

Christian Tschirner

***Rahmenwerk zur Integration
des modellbasierten Systems
Engineering in die Produkt-
entstehung mechatronischer
Systeme***

***Framework for Integrating
Model-Based Systems
Engineering into Product
Engineering Processes***

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar

© Heinz Nixdorf Institut, Universität Paderborn – Paderborn – 2016

Das Werk einschließlich seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung der Herausgeber und des Verfassers unzulässig und strafbar. Das gilt insbesondere für Vervielfältigung, Übersetzungen, Mikroverfilmungen, sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

Satz und Gestaltung: Christian Tschirner

Hersteller: Verlagshaus Monsenstein und Vannerdat OHG
Druck · Buch · Verlag
Münster

Printed in Germany

Geleitwort

Systems Engineering für den Entwurf Intelligenter Technischer Systeme ist die verbindende Leitidee des Heinz Nixdorf Instituts und des damit verbundenen Fraunhofer-Instituts Entwurfstechnik Mechatronik IEM.

Im Zeitalter der Digitalisierung führt der zunehmende Anteil an Informations- und Kommunikationstechnik zu einer Komplexitätssteigerung von mechatronischen Systemen und deren Entwicklungsprozessen. Interdisziplinäres Denken und Handeln sowie ein ganzheitliches Systemverständnis der beteiligten Akteure werden wichtiger denn je. Die technologischen und organisatorischen Herausforderungen bei der Entwicklung mechatronischer Systeme erfordern ein Model-Based Systems Engineering (MBSE). Häufig wird übersehen, dass MBSE noch in den Kinderschuhen steckt und etliche Forschungsarbeit bis zur erfolgreichen Anwendung in der Praxis notwendig ist. Insbesondere ist unklar, welche Anwender wie, wann und wofür MBSE überhaupt sinnvoll einsetzen können.

Vor diesem Hintergrund hat Herr Tschirner ein Rahmenwerk zur Integration des MBSE in die Produktentstehung mechatronischer Systeme erarbeitet. Es unterstützt die strukturierte Planung und Nutzung eines Systemmodells für spezifische Aufgaben der Produktentstehung. Grundlage bildet eine Beschreibung des Paradigmas MBSE. Hieran schließt sich ein Vorgehensmodell an, das ein MBSE-Projekt in vier Hauptaufgabenbereiche unterteilt: Modellplanung, Modellierung, Methodenkopplung und Anwendung. Als weiteren Bestandteil des Rahmenwerks hat Herr Tschirner einen Baukasten mit Vorgehensweisen, Methoden und Hilfsmittel für den erfolgreichen Einsatz von MBSE erarbeitet. Abschließend wird das Rahmenwerk anhand des Projektbeispiels „FlyPort – Kaffeemaschine“ der Gesellschaft für Systems Engineering e.V. für drei typische Methoden der Produktentstehung validiert.

Mit seiner Dissertation bewegt sich Herr Tschirner auf einem hochaktuellen und sehr herausfordernden Gebiet. Er leistet einen grundlegenden Beitrag für die Erforschung und Etablierung des MBSE in der Wissenschaft und Praxis. Die Arbeit ist ein weiterer wichtiger Baustein für unsere Paderborner Schule des Entwurfs intelligenter technischer Systeme.

Paderborn, im Dezember 2016

Prof. Dr.-Ing. J. Gausemeier

Dr.-Ing. R. Dumitrescu

**Rahmenwerk zur Integration
des modellbasierten Systems Engineering
in die Produktentstehung mechatronischer Systeme**

zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
der Fakultät Maschinenbau
der Universität Paderborn

genehmigte
DISSERTATION

von
Dipl.-Wirt.-Ing. M.Eng. Niels Christian Tschirner
aus Herdecke

Tag des Kolloquiums: 21. Oktober 2016
Referent: Prof. Dr.-Ing. Jürgen Gausemeier
Korreferent: Prof. Dr.-Ing. Reiner Anderl

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Fachgruppe Strategische Produktplanung und Systems Engineering am Heinz Nixdorf Institut (HNI) der Universität Paderborn sowie am Fraunhofer-Institut für Entwurfstechnik Mechatronik (Fraunhofer IEM). Sie ist das Ergebnis meiner wissenschaftlichen Arbeit im Rahmen von Forschungs- und Industrieprojekten.

Mein herzlicher Dank gilt Herrn Prof. Dr.-Ing. Jürgen Gausemeier, der mich stets forderte und förderte. Ich danke ihm besonders für das große Vertrauen in meine Arbeit, und dass er mein selbständiges und kreatives Forschen stets unterstützte.

Für die Übernahme des Korreferats danke ich Herrn Prof. Dr.-Ing. Reiner Anderl, dem Leiter des sehr renommierten Fachgebiets Datenverarbeitung in der Konstruktion der Technischen Universität Darmstadt.

Mein besonderer Dank gilt Dr.-Ing. Roman Dumitrescu, den ich als Kollegen am Lehrstuhl schätzen gelernt habe und der mich am Fraunhofer-Institut gefördert hat. Vielen Dank für die einzigartig konstruktive und vertrauensvolle Zusammenarbeit!

Mein größter Dank im beruflichen Umfeld gilt Dr.-Ing. Lydia Kaiser und Michael Hillebrand. Eure Begeisterung für das Systems Engineering, die Diskussionen und Ideen, die dauerhafte Motivation, der Zusammenhalt, die Offenheit, die Freundschaft: Danke!

Aber auch allen anderen Kolleginnen und Kollegen danke ich für die Zusammenarbeit. Hervorzuheben sind zunächst Sabine Illigen, Alexandra Dutschke, Meike Steffen und Daniela Peine. Danke für eure dauerhafte und außerordentliche Unterstützung! Lukas Bretz, Jörg Heihoff-Schwede, Christian Bremer, Jörg Holtmann, Matthias Greinert, Michael Bansmann: Danke für die tolle Zusammenarbeit, die intensiven Diskussionen und dass man sich stets auf euch verlassen kann! Stellvertretend für alle Studierenden, die mich unterstützt haben: Danke, Alina Linden und Tolga Ermis!

Letztlich ist es aber meine Familie, der das allergrößte Dankeschön gebührt! In erster Linie sind das meine Eltern Anne und Günter und meine Großeltern Inge und Friedhelm, die mir alle Vorbild sind und waren und denen ich unendlich dankbar bin für ihre Liebe, Zuneigung und Unterstützung. Meiner Frau Sonja danke ich für ihre Liebe, ihre Geduld, ihre Unterstützung – Du musstest häufig zurückstecken in den letzten Jahren! Danke, dass Du unseren Sohn Julius gerade auf den letzten Metern des Dissertationsprojekts so liebevoll umsorgt hast, wenn ich mal wieder zu lange am Schreibtisch saß. Dir, mein lieber Julius, ist diese Arbeit gewidmet!

Liste der veröffentlichten Teilergebnisse

- [BTD15] BANSMANN, M.; TSCHIRNER, C.; DUMITRESCU, R.: Implementing Model-Based Systems Engineering based on Shared Services concepts for industrial application. In: IEEE Systems Council (Hrsg.): IEEE International Symposium on Systems Engineering (ISSE), Rome, Italy, 2015
- [GDS+13] GAUSEMEIER, J.; DUMITRESCU, R.; STEFFEN, D.; TSCHIRNER, C.; CZAJA, A.; WIEDERKEHR, O.: Systems Engineering in der industriellen Praxis. In: GAUSEMEIER, J.; DUMITRESCU, R.; RAMMIG, F.-J.; SCHÄFER, W.; TRÄCHTLER, A. (Hrsg.): 9. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 310, Paderborn, 2013
- [GDT+11] GAUSEMEIER, J.; DUMITRESCU, R.; TSCHIRNER, C.; STILLE, K.: Modellbasierte Konzipierung eines hybriden Energiespeichersystems für ein autonomes Schienenfahrzeug. Tag des Systems Engineerings 2011 (TdSE), Hamburg, Carl Hanser Verlag, München, 2011
- [GGT13] GAUSEMEIER, J.; GAUKSTERN, T.; TSCHIRNER, C.: Systems Engineering Management Based on a Discipline-Spanning System Model. In: Conference on Systems Engineering Research (CSER 13), Elsevier B.V, Atlanta, GA, 2013
- [GRT12] GAUSEMEIER, J.; RUDTSCH, V.; TSCHIRNER, C.: A New Systems Engineering Approach as Enabler for Early Sustainability Analysis. In: Proceedings of the GCSM 2012 10th Global Conference on Sustainable Manufacturing, Istanbul, Turkey, 2012
- [GTD+12] GAUSEMEIER, J.; TSCHIRNER, C.; DUMITRESCU, R.; GAUKSTERN, T.: Integrative Konzipierung von Produkt und Produktionssystem als Basis für eine erfolgreiche Produktentstehung. In: Methoden und Werkzeuge in Produktentwicklung und Design, Dresden, 2012
- [GTD13] GAUSEMEIER, J.; TSCHIRNER, C.; DUMITRESCU, R.: Der Weg zu Intelligenten Technischen Systemen – Spitzencluster it's OWL – Mit Intelligenten Technischen Systemen an die Spitze. *Industriemanagement*, 29/2013, S. 49–52, GITO-Verlag, 2013
- [GTH16] GREINERT, M.; TSCHIRNER, C.; HOLTMANN, J.: Anwendung von Methoden der Produktentstehung auf Basis des Systemmodells mechatronischer Systeme. In: SCHULZE, S.-O.; TSCHIRNER, C. ET AL. (Hrsg.) – Tag des Systems Engineering 2016. Herzogenaurach, Carl Hanser Verlag, München, 2016
- [GTV12] GAUSEMEIER, J.; TSCHIRNER, C.; VABHOLZ, M.: Systems Thinking: Sensitizing for Systems Engineering – Experiences from academic teaching and industry workshops. In: Proceedings of the E&PDE 2012 – 14th International Conference on Engineering and Product Design Education, Artesis University College, Antwerp, Belgium, 2012
- [HBM15] HOLTMANN, J.; BERNIJAZOV, R.; MEYER, M.; SCHMELTER, D.; TSCHIRNER, C.: Integrated Systems Engineering and Software Requirements Engineering for Technical Systems. In: Proceedings of the International Conference on Software and Systems Process (ICSSP), S. 57-66, New York, NY, USA, 2015 – *best paper award*
- [TBD+15] TSCHIRNER, C.; BRETZ, L.; DUMITRESCU, R.; GAUSEMEIER, J.: Applying Model-Based Systems Engineering for Product Engineering Management – Concepts for Industrial Application. In: IEEE Systems Council (Hrsg.): 2015 IEEE International Symposium on Systems Engineering (ISSE), Rome, Italy, 2015
- [TDB+15] TSCHIRNER, C.; DUMITRESCU, R.; BANSMANN, M.; GAUSEMEIER, J.: Tailoring Model-Based Systems Engineering – Concepts for Industrial Application. In: IEEE Systems Council (Hrsg.): 2015 IEEE International Systems Conference (SysCon 2015) Proceedings. Vancouver, British Columbia, Canada, 2015
- [TKD+14] TSCHIRNER, C.; KAISER, L.; DUMITRESCU, R.; GAUSEMEIER, J.: Collaboration in Model-Based Systems Engineering based on Application Scenarios. In: Proceedings of NordDesign 2014 Conference, Aalto, 2014

Zusammenfassung

Die Herausforderungen bei der Entwicklung mechatronischer Systeme haben zu einem wachsenden Interesse am Model-Based Systems Engineering (MBSE) geführt. Dabei wird allerdings übersehen, dass das MBSE noch in den Kinderschuhen steckt und viel Forschungsarbeit zur Industrialisierung notwendig ist. Insbesondere ist unklar, welche Anwender MBSE wie, wann und für welche Aufgabenstellungen sinnvoll einsetzen können. Vor diesem Hintergrund wird ein Rahmenwerk zur Integration des MBSE in die Produktentstehung mechatronischer Systeme erarbeitet. Es unterstützt die Planung und Nutzung des Systemmodells für spezifische Aufgaben der Produktentstehung. Grundlage ist eine Beschreibung des Paradigmas MBSE. Hieran schließt sich ein Vorgehensmodell an, das ein MBSE-Projekt in vier Hauptaufgabenbereiche unterteilt: Modellplanung, Modellierung, Methodenkopplung und Anwendung. Die Modellplanung dient der Definition und Analyse des Modellierungszwecks. Zentral ist die Konzipierung des Systemmodells für den geplanten Zweck, d.h. die Definition der notwendigen Systemmodellinhalte und Modellierungsschritte. Die Bereiche Modellierung und Methodenkopplung leiten die Modellierung und Umsetzung des Zwecks an; im Bereich Anwendung geht es um die Verankerung des MBSE im Unternehmen. Die Modellplanung ist essentiell für den Projekterfolg, wird im MBSE jedoch weitgehend vernachlässigt. Aus diesem Grund adaptiert das Rahmenwerk Ansätze wie z.B. Six Sigma mit seiner Analysephase und überträgt sie auf das MBSE. Gleichzeitig ergibt sich ein Baukasten mit Vorgehensweisen, Methoden und Hilfsmittel für den Einsatz von MBSE. Das Rahmenwerk wird anhand des Projektbeispiels „FlyPort – Kaffeemaschine“ der Gesellschaft für Systems Engineering e.V. für verschiedene Methoden angewendet.

Summary

There has been a growing interest in the concept of Model-Based Systems Engineering (MBSE) in industry. Enterprises hope to improve the performance of their processes by help of this holistic approach. Nevertheless, MBSE is still in its infancy and much research to the industrialization of the approach is necessary – but often overlooked. Especially the numerous socio-technical issues of MBSE are neglected. It is not clear what stakeholders can use when, how and what for a well-sound MBSE. This includes also the corresponding model planning. For this reason, a framework for the integration of MBSE into the product engineering of mechatronic systems was developed. The basis is a description model, representing four consistent MBSE concepts confirmed in a case study. The concepts can be used for typical tasks and objectives in the product engineering process; appropriate decision support tools ensure the rational application. This is followed by an approach for the application of MBSE in the product life cycle. It is described in theory first and then operationalized through a flexible process model with a set of methods and tools. The exemplary application of the framework is done by implementing three methods of product-related project management activities in terms of the ISO15288 using the methods and tools developed.



Inhaltsverzeichnis	Seite
1 Einleitung	1
1.1 Problematik.....	1
1.2 Zielsetzung	3
1.3 Vorgehensweise	4
2 Problemanalyse	7
2.1 Technische Systeme	7
2.1.1 Evolutionsstufen technischer Systeme	7
2.1.2 Herausforderung Technik	9
2.2 Grundlagen der Produktentstehung.....	10
2.2.1 Produktentstehung und Lebenszyklus	10
2.2.2 Entwicklungsprojekte und Entwicklungsmanagement.....	12
2.2.3 Organisations- und Unternehmensgestaltung.....	15
2.2.4 Stakeholder, Rollen und Kompetenzen.....	16
2.2.5 Methodisches Arbeiten	17
2.2.6 Herausforderung Produktentstehung	20
2.3 Systems Engineering.....	21
2.3.1 Entwicklung des Systems Engineerings	21
2.3.2 Wissenschaftliche SE-Konzepte	22
2.3.3 Themenfelder und Standards des Systems Engineerings	24
2.3.4 Der Systems Engineer	26
2.3.5 Herausforderung Systems Engineering	27
2.4 Model-Based Systems Engineering.....	27
2.4.1 Vision des Model-Based Systems Engineerings.....	28
2.4.2 Allgemeine Modelltheorie als Basis für MBSE	29
2.4.3 Das Systemmodell	30
2.4.4 Sprachen, Methoden und Werkzeuge.....	32
2.4.5 Erstellung und Nutzung des Systemmodells.....	36
2.4.6 Product Lifecycle Management und Workflows	39
2.5 Herausforderung MBSE.....	41
2.5.1 Nutzen und Wertbeitrag von MBSE	41
2.5.2 Studien zur Verbreitung von SE und MBSE in der Praxis.....	43
2.5.3 Der Weg zu MBSE.....	45
2.6 Problemabgrenzung	45
2.7 Anforderungen.....	50

3	Stand der Technik	53
3.1	Beschreibungs- / Vorgehensmodelle der Produktentstehung.....	53
3.1.1	3-Zyklen-Modell der Produktentstehung	54
3.1.2	iPeM – Integriertes Produktentstehungsmodell	55
3.1.3	Modellbasierte virtuelle Produktentwicklung nach EIGNER	56
3.1.4	W-Modell nach NATTERMANN/ANDERL.....	58
3.2	MBSE-Modellierungsmethoden	60
3.2.1	CONSENS	60
3.2.2	SYSMOD	62
3.2.3	Harmony SE	63
3.2.4	OPM – Object-Process Methodology.....	65
3.2.5	oosem	66
3.2.6	LITHE.....	68
3.3	Frameworks	70
3.3.1	ISO/IEC/IEEE42010.....	70
3.3.2	DoDAF – Department of Defense Architecture Framework	72
3.3.3	Kruchten 4+1	74
3.3.4	A3 Architecture Framework	76
3.4	Prozess-, Methodenmodelle und Methodensammlungen.....	78
3.4.1	Forflow	78
3.4.2	Prozessorientiertes Methodenmodell nach BIRKHOFER.....	80
3.4.3	Münchener Methodenmodell	81
3.4.4	Situative Entwicklungsaufgaben- und Methodenauswahl	83
3.5	Organisationskonzepte im Systems Engineering	84
3.5.1	SE-Team nach FRIEDENTHAL.....	84
3.5.2	„The Systems Engineer in the Engineering Organization“	85
3.6	Handlungsbedarf	86
4	Rahmenwerk	91
4.1	Überblick über das Rahmenwerk.....	91
4.2	Konzepte des modellbasierten Systems Engineerings	93
4.2.1	Merkmale und Ausprägungen von MBSE	94
4.2.2	Klassen konsistenter Merkmalausprägungen	102
4.2.3	Detaillierung und Analyse der MBSE-Konzepte.....	104
4.3	Entscheidungsunterstützung zur Auswahl und Anwendung von MBSE-Konzepten	108
4.3.1	Lebenszyklusorientierte Auswahl und Anwendung.....	109
4.3.2	Zweckabhängige Auswahl und Anwendung	111
4.3.3	Aufgabenorientierte Auswahl und Anwendung	112
4.3.4	Kompetenzorientierte Auswahl und Anwendung	113

4.4	Konzept zur Nutzung des Systemmodells in der Produktentstehungsarbeit.....	114
4.4.1	Grundverständnis	115
4.4.2	Systemtechnisches Prinzip der Informationsverarbeitung als Grundlage zur Anwendung von Methoden mit dem Systemmodell.....	116
4.4.3	Methodenanwendung durch „Trennung von Modell und Sicht“ ..	117
4.5	Flexibles Vorgehensmodell zur Planung der Systemmodellierung in der Produktentstehung.....	119
4.5.1	Das Vorgehensmodell im Detail.....	119
4.5.2	Idealtypische Vorgehensweisen	121
4.6	Methoden und Hilfsmittel im flexiblen Vorgehensmodell	124
4.6.1	Define-Phase	124
4.6.1.1	Define – Informationserzeugung (IE)	125
4.6.1.2	Define – Informationsgewinnung, -verarbeitung, -ausgabe ..	129
4.6.2	Analyse-Phase.....	134
4.6.2.1	Analyse-Phase: Informationserzeugung	134
4.6.2.2	Analyse-Phase – Informationsgewinnung	140
4.6.2.3	Analyse-Phase – Informationsverarbeitung und -ausgabe...	146
4.6.3	Informationserzeugung – Design und Control.....	149
4.6.4	Design-Phase	151
4.6.5	Control-Phase	154
5	Anwendung und Bewertung des Rahmenwerks	155
5.1	Konzepte des MBSE	155
5.1.1	Überblick über die Untersuchungseinheiten	155
5.1.2	Klassifikation der Untersuchungseinheiten	157
5.1.3	Zusammenfassung und Bewertung	161
5.2	Anwendung von Methoden auf Basis des Systemmodells	163
5.2.1	Vorstellung des Praxisbeispiels	163
5.2.2	Gefährdungsanalyse in Anlehnung an NOHL (SysML4CONSENS).....	164
5.2.3	Komplexitätsanalyse des Kaffeeautomaten in Anlehnung an MARTI (SysML4CONSENS)	166
5.3	Bewertung der Arbeit anhand der Anforderungen	173
6	Zusammenfassung und Ausblick	177
7	Abkürzungsverzeichnis	181
8	Literaturverzeichnis	187

Anhang

A1	Ergänzungen zur Problemanalyse.....	A-1
A1.1	Prozessbereiche der ISO15288-2015	A-1
A1.2	Zuordnung von SE-Rollen zu einem Unternehmen des Mittelstand ..	A-2
A2	Ergänzungen zu den erarbeiteten MBSE-Konzepten	A-2
A2.1	Visualisierung der MBSE-Konzepte.....	A-2
A2.2	Vollständige Ausprägungsliste der MBSE-Konzepte	A-3
A2.3	Kompetenzorientierte Auswahl von MSBE-Konzepten	A-5
A2.4	Ergänzung zu Beispielprojekten ohne Quellenangabe	A-5
A3	Ergänzungen zum flexiblen Vorgehensmodell	A-7
A3.1	Kurzüberblick und Hinweise zu Six Sigma.....	A-7
A3.1.1	DMAIC-Circle	A-7
A3.1.2	Six Sigma-Rollen	A-8
A3.1.3	Reifegradmodell der Einführung von Six Sigma	A-9
A3.2	Übersicht über entwickelte, erweiterte und eingesetzte Methoden und Hilfsmittel im flexiblen Vorgehensmodell	A-10
A4	Merkmale Morphologie der Modellierungsmethoden.....	A-10
A5	Ergänzungen zu den Modellen des Kaffee-Automaten	A-14
A5.1	Umfeldmodell mit SysML4CONSENS	A-14
A5.2	Auszug Partialmodell Funktionshierarchie (CONSENS).....	A-15
A5.3	Auszüge aus dem Partialmodell Wirkstruktur (CONSENS)	A-16
A5.4	Auszug aus der Modellierung mit Sysmod.....	A-17
A6	Ergänzungen zu den gekoppelten Methoden	A-17
A6.1	Stakeholderanalyse	A-17
A6.1.1	Stakeholderanalyse nach MÜLLER-STEWENS und LECHNER	A-17
A6.1.2	Web-Applikation Stakeholderanalyse	A-18
A6.2	Gefährdungsanalyse	A-19
A6.2.1	Web-Applikation Gefährdungsanalyse.....	A-19
A6.3	Mechatronische Komplexitätsanalyse, angelehnt an MARTI	A-19
A6.3.1	Formel zur Errechnung der Komplexität nach MARTI	A-20
A7	Ergänzende Erläuterungen zur IT-Infrastruktur	A-21
A8	Auszüge aus dem Datenmodell.....	A-22

1 Einleitung

„Die Technik in ihrem Wesen ist etwas, was der Mensch von sich aus nicht bewältigt.“ [Heidegger]

1.1 Problematik

Mechatronische Systeme haben Einzug in sämtliche Bereiche unseres Lebens gefunden und werden als selbstverständlich betrachtet. Mechatronik ist kein Alleinstellungsmerkmal mehr. Dennoch: Die hohe Interdisziplinarität dieser Systeme resultiert in einer schwierig handhabbaren Produktkomplexität. Zudem ergibt sich aus der Globalisierung mit dem daraus resultierenden Wettbewerbsdruck und geographisch verteilten Kooperationen ein anspruchsvolles Anforderungsprofil: Der Markt fordert Hightech zu niedrigen Kosten, die Innovationszyklen werden kürzer, Fachleute verschiedener Nationalitäten und Kulturkreise müssen wirksam zusammenarbeiten. Das führt dazu, dass die Produktentstehung technischer Systeme immer effizienter gestaltet werden muss, obwohl die mit der Produktkomplexität einhergehende Prozess- und Organisationskomplexität nur schwer zu beherrschen ist [Lin09], [Sch05]. Die Vision von technischen Systemen mit inhärenter Teilintelligenz verstärkt diese Entwicklung [GTD13], [Dum11].

Trotz qualifizierter Fachleute, einem starken Fokus auf Wissens- und Qualitätsmanagement und einem intensiven Projektmanagement kommt es in der Produktentstehung aber immer wieder zu großen Problemen. Ursache ist die fehlende Transparenz über das technische System und das dazugehörige organisatorische System [GAC+13]. Erschwerend wirkt, dass jeder Produktentstehungsprozess einzigartig ist und somit immer wieder neuen, unbekannteren Situationen zu begegnen ist [Alb10]. Es zeigt sich, dass diesen Herausforderungen nicht mehr mit den Methoden der Fachdisziplinen begegnet werden kann, sondern eine ganzheitliche Sichtweise auf die Produktentstehung als sozio-technisches System notwendig ist [AG12].

Eine gemeinsame „Sprache“ der Produktentstehung in Ergänzung zu den Arbeitsweisen der Fachdisziplinen kann hier Abhilfe schaffen [VDI2206]. Somit ist es auch nicht verwunderlich, dass das Systems Engineering (SE) als fachdisziplinübergreifender Ansatz eine Art Renaissance erlebt [Gau10b] und mit dem modellbasierten Systems Engineering¹ (MBSE) auf eine neue Leistungsstufe gehoben werden soll [INC15]: Grundlage des MBSE ist eine fachdisziplinübergreifende Systembeschreibung auf Basis grafischer Notationen; Kern ist ein sog. Systemmodell, das der SE-Vision der INCOSE zufolge als Dreh- und Angelpunkt sämtlicher Tätigkeiten in einem Unternehmen fungieren soll und

¹ Die Begriffe *Modellbasiertes Systems Engineering*, *Model-Based Systems Engineering* sowie *MBSE* werden in dieser Arbeit weitgehend synonym verwendet.

damit entlang des Produktlebenszyklus alle dafür relevanten Daten enthält [INC15], [FMS12], [GFD+09].

Es ist allerdings unklar, wie die junge Disziplin MBSE im Sinne des Systems Engineerings in die tägliche Produktentstehungsarbeit integriert werden und der SE-Vision entsprechend wirken kann: Während Standards und Richtlinien des SE – wenn auch nicht immer einheitlich [Hon04] – sämtliche Aktivitäten des Produktlebenszyklus adressieren [ISO15288] und damit einen ganzheitlichen Ansatz stellen, greifen aktuelle Arbeiten des MBSE viel kürzer [TBD+15]: Ihr Fokus liegt meist auf der reinen Architekturbeschreibung und der sehr detaillierten Spezifikation [RFB13], häufig von Software und Embedded Systems. Ebenso werden meist nur werkzeugtechnische Formalismen adressiert [Bon14], häufig mit dem Ziel der detailgetreuen Simulation sehr spezifischer Fragestellungen. Die Unterstützung von Aufgaben des Systems Engineerings wie z.B. die Definition von Kundengruppen, Komplexitätsanalysen und Risikoanalysen als typischer Teil von Produktentstehungsprojekten werden bislang nicht adressiert – in Anlehnung an die ISO/IEC15288² sind im Prinzip sämtliche Tätigkeiten jenseits der technischen Aktivitäten der Produktentstehung ausgeschlossen [ISO15288]. Der starke Fokus auf die Weiterentwicklung von Details und Formalismen an Sprachen und Werkzeugen hat bislang viele potentielle Stakeholder von diesem vielversprechenden Ansatz ausgeschlossen, z.B. Projektmanager, Einkäufer und Produktmanager [Zin13].

Gerade deshalb empfinden selbst erfahrene SE-Praktiker das MBSE als zu kompliziert, zu werkzeuggetrieben und aufgrund der unklaren Einbindung in die Prozesse als zu wenig leistungsfähig [Clo13], [Sto12a]. Sie vermissen die Idee, dass Systems Engineering die Produktentstehungsarbeit orchestriert [LML13] und somit konkrete Tätigkeiten des Systems Engineers unterstützt; das sind bspw. im Sinne o.g. Richtlinien auch produktnahe Managementaktivitäten. Generell ist also nicht verwunderlich, dass viele Stakeholder der Produktentstehung MBSE nicht kennen, den Aufwand für die Erstellung des Systemmodells scheuen oder seinen Nutzen nicht erkennen [Kno14], [BC10], [GDS+13a], [Sto12b], [BC10]. Zudem ist durch die in kürzester Zeit entstandene Fülle von rein technisch getriebenen Arbeiten das Ziel von MBSE nicht eindeutig [Gaa10], zumal das Thema aus verschiedenen technischen Fachdisziplinen mit unterschiedlichen Beweggründen getrieben wird.

Die Darstellungen zeigen deutlich: Zwischen dem Anspruch von Systems Engineering als ganzheitlicher Ansatz der Produktentstehung und der Wirklichkeit von MBSE als seine Weiterentwicklung klafft eine Lücke. Die flächendeckende Anwendung von MBSE in der Praxis ist momentan nicht mehr als eine kühne Vision [GDS+13a]. Zwar benötigt der Transfer neuer Methoden und Arbeitsweisen in die alltägliche Produktentstehungspraxis lange Zeit; die vollständige Integration des CAD/CAM-Ansatzes hat gut

² In Anhang A1.1 sind die Prozessgebiete der ISO15288:2015 im Überblick dargestellt.

40 Jahre benötigt [RFB12a], [Sto12b] – im Vergleich betrachtet steckt das modellbasierte Systems Engineering noch in seinen Kinderschuhen.

Aus diesem Grund gilt es umso mehr, frühzeitig Konzepte und Vorgehensweisen zur Integration von MBSE in die Produktentstehung bereitzustellen und es damit auch stärker der ursprünglichen Idee des Systems Engineerings anzunähern: MBSE muss als ganzheitliches Paradigma der Produktentstehung beschrieben werden, um seinen Einsatz für die zahlreichen Aufgaben der Produktentstehung zielgerichtet gestalten zu können. Von zentraler Bedeutung ist hierbei, den Nutzen von MBSE aufzuzeigen – das kann gelingen, indem die Aufgaben der MBSE-Stakeholder individuell unterstützt werden.

Vor dem Hintergrund der dargestellten Problematik besteht Bedarf für ein *Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme*.

1.2 Zielsetzung

Ziel der Arbeit ist ein *Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme*. Durch das Rahmenwerk soll der zielgerichtete Einsatz von MBSE in der Produktentstehung möglich werden. Das umfasst auch den Einsatz des MBSE für die zahlreichen Aktivitäten in Produktentstehungsprojekten jenseits der reinen Entwicklungstätigkeiten – bspw. für das technische Management von Produktentstehungsprojekten. So soll die Idee des MBSE im Sinne der Kernidee des Systems Engineerings ausgeprägt werden – als ein ganzheitlicher Ansatz, mit dem sämtliche Aktivitäten von Produktentstehungsprojekten orchestriert werden können. Konkret soll das Rahmenwerk die folgenden Elemente umfassen:

- **Beschreibungsmodelle:** Es soll herausgearbeitet werden, was die wesentlichen Merkmale des MBSE sind. Damit soll für alle Stakeholder ein klares Verständnis zum Kern des MBSE, seinen Zielen und seiner Reichweite ermöglicht werden. Ebenso soll dargestellt werden, wie MBSE in den Produktentstehungsalltag integriert werden kann. Dabei wird davon ausgegangen, dass es ähnlich zu den zahlreichen Ausprägungen des Systems Engineerings grundsätzlich auch unterschiedliche MBSE-Konzepte geben kann.
- **Entscheidungshilfen:** Sie sollen den rationalen Einsatz von MBSE unterstützen. Dadurch soll – aufbauend auf den Beschreibungsmodellen – ein Beitrag zur zielgerichteten und bedarfsgerechten Integration des Ansatzes in die Produktentstehungsarbeit geleistet werden.
- **Vorgehensmodell, Methoden und Hilfsmittel:** Sie sollen die Planung, Modellierung und Anwendung des Systemmodells als Kern des modellbasierten Systems Engineerings für typische Aufgaben in Produktentstehungsprojekten unterstützen. Insb. soll das für technische Managementprozesse dargestellt werden, aber

auch so gestaltet sein, dass es für beliebige Anwendungen infrage kommt. Dadurch sollen konkrete Nutzenpotentiale des MBSE für die zahlreichen Stakeholder strukturiert erschlossen werden.

Das Rahmenwerk soll die Anwendung von MBSE in der täglichen Produktentstehungsarbeit befähigen. Seine Praxisrelevanz soll anhand einer empirischen Recherche zur Verbreitung des MBSE und ausgewählter Beispiele aufgezeigt werden. Auf diese Weise soll folgendem Umstand Rechnung getragen werden: MBSE befindet sich in einer jungen Entwicklungsphase; die Akzeptanz der Handlungspersonen und insb. des Managements für MBSE kann nur geschaffen werden, wenn einzelne und einfache Hilfsmittel mit direkter Wirkung bereitgestellt werden.

1.3 Vorgehensweise

Die vorliegende Arbeit ist in sechs Kapitel gegliedert. Nach der Einleitung in diesem Kapitel wird in **Kapitel 2** die skizzierte Problematik ausführlich diskutiert. Die Breite des Themenfelds erfordert aufbauend auf einer kurzen Darstellung der Eigenschaften heutiger technischer Systeme zunächst eine intensive Diskussion des **Handlungsfelds Produktentstehung** – hier soll MBSE eingesetzt werden. Somit spielen Entwicklungsprozesse und der Produktlebenszyklus ebenso eine Rolle wie Aspekte der Organisationsgestaltung, aber auch das Projektmanagement und insb. methodisches Arbeiten. Ein weiterer Schwerpunkt des zweiten Kapitels ist das **Handlungsfeld Systems Engineering**, bzw. Model-Based Systems Engineering. Die Idee und Herkunft der verschiedenen Ansätze werden ebenso ausführlich analysiert, wie grundlegende Konzepte und aktuelle Entwicklungen und Herausforderungen. Dadurch wird die skizzierte Problematik gestützt. Im Model-Based Systems Engineering wird zudem die Idee des Product Lifecycle Managements (PLM) für das Systems Engineering reflektiert. Die Betrachtungen erlauben die Konsolidierung des Handlungsbedarfs und die **Ableitung von Anforderungen** an das Rahmenwerk.

Die Ergebnisse der Literaturrecherche zum **Stand der Technik** werden in **Kapitel 3** vorgestellt. Aufbauend auf den in Kapitel 2 diskutierten Handlungsfeldern wird eine Trennung in verschiedene Analysebereiche vorgenommen. Das sind zunächst bekannte **Beschreibungsmodelle der Produktentstehung**, die den Anspruch erheben, im Sinne des Systems Engineerings ausgestaltet zu sein. Weiterhin werden ausgewählte **MBSE-Modellierungsmethoden** und **Frameworks** dahingehend untersucht, wie sehr sie die Aktivitäten der Produktentstehung und die Planung der Modellierung und Nutzung des Systemmodells berücksichtigen. Ebenso wird analysiert, wie etablierte **Prozess- und Methodenmodelle** die Idee des MBSE unterstützen. Schlussendlich stehen **Organisationskonzepte** des Systems Engineerings auf dem Prüfstand. Hier steht die Frage im Mittelpunkt, ob sie ein Arbeiten mit MBSE ermöglichen. Die Analyse der zahlreichen Ansätze kann auch dazu dienen, mögliche geeignete Einzelelemente für das Rahmenwerk

zu identifizieren. Die Bewertung der untersuchten Arbeiten und Ansätze hinsichtlich der Anforderungen erlaubt die abschließende **Bestätigung des Handlungsbedarfs**.

Kapitel 4 bildet den Kern der vorliegenden Arbeit mit den Bestandteilen des Rahmenwerks. Grundannahme ist dabei, dass es unterschiedliche Ausprägungen des MBSE gibt. Die verschiedenen Ausprägungen werden in einem Beschreibungsmodell in Form von **MBSE-Konzepten** erarbeitet. Diese bilden die Basis für die weiteren Elemente des Rahmenwerks, insb. die **Entscheidungshilfen** zum Einsatz von MBSE. Ein weiteres Beschreibungsmodell stellt die **Einbindung des modellbasierten Systems Engineerings in die Produktentstehung** dar. Hierauf baut ein flexibles **Vorgehensmodell zur Planung der Systemmodellmodellierung** auf, das die Planung, Erstellung und Nutzung des Systemmodells für typische Aufgaben der Produktentstehung ermöglicht. Das generische Vorgehen wird um zahlreiche Methoden und Hilfsmittel und ein Anwendungsbeispiel ergänzt.

Die Praxisrelevanz des Rahmenwerks und seine Eignung zur Unterstützung der Produktentstehung werden in **Kapitel 5** aufgezeigt. Eine **Studie** zu Beginn des Kapitels soll zunächst die Verbreitung der MBSE-Konzepte belegen. Im Anschluss wird anhand des **Anwendungsbeispiels** aus Kapitel 4 die Anwendbarkeit weiterer Bestandteile des Rahmenwerks dargestellt. Das Kapitel schließt mit einer Bewertung des Rahmenwerks hinsichtlich der Anforderungen aus Kapitel 2.

Kapitel 6 besteht aus einem **Resümee** der Arbeit sowie einem **Ausblick** auf zukünftige Forschungsfelder. Der **Anhang** enthält ergänzende Informationen.

2 Problemanalyse

Ziel der Problemanalyse sind Anforderungen an ein *Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme*. Hierzu ist Kapitel 2 wie folgt aufgebaut: Die hier fokussierten technischen Systeme werden in Kapitel 2.1 eingeführt, gefolgt von einer Vorstellung der Produktentstehung dieser Systeme in Kapitel 2.2. Die Produktentstehung wird dabei als sozio-technisches Handlungssystem herausgearbeitet. Die Kapitel 2.3 bis 2.5 diskutieren die Bedeutung und die Leistungsfähigkeit des Systems Engineering und des modellbasierten Systems Engineering für die Produktentstehung. In dieses Themenfeld ordnet sich die Arbeit ein. Abschließend erfolgt eine Problemabgrenzung (Kapitel 2.6) aus der die Anforderungen an ein Rahmenwerk abgeleitet werden (Kapitel 2.7).

2.1 Technische Systeme

2.1.1 Evolutionsstufen technischer Systeme

Erzeugnisse des Maschinenbaus und verwandter Branchen wie der Automobiltechnik, Bahntechnik und Medizintechnik waren lange Zeit durch mechanische Systeme geprägt. Die Anzahl an mechanischen Funktionen in diesen Systemen hatte zu Beginn der 1970er Jahre einen Anteil von gut 90% [SB09]. Die Entwicklung konnte somit aus der Perspektive der Konstruktion gesteuert werden, der geringe Anteil an Elektronik und ggf. Software hatte nur marginalen Einfluss auf Entwicklungsaktivitäten.

Heutige Systeme beruhen größtenteils auf einem engen Zusammenwirken von Mechanik, Elektronik und Softwaretechnik, was durch den Begriff Mechatronik zum Ausdruck kommt und eine elektronische Funktionserweiterung (*-tronik*) mechanischer Komponenten (*Mecha-*) bezeichnet. Mit dem Aufkommen der Mikroprozessortechnik erweiterte sich das Verständnis der Mechatronik unweigerlich um eine informationstechnische Komponente. Die VDI-Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ bezeichnet Mechatronik als das synergetische Zusammenwirken der ingenieurwissenschaftlichen Disziplinen Maschinenbau, Elektrotechnik und Informationstechnik [VDI2206]. Das umfasst auch die dazugehörige Fertigungsprozessentwicklung [HTF96].

Trotz einfacher Definition existiert bis heute kein einheitliches Bild, was Mechatronik alles umfasst. Der Begriff und der Kern mechatronischer Systeme entwickeln sich kontinuierlich weiter, wie das Bild 2-1 andeutet.

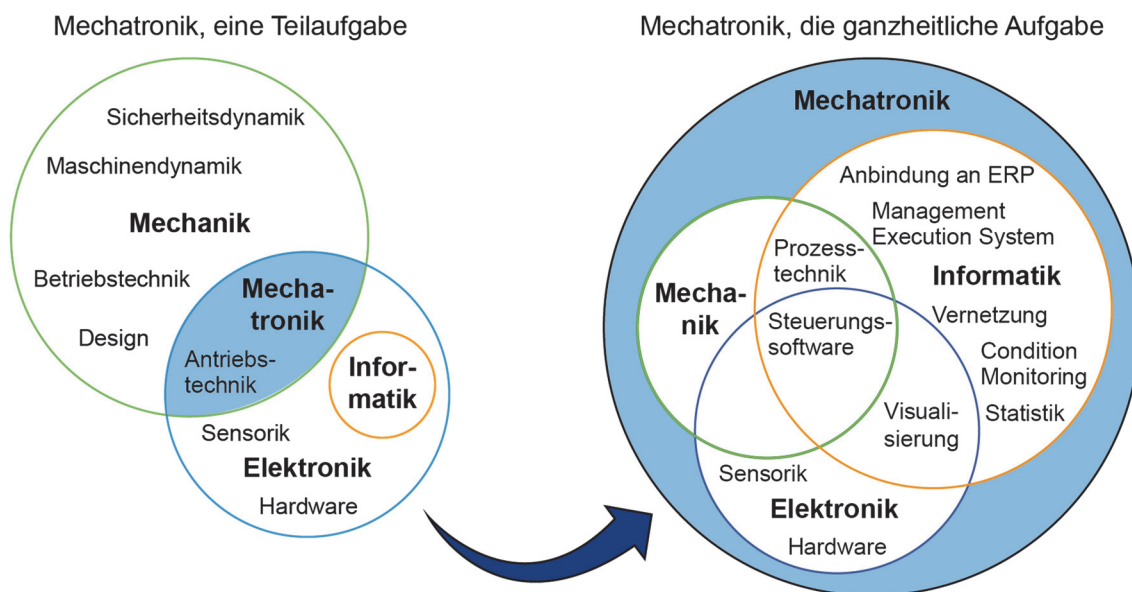


Bild 2-1: Maschinenbau gestern und morgen – Verständniswandel nach [ERZ14]

Die Struktur eines mechatronischen Systems zeichnet sich durch die funktionale und/oder räumliche Integration von einem Grundsystem, Sensoren, Aktoren und Informationsverarbeitung aus. Diese vier Elemente bilden einen systeminternen Regelkreis. Die Beziehungen zwischen den jeweiligen Elementen können mit Hilfe von Flussbeziehungen dargestellt werden. Es sind grundsätzlich drei Arten von Flüssen zu unterscheiden: Stofffluss, Energiefluss und Informationsfluss [PBF+07].

- **Stoffflüsse:** Sie beschreiben den Austausch von Stoffen, wie bspw. Festkörpern, Flüssigkeiten und Gasen zwischen Einheiten mechatronischer Systeme.
- **Energieflüsse:** Energie fließt zwischen Systemen in Form von mechanischer, thermischer oder elektrischer Energie, ebenso wie Kraft oder Strom.
- **Informationsflüsse:** Der Informationsaustausch findet beispielsweise durch Messgrößen, Steuerimpulse oder Daten statt.

An der Realisierung des mechatronischen Systems sind verschiedenste Fachdisziplinen beteiligt. Ausgehend von der Verteilung der Fachdisziplinen an der Wertschöpfung in den 1970er Jahren nimmt die Verschiebung der Anteile der einzelnen Fachdisziplinen in den technischen Systemen heute rasant zugunsten der Elektronik und der Softwaretechnik zu. Die Tiefe und die Komplexität der Mechatronisierung werden dabei von den Produkten bzw. der Branche bestimmt, wobei in heutigen Systemen von einem 70-prozentigen Anteil an Elektronik und Software ausgegangen werden muss [SB09]. Die Informationstechnik und auch nichttechnische Disziplinen, wie die Kognitionswissenschaft oder die Neurobiologie bringen eine Vielfalt an Methoden, Techniken und Verfahren hervor, mit denen sensorische, aktorische und kognitive Funktionen in technische Systeme integriert werden, die man bislang nur von biologischen Systemen kannte.

Das ermöglicht Intelligente Technische Systeme (ITS) [Gau10a], die sich durch vier zentrale Eigenschaften auszeichnen – sie sind adaptiv, robust, vorausschauend und benutzungsfreundlich [Dum11]. Dabei verschmilzt die physikalische Welt, in der diese Systeme agieren, mit der virtuellen Welt (dem Cyber-Space), über die sie miteinander vernetzt sind und kommunizieren. Derartige intelligente technische Systeme werden als Cyber-Physical Systems (CPS) bezeichnet, die Self-X-Fähigkeiten besitzen und die Grundlage für Schwarmintelligenz und die Verwirklichung einer vierten industriellen Revolution sind [GTD13], [acatech11].

Aus dieser Entwicklung lassen sich aus Sicht des Maschinenbaus drei Klassen technischer Systeme unterscheiden (vgl. Bild 2-2). Ausgehend von der Mechanik haben maschinenbauliche Systeme einen Sprung zur Mechatronik gemacht. Die dritte Klasse – Intelligente Technische Systeme – hat eine überproportional hohe Komplexität und eine höchst interdisziplinäre Entwicklung. Noch stärker als mechatronische Systeme ist die größte Herausforderung der Systementwurf. Die Entwicklung diese Klasse zeichnet sich gerade im Verlauf der sog. 4. Industriellen Revolution ab [GAC+13].

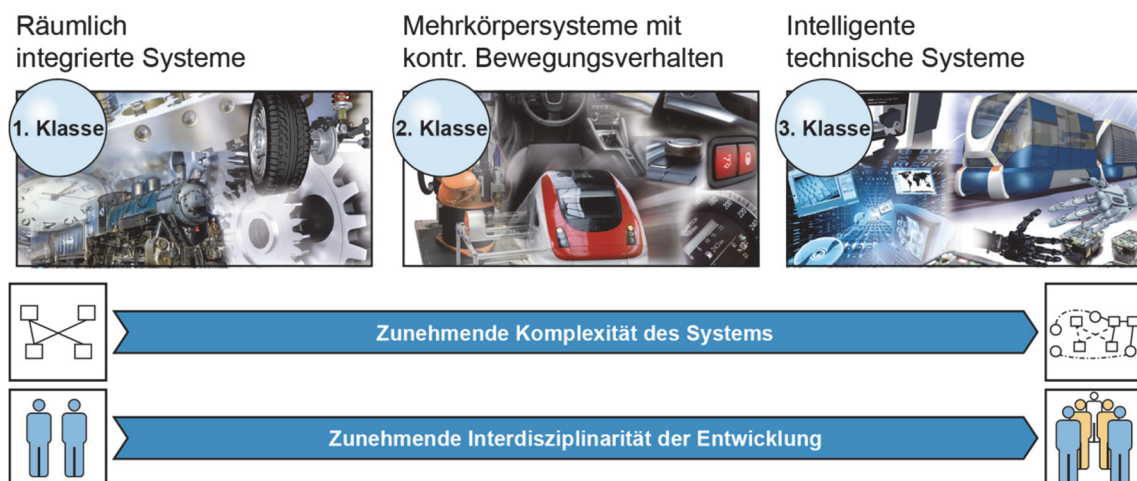


Bild 2-2: Klassen maschinenbaulicher Systeme, in Anlehnung an [GAC+13]

2.1.2 Herausforderung Technik

Die Kreation technischer Systeme steht vor dem Sprung hin zu Intelligenten Technischen Systemen. Anhand der skizzierten Klassen maschinenbaulicher Systeme sind die zunehmende Komplexität dieser Systeme und die verbundenen Herausforderungen nachvollziehbar. Die drei Hauptherausforderungen sind die wachsende Interdisziplinarität der Systeme, eine hohe Schnittstellenvielfalt zwischen den verschiedenen Systemen und Subsystemen und in gewisser Weise daraus resultierend eine hohe Anforderungskomplexität (vgl. Bild 2-3).

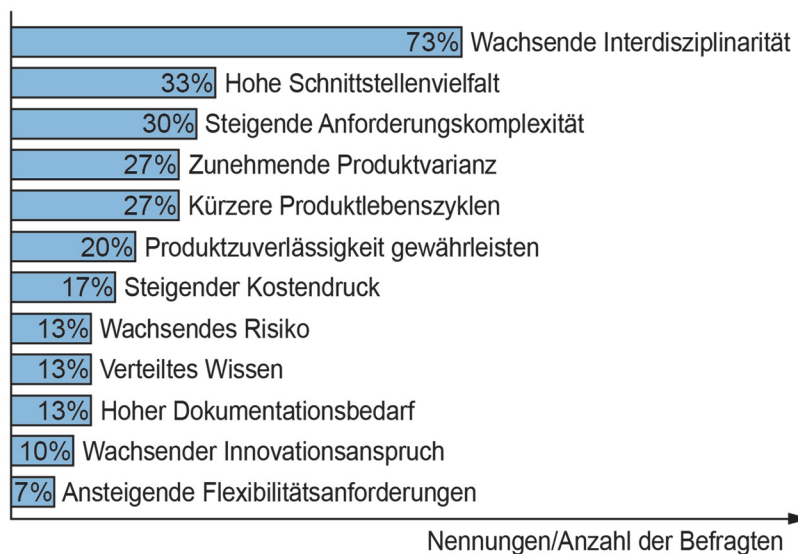


Bild 2-3: Herausforderungen in der Produktentstehung von morgen [GDS+13b]

Insb. im Falle von Intelligenten Technischen Systemen verschärfen die drei erstgenannten Komplexitätstreiber Interdisziplinarität, Schnittstellenvielfalt und Anforderungskomplexität die Situation. Klar ist: Mechatronische Systeme und noch viel mehr ITS müssen aus dem Blickwinkel mehrerer – bislang auch unbeteiligter – Fachdisziplinen betrachtet werden und können daher nicht nur mit ihrer Methodik allein entwickelt werden [GDS+13b]. Darüber hinaus gilt: Gerade weil es aus technischer Sicht enorme Herausforderungen bei der Kreation technischer Systeme gibt, rücken insb. die Schaffung eines einheitlichen Verständnisses über die Aufgaben und die Koordination der Entwicklungsaktivitäten in den Vordergrund. Gleichzeitig wird durch diese Systeme ein Wandel in der Arbeitswelt erfolgen, der zwingend eine Weiterbildung und Weiterqualifikation der Mitarbeiter/-innen der Unternehmen erfordert [And15].

2.2 Grundlagen der Produktentstehung

Die Kreation der hier adressierten Systeme erfolgt im Rahmen der Produktentstehung, die den grundsätzlichen Ablauf von der Produkt- bzw. Geschäftsidee bis zum Serienanlauf beschreibt. Nach der Produktion schließt sich die Marktphase als zweite Hauptphase des Produktlebenszyklus an. In diesem Kapitel werden die für die Arbeit wesentlichen Aspekte der Produktentstehung und des Produktlebenszyklus diskutiert.

2.2.1 Produktentstehung und Lebenszyklus

Jedes System durchläuft einen Lebenszyklus. Dieses Konzept der Betriebswirtschaftslehre beruht auf der Annahme, dass die zeitliche Entwicklung von produktspezifischen Indikatoren in charakteristische Phasen unterteilt werden kann [MR09], [Mar04], [Mar15] – im einfachsten Fall werden im Produktlebenszyklus die zwei Hauptphasen Produktentstehung und Marktphase unterschieden [GLR+00] (vgl. Bild 2-4). Die Pro-

duktentstehung wird wiederum in die Phasen Strategische Produktplanung, Produktentwicklung und Produktionssystementwicklung unterteilt [GPW14], die Marktphase gliedert sich in die vier Phasen Fertigung, Distribution, Nutzung und Rücknahme.

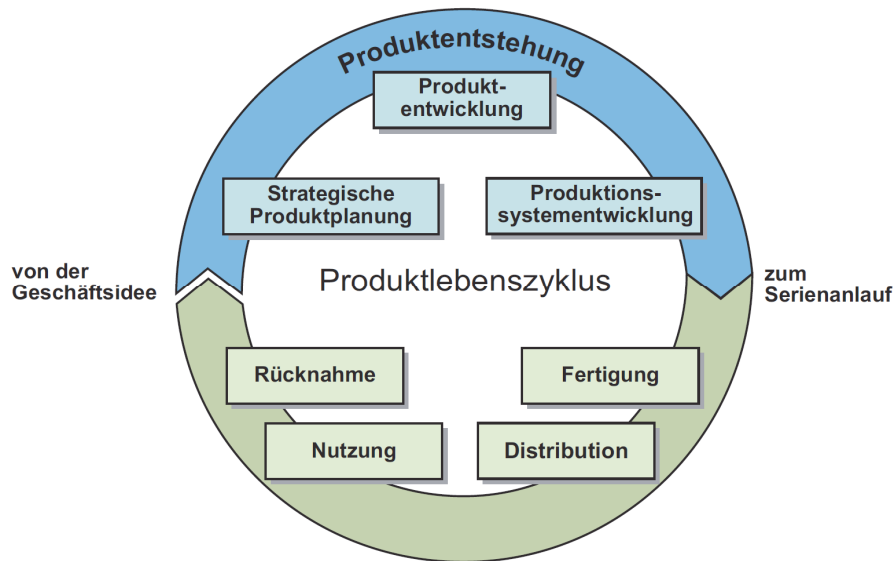


Bild 2-4: Lebenszyklus nach [GLR+00]

Basierend auf diesem generischen Produktlebenszyklus existieren zahlreiche Lebenszyklusmodelle, die sich je nach Herkunft, Branche oder Produkt marginal unterscheiden. Die Prozesse und Aktivitäten in den unterschiedlichen Phasen werden projekt- und unternehmensspezifisch so angepasst, dass spezifische Aufgaben und Ziele erreicht werden [ISO15288]. Die Phasenziele können nach Art, Zweck, Nutzung des Systems und weiterer spezifischer Umstände variieren. Jede Phase erfüllt dabei einen Zweck und erbringt einen Beitrag zur gesamten Lebensdauer, die bei der Planung und Betrieb des Systems Berücksichtigung finden [ISO15288]. Die ISO15288 versucht einen einheitlichen Produktlebenszyklus mit einheitlichen Prozessen und Begriffen für die einzelnen Phasen zu etablieren – die Verschiedenartigkeit der Produkte und die Anforderungen der jeweiligen Branchen verhindern jedoch oft diese Entwicklung (vgl. Bild 2-5).

Typischer Lebenszyklus (Generic Life Cycle (ISO 15288:2002))

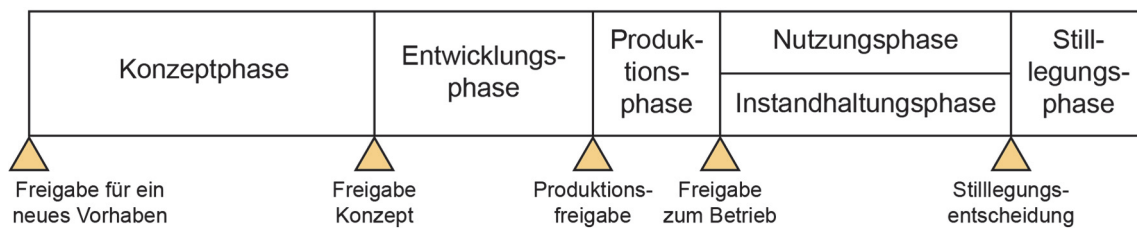


Bild 2-5: Lebenszykluskonzept der ISO 15288 [ISO15288]

Das ist auch dadurch bedingt, dass Unternehmen in Abhängigkeit ihrer Branche unterschiedlich, aber immer schneller auf sich ändernde Markt- und Gesellschaftsbedingungen reagieren müssen; durch den Marktdruck werden die Produktlebenszyklen kürzer

und neue Produkte und technologische Neuerungen müssen damit in immer kürzeren Produktentstehungsprozessen realisiert werden [Wit07]. Dazu wurde bspw. in der Automobilbranche die Entwicklungszeit stark reduziert. Bild 2-6 bringt diese Entwicklung am Beispiel des VW Golf zum Ausdruck

Generation	Zeitraum	Zykluslänge
Golf I	1974-1983	10 Jahre
Golf II	1983-1992	10 Jahre
Golf III	1991-1997	7 Jahre
Golf IV	1997-2003	7 Jahre
Golf V	2003-2008	6 Jahre
Golf VI	2009-2012	4 Jahre



Bild 2-6: Verkürzte Produktlebenszyklen am Beispiel VW Golf, www.wikipedia.de

Die Phase, in der Unternehmen Gewinne abschöpfen können, wird dadurch kürzer. Deshalb werden in der Marktphase kontinuierlich neue Varianten eingeführt, die meist nur auf kleinen Änderungen basieren. Dadurch kommt es zu starken Abhängigkeiten der Produkte untereinander, was sich noch stärker auf die Projekte auswirkt. Die Folge ist Intransparenz über das Produkt und seinen Entstehungsprozess.

2.2.2 Entwicklungsprojekte und Entwicklungsmanagement

„Ein Projekt ist ein zeitlich begrenztes Unternehmen, das unternommen wird, um ein einmaliges Produkt, eine Dienstleistung oder ein Ergebnis zu erzeugen.“ [PMI13]

Jedes Produkt – egal ob System, Modul oder Teil – durchläuft den Produktlebenszyklus und wurde im Rahmen eines Projekts im Produktentstehungsprozess geschaffen. Ein Projekt ist eine „einmalige, parallele und sequentielle Vernetzung von Aktivitäten“ mit definierter Ausgangslage und spezifiziertem Ergebnis [PR14]. Dabei sind mit Bezug auf den generischen Produktlebenszyklus nach [GLR+00] für das eigentliche Entwicklungsprojekt die Phasen Produktentwicklung und Produktionssystementwicklung relevant – wenngleich die Weichen in der Produktplanung gestellt werden. Das Projekt endet mit dem Serienanlauf, obwohl häufig ein gewisser Nachlauf zu beobachten ist, in dem überprüft wird, ob das Produkt bzw. die entsprechenden Prozesse stabil sind. Danach geht das Projekt in die Aktivität der Produktverfolgung über [PBF+13] (vgl. Bild 2-7). Die in Bild 2-5 dargestellten Konzept- oder Studienphasen sind keine Phasen des eigentlichen Entwicklungs- oder „Markteinführungs-Projekts“, sondern sind für sich genommen ein Projekt mit dem Ziel, die Machbarkeit einer Idee zu eruieren. Ebenso verhält es sich je nach Verständnis sich mit der Vorentwicklung. Über die gesamte Entwicklungszeit werden immer häufiger Partnerschaften mit Zulieferern eingegangen [PBF+13]. Die Möglichkeit zur Kostenbeeinflussung liegt damit in den frühen Phasen der Produktentwicklung: Hier werden gut 75% der Kosten festgelegt [Dil78]. wären.

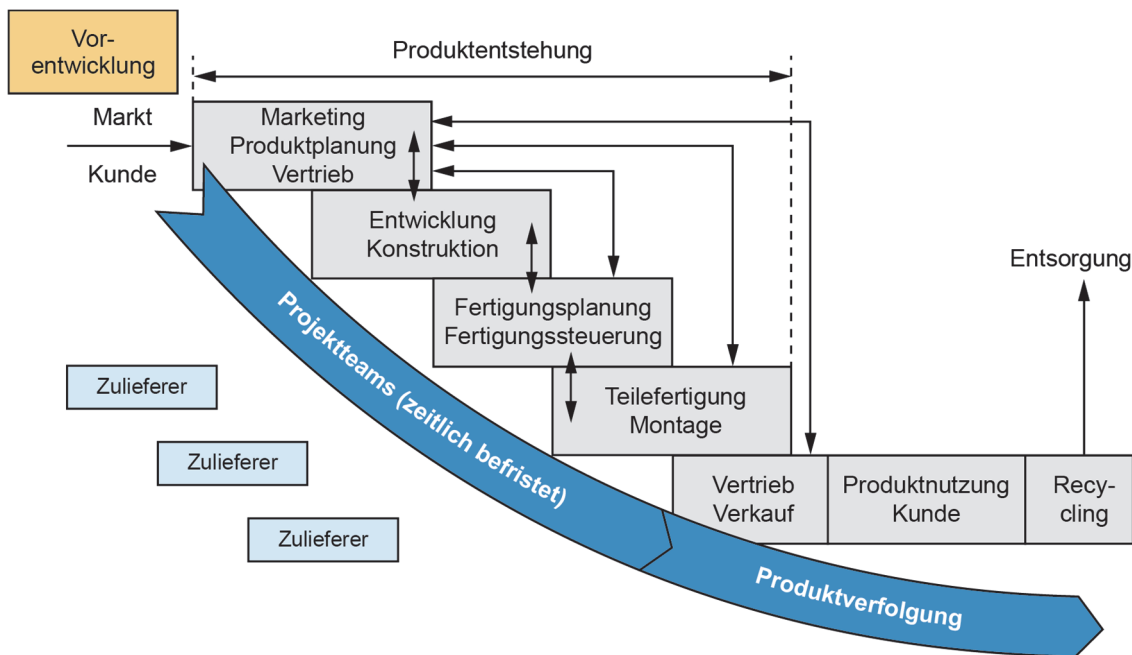


Bild 2-7: Das Projekt im Zeitverlauf, [PBF+13] mit eigenen Abänderungen

Die Ziele von Entwicklungsprojekten sind unterschiedlich: Die Einführung eines neuen Fahrzeugs („System“) ist genauso ein Projekt wie die Entwicklung eines ABS-Systems, das eine „Komponente“ im Fahrzeug ist. Die Bandbreite an Projekten lässt sich im Maschinenbau idealtypisch auf drei verschiedene Grundtypen reduzieren: Neukonstruktion, Anpassungskonstruktion und Variantenkonstruktion [PBF+07]. Die Neukonstruktion ist bislang eher die Seltenheit, wird durch die Vision intelligenter technischer Systeme aber zukünftig verstärkt aufkommen. Bislang herrschen jedoch die Anpassungskonstruktion oder die Variantenkonstruktion vor, jeweils mit spezifischen Auswirkungen auf das Projekt. Weitere Typologien detaillieren diese drei Grundkonzepte [Pau05].

Die Ausführungen zeigen: Ein Unternehmen ist durch ein Geflecht vieler, Projekte geprägt. Die Unterscheidung nach Projektarten dient daher vor allem dem projektspezifischen Einsatz von Methoden und Werkzeugen. [PR14] – die Kernaktivität des Entwickelns ist ohne ergänzende Aktivitäten und Methoden gar nicht mehr durchführbar. Aus diesem Grund wird immer häufiger die Bedeutung der Projektplanung und des Entwicklungsmanagements betont, so z.B. im iPeM [AB11]. Dennoch: Generell herrscht ein Mangel an Methoden und Richtlinien, die einfach angewendet werden können [SSG09], [PM06]. Abhilfe sollen Entwicklungshandbücher schaffen. Wie am Beispiel einer Landkarte dargestellt, kann die Unterstützung unterschiedlich detailliert ausfallen (vgl. Bild 2-8). Meist ist die Unterstützung der Aktivitäten auf Ebene 2 beschrieben, was nicht mehr als eine grobe Struktur ist. Die notwendige Beschreibung der Arbeitsschritte und Methoden für das Management der Produktentstehung zur Erreichung einheitlicher Arbeitsweisen über alle involvierten Bereiche (Ebene 4), ist nicht verfügbar³.

³ Hierbei handelt es sich um Auszüge aus dem Handbuch eines Großkonzerns. Gleiches gilt für Bild 2-9.

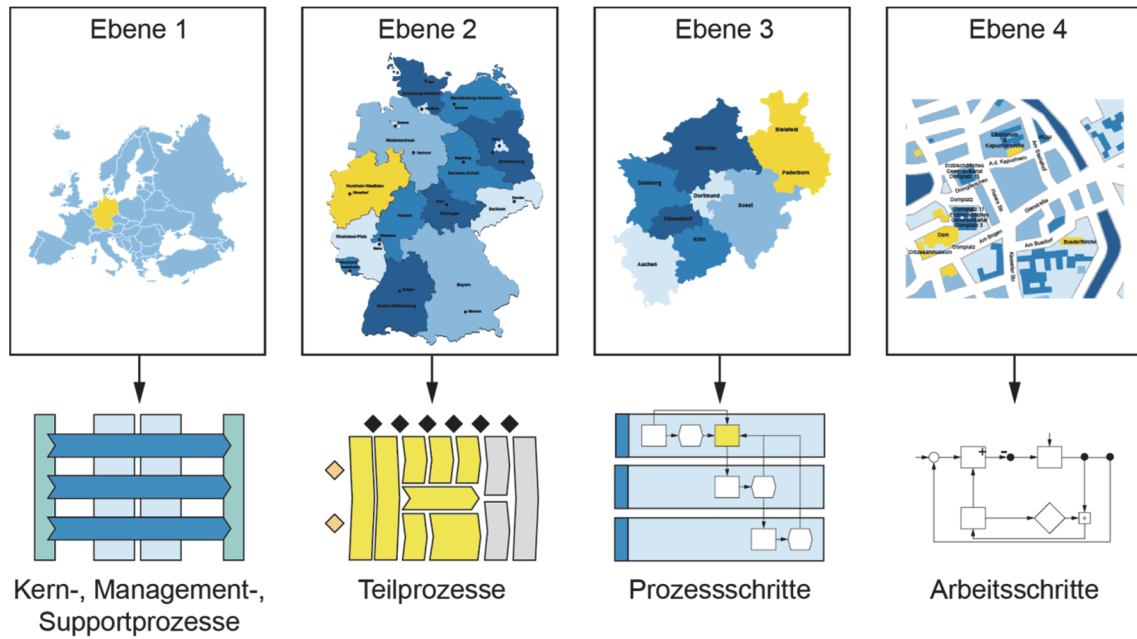


Bild 2-8 Prozessgranularität am Beispiel der Landkarte (eigene Darstellung)

Wie Prozesse stattdessen häufig dokumentiert sind, zeigt Bild 2-9. Das entspricht der Granularität der Ebene 3 aus Bild 2-8. Detailliertere textuelle Beschreibungen existieren durchaus, werden jedoch meist nicht verwendet, sondern „Pi-mal Daumen“ ausgelegt.

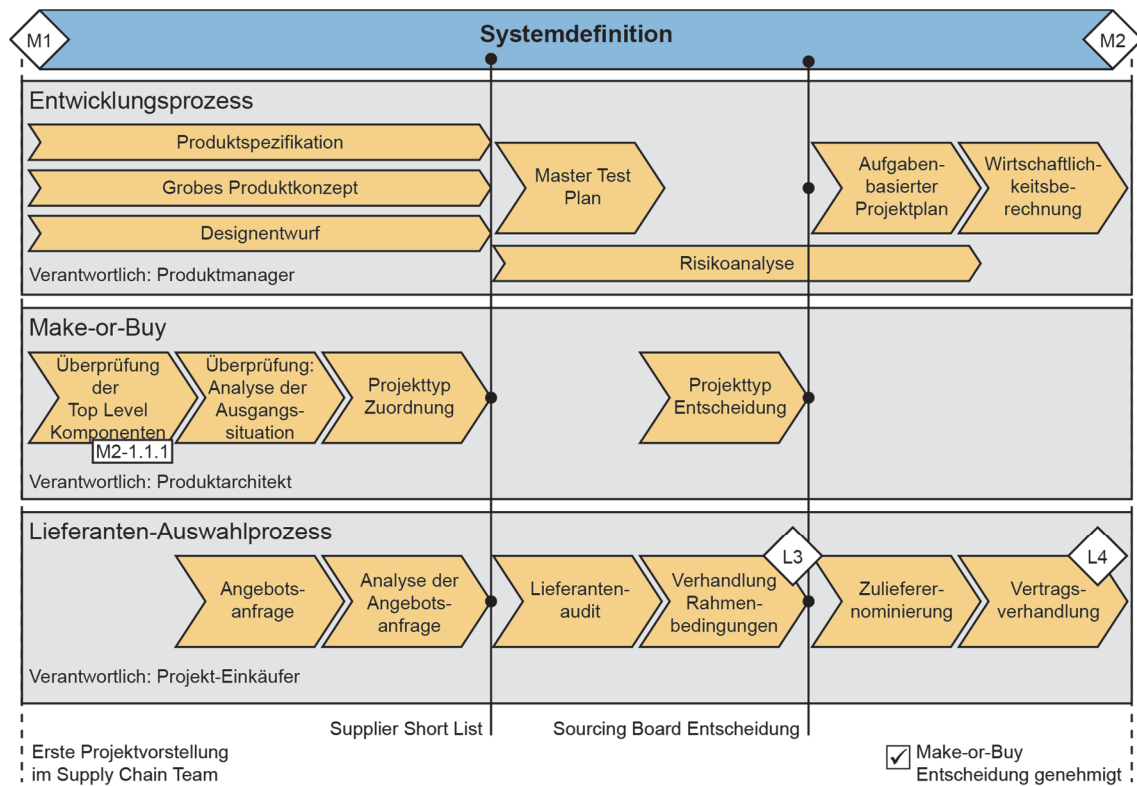


Bild 2-9: Dokumentation von Prozessen in einem Großkonzern (Ausschnitt – eigene Darstellung)

2.2.3 Organisations- und Unternehmensgestaltung

Der Organisationsbegriff unterliegt einer Vielzahl an Bedeutungen, weshalb die Organisationslehre zwischen dem institutionellen und dem instrumentellen Organisationsbegriff unterscheidet [Sch03], wenngleich sich beide bedingen: Das Unternehmen ist eine Organisation (institutionell), weil es eine Organisation hat (instrumentell) [BM14]. Im vorliegenden Kontext ist das instrumentelle Verständnis von Bedeutung; hier wird zwischen der Aufbau-, der Prozess- und der Ablauforganisation unterschieden, was auch schon durch die Darstellung in Bild 2-9 deutlich wird:

- Die Aufbauorganisation beschreibt die Gliederung einer Organisation in ein System arbeitsteiliger Organisationseinheiten und deren Beziehungen untereinander. Das organisatorische Gerüst umfasst vertikal Stellenhierarchie, Verantwortungen, Aufgaben sowie Weisungs- und Entscheidungsrechte nach den Kriterien Funktion und Objekt [Sch10].
- Die Prozessorganisation gliedert die Leistungserstellung in der horizontalen Sicht, also welche Aktivitäten wann zu tun sind. Somit zeigt sie, wie das Unternehmen anhand von Geschäftsprozessen aufgestellt ist, die durch verschiedenste mit einander verknüpfte Aktivitäten realisiert werden.
- Die Ablauforganisation bildet die Prozess- auf die Aufbauorganisation ab und regelt das räumliche, zeitliche und inhaltliche Zusammenwirken von Arbeitspersonen, Arbeits- und Betriebsmitteln, Arbeitsobjekten und dem Input des Arbeitssystems. Folglich definiert die Ablauforganisation, in welcher Reihenfolge die Aktivitäten zur Erfüllung der Arbeitsaufgaben durchgeführt werden. Ebenso werden Tätigkeitsinhalte, -umfänge und -anforderungen beschrieben [Fro04].

