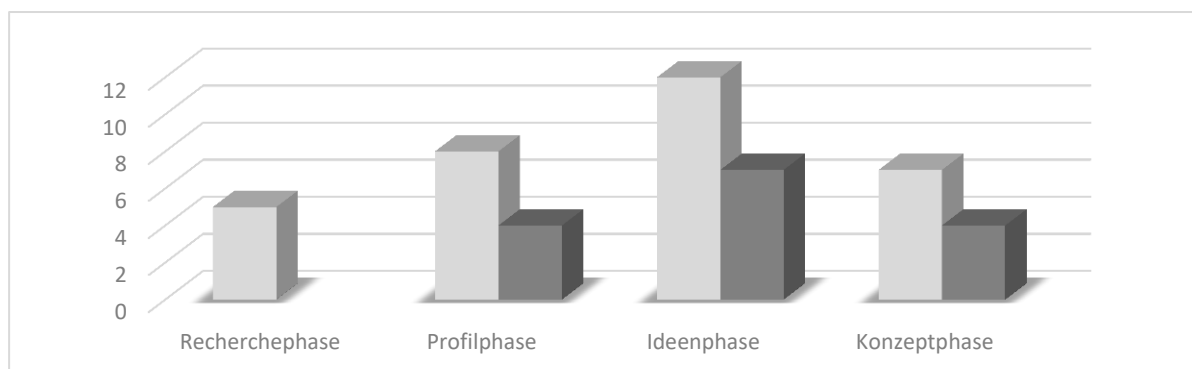


Abbildung 95: Häufigkeit von Änderungen in Projektphasen

**Frage 3: Bitte sortieren sie aufsteigend die Projektphase nach Anzahl der Änderungen.**

	Antwort Tandem																Durchschnitt
Recherchephase	3	4	2	4	1	4	4		3	3	4	3	4	4	4	4	3,4
Profilphase	1	3	1	2	4	1	3		4	4	3	1	3	3	1	2	2,4
Ideenphase	2	2	3	1	2	2	1		2	1	1	2	2	1	2	1	1,66666667
Konzeptphase	4	1	4	3	3	3	2		1	2	2	4	1	2	4	3	2,6

**Frage 4: Bitte sortieren sie die Stakeholder nach deren Einfluss auf Änderungen.**

	Antwort Tandem																Durchschnitt
Eigenes Team	2	1	1	2	4	3	3		1	3	3	1	2	3	2	3	2,26666667
Andere Teams	5	4	5	5	5	4	5		5	4	5	5	4	5	3	5	4,6
IPEK Betreuer	1	2	2	1	1	1	1		3	1	2	2	1	2	1	1	1,46666667
Inno Coaches	3	5	4	4	3	5	4		4	2	4	4	5	4	5	4	4
Projektpartner	4	3	3	3	2	2	2		2	5	1	3	3	1	1	2	2,46666667

**Frage 5: In welchen Projektphasen wurden Verantwortungsbereiche vergeben?**

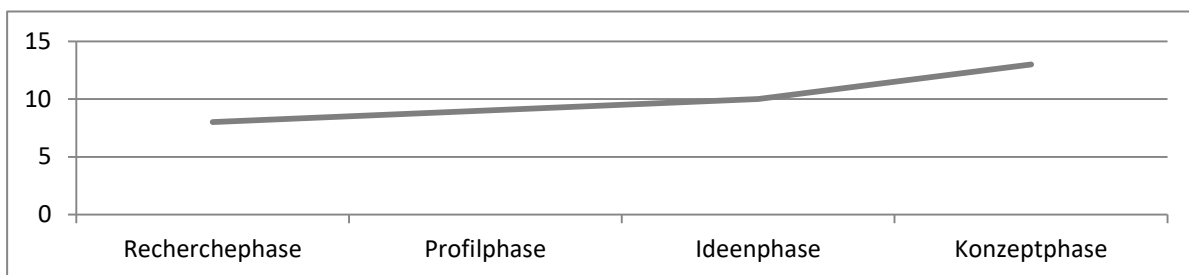


Abbildung 96: Häufigkeit der Vergabe von Verantwortungsbereichen nach Projektphasen

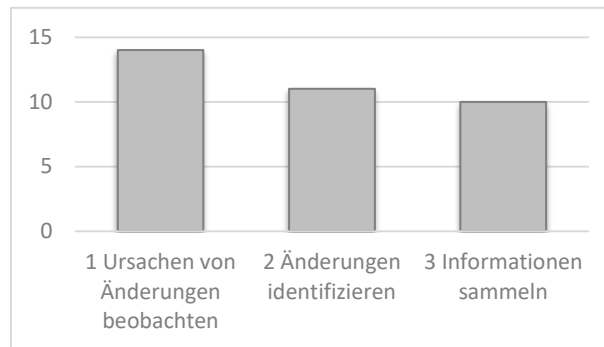
**Frage 6: Durchgeführte Aktivitäten des Management von Änderungen (je Arbeitsschritt).**