Technologische Neuerungen, Expansion in andere Märkte und soziale sowie politische Veränderungen führen zu einer kontinuierlichen Evolution der Organisationsformen. Erkennbar ist: Die Evolution der Organisationsformen verfolgt das übergeordnete Ziel, flexible Ressourcen zu kombinieren. Wesentlich sind dabei die Reaktionsfähigkeit, transparente Informationswege und v.a. die interne Flexibilität [Sch80]. Auf diese Weise konnte über die Zeit die Leistungsfähigkeit der Organisation kontinuierlich gesteigert werden. In der Literatur nimmt diese Forderung nach Flexibilität der Organisation eine besondere Rolle ein [TW10], [Bin11]. Eine erhöhte Flexibilität geht jedoch zu Lasten der Effizienz und der Fachkenntnisse. Die hier betrachteten Systeme erfordern einerseits Fachwissen in der Tiefe und andererseits Wissen in der Breite (also über Fachdisziplinen hinweg) – ein herausforderndes Anforderungsprofil an die Handlungspersonen, das i.d.R. nur selten erfüllt werden kann.

Aus diesem Grund arbeiten Unternehmen zunehmend mit externen Partnern in der Wertschöpfungskette und passen hierfür die Organisation an (vgl. Bild 2-10) [PBF+13]. Mehr und mehr geht es dabei um verteilte Entwicklungsnetzwerke [CR06]. Auf der einen Seite führt dies zu einer geringeren Fertigungstiefe sowie einer reduzierten Ent-

wicklungsleistung der OEM, da sie nur als sog. Integratoren, „Brand Owner“ oder – als ein extremes Beispiel – als Broker in einem Netzwerk agieren, in dem es nur einmalige oder befristete Verträge für eine konkrete Produktidee geht. Hier kennen die Broker teilweise nicht einmal mehr die Sublieferanten. Auf der anderen Seite führen diese Trends zu einem hohen Koordinationsaufwand in der Wertschöpfungskette, da die Produktentstehung nun über verschiedenen Zeitzonen mit unterschiedlichen Kulturkreisen stattfindet – gleichzeitig soll der Überblick behalten werden.

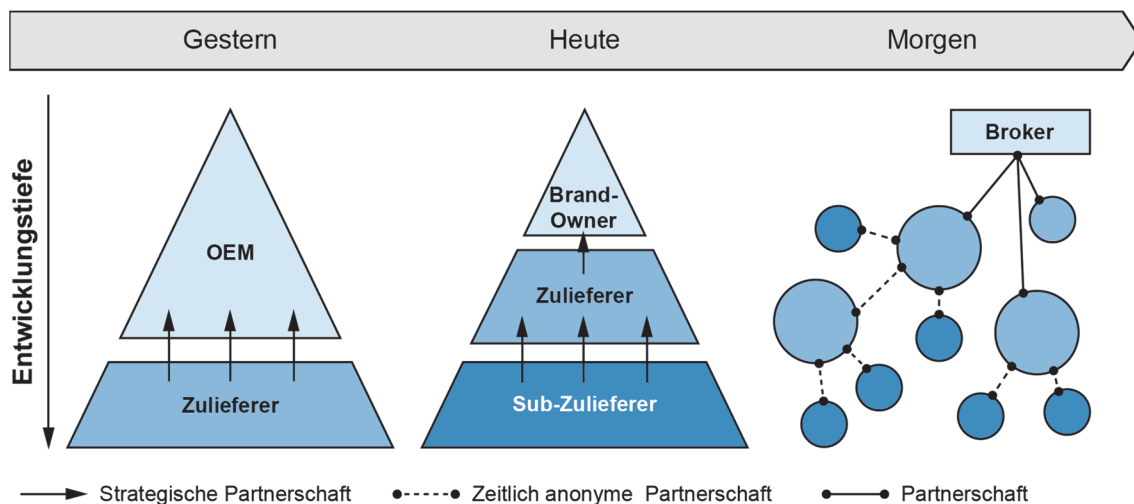


Bild 2-10: Veränderung von Wertschöpfungsstrukturen – „Organisatorischer Wandel“

2.2.4 Stakeholder, Rollen und Kompetenzen

Produktentstehung ist ein sozio-technisches System; alle Personen und Gruppen, die bei der Kreation eines Systems beteiligt sind oder berücksichtigt werden müssen, sind Stakeholder. Etymologisch betrachtet ist das der *Inhaber eines Anspruches oder Einsatzes*, der auf dem Spiel steht [Tie10]. FREEMAN nennt hier bspw. das Management, Mitarbeiter, Lieferanten, Fremdkapitalgeber, Kunden, Mitbewerber [Fre84].

Zur Ausgestaltung der Prozesse werden den Stakeholdern Rollen übertragen, die sie im Rahmen eines Projekts erfüllen müssen. Wie das genaue Zusammenspiel der einzelnen Rollen funktioniert, ist dabei nur grob skizziert – die Interaktion zwischen den Beteiligten ist jedoch ein Erfolgsfaktor für die Produktentstehung. SOSA ET AL. stellen fest, dass für den Misserfolg eines Projekts die ignorierten und die unentdeckten Schnittstellen verantwortlich sind [SER04] – hieraus folgt automatisch, dass wesentliche Abstimmungen und Tätigkeiten in einem Projekt mit großer Wahrscheinlichkeit unberücksichtigt bleiben. Um das zu verhindern wird ein Projektmanager mit der Aufgabe betraut, die Erreichung der Ziele des Projekts sicherzustellen. Der Projektmanager stellt das Projektteam zusammen, das dabei nicht nur mit Beteiligten aus der Konstruktion oder Entwicklung besetzt sein sollte [PBF+13]. Die tatsächliche Zusammensetzung ist von der Problemstellung und von der Produktart abhängig. Mit zunehmender Größe des Projekts bietet es sich an, ein kleineres Kernteam zu bilden, das verantwortliche Fachleute aus

Konstruktion, Arbeitsvorbereitung, Marketing und Vertrieb umfasst [PBF+13]. Das führt jedoch dazu, dass die Mitarbeiter auf fachlicher Ebene einem Multiprojektmanagement ausgesetzt sind und nur ein Bruchteil ihrer Arbeitszeit für ein Produkt aufwenden können. Bei gesteigener Produktkomplexität führt das zu einer überproportional gestiegen Prozess- und Organisationskomplexität.

Die notwendigen Kompetenzen für die Produktentstehung mechatronischer Systeme haben also eine enorme Spannbreite. Wesentlich sind dabei zwei Richtungen: Spezialwissen in einer Fachdisziplin bzw. in einem Anwendungsfeld – also „Wissen in der Tiefe“ und Gesamtverständnis für den Produktentstehungsprozess – „Wissen in der Breite“. Dass inzwischen das Basiswissen in jeder Fachdisziplin als mindestens genauso wichtig erachtet wird wie Spezialwissen (vgl. Bild 2-11) unterstreichen auch die Entwicklungen der Wertschöpfungsstrukturen hin zu Entwicklungsnetzwerken mit hochgradiger Aufgabenteilung. Schwierig ist: Früher wurde auf Spezialwissen einer Fachdisziplin Wert gelegt, s.d. heute weitgehend Mitarbeiter mit spezifischem Wissen, aber veraltetem Methodenwissen in den Unternehmen arbeiten – es fehlt das für komplexe technische Systeme notwendige Generalistenwissen, häufig als T-Shape bezeichnet [MSL+15].

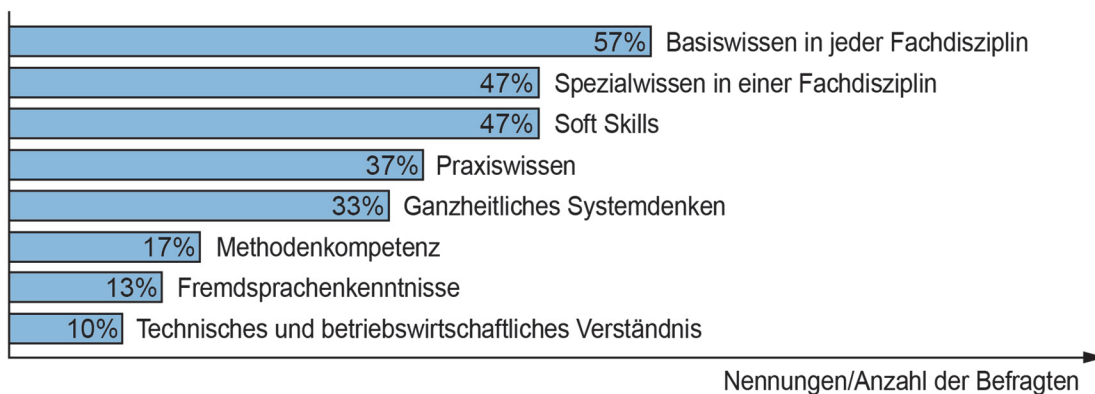


Bild 2-11: Kompetenzen der Ingenieure von morgen

2.2.5 Methodisches Arbeiten

Bislang wurde herausgearbeitet, welche Bedeutung strukturierte und detaillierte Prozesse und Methoden für die Produktentstehung und die alltägliche Arbeit in Unternehmen besitzen. Trotz eines aufkommenden Bewusstseins für methodisches Arbeiten ist dieses aber immer noch unzureichend in das Tagesgeschäft eingebunden [SL05]. Für die Anwendung von Methoden sind zielgerichtete Eingangsinformationen notwendig – vom Prinzip entspricht eine Methode einer Funktion, die einen Input in einen Output transformiert. Diese Eingangsinformationen sind im Regelfall jedoch nicht im Unternehmen verfügbar, da sie meist personengebunden sind. Falls doch, dann sind die Handlungspersonen der Produktentstehung meist mit der Suche nach diesen Informationen beschäftigt, was die Bereitschaft zum methodischen Arbeiten sinken lässt. Auch aktuelle

Studien bestätigen die Befunde der 1960er Jahre. Ingenieure sind immer noch gut die Hälfte ihrer Arbeitszeit mit der Suche nach Informationen beschäftigt – vgl. Bild 2-12.

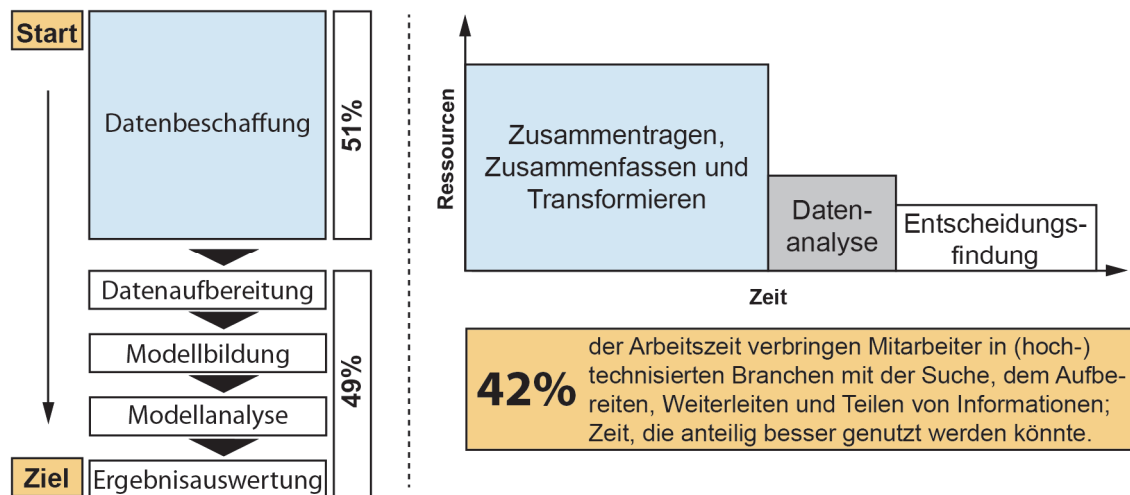


Bild 2-12 Hauptaktivität „Suchen“ bei Ingenieuren – links: [Kle03], rechts: [IDC13]

Generell ist damit weniger die aus dem Change Management bekannte Resistance-to-Change die Ursache für eine geringe Methodenakzeptanz. Stattdessen werden die meist aus der Wissenschaft hervorgehenden neuen Methoden den Anforderungen der Realität nicht gerecht [Jän07], [War13] und noch viel seltener können die Methoden aufgrund unzureichender Informationsverfügbarkeit in der Realität etabliert werden. Damit werden die generellen Bedürfnisse, Fähigkeiten, Arbeits- und Denkstile der potentiellen Anwender nicht ausreichend berücksichtigt. Ebenso werden die Methoden nicht strukturiert in die Anwendung transferiert, weshalb sie schlichtweg unbekannt sind [War13].

Die Folge: Es existiert eine Vielfalt an Methoden für die Produktentstehung [Bir11] – gleichzeitig ist die Adaption dieser Ansätze mehr als ungenügend. Das wiegt umso mehr, da für das Scheitern von Projekten meist die fehlende Methodenkenntnis der Handlungspersonen verantwortlich ist [SSG09]. Generell ist neben fehlender Kenntnis der Methoden auch die aus der notwendigen Suche resultierende fehlende Zeit zur Anwendung ein weiteres großes Hindernis [SSG09], was auf eine gewisse Komplexität der Hilfsmittel oder einer fehlende „Passfähigkeit“ in die Prozesse schließen lässt. Diesem Punkt kommt im Kontext verteilter Entwicklungsnetzwerke eine besondere Rolle zu; viele Methoden können für diese Situation nicht angemessen eingesetzt werden.

Zur Überwindung der dargestellten Hürden definieren BADKE-SCHAUB ET AL. [BDR11] drei zentrale Anforderungen, denen Methoden gerecht werden müssen:

- *Performance*: Die Leistungsfähigkeit einer Methode muss ausreichend sein und vom Anwender eingeschätzt werden können.
- *Presentation*: Die Methode ist simpel dargestellt und kann ohne intensives Vorwissen angewendet werden.

- *Process*: Die Methode muss sich nahtlos an vorliegende Bedingungen anpassen.

Dieses Konzept wird von ALBERS [ALR12] aufgegriffen. Er ergänzt es um die aus dem Change Management bekannten Blickwinkel der individuellen und organisatorischen Akzeptanz (vgl. Bild 2-13).

	Performance	Presentation	Process
Individuelle Akzeptanz	Wahrgenommene Leistungsfähigkeit	Intuitive Handhabbarkeit	Flexibilität und Anpassbarkeit
Organisatorische Akzeptanz	Aufwand-Nutzen-Verhältnis	Lehr- und Lernbarkeit	Wiederverwendbar- und Erweiterbarkeit

Bild 2-13: Das 3P-Konzept ergänzt um Aspekte nach [ALR12]

Neben diesen Herausforderungen der Methodengestaltung ist zusätzlich zu bedenken, dass die Diffusion neuer Arbeitsweisen sehr lange Zeit benötigt [War13]. So hat bspw. die vollständige Durchdringung der Produktentstehung mit Computer-Aided Design (CAD) gut 40 Jahre benötigt [Sto12a]. Noch zu Beginn der 1990er Jahre ist die Anwendung von CAD in der Breite des Maschinenbaus unterdurchschnittlich gewesen; die Einführung wurde zudem nicht strukturiert angegangen, mit zahlreichen Herausforderungen in den Dimensionen Mensch – Organisation – Technik [oV91]. Insbesondere waren die Technik und die Methodik noch nicht ausgereift, ebenso konnte aufgrund schlechter Einführungsplanung der Nutzen von CAD nicht bestimmt werden.

Damit steht generell die Frage nach dem Wertbeitrag methodischen Arbeitens im Raum: Empfundener Nutzen durch veränderte Arbeitsweise und empfundener Aufwand für die Einführung und Nutzung müssen im Gleichgewicht stehen [Sau12]. Methodisches Arbeiten bedeutet aber Overhead, wie bspw. Stabstellen, der durch Umlagen innerhalb von Unternehmen querfinanziert wird. Dies ist häufig nur in großen Unternehmen möglich, die sich für diese Aufgabe eigene Fachabteilungen einrichten. Gerade für kleine und mittelgroße Unternehmen ist das nur schwierig möglich, weshalb hier methodisches Arbeiten eine noch geringere Verbreitung hat. Grundsätzlich sollte jedoch von einem positiven Wertbeitrag methodischen Arbeitens ausgegangen werden: In der ehemaligen DDR entstanden in den 1950er Jahren motiviert durch den Mangel an Fachkräften die ersten methodischen Ansätze der Konstruktionslehre, die durch HANSEN, BISCHOFF UND BOCK als Vertreter der Ilmenauer Schule geprägt sind [Han56], [Han65]. Ziel war eine Steigerung der Effizienz bei der Entwickeln durch methodisches Arbeiten unter Berücksichtigung denkpsychologischer und heuristischer Ansätze zur Unterstützung einer besseren Arbeit der Konstrukteure [Mül90].

2.2.6 Herausforderung Produktentstehung

Neben den in Kapitel 2.1 beschriebenen technologischen Herausforderungen in der Produktentstehung ist erkennbar, dass die Produktentstehung als sozio-technisches System generell vor immensen Veränderungen steht – teilweise auch als Entgrenzung der Produktentstehung bezeichnet [SD16]. Durch diese Entgrenzung wird die Produktentwicklung zu einer gemeinschaftlichen und multidisziplinären Aufgabe [AES+12]. Diese Veränderungen führen dazu, dass Unternehmen die Produktentstehung und ihre Prozesse stärker methodisch organisieren; generell herrscht jedoch eine geringe Methodendurchdringung in Unternehmen vor – bei gleichzeitig großer Anzahl an Methoden. Gerade bei großen Unternehmen ist aber ein Trend zu eigenen Methodenabteilungen erkennbar, die hier Abhilfe schaffen sollen. Wie sehr Prozesse zukünftig strukturiert werden und wie methodisches Arbeiten die Unternehmen durchdringen soll, zeigt das folgende Praxisbeispiel: Projektarchetypen helfen, die Wertschöpfungsstruktur zu systematisieren (Bild 2-14) [TKD+14]. In Abhängigkeit der gesetzten Produktstrategie (z.B. Innovationsführer vs. Preisführer) werden Prozesse und Handlungsanweisungen für die Arbeit detailliert beschrieben, Entscheidungskriterien für die Auswahl eines Archetypen auf Basis eines Produktkonzepts definiert und so schon früh die Wertschöpfungskette organisiert. In Abhängigkeit der Kriterien wird entschieden, ob eine Komponente intern oder extern entwickelt wird und wie das Projekt zu managen ist.

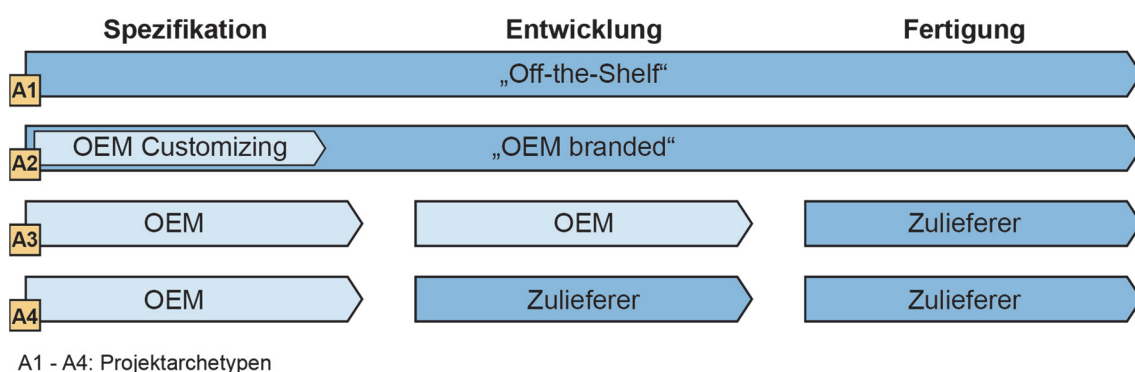


Bild 2-14: Archetypes zur Systematisierung der Wertschöpfung, nach [TKD+14]

Durch diese zahlreichen organisatorischen Maßnahmen wird gleichzeitig aber eine weitere Stufe der Komplexität erreicht – mit der Folge einer Komplexitätsspirale [LMB09]: Sämtliche Arbeitsweisen und Prozesse müssen angepasst und stringent gelebt werden. Vor dem Hintergrund einer stark verteilten Entwicklung – bspw. in den dargestellten Broker-Netzwerken – kommt zudem der Virtualisierung der Produktentstehungsprozesse eine besondere Rolle zu. Generell ist also die Verfügbarkeit von Daten Erfolgsfaktor für methodisches Arbeiten, die aufgrund der hohen Anteile von Suchaktivitäten im Arbeitsalltag nur bedingt gegeben sein kann – insb. die sehr frühen Phasen von Projekten und die entwicklungsbegleitenden Aktivitäten sind hier nicht unterstützt.

2.3 Systems Engineering

Die zahlreichen Veränderungen in der Produktentstehung und das zunehmende Bewusstsein für methodisches Arbeiten haben zu einer Suche nach unterstützenden entwicklungsmethodischen Ansätzen geführt. Somit ist es nicht verwunderlich, dass das Systems Engineering (SE) als fachdisziplinübergreifender Ansatz eine Art Renaissance erlebt. Die wohlstrukturierten Prozesse des Systems Engineerings können vor den dargestellten Herausforderungen in besonderer Weise ihre Wirkung entfalten.

2.3.1 Entwicklung des Systems Engineerings

Systems Engineering hat sich im letzten Jahrhundert als zentraler Ansatz der Produktentwicklung in der Luft- und Raumfahrt und der Verteidigungsindustrie etabliert – vornehmlich in den USA. Grundsätzlich ist das Verständnis von Systems Engineering sehr facettenreich, was sich in unterschiedlichen Ausprägungen und einer relativ großen Anzahl von Definitionen äußert. Eine der jüngsten Definitionen stammt aus dem SEBoK (Systems Engineering Body of Knowledge) – dem kleinsten gemeinsamen Nenner der Fachwelt. Sie zeigt auch die Breite des Themenfelds. Systems Engineering ist

“[...] an interdisciplinary approach and means to enable the realization of successful systems. It focuses on holistically and concurrently understanding stakeholder needs; exploring opportunities; documenting requirements; and synthesizing, verifying, validating, and evolving solutions while considering the complete problem, from system concept exploration through system disposal. [SEBoK15]

Im Wesentlichen lassen sich in Anlehnung an [GDS+13b] im englischsprachigen Raum zwei Entwicklungslinien für das Systems Engineering skizzieren – das praxisgetriebene SE und die wissenschaftlichen Konzepte. Hierbei fällt auf: Gerade die eher pragmatischen, praxisgetriebenen Ansätze prägen das Bild von SE noch heute, jedoch in einer beinahe nicht zu überblickenden Vielfalt [Kas10], [Hon05]. In vielen Handbüchern sind Good Practices dokumentiert, welche die Erfahrungen der Anwender widerspiegeln und Nutzen in den klassischen Anwendungsbereichen Militärwesen und der Luft- und Raumfahrt generieren können [Hon13] [Hon05].

Im deutschsprachigen Raum ist die Entwicklung des SE anders: Hier haben sich zunächst spezialisierte Ansätze durchgesetzt, die auf die Entwicklung innerhalb einer spezifischen Fachdisziplin ausgerichtet sind und hierfür dedizierte Vorgehensmodelle und Methoden bereitstellen, wie z.B. bei ROTH [Rot82] und KOLLER [Kol85]. Erst seit Veröffentlichung der VDI-Richtlinie 2206 [VDI2206] setzt ein Umdenken ein, das die stark durch die Systemtechnik geprägten frühen Arbeiten der Konstruktionsmethodik wieder stärker ins Bewusstsein rief, wie bspw. BEITZ [Bei70] und ROPOHL [Rop75].

Aktuell versucht das International Council on Systems Engineering (INCOSE) sämtliche Entwicklungsstränge zusammenzuführen und das SE zu vereinheitlichen. Ziel ist,

SE als generellen branchenunabhängigen Ansatz zu etablieren, wie z.B. in der SE-Vision 2025 aufgezeigt wird [INC15]. SE erhebt den Anspruch, die Akteure in der Entwicklung komplexer Systeme zu orchestrieren und eine wirtschaftliche Produktentstehung sicherzustellen. SE bündelt damit die Herausforderungen der Produktentstehung und begegnet ihnen auf Basis einer ganzheitliche Betrachtung des Systems aus Produkt und dazugehöriger Projekt-, Prozess- und Organisationsgestaltung(vgl. Bild 2-15).

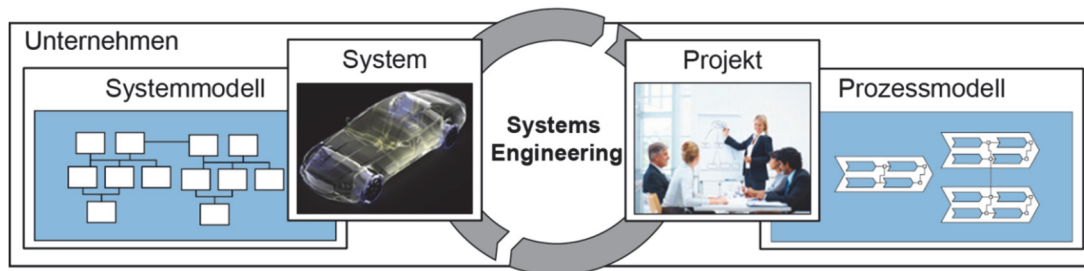


Bild 2-15: Das SE-Konzept nach [GDS+13b], erweitert um den Aspekt Unternehmen

2.3.2 Wissenschaftliche SE-Konzepte

Als Basis sämtlicher wissenschaftlicher Arbeiten im SE gilt die Allgemeine Systemtheorie des Biologen BERTALANFFY [Ber32]. Sie ist nicht auf eine bestimmte Disziplin begrenzt, sondern hat einen fachdisziplinübergreifenden Charakter und die Fähigkeit, andere Disziplinen zu integrieren. Nach PULM erfährt die sie dennoch immer eine fachdisziplinspezifische Ausprägung [Pul04] – im Ingenieurwesen dem Systems Engineering. Heute prägen drei Konzepte das Geschehen im Themenfeld. Sie bilden die Basis von Forschungsarbeiten wie bspw. dem iPEM von ALBERS [AB11], werden aber auch von der Industrie als nutzenstiftend geschätzt.

HITCHINS überträgt das Systemdenken auf die Produktentstehung mit einem 5-Ebenen-Modell [Hit07]. Jede Ebene ist integraler Bestandteil der übergeordneten Ebene und muss bei der Entwicklung berücksichtigt werden (vgl. Bild 2-16).

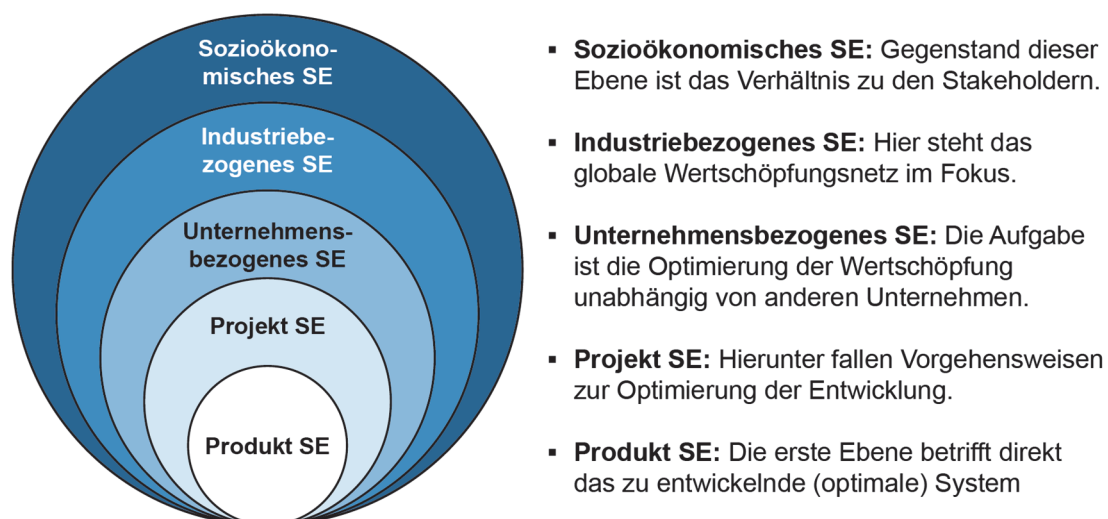


Bild 2-16: 5-Ebenen-Modell nach HITCHINS [Hit07]

Diesem abstrakten Konzept steht der Ansatz von HABERFELLNER gegenüber [HWF+12]. Schon in der ursprünglichen Version von 1976 wird das „SE-Männchen“ diskutiert (Bild 2-17): Im Kopf sind die normativen Aspekte des SE verankert: Die SE-Philosophie gliedert sich in das Systemdenken und ein Vorgehensmodell. Ziel des Systemdenkens ist die Modellierung des Systems unter der jeweils relevanten Perspektive, bspw. die umgebungs-, wirkungs- und strukturorientierte Sichtweise. Damit greift es einen zentralen Aspekt der Modelltheorie auf – die Zweckorientierung. Das Vorgehensmodell nimmt die Zweckmäßigkeit der folgenden Grundgedanken an:

- Top-Down: Vom Groben zum Detail
- Denken in Varianten: Konsequente Suche nach Alternativen
- Phasenablauf: Gliederung des Prozesses nach zeitlichen Gesichtspunkten
- Problemlösungszyklus: Unabhängig von der Phase der Systementstehung werden auftretende Probleme mit demselben Vorgehensleitfaden gelöst

Die Füße geben dem Männchen festen Stand: Hier sind die Methoden und Techniken der Systemgestaltung und des Projektmanagements verankert. Besonders unterstrichen wird der Zusammenhang zwischen Technik und Projektmanagement: Eine erfolgreiche Systemgestaltung ist nur durch ein geeignetes Projekt- und Entwicklungsmanagement möglich – deshalb sind beide Bereiche eng miteinander als Teil des ganzheitlichen Ansatzes des Systems Engineering verbunden; das bringt bereit Bild 2-15 zum Ausdruck. Dadurch ergeben sich automatisch auch Überlappungen zwischen SE und Projektmanagement. Durch das Zusammenspiel von Kopf und Füßen wird der Problemlösungsprozess aber strukturiert durchlaufen; hier wird ein Problem in eine Lösung überführt.

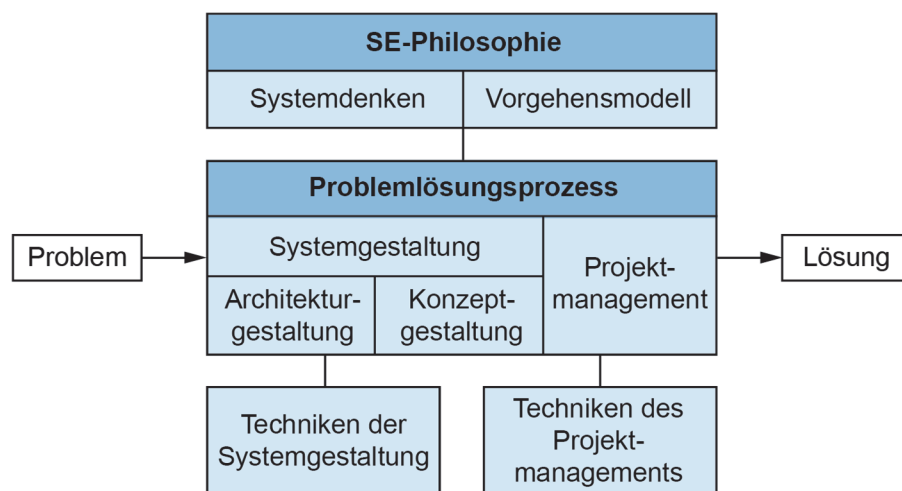


Bild 2-17: Das SE-Konzept nach HABERFELLNER

Genau diese Überführung eines Problems in eine Lösung lässt sich nach ROPOHL systemtheoretisch beschreiben [Rop75], [Rop09]. Hierfür überführt er den Ansatz der Allgemeinen Systemtheorie in ein Systemmodell der Technik und definiert den Zusammen-

menhang von Ziel-, Handlungs- und Sachsystem (vgl. Bild 2-18): Aus der Ingenieur-tätigkeit gehen Sachsysteme hervor. Die hierfür notwendigen Aktivitäten und Einrichtungen der technischen Arbeit und der Organisation ordnet ROPOHL dem Handlungssystem zu. Die Zielvorgaben der Ingenieur-tätigkeit stammen zum Teil aus dem Handlungssystem selbst, insb. aber aus dessen Umgebung – die Menge dieser Zielvorgaben bilden das Zielsystem. Generell beeinflusst das Handlungssystem das Zielsystem, während das Sachsystem auf das Handlungssystem zurückwirkt oder gar dessen Bestandteil wird.

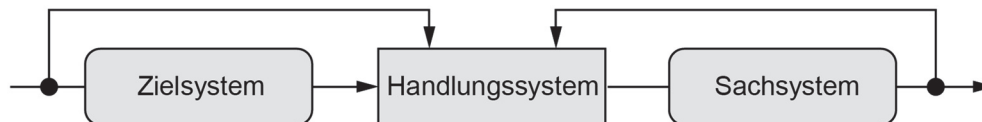


Bild 2-18: Regelkreis der Ingenieur-tätigkeit, nach ROPOHL, [Rop75]

Die drei Ansätze beschreiben zusammengenommen kompakt die wesentlichen Einfluss- und Tätigkeitsbereiche und Zusammenhänge im Systems Engineering und lassen Modell- und Systemdenken als seine zentralen Grundlagen deutlich werden. Letztlich sind sie aber nur Konzepte, die in dieser Weise nicht direkt operationalisierbar sind.

2.3.3 Themenfelder und Standards des Systems Engineerings

Die praxisgetriebenen Entwicklungslinien und SE-Standards haben zu unterschiedlichen Schwerpunkten und Inhalten geführt. Die Themenfelder der zahlreichen Normen, Standards, Richtlinien und Handbücher hat HONOUR in acht Themenfelder überführt [Hon13]. Zu Beginn eines Entwicklungsprojektes sieht HONOUR die *Mission/Purpose Definition*. Dies umfasst im Wesentlichen das Planen und Klären der Aufgabe. Es folgt das *Requirements Engineering*, d.h. die Definition, Analyse und das Management von Anforderungen. Als drittes Themenfeld wird das *Systems Architecting* genannt. Darunter wird die Definition des Systems hinsichtlich seiner Komponenten und deren Beziehungen zueinander verstanden. Dies erfolgt zeitlich vor der fachdisziplinspezifischen Entwicklung und umfasst die Entwicklung und Bewertung von Alternativen. Es folgt die *System Integration*, d.h. die Erstellung des Systems aus den entwickelten oder eingekauften Systemkomponenten. Das Themenfeld *Verification and Validation* bezieht sich auf den Abgleich des Systems mit dessen Anforderungen (Verifikation) bzw. dessen Zweck (Validierung). Die *Technical Analysis* bezieht sich auf die Bewertung der technischen Systemperformance hinsichtlich der Anforderungen. Im Zuge des *Scope Managements* wird die Wertschöpfungskette betrachtet. Das achte Themenfeld *Technical Management* umfasst Aufgaben wie Projektbeurteilung und -steuerung, Entscheidungs-, Risiko-, Konfigurationsmanagement und die Qualitätssicherung. Weiter werden technische Maßnahmen zur Kompensation von Defiziten identifiziert und durchgeführt, bspw. hinsichtlich Zeit sowie Kosten und Qualität der Ergebnisse.

Seit einiger Zeit existieren starke Aktivitäten, die Landschaft der Normen, Standards und Richtlinien im Systems Engineering zu konsolidieren. Diese sind getrieben u.a. durch das Deutsche Institut für Normung e. V. (DIN), die ISO, das IEEE – aber natürlich auch durch

INCOSE. Dabei rücken immer stärker die Bedürfnisse von Unternehmen außerhalb der klassischen Anwendungsbereiche von Systems Engineering in den Mittelpunkt. Maßgeblich sind momentan die internationalen Standards, die sich der Begriffswelt, der Strukturierung des Lebenszyklus und der Anwendbarkeit von SE in Prozessen annehmen:

- Die ISO/IEC 15288: Systems and software engineering – System life cycle processes beschreibt die Prozesse über den Lebenszyklus eines technischen Systems. Es werden vier Prozess-Gruppen inkl. entsprechender Terminologie definiert [ISO15288]⁴. Für jede Gruppe werden die für Systems Engineers relevanten Prozesse detailliert. Je nach Komplexität oder abhängig von weiteren Randbedingungen wird der Aufwand für die Prozesse verändert, was einem Tailoring entspricht. Die Prozess-Gruppen gehen dabei weit über die Modellierung der Systemarchitektur hinaus, bspw. in sog. *Technical Management Processes*.
- Die ISO 42010 Systems and Software Engineering – Architecture Description definiert die Begriffe der Architekturbeschreibung [ISO42010]. Hier werden die wesentlichen Zusammenhänge von Architektur und Anwendung der Architektur beschrieben. Vorgänger des ISO-Standards ist die IEEE1471 zur Architekturbeschreibung von Softwaresystemen mit dem Titel IEEE Recommended Practice for Architectural Description of Software-Intensive Systems [IEEE1471].
- Die ISO/IEC 29110 „Systems and Software Life Cycle Profiles and Guidelines for Very Small Entities (VSEs)“ ist die „kleine Schwester“ der ISO/IEC 15288 [ISO29110]. Sie ist bei Organisationen, Abteilungen oder Projekten bis etwa 25 Projektteilnehmern geeignet und besteht aus 5 Teilen, wobei nur der Teil 4 als „Standard“ eingestuft ist. Basis sind „Generische Profile“ mit sog. „Deployment Packages“. d.h. ausführlichen Anleitungen und Checklisten, welche die schrittweise Einführung in die SE-Prozesse in Abhängigkeit der Organisation, der Größe und der Komplexität des Projekts unterstützen [TA15a].

Die Standards bestätigen die wissenschaftlichen Darstellungen, dass es neben den rein technisch getriebenen Aufgaben im Systems Engineering viele Tätigkeiten mit großer Nähe und Überlappung zum Projektmanagement gibt. Im Kern werden diese Prozesse durch ein Set an gut gelenkten Dokumenten befähigt, die alle wesentlichen Informationen über das Produkt und das Projekt enthalten. Diese dokumentenbasierte Entwicklung führt jedoch zu großen Herausforderungen aufgrund der Intransparenz mechatronischer Systeme, die darüber hinaus in verteilten Entwicklungsnetzwerken entstehen.

⁴ Anhang A1.1 gibt einen Überblick über die Themenbereiche der ISO/IEC 15288:2015.

2.3.4 Der Systems Engineer

Der Systems Engineer orchestriert die Produktentstehung – das ist die Vision der wenigen Beschreibungen des Berufsbilds [She96], [SSG14], [PDH11]. Das Aufgabenprofil im Detail ist jedoch bislang sehr heterogen und umstritten. Für den Beruf des Systems Engineer wird eine große Anzahl an Zuständigkeitsbereichen und Rollenbeschreibungen im Produktlebenszyklus diskutiert. Eine sehr allgemeine Sicht auf den Systems Engineer wird in Bild 2-19 zusammengefasst.

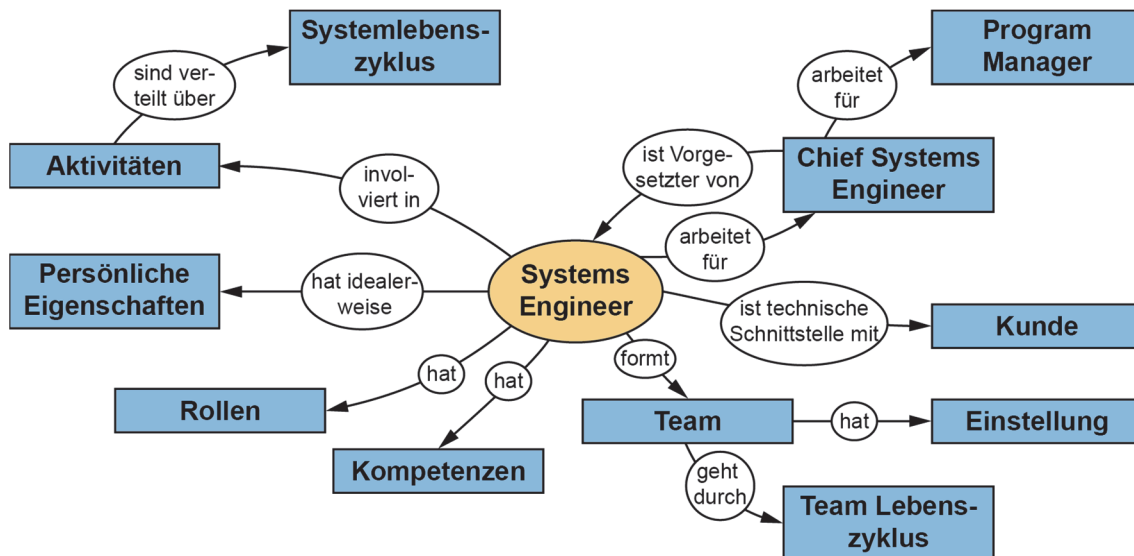


Bild 2-19: Der Systems Engineer nach [PDH11]

SHEARD fasst die verschiedenen Ansichten zum Tätigkeitsfeld des Systems Engineer in insgesamt *zwölf Rollen* zusammen [She96]. Einerseits charakterisieren diese Rollen alle Eigenschaften, Fähigkeiten und Kenntnisse, die ein Systems Engineer besitzen muss. Andererseits geben sie einen guten Eindruck davon, wie Systems Engineering verstanden werden kann und welche Kernfunktionen darin ausgeübt werden. Diese Rollen lassen sich in zwei Kategorien aufteilen: Einerseits existieren sogenannte *Life-Cycle Roles*. Aus dieser Perspektive kann Systems Engineering als eine Abfolge spezifischer Aufgaben entlang des Produktlebenszyklus angesehen werden. Hierzu zählen die Rollen: Requirements Owner, System Designer, Validation and Verification Engineer, Logistics and Operations Engineer und System Analyst. Andererseits kann Systems Engineering auch als das Management, die Koordination und Nachhaltung der fachspezifischen Systementwicklung mit Methoden der Systemtechnik verstanden werden. Diese Sichtweise wird von den *Program Management Rollen* repräsentiert. Darunter werden die Rollen: Technical Manager, Glue, Information Manager, Coordinator und Customer Interface gefasst [She96]. Je nach Unternehmensgröße sind diese Rollen durch eine oder mehrere Personen vertreten. MOEHRINGER definiert bspw. für einen Mittelständler die Verteilung der Rollen auf existierende Stellenprofile ohne dafür das Profil des Systems Engineers zu schaffen [Möh12].

2.3.5 Herausforderung Systems Engineering

Systems Engineering schickt sich an, die Vorgehensweisen in der Entwicklung technischer Systeme auf eine neue Basis zu stellen. Kern ist eine ganzheitliche Sicht auf die Aufgabe, die weit über die Grenzen des technischen Systems hinausgeht und hierzu v.a. auch stark Aspekte des Projekt- und Entwicklungsmanagements einbezieht: Systemdenken. Ein Systems Engineer übernimmt als zentrale Schaltstelle alle wesentlichen Aufgaben rund um das System – im Prinzip als „Dirigent“ der Produktentstehung und bindet die zahlreichen weiteren Stakeholder eines Projekts zielgerichtet ein. Das Verständnis über seine Aufgaben und Zuständigkeiten ist allerdings nicht einheitlich, viele Normen und praxisgetriebene Arbeiten erlauben kein einheitliches Bild. Deutlich wird aber, dass auch zahlreiche Aufgabengebiete enthalten sind, die sich mit dem Management von Projekten und der Steuerung von Prozessen jenseits reiner technischer Angelegenheiten befassen. Allerdings sind die etablierten Herangehensweisen des sog. dokumentenzentrierten Systems Engineering für die Entwicklung mechatronischer Systeme nur begrenzt geeignet.

2.4 Model-Based Systems Engineering

Wie im vorangegangenen Kapitel dargestellt, basiert Systems Engineering auf gut gelenkten Dokumenten – was jedoch an die Grenzen der Leistungsfähigkeit in den heutigen Entwicklungsprojekten und -organisationen stößt. Die voranschreitende Entwicklung der Informations- und Kommunikationstechnik hat in Verbindung mit den für die verteilte globale Wertschöpfung nicht geeigneten Ansätzen die Idee eines modellbasierten Systems Engineerings hervorgebracht (vgl. Bild 2-20): Model-Based Systems Engineering (MBSE). Anders als in der weitgehend etablierten modellbasierten Entwicklung – repräsentiert bspw. durch CAD, FEM und auch durchaus Ansätzen wie Modelica – stehen im MBSE nicht fachdisziplinspezifische Artefakte im Mittelpunkt, sondern eine fachdisziplinübergreifende Beschreibung des Systems: das Systemmodell. Je nach Ausgestaltung soll dies von allen Disziplinen für unterschiedliche Zwecke genutzt werden. Im Folgenden wird das Handlungsfeld MBSE erarbeitet.

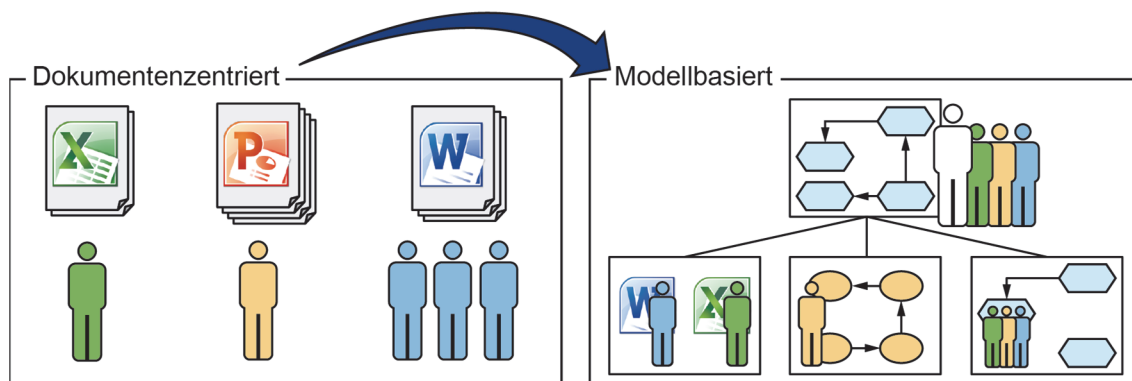


Bild 2-20: Vom dokumentenzentrierten Vorgehen zum MBSE nach [INC07]

2.4.1 Vision des Model-Based Systems Engineerings

Model-Based Systems Engineering liefert Methoden, mit denen Systemdenken wirksam unterstützt werden kann und auch abstraktes Wissen gesichert werden kann. Mit dem Systemmodell soll von Beginn des Lebenszyklus eine fachdisziplinübergreifende Systemspezifikation konsistent beschrieben und genutzt werden. Laut INCOSE ist MBSE

“the formalized application of modeling to support system requirements, design, analysis, verification, and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases” [INC07]

Damit ist der Anwendungszweck des MBSE sehr breit gefasst. Zunächst kann es dazu dienen, das Planen und Klären der Aufgabe unterstützen. Nach obiger Definition müsste es sich im Verlauf eines Lebenszyklus jedoch auch weiterentwickeln – je nach verfolgtem Zweck. Die möglichen Anwendungszwecke sind demnach mannigfaltig und werfen die Frage nach einer konkreten Ausgestaltung des MBSE auf. Nach CLOUTIER ermöglicht Modellierung als Grundlage des MBSE ein besseres Problemverständnis, das Erkennen und Beherrschen von Komplexität sowie eine verbesserte Kommunikation zwischen allen Beteiligten eines Projekts [Clo09], [Clo13]. Mit diesem Verständnis wird deutlich: MBSE ersetzt nicht die Modelle der Fachdisziplinen, sondern ergänzt sie. Gleichzeitig ist über die rein technische Sicht auch eine sozio-technische Komponente des MBSE erkennbar.

INCOSE ist überzeugt, dass sich MBSE als zentrales Paradigma der Systementwicklung etabliert wird [INC07], [INC15]. Dabei bleibt INCOSE in seiner SE-Vision 2025 jedoch sehr vage, was die konkrete Ausgestaltung des Ansatzes, die Arbeit mit dem Ansatz im Unternehmensalltag und weitere mögliche faszinierende, über die bloße Spezifikation hinausgehende, Anwendungen anbelangt [INC15]. In weiteren Veröffentlichungen wird ansatzweise dargestellt, dass der Weg von einem „ad-hoc MBSE“ Ansatz im Jahr 2007 hin zu einem institutionalisierten MBSE verfolgt wird – idealerweise im vollumfänglichen Sinne der Virtualisierung der Produktentstehung. Wie detailliert und in welcher Form dies geschieht, ist nicht festgelegt. Versteht man MBSE als logische Weiterentwicklung des klassischen Systems Engineerings müsste es im Einklang mit den Normen wie z.B. der ISO/IEC15288 auch Grundlage für die zahlreichen technischen oder techniknahen Managementaktivitäten im Lebenszyklus sein und damit viel stärker an den Entwicklungsprozess gekoppelt sein. Diese fehlende Genauigkeit in der Vorstellung von einem institutionalisierten MBSE führt zu starken Fronten in den Reihen der Systems Engineers und weiteren Stakeholdern des Themenfelds. INCOSE räumt ein, dass MBSE sich auf einem Leistungsstand wie die CaX-Bewegung in den 1950er Jahren befindet [Sto12a], [Bri98]. Aus diesem Grund fordern erfahrene Systems Engineers für das MBSE eine Besinnung auf den Kern des SE – also Ansätze, die direkten Nutzen für die Produktentstehungsarbeit bieten.

Unabhängig von der konkreten Ausprägung des MBSE herrscht aber Einigkeit, dass zur Erstellung des Systemmodells eine Beschreibungssprache, eine Methode und ein Werkzeug notwendig sind (vgl. Bild 2-21) und die Modelltheorie Grundlage des MBSE ist.

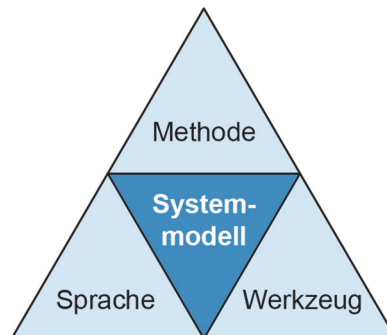


Bild 2-21: Erstellung des Systemmodells auf Basis einer geeigneten Kombination von Sprache, Methode und Werkzeug

2.4.2 Allgemeine Modelltheorie als Basis für MBSE

Trotz unterschiedlicher Meinungen zum MBSE – es herrscht Einigkeit, dass sein Ursprung in der Allgemeinen Modelltheorie liegt. Hiernach zeichnet sich ein Modell durch drei Hauptmerkmale aus [Sta73]:

- *Abbildungsmerkmal*: Modelle bilden reale Systeme mit dessen Eigenschaften ab.
- *Verkürzungsmerkmal*: Es werden nicht alle Eigenschaften des Systems erfasst, sondern nur diejenigen, die für die Stakeholder zur Erfüllung des Modellzwecks relevant sind.
- *Pragmatisches Merkmal*: Modelle erfüllen eine Ersetzungsfunktion für bestimmte Subjekte. Entsprechend der Funktion des Adressaten sollte das Modell möglichst einfach sein, ohne notwendige Informationen zu vernachlässigen.

Der Modellbegriff hat sich allerdings in den verschiedenen Ingenieurwissenschaften höchst heterogen entwickelt. In der Entwicklung mechatronischer Systeme treffen nun Fachdisziplinen mit einem unterschiedlichen Modellverständnis aufeinander. Selbst innerhalb einer Fachdisziplin existieren unterschiedliche Auslegungen – zwischen den Disziplinen sind die Unterschiede noch gravierender: Jedes der in Bild 2-22 dargestellten „Modelle“ wird vom Standpunkt des Erstellers berechtigterweise als Modell bezeichnet. Sie entsprechen im Kern der o.g. Definition nach Stachowiak und stehen somit auch nicht im Widerspruch zu dem Grundprinzip „Ganzheitlichkeit“ des Systems Engineering: Jedes der Modelle erfüllt einen Zweck. Somit können auch einzelne einfache Modelle mit Bezug auf das pragmatische Merkmal in ihrer Gesamtheit dem Anspruch der Ganzheitlichkeit gerecht werden.

Das zeigt: Der Grad der Formalisierung eines Modells und der Darstellung ist weitgehend der Grund für unterschiedliche Verständnisse. Generell muss zwischen informa-

len, semi-formalen und formalen Modellen unterschieden werden [Abu12]. Innerhalb dieser Kategorien selbst wird wiederum detaillierter unterschieden, z.B. in der semi-formalen Prozessmodellierung zwischen domänenneutralen und konzeptuellen Sprachen. Hieraus resultierende Diskussionen haben zur Folge, dass gelegentlich die Grundlagen der Allgemeinen Modelltheorie in den Hintergrund rücken und Disziplinen ihr Modellverständnis ohne Darlegung des Zwecks kommunizieren.

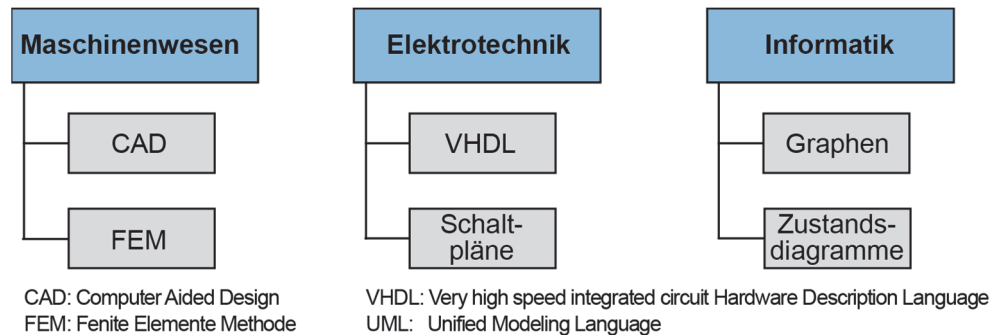


Bild 2-22: Unterschiedliches Verständnis des Begriffs "Modell", eigene Darstellung

Im Bereich des MBSE wird häufig erst dann von einem Modell gesprochen, wenn dieses eine gewisse formale Form besitzt und rechnerunterstützt abgebildet wird [Alt12]. Dieses Verständnis ist vor dem Hintergrund der virtualisierten Produktentstehung und der Langzeitvision des MBSE nachvollziehbar. Allerdings ist diese softwaretechnische Sicht schwierig mit dem Kern der Allgemeinen Modelltheorie zu vereinen und auch mit dem klassischen Maschinenbau und dem Leitbild des Systems Engineers nicht im Einklang: Nutzenpotentiale für den Maschinenbau, wie z.B. eine verbesserte Kommunikation, kann schon durch eine einfache Zeichnung erreicht werden. Andernfalls würde der Systems Engineer in einer solchen Produktentstehung mit seiner Fähigkeit die einzelnen Disziplinen zu verstehen und zu vermitteln zum bloßen „Modellverwalter“ verkommen.

2.4.3 Das Systemmodell

Obwohl das Systemmodell im Mittelpunkt des MBSE steht, wird der Begriff fast nirgendwo definiert – selbst in der SysML Spezifikation V.1.3 wird er nur drei Mal vage verwendet [OMG12]. Eine der wenigen bekannten Definitionen liefert FRIEDENTHAL:

„The system model is generally created using a modeling tool and contained in a model repository. The system model includes system specification, design, analysis, and verification information“ [FMS11].

Hier wird deutlich: Ein Datenmodell ist nützlich, jedoch nicht zwingend (vgl. idiomatische Bedeutung von „generally“). Selbstredend werden die vollen Potentiale des MBSE – bspw. eine Integration des Systemmodells in PLM-Lösungen [Sen13] – erst durch eine derartige Ausgestaltung möglich. FRIEDENTHAL weist jedoch darauf hin, dass auch eine „hybride“ Modellierung großen Nutzen gegenüber der bisherigen Praxis bedeutet

und auf lange Zeit Standard in Unternehmen sein wird [FMS08], [FMS11]. Mit hybrid ist die grafische Modellierung mit den Ansätzen des MBSE gemeint – jedoch ohne Datenmodell und als Ergänzung zu einer textuellen Spezifikation.

Trotz Unklarheiten hinsichtlich des Begriffs – über die Inhalte des Systemmodells herrscht weitgehend Einigkeit. Als Systemmodellinhalte definiert bspw. ALT die drei Bausteine [Alt12] Systemanforderungen, -architektur und -verhalten:

- *Systemanforderungen*: Es können funktionale und nichtfunktionale Anforderungen unterschieden werden. Funktionale Anforderungen spezifizieren das Verhalten des Systems und dessen Komponenten, nichtfunktionale Anforderungen definieren Eigenschaften eines Systems wie Zuverlässigkeit, Benutzbarkeit oder Änderbarkeit.
- *Systemarchitektur*: Die Architektur beschreibt die Struktur des Systems und seiner Komponenten, ebenso die Schnittstellen zwischen den Komponenten und zu den Systemgrenzen. Für den Bereich des Maschinenbaus umfasst das häufig auch erste Gestaltinformationen, z.B. in Form von Hüllkurven oder Skizzen.
- *Systemverhalten*: Durch die Architektur und die funktionalen Anforderungen ist das System beschrieben. Anforderungen sind jedoch häufig informale Textdokumente. Durch eine Formalisierung könnten Informationen rechnergestützt abgeleitet werden. Dies ist Aufgabe der Modellierung des Systemverhaltens.

Inhaltlich sollte das Systemmodell mindestens die fachdisziplinübergreifend relevanten Informationen der Entwicklungsaufgabe enthalten [IKD+13]. Genau genommen müssten das auch die für mehrere Disziplinen relevanten fachdisziplinspezifischen Daten sein – also sämtliche vom Systems Engineering betroffenen Disziplinen. Nach WEILKIENS berücksichtigt das Systemmodell die für einen bestimmten Zweck relevanten Attribute [Wei08]. Das wirft die Frage auf, welche Attribute für den jeweiligen Zweck relevant sind. Hierdurch kommt generell eine Nähe zum Begriff des Integrierten Produktmodells auf: Das Produktmodell ist nach VDI-Richtlinie 2219 eine „*formale Beschreibung aller Informationen zu einem Produkt über alle Phasen des Produktlebenszyklus hinweg in einem Modell*“ [VDI2219] – es wird hierfür in verschiedene Partialmodelle unterteilt [GAP+93]. Das bekannteste Beispiel eines integrierten Produktmodells ist das genormte Integrierte Produktmodell STEP (Standard for the Exchange of Product Model Data). Es befähigt die formale Beschreibung von Produktdaten aus verschiedenen Anwendungsbereichen für die Abbildung aller Produktinformationen aus dem Produktlebenszyklus in ein einheitliches Produktdatenmodell [Nat14]. Damit enthält es aber auch sämtliche Informationen der Produktentwicklung die nur für eine Fachdisziplin relevant sind. Die Unterscheidung zwischen Systemmodell und Produktmodell ist somit filigran. Aus diesem Grund ist es wichtig, die zentralen Stakeholder und ihre Interessen am Systemmodell zu kennen.

2.4.4 Sprachen, Methoden und Werkzeuge

Die dargestellten Diskussionen rund um das Systemmodell haben in der kurzen Entstehungsgeschichte von MBSE zu einer großen Anzahl an Modellierungssprachen und -methoden geführt. Diese Tendenz hält an, wie die jüngst veröffentlichte Sprache LML – Lifecycle Modeling Language belegt [LML13]. Zudem werden „Dialekte“ der Sprachen entwickelt, um die Sprachen und Methoden an spezifische Bedürfnisse zu adaptieren, wie bspw. in der Automobilindustrie in ersten Ansätzen zu beobachten ist [SKD+13], [Koe13]. Alle Sprachen basieren auf unterschiedlichen Sprachsystemen – bedingt durch die Herkunft aus unterschiedlichen Fachdisziplinen. Der abgestrebte SE-Standard der Systems Modeling Language (SysML) entstammt bspw. der Softwaretechnik und nutzt daher die Konzepte der Objektorientierung. Hier kommt bereits erste Kritik auf, da das Konzept der Objektorientierung genutzt wurde, um Systemdenken auf die Softwareentwicklung zu übertragen, aber generell andere technische Disziplinen keine Objektorientierung nutzen [LML13].

Ob die durch die Objekt Management Group (OMG) getriebenen Standardisierungsaktivitäten der SysML erfolgreich sein können, ist fraglich – wie am Beispiel der „Mutter der SysML“ zu erkennen ist: Die als „vereinheitlichte Modellierungssprache“ geschaffene objektorientierte UML (Unified Modeling Language) für die Softwareentwicklung hat bislang keine signifikante Verbreitung erfahren dürfen [Pet13].

Wie sich die große Anzahl an Sprachen und Methoden auf den Erfolg der maschinenbaulichen Disziplin MBSE auswirken wird, ist momentan nicht absehbar. Bild 2-23 zeigt das Dickicht an Sprachen, wie sie im erweiterten Kontext des MBSE diskutiert werden. Während Sprachen mit einem stärkeren Ursprung in der Softwaretechnik eher auf das Verhalten eines Systems eingehen, fokussieren die Ansätze mit Ursprung im Maschinenbau die multidisziplinäre Struktur des mechatronischen Systems.

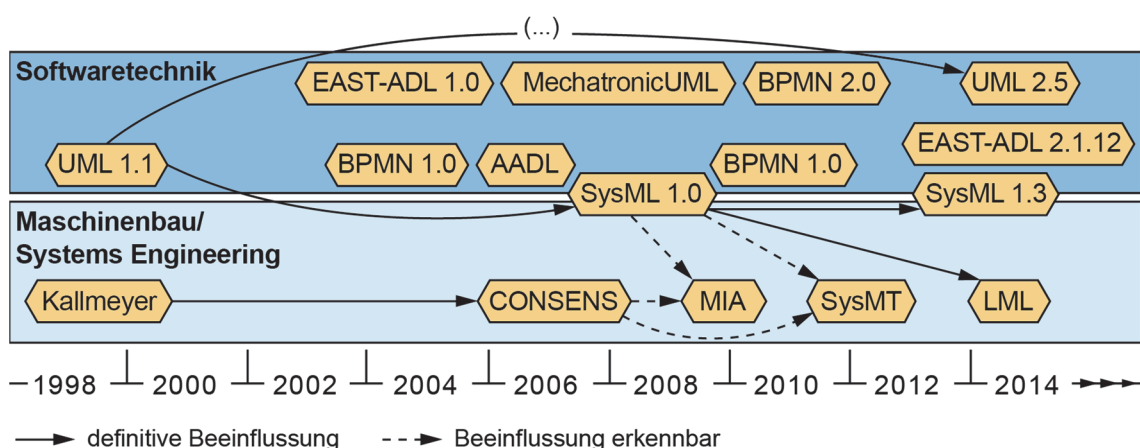


Bild 2-23: Entstehung von Modellierungssprachen im Umfeld des MBSE⁵

⁵ Zur individuellen Nachvollziehbarkeit der genannten Sprachen und Ansätze finden sich bspw. hier weiterführende Informationen [Fra06], [Kal98], [LML13], [OMG16], [Zin13].

Im Folgenden werden die für das MBSE notwendigen Elemente Sprache, Methode und Werkzeug detailliert analysiert.

Modellierungssprachen des MBSE

Wie in einer echten Sprache unterliegt der Aufbau einer Modellierungssprache gewissen Regeln [FMS08]. Dies sind die entsprechende Syntax und Semantik:

- Die Syntax legt fest, wie die einzelnen Systemelemente in ihrer Art und in ihrem strukturellen Aufbau gestaltet sind. Die abstrakte Syntax ist das Regelsystem, das die Elemente und Bezeichnungen der Sprache definiert. Ähnlich zur natürlichen Sprache (Wörter, Buchstaben, ...) sind es in den Modellierungssprachen Systemelemente, Ports, Merkmale der Elemente und die Beziehungen der Elemente untereinander. Die konkrete Syntax ist die grafische Repräsentation der abstrakten Syntax. Mit ihr werden grafische Modelle erstellt, deren Elemente auf die Elemente der abstrakten Syntax verweisen [Hap o.J.].
- Die Semantik legt fest, wie Modellkonstrukte miteinander verknüpft werden müssen, um eine Bedeutung zu haben (statische Semantik). Dies geschieht durch Bedingungen gegen die abstrakte Syntax [SVE+07]. Die Bedeutung der Modellkonstrukte ist in der dynamischen Semantik enthalten.

Die abstrakte Syntax und die statische Semantik werden in einem sogenannten Metamodell definiert [SVE+07]. Die konkrete Syntax wird separat festgelegt. Die dynamische Semantik wird ebenfalls separat in Textform oder streng mathematisch durch eine Grammatik festgehalten, was aber selbst in den formalen Sprachen der Informatik eher ein Sonderfall ist [Bal99]. Je nachdem wie Syntax und Semantik der Modellierungssprache definiert sind, ergeben sich verschiedene Möglichkeiten für die Anwendung:

- Eine reiche Modellierungssprache, d.h. umfangreiche Systemelemente, ermöglicht Automatismen [GD11], [RFB12b]. So könnten z.B. CAD-Programme auf Basis der Systemmodellldaten in grobe 3D-Modelle überführt werden oder Simulationen durchgeführt werden, wenn entsprechende Schnittstellen durch die Modellierungssprache gegeben sind. Sprachen mit strikten formalen Vorgaben sind interpretierbar und begünstigen eine automatische Datenauswertung.
- Eine weniger reiche Modellierungssprache ist einfacher zu erlernen und schneller in der Anwendung. Hierauf zielen neuere Sprachen wie OPDs/OPL und LML ab. Als Merkmale leicht erlernbarer Sprachen gelten bspw. eine geringere Anzahl an Sprachkonstrukten sowie deren übersichtliche Darstellung.

Weit verbreitet als eine Modellierungssprache des MBSE ist die Systems Modeling Language – SysML. Im Prinzip ist sie ein Dialekt der Softwaremodellierungssprache UML und wurde 2007 in ihrer ersten Version veröffentlicht. Ziel ist eine Vereinheitlichung der Sprache der Systemingenieure, um eine verbesserte Kommunikation aller Stakeholder im Projekt zu ermöglichen. Gleichzeitig soll je nach beteiligter Fachdisziplin

lin auch die Generierung von Code oder die Erstellung von Dokumentationen automatisiert werden. Während die UML als eigenständige Fachsprache etabliert werden sollte, geht es in der SysML aber um eine Ergänzung der einzelnen, in einem Projekt beteiligten Fachdisziplinen. Dennoch schlägt die Herkunft aus der Softwaretechnik voll durch, auch wenn Anstrengungen unternommen wurden, softwaretechnische Spezifika zu reduzieren. Insb. bleibt das Konzept der Objektorientierung als Basis bestehen, ebenso die Konzepte der Trennung von Modell und Sicht und das Konzept der Separation of Concerns – also vereinfacht dargestellt, dass für jeden Zweck ein eigenes Diagramm visualisiert wird und die Gesamtmenge der Einzeldiagramme erst dem Modell entspricht. Gleichzeitig ist meist eine sehr formale Modellierung gefragt, die insb. in den Details der definierten Elemente für Anwender aus Disziplinen jenseits der Softwaretechnik sehr verwirrend sein können – wozu insb. das Konzept der Ports zählt. Wie einzelne Elemente der „Muttersprache“ UML in der SysML aufgehen, zeigt Bild 4-26 [Zin13].

UML-Modellartefakt	SysML-Modellartefakt	Technische Beispiele
Class (Softwarekomponente)	Block (Systemkomponente)	Zahnrad, Welle, Steuergerät, ...
Role (Class im Systemkontext)	Block Property (Part)	Zahnrad (auf einer Welle und im Eingriff mit weiteren)
Attribut (Merkmal)	Block Property (Value)	Gewicht [kg], Modul, ...
Standard Port (Softwareschnittstelle)	Port (Systemschnittstelle)	Keilwellenprofil, Netz- anschluss, CAN-Bus-Buchse

Bild 2-24 Modifikation von UML-Artefakten in SysML (Auszug) [Zin13]

Die Auswahl einer Modellierungssprache für einen konkreten Zweck hängt damit u.a. von ihrer Handhabbarkeit ab. REBSTOCK bewertet diese auf Basis der Erlernbarkeit, Einprägsamkeit, Effektivität, Effizienz, visueller Wahrnehmbarkeit sowie Benutzerzufriedenheit [SRC10]. Die Handlungsperson muss die Sprache schnell erlernen und nach längerer Zeit ohne Anwendung immer noch verstehen können. Effizient ist eine Modellierungssprache dann, wenn der Benutzer unterstützt wird, das Modell entsprechend dem Zweck fehlerfrei und schnell zu erstellen. Die visuelle Wahrnehmbarkeit bezieht sich auf die definierten Elemente, wie Farben und Formen – daher auch häufig visuelle Syntax genannt. Alles zusammen wirkt sich auf die Benutzerzufriedenheit aus.

Eine Sprache gibt jedoch keine Vorgaben, wie sie einzusetzen ist. Von Flipcharts über Whiteboards bis hin zu rechnerunterstützten Modellen ist alles zulässig und je nach Projektkontext auch sinnvoll [Wei06]. Eine Modellierungssprache ist also ohne konkrete Modellierungsmethode und auch ohne Werkzeug nicht anwendbar [Alt12]

Modellierungsmethoden des MBSE („MBSE-Methoden“)

Eine Modellierungsmethode definiert die Abfolge von Operationen, die zur Erstellung der einzelnen Modelle durchgeführt werden [INC08], [Est08]. Der Ablauf der meisten MBSE-Methoden orientiert sich an den Schritten Anforderungsdefinition, Design, Ana-

lyse, Verifikation und Validierung. Die Modellierungsmethode muss klar vorgeben, welche Informationen zu welchem Zeitpunkt wie detailliert notwendig sind – was jedoch eher die Ausnahme ist, da die Tiefe der Modellierung weder in Sprache noch in Methode vorgeben sind. Grundsätzlich muss daher neben der Leistungsfähigkeit einer Modellierungsmethode auch immer ihre Wirtschaftlichkeit berücksichtigt werden. Sie enden mit der Erstellung der Architektur, sind aber vollkommen losgelöst von existierenden Produktentstehungsprozessen, in die sie eigentlich eingegliedert werden müssten. Es ist zu beobachten, dass das Gros der Methoden von einer Entwicklung „auf der grünen Wiese“ ausgeht und stark durch die Denkweisen der Softwaretechnik geprägt ist [RFB12a] – wie auch schon im vorangegangenen Abschnitt über die SysML als Sprache dargestellt wurde. Wie die Entstehung des Systemmodells und auch seine Pflege im Zusammenspiel mit existierenden Prozessen gelingt, wird also in keiner Weise deutlich. Die reine Erstellung des Systemmodells und die Reduktion auf den Zweck „konsistente Dokumente“ schmälern die Vision des MBSE signifikant. Die Idee einer Modellierung entsprechend einem verfolgten Zweck – bzw. einem konkreten Ziel – der über die reine Dokumentation hinausgeht, ist in aktuellen Arbeiten und Methoden nicht erkennbar. Stattdessen hat es den Anschein, dass es bei der Anwendung des MBSE um eine rein technisch getriebene Verwendung geht [Bon14].

Werkzeuge im MBSE

Mit dem zur Erstellung des Systemmodells notwendigen Werkzeug wird häufig eine Software-Lösung in Verbindung gebracht. Nach GAUSEMEIER lassen sich die verschiedenen Werkzeugklassen wie in Bild 2-25 dargestellt klassifizieren [GKP+10] und sollen je nach Zweck ausgewählt werden [Wei08].

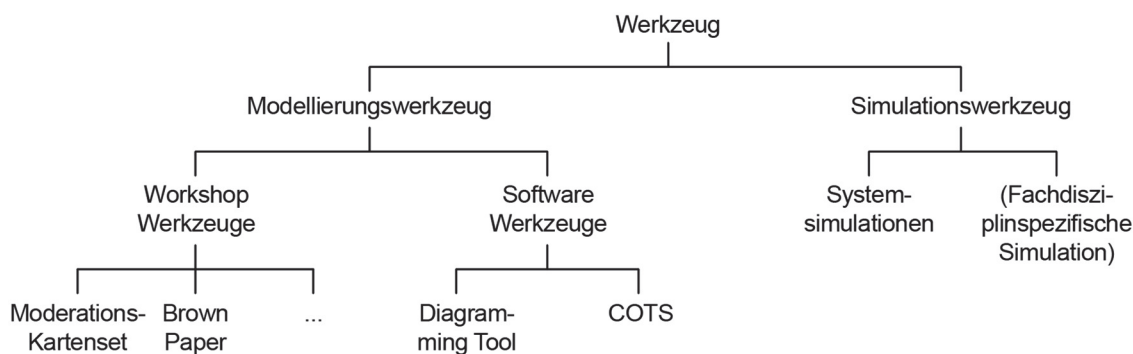


Bild 2-25: Werkzeuge des MBSE, in Anlehnung an [GKP+10]

Interessanterweise hinterlässt die Diskussion um geeignete Werkzeuge den Eindruck, dass die Werkzeugwahl eine *Entweder-oder-Entscheidung* sei. Dabei ist offensichtlich, dass sich die einzelnen Werkzeuge in gewissem Rahmen komplementär über den Lebenszyklus ergänzen können. Bei der Wahl der Werkzeuge muss selbstredend auch die Wahl einer für das Werkzeug geeigneten Sprache berücksichtigt werden.

2.4.5 Erstellung und Nutzung des Systemmodells

Die Erstellung des Systemmodells wird mit verschiedenen Sprachen, Methoden und Werkzeugen unterstützt. Der Systems Engineer soll das Systemmodell als sein zentrales Arbeitsmittel nutzen – in den verschiedenen Methoden wird allerdings nicht dargestellt, wer überhaupt für die Erstellung des Systemmodells verantwortlich ist und wer darüber hinaus Nutzen in der verteilten Zusammenarbeit daraus ziehen kann. Nach PARNELL ET AL nimmt der Systems Engineer eine „Over all“-Perspektive im Prozess ein, er muss alle Fachdisziplinen gleichermaßen berücksichtigen und ist dafür verantwortlich, dass ein System entsteht, das die Kundenanforderungen erfüllt [PDH11]. KAISER fordert in ihrem Vorgehensmodell zur Modellierung einer plausiblen Systemstruktur, dass der Systems Engineer als „Systemmodellverantwortlicher“ agiert und Workshops mit den Fachexperten moderiert – im Anschluss an diese Workshops ist er für die Aufbereitung der Arbeitsergebnisse verantwortlich [Kai14]. Dabei ist darauf zu achten, dass der Systems Engineer nicht als bloßer Modell-Verwalter angesehen wird. Da die Erstellung des Systemmodells allerdings in Abhängigkeit vom Modellierungszweck zu gestalten ist, rücken – im Gegensatz zur idealisierten Vorstellung – viele neue Stakeholder in den Mittelpunkt der Modellerstellung und -pflege, die dann für diese Tätigkeiten verantwortlich zeichnen. Für wen das Systemmodell dabei Nutzen entfalten könnte und durch wen es erstellt werden sollte, zeigt Bild 2-26 qualitativ. Hierbei fällt insb. der hohe Nutzen für die verschiedenen Rollen jenseits der Entwicklungstätigkeit auf. Der Systems Engineer ist in der Modellierung beteiligt, jedoch nicht allein verantwortlich.

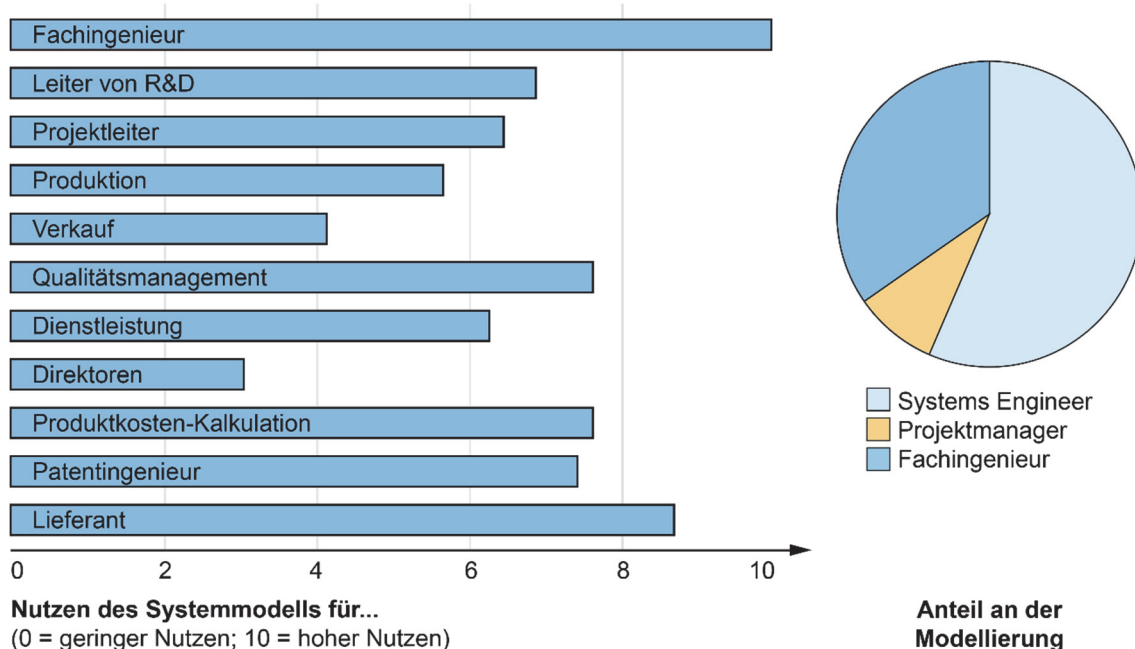


Bild 2-26: Entstehung des Systemmodells, nach [TKD+14]

In verteilten Entwicklungsorganisationen muss die Zusammenarbeit aber klar geregelt sein – diesem Thema wird im MBSE bislang nicht viel Aufmerksamkeit geschenkt. Das betrifft zunächst die bereits aufgeworfenen Fragestellungen nach dem Umfang und In-

tensität der Modellierung, insb. aber auch der Organisation der Modellierung: *Welche Stakeholder geben Input, welche Stakeholder nutzen Informationen aus dem Modell und wann?* Wie die in Bild 2-26 genannten Stakeholder, die meist nicht aktiv entwickeln, den Nutzen aus dem Systemmodell ziehen können, bleibt unklar. FRIEDENTHAL und [Zin13] nehmen für die Erstellung aber zumindest eine grobe Einteilung der Verantwortlichkeiten vor und stellen dem Kreis der Systems Engineers ein Kernteam zur Seite, das für den Großteil der Modellierungsaktivitäten verantwortlich ist. „Jedermann“ wiederum muss ein Grundverständnis haben, um erfolgreich zu modellieren. (Bild 2-27). Zunächst zeigt das, dass evtl. die Rolle des Systems Engineers im Model-Based Systems Engineering überdacht werden muss. Andererseits fällt aber auch auf, dass es sehr stark um die Modellierung geht – weniger um die Nutzung des Systemmodells bzw. der darin enthaltenen Informationen zur Erledigung von Aufgaben eines Stakeholders im Prozess, was in Bild 2-27 durch die fehlende Rückführung von Informationen aus dem Systemmodell erkennbar ist.

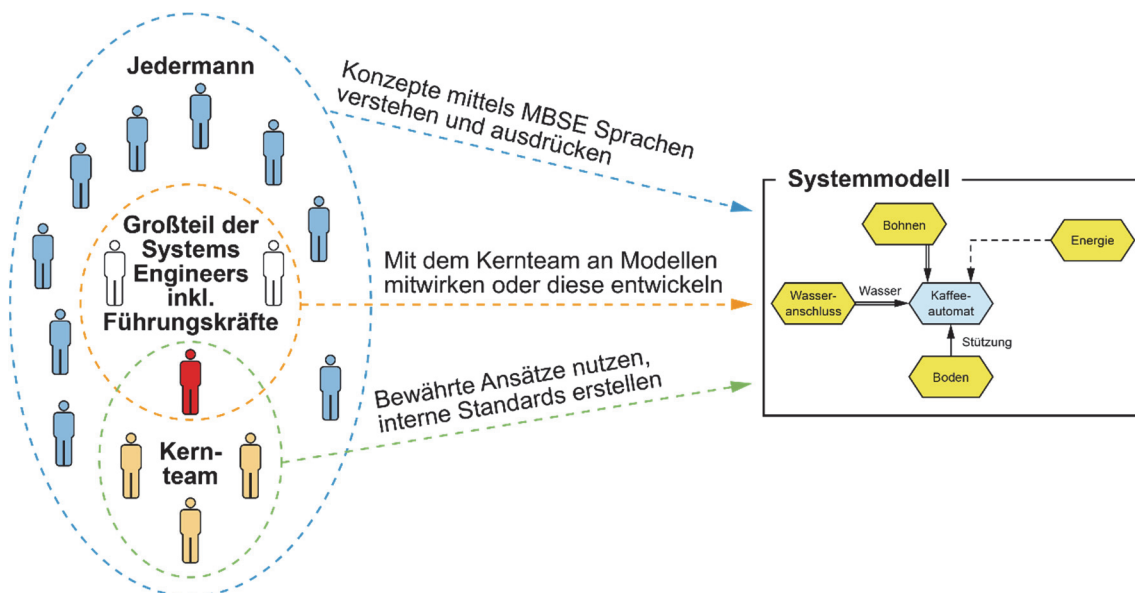


Bild 2-27 Rollen bei der Erstellung des Systemmodells nach FRIEDENTHAL

Architecture Frameworks

Wie die Modellierung des Systemmodells strukturell organisiert werden könnte, zeigen sog. Architecture Frameworks (AF). Sie strukturieren die Sichten von Stakeholdern auf Architekturen – allerdings meist IT- oder Unternehmensarchitekturen. Im ZACHMAN-Framework wird bspw. definiert, welche Stakeholder welche Aspekte bei der Gestaltung der IT-Architektur zu berücksichtigen haben [Zac87]. Generell stellen Frameworks ein Beschreibungsmodell der Architektur zur Verfügung – Prozesse und Abläufe und auch Sprachen sind nicht Teil des Frameworks [Rot12]. Auch definieren sie nur bedingt, was konkret und womit abgebildet werden muss. Dieser Aufgabe nimmt sich teilweise die ISO 42010 an: Sie definiert Begriffe der Architekturbeschreibung, sog. Concerns, Viewpoints und Views eines Stakeholders. Bei der Anwendung soll die

Überlagerung einzelner Views im Sinne der Modelltheorie eine kohärente Darstellung des technischen Systems im Sinne des Systemmodells ergeben – jede Sicht wird meist in einem eigenen Diagramm dargestellt.

Concern, Viewpoint und View

Die von STACHOWIAK definierten Merkmale *Abbildung* und *Verkürzung* eines Modells haben Auswirkung auf den Umfang der Modellierung und sind abhängig vom Stakeholder und der Lebenszyklusphase – jeder Stakeholder hat ein spezifisches Interesse an dem Systemmodell. Der Begriff Concern „*any topic of interest*“ bringt dies zum Ausdruck [ISO42010]. Jeder Stakeholder modelliert also die für ihn relevanten Information in einem bestimmten Abstraktionsgrad mit spezifischen Sprachen – der Begriff *Viewpoints* (Standpunkt) fasst dies zusammen. Das Ergebnis der Modellierung ist dann der spezifische View – also genau das, was der Stakeholder zu sehen wünscht. Der View ist „*the part of a design, constructed from a viewpoint, that represents the interest of that stakeholder. [Dij06]*“. Das Zusammenspiel von Stakeholder – Concern – Viewpoint – View wird in der ISO42010 strukturell dargestellt, ohne jedoch für spezifische Entwicklung verbindliche Vorgaben oder zumindest Beispiele zu generieren [ISO42010]. Ebenso existiert kein Bezug zum Lebenszyklus und zu den Sprachen und Methoden des MBSE.

Die grafische Darstellung eines Views im MBSE gelingt mit Diagrammen – die inhaltlich einer Sicht entsprechen. Diese Trennung des digitalen „Modells“ – häufig Repository genannt [Alt12] und der „Sicht“ stammt aus der Softwaretechnik und der virtuellen Produktentstehung. Wie verschiedene Views auf ein System entstehen und zu verstehen sind, stellt Bild 2-28 schematisch dar. Durch die rechnerinterne Darstellung könnten die Inhalte bei entsprechender Werkzeugausstattung und notwendigem Formalisierungsgrad weiterverarbeitet werden. Gerade die Nutzung dieser Informationen – im Vergleich zur Modellierung – wird bislang selten adressiert. Insbesondere der Fokus auf die Unterstützung von „Concerns“ jenseits der technischen Prozesse wie z.B. Simulation, wird überhaupt nicht adressiert.

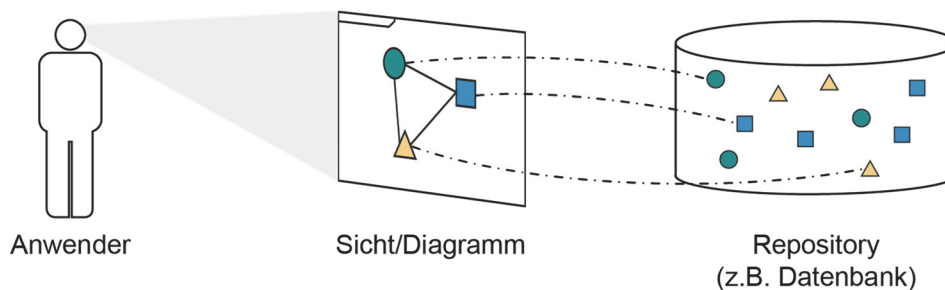


Bild 2-28: Die Trennung zwischen Diagramm und Repository nach [Alt12]

Grundsätze ordnungsgemäßer Modellierung (GOM)

Weder die MBSE-Methoden noch die Architecture Frameworks und die ISO 42010 geben konkreten Hilfestellungen, wie dem Zweck entsprechend modelliert werden soll

– eine ähnliche Problematik wie in Bild 2-8 diskutiert. In der Wirtschaftsinformatik befasst sich das Themengebiet Prozessmanagement mit ähnlichen Fragestellungen wie sie für das MBSE relevant sind: Umfang und der Intensität der Modellierung. Hier geht es ebenfalls um die Anwendung geeigneter Sprachen, Methoden und Werkzeuge [Bro09]. Ebenso identisch ist die Problematik, dass die Modelle schnell komplex und unübersichtlich werden oder häufig redundante Informationen in Teilmodellen erstellt werden, die wiederum konsistent gehalten werden müssen. Die sechs Grundsätze ordnungsmäßiger Modellierung nach BECKER fassen diese Problematik auf (vgl. Bild 2-29) und geben – in Anlehnung an die Grundsätze ordnungsgemäßer Buchführung – grobe Leitlinien bei der Modellierung vor [BPV12]. „Relevanz“ und „Wirtschaftlichkeit“ deuten an, dass der Nutzer des Modells auch für die Erstellung in gewissem Maße zuständig ist, da gerade er entscheiden kann, was wichtig ist. Im Zusammenspiel mit dem Grundsatz „Systematischer Aufbau“ wird das besonders ersichtlich: Durch das Zusammenspiel verschiedener Stakeholder entstehen verschiedene Sichten auf das System. Der Nutzen des MBSE wird gerade dann groß, wenn jede Handlungsperson der Produktentstehung die für sie relevanten Sachverhalte nicht nur beschreiben kann, sondern auch für weitere Aktivitäten ausleiten und weiterverarbeiten kann – hier zeigt sich eine enge Verknüpfung mit dem Thema Sichten. Ebenfalls ist erkennbar, dass die Erstellung des Systemmodells nur zu einem gewissen Anteil durch den Systems Engineer geleistet werden kann (vgl. Bild 2-26) und dass das Zusammenspiel bei der Modellierung und die Informationsnutzung durch klare Prozesse geregelt werden muss. Gerade diesen Grundsätzen wird in der Weiterentwicklung des MBSE wenig Aufmerksamkeit gewidmet – allerdings entscheiden genau diese über die Akzeptanz des Ansatzes.

<p style="text-align: center;">Richtigkeit</p> <ul style="list-style-type: none"> • Syntaktische Richtigkeit (formal korrekt) • Semantische Richtigkeit 	<p style="text-align: center;">Wirtschaftlichkeit</p> <ul style="list-style-type: none"> • Modellierung ist verbunden mit Aufwand (Aufwand-Nutzen Verhältnis) • Referenzprozesse einsetzen 	<p style="text-align: center;">Vergleichbarkeit</p> <ul style="list-style-type: none"> • Ein Sachverhalt mit zwei Modellierungssprachen • Zwei ähnliche Sachverhalte mit einer Sprache
<p style="text-align: center;">Relevanz</p> <ul style="list-style-type: none"> • Modellierungsziel entscheidet über Abstraktionsniveau und Relevanz der modellierten Sachverhalte 	<p style="text-align: center;">Klarheit</p> <ul style="list-style-type: none"> • Lesbar, Verständlich und Anschaulich • Adressatengerechtes Hierarchisieren, Layout und Filterung 	<p style="text-align: center;">Systematischer Aufbau</p> <ul style="list-style-type: none"> • System wird aus unterschiedlichen Sichten dargestellt • Ziel ist ein konsistentes Gesamtmodell

Bild 2-29: Grundsätze ordnungsgemäßer Modellierung

2.4.6 Product Lifecycle Management und Workflows

Im Kontext des MBSE werden immer häufiger die existierenden Produktdatenmanagement- (PDM) und Product Lifecycle Management-Ansätze (PLM) diskutiert, die als lebenszyklusbegleitende Systeme alle Informationen und Prozesse managen sollen.

Hauptgrund ist der bislang stark vernachlässigte Prozessgedanke im MBSE im Vergleich zur starken Betonung der reinen Spezifikation [ERZ14]. Viel wichtiger erscheint jedoch der Gedanke, dass mit MBSE das Konzept des ganzheitlichen lebenszyklusbegleitenden Managements aller Informationen und Prozess erstmalig tatsächlich realisiert werden kann. Streng genommen sind aktuelle PLM-Anwendung eigentlich nur PDM-Systeme, d.h. sie verwalten CAD-Daten einer Disziplin und vernachlässigen die disziplinübergreifenden Aspekte. Zudem unterstützen sie gut die Engineering Collaboration. Beides sind allerdings auch Gründe, warum diese Systeme bislang im Management und dessen Analysen keine große Bedeutung haben. Mit der Ausweitung der PLM-Aktivitäten in Richtung MBSE wird allerdings die Tür aufgestoßen in Richtung einer intensiven Managementunterstützung auf Basis der technischen Informationen über das in der Entwicklung befindliche System, die obendrein die noch häufig nicht virtualisierten Inhalte der frühen Produktentstehungsphasen umfasst. Bislang konnten also die für das Management notwendigen Informationen nicht ausreichend und detailliert genug über PLM bereitgestellt werden – ERP-Systemen als typische Management-Systeme der Produktion fehlen aber meist die technischen Informationen über das Produkt. MBSE hat das Potential die beiden Bereiche zu ergänzen, weshalb hier die Visionen von PLM, ERP und SE ineinander greifen – was aufgrund der Grundidee des Systems Engineerings nachvollziehbar ist. Primär ist dabei aber nicht nur die Verfügbarkeit und das Management der Daten, sondern auch deren Anwendung und Nutzung in den bestehenden Aufgaben des Prozesses eine Aufgabe, die das MBSE bedienen muss (vgl. Bild 2-30).

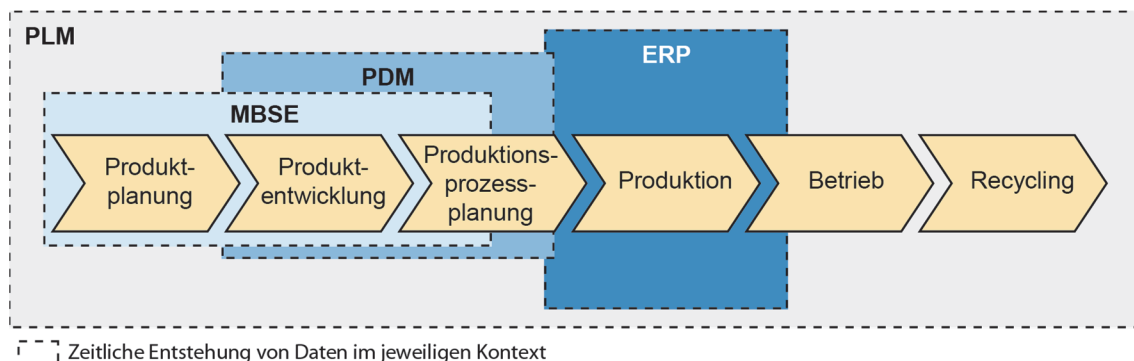


Bild 2-30: MBSE, PDM, ERP und PLM, eigene Darstellung angelehnt an [ERZ14]

Damit rücken die dargestellten Ideen nah an das Thema Workflow-Management. Eine Idee des Workflow-Managements ist, die richtigen Daten zur richtigen Zeit der richtigen Person zur Verfügung zu stellen [Sch01] – bspw. durch einen formal beschriebenen, ganz oder teilweise automatisierten Geschäftsprozess [Qui04]. So können operative Tätigkeiten automatisiert werden und ein enormer Beitrag zur Effizienzsteigerung geleistet werden. Die dahinterstehende Vision wurde schon auf das Management der Produktentstehung übertragen [Roe11]. Jedoch sind nicht alle Aufgaben gleich gut über Workflow-Management-Systeme abbildbar (vgl. Bild 2-31). Produktentwicklungsprojekte sind in der Regel schwach strukturierte Prozesse, die einen hohen individuellen und kreativen Anteil enthalten und nie in derselben Form wiederholt werden. Sie kön-

nen nur schlecht durch Workflows unterstützt werden, da hier starre Prozesse und Aufgaben nach gleichem Schema erforderlich werden. Ein Prozess kann nur dann unterstützt werden, wenn er vorausschauend koordiniert werden kann und keine unstrukturierten Aktivitäten und freien Entscheidungen enthält. Damit müssen Produktentwicklungsprojekte bislang tendenziell in den Bereich eingeordnet werden, der nur schwierig durch Workflows unterstützt werden kann. Durch das MBSE können die bislang unstrukturierten Aktivitäten und Entscheidungen bei der Entwicklung nun jedoch auf feingranularer Ebene – d.h. insb. frühzeitig verfügbaren Produktinformationen – verfolgt werden, was hier ein Zusammenwachsen der Thematik ermöglicht. Dadurch werden die im MBSE modellierten Inhalte zum Dreh- und Angelpunkt und können für sämtliche Aufgaben im Produktentstehungsprozess genutzt werden.

Teilaufgaben im Prozess Prozesstyp	Einzelfallaufgabe	Sachbezogene Aufgabe	Routineaufgabe
Einmaliger Prozess	Ad-hoc-Workflow-Management	Transaktions- Workflow-Management	Transaktions- Workflow-Management
Regelprozess			
Routineprozess			

für Workflow-Management nicht geeignet
 für Workflow-Management geeignet
 für Workflow-Management voll geeignet

Bild 2-31: Einsatz und Eignung von Workflows [Sch01], [Roe11]

2.5 Herausforderung MBSE

Im vorangegangenen Kapitel wurde das Model-Based Systems Engineering als Paradigma der Produktentstehung von morgen eingeführt. In der letzten Zeit erfährt das Thema immer mehr Zuspruch in der Industrie. Dennoch ist die Durchdringung der Produktentstehung mit MBSE eher kühne Vision als nahe Realität. In diesem Kapitel werden die Herausforderungen, die mit der Einführung und Nutzung von MBSE verbunden sind, vor dem Hintergrund der vorangegangenen Darstellungen analysiert.

2.5.1 Nutzen und Wertbeitrag von MBSE

Durch MBSE ergeben sich große Möglichkeiten – bei gleichzeitig vorhandenen Kinderkrankheiten. Schon beim Systems Engineering, aber v.a. beim MBSE ist die Diskussion über seinen Wertbeitrag noch sehr jung: Der Ansatz war durch die Anwendung in sicherheitskritischen Systemen der Luft- und Raumfahrt nicht auf den Nachweis von Wirtschaftlichkeit angewiesen. Erst mit dem aufkommenden Interesse der Privatwirt-

schaft in den letzten Jahren kam die Wirtschaftlichkeitsdiskussion auf [TA16]. Die Bewertung des Nutzens und des Wertbeitrags steht vor einem Dilemma:

- SE ist ein Lebenszyklus-Ansatz, fokussiert auf die Zeit von der Wiege eines Produkts bis zu dessen Bahre. Hierzu ist die Bereitschaft notwendig, gerade in den frühen Lebenszyklusphasen einen erhöhten Aufwand zu akzeptieren. Das soll sich in späteren Phasen ausgleichen, da u.a. die bislang üblichen zeit- und kostenintensiven Iterationsschleifen verringert werden sollen.
- Investitionen in neue Technologien und Methoden sollen einen möglichst kurzen Return-on-Investment (RoI) haben, d.h. das investierte Kapital sollte schnell durch nachfolgende Einsparungen erwirtschaftet werden. Erfahrungsgemäß wird in der Industrie von einer Amortisationsdauer von 12 bis 24 Monaten ausgegangen – im Gegensatz zu meist mehr als 60 Monaten Produktlebenszyklus.

RHODES und ROSS beschreiben die gewünschten Auswirkungen des MBSE auf die Entwicklung wie folgt: Durch das MBSE wird zu einem früheren Zeitpunkt viel Wissen über das System generiert, sodass dieses Wissen bereits vor der Festlegung des Großteils der Kosten besteht. Dadurch können bessere Entscheidungen getroffen werden [RR09]. Diese Sicht entspricht der vorherrschenden Meinung, dass in der Entwicklung nur ein geringer Teil der Kosten eines Projekts anfallen, jedoch der Großteil der zukünftigen Kosten festgelegt werden.

Aus diesem Grund ist die Diskussion um den Nutzen von MBSE und SE umso wesentlicher. Im MBSE existieren keinerlei Arbeiten zu diesem Thema, im klassischen SE sind nur wenige Arbeiten bekannt, die sich mit seinem Wertbeitrag auseinandersetzen:

- In älteren Arbeiten beim Luftfahrtunternehmen Boeing wurden für drei gleichzeitig ablaufende ähnliche Projekte SE-Aktivitäten in unterschiedlichem Umfang eingesetzt [Fra95] – hier zeigten sich bei erhöhtem SE-Aufwand ein positiver Einfluss auf die Produktqualität und die Projektlaufzeit.
- Rolls-Royce hat durch SE in mehreren Testprojekten eine Reduktion der fehlerbedingten Designänderungen von 30-70% auf etwa 2% erreicht [DYY+13].
- In einer quantitativen Studie von HONOUR wurde die Korrelation von Budget- und Terminüberschreitungen mit dem investierten SE-Aufwand untersucht [Hon13]. Das Optimum für SE-Aktivitäten lag bei etwa 15% des Projektbudgets; so erreichten die Überschreitungen ein Minimum. Sowohl weniger als auch mehr Aufwand für SE führten zu schlechteren Ergebnissen.
- ELM und GOLDENSON untersuchten in ihrer Studie die SE-Effektivität. Dazu wurde der Zusammenhang zwischen der Durchführung von SE-Aktivitäten und der Performance des entsprechenden Projektes analysiert. Untersuchte SE-Aktivitäten waren u.a. Anforderungs-, Konfigurations- oder Risikomanagement.

Dabei wurde eine positive Korrelation zwischen Anwendung von Systems Engineering und der Performance des Projektes festgestellt [EG12].

Bei allen Ansätzen wird auf die klassischen SE-Prozesse abgezielt, wie bspw. die der ISO15288. Der Aufwand für Projektmanagement-Aktivitäten ist noch nicht einberechnet. Das zeigt: Unternehmen müssen zukünftig bei der Planung von Projektbudgets umdenken und zu den PM-Aktivitäten auch Ressourcen für SE einplanen. Für MBSE sind zudem noch spezifische Software-Werkzeuge notwendig. Ebenso muss klar sein, wie das MBSE an die existierenden Prozesse und Aufgaben gekoppelt wird. Darüber hinaus müsste bei einem ganzheitlichen Ansatz wie dem Systems Engineering auch ein ganzheitlicher Ansatz zur Nutzenmessung herangezogen werden, d.h. Effekte jenseits der oben erwähnten Kriterien – doch derartige Konzepte sind vollkommen unbekannt.

2.5.2 Studien zur Verbreitung von SE und MBSE in der Praxis

INCOSE hat das MBSE zum zentralen Paradigma der Produktentstehung von morgen erkoren [INC15]. Dennoch ist der Ansatz trotz seiner propagierten Stärken in der Industrie bislang wenig verbreitet. Breit angelegte Studien zur Verbreitung existieren bislang nicht. Einige kleinere Studien bestätigen jedoch den Eindruck, dass das Interesse an und die Notwendigkeit für MBSE kontinuierlich zunehmen.

Studie von CLOUTIER und BONES [BC10]

Die Studie von CLOUTIER und BONES gibt einen Überblick über MBSE in den USA. Hierbei wurden jedoch vornehmlich Unternehmen befragt, die schon intensive Erfahrungen mit dem Ansatz hatten. Unklar ist, welcher Branche diese Unternehmen entstammen. Dennoch: Der Nutzen von MBSE wird gerade für den Systems Engineer und die Projektleitung hervorgehoben, verantwortlich für die Modellierung ist der Systems Engineer, der jedoch von einem starken Team unterstützt wird (vgl. Bild 2-32). Weiterhin ist die Anwendung von unternehmensinternen Modellierungsmethoden geprägt, von einer Integration in die Prozesse kann nicht die Rede sein.

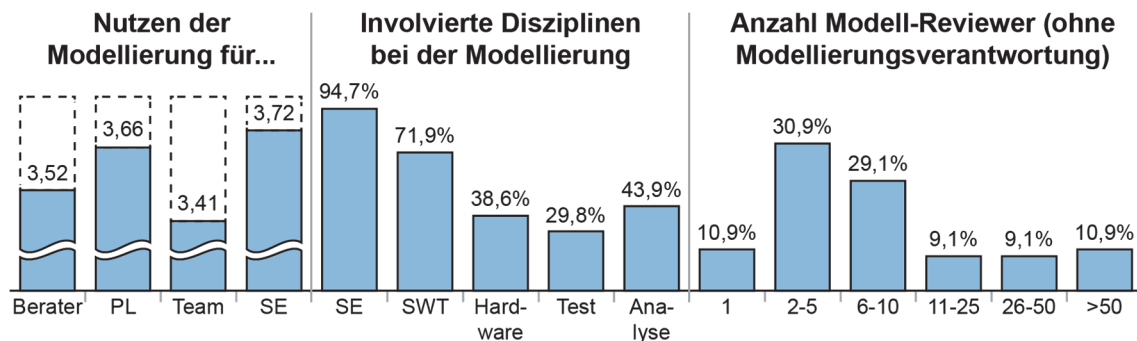


Bild 2-32: Studie von Bones und Cloutier

Studie von Gausemeier et al. [GDS+13b]

Die Studie von GAUSEMEIER ET AL. berücksichtigt insg. 32 Unternehmen aus der sog. D-A-CH-Region. Ziel ist ein Bild über das Verständnis und den Leistungsstand des SE. Allgemein fällt die ungleiche Leistungsfähigkeit zwischen den Branchen in den SE-Kernthemenfeldern auf (vgl. Bild 2-33): Der klassische Maschinenbau ist eher abgeschlagen – der Aufwand für SE wird gescheut, wenngleich durchaus Potential im SE-Ansatz gesehen wird. In der Automobilindustrie haben insb. die OEM die Bedeutung des SE erkannt und fordern dessen Weiterentwicklung für die Anforderungen variantenreicher Massenprodukte. Auch der Vergleich zwischen den einzelnen Themenfeldern ermöglicht interessante Schlussfolgerungen, insb. in der Luft- und Raumfahrt: Diese schätzen ihre Leistungsfähigkeit im MBSE (Punkt B) weitaus stärker ein, als in den vor- bzw. nachgelagerten Themen *Anforderungsmanagement* und *modellbasierte Entwicklung*. Das spricht sehr für eine softwarelastige Sicht auf MBSE.

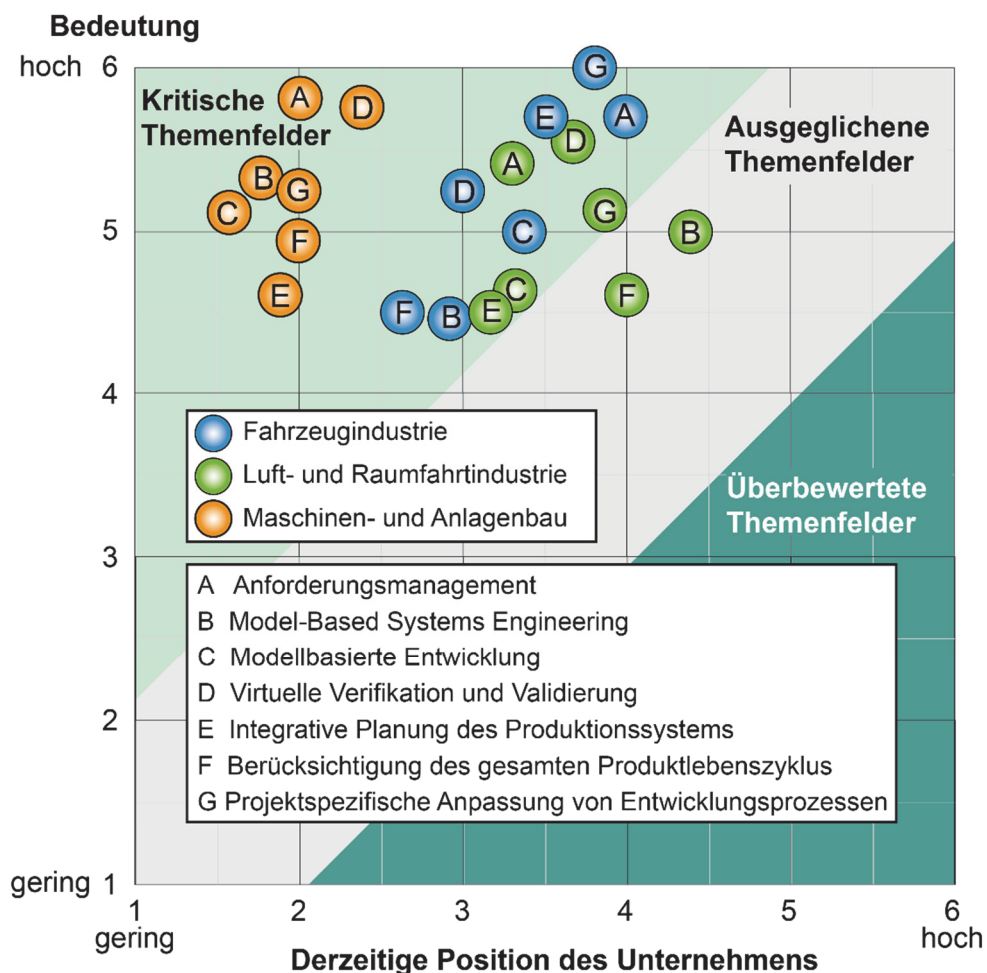


Bild 2-33: Leistungsfähigkeit unterschiedlicher Branchen in Themenfeldern des SE

2.5.3 Der Weg zu MBSE

Wesentliche Herausforderung für die Verbreitung des Systems Engineerings und des MBSE ist der Begriff selbst: In den Fachdisziplinen wird der Begriff für diesen fachdisziplinübergreifenden Ansatz unterschiedlich verwendet [Taz10], [GDS+13b]. Häufig verunsichert auch eine Sprachungenauigkeit: Systems Engineering legt den Schwerpunkt auf die Denkweisen, Methoden und Prozesse zur Verknüpfung einzelner Aktivitäten eines konkreten Projekts. Hier ist es wichtig, Wissen über jede der involvierten Disziplinen zu besitzen – aber nicht in der Tiefe eines Fachingenieurs. Dagegen erfordert System Engineering spezielles Wissen einer Disziplin, wie z.B. Mechanik oder Softwaretechnik – es ist damit die konkrete Anwendung von Methoden der Konstruktion oder Softwareentwicklung. Potentielle Anwender kennen diesen Unterschied häufig nicht und setzen – wenn überhaupt – MBSE mit den Ansätzen wie CaX gleich [TS15].

Der erfolgreiche „Weg zu MBSE“ wird allerdings auch durch einen weiteren Aspekt behindert: Die Branchen, für die MBSE Nutzen stiften kann, haben durch Konzepte wie „Lean“ oder „Six Sigma“ in der Produktion und den administrativen Bereichen schon signifikante Effekte mit einfachsten Mitteln erzielen können und eine große Anhängerschaft. Die Ansätze der Produktentstehung wie z.B. Concurrent Engineering oder allgemein das methodische Entwickeln werden dagegen häufig als nicht mehrwertig empfunden. Deshalb sehen vielleicht gerade kleine und mittelgroße Unternehmen die Standards und Vorgehen des SE als zu unhandlich für ihre Zwecke. Es mangelt also an akzeptierten Richtlinien, konsistenten Terminologien und Definitionen, sowie an Modellen, die von vielen Domänen gemeinsam und auch unternehmensübergreifend genutzt werden können [INC07], [INC12]. In einer INCOSE-Initiative werden die aktuellen Hindernisse noch deutlicher dargestellt: Neben Widerständen gegenüber neuen Methoden und eine fehlende Managementunterstützung ist der nicht direkt erkennbare Nutzen von MBSE Haupthindernis für den „Weg zu MBSE“ [Clo13].

2.6 Problemabgrenzung

„Model-Based Systems Engineering wird allerorts diskutiert, bis auf Ausnahmen fehlt es aber an konkreten Einsätzen. [...] Eine erfolgreiche Anwendung erfordert [...] noch Weiterentwicklungen bei Methoden und unterstützenden Werkzeugen“ [GAC+13].

Zwei Mega-Trends erfordern starke Veränderungen in der Produktentstehung: Die dynamische Entwicklung der Informations- und Kommunikationstechnik eröffnet die Perspektive intelligenter technischer Systeme als Weiterentwicklung mechatronischer Systeme. Diese Erzeugnisse werden in der Lage sein, sich selbstständig und flexibel an wechselnde Betriebs- bzw. Umgebungsbedingungen anzupassen. Sie werden meist aus einer Vielzahl von untereinander vernetzten Teilsystemen bestehen. Die Gesamtfunktionalität des Erzeugnisses erschließt sich erst durch das Zusammenspiel der einzelnen Teilsysteme. Vielmehr als noch bei mechatronischen Systemen ist Transparenz über die

Wirkungsweise des Systems eine der Hauptaufgaben in ihrer interdisziplinären Entwicklung. Zudem führt der starke Innovations- und Wettbewerbsdruck zu neuen Wertschöpfungsstrukturen und hohen Anforderungen an Produktentstehungsprozesse – das Konzept der „Weltfabrik“ wird zunehmend abgelöst durch agile Innovations- und Entwicklungsnetzwerke, die über Unternehmensgrenzen, Zeitzonen und Kulturkreise hinweg ihre Leistung erbringen müssen, häufig bezeichnet als Brokernetzwerk.

Die rasant gestiegene Produktkomplexität und die Veränderungen der Wertschöpfungsstrukturen erfordern neue Herangehensweisen in der Produktentstehungsarbeit: Der Wandel zu intelligenten Systemen benötigt grundlegende Veränderungen in der Entwicklungsmethodik, die neuen Wertschöpfungsstrukturen bedürfen neuartiger Kollaborationskonzepte. Die Konsequenz ist ein wachsendes Interesse an den Ansätzen des modellbasierten Systems Engineering. Mittels semi-formaler grafischer Modellierungssprachen soll das technische System fachdisziplinübergreifend und für alle Stakeholder verständlich beschrieben und die Zusammenarbeit vereinfacht werden. MBSE verfolgt das Ziel, sämtliche Aktivitäten der Produktentstehung zu unterstützen. Damit erscheint es vor den dargestellten Herausforderungen als der richtige Ansatz zur Beherrschung der aus den Veränderungen resultierenden Komplexität von Produkt und Organisation (Bild 2-34). Gleichzeitig scheint es mit seinen Grundlagen im Systems Engineering das Potential zu besitzen, die Kreation technischer Systeme auf eine neue Basis zu stellen und damit als zentrales Paradigma in der Produktentstehung akzeptiert zu werden.

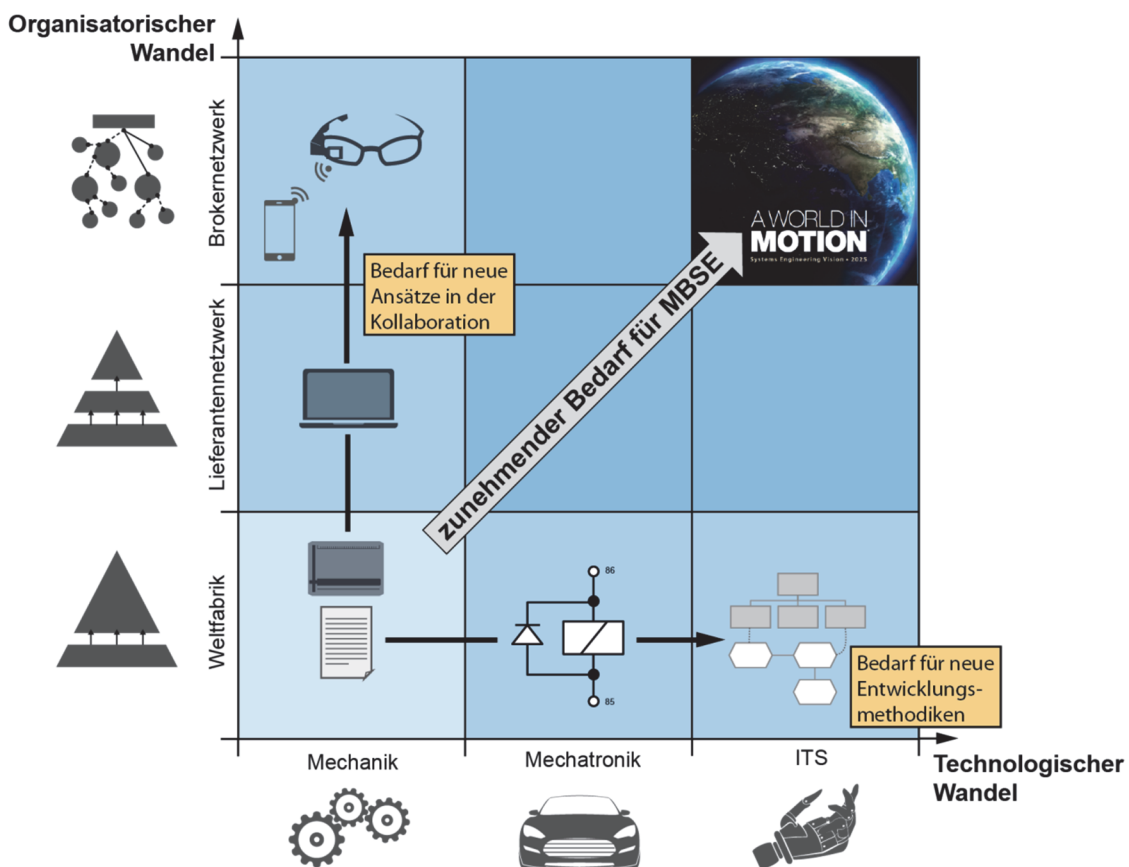


Bild 2-34 Bedarf für MBSE durch Veränderungen in Organisation und Technik

Im Mittelpunkt des MBSE steht bislang eine konsistente Spezifikation des Systems als Ausgangspunkt für die fachdisziplinspezifischen Arbeiten. Die frühzeitige Verfügbarkeit von detaillierten Informationen über das System und seine Wirkungsweise führt dabei zu hohen Erwartungen an eine Entwicklung nach dem Prinzip „1st Time Right“ und einer „Single Source of Truth“, die beliebige multidisziplinäre Systemsimulationen auf Knopfdruck ermöglichen soll. Aufgaben, die allerdings ebenfalls Kernaktivitäten des Systems Engineerings sind, werden bislang vernachlässigt – bspw. die sog. produktbezogenen Projektmanagementaktivitäten („Technical Management“ [ISO15288]).

Gleichzeitig sollte – trotz berechtigter Hoffnungen – die fachdisziplinübergreifende Systemmodellierung zunächst nicht als „Wunderwaffe“ zur Lösung aller Probleme betrachtet werden [Bro11]. Vielmehr sollte die Devise für eine MBSE-Einführung zunächst dem Motto „*Keep it simple*“ folgen. Dass dabei dennoch der ganzheitliche Charakter des Systems Engineerings berücksichtigt werden kann, kommt durch CLOUTIERIS Verständnis zur Aufgabe von MBSE zum Ausdruck [Clo09] – Modellierung soll helfen

- das Problem zu verstehen,
- Komplexität zu erkennen und zu beherrschen [Clo13] und
- die Kommunikation mit allen Beteiligten zu verbessern.

Vor diesem Hintergrund wird klar, dass das MBSE vor einigen **Herausforderungen** steht. Nach STOEWER ist das nicht verwunderlich, da es sich in einem Entwicklungsstadium wie die CaX-Technologien in den 1950er Jahren befindet [Sto12a] – mit dem Unterschied, dass mehrere Fachdisziplinen Anspruch auf das junge Thema erheben und unterschiedlichste Sichtweisen und Ziele aufeinanderprallen. Gleichzeitig ist die Durchdringung der Produktentstehung mit Methoden äußerst gering. Konkret können vor dem Hintergrund der vorangegangenen Problemanalyse die folgenden Herausforderungen identifiziert werden, die die Einbindung des modellbasierten Systems Engineerings in die Produktentstehungsarbeit behindern und sich auch stark gegenseitig bedingen:

- **Das Konzept von MBSE ist unklar.** Es gibt keine einheitliche Meinung, was MBSE ist [Clo09]. Diese Problematik wird durch den Wildwuchs an Sprachen und Vorgehensweisen verstärkt. Selbst erfahrende SE-Praktiker empfinden die Ansätze als zu kompliziert und zu werkzeuggetrieben [Sto12b], [LML13]. FRIEDENTHAL stellt fest: „*Careful tailoring [...] is essential to meet the needs of a particular project*“ [FMS08]. Erschwerend wirkt, dass Konzepte wie „Six Sigma“ mit vielen kleinen Hilfsmitteln erfolgreich sind, aber konzeptionell nicht einfach auf die Produktentstehungsarbeit übertragen werden können.
- **Es existiert eine Lücke zwischen dem MBSE und dem klassischen SE und damit auch den Aufgaben der Produktentstehung.** Der Fokus existierender Arbeiten liegt auf der Modellierung in der Konzeptphase [RFB13]. Die Modelle werden nicht in ausreichendem Maße für das Lösen konkreter Probleme oder zur

Unterstützung konkreter Aufgabe eingesetzt. Insb. die vielen Aufgaben des klassischen Systems Engineerings und des Entwicklungsmanagements spielen bislang keine Rolle im MBSE. Dadurch kommt die Anwendung von Methoden als Ausgangspunkt der ingenieurmäßigen Vorgehensweisen zu kurz [Gre03].

- **Die Nutzung des Systemmodells wird nicht angemessen geplant.** Viele Arbeiten und Vorgehensmodelle hinterlassen den Eindruck, dass ohne konkretes Ziel und ohne Aufgabe modelliert wird. Der Grundsatz der Zweckorientierung muss aber zwingend erfüllt sein, eine wohlfundierte Modellplanung ist essentiell für den Projekterfolg [VDI2212], [Kno14]. Die vielfältigen Möglichkeiten des Ansatzes werden bislang also nicht strukturiert erschlossen und Aufwand und Nutzen der Modellierung stehen nicht in einem angemessenen Verhältnis.
- **Die Integration in die Unternehmensprozesse wird nicht berücksichtigt:** Prozessuale Aspekte spielen bislang keine Rolle in den Methoden des MBSE [ERZ14]. Zuständigkeiten bei der Modellierung, die Nutzung der Modelle und die Anbindung an den Produktlebenszyklus sind jedoch essentiell – *“Scoping the model to meet its objectives [...] is essential to managing stakeholder expectations including those from program management, the customer and other member of the development team”* [FMS08]. Das zeigt auch, dass wichtige Stakeholder des interdisziplinären Ansatzes bislang ausgeschlossen sind [Zin13].

Die Darstellungen machen deutlich: MBSE wurde bislang aus einem rein technischen Blickwinkel getrieben. Das behindert die weitere Verbreitung und die Akzeptanz des Ansatzes. Vor diesem Hintergrund stellen sich die folgenden **Forschungsfragen**:

- **Wie sieht ein Konzept des MBSE aus, das die vielfältigen Facetten der Produktentstehung unterstützt?** Es ist zu beschreiben, was Kern des MBSE ist und wie der Ansatz für verschiedene Aufgaben und Ziele handhabbar wird. Dadurch soll es gelingen, die Sichtweisen und Ziele der Fachdisziplinen geeignet zu unterscheiden und zu kommunizieren.
- **Wie kann MBSE im Sinne des klassischen Systems Engineering in die Produktentstehungsarbeit integriert werden?** Dabei gilt es, Wege für die einfache und rationale Integration des Ansatzes in die Organisation bereitzustellen. Insbesondere muss das MBSE den zahlreichen bislang vernachlässigten Stakeholdern zugänglich werden – vor allem auch so, dass grade kleine Unternehmen einen Nutzen aus dem Ansatz ziehen.
- **Wie können existierende Methoden und Aufgaben aus der Produktentstehung systematisch mit MBSE unterstützt werden?** Modellierung ist kein Selbstzweck – vielmehr müssen die entstehenden Modelle für das Lösen eines konkreten Problems oder zur Unterstützung einer konkreten Aufgabe eingesetzt werden und über die zu eng gespannten Grenzen der rein technischen Anwen-

derung hinausgehen. Dadurch soll es gelingen, das Nutzenpotential von MBSE für die gesamte Produktentstehungsarbeit und im Lebenszyklus zu erschließen.

Aus der geschilderten Problematik im Zusammenspiel mit den Forschungsfragen ergibt sich der **Bedarf** für ein *Rahmenwerk zur Integration des modellbasierten Systems Engineerings in die Produktentstehung mechatronischer Systeme*. Das Rahmenwerk soll bei der zielgerichteten und zweckorientierten Anwendung des MBSE unterstützen und zunächst insb. die Tätigkeiten der bislang nicht angemessen berücksichtigten Stakeholder fokussieren. Deshalb richtet sich das Rahmenwerk neben ebendiesen insb. an Handlungspersonen mit Verantwortung für die Themenfelder Methoden und Prozesse sowie dem Change Management sofern sie MBSE zukünftig in die Prozesse ihres Unternehmens integrieren wollen. Hierzu können sie durch eine gezielte Berücksichtigung der einzelnen Elemente auch Anwender jenseits der zentralen Entwicklungsaktivitäten in das MBSE einbeziehen, also bspw. Produktmanager, Einkäufer, Qualitätsmanager – aber je nach Auslegung der Rolle (vgl. [She96]) natürlich auch den Systems Engineer. Durch diese Ausdehnung auf Aktivitäten jenseits der reinen Produktspezifikation soll die individuelle und organisatorische Akzeptanz für das Thema MBSE erhöht werden. Gleichzeitig wird dem Anspruch des Paradigmas als Dreh- und Angelpunkt der Produktentstehung Rechnung getragen, da es möglich wird, alle Aspekte des Systems Engineerings zu integrieren. Um dieses Ziel zu erreichen, soll das Rahmenwerk aufbauend auf den Erkenntnissen der Problemanalyse die folgenden Bestandteile umfassen:

- **Beschreibungsmodell „MBSE-Konzepte“:** Kern des Rahmenwerks muss eine Aufbereitung der bislang nur textuell beschriebenen Idee des MBSE sein. Es gilt herauszuarbeiten, was die wesentlichen Merkmale des MBSE sind und wie diese kombiniert werden können. Das soll einen Überblick zum Inhalt, zur Reichweite und zum Umfang des MBSE ermöglichen und ist Grundlage für eine zielgerichtete Anwendung im Produktlebenszyklus. Grundannahme ist, dass es im MBSE vergleichbar zum Systems Engineering mehrere Konzepte gibt.
- **Entscheidungsunterstützung zur Auswahl geeigneter MBSE-Konzepte:** Aufbauend auf dem Beschreibungsmodell soll die Einbindung von MBSE in die Produktentstehungsarbeit unter verschiedenen Randbedingungen gelingen. Dabei soll die Entscheidung ohne aufwändige Analysen möglich sein.
- **Konzept zur Nutzung des Systemmodells in der Produktentstehungsarbeit:** Systems Engineering umfasst technische Aktivitäten und Managementaktivitäten [HWF+12]. Die Aktivitäten greifen auf Methoden zurück, deren Eingangsinformationen weitgehend im Systemmodell vorhanden sind, bzw. mehrwertstiftend durch das Systemmodell einsetzbar sind. Das Konzept soll beschreiben, wie eine „Kopplung von Methoden“ an das Systemmodell und damit auch an die Prozesse der Produktentstehung gelingen kann.

- **Werkzeugkasten zur Modellierungsplanung, Systemmodellierung und Methodenkopplung:** Mit einem Set an Vorgehensweisen und Hilfsmitteln soll die Erstellung des Systemmodells und insb. die Anwendung von Methoden aus dem Management der Produktentstehung auf Basis des Systemmodells unterstützt werden. Das beinhaltet insb. die Planung des Modellierungsprojekts und der Modellierung inkl. der Modellinhalte zur Durchfügung einer Methode.

2.7 Anforderungen

Auf Grundlage der Problemanalyse und der Zielsetzung werden die folgenden Anforderungen an das *Rahmenwerk* gestellt:

- A1) Systems Engineering als Basis:** SE wird als das Paradigma der Produktentstehung von morgen behandelt. Die Grundbegriffe und Grundprinzipien dieses disziplinübergreifenden Ansatzes finden sich in der System- und Modelltheorie. Das sind bspw. die Orientierung am Modellbegriff und die Konzepte von HABERFELLNER. Im Laufe der Zeit wurde das Themenfeld jedoch von vielen Fachdisziplinen adressiert, die in unterschiedlichen Denkwelten beheimatet sind. Die Idee des SE wurde dabei unterschiedlich interpretiert. Das Rahmenwerk muss die Bandbreite des klassischen SE und seiner Konzepte ausfüllen und dabei nicht nur auf einzelne Aspekte fokussieren. Einschlägige Richtlinien wie die ISO/IEC15288 sollen dabei ebenso wie die o.g. Konzepte richtungsweisend sein. Insb. sollen die Managementaspekte des SE stärker in den Vordergrund rücken.
- A2) Systemmodell als Mittelpunkt sämtlicher Aktivitäten:** MBSE muss als Weiterentwicklung des Systems Engineerings verstanden werden. Das Systemmodell soll im Mittelpunkt sämtlicher Aufgaben und Tätigkeiten stehen – so die Vision. Das Systemmodell muss deshalb über die Informationen der reinen Architekturbeschreibung hinausreichen, um auch für relevante Methoden nutzenstiftend zu sein. Es ist im Rahmen dieser Arbeit aber nicht mit dem PLM-Ansatz in Verbindung zu bringen. Dieser soll keine Berücksichtigung finden, weder als Strategie noch als konkrete Werkzeugumsetzung.
- A3) Klares MBSE-Konzept:** Die Vision des MBSE ist eine vollständig und durchgängig virtualisierte Arbeitsumgebung. Darüber hinaus existieren zahlreiche Stimmen, die ein auf die jeweilige Situation, Aufgabe und Fähigkeiten der Anwender angepasstes Konzept fordern – weshalb mehrere verschiedene MBSE-Konzepte durchaus realistisch sind. Hier soll eine klare Beschreibung des MBSE entstehen, die eine eindeutige Kommunikation über das Thema ermöglicht. Dabei sollen insb. die wesentlichen Merkmale dieser Konzepte im Sinne des ganzheitlichen, lebenszyklusbegleitenden Ansatzes herausgearbeitet werden.
- A4) Sprachen- und Methodenunabhängigkeit:** Es existiert eine große Anzahl an Sprachen und Methoden im MBSE. Es ist davon auszugehen, dass dies berech-

tigt ist – bspw. für die Anwendung in unterschiedlichen Branchen. Aus diesem Grund ist es umso wichtiger, die Bestandteile des Rahmenwerks so auszulegen, dass sie für beliebige Sprachen und Methoden anwendbar sind.

- A5) Integration von MBSE in die Organisation und Prozesse:** Im klassischen SE wird der Systems Engineer als Verantwortlicher für sämtliche SE-Aktivitäten gesehen – mit unterschiedlichsten Rollenkonzepten. Im MBSE wird die Modellierung auf nahezu sämtliche Stakeholder verteilt (vgl. Kapitel 2.4.5). Es ist allerdings unklar, wie die Erstellung und Pflege des Systemmodells organisatorisch gelingen soll. Aus diesem Grund müssen konkrete Vorschläge zur Integration der Modellierungsaktivitäten und Verantwortlichkeiten dargestellt werden.
- A6) Adaption von Erfolgsprinzipien bekannter Managementansätze:** Die methodische Durchdringung der Produktentstehungsarbeit ist gering. In der Produktion haben einige Ansätze aber große Erfolge erzielt, bspw. Six Sigma. Diese zeichnen sich durch drei Eigenschaften aus: Sie zeigen eine Vision auf, bestehen aus vielen einfachen Hilfsmitteln, die zusammen und allein eingesetzt werden können und beliebig erweiterbar sind. Diese drei Eigenschaften sollen sich ebenfalls im Rahmenwerk widerspiegeln.
- A7) Rationalitätssicherungsfunktion:** Viele Entscheidungen werden intuitiv getroffen. Vor dem Hintergrund maßgeschneiderter Anwendungen des MBSE sollten Entscheidungen transparent und nachvollziehbar getroffen werden. Im Sinne des Management-Controllings geht es darum, das Spannungsfeld zwischen Intuition und Reflexion anzunähern. Damit ist zunächst nicht die Entscheidung als solche das Ziel – bspw. die Auswahl eines Werkzeugs zur Modellierung – sondern die entsprechende Unterstützung des Entscheidungsprozesses. Das Rahmenwerk soll Entscheidungen durch Hilfsmittel oder Beispiele unterstützen.
- A8) Stakeholderorientierte Modellierung und Nutzung des Systemmodells:** Bislang sind die Systemspezifikation und die Prinziplösung die zentralen Modellierungszwecke. Wenn das Systemmodell im Sinne von Anforderung A2 Dreh- und Angelpunkt der Produktentstehungsarbeit sein soll, dann sind die Arbeiten der Stakeholder durch das Systemmodell explizit zu unterstützen. Dazu muss das Modell entsprechend ihrer Zwecke gestaltet werden. Es soll dazu befähigen, einen Modellierungszweck zu identifizieren, die dazu notwendigen Informationen zielgerichtet zu identifizieren und auch entsprechend umzusetzen. Damit steht diese Anforderung im Einklang mit den Grundsätzen Relevanz und systematischer Aufbau der Grundsätze ordnungsmäßiger Modellierung.
- A9) Wohlfundierte Modellplanung:** Bei der Modellierung und Anwendung des Systemmodells gilt es, eine gründliche Planung und Klärung der eigentlichen Aufgabe vorzunehmen. Dedizierte Methoden und Hilfsmittel sollen hierfür zur Verfügung stehen und bei der Definition des Modellierungszwecks und der Modellplanung unterstützen. Dadurch kann Zeit und Aufwand eingespart werden.

3 Stand der Technik

In diesem Kapitel wird der Stand der Technik untersucht, um den in Kapitel 2 adressierten Handlungsbedarf zu bestätigen. Die junge Disziplin des MBSE erfordert eine thematisch breit gefächerte Analyse, die technische und organisatorische Aspekte aufgreift. Hierzu werden fünf Untersuchungsfelder abgesteckt und anhand bekannter Ansätze untersucht. Zunächst werden in Kapitel 3.1 etablierte Beschreibungsmodelle der Produktentstehung analysiert. Die ausgewählten Modelle haben den Anspruch, Systems Engineering zu befähigen und propagieren teilweise die Einbindung des modellbasierten Systems Engineerings. In Kapitel 3.2 werden verschiedene Systemmodellierungsmethoden analysiert. Hierbei wird insb. berücksichtigt, wie sehr die Nutzung der Modelle für Aufgaben im Lebenszyklus im Mittelpunkt steht. Daher wird auch betrachtet, wie die Planung des Modellierungsprojekts und insb. die Definition des Modellierungszwecks angegangen wird. Unter dem gleichen Blickwinkel werden die sog. Architecture Frameworks (AF) in Kapitel 3.3 analysiert. Die Einbindung in die Organisation zur individuellen Erfüllung von Aufgaben der Produktentstehung wird durch die Handlungsfelder Prozess- und Methodenmodelle (Kapitel 3.4) und Organisationskonzepte im SE (Kapitel 3.5) ergänzt. Mit den in den Kapitel 3.2 bis Kapitel 3.5 analysierten Arbeiten wird überprüft, wie MBSE bislang zur Unterstützung der typischen Aufgaben der Produktentstehung genutzt wird; ebenso sollen Implikationen zur Gestaltung des Rahmens gewonnen werden. Aus der Bewertung des Stands der Technik wird in Kapitel 3.6 der Handlungsbedarf bestätigt.

3.1 Beschreibungs- / Vorgehensmodelle der Produktentstehung

„MBSE is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases” [INC07]

Auf diese Definition beziehen sich nahezu alle Beiträge des Handlungsfeldes. Sie lässt allerdings viel Spielraum, wie MBSE ausgestaltet sein kann, wie und wofür es als Unterstützung eingesetzt wird und wie es in den Produktentstehungsprozess und den Produktlebenszyklus eingebunden wird. Dass MBSE als Weiterentwicklung des Systems Engineerings betrachtet wird, klingt in der Definition ebenfalls nicht an. Jüngere Veröffentlichungen von INCOSE kritisieren bereits die sehr starke Fokussierung auf die reine Entwurfstätigkeit [RWD14], [SF14]. In diesem Kapitel werden aktuelle Beschreibungsmodelle der Produktentstehung analysiert. Es wird insb. überprüft, ob sie ein klares Verständnis von MBSE vertreten und wie sie die Integration von MBSE in die Unternehmensprozesse und die Anwendung im Produktentstehungsprozess vor dem Hintergrund der in Kapitel 2 dargestellten Problematik adressieren.

3.1.1 3-Zyklen-Modell der Produktentstehung

Das 3-Zyklen-Modell der Produktentstehung beruht auf dem Konzept von [GLR+00] und beschreibt den Prozess von der Produkt- bzw. Geschäftsidee bis zum Serienanlauf mit den drei Hauptaufgabenbereichen Strategische Produkt- und Technologieplanung, Produktentwicklung sowie Produktionsprozess- und Produktionssystementwicklung [GPW14]. Dieser Prozess ist ein Wechselspiel von Aufgaben, die sich in drei – mit den Hauptaufgabenbereichen namensgleichen – Zyklen gliedern lassen (vgl. Bild 3-1). Ein leistungsfähiges Datenmanagement soll als Unterbau für die Anwendung dienen.

Der erste Zyklus charakterisiert das Vorgehen vom Finden der Erfolgspotentiale der Zukunft über die Erarbeitung eines erfolgversprechenden Geschäftsmodells bis zur Produkt- und Produktionssystemkonzeption (Prinziplösung des betrachteten Gesamtsystems). Er umfasst die Aufgabenbereiche Potentialfindung, Geschäftsplanung, Produktfindung, Produktkonzipierung sowie Produktionssystemkonzipierung. Der zweite Zyklus startet mit der Produktkonzeption und umfasst auch den fachdisziplinspezifischen Entwurf und die entsprechende Ausarbeitung sowie die Integration der Ergebnisse der einzelnen Fachdisziplinen zu einer Gesamtlösung. Die Produktkonzipierung verbindet also die beiden ersten Zyklen. Ausgangspunkt des dritten Zyklus bildet die Konzeption des Produktionssystems. Dabei sind die vier Aufgaben Arbeitsablaufplanung, Arbeitsmittelplanung, Arbeitsstättenplanung und Produktionslogistik, insb. Materialflussplanung integrativ zu betrachten und im Verlauf des Zyklus zu konkretisieren.

Von der Geschäftsidee...

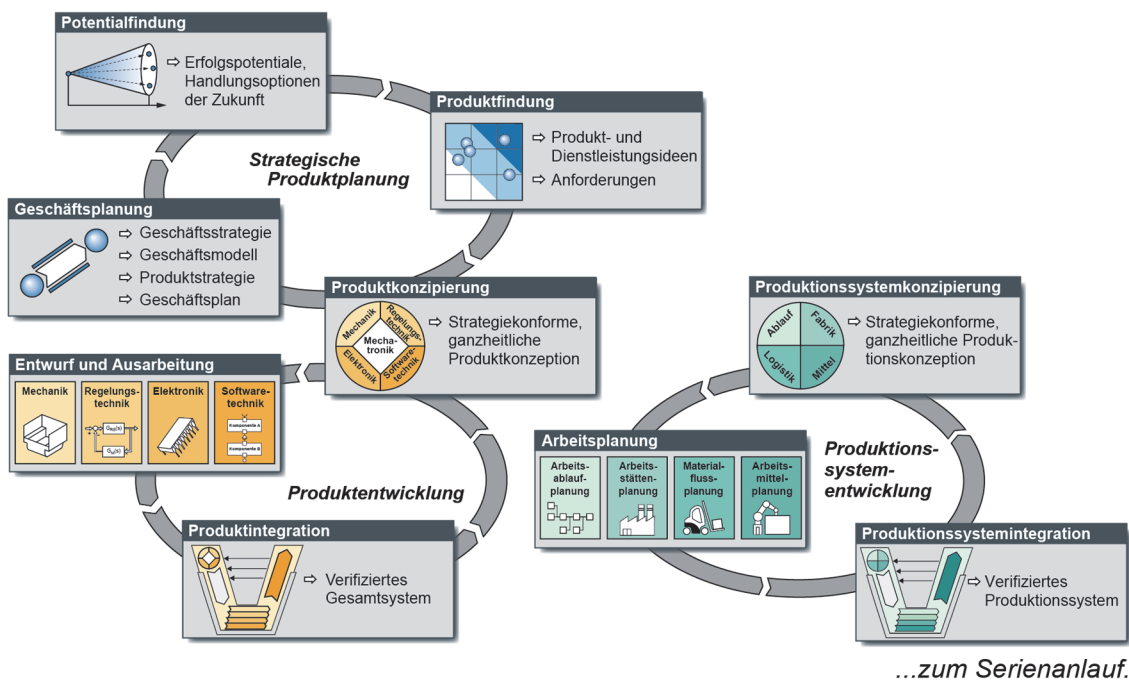


Bild 3-1 Das 3-Zyklen-Modell der Produktentstehung

Der Produkt- und Produktionssystemzyklus sind eng aufeinander abgestimmt voranzutreiben, um sicherzugehen, dass auch alle Möglichkeiten der Gestaltung eines leistungs-

fähigen und kostengünstigen Erzeugnisses ausgeschöpft werden, das die gestellten Anforderungen erfüllt [GLL12]. Aus diesem Grund werden das Produkt- und das Produktionssystemkonzept mit Hilfe fachdisziplinübergreifender Beschreibungsmittel integrativ modelliert. Ergebnis ist die Prinziplösung des Produkts und des dazugehörigen Produktionssystems, welche die Basis für die Kommunikation und Koordination der Fachleute bildet und Ausgangspunkt für die fachdisziplinspezifischen Gewerke bildet. Hierzu wurde die Spezifikationstechnik CONSENS (CONceptual design Specification technique for the ENgineering of complex Systems) entwickelt.