Abbildung 97: Ergebnisse der Untersuchungen bei der Situationsanalyse

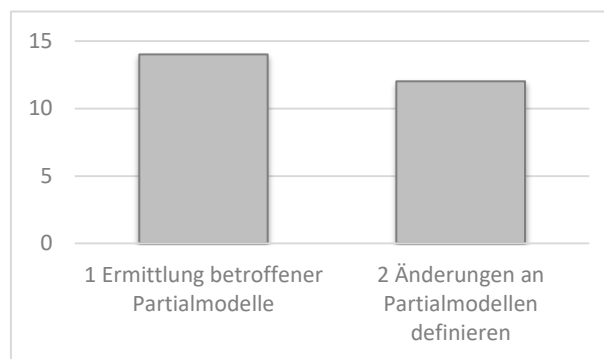


Abbildung 98: Ergebnisse der Untersuchungen bei der Problemeingrenzung

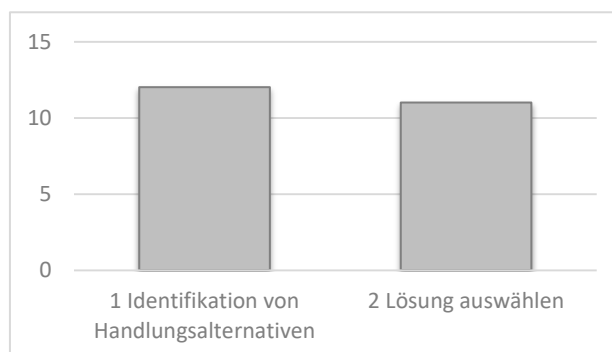


Abbildung 99: Ergebnisse der Untersuchungen bei der Lösungsauswahl



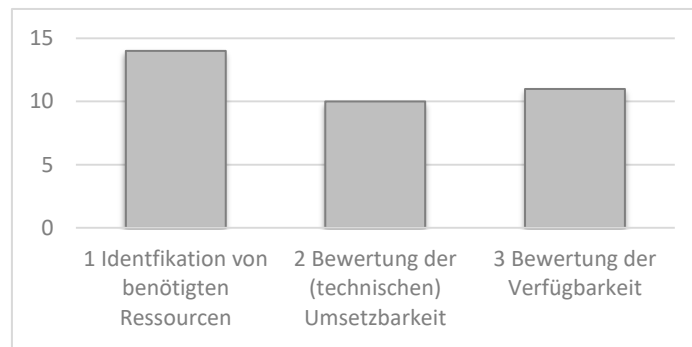


Abbildung 100: Ergebnisse der Untersuchungen bei der Tragweitenanalyse

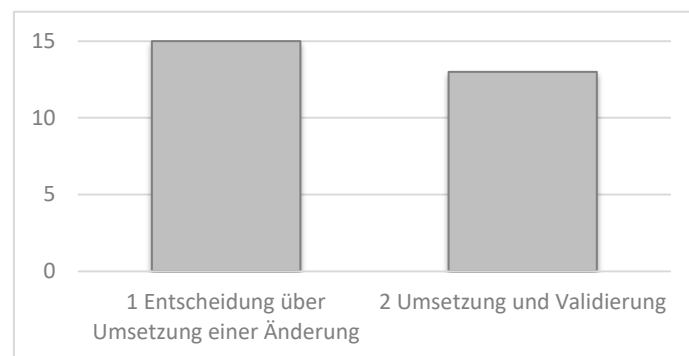


Abbildung 101: Ergebnisse der Untersuchungen beim Entscheiden und Umsetzen

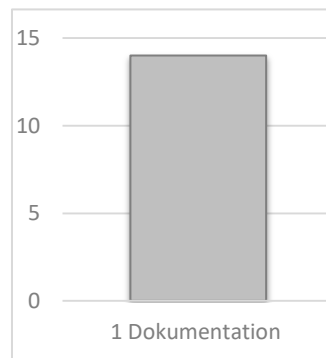
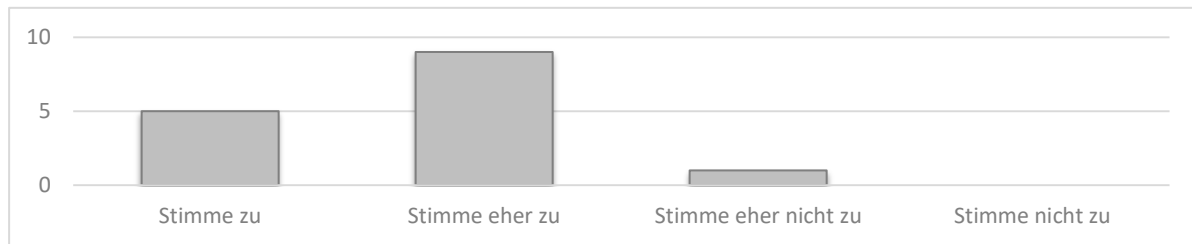


Abbildung 102: Ergebnisse der Untersuchungen bei der Dokumentation

**Frage 7: SPALTEN-Methodik / Methodik ist geeignet für das Management von Änderungen?**

# Lebenslauf

## Persönliche Daten

Name: Armin Kurrle  
Geburtsdatum: 13. April 1987  
Geburtsort: Balingen  
Staatsangehörigkeit: Deutsch  
Familienstand: ledig

## Bildungsgang

2006 Abitur am Gymnasium Neuenbürg  
2006 – 2012 Studium der Informatik (Dipl.-Inform.),  
am Karlsruher Institut für Technologie (KIT)

## Berufstätigkeit

2008 – 2012 Wissenschaftliche Hilfskraft im Sonderforschungsbereich 588  
„Humanoide Roboter“, Karlsruher Institut für Technologie (KIT)

2010 – 2012 Praktikant und Werkstudent im Application Management,  
MHP – A Porsche Company

2012 Diplomarbeit im Bereich Forschung & Entwicklung / Telematik und  
Infotainmentsysteme, Daimler AG

2013 – 2016 Doktorand im Bereich Porsche Connect, Dr. Ing. h.c. F. Porsche AG in  
Kooperation mit dem IPEK – Institut für Produktentwicklung am Karlsruher  
Institut für Technologie (KIT)

seit April 2016 Projektleiter im Bereich Porsche Connect & Big Data,  
Dr. Ing. h.c. F. Porsche AG