Bewertung: Das 3-Zyklen-Modell ist ein deskriptives Modell der Produktentstehung, das in seiner Ausgestaltung indirekt die Konzepte der Systemtechnik berücksichtigt. Es adressiert die Aufgaben der Produktentstehung von der Potentialfindung bis zum Serienanlauf und deckt die Technischen Prozesse der ISO15288 ab. Das beeinflusst auch den Fokus der vorgeschlagenen fachdisziplinübergreifenden Spezifikationstechnik CONSENS, die als Zweck die Prinziplösung von Produkt und Produktionssystem verfolgt: Sie verbindet alle drei Zyklen, hat ihre Schwerpunkte allerdings im zweiten und dritten Zyklus – wenngleich sie auch stärker in den ersten Zyklus integriert werden könnte. Die Anwendung und Integration von Methoden und Werkzeugen in das 3-Zyklen-Modell wird skizziert. Insgesamt bietet das Modell ein gutes systemisches Verständnis der Produktentstehung, hat aber nicht den Anspruch, auch die Organisation der Produktentstehung zu gestalten.

3.1.2 iPeM – Integriertes Produktentstehungsmodell

Die Grundidee des iPeM – integriertes Produktentstehungsmodell liegt in einer systemischen Betrachtungsweise der Produktentstehung, die die Arbeiten von ROPOHL zur Systemtheorie der Technik [Rop75] aufgreift und die Produktentstehung in die drei Teilsysteme Zielsystem, Handlungssystem und Objektsystem unterteilt (vgl. Kap 2.3.2, Bild 3-2). Das Handlungssystem generiert die Ziele des Zielsystems und transformiert diese in Objekte des Objektsystems. Diese Transformation gelingt im Handlungssystem durch verschiedene Aktivitäten und zugeordnete Ressourcen, wie bspw. Mitarbeiter, Methoden oder IT-Werkzeuge. Das Objektsystem besteht aus den Ergebnissen der Aktivitäten, den Objekten. Dies sind bspw. Modelle, Ergebnisse von Simulationen und von Validierungsversuchen, aber auch mindestens ein marktfähiges Produkt. Im iPeM steht nicht die Bereitstellung eines allgemeingültigen Modells im Vordergrund. Vielmehr soll der Prozess zur Erstellung zweckmäßiger Modelle für jeweils einzigartige Produktentstehungsprozesse bestmöglich unterstützt werden [Alb10]. Die Prozessmodellierung im Handlungssystem des iPeM ist dafür in eine logische und eine zeitliche Sicht unterteilt. Die logische Sicht wird gebildet durch die Aktivitäten der Produktentstehung und die Aktivitäten des Problemlösungsansatzes SPALTEN (Situationsanalyse, Problemeingrenzung, Alternative Lösungssuche, Lösungsauswahl, Tragweitenanalyse, Entscheiden und Umsetzen, Nachbereiten und Lernen), die zusammen eine Aktivitätenmatrix aufspannen. Diese Matrix bildet einen universellen Rahmen für die Modellierung von Pro-

duktentstehungsprozessen [AB11]. Dadurch, dass im Phasenmodell die Aktivitäten in einen zeitlichen Bezug zueinander gesetzt werden, ermöglicht das iPeM die Abbildung von branchen-, unternehmens- oder produktspezifischen Referenzmodellen sowie von Implementierungs- und Anwendungsmodellen.

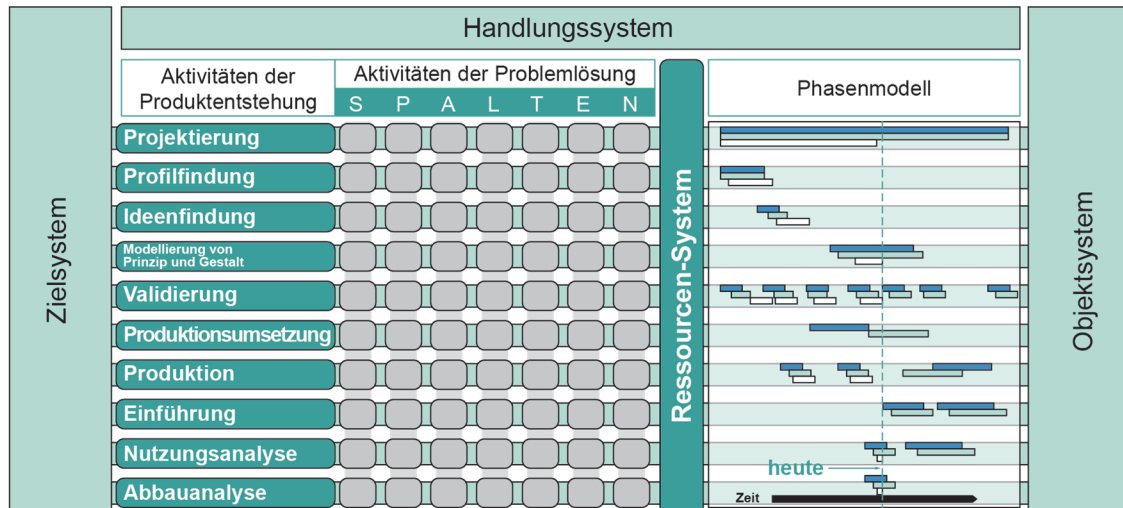


Bild 3-2 Integriertes Produktentstehungsmodell – iPeM

Bewertung: Das iPeM überwindet ein phasenorientiertes Denken und hat gleichzeitig einen deskriptiven und einen präskriptiven Charakter. Hervorzuheben ist, dass das iPeM die Produktentstehung als sozio-technisches System versteht, das nicht nur die Perspektiven von Entwicklern und ihre Aufgaben aufgreift, sondern auch Projektmanagern einen entsprechenden Rahmen für die Produktentstehungsarbeit bietet. Das unterstreicht den Bezug zu den Konzepten der System- und Modelltheorie in besonderer Weise. Durch die allgemeine Problemlösungssystematik können Vorgehensmodelle und Methoden im Prinzip beliebig eingebunden werden, wenngleich dabei nicht das Systemmodell im Mittelpunkt sämtlicher Aktivitäten steht. Die Orchestrierung der Produktentstehungsarbeit wird stattdessen durch die Elemente des Handlungssystems gestaltet. Hierbei werden im Sinne eines Managementwerkzeugs einige kleinere Werkzeuge und Hilfsmittel bereitgestellt, wie z.B. das Phasenmodell, mit dem Projekt-Setup und -steuerung unterstützt werden soll. Gleichzeitig dient das Phasenmodell weit gefasst auch in gewissem Sinne der Rationalitätssicherung, da hierüber Wissen über vorangegangene Projekte bereitgestellt wird. Der besondere Charme liegt in der eindeutigen Darstellung der anzugehenden Tätigkeiten in der Aktivitätenmatrix und die hierfür gegebene problemlose Erweiterbarkeit.

3.1.3 Modellbasierte virtuelle Produktentwicklung nach EIGNER

Die Idee der virtuellen Produktentstehung von EIGNER [ERZ14] beschreibt im Prinzip das V-Modell nach VDI-Richtlinie 2206 und betont die besondere Bedeutung des linken Astes, also der Interdisziplinären Systementwicklung. Dazu wird in Ergänzung zum

klassischen V-Modell noch der Verweis auf den sog. RFLP-Ansatz vorgenommen; ein Produktlebenszyklus wird abgebildet, ohne jedoch eine exakte Zuordnung zum V-Modell vorzunehmen (vgl. Bild 3-3). Dieser Produktlebenszyklus umfasst die Phasen Anforderungsdefinition, Produktplanung, Entwicklung, Prozessplanung, Produktion, Betrieb und Recycling. Von Anfang an soll er durch ein entsprechendes PLM-Backbone unterstützt werden, was später auch eine durchgängige Modellbildung und -analyse umfassen soll. Der Ansatz ist insgesamt durch die PLM-Sicht geprägt. Das Modell adressiert sehr stark die interdisziplinäre Systementwicklung mit MBSE und fordert hier den Ansatz der SysML. Wie die Anwendung stattfinden soll wird nicht geklärt, wengleich drei Ebenen der Modellierung grob unterschieden werden:

- Modellbildung und Spezifikation: Ein System wird mit qualitativen Modellen beschrieben. Diese beinhalten Anforderungs-, Funktions- oder Systemstrukturen. Die Modelle sind beschreibend und können nicht simuliert werden.
- Modellbildung und erste Simulation: quantitative, simulierbare Modelle beziehen mehrere Disziplinen ein, z.B. multi-physikalische Simulationsmodelle.
- Disziplinspezifische Modellbildung: Das sind bspw. CAD-Modelle.

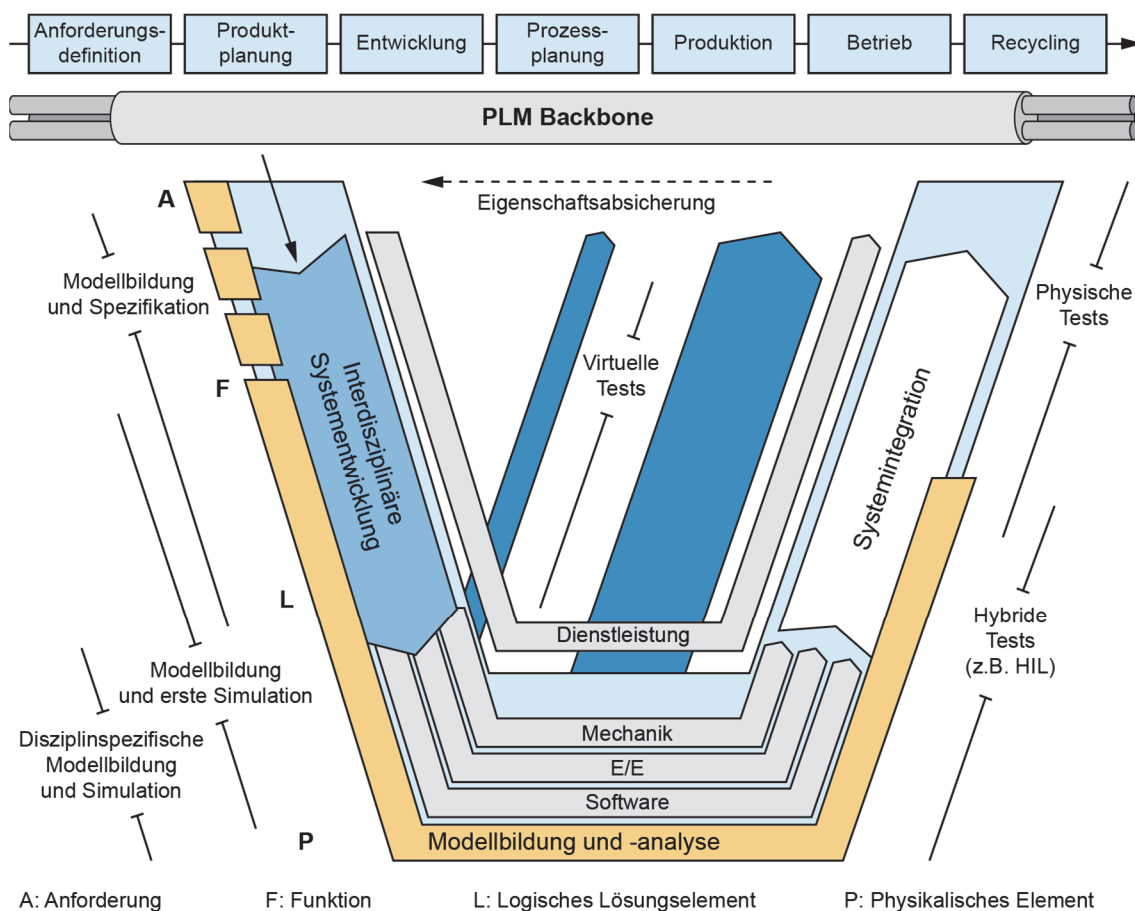


Bild 3-3 Virtuelle Produktentwicklung nach EIGNER

Bewertung: Das Modell von EIGNER bezieht sich durch seine Anleihe am V-Modell der VDI-Richtlinie 2206 indirekt auf grundlegende Konzepte der System- und Modelltheorie, was jedoch nicht explizit erwähnt wird. Vielmehr stellt es mit dem sog. PLM Backbone die PLM-Philosophie in den Mittelpunkt sämtlicher Aktivitäten. Das Systemmodell dient gemäß dieser vielmehr dazu, das Backbone aufzubauen. Wie der Ansatz mit dem Systemmodell als Dreh- und Angelpunkt ausgestaltet sein soll wird – wenn überhaupt – nur vage angedeutet. Zur Modellierung des Systemmodells wird die SysML als Mittel der Wahl dargestellt; wie die Arbeit mit der SysML oder weiteren Sprachen geplant und gestaltet werden kann, kommt nicht zum Ausdruck – erscheint jedoch realisierbar. Die vielfältigen Möglichkeiten einer solchen „PLM-Plattform“ werden skizziert; in Bezug auf die Operationalisierbarkeit existieren jedoch keine Ansätze, insb. nicht für das Management von Produktentstehungsprozessen.

3.1.4 W-Modell nach NATTERMANN/ANDERL

In der etablierten VDI-Richtlinie 2206 werden aufbauend auf einer Analyse der Anforderungen die Fachdisziplinen einzeln mit ihren spezifischen Arbeiten beauftragt. Diese werden weitgehend parallel und unabhängig voneinander bearbeitet. Daran schließt sich die Integration auf Subsystemebene und Systemebene. Nach wie vor führt dieses Vorgehen zu starken Integrationsproblemen. Hier setzt das W-Modell an [NA13]. Beginn und Ende des W-Modells und des V-Modells sind weitgehend identisch. Das W-Modell weicht in seinem Vorgehen jedoch gerade in der disziplinspezifischen Entwicklung stark vom V-Modell ab: Es findet eine Integration digitaler Modelle der Fachdisziplinen zur virtuellen Absicherung statt, wobei insb. die Simulation eine besondere Rolle spielt. Ergänzt wird das W-Modell durch ein spezielles Datenmanagement („DM“), das auch eine Steuerung der Entwicklungsarbeiten vorsieht und zudem in der Lage ist, die disziplinspezifischen Daten zu analysieren und disziplinübergreifend zu synchronisieren. Hierbei wird die Idee des Systemmodells aus dem MBSE aufgegriffen. Als Ergebnis liegt das validierte Gesamtsystem vor.

Wissenschaftliche Grundlage des W-Modells sind die Prinzipien des Systems Engineerings. Das W-Modell unterteilt sich in fünf Schritte, die auf drei Ebenen angeordnet sind: Systemebene, Subsystemebene und domänenspezifische Ebene (vgl. Bild 3-4). Die fünf Schritte des W-Modells sind im Einzelnen:

- Schritt 1 – Systemanalyse
- Schritt 2 – Analyse von Lösungen und Abhängigkeiten
- Schritt 3 – Virtuelle Systemintegration
- Schritt 4 – Modellanalyse und detaillierte Konstruktion
- Schritt 5 – Systemintegration:

Wie angedeutet ist das zentrale Element die Virtuelle Systemintegration in Schritt 3. Zunächst erfolgt jedoch die domänenspezifische Definition und Eingrenzung des Lösungsraums in Schritt 2 zzgl. einer Analyse der jeweiligen Lösungen hinsichtlich ihrer funktionalen Eigenschaften und der beeinflussenden konstruktiven Merkmale. In der virtuellen Integration werden dann die domänenspezifischen Lösungsräume in einem Systemlösungsraum zusammengebracht. Der Systemlösungsraum beschreibt die Gesamtheit der möglichen Kombinationen der domänenspezifischen Komponentenlösungen. Für jede Lösungsvariante werden generische Systemmodelle erstellt, die mit den globalen Anforderungen und Funktionen und den variantenspezifischen Elementen der Systemarchitektur verknüpft werden. Für die virtuelle Absicherung werden Simulationen des Systemverhaltens vorgeschlagen. Ebenso ist es möglich, Aussagen über den Reifegrad und die Anforderungserfüllung zu geben und hierauf aufbauend die domänenspezifische Konstruktion zu beginnen. Durch die Virtuelle Systemintegration kann die domänenspezifische Konstruktion in einem ganzheitlichen Systemkontext abgebildet werden, ohne eine manuelle Systemintegration der Fachdisziplinen durchzuführen.

Bewertung: Das W-Modell ist ein Vorgehensmodell, das in seinen Grundzügen auf dem Systems Engineering aufbaut und zusätzlich Vorgehensmodelle wie das V-Modell der VDI-Richtlinie 2206 sinnvoll erweitert. Die besondere Rolle der Virtuellen Systemintegration greift dabei in Richtung des MBSE. Jedoch werden diese Konzepte nicht explizit angesprochen. Das W-Modell ist vollkommen sprach- und methodenunabhängig. Als rein deskriptives Modell ist gibt es keine weiteren Hilfestellungen zur Operationalisierung.

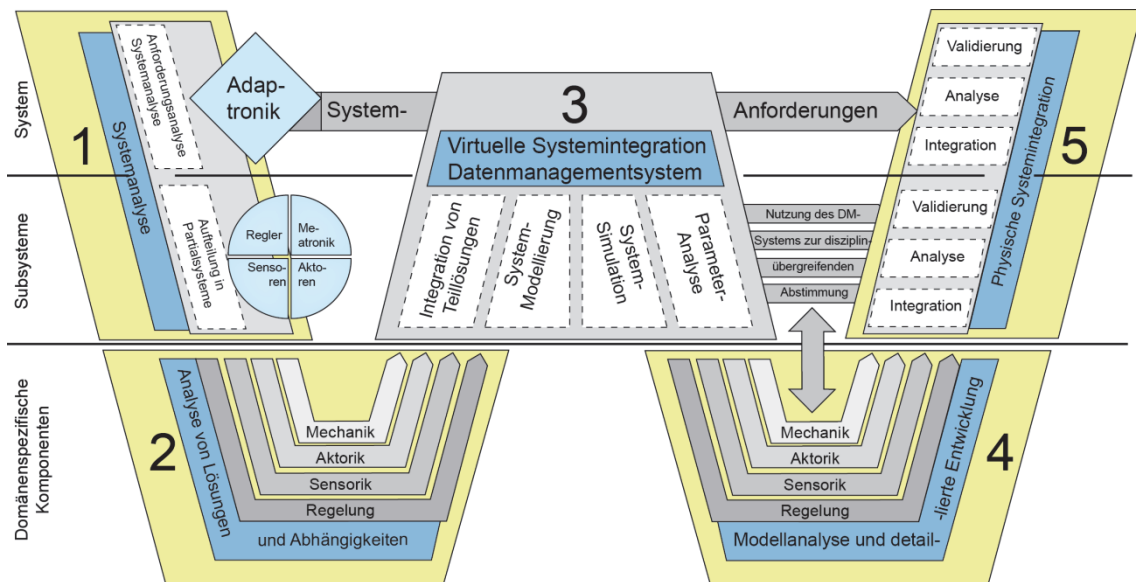


Bild 3-4 Das W-Modell nach [Nat14]

3.2 MBSE-Modellierungsmethoden

Ein wesentlicher Baustein der MBSE-Anwendung sind die Modellierungsmethoden. Ihr Ergebnis ist das Systemmodell – häufig auch als Systemspezifikation oder Prinziplösung bezeichnet. Während in der Fachwelt weitgehend Einigkeit herrscht, dass die Ansätze der objektorientierten Sprachen wie sie bspw. die SysML nutzt, Ausgangspunkt für die fachdisziplinübergreifende Systemmodellierung sind, herrscht bei den Modellierungsmethoden große Uneinigkeit. Die Folge: In kürzester Zeit ist ein unüberschaubarer Wildwuchs entstanden. In diesem Kapitel werden einige repräsentative Methoden auf ihre Eignung untersucht, ein Systemmodell zu erzeugen, das der Vision des lebenszyklusbegleitenden Ansatzes des MBSE gerecht wird. Dabei werden insb. auch Aspekte der Planung und Nutzung des Systemmodells für verschiedene Aufgaben berücksichtigt, was die Idee in den Mittelpunkt stellt, das Systemmodell als Dreh- und Angelpunkt sämtlicher Aufgaben im Produktentstehungsprozess zu nutzen.

3.2.1 CONSENS

CONSENS – CONceptual design Specification technique for the ENgineering of complex Systems ist eine Spezifikationstechnik für die fachdisziplinübergreifende Modellierung mechatronischer Systeme. Modellierungszweck ist die Prinziplösung. CONSENS wird insb. im Kontext des in Kapitel 3.1.1 diskutierten 3-Zyklenmodells propagiert. Genaugenommen besteht CONSENS aus einer Sprache und einer Methode. Beides baut auf den Arbeiten von KALLMEYER [Kal98] und FRANK [Fra06] auf. Als sprachliche Grundlage werden aber immer mehr die Konzepte der SysML aufgegriffen [IKD+13], so dass die Sprache CONSENS als Profil der SysML interpretiert werden kann. Im Rahmen dieser Analyse steht die Methode CONSENS im Vordergrund.

Bild 3-5 gibt einen Überblick über das Vorgehen nach CONSENS. Der Phasenablauf unterscheidet zwischen zwei Phasen: Analyse des Problems und Definition der Systemarchitektur. Das Vorgehen ist iterativ. Die Prinziplösung mechatronischer Systeme umfasst nach CONSENS sieben verschiedene Aspekte, die rechnerintern abgebildet ein kohärentes System von Partialmodellen ergeben.

Die Modellierung der Prinziplösung beginnt mit der Problemanalyse. Dazu gehören die Ermittlung und Darstellung der Interaktionen des Systems mit seiner Umwelt, das Auflisten von Anwendungsszenarien für das System, die Definition der Anforderungen an das System und das Erstellen einer Funktionshierarchie [GFD+08]. Während der Umfeldanalyse werden die Wechselwirkungen zwischen System und Umwelt mit Hilfe des „Black-Box“-Gedankens abstrakt ermittelt. Störgrößen werden dabei besonders hervorgehoben. Die Anwendungsszenarien stellen das Verhalten des Systems sowohl unter normalen Betriebsbedingungen als auch in Störfällen dar. Das ideale Verhalten des Systems als Reaktion auf die Szenarien wird als erster grober Lösungsansatz definiert. Die Anforderungsanalyse überführt die Anforderungen in funktionale und nicht-funktionale Anforderungen des Systems [GFD+09].

Das Ergebnis der Problemanalyse ist die Funktionshierarchie des Systems. Um von der Funktionshierarchie zu einer Systemarchitektur zu gelangen, sollen die Funktionen zunächst mit Hilfe des Morphologischen Kastens in die Wirkstruktur überführt werden. Hiervon ausgehend werden die weiteren Partialmodelle, die zusammen die Systemarchitektur bilden, erarbeitet. Die Systemarchitektur besteht zusammengenommen aus:

- der Analyse der Wechselwirkungen innerhalb des Systems und mit der Umwelt;
- der Bestimmung des Systemverhaltens;
- einer ersten Darstellung der Systemgestalt.

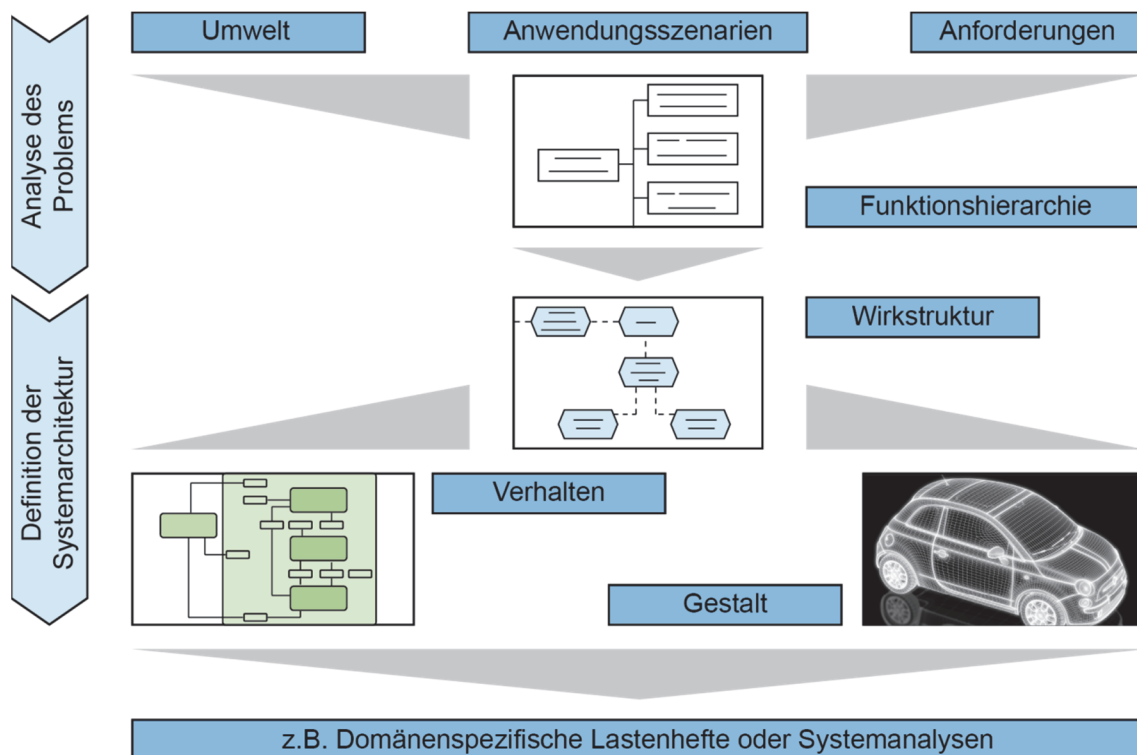


Bild 3-5 Aspekte und Vorgehen der Spezifikationstechnik CONSENS

Wie bei anderen Modellierungsmethoden auch, wird bei CONSENS von einem „Black-Box“-Prinzip ausgehend eine konkrete Darstellung der inneren Subsysteme als „White-Box“ erzeugt. Diese Subsysteme und ihre Wechselwirkungen werden in einer Wirkstruktur modelliert. Die Wechselwirkungen können auf physikalische Grundprinzipien (Energieerhaltungssatz, Massenerhaltungssatz) überprüft und korrigiert werden – ein Set an Regeln unterstützt die Modellierung einer plausiblen Systemstruktur als Summe aus Umfeldmodell und Wirkstruktur [Kai14]. Im Systemverhalten werden die Zustände des Systems festgehalten, aber auch diejenigen Systemaktivitäten, die Zustandsänderungen herbeiführen. Zuletzt kann die Gestalt der Systemarchitektur mit Hilfe eines ersten einfachen CAD-Modells oder Hüllkurven veranschaulicht werden.

Bewertung: CONSENS ist eine der wenigen bekannten MBSE-Modellierungsmethoden, die aus einem maschinenbaulichen Umfeld stammen. Das kommt insb. durch das Gestaltmodell zum Ausdruck. Von der Vorgehensweise erweitert CONSENS die mechanische Konstruktionsmethodik nach PAHL/BEITZ auf mechatronischer Systeme. Die klare Definition der Partialmodelle erlaubt dabei zunächst eine gute Orientierung für die Modellierung. Eigentlicher Modellierungszweck ist die Prinziplösung mechatronischer Systeme, was im Sinne des Systems Engineerings zu kurz greift. Dennoch wird die Skalierbarkeit des Ansatzes betont und meist die Bedeutung der Wirkstruktur hervorgehoben, da sie Grundlage für diverse Analysen in Kontext der Produkt- und Produktionssystemkonzipierung ist [Gau10a], [BC10] – bei diesen Analysen werden allerdings detailliertere Modelle genutzt, als für eine Prinziplösung üblich. Ein generelles Vorgehen und konkrete Anleitungen und Hilfestellungen zur Planung der Modellierung, Erzeugung und zur Nutzung der Partialmodelle und der möglichen Analysen existieren jedoch nicht – abgesehen von Modellierungsregeln zur Formalisierung der Wirkstruktur. Dennoch ermöglicht CONSENS einen pragmatischen Einsatz für unterschiedliche Zwecke. Die Bedeutung von MBSE als lebenszyklusbegleitender Ansatz wurde damit im Ansatz erkannt, eine stakeholderorientierte Modellierung und Nutzung des Systemmodells inkl. einer wohlfundierten Modellplanung wird aber nicht unterstützt. Ebenso bleibt offen, wie die Integration in die Organisation zur Erfüllung der Bedarfe der Stakeholder gelingen kann.

3.2.2 SYSMOD

SysMod – Systems Modeling Process ist eine Modellierungsmethode zur Anwendung von UML bzw. SysML [Wei15]. Ziel ist ein pragmatischer Ansatz zur Modellierung von Anforderungen, der funktionalen und physikalischen Architektur eines Systems. Beim Phasenablauf unterscheidet SysMod zwischen der Analyse und dem Design des Systems. Das Vorgehen unterteilt sich in mehrere Schritte: Anforderungen ermitteln, Systemkontext modellieren, Anwendungsfälle modellieren, Fachwissen modellieren, Glossar erstellen, Anwendungsfälle realisieren. In Ergänzung zu den einzelnen Modellierungsschritten existieren Hinweise und Leitfragen. Generell ist SysMod sehr flexibel und erlaubt unterschiedliche Detaillierungstiefen der Modelle [Mur12]. Zudem betont WEILKIENS, dass einzelne Schritte der Methode modifiziert oder vollständig ausgelassen werden können [Wei08] – wobei offen bleibt, unter welchen Umständen und mit welchem Zweck das geschehen kann. Auch wenn SysMod für beliebige Systeme einsetzbar sein soll, die Beschreibung von softwaretechnischen Systemen dominiert.

Die Analyse-Phase beginnt mit der Beschreibung des Projektkontexts. Dabei bedient sich SysMod der „Black-Box“-Darstellung: Zunächst wird eine Systemgrenze identifiziert und das Umfeld des Systems betrachtet. Danach werden die internen Subsysteme sequentiell und sukzessive erschlossen. In diesem Schritt werden u.a. die Ziele und Rahmenbedingungen in Textform festgelegt. Um die Anforderungen des Systems zu ermitteln, empfiehlt SysMod zunächst die Stakeholder des Systems zu identifizieren

und diese zu priorisieren. Diese Schritte werden jedoch methodisch nicht unterstützt. Danach werden die Anforderungen der Stakeholder erfasst und gewichtet.

Auf Basis des Projektkontextes und der Anforderungen wird der erste Ansatz des Systemkontextes als Strukturdiagramm modelliert. Danach fokussiert sich SysMod auf die Anwendungsfälle. Zunächst werden die Systemakteure identifiziert und dann die Interaktionen zwischen diesen Akteuren mit dem System erfasst, wobei insb. Interaktionspunkte betrachtet werden. Daraus leiten sich die Anwendungsfalldiagramme ab, die auf mögliche Redundanzen oder Konflikte überprüft werden. Akteure sind dabei keine Personen, sondern im Sinne eines mechatronischen Systems Aktoren. Sie stehen bei SysMod außerhalb der Systemgrenze. Die Ergebnisse der Analyse-Phase sind damit ein erhöhtes, domänenspezifisches Wissen, ein umfangreiches Verständnis für den Systemkontext und ein Glossar als einheitliche Kommunikationsbasis.

Während in der Analyse das System bezüglich der Informationsflüsse betrachtet wird, kommen in der Designphase nun die Objektflüsse hinzu. In der Designphase werden die Systemstruktur und Zustandsmodelle erstellt. Die Konsistenz der Anforderungen wird zwar durch die ständige Rückführung auf das Anwendungsfalldiagramm gesichert, die Anforderungen selbst werden zunächst jedoch nicht durch Anwendungsfälle ermittelt.

Bewertung: SysMod hat seine Wurzeln in der Softwaretechnik, wird mittlerweile aber als eine der führenden Modellierungsmethoden des Model-Based Systems Engineering wahrgenommen [Mur12]. Insgesamt wirkt die Methode durch zahlreiche Hilfsmittel sehr abgeschlossen. Zudem wird deutlich, dass SysMod klar auf die UML und die SysML zugeschnitten ist und in erster Linie die Systemspezifikation softwareintensiver Systeme adressiert – das wird an der Definition der Schnittstelle deutlich [Wei06]: „Die Schnittstelle definiert ein Verhalten mit einer Liste von Operationen.“ Damit wird die ganzheitliche Idee des SE bzw. des MBSE zu stark auf eine Fachdisziplin reduziert. Aus diesem Grund ist die Idee, das Systemmodell als Grundlage sämtlicher Aktivitäten im Prozess zu nutzen, nur schwierig realisierbar. Hervorzuheben ist, dass bei SysMod in leichten Grundzügen eine Modellplanung erkennbar ist und aufgrund der detaillierten Hilfsmittel die Eingliederung in die Organisation in gewissem Rahmen vorstellbar ist.

3.2.3 Harmony SE

Die Modellierungsmethode Harmony Systems Engineering (Harmony SE) ist Teil des übergeordneten Rational Integrated Systems/Embedded Software Development Process und explizit für die Entwicklung von Software mittels der Modellierungssprachen UML und SysML entwickelt worden [Hof08] Wie in Bild 3-6 dargestellt, deckt Harmony SE damit nur den linken Ast des gesamten Rational Prozesses ab, der wiederum einer Variante des V-Modells folgt [INC08]. Ziel ist die sog. System Architecture Baseline. Das ist eine Beschreibung des Systems, von der ausgehend die Spezifikation des Systems als Dokument generiert werden soll und weitere Analysen möglich werden. Das Besondere: Immer wenn sich Kundenwünsche (request) ändern, startet der Prozess erneut mit der

Anforderungsanalyse, der Begriff „request-driven“ bringt das zum Ausdruck und soll die Konsistenz des Repository garantieren [INC08].

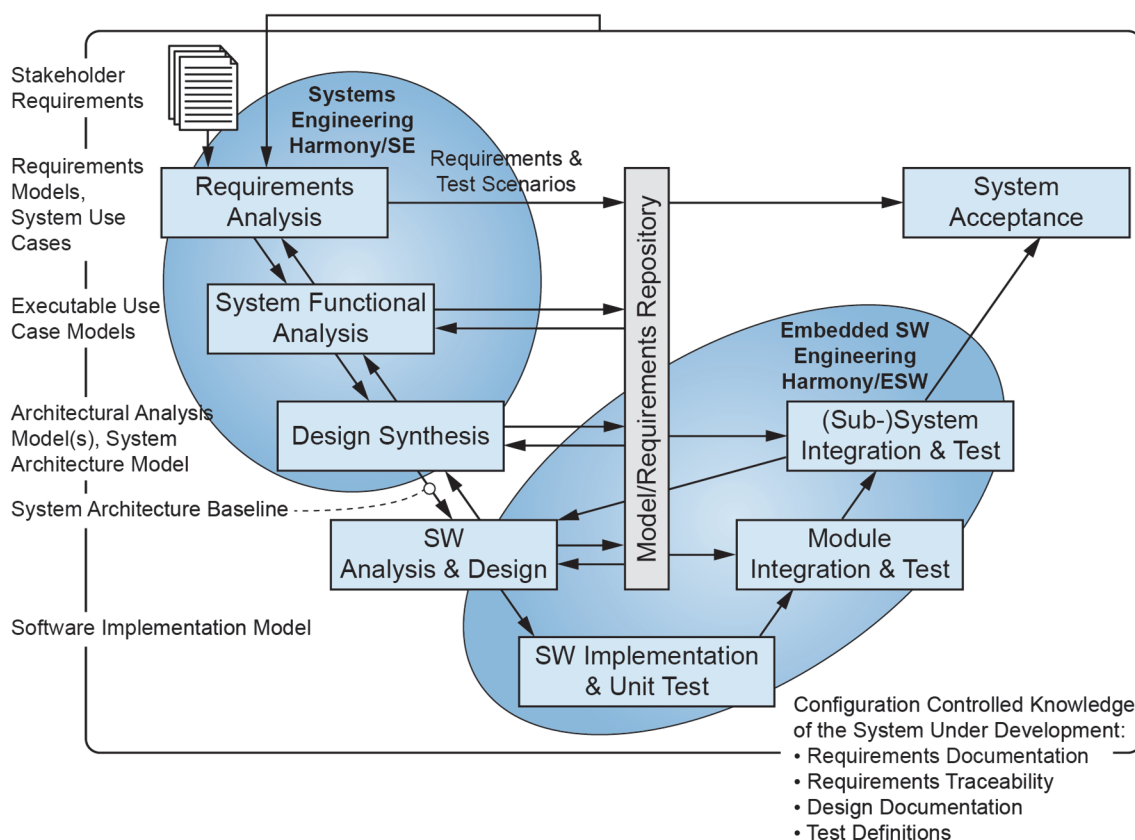


Bild 3-6 Harmony SE – Teil des Integrated System/Software Development Process

In der Anforderungsanalyse werden die Anforderungen der Stakeholder in die funktionalen und qualitativen Systemanforderungen übersetzt. Die Anwendungsfälle des Systems werden ermittelt und hierarchisch strukturiert. Harmony SE bedient sich dafür des Anwendungsfalldiagrammes von UML/SysML und ggf. des Anforderungsdiagrammes von SysML. Die Anforderungen werden zum Schluss priorisiert, s.d. die sich ergebende Reihenfolge den weiteren Ablauf des Vorgehens bestimmt. Die funktionalen Anforderungen an das System werden dann in der Funktionalen Systemanalyse in die Systemoperationen überführt. Die Anwendungsfalldiagramme werden den Systemoperationen entsprechend angepasst. Die Diagramme sind als „Black-Box“ erstellt, um eine lösungsneutrale Darstellung des Systems zu gewährleisten. Von den Anwendungsfällen wird das Systemverhalten bestimmt. Dadurch wird das Verhalten des Systems mit den Anforderungen verifiziert. In der Design-Synthese werden die abstrakten, funktionalen Anforderungen in konkrete Strukturen überführt: Aus den „Black-Boxes“ werden „White-Boxes“. Die Überführung in eine konkrete physikalische Struktur geschieht in den Subsystemen. Die Auswahl zwischen Alternativen orientiert sich an dem Erfüllungsgrad der vordefinierten Leistungskriterien – z.B. Measures of Effectiveness (MOEs).

Bewertung: Harmony SE wurde speziell für die Sprachen der Object Management Group entwickelt. Basierend auf den Modellierungssprachen UML und SysML kann Harmony SE somit eine große Zielgruppe ansprechen. Durch die explizite Darstellung des Repository in der Mitte des V-Modells ist die Idee zur Nutzung des Systemmodells als Dreh- und Angelpunkt in Ansätzen existent, jedoch wird seine Anwendung nicht dargelegt. Durch die Definition der Measures of Effectiveness (MoE) soll eine rationale Entwicklung ermöglicht werden. Für die eigentliche Modellierung existiert ein kleines Set an Richtlinien. Die individuelle Modellierung vor dem Hintergrund eines Concerns wird nicht dargestellt, ebenso nicht die Modellplanung.

3.2.4 OPM – Object-Process Methodology

Die Object-Process Methodology (OPM) ist eine Modellierungsmethode für die Entwicklung von Software-/Informationssystemen und ihre fortlaufende Pflege im Produktlebenszyklus. OPM ist die korrespondierende Methode zur Sprache OPD/OPL von DORI und dem Werkzeug OPCAT [Dor02]. OPM basiert auf der Idee, dass alles entweder Objekt oder Prozess ist und alle Objekte einen Zustand haben – deshalb wird OPM auch als objekt- und gleichzeitig prozessorientiert bezeichnet [DR03]. Mit den drei Elementen Objekt, Prozess und Zustand lassen sich nach DORI sämtliche Systeme zusammenhängend beschreiben: Prozesse wirken auf Objekte und können dabei den Zustand eines Objektes in andere Zustände überführen [INC08]. Um alle Zusammenhänge zwischen Objekten, Prozessen und Zuständen hervorzuheben, existiert in OPM nur ein einziges Modell [Dor02]. Die Methode selbst ist stark auf die eigene Software OPCAT zugeschnitten, was bspw. durch die sog. Mechanismen deutlich wird: Aufgrund der Reduktion auf ein einziges Modell werden die Mechanismen *In-/Out-Zooming*, *Ex-/Subpressing* und *Un-/Folding* definiert, die die Beherrschung der im Modell dargestellten Informationen unterstützen sollen [DR03].

Bild 3-7 gibt einen Überblick über das Vorgehen, die relevanten Stakeholder und die Entwicklungsartefakte der Modellierungsmethode OPM in der für diese Sprache und Methode üblichen Darstellungsweise, dem OPM-Ei:

- Die Phase „Spezifikation der Anforderungen“ beginnt mit der Problemdefinition durch den Systemarchitekten und den Auftraggeber. Das Ergebnis wird in einem Anforderungsdokument hinterlegt und kann teilweise mittels der Sprache OPL dargestellt werden. Danach werden die Anforderungen an das System identifiziert.
- Die zweite Phase von OPM befasst sich mit der Analyse und Design – Ziel ist das sog. *skeleton* als abstrakte Darstellung der Systemstruktur. Wenn die Anforderungen mit OPL erstellt wurden, können Compiler den Vorgang automatisieren.
- In der dritten Phase wird das System implementiert, d.h. das *skeleton* wird in seine vollständige Systemstruktur übersetzt. Hierbei wird die Softwareausrichtung deutlich, da in dieser Phase der Programmiercode erstellt wird.

- Die letzte Phase beinhaltet die Anwendung und Instandhaltung, worunter Anpassungen der Systemstruktur während des Betriebs fallen. Dadurch wird deutlich, dass die OPM letztlich nie abgeschlossen ist.

Bewertung: Mit Hilfe des OPM lassen sich laut Dori alle Arten von Systemen abbilden. In der Praxis liegen die Anwendungsfälle von OPM jedoch eindeutig in der Softwareentwicklung [LG04]. OPM ist einfach und intuitiv handhabbar [RFB13]. Eine detaillierte Modellierung ist aufgrund der geringen Anzahl verschiedener Konstrukte allerdings nicht möglich. Dadurch eignet sie sich zur Modellierung von Systemen auf hohem Aggregationsniveau. Insbesondere für Stakeholder mit geringer Modellierungserfahrung ist diese Sprache damit gut geeignet. Hervorzuheben ist die Lebenszyklusorientierung. Für Anwendungen und die Planung für Anwendungen jenseits der automatisierten Softwarecode-Erzeugung erscheint OPM allerdings nicht gerüstet.

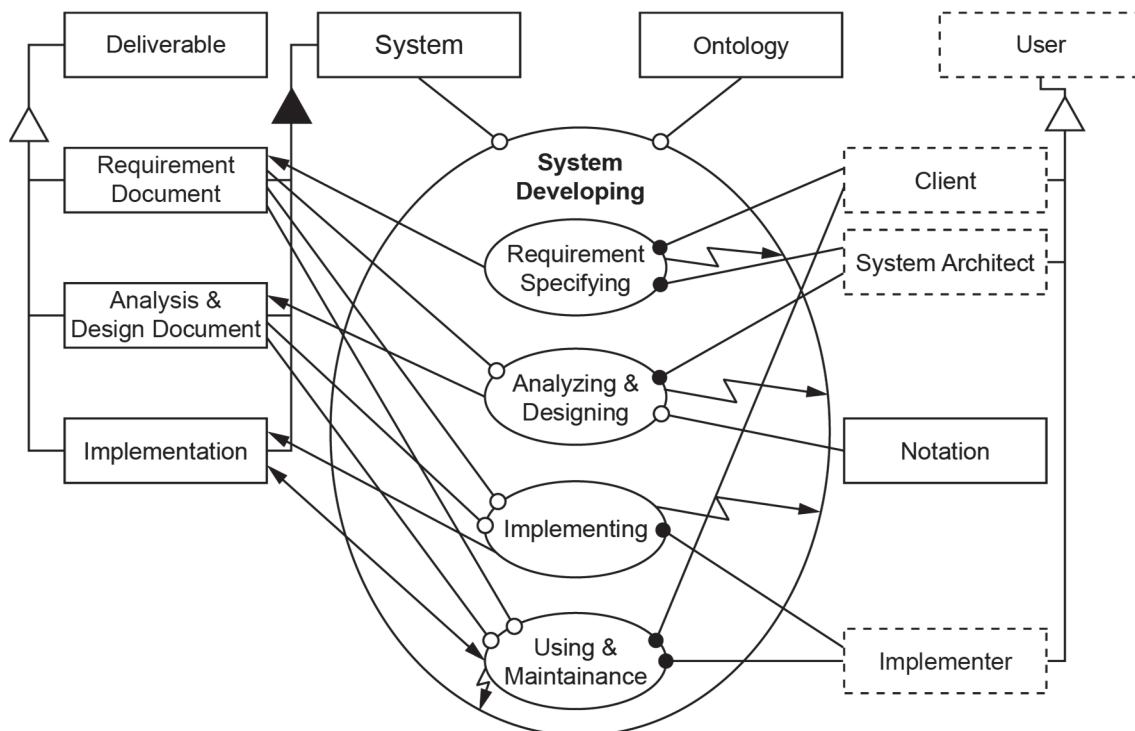


Bild 3-7 Das „OPM-Ei“ mit OPD-typischen Diagrammen

3.2.5 oosem

Die Object-Oriented Systems Engineering Method (oosem) ist eine Modellierungsmethode der Object Management Group. Sie bildet weitgehend die technischen Prozesse der [ISO15288] ab. oosem wurde mit dem Ziel entwickelt, die Integration von Hardware und Software Komponenten zu vereinfachen [Mur12]. Seit Einführung der SysML im Jahr 2006 unterstützt oosem zudem auch das Model-Based Systems Engineering [PH12]. Generell ist oosem ein Top Down-Ansatz. Die einzelnen Hauptphasen können sich teilweise überschneiden bzw. mit den unterstützenden Phasen gleichzeitig erfolgen.

Das grobe Vorgehen ist in Bild 3-8 dargestellt. Auf Seite der Hauptphasen steht zunächst die Bedarfsanalyse. Hier findet bspw. die Analyse der Stakeholder und ihrer Anforderungen statt. oosem nutzt dafür die Anwendungsfalldiagramme von UML bzw. SysML. In der System-Anforderungsanalyse werden die Stakeholder-Anforderungen in Systemanforderungen übersetzt. Dafür wird das System zunächst als eine „Black-Box“ betrachtet, d.h. alle internen Eigenschaften und Faktoren des Systems werden ausgeblendet. Danach werden die Wechselwirkungen zwischen dem System und seinem Umfeld ermittelt und dargestellt. Nur wenn alle relevanten Interaktionen korrekt ermittelt und dargestellt werden, kann die Konsistenz der Anforderungen garantiert werden. Da die Anforderungen auf den Anwendungsfällen basieren und die Konsistenz der funktionalen Anforderungen durch eine hohe Gewichtung der Anwendungsfälle des Systems gewährleistet wird, wird dieser Design-Ansatz als „usage-driven“ bezeichnet [NBB+08]. Aus den Anforderungen werden Funktionen abgeleitet. Nach dem Prinzip „vom Abstrakten zum Konkreten“ wird die funktionale Struktur des Systems in logische Strukturen überführt. Statt einer einzelnen logischen Struktur werden jedoch mehrere, gleichwertige Strukturalternativen erstellt. Durch diese Maßnahme wird eine lösungsneutrale Sichtweise des Systems lange Zeit forciert, um Standardlösungen zu hinterfragen und voreilige Schlüsse zu vermeiden. Von jeder logischen Struktur werden dann in der Synthese-Phase mehrere physikalische Strukturen ermittelt. In den physikalischen Strukturen des Systems wird jedes Element der logischen Struktur durch ein konkretes Bauteil ersetzt, sodass die gesamte physikalische Struktur am Ende ein mögliches Gesamtprodukt darstellt [PH12].

Neben diesen vier Hauptphasen existieren zwei unterstützende Phasen: Optimieren und Bewerten von Alternativen, sowie Validierung und Verifizierung des Systems. Zu letzterer zählen die Tests des Systems und der Anwendungsfälle, die mit Daten versorgt werden müssen. oosem fordert daher eine Identifizierung der „Measures of Effectiveness“ (MOEs), also eine Beurteilung der Leistungsfähigkeit des Systems in den Anwendungsfällen. Die MOEs werden als Elemente in ein Diagramm integriert und verkörpern alle relevanten Eigenschaften eines Leistungsindikators, wie z.B. Verfügbarkeit, Schnelligkeit, Sicherheit oder Kosten [Hau11]. Dadurch wird nicht nur ein Indikator über die Fähigkeit des Systems gewonnen, sondern es können auch die wichtigen – weil besonders kritischen – Anwendungsfälle identifiziert und optimiert werden.

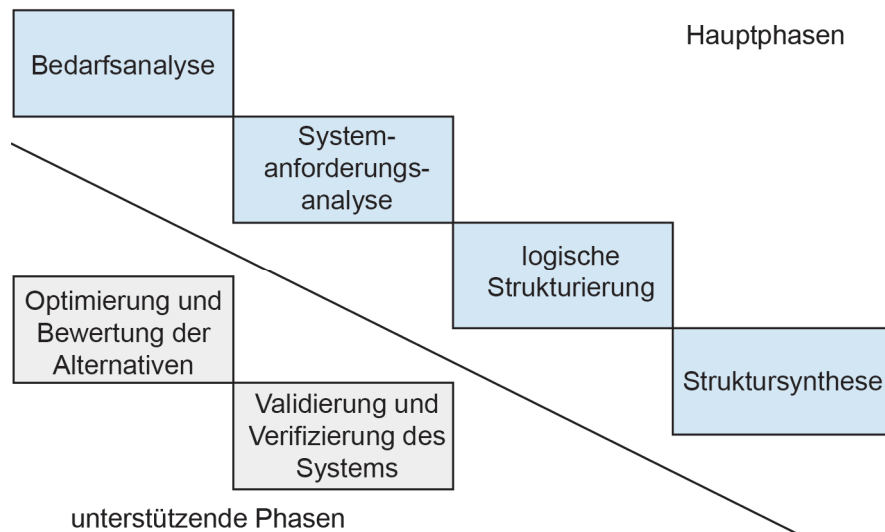


Bild 3-8 Hauptphasen und unterstützende Phasen in oosem

Bewertung: Obwohl oosem eine große Nähe zur UML und der SysML hat, ist es vom Vorgehen auch für andere Sprachen geeignet – nicht zuletzt, da es werkzeugneutral gehalten ist [PH12]. Durch die Verwendung der Standardsprachen und durch die Rücken- deckung der INCOSE hat oosem einen hohen Bekanntheitsgrad erlangt. Die Einbindung von MOE leistet einen Beitrag zur effektiven Produktentwicklung, was auch die Rele- vanz der Modellierung begünstigt. Die Aspekte der Rationalitätssicherung, Modellpla- nung und stakeholderorientierte Modellierung sind nicht explizit ausgeprägt.

3.2.6 LITHE

LITHE ist eine Modellierungsmethode des Model-Based Systems Engineering und wurde von RAMOS als Kombination der Methoden Harmony SE, OOSEM und OPM sowie dem generellen SE-Prozess SIMILAR [BG98] entwickelt [RFB10]. Ziel ist die agile Entwicklung des Systemmodells. Das sollen Prinzipien sein, die eine schnelle, fortlaufende Kommunikation zwischen allen Stakeholdern eines Systems über den ge- samten Produktlebenszyklus ermöglicht [Ram08], [RFB12b], [RFB13]. Das integrierte und ganzheitliche Systemmodell dient als Hauptartefakt der Entwicklungsumgebung.

Der LITHE Prozess ist an den SIMILAR Entwicklungsprozesses von BAHILL angelehnt und besteht aus vier Hauptphasen und zwei unterstützenden Phasen (Bild 3-9). LITHE wird sowohl iterativ als auch inkrementell durchschritten – vergleichbar mit dem Spi- ralmodell. Die vier Hauptphasen „Problem definieren“, „Alternative finden“, „Integrie- ren“ und „System starten“ werden durch einfache Methoden umgesetzt. In jeder Haupt- phase können Modelle mit den Modellierungssprachen SysML und OPDs/OPL oder anderen Darstellungsmöglichkeiten wie z.B. Prototypen erstellt werden. Diese Haupt- phasen werden durch die transversalen Prozesse *Assess performance* sowie *Re-evaluate* und *Model the System* ergänzt. Hier werden die einzelnen Hauptphasen ausgewertet, und ggf. Maßnahmen zur Anpassung durchgeführt. Dies deckt sich im Prinzip mit dem

Grundsatz der kontinuierlichen Verbesserung des Qualitätsmanagements [DGQ10]. Daneben gibt es noch die Messung der Leistung, die sich auf die in OOSEM vorgestellten Leistungsindikatoren (MOEs) stützt. Diese Leistungsindikatoren werden aus den Anforderungen des Systems übernommen und dienen schließlich als eine objektive Überwachung und Ermittlung von Verbesserungspotentialen [RFB13].

Zur Integration verschiedener Modellierungstypen in den Entwicklungsprozess schlägt RAMOS GRAPHITE vor (GRAPHICAL Tool for stakeholder's interaction). Es enthält Matrizen für die Phasen *State the problem*, *Investigate alternatives* und *Integrate* des SIMILAR-Prozesses. Dabei wird in den Zeilen die Modellierungsfähigkeit der Stakeholder dargestellt, in den Spalten die durchzuführenden Aktivitäten. Die Eintragungen schlagen vor, welche Modellierungselemente geeignet sind. Es werden Stakeholder mit und ohne Modellierungskennntnisse unterschieden. Das Tool wird vom Systems Engineer angewendet. Je nach Phase, Aktivität und Art des Stakeholders wählt der SE die zu verwendende Modellierungssprache und Modell aus. Die Matrizen können auf System-, Subsystem-, Komponenten- und Bauteil-Level angewendet werden [RFB13].

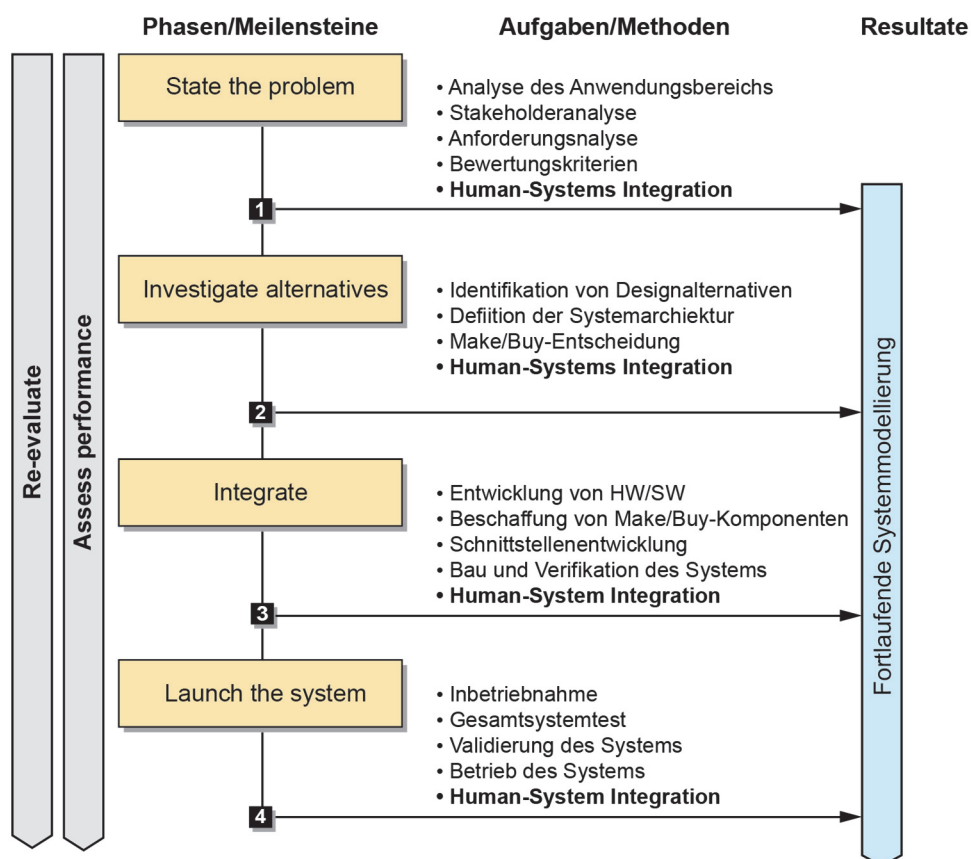


Bild 3-9 Der LITHE-Prozess, angelehnt an [RFB13]

Bewertung: Mit LITHE existiert ein sehr umfassendes Konzept zur Schaffung einer modellbasierten Entwicklungsumgebung. Es erhebt für sich den Anspruch, *lean* und

customizable zu sein und greift damit die Erfolgsprinzipien bekannter Managementansätze auf. LITHE will mittels einfacher Visualisierungstechniken im gesamten Produktlebenszyklus unterstützen und dabei auch die kontinuierliche Weiterentwicklung des Systemmodells als zentrales Element in der Produktentstehung forcieren. Hauptanwender ist ein Systems Engineer – je nach Fähigkeitslevel können mehrere Modellierungssprachen zur Erstellung des Systemmodells genutzt werden. LITHE als eine der jüngsten MBSE-Modellierungsmethoden geht mit vielen Ideen in Richtung eines ganzheitlichen Entwicklungskonzepts, das auf MBSE aufbaut. Wie das Konzept aber im Detail ausgestaltet ist, bleibt unklar. Dadurch kommen die Modellplanung („Analyse des Anwendungsbereichs“) und die Nutzung des Modells für Aufgaben im Projekt kurz. Die Integration in die Organisation wird ebenfalls nicht aufgegriffen.

3.3 Frameworks

Es existieren zahlreiche Architekturrahmenwerke – sie definieren meist die für eine Aufgabe zu erzeugenden Entwicklungsartefakte. Das standardisiert die Entwicklung und Dokumentation von Architekturen. Ein solches Rahmenwerk gibt jedoch weder die strukturelle Beschaffenheit des betrachteten Systems vor, noch macht es im Regelfall Vorgaben hinsichtlich geeigneter Hilfsmittel [Rot12]. Da hier aber die Architekturbeschreibung eines Systems im Mittelpunkt steht, sind hier grundsätzlich zahlreiche Analysen auf Basis dieser Architekturbeschreibung möglich.

3.3.1 ISO/IEC/IEEE42010

Der International Standard ISO/IEC/IEEE42010 – Systems and Software Engineering – Architecture Description definiert die grundlegenden Begriffe einer Systemarchitekturbeschreibung und ihre Zusammenhänge [ISO42010]. Unter einer Systemarchitektur versteht die ISO/IEC42010 die fundamentalen Konzepte oder Eigenschaften eines Systems, die in einer Architekturbeschreibung ausgedrückt und dokumentiert werden. Ziel der ISO/IEC42010 ist, die Entwicklung von Systemarchitekturen, ihre Nutzung für verschiedene Aufgaben und ihre Pflege auf Basis einer Architekturbeschreibung und eine eindeutige Nomenklatur zur erleichtern. Prozesse oder Methoden zur Erstellung solcher Architekturbeschreibungen werden nicht beschrieben.

Kern der ISO ist die in Bild 3-10 dargestellte Ontologie, deren Begriffe diskutiert werden – insb. der Begriff *Architekturbeschreibung*. Die o.g. Unterscheidung zwischen der Architektur und der Architekturbeschreibung ist essentiell; letztere ist lediglich ein Artefakt der Entwicklungsaktivitäten. Die Verwendung der Architekturbeschreibung ist abhängig vom Stakeholder und des Lebenszykluszeitpunkts. Mögliche Verwendungszwecke sind u.a. die Grundlage für die Systementwicklung, die Analyse und Bewertung von alternativen Architekturen, die Dokumentation der Entwicklung und/oder Instandhaltung, der Input für Werkzeuge (Simulationen, etc.) und generell die Grundlage für Bewertung, Analyse und Evaluierung des Systems über dessen Lebenszyklus.

Das Zusammenspiel zwischen Stakeholder und Architekturbeschreibung ist elementar: Die verschiedenen Stakeholder der Architekturbeschreibung haben unterschiedliche Interessen an dem in Entwicklung befindlichen System. Über die sog. Concerns sollen sie das Interesse auf Basis der Architekturbeschreibung befriedigen. Ein Concern ist definiert als „any topic of interest“, wird einem oder mehreren Stakeholdern zugeordnet und adressiert bspw. laut ISO die Funktionalität, die Struktur, das Verhalten o.ä. Zur Befriedigung eines Concerns wendet die ISO42010 das Konzept der „Separation of Concerns“ an – jeder Concern wird durch einen einzelnen View („Sicht“) befriedigt. Die Inhalte der Sicht beziehen sich auf den Concern, sind aber erst durch den Viewpoint bestimmt – er definiert die Konventionen zur Erstellung, Interpretation und Analyse des Views. Das sind bspw. Sprachen, Notationen, Modellart, Modellierungsmethoden oder Analysetechniken. Ein *Architecture Viewpoint* ist somit im Prinzip eine Methode zur Erstellung des konkreten Views – hier wird die ISO jedoch nicht konkret.

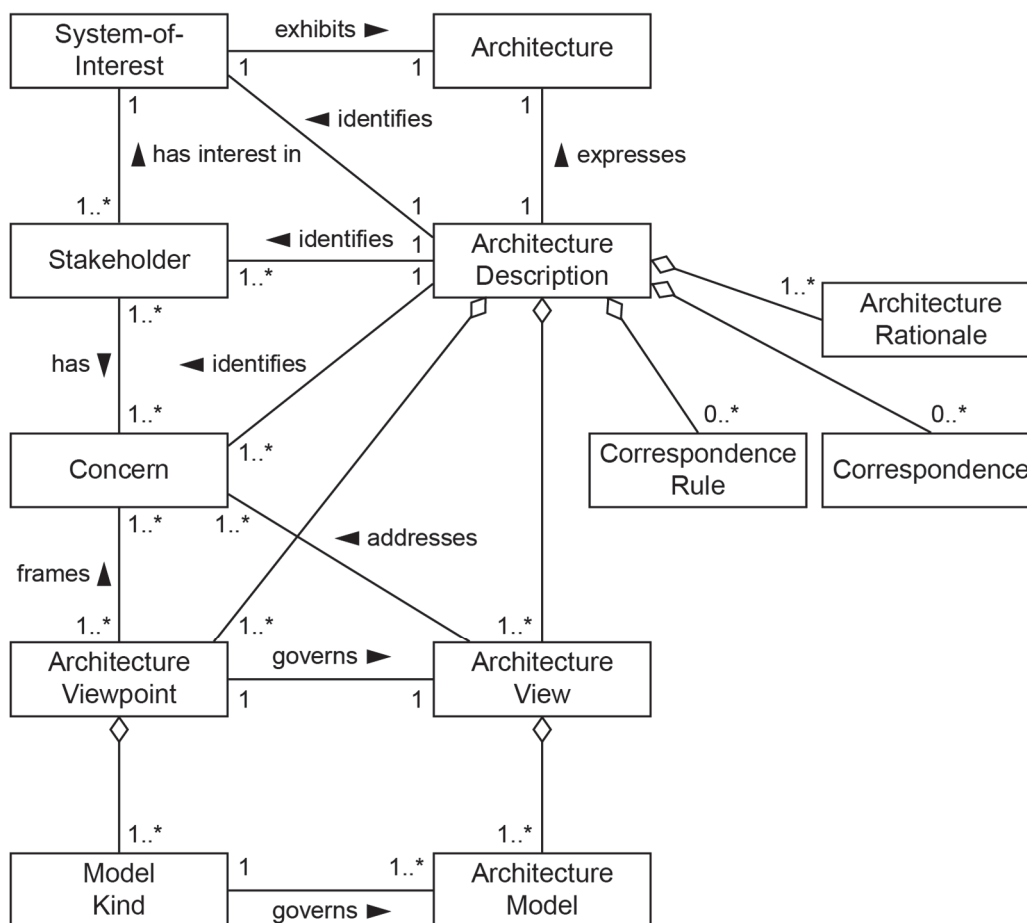


Bild 3-10 Grundkonzept der ISO/IEC/IEEE 42010

Bewertung: Die ISO/IEC/IEEE42010 liefert einen geschlossenen Rahmen zur Vereinheitlichung von Begriffen, die insb. in der Architekturentwicklungsphase eine wesentliche Rolle spielen. Dabei werden die Begriffe unabhängig von einer Modellierungssprache definiert und berufen sich aufgrund der Entstehungsgeschichte der Norm nicht

zwangsläufig auf die Idee des MBSE. Die ISO ist insgesamt generisch und beinhaltet deshalb gut übertragbare Konzepte. Diese müssen jedoch für die konkrete Anwendung des MBSE stark erweitert und ausdetailliert werden: So wird bspw. der Begriff Concern nicht ausreichend detailliert, ihr Auftreten und die Kopplung an den Lebenszyklus und die Produktentstehungsprozesse sind nicht geklärt. Sämtliche Aspekte zur Operationalisierung (Modellplanung, stakeholderorientierte Modellierung, ...) werden nicht berücksichtigt. Auch wenn die ISO/IEC42010 einen guten Ansatzpunkt zur Beschreibung des MBSE-Konzepts und seine Anwendung in Prozessen darstellt – konkrete Ansätze liefert sie nicht und die Kopplung mit dem MBSE ist nicht beschrieben.

3.3.2 DoDAF – Department of Defense Architecture Framework

Das Department of Defense Architecture Framework (DoDAF) ist ein Architekturrahmenwerk des US-amerikanischen Verteidigungsministeriums. Es bezieht sich ab Version 2.0 auf komplexe Systeme beliebiger Art, bei denen die Integration und Kollaboration verschiedener Disziplinen eine der zentralen Herausforderungen darstellen [DoDAF11] – vgl. Bild 3-11. Das DoDAF wurde entwickelt, um die Anforderungen des US-amerikanischen Verteidigungsministeriums Department of Defense (DoD) bei der Systementwicklung zu erfüllen. Die zahlreichen Lieferanten des Ministeriums sind so gezwungen, Systeme einheitlich zu beschreiben, was u.a. die Vergleichbarkeit der Angebote erhöht. Zusätzlich wurde in der Version 2.0 ein datenzentrierter Ansatz entwickelt – im Gegensatz zum bisher üblichen produktzentrierten Ansatz. Durch den Fokus auf Daten sollen für jeden Stakeholder die besonderen Informationen über eine Architektur durch Sichten realisiert werden können. Damit das Konzept realisierbar ist, hat das DoD passend zum Framework ein eigenes Metamodell entwickelt, das zudem durch ein Profil der Softwaremodellierungssprache UML verfügbar ist.

Die Nutzung des DoDAF wird durch ein grobes sechsstufiges Vorgehen angeleitet:

- 1) Bestimmung des Verwendungszwecks der Architektur
- 2) Bestimmung des Umfangs der Architektur
- 3) Bestimmung benötigter Informationen und Daten zur Unterstützung der Architekturentwicklung
- 4) Durchführung von Analysen zur Erreichung der architekturenspezifischen Ziele
- 5) Dokumentation von Ergebnissen in Übereinstimmung mit den Anforderungen von Entscheidern

Das DoDAF greift die Terminologie der ISO/IEC42010 auf. Die zu liefernden Views sind durch Viewpoints definiert; jeder Viewpoint erfüllt einen oder eine Kombination der folgenden Zwecke:

- High-Level Concept: Breite Zusammenfassung über das gesamte System
- Low-Level Concept: Spezifizierte Information für einen bestimmten Zweck
- Connection-Level Concept: Informationen über Beziehungen der Systemelemente

Konkret sind im DoDAF die folgenden Viewpoints mit entsprechenden Views (bzw. hier: Modellen) definiert:

- All Viewpoint: Dieser Viewpoint beschreibt alle übergeordneten Aspekte, die einen Bezug zu weiteren Viewpoints haben. Innerhalb dieses Viewpoints werden zur Architekturbeschreibung die Modelle Overview and Summary Information sowie Integrated Dictionary verwendet. Im ersten Modell werden die Vision, Ziele, Aktivitäten, etc. dargestellt. Das zweite Modell enthält die Definitionen aller zur Architekturbeschreibung notwendigen Begriffe und Daten.
- Capability Viewpoint: Die Anforderungen an das System werden in diesem Viewpoint definiert. Dazu stehen sieben Beschreibungsmodelle zur Verfügung.
- Data and Information Viewpoint: Durch diesen Viewpoint werden die Beziehungen zwischen Daten zur Darstellung der Architekturen aus der Sicht des Capability, Operational, Service und Systems Viewpoint darzustellen, definiert. Dies basiert auf einem konzeptionellen, logischen und physikalischen Datenmodell.
- Operational Viewpoint: Im Fokus dieses Viewpoints stehen Anwendungsszenarien und Aktivitäten, die zur Erfüllung der Anforderungen notwendig sind. Aktivitäten sind hierbei alle Elemente und Ressourcenflüsse, die zur Durchführung von Operationen erforderlich sind. Zehn Modelle zur Beschreibung von Umgebung, Ressourcenflüssen, Aktivitäten, und Beziehungen können verwendet werden.
- Project Viewpoint: Dieser Viewpoint beschreibt die Abhängigkeiten zwischen Organisation und durchgeführten Projekten, die Zeitpläne der Projekte sowie eine Zuordnung zwischen Projekt und den damit zu erfüllenden Anforderungen. Drei Modelle werden dabei herangezogen.
- Service Viewpoint: Durch diesen Viewpoint werden die Beziehungen zwischen dem Anwender, Anwendungsszenarien und Services beschrieben. 13 verschiedene Modelle werden dazu eingesetzt. Beispielhaft seien Modelle zur Darstellung der Service-Funktionen sowie der Entwicklung der Technologien, welche das zukünftige Service-Angebot beeinflussen, genannt.
- Standard Viewpoint: Die verwendeten Richtlinien und Standards sowie ein Ausblick auf deren Entwicklung sind Inhalt dieses Viewpoints. Zwei Beschreibungsmodelle werden dazu verwendet.
- Systems Viewpoint: Die Beschreibung des Systems, dessen Zusammensetzung aus Subsystemen und deren Beziehung zueinander und zur Umfeld steht im Fokus dieses Viewpoints. Ziel ist die Umsetzung der in den Operational und Capability

Viewpoint definierten Anforderungen und Anwendungsszenarien. 13 Modelle stehen zur Verfügung.

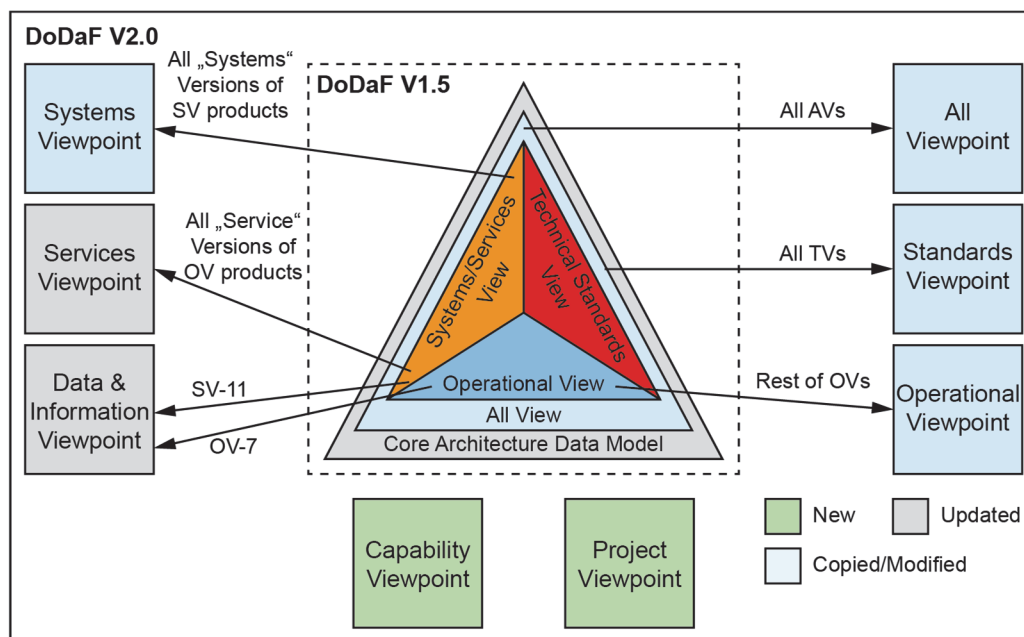


Bild 3-11 DoDAF Framework nach DoDAF

Zusätzlich werden im DoDAF einige Hinweise zur Visualisierung der Views und zur Nutzung gegeben. Während die grafischen Darstellungen der UML für erfahrene Architekten im Regelfall geeignet sind, sollten für weitere Stakeholdergruppen spezifische Darstellungsformen gefunden werden. Hierzu werden einige grundlegende Darstellungsformen genannt, bspw. Tortendiagramme. Hinsichtlich der Nutzung dieser Darstellungsformen wird kurz erklärt, dass sie zur Verbesserung und Gestaltung der Architektur genutzt werden können.

Bewertung: Das Vorgehen stammt aus der Militärtechnik, erhebt aber den Anspruch, auch in zivilen Branchen einsetzbar zu sein. Es stellt ein datenzentriertes Vorgehen in den Mittelpunkt, was die Adressierung spezifischer Stakeholderinteressen ermöglichen soll. Die Idee des modellbasierten SE wird nicht explizit aufgegriffen, die empfohlenen Modelle (bspw. Datenflussdiagramme) und ein UML-Profil zeigen jedoch eine Nähe. Insgesamt werden zwar insb. die zahlreichen Viewpoints sehr detailliert beschrieben, erscheinen in der Summe jedoch aufgrund des sehr generischen Vorgehensmodells nicht handhabbar für ein individuelles und organisatorisches akzeptiertes Vorgehen, das insb. auch die Nutzung von Methoden in den Mittelpunkt stellt. Die Mächtigkeit des Ansatzes schränkt in somit gleichzeitig für die Anwendung selbst ein.

3.3.3 Kruchten 4+1

Das Framework Kruchten 4+1 bildet die Grundlage der Dokumentation von Architekturen im Rational Process (vgl. Kapitel 3.2.3), da beide maßgeblich von PHILIPPE KRUCH-

TEN entwickelt wurden (Bild 3-12). Zentrale Idee ist, den Architekturentwicklungsprozess über ein Framework zu strukturieren. Hierzu sollen mehrere Sichten definiert werden, die unterschiedlichen Concerns adressieren. Insgesamt verwendet dieses 4+1-Sichtenmodell nach KRUCHTEN fünf Sichten, die die Betrachtung von Interessen verschiedener Stakeholder ermöglichen. Die Sichten sind nicht unabhängig und überschneiden sich auch teilweise. Sichten, die in der Softwareentwicklung ggf. keine Rolle spielen, können auch weggelassen werden – ein Tailoring ist somit möglich. Bild 3-12 lassen sich die Sichten entnehmen, mit denen die Concerns der verschiedenen Stakeholder adressiert werden [Kru95]:

- **Logical View:** Die logische Sicht unterstützt die funktionalen Anforderungen an das System. Dazu wird das System in Schlüssel-Abstraktionen in Form sogenannter objects oder object classes unterteilt. Das ermöglicht zum einen Funktionsanalysen, zum anderen die Identifizierung der verwendeten Mechanismen und Elemente.
- **Process View:** Die Prozess-Architektur kann in verschiedenen Abstraktionsniveaus beschrieben werden. Dabei werden auf jedem Level unterschiedliche Concerns adressiert. Unter einem Prozess wird in diesem Zusammenhang eine Bündelung von Aufgaben verstanden, die eine ausführbare Einheit bilden.
- **Development View:** In dieser Sicht steht die Entwicklungsumgebung im Fokus. Dabei wird die Anordnung der Module innerhalb der Software betrachtet. Diese ist in Subsysteme untergliedert, die von Entwicklungsteams entwickelt werden. Die Entwicklungs-Architektur zeigt dabei die Export- und Import-Beziehungen zwischen den Subsystemen.
- **Physical View:** Diese Sicht ordnet der Software die erforderliche Hardware zu. Dabei kann es mehrere Architekturen geben, z.B. zur Entwicklung, zum Testen, zur Implementierung oder für verschiedene Kunden. Idealerweise ist die physikalische Architektur flexibel aufgebaut und hat wenig Einfluss auf den Quellcode.

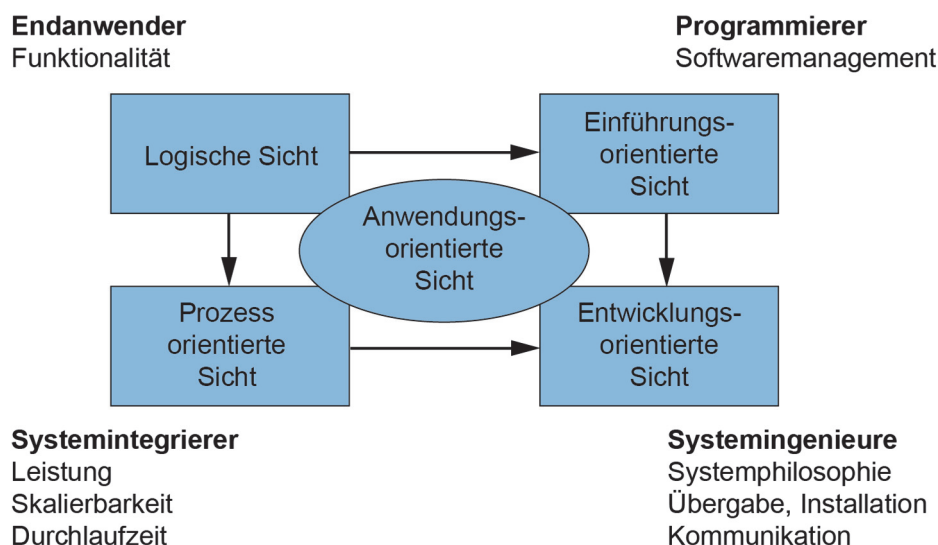


Bild 3-12 Das 4+1 Sichtenmodell [Kru95]

Für jede der Sichten werden die verwendeten Elemente festgelegt. Jede Sicht hat eine eigene Notation. Darüber hinaus beschreibt KRUCHTEN Regeln, nach denen Elemente der verschiedenen Sichten miteinander verbunden sind.

Das „+1“ im Namen steht für die sog. anwendungsorientierte Sicht – auch Anwendungsszenario genannt. Diese fünfte Sicht stellt das Zusammenwirken der anderen vier Sichten dar. Die *scenarios* sind eine abstrakte Form der wichtigsten Anforderungen und bringen Abläufe zwischen Prozessen und Komponenten zum Ausdruck. Diese Sicht ist redundant zu den anderen. Sie ermöglicht die Identifizierung von Architekturelementen während des Entwurfs der Architektur, dient aber auch der Validierung und Visualisierung des Architekturentwurfs.

Bewertung: Das Framework definiert sehr genau die für jede der Sichten zu verwendenden Elemente. Jede Sicht hat eine eigene Notation. Darüber hinaus beschreibt der Ansatz Regeln, nach denen Elemente der verschiedenen Sichten miteinander verbunden sind. Aspekte der Integration in die Organisation und Prozesse berücksichtigt der Ansatz nicht, ebenso kann trotz des Bezugs auf den Rational Prozess nicht von einer Berücksichtigung der Erfolgsprinzipien bekannter Managementansätze ausgegangen werden, ebenso nicht von einer Rationalitätssicherungsfunktion.

3.3.4 A3 Architecture Framework

A3 Architecture Overview (A3AO) ist ein Konzept für den Wissenstransfer und die wirksame Kommunikation von architekturelevantem Wissen [Bor10]. Es wurde entwickelt, um auf Basis eines Frameworks strukturiert Wissen über die Architektur eines beliebigen Systems im Rahmen von Reverse Engineering Prozessen abzubilden. A3AO konzentriert sich dabei auf die Darstellung und Vermittlung von Informationen und geht nicht näher auf deren Generierung ein. Der Begriff A3 stammt vom Papierformat DIN A3 – dieses hat sich nach BORCHES bereits im Lean-Ansatz von Toyota hervorragend bewährt und soll nun auch im Rahmen der Entwicklung genutzt werden.

Bild 3-13 zeigt den Aufbau der A3AO-Architekturdarstellung im Überblick. In Summe werden für den Ansatz zwei DIN A3 Seiten genutzt. Durch die Beschränkung des verfügbaren Platzes wird begünstigt, dass nur wirklich relevante Informationen kommuniziert werden: Die Vorderseite fasst textuell Informationen über das System und das Projekt zusammen („A3 Summary“) – die zweite Seite zeigt ein strukturiertes Modell des Systems („A3 Model“) und besteht aus verschiedenen miteinander verbundenen Sichten. Die Beziehungen zwischen den Sichten werden explizit durch Pfeilbeziehungen visualisiert. So werden bspw. Hauptfunktionen mit den entsprechenden realisierenden physischen Elementen verbunden, *Design Annotations* werden durch kleine Sternchen hervorgehoben. Die Sichten – bzw. die A3-Bögen allgemein – werden mit Leserichtung von links oben nach rechts unten ausgerichtet. Zusätzlich sind links oben angeordnete Informationen stabiler gegenüber Änderungen, als Informationen rechts unten. Ein Beispiel anhand der A3 Summary: Problem und Hintergrund sind wichtiger und werden

sich tendenziell weniger ändern als bspw. Zusatzinformationen, die rechts unten auf der Vorderseite abgebildet sind; ähnlich verhält es sich im A3 Model. Die Modellierungssprache für das A3 Model ist nicht festgelegt, u.a. um unproduktive Diskussionen über die korrekte Darstellungsform zu vermeiden.

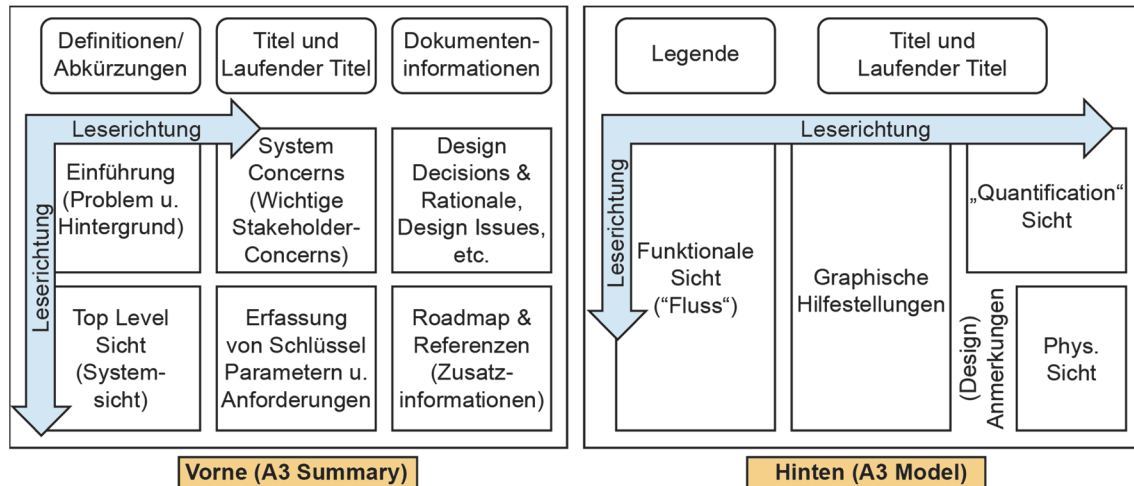


Bild 3-13 Aufbau des A3AO (Vorder- und Rückseite) nach [Bor10]

BORCHES sieht in der Praxis keinen echten Bedarf für eine automatisierte oder intelligente Softwareunterstützung zur Erstellung der Overviews, insb. da solche bei der kreativen Tätigkeit der Architekturgestaltung häufig nicht gewünscht sind [Bor10]. Entsprechend ist für die Methode keine Softwareunterstützung vorgesehen, sondern lediglich Papierbögen. Das ist ggf. auch der Grund, warum das A3AO keinen Anspruch auf Vollständigkeit, Formalität der Modelle oder Ausführbarkeit bzw. Transformierbarkeit der generierten Modelle erhebt. In erster Linie geht es darum, das Spannungsfeld zwischen sehr detailliertem Wissen und abstrakten Visualisierungen einer Architektur innerhalb einer Darstellungsform auf leichte Art und Weise mit wenig Zusatzaufwand zu überbrücken und dadurch in Besprechungen oder der alltäglichen Arbeit unterstützen. Inzwischen existieren erste Ideen, die Bögen digital zu erzeugen [Bor10]. Zudem ist der Einsatzzweck nicht mehr auf das Reverse Engineering festgelegt. Vielmehr wird A3AO nun auch in der Architekturentwicklung eingesetzt, aber auch bspw. um die Kommunikation über Simulationen und Test zu erleichtern [HB15].

Bewertung: Mit dem A3AO liegt ein Konzept vor, dass die Architekturbeschreibung eines Systems mit einfachen Mitteln zur Unterstützung von unterschiedlichen Aufgaben bereitstellt und sich dabei an den Bedarfen der Stakeholder orientiert und ihre Interessen berücksichtigt. Dennoch ist der Anwendungsbereich momentan noch sehr stark auf die Architekturentwicklung und nahestehende Tätigkeiten konzentriert. Wie die Informationen generiert werden, bleibt jedem Stakeholder selbst überlassen. Der A3AO bietet lediglich das Kommunikationsmedium, mit dem aber auch Aufgaben einer Modellplanung angegangen werden. Hervorzuheben ist die pragmatische Herangehensweise: Einerseits bezieht sich das auf die Sprachen- und Methodenunabhängigkeit, andererseits aber insb. auf die Adaption der Erfolgsprinzipien des Lean-Ansatzes.

3.4 Prozess-, Methodenmodelle und Methodensammlungen

Durch die Mechatronisierung sind die Anforderungen an die Produktentstehungsarbeit gestiegen. Die Nutzung von dedizierten Prozessen und Methoden im Rahmen von Produktentstehungsprozessen wird in der Wissenschaft als essentiell angesehen – wenngleich die tägliche Produktentstehungsarbeit bislang nur wenig methodisch unterstützt wird. Zur Verbesserung dieser Situation haben zahlreiche Initiativen verbesserte Prozessmodelle für mechatronische Entwicklungsprojekte und v.a. auch Modelle zur standardisierten Methodenbeschreibung und -darstellung entwickelt. Sie helfen, die spezifischen Aufgaben von Mitarbeitern in der Produktentstehung besser zu verstehen und zu unterstützen. Das Systemmodell selbst spielt bei diesen Ansätzen meist keine Rolle. Jedoch fokussieren sie die Integration von Vorgehensmodellen, Methoden und Hilfsmitteln in die Organisation und die Prozesse.

3.4.1 Forflow

Im Projekt FORFLOW wurde eine neuartige Möglichkeit geschaffen, in multidisziplinären Entwicklungsprojekten eine situationspezifische Unterstützung für die Entwicklung anzubieten. Bild 3-14 zeigt die Lösung als Schema, konkret bedeutet das ...

„ ... eine Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung, ... [die] ... den Entwickler durch eine zielgerichtete Führung in den Entwicklungssituationen unterstützt.“ [MHJ+10]

Die Grundlage der Lösung ist das FORFLOW-Prozessmodell mit dem Prozessnavigator. Es bietet u.a. eine situationspezifische Prozessplanung, eine verbesserte Informationsversorgung der Entwickler, die Integration von CAX-Werkzeugen und die Verknüpfung mit während der Entwicklung entstehenden Produktmodellen. Das Prozessmodell besteht aus einem Drei-Level-Konzept, das durch den Prozessnavigator gesteuert wird. Level 1 bildet die Hauptebene mit sechs Prozessphasen. Die Prozessphasen beginnen mit dem *Klären der Aufgabe* und enden mit dem *Start und Steuerung der Serienproduktion*. Auf dem zweiten Level wird die Hauptebene detailliert, so dass sich schlussendlich 90 Teilprozesse auf Level 3 befinden. Während für die Aktivitäten auf Level 1 und Level 2 eine Reihenfolge vorgesehen ist, existiert für die Aktivitäten der dritten Ebene keine stringente Abfolge.

Eine zentrale Idee ist, den Entwicklungsprozess über Produktmodelle zu steuern. Mit diesem Anspruch wurden in den zwei maßgeblichen Projektbereichen Produkt & Prozess sowie Methoden & Werkzeuge verschiedene Konzepte und Methoden erarbeitet:

- Eine prozessorientierte Beschreibung von Produktmodellen soll die Einbindung von Produktmodellen und Produktdaten in den Entwicklungs-Workflow ermöglichen. Produktmodelle enthalten spezifische Produktinformationen, die im Prozess erzeugt werden. Produktmodelle sind im Sinne des Projekts Dokumente je-

der Art, die Informationen zum Produkt enthalten, also bspw. CAD-Modelle und Ergebnisse von FE-Berechnungen. Diese Modelle sind nicht mit einem integrierten Produktdatenmodell zu verwechseln. Da jeder Fachbereich seine eigenen Herangehensweisen und Werkzeuge nutzt und die dabei entstehenden Informationen stark vernetzt sind, führt das zu zahlreichen Daten- und Informationsverlusten in der Praxis. Ein parameterbasierter Ansatz hilft, die Integration der Produktmodelle in den Prozess zu bestimmen und fortlaufend neu zu berechnen. Die Parameter Inhalt, Verwendungszweck, Konkretisierungsgrad, Entwicklungsstand und Vernetzungsgrad spannen einen 5-dimensionalen Raum auf; die Produktmodelle werden hier eingeordnet. Auf diese Weise soll die Zuordnung zu Entwicklungsworkflows gelingen [Lau10].

- Projekte mit dem Fokus der systematischen Einbindung von CAE-Werkzeugen in den Entwicklungsprozess verfolgen das Ziel, den Datenaustausch von Produktmodellen in unterschiedlichsten Formaten wirksam zu ermöglichen. Eine Vielzahl von Datenmodellen erschwert das jedoch ungemein, mit der Folge von erheblichem Mehraufwand in der Zusammenarbeit. Aus diesem Grund wurden auf Basis einer Analyse wesentlicher Werkzeuge der Produktentstehung und ihrer unterstützten Datenschnittstellen für den Austausch von 3D-Geometrien wichtige Formate extrahiert und als Standardschnittstellen für den Prozessnavigator definiert. Ein Bewertungsmodell empfiehlt unter Berücksichtigung allgemeiner, aber auch unternehmensspezifischer Randbedingungen eine geeignete Schnittstelle für die geplante Anwendung [Tro10].

Da immer mehr Entwicklungsaktivitäten durch computergestützte Verfahren unterstützt werden können, wird die Kopplung und prozessorientierte Verknüpfung von Methoden und Werkzeugen ebenfalls angegangen. Generell gestaltet sich ihr Einsatz durch die Vielzahl existierender Produkte sehr schwierig aufgrund der Unübersichtlichkeit. Meist müssen auch mehrere CAX-Programme kombiniert eingesetzt werden – wozu diese entsprechend dem geplanten Einsatzzweck zunächst entsprechend kombiniert werden müssen. Ergebnis in Forflow ist ein Mehrebenenansatz zur Benutzerunterstützung, sodass auf diese Weise ein standardisierter Prozess und Werkzeugunterstützung erstellt wird, der in ähnlichen Projekten ebenfalls eingesetzt werden kann [Zap10].

Bewertung: Forflow bietet einen klassischen Ansatz zur Planung und Durchführung mechatronischer Produktentstehungsprozesse und zum Arbeiten im Produktlebenszyklus. Klassisch heißt in diesem Zusammenhang: Die Herkunft aus der klassischen Konstruktionsmethodik und der damit verbundenen Herangehensweisen ist erkennbar, die Idee des Systems Engineering und des Systemmodells spielen keine Rolle, wenngleich das modellbasierte Systems Engineering grundsätzlich integrierbar wäre. Hervorzuheben sind viele pragmatische Ansätze – sie unterstützen den rationalen Einsatz von Prozessen und Methoden.

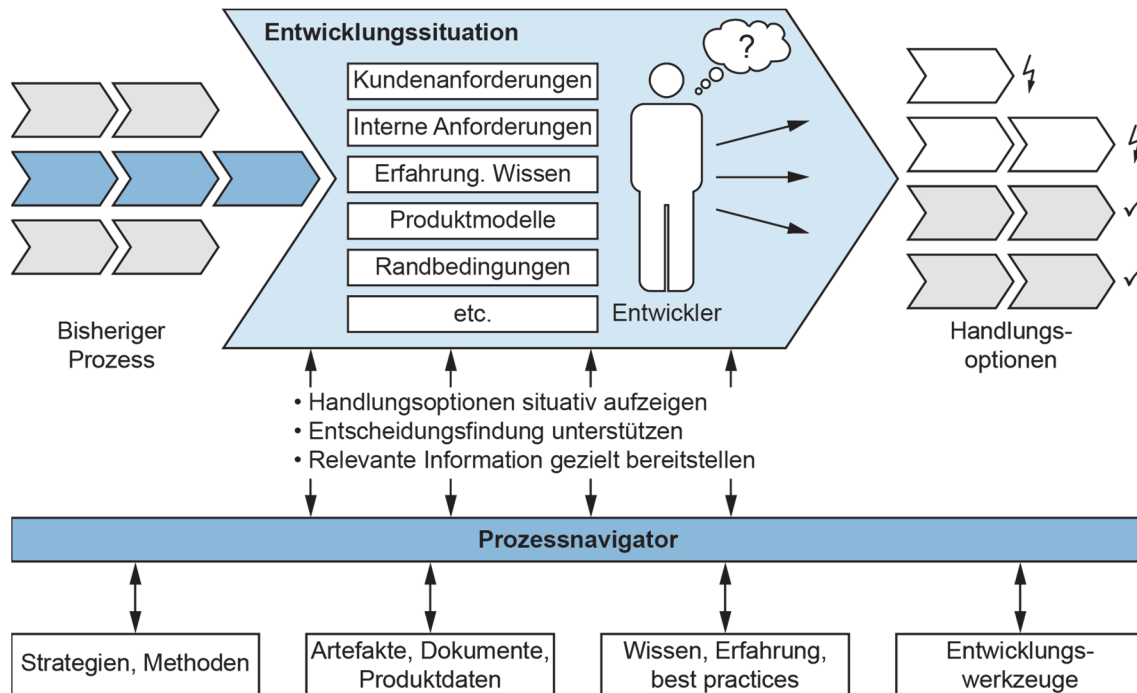


Bild 3-14 Forflow

3.4.2 Prozessorientiertes Methodenmodell nach BIRKHOFFER

Im Projekt „thekey to innovation“ (thekey) entstand das Prozessorientierte Methodenmodell (PoMM) nach BIRKHOFFER (Bild 3-15). Ziel war eine standardisierte Methodendarstellung auf Basis eines weitreichenden Frameworks, um so die methodische Durchdringung der Produktentstehung zu verbessern. Dabei sollten zwei entscheidende Aspekte berücksichtigt werden: Mit dem Framework müssen unterschiedliche Darstellungsweisen der Methoden möglich sein, um die vielen potentiellen Nutzer in den unterschiedlichen Situationen und verschiedensten Anwendungen mit unterschiedlichen Fähigkeiten adäquat zu unterstützen. Zudem sollten die Methoden in einem Prozessmodell aufbereitet sein, da eine Methode einerseits wie ein Prozess betrachtet werden kann, andererseits um auch einen geeigneten Bezug zu den entsprechenden Zielanwendungen der Methoden sicherzustellen [BKB+02], [SBB02]. Neben dem Verständnis, dass die Anwendung einer Methode als Prozess verstanden werden kann, sind weitere Einflussparameter zur standardisierten Beschreibung notwendig, z.B. Informationen über die Anwender oder Hilfsmittel.

Die konstituierenden Elemente des PoMM sind die Prozessmodule (blau) und die Zugangsmodule (weiß). Die Prozessmodule sind prozessorientiert beschrieben, ihr Inhalt hat einen direkten Einfluss auf die Methode. Diese Prozessmodule sind der Input, der Output, die Vorgehensweise, Einflüsse durch den Nutzer, generelle Randbedingungen sowie Hinweise und Hilfen. Die Zugangsmodule sind für eine flexible und detaillierte Suche gedacht, ebenso für die Verlinkung zu ähnlichen Methoden. Im Prinzip entsprechen die Zugangsmodule einer Methodenklassifikation.

Gerade die Prozesssicht auf die Methoden sollte hier herausgestellt werden: Ein Input wird durch eine explizite Vorgehensweise in einen Output überführt. Das PoMM sieht für Input und Output kurze Beschreibung des Startzustands und des Endzustands vor, ebenso gewünschte Ergebnisse am Methodenende. Die Vorgehensweise, die den Input in den Output transformiert, sollte als Struktur beschrieben werden, z.B. als Flussdiagramm und dazugehörigen textuellen Informationen. Durch diese beiden unterschiedlichen Beschreibungsformen sollen den unterschiedlichen Lern- und Anwendungsbedürfnissen der Methodenanwender Rechnung getragen werden.

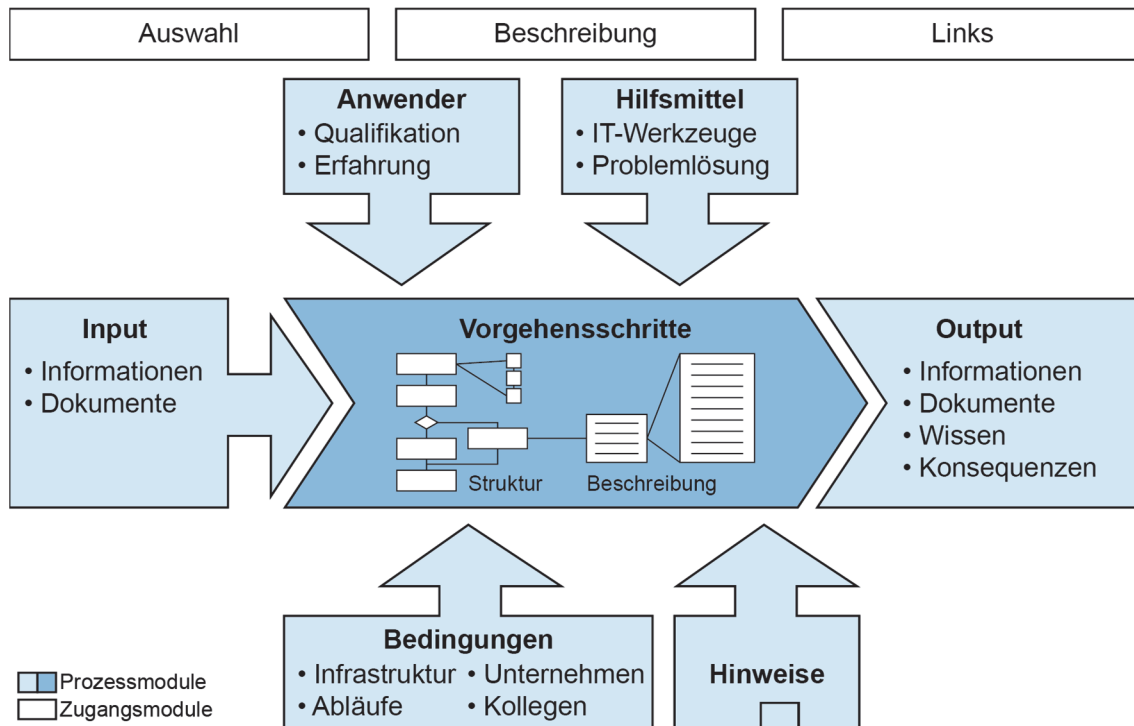


Bild 3-15 Prozessorientiertes Methodenmodell – PoMM nach [BKB+02]

Bewertung: Im PoMM spielen Systems Engineering und die damit verbundenen Konzepte keine Rolle. Dennoch ist es hervorragend geeignet, die Rahmenbedingungen für die Auswahl und Integration von Methoden in den Entwicklungsprozess zu integrieren. Durch die detaillierte und standardisierte Beschreibung von Methoden unterstützt es eine gute Modellplanung.

3.4.3 Münchener Methodenmodell

Wie der Einsatz von Methoden geplant werden kann, stellt das Münchener Methodenmodell dar [Lin09]. Unter dem Begriff Methodeneinsatz werden die weiteren Aspekte Methodenauswahl, Methodenanpassung und Methodenanwendung zusammengefasst; zunächst gilt es jedoch, den notwendigen Methodeneinsatz exakt zu klären. Diese vier Aspekte werden in vier Schritten gemäß Bild 3-16 durchlaufen.

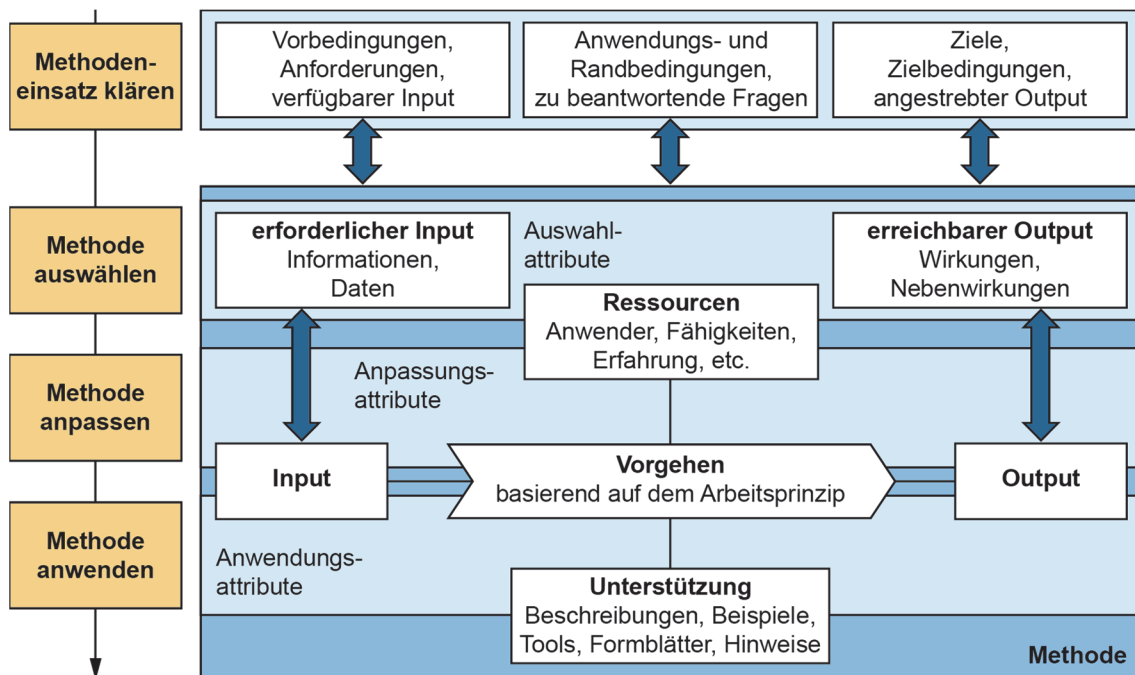


Bild 3-16 Münchener Methoden Modell – MMM

Im ersten Schritt wird der Methodeneinsatz geklärt. Es geht also u.a. um Vorbedingungen, den notwendigen Input, Anwendungs- und Randbedingungen und insb. Ziele, Zielbedingungen und den angestrebten Output. Für den Fall, dass ein Methodeneinsatz sinnvoll erscheint, werden die weiteren Schritte durchlaufen:

- Es muss für die vorliegende Situation eine geeignete Methode ausgewählt werden. Dabei wird überprüft, ob die anstehende Aufgabe von der Methode unterstützt wird und ob die angestrebten Zielen der Aufgabe und die Ergebnisse der Methode übereinstimmen. Dazu wird der notwendige Input ebenso definiert, wie der Output. Der Output entspricht der Wirkung zuzüglich weiterer Nebenwirkungen in Abhängigkeit des Inputs. Zusätzlich haben verschiedene Ressourcen eine enorme Einwirkung auf die Methode, insb. die Qualifikation der Anwender.
- Da im Regelfall die existierenden Methoden nicht unverändert auf die geplante Einsatzsituation übertragen werden können, ist die Methodenanpassung eminent wichtig. Diese sollte vor der Methodenanwendung durchgeführt werden. Es ist möglich, über Anpassungsattribute bspw. Teilschritte anderer Methoden zu integrieren oder einzelne Module einer Methode wegzulassen.
- Die Anwendung erfolgt durch Bearbeitung der geplanten Aufgabe. Dabei soll jede Methode durch Formblätter, Werkzeuge oder weitere Hilfsmittel unterstützt werden. Zusätzlich kann die Methode auch weiterhin angepasst werden.

Bewertung: Das Münchener Methoden Modell zeigt einen Weg auf, methodisches Arbeiten in der Produktentstehung zu etablieren, bezieht sich dabei aber nicht auf ein Systemmodell. Zentral sind aber die vorzeitige Klärung, ob ein Methodeneinsatz sinnvoll

ist und die dann folgende individuelle Anpassung, mit deren Hilfe die Methoden geeignet unterstützen sollen. Die Methode selbst wird wie im PoMM als Transformation eines Inputs über ein dediziertes Vorgehen in einen Output verstanden.

3.4.4 Situative Entwicklungsaufgaben- und Methodenauswahl

PONN ermöglicht mit seiner Systematik die zielgerichtete Methodenauswahl im Rahmen der Konzeptentwicklung technischer Produkte – abhängig von der vorliegenden Entwicklungssituation und Aufgabe [Pon07]. Die Systematik umfasst drei Bestandteile, die für die Konzeptentwicklung beliebiger technischer Produkte ausgelegt sind:

- Ein **Beschreibungsmodell** definiert die drei Elemente Entwicklungssituation, Entwicklungsaufgabe und Methode und stellt ihr Zusammenspiel dar: Durch die Beschreibung der Verknüpfungen von Situationen mit Aufgaben, Aufgaben mit Methoden und Situationen mit Methoden soll die Auswahl einer Methode möglich werden (vgl. Bild 3-17).
- Eine **Anwendungsmethodik** definiert einen vierstufigen Prozess (Situationsanalyse, Ausgabenwahl, Methodenauswahl, Methodenanwendung), mit dem auf Basis der aktuellen Entwicklungssituation die anstehenden Aufgaben und geeignete Methoden identifiziert werden können.
- Eine **Informationssammlung** stellt ausgehend vom Beschreibungsmodell Prozessbausteine, Methoden und deren Verknüpfungen bereit. Die Verknüpfung erfolgt anhand spezifischer Faktoren zur Beschreibung des direkten und indirekten Kontext der Entwicklungssituation. Diese wurden in einem webbasierten Werkzeug prototypisch implementiert.

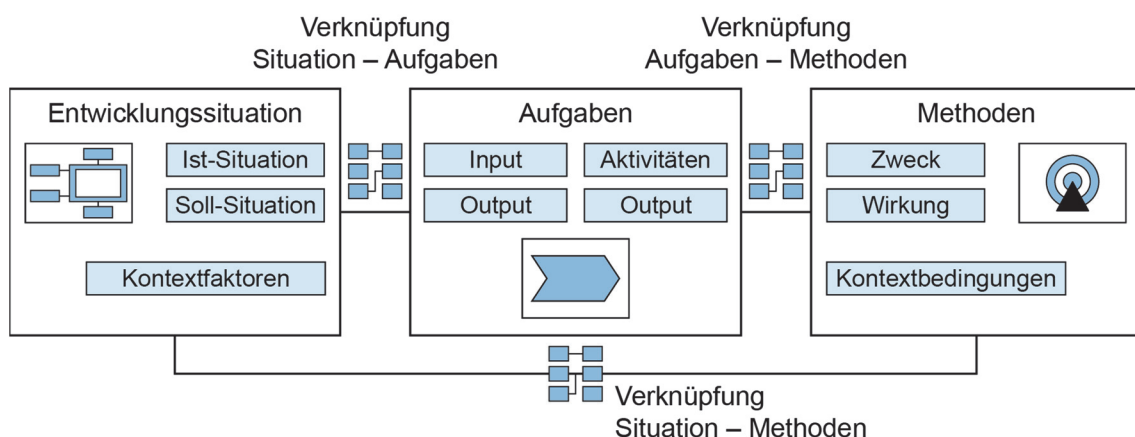


Bild 3-17 Beschreibungsmodell nach PONN

Grundlage der Systematik ist das Beschreibungsmodell: Entwicklungssituationen werden durch einen direkten und einen indirekten Kontext beschrieben. Der direkte Kontext charakterisiert das Betrachtungsobjekt des Ingenieurs und dessen Wissensstand. Die dazugehörigen Kontextfaktoren können bspw. Ergebnisse der Tätigkeit sein, für die

nach Ausführung einer Aufgabe ein Ist- und ein Soll-Zustand zu definieren sind. Die indirekten Kontextfaktoren betreffen in erster Linie die Methodenauswahl, haben allerdings keinen direkten Einfluss auf die Entwicklungsaufgabenauswahl. Das sind bspw. Faktoren aus dem Entwicklungsprozess, der Entwickler oder Rahmenbedingungen.

Die Entwicklungsaufgaben werden durch Prozessbausteine beschrieben, die durch die Merkmale Name – Kurzbeschreibung – Abbildung – Aktivität – Input – Output standardisiert dargestellt werden. Ähnlich sind die Methoden aufbereitet; ihre Beschreibung umfasst Namen – Kurzbeschreibung – Abbildung – Zweck – Voraussetzungen – Wirkung – Kontextbeziehungen.

Die Verknüpfung von Entwicklungssituation – Aufgabe – Methode wird über eine Verknüpfungsstruktur ermöglicht, wie sie auch in Bild 3-17 dargestellt ist. Situationen und Aufgaben werden über die Ist- bzw. Soll-Situation und die Methodeneingangs- und Ausgangsgrößen (Input/Output) verknüpft. Aufgaben und Methoden können über drei Wege miteinander verknüpft werden: direkt, über Tätigkeiten und über Objekte. Die Situationen werden über die indirekten Kontextfaktoren und den Kontextbedingungen mit den Methoden verknüpft.

Bewertung: Im Ansatz von PONN spielen das Systems Engineering und die damit verbundenen Konzepte keine Rolle. Er dient aber als gutes Beispiel für die detaillierte Planung des Methodeneinsatzes in Produktentwicklungsprojekten anhand verschiedener Einflussfaktoren. Das dient der Rationalitätssicherung.

3.5 Organisationskonzepte im Systems Engineering

Die Einführung von MBSE und die Integration in die Organisation und die Prozesse sind elementar für den Erfolg des Ansatzes. Bislang sind keine Ansätze bekannt, die dieses Thema adressieren. Im klassischen Systems Engineering existieren aber Konzepte, wie der Systems Engineer oder ein entsprechendes Team in der Organisation verankert werden können.

3.5.1 SE-Team nach FRIEDENTHAL

FRIEDENTHAL schlägt zur Integration von SE in die Organisation ein multidisziplinäres Systems Engineering Team vor. Dieses soll dem breiten Anforderungsprofil an den Systems Engineer gerecht werden. Dazu empfiehlt er fünf Einheiten, aus denen sich eine Systems Engineering-Abteilung zusammensetzt [FMS12]. Das SE-Management Team verantwortet die Management-Prozesse zur Planung und Kontrolle des technischen Aufwands. Ein Requirements Team befasst sich mit der Analyse der Stakeholder-Anforderungen, mit der Identifikation von Anwendungsszenarien („Conops“) sowie mit der Spezifikation und Validierung von Anforderungen auf Systemebene. Das Architecture Team entwirft die Architektur des Systems. FRIEDENTHAL versteht darunter die Definition der Komponenten sowie deren Beziehungen zueinander. Darüber hinaus sind die Anforderungen den erfüllenden

Komponenten zuzuordnen. Zur Bewertung des Systems aus technischer Sicht wird das System Analysis Team eingesetzt. Es beurteilt Aspekte wie die Verlässlichkeit, die Performance oder die Kosten. Zuletzt beschäftigt sich das Integration and Test Team mit der Entwicklung von Testplänen und -abläufen sowie deren Durchführung. Bild 3-18 gibt einen Überblick über den Teamaufbau.

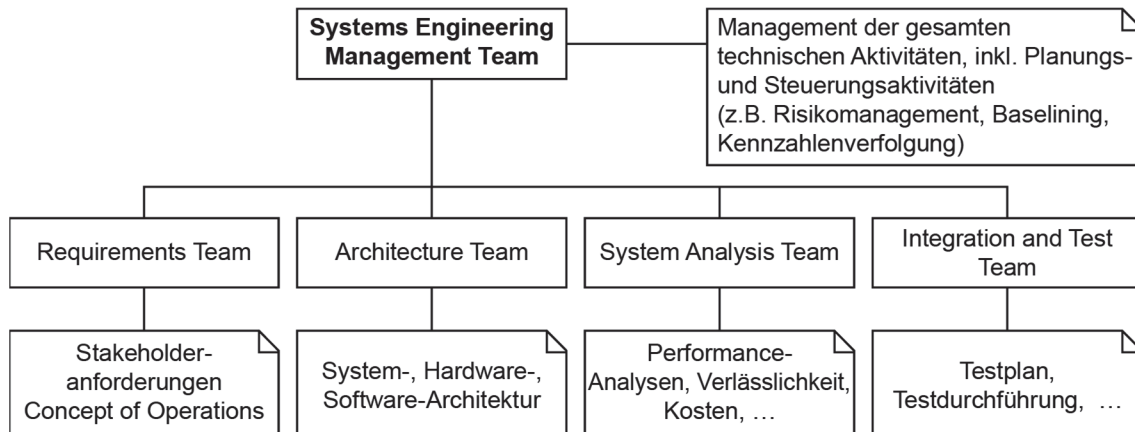


Bild 3-18 Das Systems Engineering-Team

Bewertung: Der Ansatz von FRIEDENTHAL gibt einen Überblick darüber, welche Leistungen eine SE-Abteilung im Lebenszyklus erbringen muss. Das Systemmodell spielt in dem Konzept jedoch keine Rolle, weshalb ein Großteil der gestellten Anforderungen nicht beurteilt werden kann. Allerdings zeigt FRIEDENTHAL auf, dass ein SE-Team durchaus auch als Vorlage für ein „MBSE-Team“ dienen kann.

3.5.2 „The Systems Engineer in the Engineering Organization“

PARNELL ET AL. beschreiben sehr detailliert das Zusammenspiel des Systems Engineers mit anderen Rollen in der Organisation; dazu gehört insb. das Zusammenspiel mit dem Program Manager bzw. dem Produktmanager. Prinzipiell kann dabei die Rolle des Systems Engineers als „Technical Leader“ verstanden werden, der den Program Manager signifikant unterstützt [PDH11]. Dabei geht es also um die Koordination sämtlicher technischen Aktivitäten über den gesamten Produktlebenszyklus, insb. jedoch von Projektbeginn bis zur Integration der disziplinspezifischen Arbeiten. Hier stehen bspw. Aufgaben wie Kundenanalysen, Anforderungsdefinition und Auswahl von Komponenten für das System an. Wesentlich bei der Arbeit ist jedoch, dass der SE einerseits diese Arbeiten aktiv durchführt, aber auch die anderen Rollen unterstützt und berät.

Bewertung: Die Darstellungen von PARNELL sind sehr allgemein, so dass die genaue Einbindung des Systems Engineers in die Organisation und den Prozess nur interpretiert werden kann. Die Ansätze des Model-Based Systems Engineerings als zentrales Werkzeug des Systems Engineers oder der anderen Stakeholder werden nicht berücksichtigt. Generell muss aber aus den Beschreibungen geschlussfolgert werden, dass das Zusammenspiel der einzelnen Rollen auf Basis des Systemmodells detailliert anhand von Ver-

antwortlichkeiten und Aufgaben zu beschreiben ist und die gemeinschaftliche Erstellung und Nutzung des Systemmodells eine realistische Annahme darstellt.

3.6 Handlungsbedarf

In den Kapiteln 3.1 bis 3.5 wurde jeder vorgestellte Ansatz einzeln bewertet. In diesem Kapitel erfolgt eine zusammenfassende Bewertung aus Sicht der in Kapitel 2.7 gestellten Anforderungen. Die Übersicht der Bewertungen ist in Bild 3-19 aufgezeigt. Zusammenfassend lassen sich die Anforderungen wie folgt bewerten.

- A1) Systems Engineering als Basis:** Weitgehend alle bewerteten Ansätze berufen sich auf das Systems Engineering. Genau genommen adressieren die meisten Ansätze jedoch nur Teilaspekte und legen implizit einen Schwerpunkt. Meist ist der Schwerpunkt die Architekturentwicklung, wie in den untersuchten Frameworks und den Ansätzen zur Modellierung des Systemmodells. Letztere haben zusätzlich oft eine eher fachdisziplinspezifische Herkunft, da sie häufig aus der Softwaretechnik stammen. Die Beschreibungsmodelle reduzieren meist auf die technischen Aufgabenbereiche der Produktentstehung und vernachlässigen somit Teile des Konzepts von HABERFELLNER und der ISO/IEC15288.
- A2) Systemmodell als Mittelpunkt sämtlicher Aktivitäten:** Das Systemmodell als Dreh- und Angelpunkt sämtlicher Aktivitäten wird lediglich im Konzept von EIGNER adressiert – wobei hier die Bedeutung des PLM betont, aber das Zusammenspiel nicht klar dargestellt wird. Im W-Modell wird nicht explizit auf ein Systemmodell abgezielt, vielmehr auf ein Datenmanagement-System zur disziplinübergreifenden Abstimmung – was aber der gleichen Idee entspricht. Klarer wird dies im 3-Zyklenmodell, allerdings mit Einschränkung auf die Produkt- und Produktionssystemkonzipierung und die damit verbundenen Tätigkeiten. Im iPeM kann die Idee erahnt werden, dringt aber durch die Struktur des Systemtripels nicht hervor und wird in den konstituierenden Arbeiten auch nicht erwähnt. Bei den Modellierungsmethoden und den Frameworks ist die Idee des Systemmodells als Mittelpunkt sämtlicher Aktivitäten nicht stark ausgeprägt. Das liegt insb. an der Fokussierung auf die Architekturentwicklung. Wird das Systemmodell für weiterführende Aktivitäten verwendet, dient es der Generierung fachdisziplinspezifischer Artefakte (insb. Softwarecode) oder der Verbesserung der Architektur und nicht weiterführender Aufgaben im Produktlebenszyklus.
- A3) Klares MBSE-Konzept:** Genau genommen präsentiert keiner der untersuchten Ansätze ein nachvollziehbares MBSE-Konzept. Selbst die Modellierungsmethoden gehen – wenn überhaupt – nicht über die Nennung der Definition von INCOSE und die Nennung der Merkmale *Anforderungen, Struktur, Verhalten, Parameter* (CONSENS: Gestalt) hinaus. Vereinzelt wird auf die Unterscheidung zwischen Spezifikation und erster Simulation hingewiesen. Welche Merkmale

aber den Ansatz MBSE ausmachen oder beeinflussen wird nicht herausgearbeitet. Das behindert die Verbreitung des Ansatzes.

- A4) Sprachen- und Methodenunabhängigkeit:** Bei den untersuchten Arbeiten zeichnet sich kein klares Bild hinsichtlich der gestellten Anforderung ab. Die Beschreibungsmodelle sind weitgehend sprach- und methodenunabhängig, geben jedoch vereinzelt Empfehlungen, so z.B. das Modell nach EIGNER mit klarer Tendenz zur SysML. Die Frameworks sind teilweise explizit auf Sprachen und Methoden ausgelegt: DoDAF und Kruchten 4+1 auf die UML, wobei sie prinzipiell auch unabhängig von der UML nutzbar wären.
- A5) Integration von MBSE in die Organisation und Prozesse:** Keiner der dargestellten Ansätze kann ein Konzept vorlegen, dass die Modellierungsaktivitäten zusammen mit einem Konzept zur Verteilung auf Aufgaben in der Organisation zusammenbringt. Lediglich die Ansätze CONSENS, SysMod und A3AO nennen mögliche Verantwortlichkeiten. Die Organisationskonzepte des SE stellen die Einbindung in die Organisation dar, jedoch nicht in Bezug auf das MBSE.
- A6) Adaption von Erfolgsprinzipien bekannter Managementansätze:** Systems Engineering und Model-Based Systems Engineering sind von der Idee her auf einer Stufe mit Ansätzen wie Six Sigma und Lean Management. Der Ansatz LITHE stößt mit der Vision einer Entwicklungsumgebung und verschiedenen Hilfsmitteln in eine ähnliche Richtung, ebenso das iPeM und Forflow – insgesamt ist hier jedoch die Vision stärker ausgeprägt als die Anzahl an Hilfsmitteln. CONSENS, DoDAF und A3AO nennen zwar keine klare Vision, aber erkennen den Bedarf an Flexibilität. Das W-Modell greift Aspekte bekannter Modelle auf, wie z.B. das V-Modell der Softwaretechnik oder die Strukturierung von Ebenen. Letztlich befinden sich die Ansätze alle noch in einem frühen Entwicklungsstadium. Zusätzlich schränken sich die Ansätze durch das fehlende klare Verständnis von MBSE in ihrer Umsetzbarkeit ein.
- A7) Rationalitätssicherungsfunktion:** Explizite Hilfsmittel zur Rationalitätssicherung bieten die analysierten Prozess- und Methodenmodelle der Produktentstehung, die im Prinzip auch in das Rahmenwerk integriert werden könnten – wenngleich sie sich im Regelfall auf die Auswahl allgemeiner Methoden und Werkzeuge beziehen. Mit Bezug auf die Rationalitätssicherung bei der Anwendung von MBSE bietet lediglich der Ansatz LITHE erste grobe Hilfsmittel durch die Kombination der Sprachen SysML und OPL für unterschiedlich erfahrene Anwender. Im Prinzip wird das Thema vollständig vernachlässigt.
- A8) Stakeholderorientierte Modellierung und Nutzung des Systemmodells:** In den analysierten Frameworks klingt die dargestellte Idee konzeptionell an. Die ISO/IEC42010 beschreibt konzeptionell eine stakeholderorientierte Modellierung anhand des Zusammenspiels von Stakeholdern und ihrer Concerns. DoDAF und A3AO bieten für fest definierte Anwendungen gute Hilfsmittel. Sysmod lie-

fert einige Beispiele, welche Diagramme für welchen Stakeholder interessant sein können. Allerdings bietet kein Ansatz Vorgehensweisen zur Modellierung in Abhängigkeit eines Stakeholder-Interesses oder allgemein der Nutzung des Systemmodells. Insb. die Nutzung etablierter Methoden aus dem Management von Produktentstehungsprojekten ist vollkommen vernachlässigt.

- A9) Wohlfundierte Modellplanung:** Bei den untersuchten Ansätzen findet eine wohlfundierte Modellplanung nicht statt. Lediglich die Ansätze der Prozess- und Methodenmodelle nehmen eine gründliche Analyse der Ist-Situation und der Soll-Situation zwecks geeigneter Methodenauswahl vor. Dies ist jedoch losgelöst vom MBSE; LITHE subsummiert zwar im ersten Schritt seines Vorgehens eine Analyse der Ausgangssituation – die übrigen Ansätze zur Modellierung des Systemmodells starten direkt mit Aktivitäten, die der Spezifikation des Produkts dienen, i.d.R. der Definition der Systemgrenze. Organisatorische Aspekte zur Planung des Modellierungsprojekts werden bspw. nicht berücksichtigt.

Somit ist festzustellen: Keiner der untersuchten Ansätze und auch keine triviale Kombination der Ansätze erfüllen ganzheitlich die gestellten Anforderungen für ein Rahmenwerk. Wie in Kapitel 2.6 aufgezeigt, mangelt es zunächst am konkreten Verständnis des modellbasierten Systems Engineerings. Ferner sind die Modellierungsansätze rein technisch geprägt: Sie fokussieren einen begrenzten Aufgabenbereich der Produktentstehungsarbeit, schließen damit weite Teile des Systems Engineerings aus und vernachlässigen den sozio-technischen Charakter der Produktentstehung. Zusätzlich mangelt es an Entscheidungs- und Planungshilfen für die Modellierung. Der herausgearbeitete Handlungsbedarf für ein *Rahmenwerk zur Integration des modellbasierten Systems Engineerings in die Produktentstehung mechatronischer Systeme* hat also weiterhin Bestand.

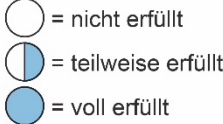

Bewertung der untersuchten Ansätze hinsichtlich der gestellten Anforderungen. Bewertungsskala:  Legende: PE = Produktentstehung SE = Systems Engineering  = nicht bewertbar/nicht relevant		Anforderungen								
		Systems Engineering als Basis	Systemmodell als Mittelpunkt	Klares MBSE-Konzept	Sprachen- und Methoden-unabhängigkeit	Integration von MBSE in die Organisation und Prozesse	Erfolgsprinzipien bekannter Managementansätze	Rationalitätssicherungsfunktion	Stakeholderorientierte Modellierung und Nutzung des Modells	Wohlfundierte Modellplanung
		A1	A2	A3	A4	A5	A6	A7	A8	A9
Beschreibungs-/Vorgehensmodelle	3-Zyklenmodell nach GAUSEMEIER									
	iPeM – Integriertes Produktentstehungs-Modell nach ALBERS									
	Modellbasierte virtuelle Produktentwicklung nach EIGNER									
	W-Modell nach NATTERMANN/ANDERL									
Ansätze zur Modellierung des Systemmodells	CONSENS nach GAUSEMEIER									
	SysMod nach WEILKIENS									
	Harmony SE nach IBM									
	OPM nach DORI									
	oosem nach OMG									
	LITHE nach RAMOS									
Frameworks	ISO 42010									
	DoDAF									
	KRUCHTEN 4+1									
	A3AO-Framework nach BORCHES									
Prozess-/ Methodenmodelle der PE	Forflow									
	Situative Entwicklungsaufgaben- und Methodenauswahl									
	PoMM nach BIRKHOFFER									
	Münchener Methoden-Modell									
Organisation	SE-Teams nach FRIEDENTHAL									
	SE in der Organisation nach PARNELL									

Bild 3-19 Bewertung Stand der Technik anhand der gestellten Anforderungen

4 Rahmenwerk

*Je mehr System-Ingenieure eine gemeinsame Begriffswelt akzeptieren und benutzen, desto mehr Fortschritt bei der Kommunikation, dem Verstehen und schließlich bei der Produktivität werden wir erfahren.“
[INC12]*

Dieses Kapitel bildet den Kern der vorliegenden Arbeit. In seinem Verlauf wird ein *Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme* erarbeitet. Die einzelnen Elemente des Rahmenwerks haben den Anspruch, dem aufgezeigten Handlungsbedarf aus Kapitel 2 und Kapitel 3 gerecht zu werden.

Das Kapitel ist wie folgt aufgebaut: In Kapitel 4.1 wird ein Überblick über das Rahmenwerk gegeben und seine Bestandteile vorgestellt. In den Kapiteln 4.2 bis 4.6 werden die Bestandteile detailliert erarbeitet. Alle Bestandteile des Rahmenwerks können weitgehend unabhängig voneinander angewendet werden. In Kapitel 4.6 wird die Möglichkeit aufgezeigt, sie zu verknüpfen. Das Rahmenwerk wird in Kapitel 4 konzeptionell entwickelt und – wo sinnvoll – um Anwendungsbeispiele ergänzt. Zusätzlich findet eine Anwendung und die dazugehörige Bewertung der Bestandteile in Kapitel 5 statt.

4.1 Überblick über das Rahmenwerk

MBSE ist bislang durch viele Stakeholder mit fachdisziplinspezifisch unterschiedlich ausgeprägten Sichten auf das Themenfeld gekennzeichnet. Die Folge: Es mangelt an einem einheitlichen Verständnis für MBSE und die damit verbundenen Möglichkeiten. Das Rahmenwerk hilft, das Themenfeld zu strukturieren. Darüber hinaus schafft es die Möglichkeit, die Nutzung des Systemmodells als Kern der Produktentstehung zielgerichtet zu planen und es für verschiedene Aufgaben im Produktlebenszyklus einzusetzen. Dabei stehen gerade die Stakeholder im Fokus, die bislang noch nicht ausreichend in den Ansatz eingebunden sind, deren Aufgaben sich aber mit dem klassischen Systems Engineering decken oder zumindest thematisch nahestehen. Das sind bspw. die Ansätze aus dem Technical Management nach [ISO15288].

In Kapitel 2 wurde gezeigt, dass im klassischen Systems Engineering zahlreiche gleichberechtigte Konzepte und Ausgestaltungen existieren. Der Vergleich legt also nahe, dass im Model-Based Systems Engineering ebenfalls mehrere gleichberechtigte MBSE-Konzepte existieren. Diese können bspw. in Abhängigkeit der Anwender, der Aufgabe oder weiterer Kriterien unterschiedlich Nutzen stiften. FRIEDENTHAL bekräftigt die für diese Arbeit grundlegende Annahme durch den Hinweis auf eine notwendige situationgerechte Anpassung des MBSE bei der Anwendung [FMS08], [TA15b]. Im Prinzip entspricht das dem im Systems Engineering üblichen Tailoring.

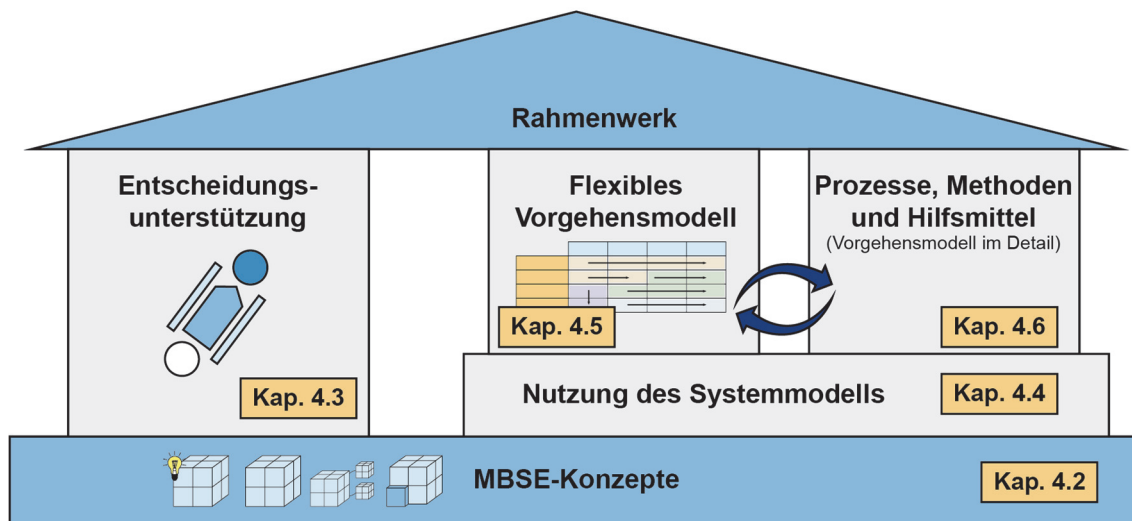


Bild 4-1 Das Rahmenwerk im Überblick

Bild 4-1 zeigt das Rahmenwerk im Überblick. Die Bestandteile werden zusammen als Rahmenwerk aufgefasst und in den folgenden Unterkapiteln schrittweise erarbeitet:

- Die situationsgerechte Anwendung von MBSE wird durch sog. **MBSE-Konzepte** ermöglicht (Kapitel 4.2). In Anlehnung an die Klassifizierungstheorie nach TAFRESCHI [Taf06] werden idealtypische Prototypen des MBSE entwickelt, hier *MBSE-Konzept* genannt. Ein MBSE-Konzept besteht aus einer konsistenten Kombination mehrerer MBSE-relevanter Merkmale und ihrer Ausprägungen, z.B. „*Formalisierungsgrad der Sprache*“ und „*Werkzeugart*“. Zur Herleitung wird der aus der Strategieentwicklung stammende Ansatz Vitostra verwendet [Bät04]⁶. Die MBSE-Konzepte erhöhen das Verständnis für MBSE und geben dem Paradigma Struktur, was die Planung und Anwendung des MBSE-Einsatzes unterstützt.
- Eine **Entscheidungsunterstützung** hilft bei der Auswahl und dem zielgerichteten Einsatz der Konzepte im Produktentstehungsalltag (Kapitel 4.3). Da bei der prototypischen Klassifizierung schon eine Ähnlichkeit des Klassifizierungsobjekts mit dem Prototypen die Zuordnung ermöglicht, dient die Entscheidungsunterstützung ergänzend und als Empfehlung. Damit hat sie zwar keinen verbindlichen Charakter, ermöglicht aber potentiellen und insb. unerfahrenen Anwendern eine rationale Entscheidung anhand unterschiedlicher Bewertungskriterien.
- In Kapitel 4.4 steht die **Nutzung des Systemmodells** für verschiedene Aufgaben im Vordergrund – es wird unterschieden in die Modellplanung, Modellierung,

⁶ Mit Vitostra können konsistente Strategiealternativen entwickelt werden. Dieser Ansatz hat bei Themenstellungen wie der Strategieentwicklung [Leh14] oder der Klassifizierung von Reifegradmodellen seine Eignung unter Beweis gestellt [Chr09]. Für das Model-Based Systems Engineering existieren hierzu bislang keinerlei Arbeiten jenseits der eigenen Vorarbeit [TBD+15].

Pflege und Anwendung des MBSE. Bei der Anwendung wird unterschieden in die technische Anwendung des Systemmodells und die Management-Anwendung. Angelehnt an das Verständnis, dass die Anwendung von Methoden durch die Aktivitäten der Informationsverarbeitung [PBF+05] dargestellt werden können, wird ein Konzept erarbeitet, das die Kopplung von Methoden und Aufgaben mit dem MBSE und dem Produktlebenszyklus beschreibt: Die Aktivitäten der Informationsverarbeitung werden zudem auf die ISO/IEC42010 übertragen; diese wiederum wird um eine Prozess- und Aufgabensicht ergänzt. Dadurch erhält das MBSE eine zeitliche Komponente; ebenso wird die Lücke zwischen den Aufgaben der Produktentstehung und dem MBSE geschlossen.

- Aufbauend auf Kapitel 4.4 wird in Kapitel 4.5 ein **Flexibles Vorgehensmodell** erarbeitet. Es leitet die Systemmodellierung an, mit Fokus auf die **Planung der Modellierung** und die **Anwendung von Methoden** auf Basis des Systemmodells. In einer 4x4-Matrix werden die Phasen des etablierten Management-Ansatzes Six Sigma mit den Aktivitäten der Informationsverarbeitung kombiniert. Jede der resultierenden 16 Matrix-Zellen enthält Anleitungen, Prozesse, Methoden und Hilfsmittel für o.g. Aufgaben. In der Matrix ergeben sich dadurch implizit die vier o.g. Hauptaufgabenbereiche mit Schwerpunkten in dieser Arbeit auf der Modellplanung und der Anwendung/Methodenkopplung. Die Hauptaufgabenbereiche können in Abhängigkeit verschiedener Rahmenbedingungen unterschiedlich durchlaufen werden.
- In Kapitel 4.6 werden zusätzlich zu den in den vorangegangenen Kapiteln erarbeiteten Elementen weitere **Methoden und Hilfsmittel** erarbeitet und in das flexible Vorgehensmodell aus Kapitel 4.5 integriert. Dadurch ergibt sich ein Baukasten, der insb. bei der Modellplanung und Methodenkopplung unterstützt, aber auch bei der Integration des MBSE in die Organisation hilft. Die Elemente werden am Beispiel der Stakeholderanalyse eines Kaffeeautomaten erklärt.

4.2 Konzepte des modellbasierten Systems Engineerings

In der Problemanalyse und im Stand der Forschung wurde gezeigt, dass das modellbasierte Systems Engineering zahlreiche Facetten und kein einheitliches Verständnis aufweist – erkennbar bspw. an der Vielfalt der Modellierungsmethoden. In diesem Kapitel werden MBSE-Konzepte entwickelt, die das Verständnis für MBSE fördern und seine Ausgestaltung für unterschiedliche Aufgaben, Situationen und Stakeholder unterstützen⁷. In Anlehnung an die Klassifizierungstheorie nach TAFRESCHI werden prototypische MBSE-Konzepte entwickelt [Taf06]. Anwendungen des modellbasierten Systems

⁷ Grundlegende Arbeiten hierzu wurden auf der IEEE-Konferenz SysCon 2015 – Systems Conference in Vancouver veröffentlicht [TBD+15].

Engineerings können dann als typischer Vertreter eines Konzepts interpretiert werden, wenn eine Ähnlichkeit mit einem MBSE-Konzept existiert.

Das Vorgehen basiert im Kern auf einer Konsistenzanalyse und einer Multidimensionalen Skalierung, über die einzelne Merkmale – bzw. ihre Ausprägungen – in konsistenten Merkmalklassen zusammengefasst werden. Ein MBSE-Konzept ist in diesem Verständnis eine konsistente Kombination mehrerer MBSE-relevanter Merkmalausprägungen. Zunächst werden deshalb MBSE-relevante Merkmale extrahiert und ihre möglichen Ausprägungen identifiziert (Kapitel 4.2.1). Die Merkmale werden in Anlehnung an das sozio-technische Verständnis der Produktentstehung unterteilt in technische und sozio-technische Merkmale. Die technischen Merkmale sind der Kernidee des MBSE zuzuschreiben und lassen sich bspw. aus dem MBSE-Dreieck ableiten (vgl. Kapitel 2.4). Sozio-technische Merkmale sind relevant für die Anwendung von MBSE durch den Nutzer oder in der Organisation. Bei der Auswahl der Merkmale bietet sich erfahrungsgemäß eine Beschränkung auf etwa 15 bis 20 Merkmale an. Die Merkmalausprägungen werden paarweise auf Konsistenz bewertet, gefolgt von dem o.g. Vorgehen. Als Ergebnis liegen konsistente Merkmalklassen vor, die jeweils als MBSE-Konzept detailliert und interpretiert werden (Kapitel 4.2.2).

4.2.1 Merkmale und Ausprägungen von MBSE

Die Berücksichtigung von technischen und sozio-technischen Merkmale ist der Ausgangspunkt für die Entwicklung der MBSE-Konzepte. Für die bessere Übersicht werden die einzelnen Merkmale in verschiedenen Hauptklassen zusammengefasst. Bei den technischen Merkmalen spielen die Hauptklassen Modellierungssprache, Modellierungsmethode und Werkzeug eine zentrale Rolle, da sie die konstituierenden Elemente des MBSE sind („MBSE-Dreieck“). Die relevanten technischen Aspekte sind also durch die Idee der fachdisziplinübergreifenden Modellierung selbst beeinflusst. Dem gegenüber stehen die sozio-technischen Merkmale in den Hauptklassen Organisation und Anwendung⁸. In Bild 4-2 ist die entsprechende Strukturierung der Hauptklassen und Merkmale als Übersicht dargestellt.

Insgesamt wurden auf Basis der Arbeiten in Kapitel 2 siebzehn Merkmale für die Definition von MBSE-Konzepten herausgearbeitet. Die Analyse der Merkmale zur Identifikation relevanter Merkmalausprägungen erfolgt in der Reihenfolge der in Bild 4-2 vermerkten Nummerierung.

⁸ Bei den sozio-technischen Merkmalen werden nur robuste Merkmale und Ausprägungen berücksichtigt, d.h. für die zukünftig keine Änderungen zu erwarten sind. Dagegen sind bspw. die Merkmale *Kompetenz* und *Modellierungszweck* in Kapitel 4.3. nicht robust, obwohl sie im Prinzip der Hauptklasse Anwendungsaspekte zugeordnet werden könnten. Zukünftig ist jedoch eine höhere Kompetenz im Themenfeld MBSE als heute zu erwarten, s.d. die Ausprägung „geringe Kompetenz“ dann keine Rolle mehr spielen würde. Die Anzahl der Modellierungszwecke ist wiederum nahezu unendlich.

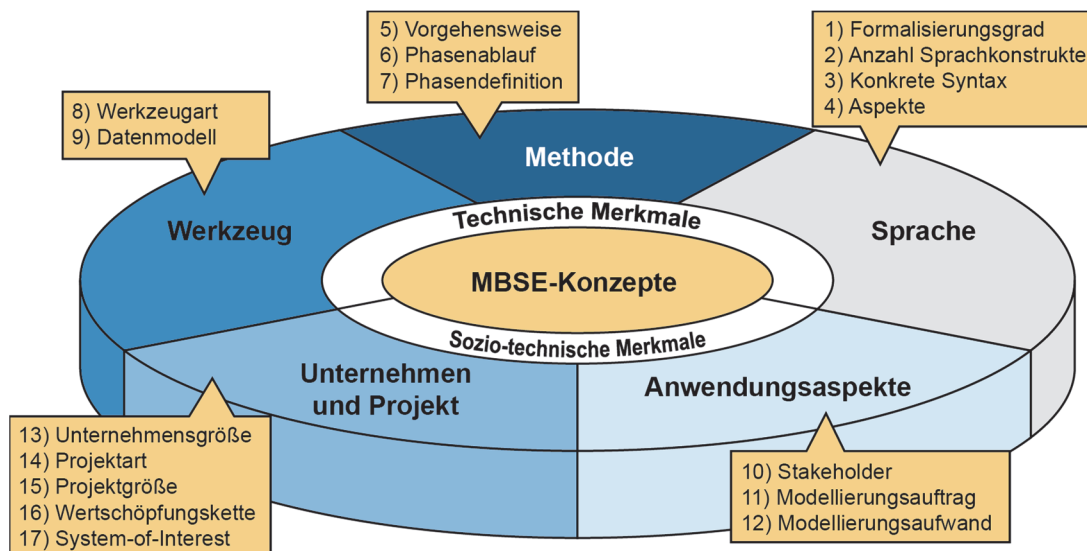


Bild 4-2 Struktur der Hauptklassen und Merkmale zur Bestimmung von MBSE-Konzepten

Formalisierungsgrad (Technische Merkmale, Sprache)

Sprache basiert auf einem Regelsystem von Syntax und Semantik (vgl. Kapitel 2.4). Die Syntax verknüpft in einer natürlichen Sprache Vorstellungen mit Lautbildern, die Semantik regelt die Anordnung der Lautbilder, bspw. im Satz. Die Genauigkeit dieser Sprachdefinition beeinflusst den Kommunikationserfolg. Künstliche Sprachen werden mit einem Zweck entwickelt und können in Plansprachen und formale Sprachen unterschieden werden. Damit sie interpretierbar sind, muss die Struktur einer künstlichen Sprache ebenfalls Syntax und Semantik aufweisen. Deshalb wird bei künstlichen Sprachen in der Entwicklung ein Regelsystem definiert – das Metamodell. Formale Sprachen zeichnen sich zusätzlich durch einen strikt logischen Algorithmus aus, dem mathematische Modelle von Grammatiken zu Grunde liegen.

Die Sprachen des modellbasierten Systems Engineerings beinhalten gleichzeitig Aspekte einer formalen Sprache und einer Plansprache, teilweise sogar ergänzt um Eigenschaften einer natürlichen Sprache. Ihre Modelle verknüpfen durch ihre abstrakte, grafische Darstellung der Wirklichkeit Lautbilder und Vorstellung.

Entscheidend für eine Klassifikation ist der Formalisierungsgrad der Sprache als Maß an formalen Regeln der Sprachsyntax und -semantik. Der Formalisierungsgrad hat Einfluss auf die Anwendung der Sprache und die weitere Verwendung. Mit steigendem Formalisierungsgrad nimmt die Möglichkeit zu, Sprache für einen Rechner interpretierbar zu machen; formale Sprachen überfordern jedoch häufig den menschlichen Anwender aufgrund der detaillierten und komplexen Dateneingabe [Bal99]. Ein Mehr an Formalisierung bedingt damit auch eine geringere Verständlichkeit. Genaugenommen ist die bekannteste Sprache des modellbasierten Systems Engineerings SysML eine formale Sprache. Das andere Extremum sind nicht-formale Sprachen; sie haben kein Regelwerk, weshalb die Darstellungsformen kaum eingeschränkt sind. Einfache Schaubilder,

z.B. Pfeildiagramme, sind Beispiele für Spezifikationen mit informalen Sprachen, ebenso wie sämtliche Beschreibungen mit natürlicher Sprache.

Die Sprachen des modellbasierten Systems Engineerings werden in der Regel als sog. semi-formale Sprachen bezeichnet. Damit platzieren sie sich im Prinzip zwischen den Extrema *formal* – *nicht-formal*. Sie haben eine definierte, eindeutige abstrakte und konkrete Syntax. Das kann bspw. eine grafische Notation sein, ausgestattet mit präzisen Regeln zur Spezifikation oder eine rein textuelle Notation mit ähnlichen Regeln. Auf dieser Basis lassen sich für die Klassifikation drei Merkmalausprägungen ableiten: *nicht-formale Sprache* – *semi-formale Sprache* – *formale Sprache*.

Anzahl der Sprachkonstrukte (Technische Merkmale, Sprache)

Die Anzahl der Konstrukte einer Sprache wird in der abstrakten Syntax definiert (vgl. Kapitel 2.2.4). Unter der Anzahl der Sprachkonstrukte wird der Umfang an unterschiedlichen Symbolen der Sprache verstanden; diese beeinflusst den Sprachumfang. Umfangreiche Sprachen helfen, Systeme differenziert und detailliert darzustellen, sind im Umkehrschluss jedoch schwerer zu erlernen [MHM10]. Für eine Klassifikation kann nur qualitativ unterschieden werden, da die Anzahl der Sprachkonstrukte relativ zu anderen Sprachen zu sehen ist. Daher wird unterschieden in *sehr wenige Sprachkonstrukte* – *wenige Sprachkonstrukte* – *viele Sprachkonstrukte*. So hat die SysML [OMG12] bspw. viele, CONSENS wenige und die OPD/OPL [Dor02] sehr wenige Sprachkonstrukte.

Konkrete Syntax (Technische Merkmale, Sprache)

Die grafische Repräsentation der Sprachkonstrukte wird über die konkrete Syntax visualisiert (vgl. Kapitel 2.2.4). Experten sind der Meinung, dass grafische Spezifikationen verständlicher sind als textbasierte Spezifikationen; eine Kombination aus Grafik und Text ist wiederum verständlicher als rein grafische Sprachen [GD11], [Dor02]. Das unterschiedliche Aussehen der Sprachkonstrukte für unterschiedliche Bedeutungen beeinflusst zusätzlich die Erlernbarkeit und Interpretierbarkeit einer Sprache: Je deutlicher die Bedeutung der Symbole durch Formen und Farben herausgestellt wird, desto verständlicher wirkt die Sprache [MHM10]. Diese spezielle Eigenschaft der konkreten Syntax wird häufig als „Visual Syntax“ oder „Visual Notation“ bezeichnet, die mindestens unterschiedliche Form- und Farbgebung muss in der Sprachsyntax festgelegt sein. MOODY ET AL. stellen neun Prinzipien auf, nach denen eine unterschiedliche Visualisierung der Sprache gewählt werden kann [MHM10]. Die geometrischen formgebenden Unterschiede sollten signifikant sein, d.h. mit einem Blick klar unterscheidbar. Die Sprachkonstrukte können demnach mindestens drei unterschiedliche Ausprägungen aufweisen: *keine definierte Visual Syntax* – *Form oder Farbe* – *Form und Farbe*. Alle Ausprägungen sind bei existierenden Sprachen beobachtbar; erstere natürlich bei nicht-formalen Skizzen. Im Meta-Modell der Spezifikationstechnik CONSENS wird bspw. explizit die Definition der Farbe *rot* für Störbeziehungen festgelegt, wohingegen die anderen Beziehungen *schwarz* sind. Der Ansatz OPD/OPL unterscheidet nur Formen.

Aspekte (Technische Merkmale, Sprache)

Die Elemente der verschiedenen Sprachen zur fachdisziplinübergreifenden Systemmodellierung werden im Metamodell definiert. Hierbei werden unterschiedliche Aspekte betrachtet; dies sind nach KALLMEYER bei mechatronischen Systemen seine Struktur, sein Verhalten und seine Gestalt [Kal98]. In der Sprache SysML werden diese Aspekte als „Säulen der SysML“ bezeichnet und umfassen Anforderungen, Struktur, Verhalten und Parameter. Die tatsächlich verwendeten Aspektmodelle, bzw. Diagrammarten [Alt12] variieren in Abhängigkeit der Modellierungssprache. Bild 4-3 gibt hierzu ein Beispiel, aus dem die Herkunft der SysML und der OPD/OPL aus der Informatik ersichtlich werden, da sie nicht den für mechatronische Systeme notwendigen Aspekt Gestalt berücksichtigen. Generell lassen sich damit bei der Entwicklung technischer Systeme die Aspekte *Anforderungen – Struktur – Verhalten – Gestalt – Parameter* unterscheiden, die aber nicht für jede Entwicklungsaufgabe eine Rolle spielen:

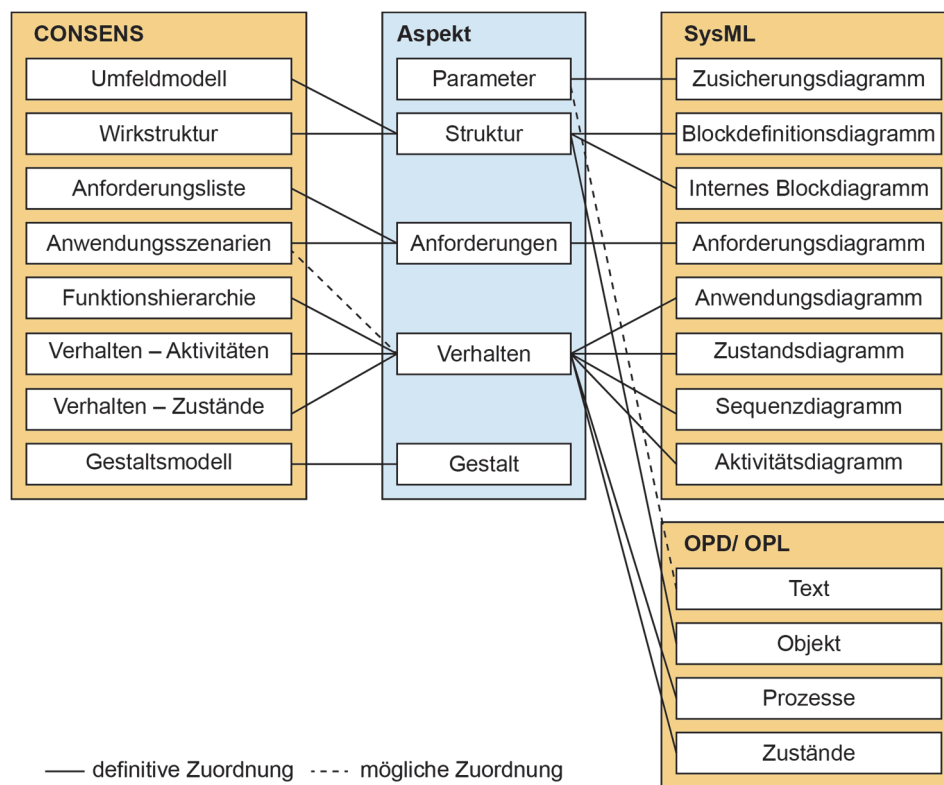


Bild 4-3: Adressierte Aspekte innerhalb verschiedener Sprachen (Beispiel)

Vorgehensweise (Technische Merkmale, Methode)

Im Vorgehensmodell nach HABERFELLNER wird das Vorgehen nach dem Top-Down-Gedanken als zweckmäßig betrachtet [HWF+12]. Das Betrachtungsfeld wird zunächst grob und dann erst mit fortlaufender Entwicklung enger gefasst. Grundsätzlich lassen sich im entwicklungsmethodischen Vorgehen zwei klassische Vorgehensweisen voneinander unterscheiden [PBF+07], [HWF+12]: Neben dem *Top-Down*-Ansatz, der bei Neuentwicklungen angewendet werden sollte, existiert ein *Bottom-Up*-Vorgehen – im

Prinzip ein Vorgehen ausgehend von existierenden Komponenten, die zu einem System integriert werden. Als Drittes lässt sich das *Zig-Zagging* bezeichnen, das Hin- und Her-springen mit unterschiedlichen Abstraktionsgraden [Wei08].

Phasenablauf (Technische Merkmale, Methode)

Unter Phasenablauf wird die Vorgabe zur Abfolge der Arbeitsschritte einer Methode verstanden. Im Prozess können die einzelnen Phasen schrittweise durchlaufen werden und/oder in regelmäßigen Abständen wiederkehren. Der Phasenablauf einer Methode kann demnach die Ausprägungen: *linear* oder *iterativ* annehmen. In diesem Kontext werden Methoden als iterativ betrachtet, wenn die Modellierungsmethode keine abgeschlossenen Phasen definiert. Streng lineare Methoden sind einfacher in der Anwendung. Iterative Methoden hingegen ermöglichen den Anwendern neue Erfahrungswerte in vorhergegangene Entwicklungsphasen mit einfließen zu lassen. In der modernen Entwicklung gibt es Methoden mit sowohl linearen als auch iterativen Phasenablauf [RFB12b]. Der Phasenablauf ist Bestandteil der Vorgehensweise einer Methode.

Phasendefinition (Technische Merkmale, Methode)

Unter der Phasendefinition wird die Genauigkeit der Vorgaben zu den einzelnen Arbeitsschritten einer Methode verstanden. Die Definition der Phasen hat Einfluss auf die Anwendung einer Methode. Methodisches Vorgehen ist nach BALZER „eine Gradwanderung zwischen Formalismus und Formlosigkeit“. Methoden die eher formlos sind begünstigen die Kreativität auf Kosten eines strukturierten Vorgehens. Zu formale Methoden dämmen zwar einen chaotischen und undurchsichtigen Ablauf ein, hemmen jedoch zugleich eigene Ideenansätze [Bal99]. Die Phasendefinition einer Modellierungsmethode wird auf einer Achse eingetragen, mit deren Extrema: *formlos* – *formal*. OPM von DORI ist ein Beispiel für ein sehr formales Vorgehen,

Werkzeugart (Technische Merkmale, Werkzeug)

Für das modellbasierte Systems Engineering existieren aufgrund der großen Nähe zum modellbasierten Softwareentwurf UML-Werkzeuge, die durch entsprechende Profile auf beliebige Sprachen des MBSE angepasst werden können. Grundsätzlich hängt die Wahl des Werkzeuges aber von der konkreten Situation ab, in der MBSE angewendet werden soll [Wei08]. In Anlehnung an GAUSEMEIER [GKP+10] und EIGNER [ERZ14] werden für die Klassifikation vier Merkmalausprägungen definiert

- **Brown Paper:** Brown Paper ist bspw. für Workshops geeignet [Wei08] und umfasst in Anlehnung an das Lean Management sämtliche Papiernotationen, z.B. die DIN A3 Bögen im A3AO nach [Bor10].
- **Diagramming Tools:** Das ist übliche Büro-Software, mit denen bspw. Präsentationen erstellt werden. Diese haben – vereinfacht ausgedrückt – ein simples Datenmodell, das sog. Shapefiles nutzt. Die hierin enthaltenen Informationen können aber für einfache Aufgaben ausgelesen und weiterverwendet werden.

- Commercial-off-the-Shelf (COTS): Diese sind meist auf breite Anwendungsbereiche ausgelegt und bedürfen immer einer speziellen Anpassung an Workflows und branchenspezifische Vorgehensweisen. Es sind in der Regel UML-Werkzeuge. Oft haben COTS-Werkzeuge Schnittstellen zu anderen Werkzeugen, was die horizontale und vertikale Durchgängigkeit der Entwicklung unterstützt – allerdings ist das i.d.R. auf die Softwareentwicklung beschränkt, wengleich Ansätze existieren, die Informationen der COTS-Werkzeuge mit Werkzeugen anderer Disziplinen zu koppeln. Die meisten SysML-Werkzeuge sind COTS. Auch dedizierte Modellierungswerkzeuge, wie z.B. der Mechatronic Modeller werden hier als COTS aufgefasst, da sich solche Systeme im Prinzip auch einfach über den Profilmechanismus der SysML in COTS erzeugen lassen.
- Simulationstool: Die Simulationstools im vorliegenden Sinne sind bspw. Produkte wie Matlab Simulink. Sie simulieren durchaus Verhalten auf der Systemebene, gehen aber genaugenommen von einer Bottom-Up-Sicht aus. Derartige Systeme sollen bspw. nachgeschaltet zu den COTS gekoppelt werden.

Genaugenommen sind die vier vorgestellten Merkmalausprägungen jedoch nicht direkt vergleichbar. Zunächst müssten eigentlich Modellierungs- bzw. Synthesewerkzeuge und Analysewerkzeuge unterschieden werden [GHK+06], [ERZ14]. Danach ergibt sich die in Bild 4-4 dargestellte Unterteilung, die im Kontext des MBSE eine plausible Zuordnung ermöglicht. COTS können sowohl zur Synthese als auch zur Analyse herangezogen werden. Werkzeuge wie Matlab Simulink werden oft als Synthesewerkzeuge bezeichnet – aus der Perspektive des Systems Engineering handelt es sich aber um Analysewerkzeuge, da Simulation eine Analysetätigkeit ist [ERZ14].

Modellierung- bzw. Synthesewerkzeuge		Analysewerkzeuge	
Brownpaper	Diagramming Tool	COTS	Simulationstool
			

Bild 4-4: Werkzeugarten im Kontext der Systemmodellierung

Datenmodell (Technische Merkmale, Werkzeug)

Das Systemmodell als Kern der MBSE-Idee wird häufig auch als „Single Source of Truth“ bezeichnet. In ihm sollen alle fachdisziplinübergreifenden Informationen über das System dargestellt werden. Konkrete Mindestanforderungen an das Systemmodell existieren jedoch nicht, was zu einem uneinheitlichen Verständnis des MBSE führt: Mit Bezug auf die grafische Modellierung ist die Visualisierung der fachdisziplinübergreifenden Informationen das Modell – in Anlehnung an STACHOWIAK ist bereits eine Papierskizze ein Modell. Die Entwicklung der Softwaretechnik hat zu einer weiteren Mei-

nung geführt: Hier wird von einem Systemmodell gesprochen, wenn ein Datenmodell als Sammlung aller relevanten Informationen vorliegt. In den COTS-Werkzeugen wird dieses Datenmodell häufig *Repository* oder Datenbank genannt. Die grafische Darstellung der Systemzusammenhänge ist dann lediglich die grafische Repräsentation einer Datenbank. Im Sinne des klassischen SE sollen beide Ausprägungen als geeignet für das modellbasierte Systems Engineering sein; als Merkmalausprägung wird der Einfachheit halber unterschieden in *Datenmodell vorhanden* – *kein Datenmodell vorhanden*.

Stakeholder (Sozio-technische Merkmale, Anwendung)

MBSE adressiert sämtliche Unternehmensbereiche, wodurch eine Vielzahl an Schnittstellen entsteht. Die Anzahl und Verschiedenartigkeit der möglichen Stakeholder von MBSE ist damit prinzipiell unendlich. Zudem zieht jeder Bereich einen anderen Nutzen aus der fachdisziplinübergreifenden Modellierung: Der Technische Vertrieb kann bspw. bei der Klärung von Aufträgen mit einfachen Mitteln Use Cases und Anwendungsszenarien generieren, die für andere Unternehmensbereiche Mehrwert bieten. Für die Klassifikation können der Einfachheit halber sämtliche potentielle Stakeholder unter den drei Merkmalausprägungen *Technik* – *Unternehmensleitung* – *Befähiger* zusammengefasst werden. Technik umfasst zumindest alle fachdisziplinspezifischen Bereiche. Unter Enabler werden hier bspw. der Technische Einkauf, der Service-Bereich, Qualität, Produktmanagement, das Projektmanagement und der Architekt verstanden – also Stakeholder, die stark in das Management und die Unterstützung von Produktentstehungsprojekten involviert sind. Nach CLOUTIER ist der Nutzen der Modellierung insb. für die beiden letztgenannten Stakeholder sehr hoch [BC10].

Modellierungsverantwortung (Sozio-technische Merkmale, Anwendung)

Die Anwendung von MBSE kann auf unterschiedliche Weisen gelingen und ist stark von der Unternehmensorganisation und der Situation im Lebenszyklus abhängig. Grundsätzlich kann die Erarbeitung der relevanten Informationen für das Systemmodell durch sämtliche Disziplinen geschehen. Mit Bezug auf [TKD+14] und Bild 2-26 kann ein Großteil der Modellierungsarbeit oder -unterstützung im Bereich der Fachdisziplin liegen: So unterstützt der *Fachspezialist* zu einem Drittel die Erstellung des Systemmodells, bspw. im Rahmen der Komponentenspezifikation. [Kai14] sieht den sog. Systemmodellverantwortlichen als Haupthandlungsperson, der die Modellierung in Workshops verantwortet. ZINGEL nennt den Modellierspezialisten bzw. das Kernteam [Zin13]. Beide Sichten werden zusammengefasst in der Rolle des Architekten (vgl. [She96]). Somit lassen sich für das Merkmal Modellierungsverantwortung die Ausprägungen *Workshop* – *Architekt* – *Spezialist* definieren.

Modellierungsaufwand (Sozio-technische Merkmale, Anwendung)

Der Modellierungsaufwand steht generell im Zusammenhang mit der Anwendung der Sprache und zusätzlich mit ihrer Anwendung für unterschiedliche Zwecke. Der Grund: Die Modellierung mit einer formalen Sprache mit dem Zweck der automatisierten

Code-Generierung erzeugt einen höheren Aufwand als die Anwendung einer weniger formalen Sprache zur interdisziplinären Kommunikation, bspw. im Rahmen der Phase Planen und Klären der Aufgabe. Damit sollte der Modellierungsaufwand zumindest qualitativ unterschieden werden in *niedrig – mittel – hoch*.

Unternehmensgröße (Sozio-technische Merkmale, Organisation)

Die Unternehmensgröße ist ein Merkmal, das erhebliche Auswirkungen auf den Einsatz von MBSE hat. In kleinen Unternehmen existieren kurze Kommunikationswege, persönliche Abstimmungen sind einfacher möglich – das impliziert andere Möglichkeiten als es bspw. in Konzernen mit Entwicklungsabteilungen in unterschiedlichen Zeitzonen der Fall ist. Da es keinen weltweit einheitlichen Bewertungsmaßstab für die Größe eines Unternehmens gibt, wird der Einfachheit halber zwischen *kleinen Unternehmen* bis etwa 500 Mitarbeitern und *großen Unternehmen* mit mehr als 500 Mitarbeitern unterschieden. Die Unternehmensgröße bezieht sich hierbei auf Unternehmensbereiche, die theoretisch von der Arbeit mit MBSE profitieren können und entspricht damit nicht zwingend der Gesamtgröße des Unternehmens.

Projektart (Sozio-technische Merkmale, Organisation)

Der Einsatz von MBSE für konkrete Projektarten ist ein weiteres wichtiges Merkmal. Idealerweise beginnt der Einsatz in der *Vorentwicklung*. Diese Projekte zeichnen sich durch einen hohen Unsicherheitsgrad aus und beziehen sich selten auf ganze Systeme, sondern meist auf Module. Hier werden notwendigerweise auch noch viele physikalische Tests durchgeführt. Bei gegebener technologischer Reife entstehen dann *Serienentwicklungsprojekte*, meist bei kundenanonymen oder variantenreichen Serienprodukten. Dem gegenüber stehen *kundenspezifische Entwicklungen*, bspw. im Maschinen- und Anlagenbau mit einer geringen Variantenanzahl und meist „Stückzahl 1“.

Projektgröße (Sozio-technische Merkmale, Organisation)

Bei dem Merkmal Projektgröße verhält es sich wie mit dem Merkmal Unternehmensgröße. Hier sollte jedoch als Merkmalausprägung die Anzahl möglicher Nutzer je Projekt in Betracht gezogen werden. [Kac13] und [BEJ06] zeigen, dass Entwicklungsprojekte selten mehr als 30 Teammitglieder umfassen und im Mittel meist bei etwa 10 Mitgliedern liegen. Aus diesem Grund werden drei Klassen für die Projektgröße unterteilt: *klein* (≤ 10 MA) – *mittel* ($10 < x < 30$ MA) – *groß* (≥ 30 MA).

Einbindung der Wertschöpfungskette (Sozio-technische Merkmale, Organisation)

MBSE sollte ein lebenszyklusbegleitender Ansatz sein. Heutige Anwendungen sind meist auf einzelne Projekte beschränkt, die Anwendung des MBSE ist aktuell noch meist ergänzend [FMS12] und nicht obligatorisch. Die Idee setzt aber eigentlich auf eine unternehmensübergreifende Anwendung entlang der Wertschöpfungskette. Zwischen diesen Extrema lässt sich noch eine unternehmensinterne Anwendung annehmen,

s.d. insg. drei mögliche Ausprägungen zur Anwendung in der Wertschöpfungskette möglich sind: *projektintern – unternehmensintern – wertschöpfungsstufenübergreifend*.

System-of-Interest (Sozio-technische Merkmale, Anwendung)

Die Art des in der Entwicklung befindlichen Systems „System-of-Interest – SoI“ hat Einfluss auf die Anwendung von MBSE. Es gilt im Prinzip darzustellen, auf welcher Produktstrukturstufe sich das SoI befindet. Das ist nicht immer eindeutig in Unternehmen geregelt und wird als künstliches Konstrukt nicht den technischen Merkmalen zugeordnet. Bei dem Merkmal sollten drei Abstufungen betrachtet werden: *System-of-Systems (S-o-S) – System – Modul/Komponente*. Ein Bauteil steht nicht im Fokus, da es die „kleinste nicht-zerlegbare Einheit eines Systems“ ist. Bei den drei Produktstrukturstufen ist davon auszugehen, dass bei der Anwendung von MBSE unterschiedliche Aufgaben im Vordergrund stehen: Bei S-o-S geht es u.a. um die strukturelle Einbindung in das übergeordnete Gesamtsystem und ggf. die Kommunikationsbeziehungen in unterschiedlichen Use Cases – jedoch durchaus auf sehr abstrakter Ebene. Bei einem Modul dagegen wird sehr ausführlich spezifiziert und auch fachdisziplinspezifische Informationen werden stärker berücksichtigt. Das Ziel der Modellierung ist dann bspw. die automatische Generierung von Software-Code.

4.2.2 Klassen konsistenter Merkmalausprägungen

Im Rahmen der vorangegangenen Analyse wurden siebzehn Merkmale für MBSE-Konzepte mit 50 Merkmalausprägungen identifiziert. Zur Ableitung der MBSE-Konzepte werden nun alle Merkmalausprägungen paarweise in einer Konsistenzmatrix auf Konsistenz bewertet. Hierbei wird untersucht, ob die einzelnen Merkmalausprägungen sich gegenseitig unterstützen oder ausschließen. Ein Beispiel: Bei Auswahl einer Modellierungssprache mit sehr hohem Formalisierungsgrad empfiehlt es sich nicht, Brown Paper zur Dokumentation zu nutzen; das ist nicht plausibel, da Aufwand und Nutzen in keinem angemessenen Verhältnis zueinander stehen. Beide Ausprägungen schließen sich also normalerweise gegenseitig aus und werden daher als inkonsistent bezeichnet. Stattdessen ist die Nutzung einer eher informalen Notation sinnvoller. Diese Kombination (Brown Paper und informale Notationsform) ist bspw. immer wieder im Rahmen von Arbeitstreffen zu beobachten. Die Kombination ist daher sehr konsistent.

Die Konsistenzbewertung der Merkmalausprägungen erfolgt anhand einer Skala von „1“ (totale Inkonsistenz – beide Merkmalausprägungen schließen sich gegenseitig aus) bis „5“ (beide Merkmalausprägungen verstärken sich positiv). Da es sich bei der Konsistenzbewertung um eine ungerichtete Beziehung handelt, muss nur eine Hälfte der Matrix bewertet werden. Die Auswertung der Konsistenzmatrix wird dann durch die verwendete Software vorgenommen: Hochkonsistente Kombinationen werden identifiziert und mit der Methode der Clusteranalyse zusammengefasst. Die ausgefüllte Konsistenzmatrix bildet die Grundlage für die weiteren beschriebenen Schritte.

Am Ende dieses Vorgehens liegen verschiedene theoretisch mögliche, konsistente MBSE-Konzepte vor. Die Konsistenzbewertung erfolgte zusammen mit erfahrenen Wissenschaftlern im Umfeld des Spitzencluster-Querschnittsprojekts Systems Engineering. Bild 4-5 zeigt einen Auszug aus der ausgefüllten Konsistenzmatrix.

Konsistenzmatrix		Ausprägungen						Werkzeugart				Projektgröße						
Fragestellung: „Wie verträgt sich Merkmalsausprägung i (Zeile) mit Merkmalsausprägung j (Spalte)?“		Bewertungsskala: 1 = totale Inkonsistenz 2 = partielle Inkonsistenz 3 = neutral oder voneinander unabhängig 4 = gegenseitiges Begünstigen 5 = starke gegenseitige Unterstützung																
Merkmalsausprägung	Nr.	1a	1b	1c	2a	2b	2c	8a	8b	8c	8d	15a	15b	15c	16a	16b	16c	
Formalisierungsgrad	informat	1a																
	semi-formal	1b																
	formal	1c																
Anzahl Sprachkonstrukte	sehr wenige	2a	1	3	2													
	wenige	2b	3	3	2													
	viele	2c	5	5	5													
Werkzeugart	Brown Paper	8a	5	4	2	5	4	1										
	Diagramming Tools	8a	4	4	2	5	3	2										
	COTS/ Delicated Tools	8a	1	4	4	5	5	2										
	Simulationswerkzeug	8a	1	3	5	2	2	5										
Projektgröße	klein	15a	5	4	2	5	5	2	4	5	2	3						
	mittel	15b	2	5	3	5	5	4	3	5	4	3						
	groß	15c	1	1	4	2	5	3	2	3	5	4						
Wertschöpfungskette	projekintern	16a	4	5	3	4	3	3	5	4	3	5	5	4	2			
	unternehmensintern	16b	2	5	3	3	3	4	2	5	5	4	3	4	5			
	Wertschöpfungskette	16c	1	4	5	2	4	2	1	5	4	2	1	4	4			
System-of-Interest	System-of-System	17a	4	5	2	4	4	2	4	4	2	1	1	4	5	1	3	5
	System	17b	2	5	3	3	4	2	3	5	5	2	3	4	5	2	4	4
	Modul	17c	1	3	5	1	4	5	2	2	4	5	4	3	2	5	2	1

1 Eine „informale“ Notation und die Anwendung von „Simulationswerkzeugen“ schließen sich gegenseitig aus, es liegt eine totale Inkonsistenz vor.

5 Große Projekte sind sehr gut geeignet für den Einsatz von COTS-Software, bzw. eines dedizierten Modellierungswerkzeugs – und umgekehrt.

Bild 4-5 Auszug aus der Konsistenzbewertung der Merkmalausprägungen

Nach Anwendung der Software und Bestimmung einer geeigneten Anzahl an Clustern mit Hilfe eines Scree-Diagramms ergeben sich vier mögliche MBSE-Konzepte⁹. Mit

⁹Genaugenommen ergeben sich auf Basis der Konsistenzmatrix zunächst sog. Projektions- bzw. Ausprägungsbündel. Bei der vorliegenden Anzahl an Merkmalausprägungen entstehen einige Millionen möglicher Bündel. Mit der verwendeten Software Vitostra wird eine Reduktion der Bündel vorgenommen, s.d. schließlich nur noch einige etwa hoch konsistente Bündel vorliegen. Da viele dieser Bündel ähnlich sind, werden sie aus Gründen der Komplexitätsreduktion noch einmal mit einer Clusteranalyse zusammengefasst und das Ergebnis hier als MBSE-Konzept interpretiert. Vgl. hierzu auch [Bät04].

dem Scree-Diagramm ergibt sich, dass der Informationsverlust in Relation zur Anzahl der Cluster bei vier Clustern möglichst gering ist. Die Cluster werden mittels der multidimensionalen Skalierung (MDS) in einem mehrdimensionalen Raum visualisiert und zusätzliche als zweidimensionale Darstellung bereitgestellt.

Die zweidimensionale Darstellung der Cluster ist in Bild 4-6 dargestellt. Ähnliche Cluster liegen eng beieinander, je größer der Unterschied zwischen den Clustern (Konzepten), desto größer also ihre Entfernung zueinander. Konzept 1 und Konzept 4 sind also maximal unterschiedlich, was auch durch die beispielhaft eingezeichneten Hauptunterscheidungsmerkmale deutlich wird; Konzept 1 setzt auf einen Brown Paper-Ansatz, wohingegen Konzept 4 durch Simulationswerkzeuge geprägt ist. Zusätzlich zur Entfernung der Konzepte untereinander haben die Kugeln eine besondere Bedeutung: Jede Kugel ist ein konsistentes Bündel, ihr Durchmesser ein Indikator für seine Konsistenz.

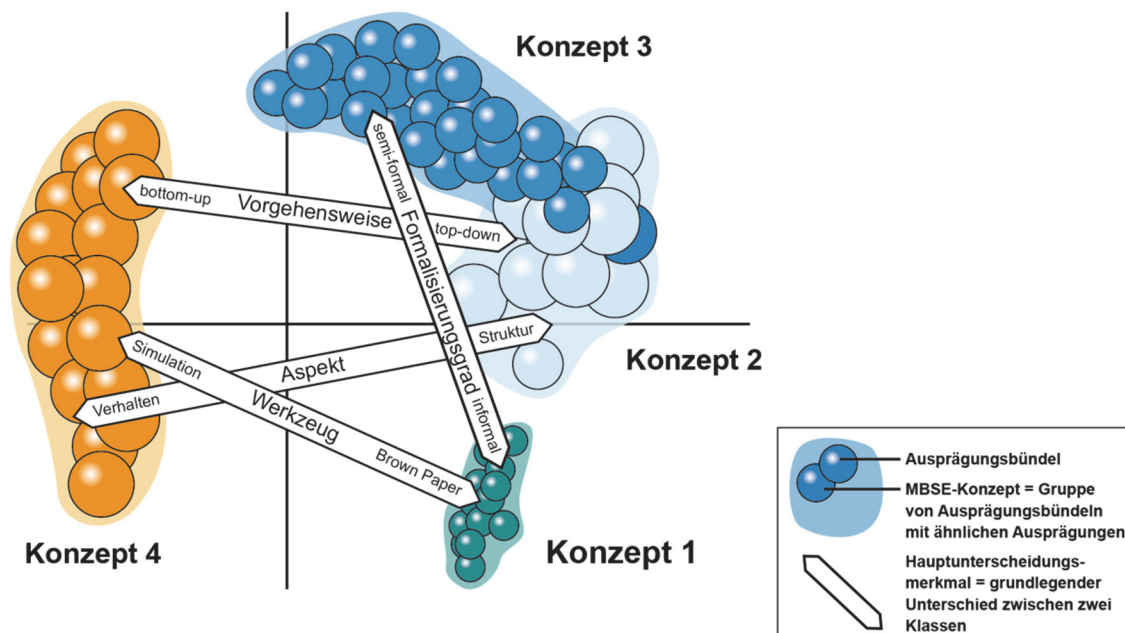


Bild 4-6: Visualisierung der MBSE-Konzepte mittels einer Multidimensionalen Skalierung (MDS) und Beispiele wesentlicher Unterscheidungsmerkmale

4.2.3 Detaillierung und Analyse der MBSE-Konzepte

Die Visualisierung der MBSE-Konzepte in Bild 4-6 mit der multidimensionalen Skalierung hat einen ersten Eindruck über die Schwerpunkte der Konzepte und ihre Unterschiede vermittelt. Zusätzlich zu der zweidimensionalen Darstellung der Konzepte erstellt das verwendete Werkzeug automatisch eine Ausprägungsliste mit den Ergebnissen der Konsistenzanalyse. Diese enthält also die identifizierten Merkmale inklusive der dazugehörigen Ausprägungen und gibt Aufschluss über die Auftretenshäufigkeit einer konkreten Ausprägung in den Konzepten. Bild 4-7 zeigt die Ausprägungsliste mit einem

Auszug markanter Ausprägungen der 17 Merkmale¹⁰. Mit diesen Ausprägungen kann nun die „DNA“ der MBSE-Konzepte herausgearbeitet werden.

Merkmal	Ausprägungen	Konzept	Konzept	Konzept	Konzept
		1	2	3	4
Formalisierungsgrad	A informal	70	0	0	0
	B semi-formal	30	100	64	4
	C formal	0	0	35	95
Aspekte	A Anforderungen	100	20	8	0
	B Struktur	0	80	62	6
	C Gestalt	0	0	0	16
	D Verhalten	0	0	27	62
	E Parameter	0	0	1	16
Vorgehensweise/ Phasendefinition	A Top-Down	80	100	93	0
	B Bottom-Up	0	0	0	100
	C Zigzagging	20	0	6	0
Werkzeugart	A Brown Paper	70	0	8	0
	B Diagramming Tools	30	60	32	0
	C COTS/Dedicated Tools	0	40	59	20
	D Simulationswerkzeug	0	0	0	79
Reichweite Wertschöpfungskette	A projektintern	100	0	0	100
	B unternehmensintern	0	0	30	0
	C Wertschöpfungskette	0	100	69	0
System-of-Interest	A S-o-S	0	0	24	0
	B System	100	100	75	0
	C Modul	0	0	0	100

eindeutige Ausprägung
 dominante Ausprägung
 alternative Ausprägung
 schwache Ausprägung/ Ausprägung tritt nicht auf

Bild 4-7: Ausprägungsliste der MBSE-Konzepte (Auszug)

Bei der Analyse der MBSE-Konzepte wird schnell deutlich, dass jedes Konzept einen eindeutigen Charakter besitzt und unterschiedliche Schwerpunkte legt. So verändert sich bspw. beim Merkmal Formalisierungsgrad der Fokus über die Konzepte, was anhand des prozentualen Anteils einer Merkmalausprägung in einem Konzept deutlich wird: Die Merkmalausprägung *semi-formal* ist zu 100 Prozent in Konzept 2 enthalten, was einer eindeutigen Ausprägung gleichkommt. Ebenso ist die Ausprägung *formal* mit 95 Prozent dominant in Konzept 4.

Markant ist auch, dass jedes MBSE-Konzept bei den „Säulen des MBSE“ einen anderen Schwerpunkt legt; diese Schwerpunkte sind in Bild 4-8 auf Basis der Ausprägungsliste in einem Balkendiagramm dargestellt. Je nach gewählter MBSE-Methode ist der Schwerpunkt ein Hinweis dafür, welche Diagramme oder Aspektmodelle besonders im

¹⁰ Eine vollständige Ausprägungsliste über alle Merkmale befindet sich im Anhang A2.2.

Fokus der Modellierung stehen sollten: Konzept 1 bezieht sich bspw. auf die Anforderungen. Für CONSENS umfasst dies also insb. die Anforderungsliste und die Anwendungsszenarien, aber auch das Umfeldmodell. Konzept 2 ist sehr strukturorientiert (z.B. Wirkstruktur), Konzept 3 nimmt zusätzlich noch das Verhalten (z.B. Aktivitätsdiagramm) auf. Eine gewisse Ähnlichkeit der beiden Konzepte fällt ebenfalls in der MDS in Bild 4-6 durch die räumliche Nähe der beiden Wolken auf.

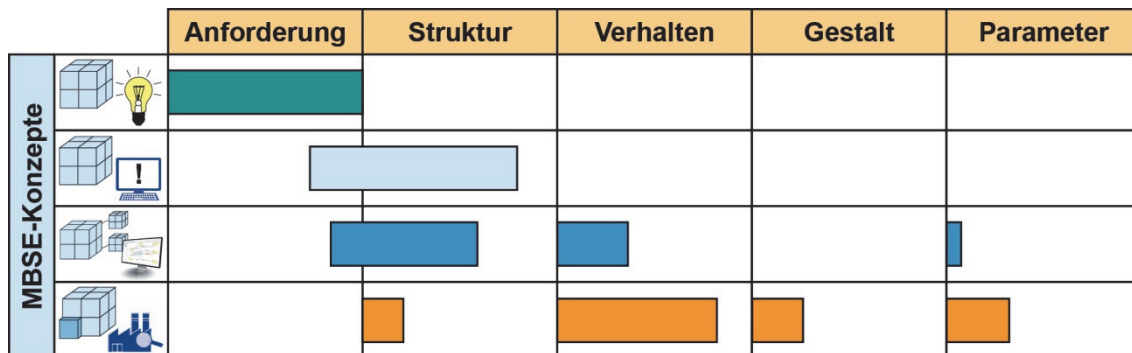


Bild 4-8 *Schwerpunktbildung auf unterschiedliche Aspekte – "Säulen des MBSE"*

Unter Berücksichtigung der durch CLOUTIER genannten Ziele der Modellierung (Problemverständnis, Kommunikation, Komplexitätsbeherrschung) und der übergeordneten Vision des SE, die Produktentstehung auf eine neue Leistungsstufe zu heben (Innovation, vollständige Virtualisierung) können die vier Konzepte für die drei Ziele Innovation (=Problemverständnis, Innovation), Kommunikation und Koordination, sowie Virtualisierung verteilt Nutzen stiften. Bild 4-9 bringt dieses Verständnis zum Ausdruck. So lässt sich über alle Ziele nun eine detailliertere Charakterisierung der Konzepte ableiten. In der Szenario-Methodik – dessen Vorgehen hier adaptiert wurde – werden die einzelnen Szenarien zugunsten der schnelleren Verständlichkeit visualisiert – z.B. mittels Collagen in sog. Zukunftsbildern [GPW14]. In Anlehnung an dieses Vorgehen werden im Anhang A1-3 die verschiedenen MBSE-Konzepte ebenfalls visualisiert; dabei werden gerade die dominanten und die eindeutigen Merkmalausprägungen berücksichtigt.

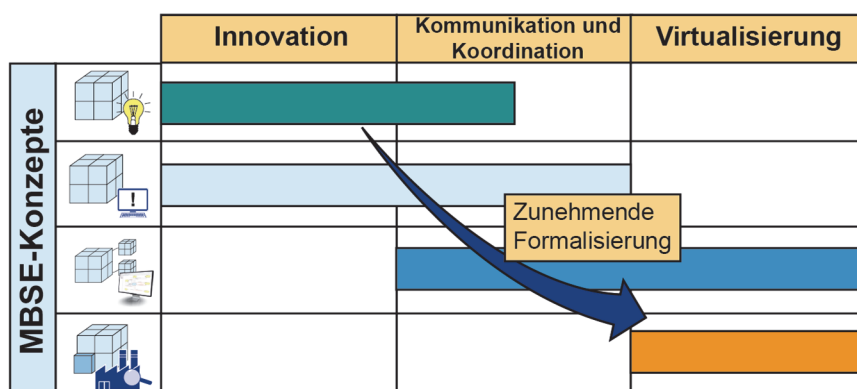


Bild 4-9 *Ziel des MBSE und Eignung der MBSE-Konzepte*

Hier dient zunächst eine Charakterisierung in Prosaform. Die Namen der Konzepte vermitteln dabei schon einen Hauptanwendungszweck:

- **Konzept 1 – Innovationswerkstatt:** Das Konzept ist durch seine gute Unterstützung für die Ideenfindung und beim Planen und Klären der Aufgaben in Systementwicklungsprojekten gekennzeichnet. Dafür sprechen zunächst die Erarbeitung von anforderungsrelevanten Aspekten mit klassischen Hilfsmitteln für Workshops wie dem Brown Paper und einer informalen Sprache mit wenig definierter Visual Syntax – was die Kreativität unterstützt. Die Ergebnisse werden zusätzlich mit üblichen Diagramming-Tools gesichert. Das Vorgehen ist sehr iterativ, von der Idee her top-down aber mit einem gewissen zigzagging-Anteil, was auch durch die formlose Phasendefinition unterstrichen wird. Durch die nicht immer eindeutige Dokumentation ist das Vorgehen insb. für kleine Vorentwicklungsprojekte gedacht, zur Anwendung innerhalb des Teams. Das begrenzt den Aufwand. Die Zusammenarbeit wird somit durch ein eher informales „paper-based“ MBSE unterstützt.
- **Konzept 2 – Mechatronic Sketch [TA15b]**¹¹ zeichnet sich durch seinen klaren Fokus auf die mechatronische Struktur des Systems aus. Es basiert auf einer semi-formalen Sprache mit klarer, eindeutiger Visual Syntax. Die Anzahl an Sprachkonstrukten ist gut überschaubar und deshalb auch mit Diagramming-Tools noch handhabbar. Das methodische Vorgehen bei der Modellierung ist gemäß Systems Engineering ein top-down-Ansatz, erlaubt jedoch Iterationen. Innerhalb einer jeden Phase wird sehr konkret vorgegeben, was zu tun ist. Aus methodischer Sicht eignet sich der Ansatz für den Einsatz über Wertschöpfungsstufen von kleinen Unternehmen hinweg bei mittelgroßen, kundenspezifischen Systementwicklungsprojekten. Verantwortlich für die Modellierung ist der Systemarchitekt, der die Modelle insb. für Adressaten wie z.B. den Einkauf, das Projektmanagement, das Produktmanagement oder den Vertrieb erstellt und dabei einen hohen Pflegeaufwand akzeptieren muss.
- **Konzept 3 – Mechatronische Systemmodellierung:** Das Konzept entspricht zunächst weitgehend der Idee, die bislang mit MBSE verbunden wird. Auf Basis einer sehr genau definierten Sprache mit vielen Sprachkonstrukten und eindeutiger Visual Syntax wird insb. die mechatronische Struktur des Systems definiert. Es existiert aber auch eine starke Tendenz zur Spezifikation des Systemverhaltens und eine erste stärkere Berücksichtigung von Parametern. Durch die überwiegende Nutzung von COTS mit einem Datenmodell wird der Grundstein für die PLM-Integration gelegt. Das Top-Down Vorgehen bei der Modellierung ist sehr sequentiell und insb. durch formales Arbeiten in den einzelnen Phasen geprägt. Damit können große Serienentwicklungsprojekte in großen Unternehmen beherrscht werden, die auch stark in Wertschöpfungsnetzwerken agieren. Das erklärt den leichten Fokus auf System-

¹¹ In Anlehnung an die häufige Bezeichnung für die Anwendung der Spezifikationstechnik CONSENS in Industrieprojekten, die eine ähnliche Merkmalausprägung aufweist und auch an [TA15b].

of-Systems. Es sollte hier jedoch in gewissem Grad als „Integrationsprojekt“ verstanden werden, also eine stark zerteilte Wertschöpfung unter Einbindung von Modul- und Komponentenlieferanten. Das wird unterstrichen durch den hohen Aufwand bei der Modellierung und Modellpflege, der in erster Linie von einem Architekten getragen werden soll. Die Adressaten der Modelle entstammen weitgehend dem Kreis der Enabler, also bspw. Einkäufer.

- **Konzept 4 – Simulationsbasierte Systemanalyse:** Dieses MBSE-Konzept ist stark anders als die vorangegangenen Konzepte ausgerichtet, was auch durch die Lage in der MDS zum Ausdruck kommt. Der Fokus dieses MBSE-Konzept liegt im Gegensatz zu den anderen Konzepten auf dem Verhalten eines Moduls. Mit Hilfe einer sehr formalen Sprache und vieler Sprachkonstrukte wird so in einem Bottom-Up Ansatz ein Modell durch Spezialisten einzelner Fachbereiche für Fachbereiche erzeugt. Das wird deutlich unterstrichen durch den Einsatz von Simulationswerkzeugen, der projektintern stattfindet. Dieser Ansatz dient somit der Analyse und wird aufgrund des hohen Aufwands bei der Modellierung größtenteils in der Serienentwicklung großer Unternehmen eingesetzt.

4.3 Entscheidungsunterstützung zur Auswahl und Anwendung von MBSE-Konzepten

Die MBSE-Konzepte fördern ein einheitliches Verständnis für das modellbasierte Systems Engineering. In diesem Kapitel wird die Auswahl und Anwendung eines Konzepts unter verschiedenen Blickwinkeln analysiert:

- **Lebenszyklusabhängige Auswahl und Anwendung:** Systems Engineering ist ein Lebenszyklusansatz. Der Großteil der Arbeiten im MBSE berücksichtigt lediglich die reine Entwicklungsphase. Die MBSE-Konzepte können – und sollten – jedoch zu unterschiedlichen Phasen im Produktlebenszyklus eingesetzt werden. Durch die Zuordnung der Konzepte zu einzelnen Lebenszyklusphasen wird das Verständnis über MBSE erhöht und der Einsatz der Konzepte im Projekt kann für unterschiedliche Aufgaben vorbereitet werden.
- **Zweckabhängige Auswahl und Anwendung:** Jedes Modell erfüllt einen Zweck. Dass dies bislang selten berücksichtigt wird, wurde im Stand der Technik aufgezeigt. Die MBSE-Konzepte sind für unterschiedliche Modellierungszwecke unterschiedlich gut geeignet.
- **Aufgabenorientierte Auswahl und Anwendung:** Der Einsatz der MBSE-Konzepte ist für unterschiedliche Aufgaben denkbar. Es werden Aufgaben im Entwicklungsmanagement dargestellt und geeignete Konzepte zugeordnet.
- **Kompetenzorientierte Auswahl und Anwendung:** Die Fähigkeiten für die Anwendung von SE und insb. MBSE sind in der Industrie noch relativ gering [GDS+13b] – wenngleich die Anforderungen an die Rolle des Systems Engineer

in den klassischen Branchen generell sehr hoch sind. Aus diesem Grund spielen die Ausbildung, Erfahrung und Fähigkeiten potentieller Anwender eine besondere Rolle für den erfolgreichen Einsatz der MBSE-Konzepte.

Diese vier Blickwinkel wurden bewusst nicht zur Herleitung der MBSE-Konzepte berücksichtigt. Anders als bei den Merkmalen in Kapitel 4.2 sind sie nicht robust, d.h. sie können sich bspw. mit der Zeit verändern, branchenspezifisch sein oder generell an Bedeutung verlieren. Für die lebenszyklusabhängige Auswahl wird bspw. in Anlehnung an Kapitel 2.2.1 ein vereinfachter Lebenszyklus gewählt. Für eine konkrete Branche kann dieser ggf. ungeeignet sein. Bei den Kompetenzen könnte es passieren, dass sich mit fortschreitender Durchdringung der Produktentstehung mit MBSE dieses Kriterium generell erübrigt. So könnte es passieren, dass die MBSE-Konzepte nicht eindeutig sind und nach einiger Zeit die Analyse erneut durchgeführt werden müsste.

4.3.1 Lebenszyklusorientierte Auswahl und Anwendung

Die konzeptspezifischen Merkmalausprägungen (vgl. bspw. Bild 4-7) geben einen Hinweis, wann die MBSE-Konzepte sinnvoll im Produktlebenszyklus einsetzbar sind. Die zu stellende Leitfrage zur qualitativen Beurteilung der Eignung eines Konzepts für eine bestimmte Lebenszyklusphase lautet: „*Wie gut können bei gegebener Merkmalausprägung eines MBSE-Konzepts die Aufgaben einer Lebenszyklusphase erfüllt werden?*“ Das kann mit einer Einflussmatrix beurteilt werden. Die Passivsumme gibt Aufschluss über die Eignung des Konzepts für die Lebenszyklusphase. Eine hohe Passivsumme ist dann ein Indikator für eine gute Unterstützung der Lebenszyklusphase durch das Konzept. Bspw. ist MBSE-Konzept 1 gekennzeichnet durch

- den Einsatz nicht-formaler Sprachen,
- die projektinterne Anwendung in Workshops,
- die Nutzung von klassischen Moderationswerkzeugen.

Diese Ausprägungen ermöglichen eine gute Unterstützung der typischen Aufgaben der Phase Produktfindung, wie z.B. Ideenworkshops, Umfeldanalysen und Stakeholderanalysen. Damit liegt ein starker positiver Einfluss vor. Gleiches gilt mit begrenzten Ausmaß für die Marktphase – die Ausprägungen ermöglichen bspw. adhoc-Workshops. So können die erstellten Umfeldanalysen und die damit vorhandenen Informationen als Input für die Fehleranalyse dienen, wie bspw. Ishikawa-Diagramme aus dem Six Sigma-Umfeld. Für die Phase Entwurf und Ausarbeitung ist der Einfluss des Konzepts dagegen im Prinzip nicht vorhanden. Hier sind für die Hauptaufgaben bspw. formale Sprachen und fachdisziplinspezifische Werkzeuge notwendig – somit kann Konzept 1 in dieser Phase nicht sinnvoll unterstützen.

Bei einem derartigen Vorgehen ergibt sich die in Bild 4-10 dargestellte vereinfachte, qualitative Einordnung der Konzepte in den Produktlebenszyklus. Dabei fällt auf: Die

Konzepte bauen logisch entlang des Lebenszyklus auf. Die Überlappung der Phasen deutet auf die Konsistenz der vier Konzepte und ihre komplementäre Ergänzung hin.

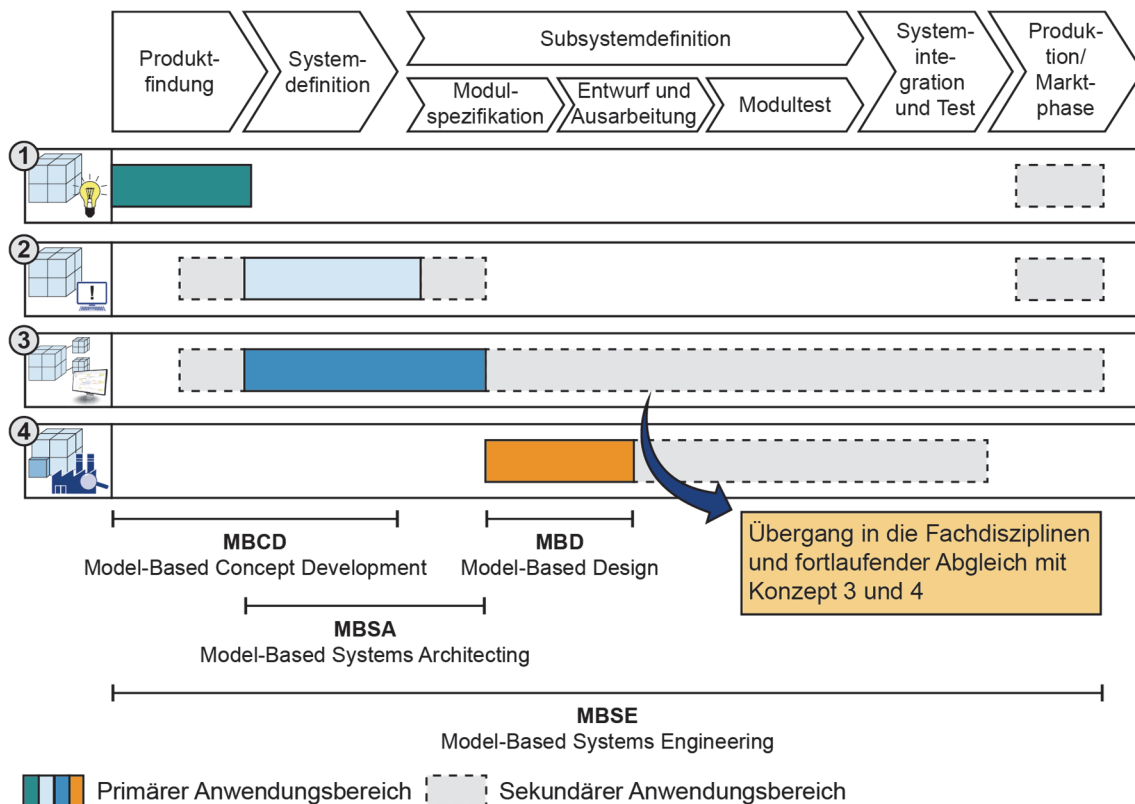


Bild 4-10 Einsatz der MBSE-Konzepte im Produktlebenszyklus

Der primäre Anwendungsbereich zeigt, welche Lebenszyklusphasen hauptsächlich unterstützt werden: Da in der frühen Phase Produktfindung viele Informationen über das Produkt noch nicht verfügbar sind [Ram11], bietet der „hands-on“ Ansatz von Konzept 1 einen pragmatischen Weg, die Informationen für die zukünftigen Aufgaben zu sammeln und mit Methoden der Systemtechnik und des MBSE so aufzubereiten, dass sich eine Formalisierung dieser Informationen mit den Konzepten 2 und 3 nahtlos anschließen kann. Der Fokus auf den Aspekt Struktur in Konzept 2 „Mechatronic Sketch“ macht bspw. den Einsatz in der Phase Systemspezifikation sinnvoll (primäre Anwendung). Das erstellte Systemmodell unterstützt jedoch auch bei Analysen in der Produktion- und Marktphase, z.B. den Servicetechniker bei der Analyse von Fehlerursachen im System (sekundärer Bereich). Konzept 3 erweitert diesen Gedanken hin zu einem lebenszyklusbegleitenden Systemmodell. Konzept 4 legt den Fokus auf die Analyse von Verhalten. Im Wechselspiel mit Konzept 3 kann schon in der Phase Modulspezifikation analysiert werden (Zusammenspiel von Synthese und Analyse, „Frontloading“) und in der Phase Entwurf das Verhalten noch weiter detailliert werden, s.d. beide Konzepte zusammen den Übergang in die Fachdisziplinen vorbereiten.

Aus dieser Zuordnung kann die lebenszyklusorientierte Betrachtung der MBSE-Konzepte in drei wesentliche Phasen unterschieden werden:

- Model-Based Concept Development¹²
- Model-Based Systems Architecting
- Model-Based Design

Sie ergänzen das etablierte Konzept der modellbasierten Entwicklung in den Fachdisziplinen mit Schwerpunkt im Entwurf und der Ausarbeitung nach vorn. Die Unterscheidung und der logische Aufbau der Konzepte ermöglichen die Trennung von Aufgaben und Verantwortlichkeiten; gemeinsam spiegeln die Konzepte die Idee des MBSE wider und ergänzen sich mit den Arbeiten der virtuellen Produktentwicklung.

4.3.2 Zweckabhängige Auswahl und Anwendung

Die Definition eines geeigneten Zwecks ist der Zugang zu einer zielgerechten Modellierung. Im Stand der Technik wurde gezeigt, dass die Definition des Modellierungszwecks im Regelfall nicht stattfindet, bzw. maximal implizit angenommen wird.

Tabelle 4-1 zeigt eine Sammlung von Modellierungs- bzw. Verwendungszwecken, die im Sinne der Problemanalyse zentral sind. Zusätzlich wird zur Rationalitätssicherung bei der Auswahl dargestellt, welche Konzepte diese Zwecke prinzipiell gut unterstützen – wenngleich es je nach Ausgestaltung hier selbstredend Abweichungen geben kann.

Tabelle 4-1 Modellierungszwecke und prinzipielle Eignung der MBSE-Konzepte

		Modellierungszwecke																				
		Systemgrenzen definieren	Problemverständnis sicherstellen	Kommunikation befähigen	Kooperation unterstützen	Innovation befähigen	Interdisziplinäre Systembetrachtung	Architektur beschreiben	Lösungen kommunizieren	Produktkomplexität beherrschen	Dokumentation bereitstellen	Impactanalysen durchführen	Traceability ermöglichen	Übergang in die Konkretisierung	Verifikation und Validierung	Analysen unterstützen	Methoden unterstützen	Simulation unterstützen	Qualität verbessern	Risiko verringern	Entscheidungsunterstützung (Techn.)	Entscheidungsunterstützung (Techn., Mmgt.)
		●	●	●		●													●			
		●	●	●	●	●	●	●	●		●			●		●	●		●	●		●
				●			●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
						●									●	●		●	●	●	●	

● Prinzipiell gute Eignung eines Konzepts zur Unterstützung des Modellierungszwecks

¹² Vgl. hierzu auch die Diskussion in [SF14].

4.3.3 Aufgabenorientierte Auswahl und Anwendung

MBSE unterstützt die Aufgaben im Produktlebenszyklus: „*MBSE emphasizes the application of rigorous visual modeling principles and best practices to Systems Engineering activities [...]“ [oV15].* Damit wird klar, dass MBSE nicht nur bei der Spezifikation und technischen Analyse eines Systems unterstützen soll – wie auch schon im Kapitel 2.3.3 aufgezeigt, sondern generell die Anwendung von Methoden auf breiter Basis unterstützt. Dennoch liegt momentan der Fokus auf der rein technischen Spezifikation des Systems [Bon14]; die – sehr vereinzelt – Anwendung von Methoden dient im Regelfall der frühzeitigen Analyse von Produktverhalten (Stichwort „Frontloading“), was wiederum eine bessere Systemspezifikation ermöglichen soll.

In Ergänzung zu den Modellierungszwecken ermöglicht MBSE auch die Unterstützung der zahlreichen Aufgaben und Methoden der Produktentstehung, die nicht direkt die Spezifikation verfolgen – aber auch Aufgabe des Systems Engineering und damit der Produktentstehung sind. Das lässt sich anhand von Bild 2-8 und Bild 2-9 nachvollziehen, in denen eine Untermenge von Aufgaben und Prozessen der Produktentstehung genannt sind: In Bild 2-9 sind – obwohl es sich nur um einen abstrahierten Ausschnitt eines realen Prozesses handelt – siebzehn Aufgaben genannt, die in der Produktentstehung in der Phase Systemdefinition von potentiellen Stakeholdern des MBSE zu erledigen sind und den in Bild 2-8 angedeuteten Kern-, Management- und Supportprozessen zuzuordnen sind. Die Methoden, die hier unterstützen können, sind dabei noch nicht aufgelistet. Bild 4-11 gibt in Anlehnung an den verallgemeinerten Produktlebenszyklus aus Bild 4-10 einen Überblick über einige Methoden, die in den ersten beiden Phasen mit Hilfe eines modellbasierten Systems Engineering strukturiert Anwendung finden könnten und bislang – wenn überhaupt – papierbasiert durchgeführt werden.

Produktfindung		Systemdefinition	
• Stakholderanalyse	• Use Cases	• Make-or-Buy	• House of Quality
• Carry-over-Part	• Verkaufspreis	• Konzept-FMEA	• Projektbudget
• Wettbewerbsanalyse	• Projekttyp	• Konzept-Workshop	• Team-Setup
• Schutzrechte	• Conjoint-Analyse	• Herstellkosten	• Master-Testplan
• Projektrisiken	• QFD	• Beschaffungsstrategie	• Fertigungskonzept
• Technologieradar	• ...	• Techn. Risiken	• Projektplan
• Target Costing		• Systemkomplexität	• ...

Bild 4-11 Beispielhafte Aufgaben und Methoden des Entwicklungsmanagements in den frühen Phasen der Produktentstehung

Grundlage für das enorme Unterstützungspotential sind die im Systemmodell vorhandenen Strukturinformationen. Diese sind Eingangsinformation für die meisten der o.g. Aufgaben- und Methodenarten und werden in der Regel implizit bei der Spezifikation modelliert. Ihre Nutzung für die Aufgaben aus dem Management der Produktentstehung aufgrund der großen Verfügbarkeit struktureller Informationen ist damit möglich – bleibt jedoch weitgehend aus [Zin13]. Gleichzeitig ergeben sich durch einen höheren Detailgrad der beschriebenen Strukturinformationen – z.B. durch die detaillierte Be-

schreibung unterschiedlicher Flusstypen zwischen Elementen – die Chance, vollkommen neue Methoden zu entwickeln oder existierende komplizierte Methoden handhabbar anzupassen. Schon länger im Fokus sind dagegen die frühzeitige Erprobung und Absicherung von Produkteigenschaften auf Basis experimentierfähiger Modelle, bspw. zur Nachbildung des dynamischen Produktverhaltens im Sinne des Frontloadings. Bild 4-12 zeigt qualitativ, wie gut die beschriebenen MBSE-Konzepte bei den strukturbasierten Analysen im Vergleich zu den meist verhaltensbasierten Analysen unterstützen können. Die ersten drei Konzepte weisen eher einen Fokus auf die Modellierung von Strukturinformationen auf (vgl. auch Bild 4-8, wobei Anforderungen als Struktur zu verstehen sind). Mit Konzept 3 beginnt gleichzeitig auch die Modellierung von Verhaltensaspekten. Konzept 4 fokussiert vollkommen auf die Verhaltensanalyse.

		Analysetypen	
		Strukturbasierte Analyse	Verhaltensanalyse
MBSE-Konzepte		+	-
		++	-
		+++	++
		-	+++

sehr gut geeignet
 nicht geeignet

Bild 4-12 Eignung der MBSE-Konzepte zur Unterstützung unterschiedlicher Analysen

4.3.4 Kompetenzorientierte Auswahl und Anwendung

Ebenso wie der Einsatzzeitpunkt der Konzepte müssen auch die notwendigen individuellen Fähigkeiten der Stakeholder zur Anwendung der Konzepte klar sein. In Kapitel 2.3.4 wurde gezeigt, dass der Systems Engineer ein breites Fähigkeitsprofil erfüllen muss – seine Rolle gleichzeitig aber sehr unterschiedlich ausgeprägt sein kann. Die Fähigkeiten für die Anwendung von MBSE wurden bislang noch nicht untersucht.

Im Folgenden werden Minimalanforderungen zur Anwendung der MBSE-Konzepte erarbeitet, die die Einbindung möglichst vieler Stakeholder in die Produktentstehung auf Basis des modellbasierten Systems Engineerings ermöglichen. Ausgangspunkt ist das vom INCOSE UK Advisory Board erarbeitete Systems Engineering Competencies Framework (SECF) [INC10]. Relevant sind alle drei Beurteilungsklassen *Systems Thinking*, *Holistic Lifecycle View* und *Systems Engineering Management*.

Aus der Analyse und Bewertung des SECF ergeben sich für die vier MBSE-Konzepte die in Bild 4-13 dargestellten Minimal-Anforderungen – relativ zum geforderten Profil des Systems Engineers. Dabei fällt auf, dass gerade die Anforderungen an die Anwendung der Konzepte 1 und 2 weitaus geringer sind, als die vom INCOSE UK Advisory

Board definierten Anforderungen an den Systems Engineer. Konzept 4 unterscheidet sich ebenfalls durch den sehr starken Fokus auf die Simulationsaspekte. Gerade diese Unterscheidung und die klare Darstellung durch die MBSE-Konzepte öffnen den Ansatz hin zu einem Paradigma für die Produktentstehung.

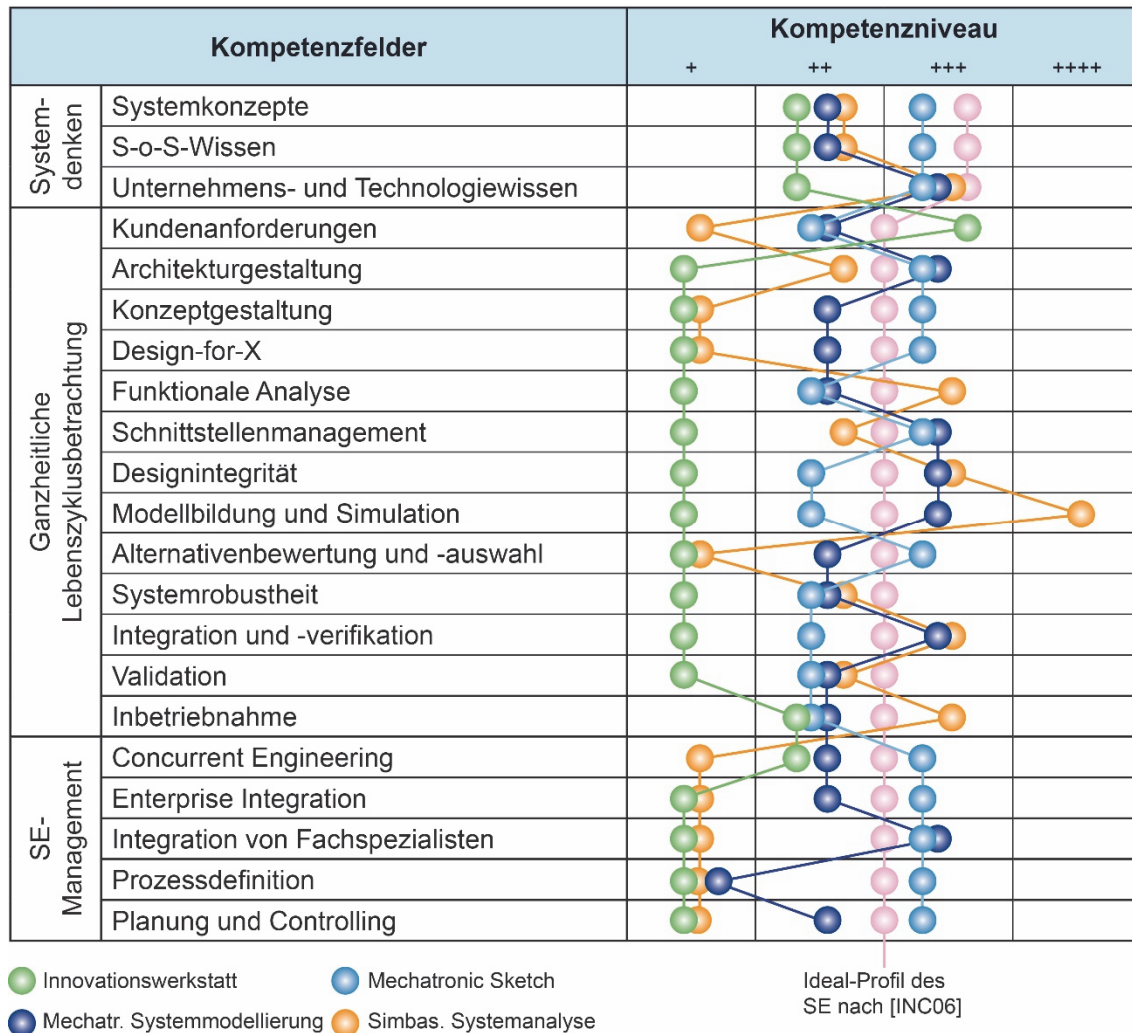


Bild 4-13 Anforderungen zur Anwendung der MBSE-Konzepte im Vergleich zum Qualifikationsprofil des Systems Engineers, angelehnt an [INC10]

4.4 Konzept zur Nutzung des Systemmodells in der Produktentstehungsarbeit

In diesem Kapitel wird ein Konzept erarbeitet, das als Kern der Arbeit mit dem Systemmodell die Durchführung typischer Aufgaben in Produktentstehungsprojekten versteht. Damit wird das MBSE stärker in Richtung des klassischen SE-Verständnisses gerückt (vgl. Kapitel 2.3), die Idee einem größeren Stakeholderkreis geöffnet und eine stärkere methodische Durchdringung der Produktentstehungsarbeit ermöglicht. Konkret steht die Erzeugung und Verwendung von Strukturinformationen des Systemmodells im Vordergrund. Diese werden durch die Modellierung frühzeitig verfügbar und als mehr-

wertig betrachtet [BC10]. Zunächst wird das Grundverständnis dieser Sichtweise auf das MBSE erläutert (Kapitel 4.4.1). Anschließend wird dargestellt, wie die Nutzung von Methoden auf Basis des Systemmodells organisiert werden kann.

4.4.1 Grundverständnis

Die Aufgabe des Systems Engineers und nahestehender Stakeholder ist die Nutzung des Systemmodells zur Unterstützung der alltäglichen Arbeiten und Aufgaben der Produktentstehung: „*MBSE ist die formalisierte Anwendung der Modellierung zur Unterstützung von Aktivitäten wie [...]*“ [INC07], so dass im Sinne der ganzheitlichen SE-Idee sämtliche Aktivitäten der Produktentstehung mit dem Systemmodell orchestriert werden sollen. Bislang wird die Erstellung des Systemmodells durch Vorgehensmodelle wie z.B. CONSENS unterstützt; Ziel ist eine konsistente Systemarchitektur. Diese Architektur kann jedoch – ggf. ergänzt um weitere Informationen – als Grundlage für zahlreiche Aufgaben und Methoden des SE dienen. Dafür muss sie zielgerichtet erzeugt werden, bzw. die vorhandenen Informationen müssen zielgerichtet genutzt werden können. In der Systemtheorie entspricht das dem Konzept der Zweckorientierung. Die etablierten Modellierungsmethoden des MBSE berücksichtigen die durch das Systemmodell zusätzlich gegebenen Möglichkeiten jedoch nicht. Das verhindert die Nutzung des Systemmodells für die zahlreichen Aufgaben der Produktentstehung; wodurch das Potential des MBSE nicht konsequent erschlossen wird – wie in Kapitel 2 aufgezeigt.

Das hier propagierte Verständnis der fachdisziplinübergreifenden Systemmodellierung geht über das heutige Verständnis des MBSE hinaus und sieht als Kernaufgabe eine Nutzung des Systemmodells in Abhängigkeit des Lebenszyklus und der anstehenden Aufgaben. Dadurch muss das systemtheoretische Konzept der „Zweckorientierung“ bei der Modellierung stärker in den Vordergrund gerückt werden – in Abhängigkeit des Projekt- oder Produktlebenszyklus muss eine konkrete Aufgabe unterstützt werden können. Dazu können bspw. eine „Technik-Sicht“ und eine „Management-Sicht“ auf das Systemmodell existieren. In der Technik-Sicht spielt das Verhalten des Systems eine stärkere Rolle, z.B. bei Ablaufsimulationen. Bei der Management-Sicht geht es vorrangig um die Strukturinformationen, da sie – wie dargestellt – Ausgangspunkt für viele der hier adressierten Methoden sind [BC10]. Mit Bezug auf das klassische Systems Engineering werden dann auch die Themenbereiche jenseits der Architekturgestaltung adressiert, bspw. die Technischen Managementprozesse gem. ISO15288. Zusätzlich können bislang vom MBSE ausgeschlossene Stakeholder wie bspw. das Produktmanagement und der Einkauf am MBSE partizipieren und dessen Nutzen erkennen. Bild 4-14 stellt dieses Verständnis dar: In diesem Konzept bekommt neben der Modellierung (und Pflege) nun die Anwendung der Modelle für Aufgaben und Methoden eine Rolle. Dazu sind insb. die Modellplanung und die Methodenkopplung unabdingbare Tätigkeiten für einen erfolgreichen und strukturierten Modellierungsprozess. Mit dieser Sicht kann die Anwendung des Systemmodells zugunsten der Aufgaben der Stakeholder im Lebenszyklus gelingen.

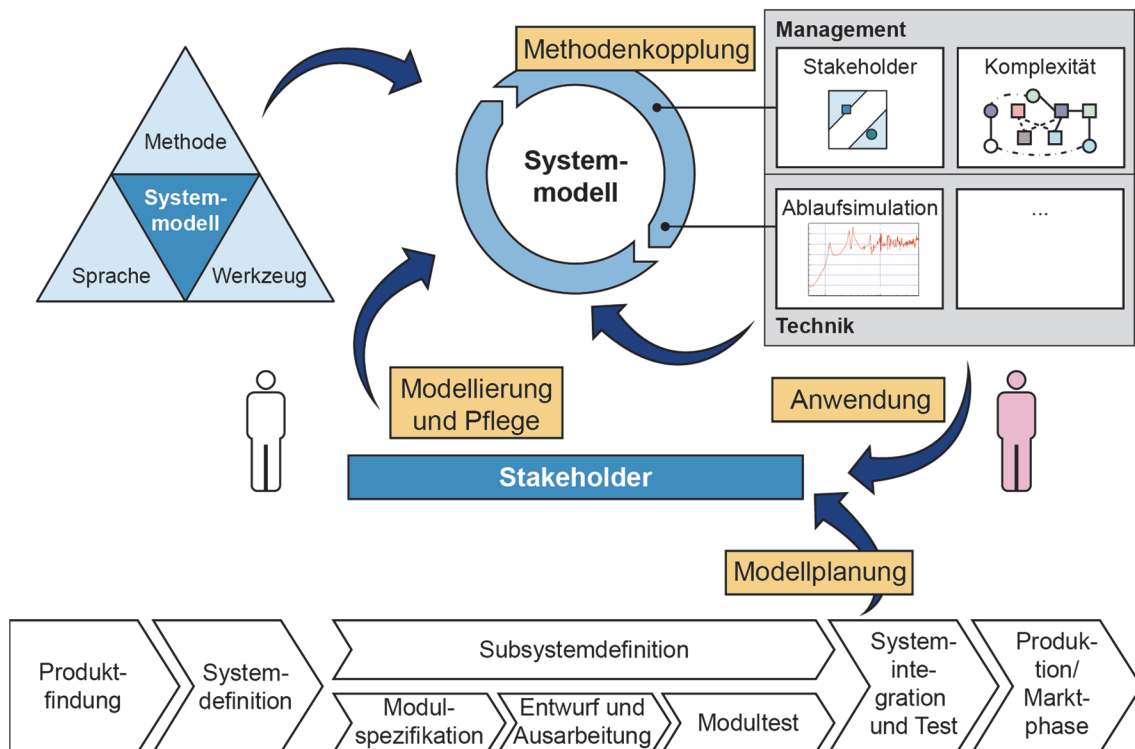


Bild 4-14 Grundverständnis zum MBSE im Produktlebenszyklus

4.4.2 Systemtechnisches Prinzip der Informationsverarbeitung als Grundlage zur Anwendung von Methoden mit dem Systemmodell

Die beschriebene Anwendung des Systemmodells zur Bearbeitung einer bestimmten Aufgabe oder zur Durchführung einer Methode wird nun detailliert. Grundsätzlich entspricht die Anwendung einer Methode dem Ansatz des Problemlösens. Unter einem systemtheoretischen Blickwinkel kann das als klassische Informationsverarbeitung aufgefasst werden. Damit ergibt sich das in Bild 4-15 abstrahierte Vorgehen für die Kopplung des Systemmodells mit Methoden, angelehnt an [PBF+05]:

- Etablierte Methoden des MBSE verfolgen den Zweck der Systemspezifikation. Implizit werden im Idealfall auch die für das Problemlösen notwendigen Eingangsinformationen während der Systemmodellierung erzeugt. Die Systemmodellierung ist gleichzusetzen mit der Informationserzeugung.
- Die für die Anwendung der Methode notwendigen Eingangsinformationen können aus dem Systemmodell gewonnen werden, wobei das im Regelfall nur einer Teilmenge der modellierten Inhalte entspricht. Ggf. sind auch zusätzliche Informationen durch Spezialisten notwendig, bspw. zur Bewertung. So kann eine Gewichtsanalyse auf Basis der im Systemmodell vorhandenen Informationen automatisch erfolgen, eine Stakeholderanalyse dagegen bedarf zusätzlicher Bewertungsmaßnahmen durch Anwender. Welche Informationen das sind, muss klar definiert werden. Der Schritt der Informationsgewinnung dient somit der Identifikation und Extraktion relevanter Informationen aus dem Systemmodell.

- Der Schritt der Informationsverarbeitung bezeichnet die Überführung der identifizierten Eingangsinformationen in entsprechende Ausgabeinformationen, was nach MÜLLER dem Prinzip der Methode entspricht [Mül90]. Methoden sind vereinfacht dargestellt eine Abfolge von definierten Bewertungsschritten, in denen vordefinierte Mechanismen abgearbeitet werden. Das wurde bspw. auch durch das Prozessorientierte Methodenmodell dargestellt [BKB+02].
- Die Bereitstellung und Anwendung der Ergebnisse geschieht über die Informationsausgabe, bei den adressierten Methoden meist über eine Visualisierung: Die Handlungsperson greift das Methodenergebnis auf und verwertet es im Prozess. Das umfasst auch das Zurückspielen der Ergebnisse in das Systemmodell.

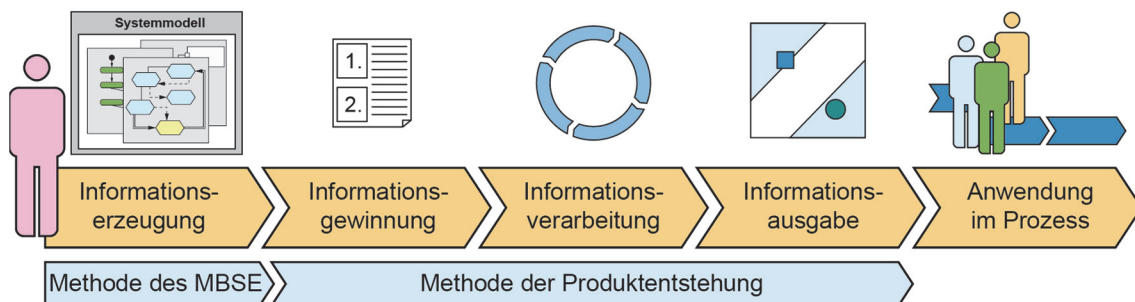


Bild 4-15 Systemtechnisches Prinzip des Problemlösens übertragen auf die Anwendung von Methoden der Produktentstehung mit dem Systemmodell

4.4.3 Methodenanwendung durch „Trennung von Modell und Sicht“

Das vorangegangene Kapitel 4.4.2 macht deutlich, dass Methoden der Produktentstehung mit dem Systemmodell unterstützt werden und angelehnt an das Konzept „Trennung von Modell und Sicht“ umgesetzt werden können – für eine Methode ist nur eine Untermenge der modellierten Inhalte notwendig. Die Sicht bietet die Möglichkeit, die mindestens für eine Methode relevanten Eingangsinformationen zu modellieren, die dann aus dem Datenmodell informationstechnisch weiterverarbeitet werden.

Bild 4-16 stellt im oberen Bereich dieses Verständnis im Einklang mit den Konventionen der ISO42010 und dem Prinzip der Informationsverarbeitung dar; zunächst werden die notwendigen Informationen erzeugt und hierfür die etablierten MBSE-Methoden verwendet; Ergebnis der Modellierung ist das Modell (bislang lt. ISO42010: Architecture Description). Hieraus bilden sich Sichten in Abhängigkeit der Concerns eines Stakeholders. Der Detaillierungsgrad eines Concerns – „any topic of interest“ – ist in der ISO42010 nicht definiert [ISO42010]. Allgemein ist es der „Zweck“ der Modellierung und kann daher als die anzuwendende Methode interpretiert werden. Die Sicht stellt die relevanten Informationen zur Befriedigung des Concerns dar, die als Teilmenge des Modells relevant sind. Entscheidend ist in dieser Darstellung der Viewpoint: Er enthält gemäß Verständnis der ISO42010 alle notwendigen Informationen, wie z.B. Modellie-

rungsmethoden, Analysetechniken, Konzepte und Visualisierungsvorgaben, die zur Gewinnung des Views und zur Befriedigung des Concerns notwendig sind.

Dieses Verständnis des MBSE klammert also die Kopplung des Systemmodells mit dem Lebenszyklus aus. Deshalb erfährt die Darstellung in Bild 4-16 im Einklang mit dem in Kapitel 4.4.1 dargestellten Grundverständnis die hierfür notwendigen Erweiterungen: Eine Methode ist relevant in einem Prozess in einer Lebenszyklusphase. Der Prozess wird von einer Rolle ausgeführt, die hierfür einen spezifischen Detaillierungsgrad der Sicht benötigt. Der Lebenszykluszeitpunkt entscheidet über den möglichen oder notwendigen Detaillierungsgrad der Sicht. Erst so ist die Anwendung des MBSE im Lebenszyklus beschrieben und eine systemtheoretische Ausrichtung des MBSE gegeben.

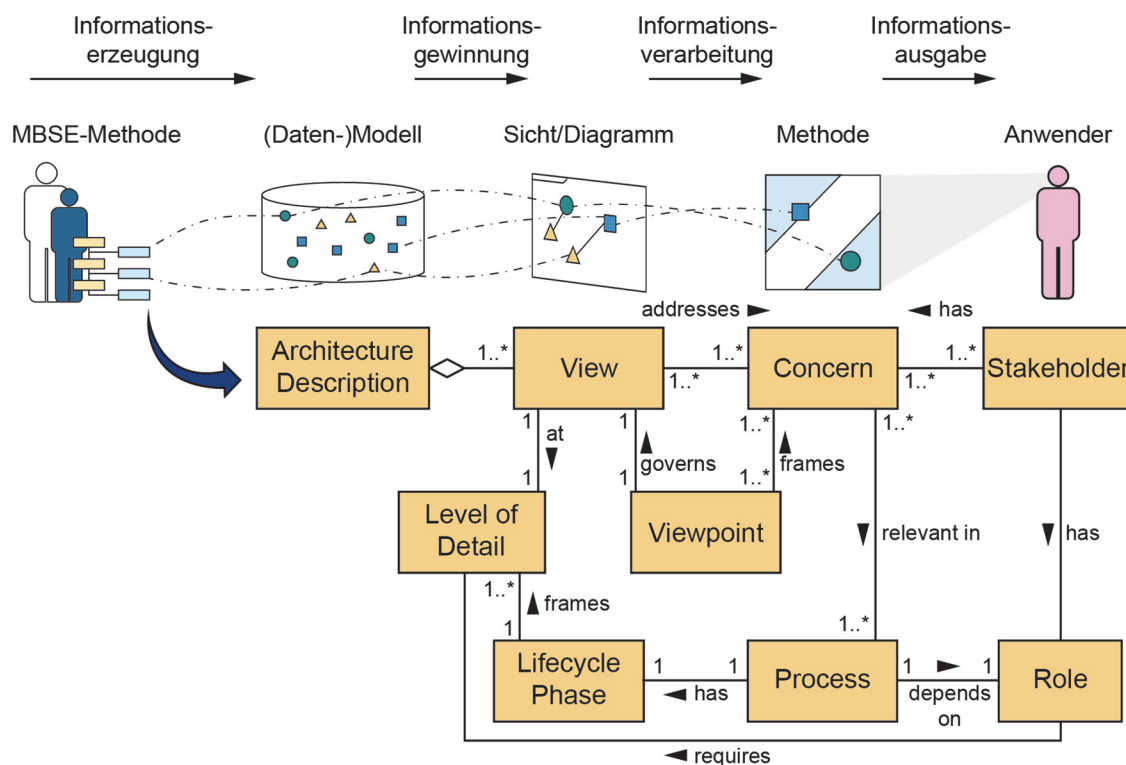


Bild 4-16 Nutzung des Konzepts „Trennung von Modell und Sicht“

Damit ist gezeigt: Die Nutzung von Methoden ist möglich auf Basis des Systemmodells – muss aber mit entsprechenden Hilfsmitteln angeleitet werden. Sicht und Standpunkt haben eine Schlüsselrolle. Sie beschreiben die geplante Anwendung der Methode und leiten bei der Kopplung von Systemmodell und Methode an – im vorliegenden Fall wird hierzu ein flexibles Vorgehensmodell erarbeitet, das für jede Art der Modellierung und Methoden-anwendung die entsprechenden Hilfsmittel bereitstellen kann. Im Prinzip integriert das Vorgehensmodell sämtliche Modellierungsmethoden und Hilfsmittel und stellt sie bedarfsorientiert bereit, da es zur Planung des Modellierungszwecks, der Modellierung und zur Anwendung der Modelle im Prozess befähigt.

4.5 Flexibles Vorgehensmodell zur Planung der Systemmodellierung in der Produktentstehung

In diesem Kapitel wird ein Vorgehen entwickelt, das die Planung, Erstellung und Nutzung des Systemmodells anleitet und dabei insb. die Anwendung von Methoden der Produktentstehung auf Basis des Systemmodells aufgreift: Kern ist ein flexibles Vorgehensmodell. Eine 4x4-Matrix strukturiert die einzelnen Aufgaben und Methoden von der Modellplanung bis hin zur Anwendung der Methode in der Produktentstehung. Es integriert im Prinzip sämtliche Modellierungsmethoden in einem Metamodell der Modellierung – je nach Ziel werden dazu unterschiedliche Hilfsmittel verwendet.

Zunächst wird das flexible Vorgehensmodell vorgestellt, inkl. eines idealtypischen Wegs (Kap. 4.5.1). Im Anschluss werden alternative Wege durch das Vorgehensmodell gezeigt (Kap. 4.5.2). Das Vorgehen unterstützt besonders die Anwendung der MBSE-Konzepte Mechatronik Sketch und Mechatronische Systemmodellierung.

4.5.1 Das Vorgehensmodell im Detail

Bild 4-17 gibt einen Überblick über den strukturellen Aufbau des flexiblen Vorgehensmodells. Die Darstellung als Matrix ist u.a. angelehnt an das iPeM und koppelt ebenfalls einen Marko- und einen Mikroprozess: Die Zeilen orientieren sich an den Vorgehensweisen der Six Sigma-Ansätze (DMAIC und DMADV)¹³ [TK08] – das entspricht dem Makroprozess. Die Spalten lehnen sich an das erweiterte Konzept der Informationsverarbeitung aus Bild 4-15 an und bilden damit einen Mikrozyklus. In der resultierenden 4x4-Matrix werden somit die etablierten Phasen von Six Sigma mit den grundlegenden Aktivitäten der Informationsverarbeitung gekoppelt. Ähnlich zur Idee von Six Sigma und des iPeM resultiert dadurch ein Baukasten, in den sich verschiedene Vorgehensweisen, Methoden und Hilfsmitteln für das MBSE einordnen. Zweck ist die Befähigung einer rationalen Planung der Modellierung, ihrer Durchführung und der anschließenden Nutzung des Systemmodells. Deshalb integrieren sich in die 4x4-Matrix auch gemäß des Konzepts zur Nutzung des Systemmodells aus Kapitel 4.4 die vier Hauptaufgabenbereiche Modellplanung, Modellierung, Methodenkopplung und Anwendung/Verstetigung. Deren Aufgaben können wie folgt zusammengefasst werden:

- **Modellplanung:** Sie dient der Abgrenzung und Definition des Problems, der Auswahl der Modellierungsmethode und der Konzipierung des Systemmodells.
- **Modellierung:** Das entspricht der Realisierung des Systemmodells mit den Sprachen und Methoden des modellbasierten Systems Engineerings. Dabei kommen auch Modellierungsrichtlinien wie z.B. nach [Kai14] zum Einsatz, bspw. um die Plausibilität der modellierten Inhalte zu prüfen.

¹³ Der DMAIC-Circle steht für Define – Measure – Analyze – Improve – Control im Six Sigma, im Design for Six Sigma steht DMADV für Define – Measure – Analyze – Design – Verify.

- **Methodenkopplung:** Die Nutzung der Systemmodellinhalte wird hier befähigt. Hiervon abhängig werden auch die notwendigen Werkzeuge zur Methodenanwendung erstellt.
- **Anwendung und Verstetigung:** In erster Linie geht es um die Nutzung des Systemmodells für die geplante Aufgabe, also die Nutzung der Systemarchitekturbeschreibung zur Befriedigung eines konkreten Concerns (vgl. [ISO42010]). Mit jeder erfolgreichen Methodenkopplung wird eine Veränderung des Produktentstehungsprozesses vorgenommen. Somit ist bspw. im Sinne einer erfolgreichen Methodenanwendung eine ausführliche Schulung der Nutzer sicherzustellen.

	Informations- erzeugung	Informations- gewinnung	Informations- verarbeitung	Informations- ausgabe
Define				
Measure/Analyze				
Design				
Control				

	Modellplanung		Modellierung
	Methodenkopplung		Anwendung u. Verstetigung

Bild 4-17 Flexibles Vorgehensmodell im Überblick und idealtypisches Vorgehen

Durch diese Hauptaufgabenbereiche erstrecken sich die Phasen des Makroprozesses, wobei die fünf Phasen des DMAIC-Circles in vier Phasen zusammengefasst wurden:

- **Define – D:** Hier geht es um das Modellierungsprojekt; die Ziele, Rollen und Verantwortlichkeiten des Projekts werden definiert. Insbesondere wird die Nutzung des Systemmodells bestimmt und geklärt, wie es in die Produktentstehungsarbeit eingebunden werden kann. Ebenso soll die Phase das Bewusstsein für die anstehende Veränderung durch MBSE schaffen. Initiiert wird sie bspw. in Anlehnung an die Rollen nach [She96] durch den Technical Manager. Im Six Sigma heißt diese Rolle „Champion“ und gehört häufig der Unternehmensführung an, was dem Ansatz eine hohe Akzeptanz zukommen lässt.
- **Measure/Analyze – M/A:** Im Six Sigma werden Prozesse aufgenommen und in der Phase Analyze analysiert. Im Design for Six Sigma (DfSS) werden Kundenbedürfnisse ermittelt und Anforderungen an das Produkt gesammelt. Dann werden mögliche Produktideen verglichen und das zu realisierende Konzept ausgewählt. Die Trennung der beiden Phasen dient der Risikominderung – am Ende der Measure-Phase ist ein Abbruchkriterium platziert, falls die Messungen nicht die erhofften Einsparpotentiale widerspiegeln. Im dargestellten Vorgehen werden die Phasen Measure und Analyze zusammengefasst. Einerseits sind die Aufgaben in den Phasen sehr ähnlich, andererseits sind im Six Sigma die Auf-

wände für die Analyse-Phase überproportional hoch, da hier eine intensive Ursachen-Wirkungsforschung durchgeführt wird. Die Aktivitäten der hier propagierten kombinierten Phase beziehen sich auf die Modellplanung und auf die Planung der Methodenkopplung. Mit Bezug auf die Modellplanung wird bspw. zunächst geklärt, welche Modellierungsmethode verwendet wird. In der Aktivität Informationsgewinnung werden die zu modellierenden Daten definiert, s.d. das Systemmodell den geplanten Zweck erfüllen kann. In der Methodenkopplung wird definiert, wie die Infrastruktur und die Ergebnisdarstellung aussehen.

- **Improve/Design – I/D:** Im DfSS – dessen Phasenbegriff „Design“ hier gewählt wird – finden Entwurf und Ausarbeitung des gewählten Konzepts statt. Es geht hier also zunächst um die Erstellung des Systemmodells mit der Bewegungsrichtung Modellierung (Design – Informationserzeugung), dann um die Implementierung der Methode zur Nutzung auf Basis des Systemmodells.
- **Control – C:** Im Six Sigma – dessen Phasenname Control hier verwendet wird – geht es um die Verstetigung der Ergebnisse, die Schulung der Mitarbeiter und den Projektabschluss, inkl. der Verwendung des gewonnenen Wissens für weitere Projekte. Hier geht es in erster Linie um die Anwendung der Methode und die Verstetigung der Arbeiten, also bspw. die Einführung der neuen Vorgehensweise in den Unternehmensprozess durch eine Aktualisierung der Prozessbeschreibung (vgl. Bild 2-9). In der Aktivität der Informationsgewinnung geht es zunächst aber um eine Überprüfung der Modellierung, bspw. unter Anwendung der durch [Kai14] definierten Regeln. Ebenso sollten Entscheidungen dokumentiert werden, bspw. die Begründungen für eine konkrete Modellierungsart. Hinsichtlich der Anwendung und Verstetigung geht es um die Durchführung der Methode auf Basis der implementierten Lösung und die Dokumentation.

Für die Anwendung von Methoden auf Basis des Systemmodells sind die Hauptaufgabenbereiche Modellplanung und Methodenkopplung von besonderer Bedeutung und grundsätzlich neuartig für die existierenden MBSE-Methoden. Die Hilfsmittel dieser Phasen sollten die existierenden Modellierungsmethoden somit ergänzen. Die konkret durchzuführenden Aufgaben werden jeweils in einem ergänzenden Phasen-Meilenstein-Diagramm benannt (vgl. Kap. 4.6).

4.5.2 Idealtypische Vorgehensweisen

Die Matrix aus Bild 4-17 weist in den Hauptaufgabenbereichen abgetrennte Felder auf. Diese trennen einzelne logische Bereiche der Hauptaufgaben voneinander, die unterschiedliche Wege durch das Flexible Vorgehensmodell ermöglichen. Im Rahmen der Arbeit wurden fünf mögliche Wege durch die 4x4-Matrix identifiziert, die situationspezifisch ausgeprägt werden können (vgl. Bild 4-18).

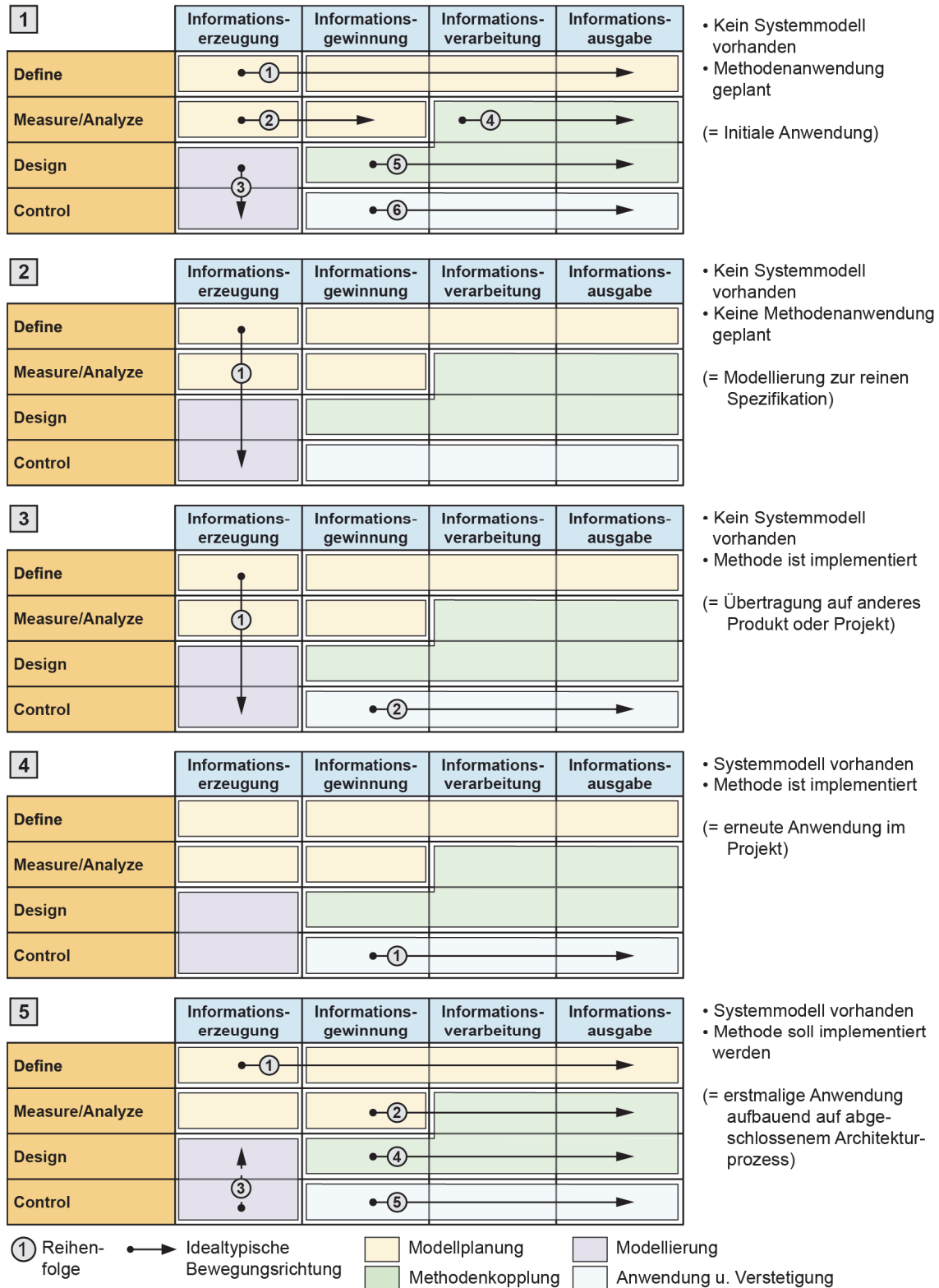


Bild 4-18 Idealtypische Vorgehensweisen im flexiblen Vorgehensmodell im Überblick

Je nach Intention zur Nutzung wird ein anderer Weg durch die 4x4-Matrix zurückgelegt. Dieser ist ebenfalls im Bild durch die idealtypische Bewegungsrichtung angedeutet; die Zahlen stellen die empfohlene Reihenfolge dar. Möglich sind im Einzelnen:

- 1) **Kein Systemmodell vorhanden – Methodenapplication geplant.** Dies wird die häufigste Variante durch die 4x4-Matrix sein. Hierbei wird von einem komplett neuen Projekt ausgegangen. Da alle Bereiche des flexiblen Vorgehensmodells bearbeitet werden, ist es häufig die erstmalige Anwendung des Systemmodells zur Unterstützung von Methoden, da keine gekoppelten Methoden vorliegen. Ebenso könnte es sich aber auch um die erstmalige Anwendung in einem Projekt nach einer punktuellen Überarbeitung des Produktentstehungsprozesses handeln, bei dem eine neue Methode hinzugefügt wurde.
- 2) **Kein Systemmodell vorhanden – Keine Methodenapplication geplant.** Das entspricht dem Zweck der Architekturbeschreibung, ggf. mit dem erweiterten Zweck der Dokumentation. Die Methoden und Hilfsmittel des flexiblen Vorgehensmodells ergänzen die Planung und Dokumentation der Modellierung. Idealtypisch muss hier nur Bewegungsrichtung (1) durchlaufen werden.
- 3) **Kein Systemmodell vorhanden – Methode ist implementiert.** Das deutet auf ein erfolgreiches Vorprojekt hin, das den Nutzen des MBSE bestätigt hat. Nun wird die Anwendung auf ein weiteres Projekt übertragen. Zuvor getroffene Entscheidungen haben weiterhin Bestand.
- 4) **Systemmodell vorhanden – Methode ist implementiert.** Hierbei kann es sich um die erneute Anwendung einer Methode in einem laufenden Projekt handeln.
- 5) **Systemmodell vorhanden – Methode soll implementiert werden.** Dieser Fall kann im MBSE häufig vorkommen, da bislang der Modellierungszweck nicht ausreichend definiert wurde; ein existierendes Modell muss nun für die erstmalige Anwendung einer Methode „nachbereitet“ werden.

4.6 Methoden und Hilfsmittel im flexiblen Vorgehensmodell

In diesem Kapitel werden die Schritte des Vorgehensmodells aus Kap. 4.5 detailliert. Je Schnittpunkt von Makro- und Mikroprozess ergeben sich Aufgaben, die beschrieben und – wo sinnvoll – anhand einer Stakeholderanalyse¹⁴ für einen Kaffee-Vollautomaten erläutert werden¹⁵. Zur Bearbeitung der Aufgaben werden Methoden und Hilfsmittel entwickelt bzw. existierende Methoden und Hilfsmittel in das Rahmenwerk integriert. Weitere können jederzeit ergänzt werden. Die Reihenfolge der Vorstellung orientiert sich an den in Bild 4-17 dargestellten Bewegungsrichtungen. Auf die Aktivitäten der Design- und Control-Phase wird der Vollständigkeit halber eingegangen; der Fokus liegt im Rahmen der Arbeit aber auf der Define- und Measure/Analyse-Phase.

4.6.1 Define-Phase

Die Define-Phase zieht sich über alle Aktivitäten der Informationsverarbeitung und umfasst den ersten Teil der Modellplanung. Im Six Sigma kommt der Phase trotz ihrer Kürze eine sehr hohe Bedeutung zu: Es werden der Projektinhalt bestimmt, im sog. Fact-Sheet alle verfügbaren Daten gesammelt und mit dem S-I-P-O-C¹⁶ ein grobes Verständnis über den Prozess erzeugt. Zuletzt wird die Project-Charter als Projektplan mit klarem Arbeitsauftrag sowie messbaren Zeit- und Kostenzielen verabschiedet.

Analog zum Vorgehen im Six Sigma wird für das Modellierungsprojekt ein Überblick über die Aufgabe erarbeitet. Alle wesentlichen Informationen über das Modellierungsprojekt werden in Steckbriefen zusammenfasst, die auch über diese Phase hinaus gepflegt werden sollten. Die Informationen dazu stammen aus unterschiedlichen Quellen, bspw. üblichen Projektdokumentationen oder werden extra erarbeitet.

Teil 1 der Define-Phase adressiert die Informationserzeugung und sollte obligatorische Ergänzung für sämtliche Modellierungsmethoden sein. Teil 2 umfasst Informationen zur Informationsgewinnung, -verarbeitung und -ausgabe. Hier werden Informationen vor einem konkreten Modellierungszweck zusammengetragen, wie bspw. hier vor dem Hintergrund der Anwendung von Managementmethoden. Insgesamt erhöhen die Steckbriefe die Nachvollziehbarkeit des Modellierungsprojekts und der getroffenen Entscheidungen nach Projektende. Verantwortlich sollte in dieser Phase ein Mitglied des Kernteams sein, das ähnlich zum Six Sigma auch als Projektsponsor bzw. -pate auftritt.

¹⁴ Die Stakeholderanalyse wird häufig als erste Methode in Projekten eingesetzt – allerdings i.d.R. informell. Die Ergebnisse können dann nicht in weitere Aktivitäten überführt werden. Wird sie mit dem Systemmodell unterstützt, sind die Informationen für alle Phasen der Produktentstehung verfügbar. So kann anhand dieser einfachen Methode sehr schnell der Nutzen des MBSE vermittelt werden.

¹⁵ Hierbei handelt es sich um das Referenzbeispiel des sog. Tool Vendor Projects der GfSE e.V., das der Autor zusammen mit dem Vorstand der GfSE eingeführt hat. Vgl. hierzu u.a. auch www.tdse.org.

¹⁶ S-I-P-O-C stammt aus dem Six Sigma-Umfeld und wird hier als einfache Prozessbeschreibungssprache genutzt. S-I-P-O-C steht für S – Supplier, I – Input, P – Process, O – Output, C – Customer.

4.6.1.1 Define – Informationserzeugung (IE)

Hier geht es um die Planung des Modellierungsprojekts. Es stehen allgemeine Informationen über das Modellierungsprojekt und auch das übergeordnete Produktentstehungsprojekt im Fokus. Die Informationen werden in den drei Schritten Projektanalyse – Zweckdefinition – Methodenauswahl zusammengetragen, wobei letzterer nur bei entsprechendem Modellierungszweck durchgeführt wird. Das Phasen-Meilenstein-Diagramm in Bild 4-19 fasst die dazugehörigen Aufgaben und Methoden zusammen.

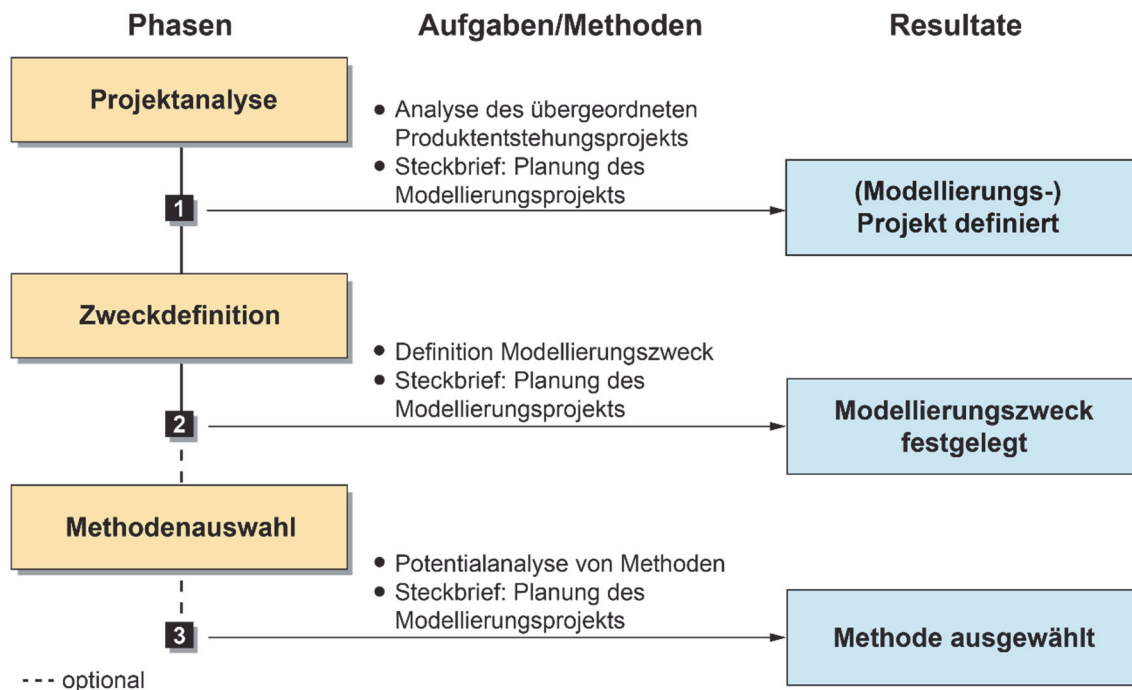


Bild 4-19 Aufgaben in der Define-Phase – Informationserzeugung

Projektanalyse

Es werden allgemeine Informationen zum Produkt, dem Projekt und den Verantwortlichkeiten dokumentiert. Ebenso werden der geplante Projektverlauf, Meilensteine und zu liefernde Artefakte skizziert – wie im jeweiligen Entwicklungsprozess definiert.

Zweckdefinition

Zusätzlich wird der Modellierungszweck explizit definiert – im Gegensatz zu den etablierten MBSE-Methoden, bei denen dieser nur implizit angenommen werden kann. Konkret können das die in Kap. 4.3.2 herausgearbeiteten Modellierungszwecke sein. Das Wissen über den Modellierungszweck hilft in den weiteren Phasen bei der rationalen Auswahl der Modellierungssprachen, -methoden und -werkzeuge.

Die Ergebnisse der beiden Phasen sind im Steckbrief dokumentiert (Bild 4-20) – hier für das fiktive Referenzbeispiel der Kaffeemaschine. Zusätzlich zur Projektanalyse und der Zweckdefinition findet in Abhängigkeit des Modellierungszwecks ggf. noch eine Methodenauswahl statt. Das bedeutet: Es wird analysiert und festgelegt, welche Metho-

den mehrwertstiftend auf Basis des Systemmodells im eigenen Prozess angewendet werden können. Dazu unterstützt der im Anschluss dargestellte Quick Check.

Steckbrief (Planung des Modellierungsprojekts)


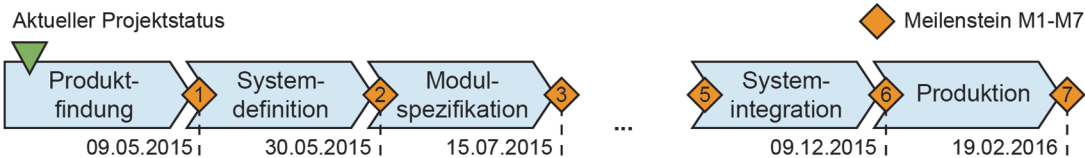
Projektname: Tool Vendor Project Kunde: Flyport Projektnr. ABCXYZ123		Bearbeiter: Christian Tschirner Stand: 15.12.2015																					
Produkt- und Projektinformationen		Verantwortlichkeit																					
Produkt: Kaffeeautomat Flughafen Projektname: TVP GfSE Systemebene: System Projektart: Kundenindividuell Projektstart: 03.03.2015 Liefertermin: 19.02.2016		Kunde: Flyport Key Account: Elisabeth Produktmanager: Julius Projektleiter: Richard Lead Architekt: Hans Kernteam-Pate: Magda																					
Produkt- und Projektinformationen																							
<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>Aktueller Projektstatus</p>  </div> <div style="width: 35%; text-align: right;"> <p>◆ Meilenstein M1-M7</p> </div> </div> <div style="margin-top: 5px;"> <ul style="list-style-type: none"> ▪ Stakeholder ▪ Projektrisiko ▪ Use Cases ▪ ... </div> <div style="margin-top: 5px;"> <ul style="list-style-type: none"> ▪ Funktionen ▪ Top-Komponenten ▪ Systemkomplexität ▪ ... </div> <div style="margin-top: 5px;"> <ul style="list-style-type: none"> ▪ Systemkomplexität ▪ ... </div>																							
Modellierungszweck		Methoden auf Basis des Systemmodells																					
Architekturbeschreibung <input type="checkbox"/> Dokumentation <input type="checkbox"/> Methodenunterstützung <input checked="" type="checkbox"/> Simulation <input type="checkbox"/> Kommunikationssupport <input type="checkbox"/> ... <input type="checkbox"/>		<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>einmalig</th> <th>mehrfach</th> <th>Meilenstein</th> </tr> </thead> <tbody> <tr> <td style="text-align: left;">Stakeholderanalyse</td> <td>X</td> <td></td> <td>M1</td> </tr> <tr> <td style="text-align: left;">Projektrisiko</td> <td></td> <td>X</td> <td>M1-M4</td> </tr> <tr> <td style="text-align: left;">Complexity</td> <td></td> <td>X</td> <td>M2/M3</td> </tr> <tr> <td style="text-align: left;">Make-or-Buy</td> <td>✓</td> <td></td> <td>M2</td> </tr> </tbody> </table>			einmalig	mehrfach	Meilenstein	Stakeholderanalyse	X		M1	Projektrisiko		X	M1-M4	Complexity		X	M2/M3	Make-or-Buy	✓		M2
	einmalig	mehrfach	Meilenstein																				
Stakeholderanalyse	X		M1																				
Projektrisiko		X	M1-M4																				
Complexity		X	M2/M3																				
Make-or-Buy	✓		M2																				

Bild 4-20 Beispielhafter Steckbrief „Planung des Modellierungsprojekts“ – Teil 1/2

Methodenauswahl

Die Methodenauswahl sollte im Rahmen der Define-Phase durch einen Quick Check unterstützt werden. Idealerweise wird auf eine existierende Prozessbeschreibung zurückgegriffen. Auf dieser Basis können Methoden identifiziert werden, die im Unternehmen bereits eingesetzt werden, und ihre Eignung zur Kopplung mit dem Systemmodell bewertet werden.

Der Erfolg einer veränderten Arbeitsweise kann in dieser frühen Planungsphase allerdings nur anhand einer qualitativen Vorbewertung bestimmt werden. In Anlehnung an SAUTER [SAU12] wird ein Portfolio mit den zwei Bewertungsdimensionen „Empfundener Nutzen durch veränderte Arbeitsweise“ und „Empfundener Aufwand für die Einführung und Nutzung“ mit dem dazugehörigem Bewertungsmaßstab in Anlehnung an das

3P-Modell nach BADKE-SCHAUB [BDR11] definiert. Beispielhaft sollen einige der in Bild 4-11 im Rahmen von Industrieprojekten genutzten Methoden beurteilt werden.

Tabelle 4-2 zeigt die Kriterien und Bewertungsmaßstäbe der Dimension „Empfundener Nutzen“ im Überblick. Konkret handelt es sich um die folgenden vier Kriterien:

- **Häufigkeit im Prozess:** Das Kriterium bewertet, wie häufig die betrachtete Methode in einem Prozess angewendet wird. Je häufiger, desto besser – insb. wenn die Methode verbindlich vorgeschrieben ist. Das Gegenteil könnte bspw. der Fall sein, wenn die Methode lediglich optional ist.
- **Zusatznutzen für die fachdisziplinübergreifende Zusammenarbeit:** Hier wird beurteilt, ob die Methode einen Nutzen für die fachdisziplinübergreifende Zusammenarbeit stiftet. Das tun im Regelfall systemtechnische Methoden wie sie bspw. in der VDI-Richtlinie 2221 dargestellt sind [VDI2221]. Der Nutzen steigt, je mehr Disziplinen in der Anwendung involviert sind. Im Idealfall können die Ergebnisse sogar als Input für weitere Anwendungen genutzt werden.

Tabelle 4-2 Bewertung y-Achse: Empfundener Nutzen durch Einführung

Empfundener Nutzen		Bewertungsmaßstab				
		0	1	2	3	
Bewertungskriterien	1	Häufigkeit im Prozess	Methode nicht vorgeschrieben	Methode ist lediglich optional	Methode ist einmalig Pflicht	Methode regelmäßig im Projekt eingesetzt
	2	Zusatznutzen für Zusammenarbeit	kein Zusatznutzen	Zusatznutzen für Zusammenarbeit	hoher Zusatznutzen für mehrere Disziplinen	weitere Stakeholder benötigen Ergebnisse
	3	Automatisierungspotential	kein Potential	Methode unterliegt „Wenn-Dann-Beziehungen“	Methode durch eindeutiges Vorgehen bestimmt	größtenteils automatischer Informationsinput aus SM möglich
	4	Virtualisierungspotential	kein Potential	„hybride“ Arbeitsweise	dokumentenzentrierte Arbeit abgelöst	Durchgängigkeit eines Prozessschritts möglich

SM: Systemmodell

- **Automatisierungspotential:** Der Zusammenhang zwischen MBSE und dem Workflowmanagement wurde in Kapitel 2 dargestellt. Generell sind Methoden der Produktentstehung schwierig automatisierbar, da sie bspw. hohe kreative Anteile aufweisen [Roe11]. Je eindeutiger das Vorgehen einer Methode ist und je mehr Informationen zur Durchführung der Methode aus dem Systemmodell direkt gewonnen werden können, umso größer ist ihr Automatisierungspotential.
- **Virtualisierungspotential:** Die rechnergestützte Anwendung der Methoden bietet großen Nutzen für die Produktentstehungsarbeit. Das Potential ist größer, je stärker auf ergänzende Papierdarstellungsformen verzichtet werden kann und bspw. eine virtuelle Durchgängigkeit des betrachteten Prozessschrittes gelingt.

Für die Bewertungsdimension „Empfundener Aufwand für die Einführung und Nutzung“ werden die in Tabelle 4-3 dargestellten Kriterien herangezogen:

- **Dokumentation:** Auf die Bedeutung gut dokumentierter Prozesse und Methoden wurde bereits in Kap. 2 eingegangen. Der Aufwand für die Einführung und Nutzung ist selbstredend umso größer, je weniger über die Prozesse und anzuwendenden Methoden dokumentiert ist. Mit Bezug auf Bild 2-8 sollte gelten, dass eine detaillierte Prozess- und Methodenbeschreibung den Aufwand senkt.
- **Breite der Veränderung:** Der Lean-Ansatz ist erfolgreich, da er fokussierte Veränderungen anstrebt; eine Veränderung der Unternehmensstrukturen ist nicht der Fall. In Anlehnung an SERVATIUS ist die schrittweise Veränderung des Unternehmens durch neue Arbeitsweisen verbunden mit einer behutsamen Einstellungsveränderung der Betroffenen Erfolgsgarant für die nachhaltige Veränderung [Ser94]. Das Gegenteil ist der Neuaufbau des Gesamtunternehmens.
- **Tiefe der Veränderung:** Der evolutionären Veränderung im Sinne einer kontinuierlichen Verbesserung wird grundsätzlich eine größere Erfolgchance eingeräumt als dem radikalen Wandel. Die schrittweise Veränderung erhöht die Akzeptanz, die Wirtschaftlichkeit und die Humanität der Veränderung. Bei der radikalen Veränderung werden die Bedürfnisse der adressierten Stakeholder nicht ausreichend berücksichtigt – sie sind im Prinzip „Manövriermasse“.

Tabelle 4-3 Bewertung x-Achse: Empfundener Aufwand bei der Einführung

Aufwand bei Einführung			Bewertungsmaßstab			
			0	1	2	3
Bewertungskriterien	1	Dokumentation	Prozess- und Methodenbeschreibung vorhanden	Methodenbeschreibung vorhanden	Prozessbeschreibung vorhanden	keine Dokumentation vorhanden
	2	Breite der Veränderung	Eingegrenzter Bereich	Projekt	Unternehmensbereich	Neuaufbau des Unternehmens
	3	Tiefe der Veränderung	„KVP“ – Kontinuierlicher Verbesserungsprozess	mittelstarke Veränderung einzelner Prozesse	starke Veränderung wesentlicher Prozesse	Radikale Veränderung
	4	Modellierungserfahrung	Methodennutzer sind geschult	MBSE-Kernteam	einige Mitarbeiter mit Erfahrung	keine Kompetenz im Unternehmen

- **Modellierungserfahrung:** MBSE ist selbst für erfahrene Systems Engineers Neuland. Der Aufwand für die Änderung der Arbeitsweisen wird also auch durch die Modellierungserfahrung beeinflusst. Er ist dabei umso niedriger, je größer die Erfahrung mit MBSE und der Arbeit auf Basis eines Systemmodells im Unternehmen ist. Im Idealfall sind die Stakeholder im MBSE geschult.

Aus der Bewertung wird eine Punktzahl über alle Kriterien gemittelt. Letztlich ergibt sich eine Aussage, welche Methoden auf Basis des Systemmodells zunächst eingeführt werden sollten. Bild 4-21 zeigt einige Methoden aus Bild 4-11 im resultierenden Bewertungs-Portfolio. Methoden links oben eignen sich zur schnellen und einfachen Anwendung auf Basis des Systemmodells. Das ist z.B. die Stakeholderanalyse. Methoden rechts unten im Portfolio sollten vorerst nicht umgesetzt werden – wie z.B. die Carry-Over Analyse. Insgesamt zeigt sich, dass bspw. die Methoden der Stakeholderanalyse, der Risikobewertung und der Systemkomplexitätsbewertung gut auf Basis des Systemmodells anwendbar sind. Dies sollte im Steckbrief entsprechend vermerkt werden, inkl. der Zusatzinformationen, zu welchem Meilenstein im Projekt die Methode relevant ist und ob sie einmalig oder mehrmals im Projekt durchgeführt werden muss.

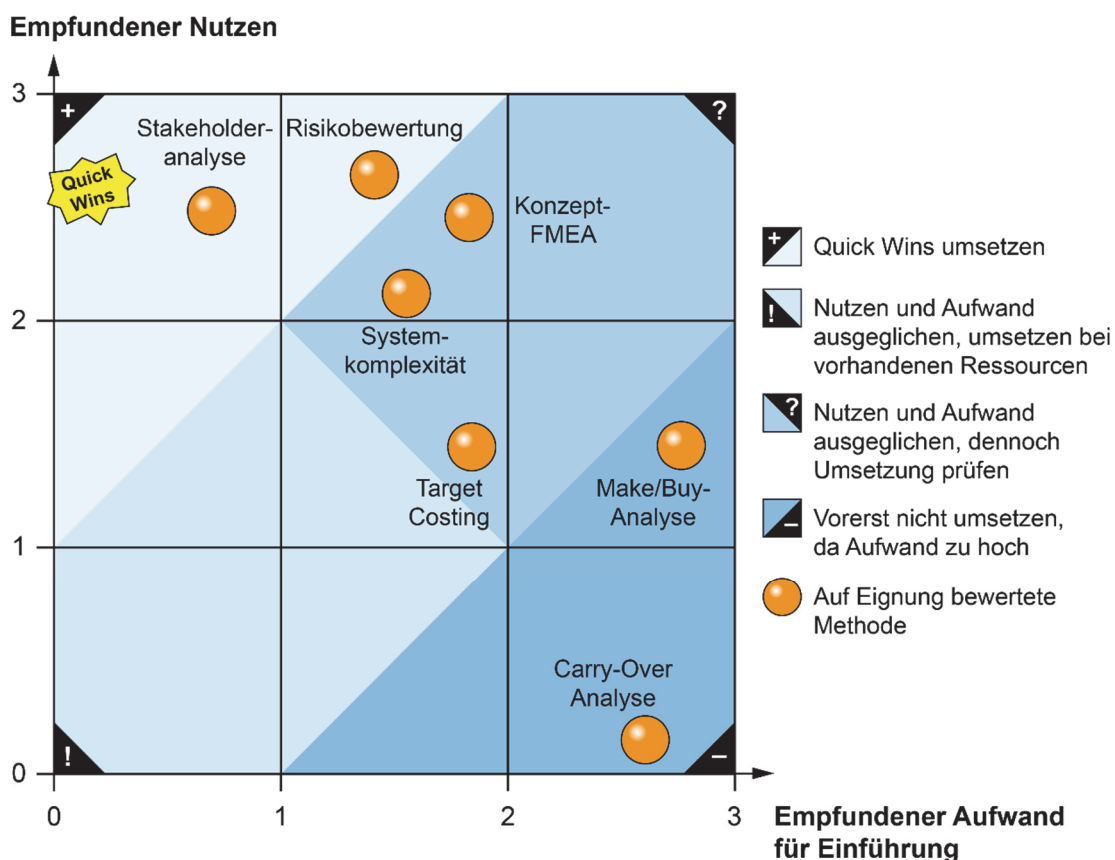


Bild 4-21 Anwendung des Quick Checks für ausgewählte Methoden aus Bild 4-11

4.6.1.2 Define – Informationsgewinnung, -verarbeitung, -ausgabe

In der Define-Phase adressieren die Aktivitäten der **Informationsgewinnung** (IG) Aspekte der zu koppelnden Methode. **Informationsverarbeitung** (IV) und **-ausgabe** (IA) stehen für die informationstechnische Realisierung der Methodenkopplung. Die Aktivitäten werden nur bei entsprechendem Modellierungszweck durchlaufen (vgl. Bild 4-18). Die Informationen werden in den Schritten Grobanalyse der Methode – Analyse der Stakeholderinteraktionen – Grobkonzept IT-Infrastruktur zusammengetragen. Das Pha-

sen-Meilenstein-Diagramm in Bild 4-22 fasst die spezifischen Aufgaben und Methoden zusammen, die Zugehörigkeit zur entsprechenden Aktivität ist dargestellt.

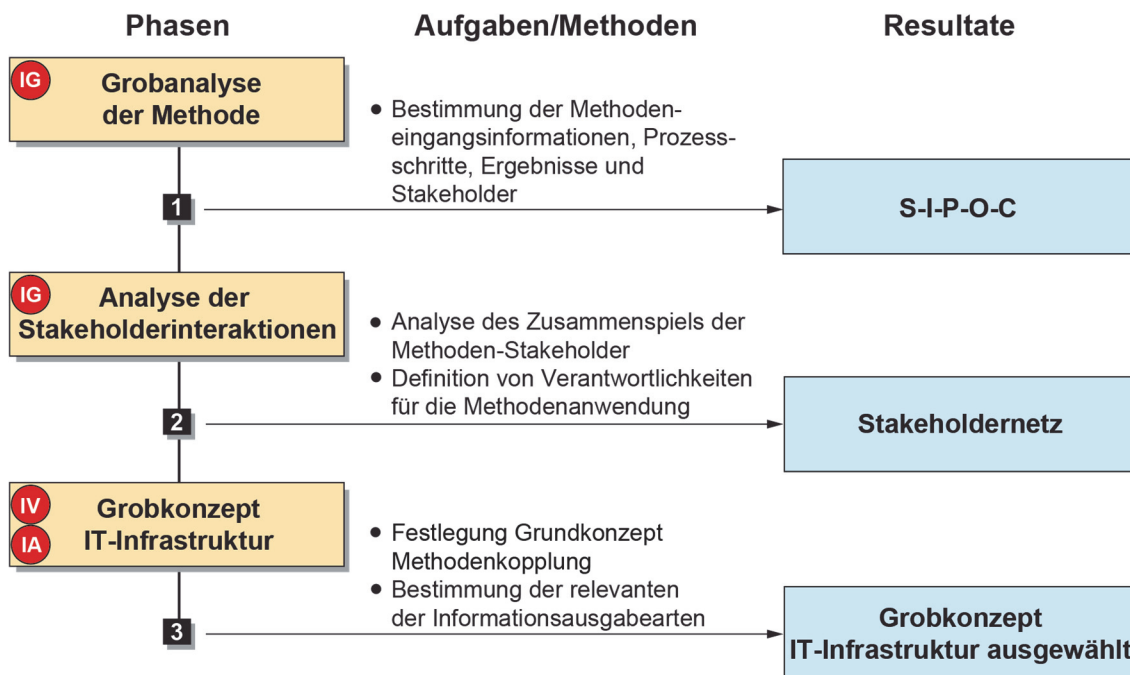


Bild 4-22 Aufgaben in der Define-Phase für die Aktivitäten Informationsgewinnung, -verarbeitung und -ausgabe

Grobanalyse der Methode – S-I-P-O-C

Mit dem S-I-P-O-C existiert im Six Sigma ein einfaches Hilfsmittel zur Prozessanalyse, mit dem die zentralen Stakeholder der Methode identifiziert und deren Inputs bzw. Outputs verknüpft werden¹⁷. Bei der Analyse mit S-I-P-O-C sollten beim vorgestellten Vorgehen zusätzlich auch Stakeholder aus dem Kontext der Systemmodellierung (z.B. nach SHEARD) als *Customer* ermittelt werden. Das bedeutet: Die Methode kann dann nicht nur auf Basis des Systemmodells angewendet werden, sondern ihre Ergebnisse spielen auch eine Rolle für die weitere Systemmodellierung. In Bild 4-23 ist ein derartiges S-I-P-O-C für die Stakeholderanalyse dargestellt. Die Ergebnisse der Stakeholderanalyse sind bspw. für den Systemanalysten von Bedeutung (hier fett markiert, grün als Abschluss des S-I-P-O-C): Er formuliert die Use Cases für das Produkt, die stark von den Stakeholdern abhängig sind. Die Anzahl möglicher Use Cases eines Systems ist allerdings grundsätzlich sehr hoch; durch die Priorisierung der Stakeholder können nun auch die Use Cases entsprechend priorisiert werden.

¹⁷ Im Prinzip wird auf diese Weise ein Phasen-Meilenstein-Modell für eine Methode abgebildet, ohne jedoch die Methode durchführbar zu machen.

Analyse der Stakeholderinteraktionen – Stakeholdernetz

Auf Basis des S-I-P-O-C werden in einem weiteren Schritt die Zusammenhänge zwischen den identifizierten Stakeholdern beschrieben. Das kann bspw. mit einem Stakeholdernetz geschehen: Die Stakeholder (Knoten) werden über Beziehungspfeile (Kanten) miteinander verbunden. Die Pfeilrichtung definiert, ob die Stakeholder einen Input zur Methode beisteuern oder einen Output ziehen. Die Informationen zur Erstellung des Stakeholdernetzes stammen aus dem S-I-P-O-C. Das Element mit den meisten eingehenden Pfeilen ist oft der „Methodeneigner“ und sollte deshalb auch maßgeblich in die Modellierung eingebunden sein. Die hier vorgeschlagene abstrakte und informale Darstellungsform ist in dieser Phase absolut ausreichend.

Zusätzlich zur Darstellung der Zusammenhänge im Stakeholdernetz sind die Verantwortlichkeiten im Kontext der Methodenanwendung und der Systemmodellierung festzulegen. Die Zuordnung von Verantwortlichkeiten bei der Methodenanwendung gelingt bspw. mit der RASCI-Nomenklatur:

- **R** – *responsible*, verantwortlich für die Aufgabe, führt sie auch operativ durch und berichtet direkt an „A“
- **A** – *accountable*, gesamtverantwortlich für sämtliche Tätigkeiten
- **S** – *supportive*, unterstützt die verantwortlichen Stakeholder
- **C** – *consulted*, ist einzubinden bevor Entscheidungen getroffen oder wesentliche Maßnahmen eingeleitet werden
- **I** – *informed*, wird über das Ergebnis in Kenntnis gesetzt, hat aber keinen Einfluss auf das Ergebnis

Zusätzlich zum S-I-P-O-C sind in Bild 4-23 auch die Stakeholderinteraktionen und die Verantwortlichkeiten dargestellt. Im vorliegenden Fall wird identifiziert, dass der Systemanalyst über die Ergebnisse der Stakeholderanalyse informiert werden muss – die Beweggründe hierfür wurden bereits erläutert. Da die Festlegung solcher Zusammenhänge eine gewisse Erfahrung benötigt, ist aber ähnlich zum Six Sigma ein Pate verantwortlich für diese Phase. Ebenso wird deutlich, dass der Produktmanager für die Durchführung der Methode verantwortlich ist, aber auch selbst die Ergebnisse für weitere Aufgaben benötigt: Sämtliche Kanten liegen am Knoten Produktmanager an. Zur Durchführung der Methode wird zudem Unterstützung aus den Bereichen Marktanalyse und der Rechtsabteilung benötigt. Eine derartige Analyse sollte für jede ausgewählte Methode durchgeführt werden.

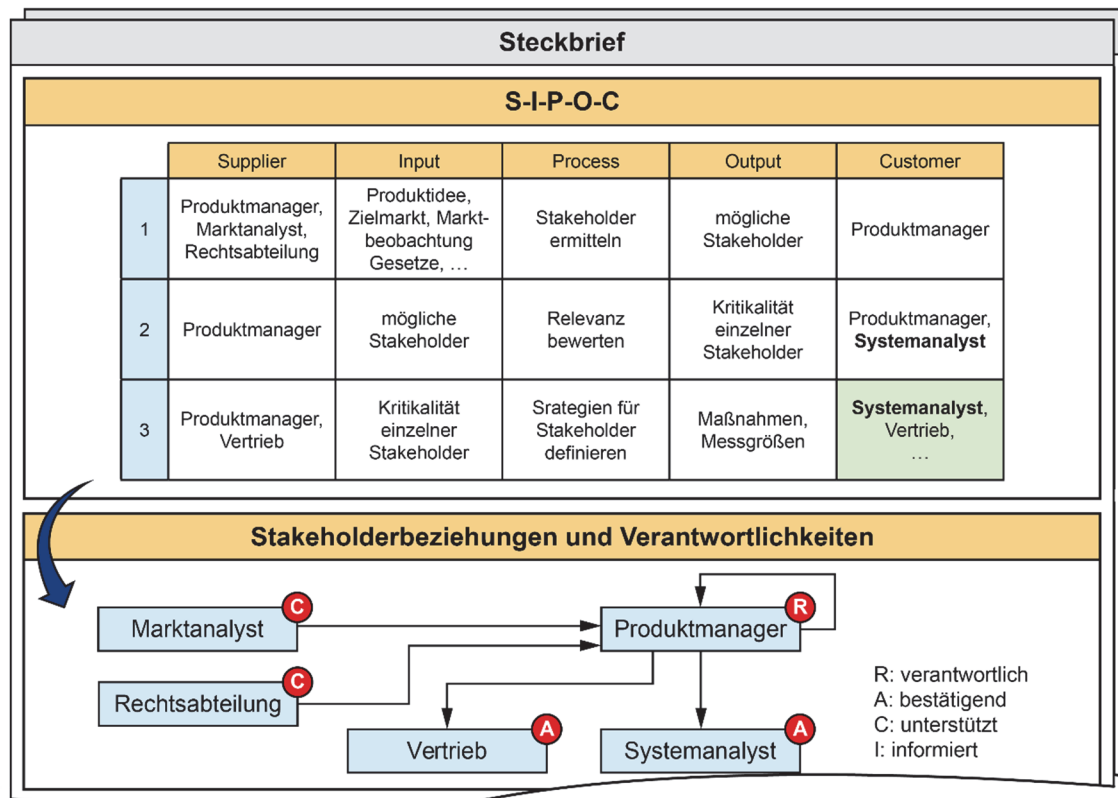


Bild 4-23 Beispielhafter Steckbrief „Planung des Modellierungsprojekts“ Teil 2/2

Grobkonzept IT-Infrastruktur

Es muss frühzeitig geklärt werden, wie die Informationen des Systemmodells für unterschiedliche Aufgaben genutzt werden sollen. Hinsichtlich der Unterstützung von Methoden stehen zwei grundsätzliche Ansätze zur Auswahl: Ein integrierter und ein föderierter Ansatz. Bild 4-24 zeigt den konzeptionellen Unterschied dieser beiden Ansätze.

- Übertragen auf das MBSE werden beim integrierten Ansatz die Methoden direkt im Modellierungswerkzeug umgesetzt. Das kann bspw. direkt mit der gewählten Modellierungssprache gelingen: Baut die Sprache auf der SysML auf, kann mittels Parameterdiagramm die Methode direkt in das Werkzeug (bzw. das Modell) integriert werden. Alternativ kann die Methode auch in den gleichen Werkzeugen mit den häufig integrierten Programmiersprachen umgesetzt werden, z.B. JavaScript oder Visual-Basic.
- Beim föderierten Ansatz werden die Methoden außerhalb des Modellierungswerkzeugs ausgeführt – mittels spezifischer Anwendungswerkzeuge. Die für die Methode benötigten Inputs müssen aus dem Systemmodell über Schnittstellen in eine externe Anwendung übertragen werden. Dann können Daten sowohl im als auch außerhalb des Werkzeugs manipuliert werden.

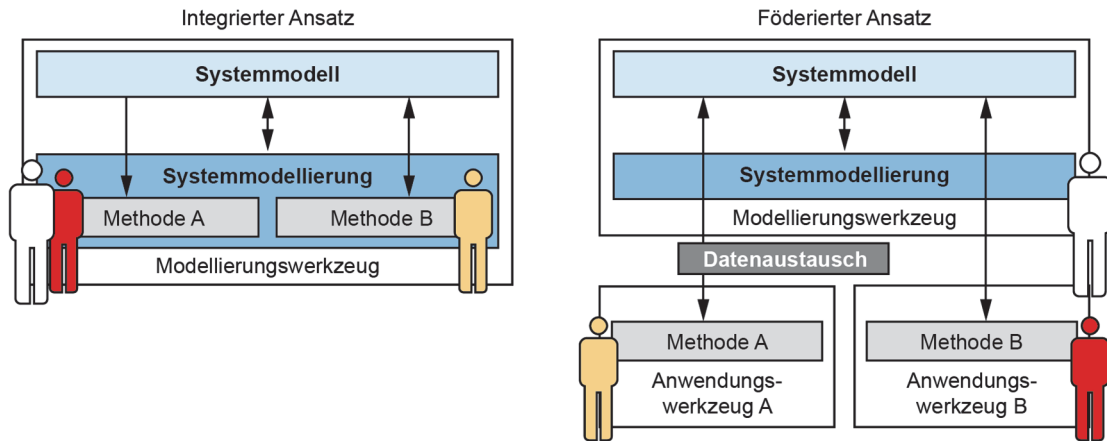


Bild 4-24 Systemmodellzugriff – zwei Alternativen

Vor dem Hintergrund der in Kapitel 2 skizzierten Veränderungen und Herausforderungen stellt sich der integrierte Ansatz tendenziell nachteilig dar: Die meist aus der Softwaretechnik stammenden Werkzeuge werden zum zentralen Arbeitsmedium sämtlicher Stakeholder. Beim föderierten Zugriff können die Stakeholder dagegen ihre Werkzeuge nutzen. Die Modellierung wird durch eine SE-spezifische Rolle durchgeführt (vgl. bspw. [She96], [Zin13], [Kai14]), die übrigen Stakeholder unterstützen oder führen ergänzende Aktivitäten durch. Letzteres ist deshalb wichtig, da die hier betrachteten Methoden im Regelfall nicht vollständig auf Basis der im Systemmodell vorhandenen Daten ausgeführt werden können, wie anhand des in Bild 4-23 dargestellten S-I-P-O-C für die Stakeholderanalyse nachvollzogen werden kann. So können sämtliche Stakeholder vom Konzept des MBSE profitieren, ohne detailliert mit den Konzepten der Systemmodellierung vertraut zu sein.

Wie eine Methode konkret auf Basis des Systemmodells dargestellt wird, wird in der Aktivität Informationsausgabe festgelegt. Bild 4-25 stellt unterschiedliche Ausgabeformate dar, die im Kontext der Produktentstehung üblich sind. Für die hier betrachteten Methoden ist die Informationsausgabe im Regelfall die Visualisierung.



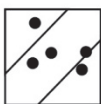
Ausgabeformat	Erklärung
Dokument 	Ein Aspekt der Grundidee des MBSE ist die Ausleitung konsistenter Dokumente auf Basis der grafischen Systemspezifikation. Häufig ist ein Textdokument Standard in der Produktentstehung - Begriffe wie „Lastenheft“ unterstreichen dies.
Datenformat 	Im Zuge der Virtualisierung können Methodenergebnisse in unterschiedlichen Datenformaten gesichert werden. Das sind meist standardisierte Formate, wie z. B. xml. Zum Austausch von Simulationsmodellen hat sich FMI (Functional Mock-Up Interface) etabliert.
Visualisierung 	Bei Methoden im Kontext des Entwicklungsmanagements ist es üblich, die Ergebnisse einer Methode zu visualisieren. Das dient dem einfacheren Verständnis der Ergebnisse und einer leichteren Kommunikation über das Ergebnis. <input checked="" type="checkbox"/>

Bild 4-25 Ausgabeformate

4.6.2 Analyse-Phase

Die Analyse-Phase erstreckt sich über alle Aktivitäten der Informationsverarbeitung. Sie umfasst den zweiten Teil der Modellplanung, adressiert aber auch den Hauptaufgabenbereich der Methodenkopplung. In diesem Kapitel stehen die Aktivitäten der Informationserzeugung und Informationsgewinnung im Vordergrund. Sie gehören noch zur Modellplanung (vgl. Bild 4-17) und dienen somit der Vorbereitung und Planung des Systemmodells. Die Ergebnisse sollten in den Steckbriefen nachgepflegt werden.

4.6.2.1 Analyse-Phase: Informationserzeugung

Die Aktivität der Informationserzeugung (IE) adressiert weiterführende Planungen des Modellierungsprojekts. Aufbauend auf der Define-Phase steht nun – ähnlich zum Six Sigma – eine Detailanalyse an. Ziel ist, sämtliche organisatorische Vorbereitungen für die tatsächliche Modellierung vorzunehmen. Das Phasen-Meilenstein-Diagramm in Bild 4-22 stellt die Aufgaben und Methoden für die Aktivität Informationserzeugung dar. Das betrifft zunächst die Auswahl eines geeigneten MBSE-Konzepts. Hier unterstützen die in den Kapiteln 4.2 und 4.3 erarbeiteten Hilfsmittel. Es folgen die Auswahl einer Modellierungsmethode und die Organisation der Modellierung.

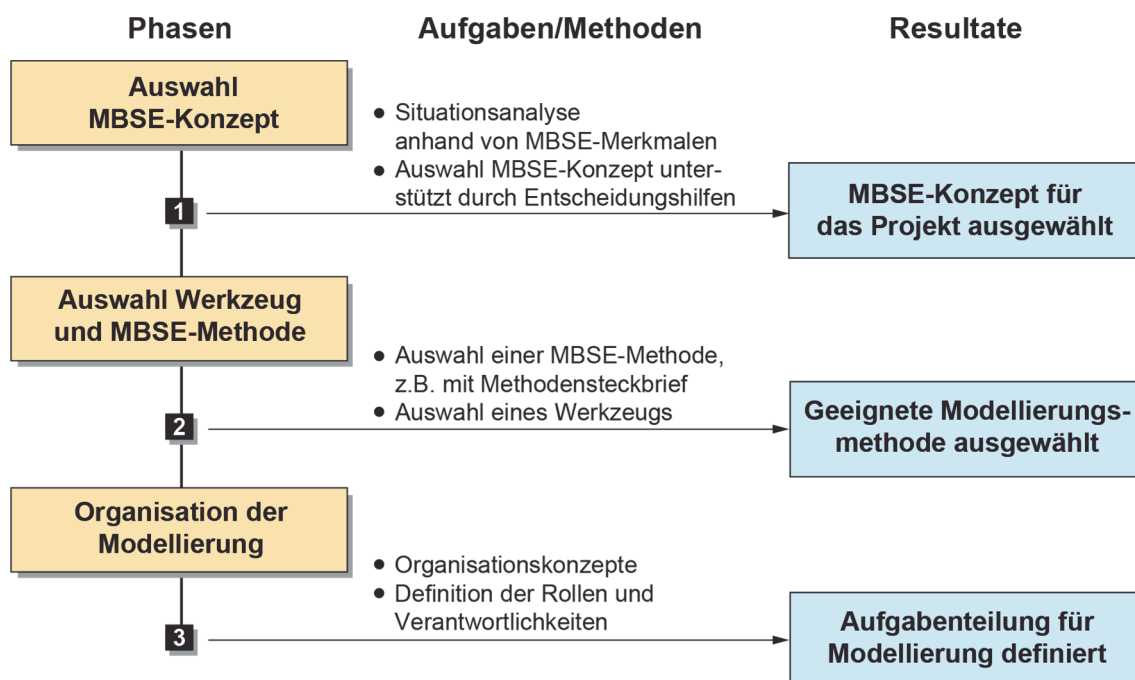


Bild 4-26 Aufgaben in der Analyse-Phase – Informationserzeugung

Auswahl MBSE-Konzept

Die Auswahl eines MBSE-Konzepts gelingt unter Rückgriff auf die in Kap. 4.2 und 4.3 bereitgestellten Hilfsmittel. Die Entscheidung sollte sorgfältig abgewogen werden:

- Für die langfristige und weitreichende Integration bietet sich Konzept 3 an.
- Für das prototypische Ausprobieren ist auch Ansatz 2 geeignet.

Die analytische Auswahl eines Konzepts kann mit einem morphologischen Kasten als einfachem Hilfsmittel unterstützt werden: Bei der prototypischen Klassifikation reicht die Ähnlichkeit zu einem MBSE-Konzept für die Zuordnung aus. Bild 4-27 zeigt einen Ausschnitt eines morphologischen Kastens mit fiktiven (Unternehmens-)Daten für das Kaffeemaschinenprojekt – hierbei stehen die drei Hauptklassen *Werkzeug*, *Unternehmen und Projekt* und *Anwendungsaspekte* der in Kapitel 4.2.1 erarbeiteten Merkmale im Mittelpunkt. Demnach könnte auch das Konzept 2 Mechatronic Sketch geeignet sein.

		Merkmalsausprägungen	Merkmalausprägungen			
Werkzeug	1	Werkzeugart	Brown Paper	Diagramming Tool	COTS/ Dedicated Tool	Simulations- werkzeug
	2	PLM Unter- stützung	ja		nein	
Unternehmen und Projekt	3	Unternehmensgröße	klein		groß	
	4	Projektgröße	klein	mittel	groß	
	5	Projektart	Vorentwicklung	Kundenspezifische Entwicklung	Serienentwicklung	
	6	Wertschöpfungs- kette	projektintern	unternehmensintern	Wertschöpfungskette	
Anwendung	7	System-of- Interest	SoS	System	Modul	
	8	Modellierungs- verantwortung	Workshop	Architekt	Fachspezialist	
	9	Anwendungs- bereich	Management	Technik	Enabler	

 bewertet im Rahmen von Projekt TVP GfSE  idealtypische Ausprägung von MBSE-Konzept 2 im Vergleich (vereinfachte Darstellung)

Bild 4-27 Bewertung des Projekts im morphologischen Kasten

Für eine rationale Entscheidung sollte die Auswahl durch die in Kapitel 4.3 erarbeiteten Entscheidungshilfen abgesichert werden: Mit den verschiedenen Blickwinkeln können sich für die geplanten Methoden *Stakeholderanalyse*, *Risikobewertung* und *Komplexitätsbewertung* (vgl. Bild 4-21) durchaus andere Empfehlungen ergeben. Sollte danach keine eindeutige Empfehlung vorliegen, ist eine strategische Entscheidung zu treffen: Ist eine dauerhafte Integration des MBSE in die Produktentstehungsarbeit geplant, wird Konzept 3 das Mittel der Wahl sein; die Konzepte 1 und 2 werden dagegen für spezifische Situationen gewählt (z.B. Workshops, Abstimmungen mit Kunden, ...).

Aus Gründen der Einfachheit und zur Demonstration der Vorgehensweise werden im weiteren Verlauf die betrachteten Methoden mit Konzept 2 – Mechatronic Sketch und ebenso Konzept 3 abwechselnd angewendet. Die Planung ist bei beiden Ansätzen identisch, die operative Umsetzung unterscheidet sich jedoch.

Auswahl einer MBSE-Modellierungsmethode

Unabhängig vom gewählten MBSE-Konzept sollte eine Modellierungsmethode bewusst ausgewählt werden. Für einen besseren Überblick über die große Masse unterschiedlicher Modellierungsmethoden sieht das Rahmenwerk eine Struktur für einen Steckbrief vor, der kontinuierlich gepflegt werden sollte (vgl. Bild 4-28 und Bild 4-29). Der Steckbrief umfasst alle wesentlichen Daten zur Methode: Das sind zunächst Metadaten zur Methode (Bild 4-28), die zwar einen gewissen Einfluss auf die Auswahlentscheidung haben, im direkten Methodenvergleich jedoch nur begrenzt helfen – bspw. Informationen zum Erscheinungsjahr, zum aktuellen Release und ein Kurzüberblick mit Angaben zur Anzahl an Phasen, empfohlenen Diagrammen und zum Verbreitungsgrad. Diese Angaben sollten regelmäßig auf Aktualität überprüft werden.





Methoden-Steckbrief CONSENS – Metadaten		Stand: 3. Nov. 2015	
Methoden-Historie		Kurzüberblick	
Name:	CONceptual design Specification technique for the ENgineering of complex Systems	# Phasen:	2 Hauptphasen • Analyse des Problems • Definition d. Systemarchitektur
Akronym:	CONSENS	# Diagrammarten:	7 (bzw. 8 bei ITS) Partialmodelle
Erscheinungsjahr:	2006	Diagrammarten:	Umfeld Anwendungsszenarien Anforderungen Funktionshierarchie Wirkstruktur Verhalten Gestalt (Zielsystem)
Fachlicher Ursprung:	Maschinenbau	Verbreitungsgrad:	hoch lokal / mittel national
Entwickler:	Heinz Nixdorf Institut (PB) Fraunhofer IEM (PB)		
Aktuelles Release:	-		
Nächstes Release:	keine Release-Strategie bekannt		
Release-Strategie:	-		
Informationen:	www.selive.de		
Werkzeugunterstützung		Kurzbeschreibung	
MBSE-Konzept	 • CONSENS-Kartenset	CONSENS ermöglicht die domänenübergreifende Beschreibung der Prinziplösung mechatronischer Systeme. Die Prinziplösung legt den grundsätzlichen Aufbau und die Wirkungsweise des Systems fest. Dabei berücksichtigt CONSENS auch die Entwicklung Intelligenter Technischer Systeme (ITS). Die Prinziplösung ist die Grundlage für die weitere Konkretisierung der Entwicklungsarbeit. Die Prinziplösung eines mechatronischen Systems wird durch CONSENS in acht Aspekte gegliedert. Die Aspekte	
	 • MS Visio-Shapes4CONSENS		
	 • SysML4CONSENS Profil • Mechatronic Modeller		
	 • -		

Bild 4-28 MBSE-Modellierungsmethoden: Beispielhafter Steckbrief 1/2 mit Metadaten

Merkmale der Methode, die eine Vergleichbarkeit ermöglichen, werden zusätzlich in einer Morphologie aufbereitet. Der Steckbrief in Bild 4-29 schlägt eine erste Morphologie für die Modellierungsmethoden des MSBE vor, hier beispielhaft für CONSENS angewendet. Sie umfasst aktuell 14 Merkmale, die in die vier Hauptklassen *Methodeninhärente Merkmale*, *Anwendung*, *Organisation* und *Werkzeug* strukturiert sind.

- *Methodeninhärente Merkmale* sind im Kern der Methode verankert. Sie können nicht verändert werden, ohne den Charakter der Methode in Frage zu stellen.

- Die Hauptklasse *Anwendung* beschreibt Merkmale, die im Rahmen der Entwicklung der Methode explizit festgelegt wurden und im Rahmen einer Weiterentwicklung veränderbar sind.
- *Organisation* fasst zusammen, wie die Modellierung durch die Stakeholder genutzt und gesteuert wird.
- Die Klasse *Werkzeug* adressiert Aspekte, die im Zusammenhang von Methode und Softwareunterstützung stehen.

In diesen Hauptklassen sind die im Rahmen der MBSE-Konzeptentwicklung in Kapitel 4.2 identifizierten Merkmale enthalten¹⁸. Die Merkmale resultieren aus der Analyse der Methoden in Kapitel 3 und können um weitere Merkmale ergänzt werden – die Punkte in der letzten Zeile einer Klasse bringen das zum Ausdruck. Durch diese erste einfache Morphologie wird aber schon ersichtlich, wie sich die Modellierungsmethoden unterscheiden: So fällt bspw. bei Sysmod auf, dass es sich in Ansätzen mit Rollenprofilen beschäftigt – die Methode CONSENS dagegen nicht.

	Merkmale	Ausprägungen		
Methodeninhärente Merkmale	Phasenablauf	<i>iterativ</i>		inkrementell
	Phasendefinition	formlos	formal	
	Sprachenkompatibilität	eine	<i>mehrere</i>	
	Sprachausrichtung	schwach	<i>stark</i>	
	Skalierbarkeit	<i>gegeben</i>		nicht gegeben
	...			
Anwendungsbezogene Merkmale	Modellierungszweck	<i>Prinziplösung</i>	<i>Kommunikation</i>	<i>Dokumentation</i>
	Anforderungsermittlung	<i>request-driven</i>	feature-driven	usage-driven
	Systemvalidierung	<i>nicht Bestandteil</i>	Leistungskriterien	Testvorgaben
	Variantenmodellierung	ja	<i>nein</i>	
	Hilfestellungen	keine	<i>Hinweise</i>	dedizierte Vorgaben
	...			
Organisation	Methodeneigner	nicht definiert	Systemarchitekt	<i>SM-Verantwortl.</i>
	Rollenprofile	<i>keine Vorgaben</i>	idealis. Vorschläge	klar definiert
	...			
Werkzeug	Werkzeugabhängigkeit	<i>niedrig</i>		hoch
	Softwarezugänglichkeit	einfach	<i>schwierig</i>	
	...			

Bild 4-29 MBSE-Modellierungsmethoden: Steckbrief 2/2 mit Morphologie

¹⁸ Die Merkmale und ihre Ausprägungen werden in Anhang A4 beschrieben, sofern nicht anderswo in dieser Arbeit bereits aufgegriffen.

Auswahl eines Modellierungswerkzeugs

Die Auswahl eines für die Modellierung geeigneten Werkzeugs wird durch das ausgewählte MBSE-Konzept beeinflusst und kann über das Merkmal Werkzeugart und die entsprechenden Merkmalausprägungen unterstützt werden (vgl. Bild 4-7). Noch stärkeren Einfluss auf die Werkzeugauswahl hat die zuvor ausgewählte Methode: Häufig existieren schon Profile einer MBSE-Methode für die auf dem Markt verfügbaren Werkzeuge. Für die hier ausgewählte Methode CONSENS existieren bspw. Modellierungs-Shapes für das Diagramming-Werkzeug MS Visio – das ermöglicht die einfache Anwendung von CONSENS im Konzept 2 Mechatronic Sketch. Ebenso existiert ein Profil für das COTS-Werkzeug Enterprise Architect [IKD+13]. Für die in Kapitel 3 ebenfalls vorgestellte Methode SysMod existieren bspw. SysML-Profile für das COTS-Werkzeug Cameo Systems Modeler und ebenso für das Werkzeug Enterprise Architect. Die Anwendung von Konzept 2 Mechatronic Sketch wird dadurch tendenziell nicht so gut unterstützt wie bei CONSENS, was bspw. „*das Erlernen*“ des MBSE in kleineren Projekten nicht so begünstigt. Diese Informationen sind ebenfalls in den Metadaten der Steckbriefe dokumentiert und sollten fortlaufend aktuell gehalten werden.

Organisation der Modellierung

Schlussendlich muss die Erstellung des Systemmodells organisiert werden, d.h. die Verantwortlichkeiten und die Zusammenarbeit einzelner Stakeholder während der Modellierung müssen geklärt sein. Die in Kapitel 3 analysierten Modellierungsmethoden zeigen keine Formen der Arbeitsorganisation auf. FRIEDENTHAL, WEILKIENS und KAISER nennen rudimentäre Rollen, die in Kapitel 2 zitierte Umfrage unter Systems Engineers [TKD+14] zeigt Ansätze der Aufgabenverteilung – wengleich alle kein eindeutiges Konzept präsentieren. In Bild 4-30 sind sechs Möglichkeiten zur Organisation der Modellierung dargestellt, die im Rahmen der Arbeit gestaltet wurden:

- 1) Klassisches, in der Literatur geprägtes Bild der Modellierung; der Systems Engineer modelliert und nutzt gleichzeitig das Modell für seine Aufgaben.
- 2) Sehr stark an die Rolle des Systemmodellverantwortlichen von KAISER [Kai14] angelehnt; der SE konsolidiert Wissen verschiedener disziplinspezifischer Rollen und bereitet es auf. Das im Systemmodell bereitgestellte Wissen dient sowohl der eigenen Arbeit als auch den übrigen Wissensträgern.
- 3) „Shared Service“ – Angelehnt an das aus der Dienstleistung stammende gleichnamige Konzept [Dre07]; einzelne und sehr spezielle Modellierungsaufgaben werden von einer zentralen Gruppe von Spezialisten übernommen und für Dritte allgemein bereitgestellt. Die Spezialisten sind im dem Fall bspw. die SE-Rollen nach SHEARD [She96], oder wie im SE-Team nach FRIEDENTHAL [FMS12].
- 4) Unterstützung von fachlichen Rollen durch SE-Spezialisten im Sinne von Coaching; als Erweiterung von 3) wird das Systemmodell gemeinsam zur Nutzung durch die fachliche Rolle erstellt.

- 5) „Kollaboratives MBSE“ – Angelehnt an [TKD+14] erstellen sämtliche Stakeholder der Produktentstehung das Systemmodell zur gemeinschaftlichen Nutzung. Die erzeugten Informationen können dabei auch Dritten zur Verfügung stehen – natürlich prinzipiell auch dem Systems Engineer.
- 6) Als Mischkonzept aus 2) und 5) legt ein Systems Engineer den Grundstein der Systemmodellierung, im Zeitverlauf werden Inhalte des Systemmodells für fachspezifische Aufgaben genutzt und neue Inhalte durch diese Rollen ergänzt.

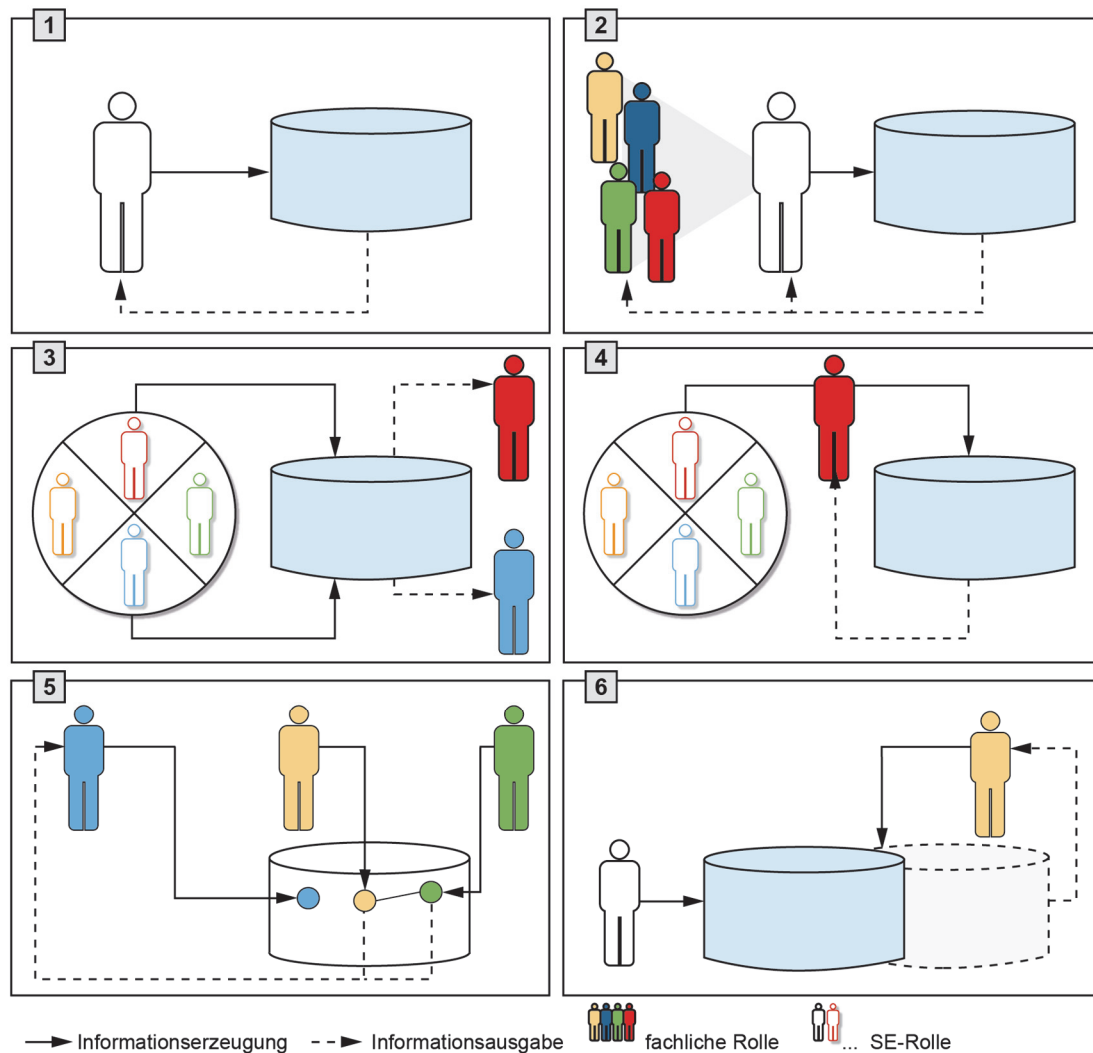


Bild 4-30 Möglichkeiten zur Organisation der Modellierung in MBSE-Projekten

Gerade die im Rahmen der Organisationskonzepte 4) bis 6) dargestellte Rollenteilung in der Modellierung ist für die in Kapitel 2 beschriebenen Herausforderungen und die hier angedachte Nutzung des Systemmodells zur Unterstützung der Produktentstehungsarbeit geeignet. Auf Basis des S-I-P-O-C und des Stakeholdernetzes in Kapitel 4.6.1 sticht das hier dargestellte Konzept 5) als geeignet für die Stakeholderanalyse hervor.

4.6.2.2 Analyse-Phase – Informationsgewinnung

Hier geht es darum, eine Methode so aufzubereiten, dass sie auf Basis des Systemmodells unterstützen kann. Am Beispiel der in Kapitel 4.6.1 ausgewählten Stakeholder-Analyse wird das dazugehörige Vorgehen weiter detailliert. Konkret heißt das: Informationen, die bislang nicht virtualisiert wurden, müssen einerseits im Systemmodell abgebildet werden, andererseits für weiterführende Aufgaben – je nach Infrastruktur – auch in anderen Anwendungsprogrammen nutzbar sein. Konkret stehen in dieser Phase die in Bild 4-31 genannten Aufgaben an: Methodenanalyse und die Sichtenplanung.

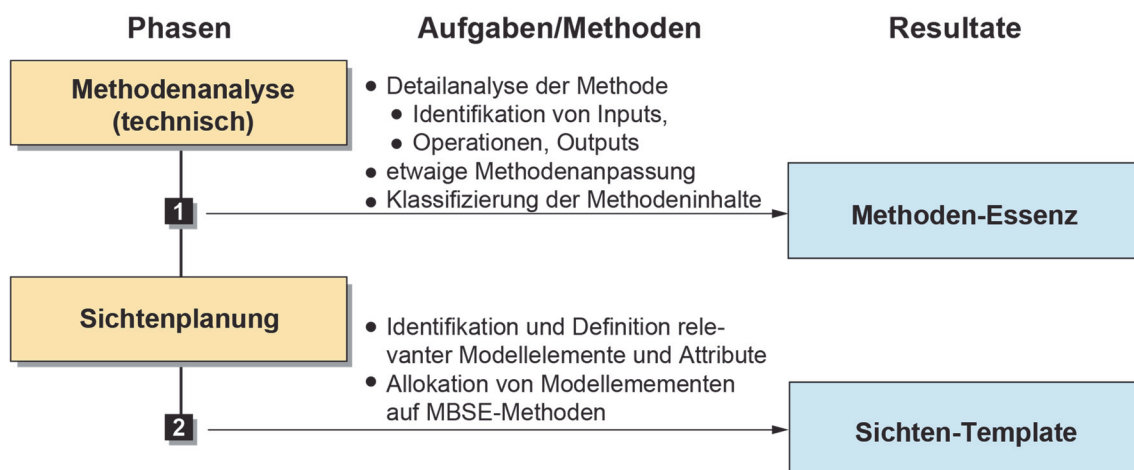


Bild 4-31 Aufgaben in der Analyse-Phase – Informationsgewinnung

Technische Methodenanalyse

Die Stakeholder der adressierten Methode, ihre Wirkbeziehungen und der grobe Ablauf sind bereits bekannt (vgl. Kapitel 4.6.1). Für die erfolgreiche Kopplung der Methode mit dem Systemmodell werden nun ihre technischen Aspekte näher betrachtet. Das PoMM nach BIRKHOFER hat gezeigt, dass der Methodenkern ihre Struktur ist, mit der die Transformation eines Inputs in einen Output gelingt. Diese wird nun erarbeitet.

Mit dem entwickelten Vorgehen zur Methodenanalyse wird die Transformation des Methodeninputs in den gewünschten Output detailliert untersucht, so dass das Systemmodell zur Methodendurchführung genutzt werden kann. Als Ergebnis ist klar,

- welche Inhalte der Methode im Systemmodell abgebildet,
- welche Aufgaben durch eine dazugehörige Anwendung unterstützt,
- welche Informationen im Systemmodell zusätzlich gesichert werden,
- welche Ergebnisse im Entwicklungsprojekt weiterverwendet werden sollen.

Rund um die Beschreibung der eigentlichen Methodenschritte sind in den üblichen Darstellungen zahlreiche Informationen verfügbar, die nichts mit der eigentlichen Methodenstruktur zu tun haben. Die Beschreibungen werden mit der technischen Metho-

denanalyse in einem ersten Schritt so reduziert, dass die *Essenz* der Methode vorliegt, wobei hier auch eine Anpassung der Methode erfolgen kann. Als *Essenz* wird hier der Teil der Methode verstanden, der sich auf die reine Erzeugung, Gewinnung, Verarbeitung und Ausgabe der methodenrelevanten Informationen bezieht.

Bei der Analyse der Methode wird auffallen, dass sich unabhängig von der Beschreibungsart diese *Essenz* als eine Kombination von Substantiven und Verben darstellen lässt. Das ist auf das Konzept der Informationsverarbeitung zurückzuführen. Bild 4-32 stellt dieses Verständnis anhand der ersten Phase der Stakeholderanalyse nach MÜLLER-STEWENS und LECHNER [ML11] beispielhaft dar: Der erste Schritt „*Auflistung aller relevanten Stakeholder*“ ist im Gegensatz zum zweiten essentiell für die Methodenanwendung und kann in eine *Substantiv-Verb-Kombination* umformuliert werden.

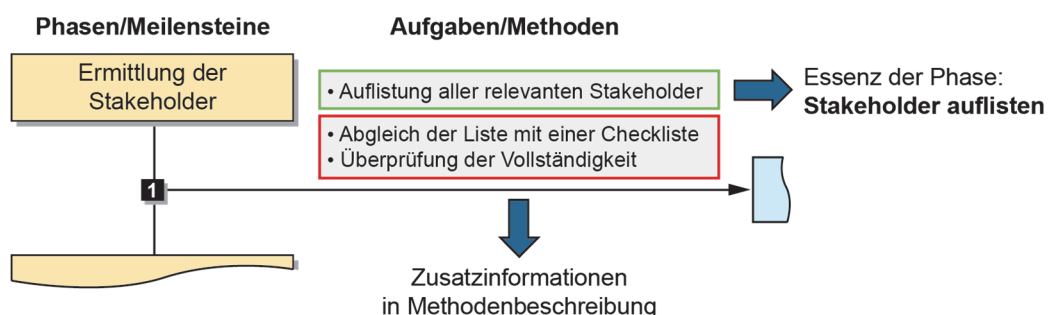


Bild 4-32 Essenz der Phase Ermittlung der Stakeholder

Wie die *Essenz* weiter genutzt wird, ist in Bild 4-33 dargestellt. Hierbei ist wichtig, zusätzlich die Substantive und Verben differenziert zu unterscheiden. Gerade bei Verben sticht deutlich hervor, dass sich i.w.S. drei unterschiedliche Arten von Verben unterscheiden lassen: Verben des Sammelns, des Bewertens und des Schlussfolgerns.

- Verben des Sammelns (V_S), bspw. *auflisten*, *suchen*, *identifizieren*. Sie beziehen sich auf Substantive, die im Regelfall zu modellierende Systemelemente sind.
- Verben des Bewertens (V_B), z.B. *bewerten*, *klassifizieren*. Sie deuten auf Aktivitäten, die die Transformation des Methodeninputs in den -output beschreiben.
- Verben des Schlussfolgerns (V_{SF}): Sie deuten auf einen Methodenoutput, der im Prozess oder anderen Methoden verwendet wird, z.B. das Verb *ableiten*.

Teilweise ist aufgrund der Sprachvielfalt eine Zuordnung zu einer der Klassen nicht ganz eindeutig. Dann gilt es im Kontext zu entscheiden. Oft sind – wie im obigen Beispiel – auch keine Verben vorhanden, s.d. eine Desubstantivierung der relevanten Substantive vorgenommen werden muss; so wird aus *Auflistung* das Verb *auflisten*.

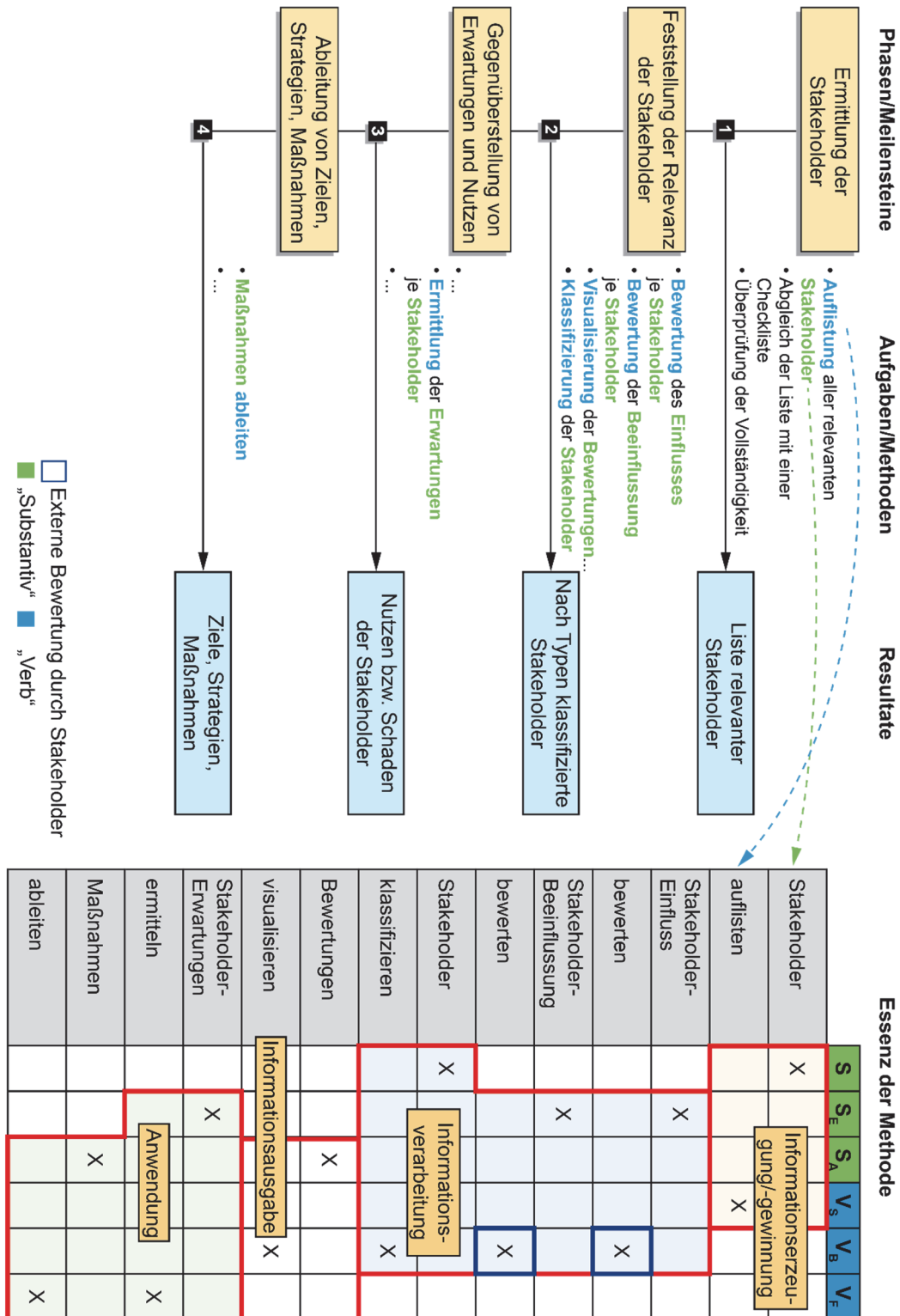


Bild 4-33 Essenz der Methode und Klassifikation am Beispiel der Stakeholderanalyse

Wie bei der Beschreibung der unterschiedlichen Verbarten, können ebenfalls unterschiedliche Arten von Substantiven festgestellt werden:

- Substantive (S), die im Kontext der Methode zentral sind. Bei der Stakeholderanalyse ist das natürlich der *Stakeholder*.

- Substantive (S_E), die eine Eigenschaft des übergeordneten Substantivs adressieren, bspw. *Stakeholder-Einfluss*. Zusammen mit den Verben des Bewertens sind das Merkmale – bzw. Attribute – des übergeordneten Substantivs. Sie müssen also im Bewertungsverfahren der Methode berücksichtigt werden und sollten je nach MBSE-Konzept auch im Systemmodell enthalten sein.
- Substantive (S_A), die eine weiterführende Aktion mit Bezug auf die Methode implizieren. Ein Beispiel ist das Wort *Maßnahme*.

Mit dieser Zerlegung können die Methoden des Technischen Managements analysiert und geklärt werden, wie die Methode zur Nutzung auf Basis des Systemmodells ausgestaltet sein muss. Es lassen sich die Elemente identifizieren, die im Systemmodell abgebildet werden müssen, im Rahmen der Methode eine Transformation erfahren oder weiterverwendet werden. Ebenfalls wird erkannt, welche Aktivitäten durch den Anwender durchgeführt werden müssen; für die Stakeholderanalyse sind das bspw. die Modellierung der Stakeholder und ihrer Attribute (vgl. Bild 4-33); für ihre Bewertung kann auf eine externe Anwendung gesetzt werden. Ebenso sollten die Ergebnisse der Methode im Systemmodell gesichert werden, da sie von fachdisziplinübergreifender Relevanz sind.

Sichtenplanung

Mit den Ergebnissen der technischen Methodenanalyse, des S-I-P-O-C aus Kapitel 4.6.1.2 und den Informationen der Steckbriefe aus Kapitel 4.6.2.1 kann die Erstellung des Systemmodells zur Unterstützung der ausgewählten Methode nun vorangetrieben werden und die Modellplanung abgeschlossen werden. Wie in Kapitel 4.4.3 herausgearbeitet, handelt es sich hierbei um die Erstellung der entsprechenden Sicht. Bei existierendem Systemmodell handelt es sich analog um die Extraktion der Sicht.

Hierzu wird das Rahmenwerk mit einem weiteren einfachen Hilfsmittel unterstützt: Ein Sichten-Template strukturiert die Umsetzung der Sicht. Das Template baut auf der Idee der Multi-Domain-Matrix (MDM) auf [LMB09]. Es wird vorgeschlagen, dass es die bislang unabhängig voneinander erarbeiteten methoden- und prozessspezifischen Zusammenhänge und modellspezifischen Zusammenhänge zusammenfasst. Im Prinzip geht es ausgehend von der Methoden-Essenz um die Identifikation und Definition relevanter Modellelemente, Attribute/Parameter und die Strukturierung des Modells in Abhängigkeit der Modellierungsmethode – und die Visualisierung ihrer Zusammenhänge.

Als Ergebnis liegt in Form des Sichten-Templates ein für das jeweilige Modellierungsvorhaben individuelles Vorgehensmodell vor, inkl. der wichtigsten prozessbezogenen Zusammenhänge. Bild 4-34 stellt das Template mit seinen sechs Matrizen und ihr Zusammenwirken als Schema dar. Die Leserichtung ist europäisch unidirektional, also von der Zeile ausgehend auf die Spalte. Die ersten drei Matrizen repräsentieren prozess- bzw. methodenspezifische Zusammenhänge, die hinteren Matrizen modellspezifische Zusammenhänge:

- Die Beziehungsmatrix ist Ausgangspunkt des Templates und wird aus der Methoden-Essenz gewonnen. Sie umfasst die Systemmodellinhalte und beantwortet die Frage: „*Wie stehen die Elemente zueinander in Beziehung?*“ Für die Stakeholder-Analyse sind das z.B. *Stakeholder*, *Stakeholder-Einfluss*, *Stakeholder-Beeinflussung*, *Bewertung*, idealerweise auch der *Kaffeeautomat*. Beziehungen und Abhängigkeiten der Elemente können so zwar nur grob, aber schnell und überschaubar geplant bzw. dokumentiert werden.
- Bei den Systemmodellinhalten ist die Unterscheidung von Methodeninputs und -outputs wichtig. Dies wird mittels S-I-P-O-C unterstützt. Inputs für die Methode sind zwingend zu modellieren, bei Outputs muss die Relevanz abgewogen werden. Alternativ wird das Ergebnis jedoch nicht gesichert. So sollte bspw. die *Bewertung* („Stakeholder-Klassifikation“) als relevant eingestuft werden. Sie hat fachdisziplinübergreifende Relevanz, da bspw. eine Priorisierung von Use Cases in Abhängigkeit der Bedeutung der Stakeholder durch den Systemanalysten stattfinden kann – wie auch das S-I-P-O-C zum Ausdruck bringt.
- Im Bereich Verantwortlichkeiten werden Zuständigkeiten bei der Modellierung dargestellt. Hierbei kann der gemäß S-I-P-O-C nominierte Stakeholder verantwortlich sein, oder ein Organisationskonzept aus Bild 4-30 umgesetzt werden.
- Modellierungsschritte beziehen sich auf das Vorgehen innerhalb der gewählten Modellierungsmethode. In ihrem Rahmen ist die Methode bzw. „Sicht“ organisiert. Für die Stakeholderanalyse ist das bei CONSENS bspw. der Schritt „Umfeldanalyse“, bei Sysmod dagegen „Identifikation der Stakeholder“.
- Die Modellelemente bilden im Einklang mit der ausgewählten Sprache und Methode die Systemmodellinhalte (datentechnisch) ab. Die Sprache definiert i.d.R. die nutzbaren Elemente. Jede Sprache hat ein Metamodell, das auf unterschiedlichen Wegen entwickelt werden kann: Es kann eine spezielle Sprache mit eigenem Metamodell entwickelt werden, in der MBSE-Community ist jedoch meist die Definition einer Sprache durch Stereotypen als leichtgewichtiger Ansatz möglich. Jedes Mal entsteht eine sog. domänenspezifische Sprache (DSML), die dann Besonderheiten des Anwendungsbereichs ausweisen. Bei den leichtgewichtigen Ansätzen handelt es sich um sog. Profile, die dann auch neue Modellelemente enthalten, so z.B. in CONSENS und Sysmod.
- Die Zuordnung zu einem Diagramm ist abhängig von der Modellierungsmethode. Idealerweise gibt es eine Vorgabe, welche Inhalte in welchem Diagramm zu modellieren sind. Bei CONSENS geschieht die Modellierung der Stakeholder in einem Umfeldmodell, was im korrespondierenden SysML-Profil in einem Blockdefinitionsdiagramm oder dem internen Blockdiagramm umgesetzt würde. Bei Sysmod dagegen werden die Stakeholder in einem Blockdefinitionsdiagramm und einem Use Case-Diagramm modelliert. Diese Zuordnung sollte im Sinne weiterer Modellierungsvorhaben also immer definiert sein.

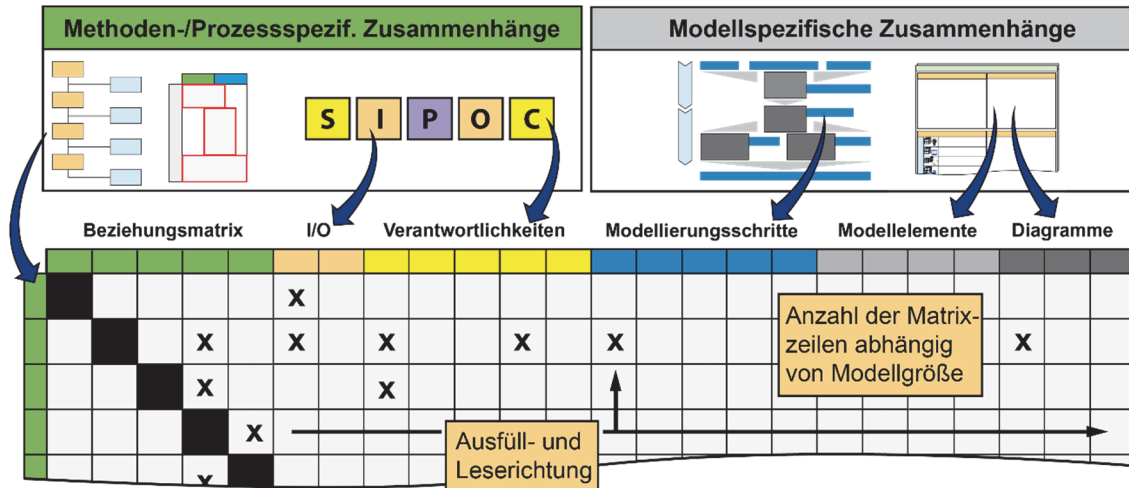


Bild 4-34 Schema des Sichten-Templates

Für die Methoden CONSENS und Sysmod sind die Ergebnisse in Bild 4-35 beispielhaft anhand der Stakeholderanalyse dargestellt. Die methoden- bzw. prozessspezifischen Zusammenhänge der Modellierungsmethoden sind aufgrund ihrer Allgemeingültigkeit identisch (oben); das Ergebnis der modellbezogenen Zusammenhänge ist aber anders, was aus der vorangegangenen Beschreibung deutlich wurde. Der modellspezifische Teil des Sichten-Templates ergibt sich somit aus einer strukturierten Aufbereitung der Modellierungsmethode für diese Matrixdarstellung – dadurch werden sämtliche Elemente der meist schwierig zu erfassenden Methoden übersichtlich dargestellt und können für die Planung wie eine Checkliste genutzt werden. Zudem ergibt sich der Vorteil bei einer häufigeren Anwendung einer Methode in unterschiedlichen Projekten, dass auf Basis dieser Darstellung Vorzugselemente vorgegeben werden können – hier also relevante Stakeholder für eine bestimmte Projektart.

Die Anwendung erfordert Erfahrung im MBSE und in der ausgewählten Modellierungsmethode; das Sichten-Template unterstützt die strukturierte Modellplanung und den Aufbau vergleichbarer Modelle. Auf diese Weise werden Prozess- und Modellaspekte gekoppelt und dienen als Anleitung während der Umsetzung – für alle Beteiligten sind die Aufgaben und Verantwortlichkeiten transparent dokumentiert. Hierauf aufbauend können die entsprechenden Systemmodelle erstellt werden, bzw. auch die Methode implementiert werden.

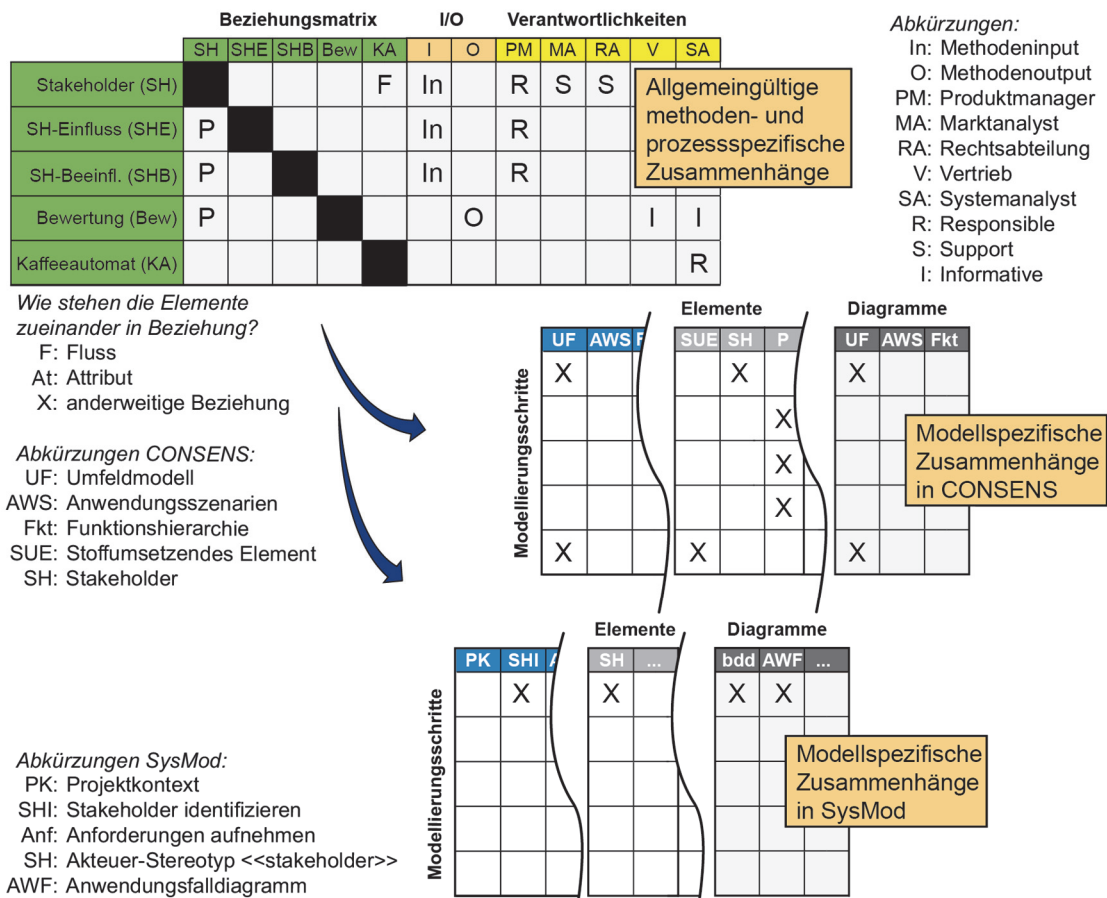


Bild 4-35 Auszug aus dem Sichten-Template – CONSENS (Mitte) und SysMod (unten)

4.6.2.3 Analyse-Phase – Informationsverarbeitung und -ausgabe

Die Aktivitäten Informationsverarbeitung und -ausgabe sind vorbereitend für die Methodenimplementierung. Sie sind die ersten Aktivitäten des Hauptaufgabenbereichs Methodenkopplung. Konkret müssen die Aufgaben aus Bild 2-33 durchgeführt werden: Detaillierung der IT-Infrastruktur und Gestaltung der Darstellungsform der Methode.

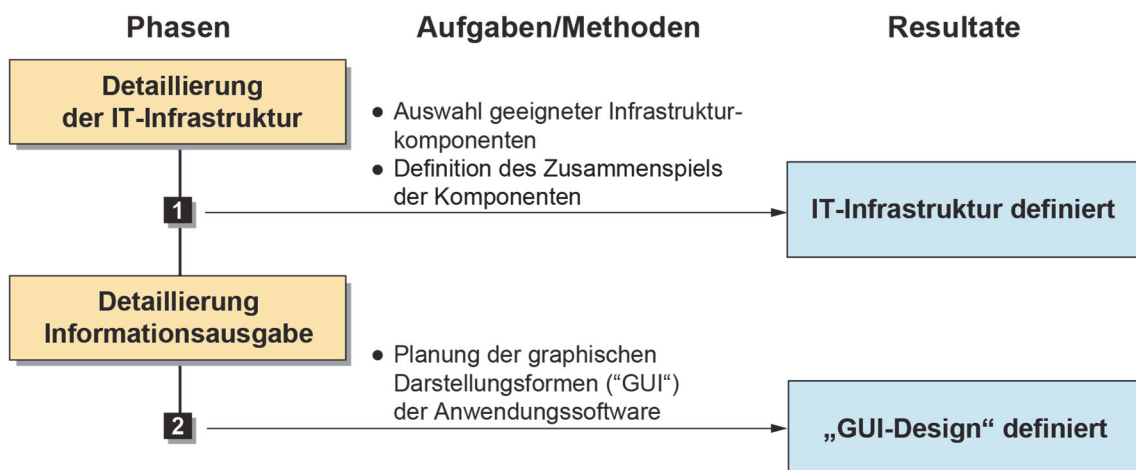


Bild 4-36 Aufgaben in der Analyse-Phase – Informationsverarbeitung und -ausgabe

Detaillierung der IT-Infrastruktur

Das Systemmodell und hierauf aufbauende Methoden sollen von mehreren Anwendern genutzt werden – die eventuell auch weiterhin ihre eigenen Anwendungswerkzeuge nutzen wollen; das föderierte Konzept unterstützt diese Zusammenarbeit auf geeignete Weise (vgl. Bild 4-24).

In den üblichen MBSE-Werkzeugen ist das Systemmodell im sog. Repository verfügbar. Um außerhalb des Werkzeugs auf die Inhalte des Systemmodells zugreifen zu können, bedarf es zusätzlich einer Serverstruktur und der entsprechenden Anwendungssoftware. Bei letzterer bietet sich vor dem Hintergrund der verteilten Arbeit eine web- bzw. browserbasierte Anwendung an, da so orts- und plattformunabhängig mit einem Web-Browser auf Datenbanken zugegriffen werden kann. Bei der Detaillierung der IT-Infrastruktur müssen also konkret die folgenden Fragen beantwortet werden:

- Wie sollen die Informationen aus dem Systemmodell extrahiert werden?
- Wie werden die Informationen gespeichert?
- Wie sollen die Informationen weiterverarbeitet werden?
- Wie sollen die Ergebnisse der Methode ausgegeben werden?

Je nach Rahmenbedingungen fällt die Beantwortung dieser Fragen unterschiedlich aus. Für die geplante Anwendung wird eine möglichst unkomplizierte Infrastruktur beispielhaft definiert – der Fokus liegt auf dem Vorgehen selbst und nicht auf der Einrichtung einer Infrastruktur. Als Werkzeug wird das Produkt Enterprise Architect genutzt, da es in vorangegangenen Forschungsprojekten bereits angewendet wurde [IKD+13], [IGB+15]. Zudem existieren für das Werkzeug SysML-Profile der hier betrachteten Methoden CONSENS und Sysmod. Der Zugriff auf das Systemmodell kann beim ausgewählten Werkzeug beispielsweise über die folgenden vier Zugriffsarten gelingen:

- XMI-Import und –Export sowie auf XMI basierende Modelltransformationen
- API
- EA-UML-Bridge als Plugin zur Verknüpfung mit EMF
- SQL (Structured Query Language) zum Direktzugriff auf die Datenbank

Durch diese Schnittstellen kann die Integration des Werkzeugs in eine Infrastruktur gelingen, die die Methodenanwendung und die Zusammenarbeit im Rahmen des MBSE auf einfache Weise demonstriert. Bild 4-37 skizziert schematisch die im Rahmen dieser Arbeit für das MBSE-Konzept 3 konzipierte Infrastruktur und das Zusammenspiel der einzelnen Elemente: Kern ist ein MySQL-Server, über den die Daten zwischen Modellierungswerkzeug und Methodenanwendung ausgetauscht werden. Die PHP-Anwendung kommuniziert mit dem SQL-Server, der das Systemmodell aus Enterprise Architect abbildet und dient zusätzlich als Schnittstelle mit dem Stakeholder – sowohl

für die methodenspezifische Informationseingabe (Informationen, die nicht aus dem Systemmodell stammen, bspw. Bewertungsinformationen), als auch für die Informationsausgabe. Das bietet für die Methodenanwender entsprechend einfache Zugänge.

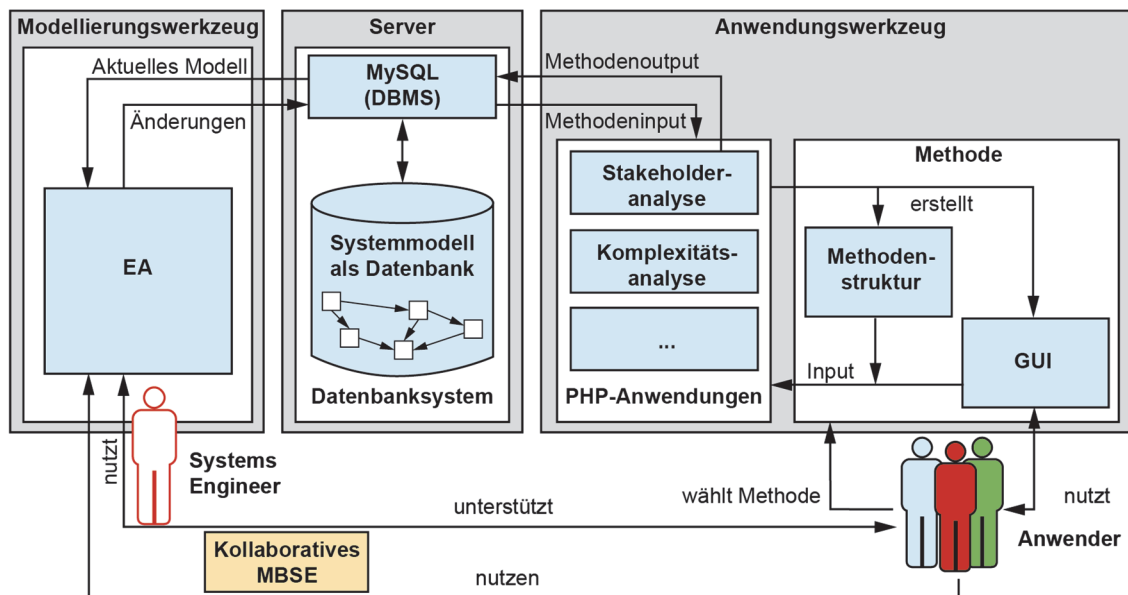


Bild 4-37 Beispielhaft implementiertes Infrastrukturkonzept

GUI-Design

Als weitere Aufgabe dieser Phase steht die Definition einer geeigneten Visualisierungsform der Methode an. In Bild 4-38 sind einige typische Visualisierungsformen von Methodenergebnissen zusammengetragen. Für die adressierte Stakeholderanalyse ist ein Portfolio geeignet. Dieses sollte mit geeigneten grafischen Methoden geplant werden, hierbei ist insb. auch auf eine geeignete Skalierung der Achsen zu achten. Durch die Verwendung von PHP zur Umsetzung der Methode ist hier eine einfache Möglichkeit zur Manipulation von Grafiken gegeben, was das Erstellen von Portfolios vereinfacht.

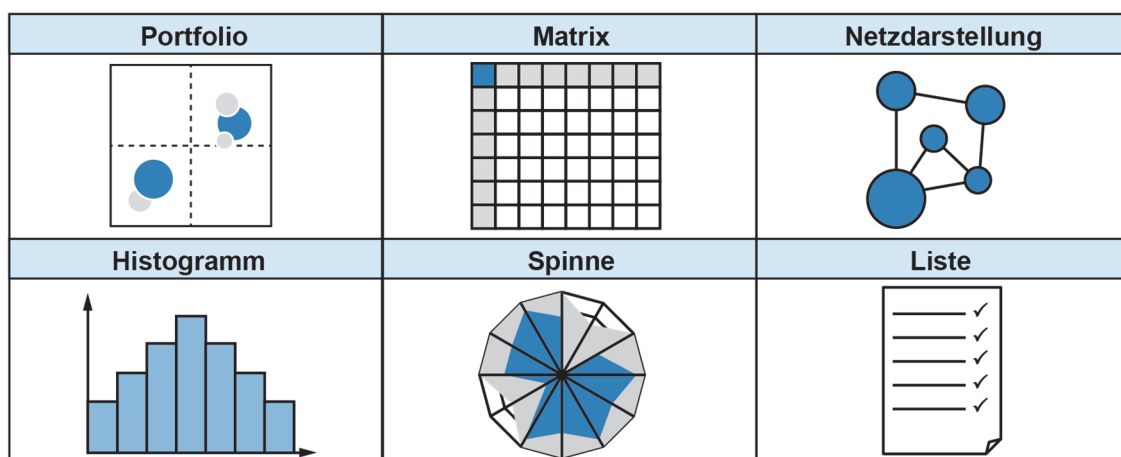


Bild 4-38 Typische Arten zur Visualisierung von Methodenergebnissen

4.6.3 Informationserzeugung – Design und Control

Hier steht die Modellierung der relevanten Modellinhalte an – die Sicht. Dabei kommen die Spezifika der Modellierungsmethode zum Tragen. Zur Durchführung der Stakeholderanalyse wurde mit dem Sichten-Template gezeigt, dass im Umfeldmodell alle Inhalte abgebildet werden können. Im Prinzip reichen in Anlehnung an KAISER [Kai13] die Umfeldelemente „Nicht-Technisches Element“ aus, zzgl. einer logischen Beziehung zum Systemelement Kaffeeautomat. Zur Modellierung mit dem Konzept Mechatronic Sketch wurden die existierenden CONSENS-Visio Shapes so erweitert, dass das Profil nach KAISER umsetzbar ist [TBD+15]. Zusätzlich wurde der Elementtyp <<Stakeholder>> ergänzt. Dadurch ergibt sich für das MBSE-Konzept 2 „Mechatronic Sketch“ die Möglichkeit, die Methode auf Basis des Systemmodells durchzuführen.

In Bild 4-39 ist dargestellt, wie einem Umfeldelement der Typ <<Stakeholder>> über das Auswahlmenü der Visio-Shapedaten zugewiesen wird. Im Verlauf der Modellierung wird somit für jedes Umfeld- oder Systemelement die Elementart definiert und die Beziehungen an den Shape-Konnektoren angebracht. Auf diese Weise werden über einen Shape-Bericht die Informationen dieses Visio-Modells ausgelesen. (vgl. Kapitel 4.6.4).

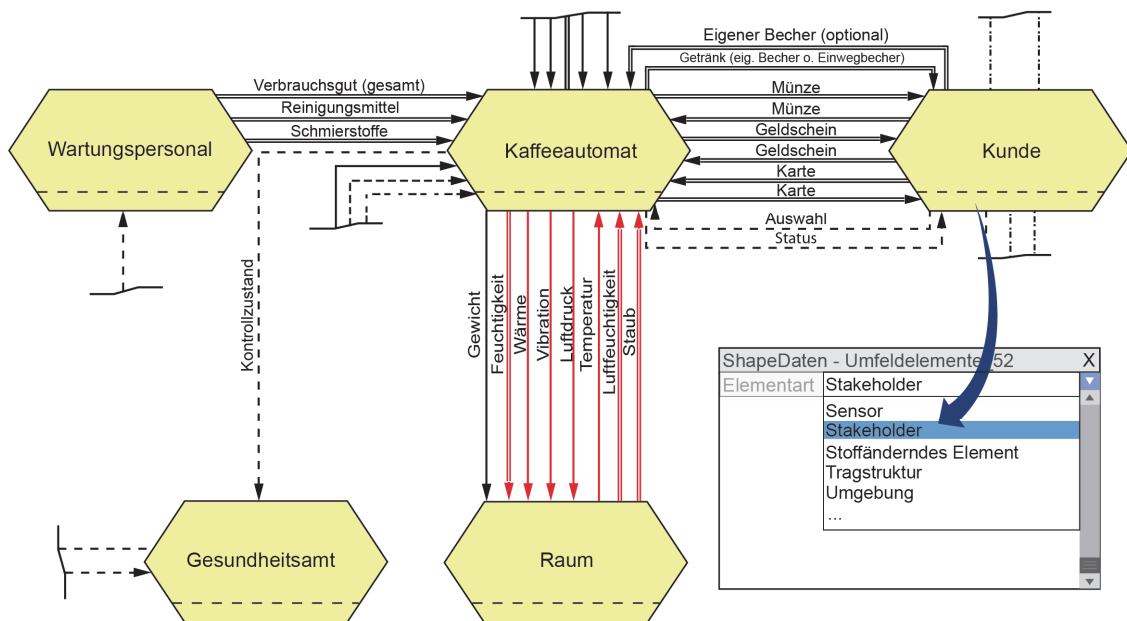


Bild 4-39 CONSENS-Profil im MBSE-Konzept 2 Mechatronic Sketch, inkl. umgesetztem und erweitertem Profil nach KAISER [Kai14]

Je nach Projektfortschritt sieht das Umfeldmodell unterschiedlich aus, wobei die o.g. Darstellungen ausreichen – Bild 4-40 (vorne) zeigt diese resultierende Stakeholder-Sicht. Bei der Modellierung mit Diagramming-Tools sind jedoch mehr Informationen im Umfeldmodell üblich als notwendig, da die Trennung von Modell und Sicht aufgrund des fehlenden Datenmodells nicht angewendet werden kann (Bild 4-40 – hinten). Durch die Elementklassifikation kann aber trotzdem die Analyse durchgeführt werden.

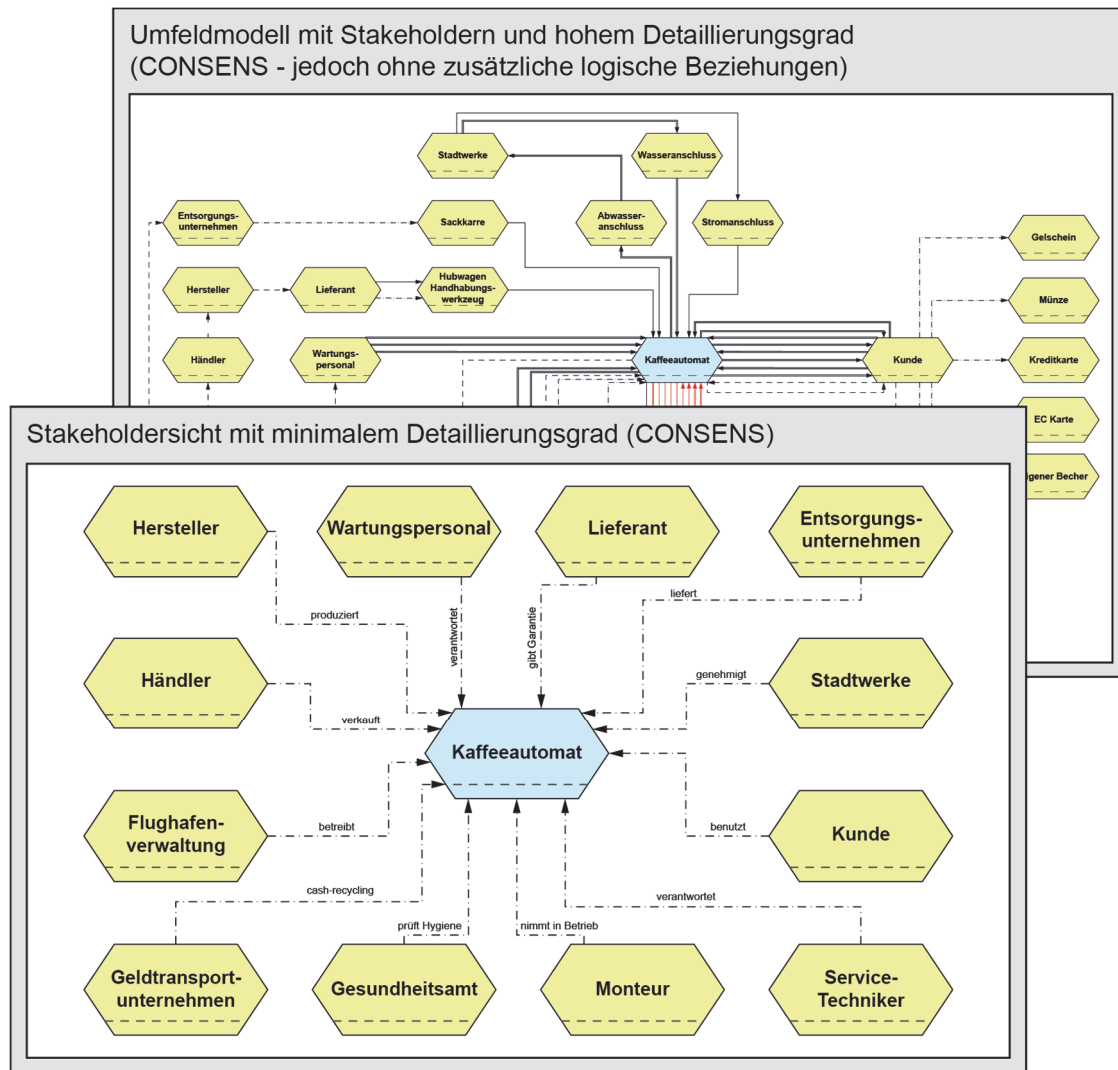


Bild 4-40 Stakeholder-Sicht im Umfeldmodell mit minimalen Informationen (vorne), detailliertes Umfeldmodell (hinten) – MBSE-Konzept 2

Im Konzept 3 Mechatronische Systemmodellierung sieht die Modellierung wie in Anhang A5.1 dargestellt aus: Es wurde das Profil SysML4CONSENS für das Werkzeug Enterprise Architect genutzt. Stakeholder konnten als Stereotyp Umfeldelement angelegt werden. Die weiteren Informationen wurden über Attribute realisiert, da das Profil keine weitere Unterscheidung der Umfeldelemente anbietet. Damit ist der Stakeholder ebenso ein Attribut, wie die Ergebnisse der Stakeholderanalyse „Beeinflussbarkeit“ und „Einfluss“ – um die Resultate der Analyse speichern zu können. Das tatsächliche Ergebnis der Methode – die Art des Stakeholders – wird über das Attribut Stakeholder gespeichert. Bild 4-41 zeigt einige der im Blockdefinitionsdiagramm „bdd“ angelegten Blöcke des Umfeldmodells. Die Blöcke zeigen die notwendigen Ports und die Attribute des Blocks. So hat z.B. der Block/das Umfeldelement *Kunde* aktuell elf Flüsse angelegt, als Attribute sind angelegt *Beeinflussbarkeit*, *Einfluss* und *Stakeholder*.

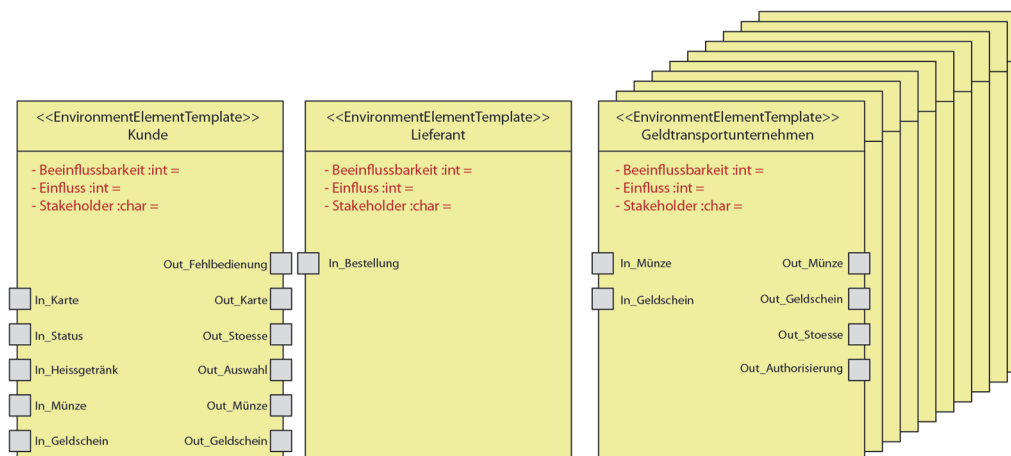


Bild 4-41 Elemente des Blockdefinitionsdiagramms der Kaffeemaschine

4.6.4 Design-Phase

In der Aktivität Informationserzeugung der Design-Phase wurde das Systemmodell erstellt, in den übrigen Aktivitäten der Phase steht die Umsetzung der IT-Infrastruktur an. Ergebnis ist eine funktionsfähige IT-Infrastruktur. Damit wird der Hauptaufgabenbereich der Methodenkopplung technisch realisiert und abgeschlossen.

Für das MBSE-Konzept 2 „Mechatronic Sketch“ wurde in [TBD+15] ein mögliches IT-Konzept vorgestellt. Dieses verbindet ein Diagramming-Tool mit weiteren Werkzeugen aus der Office-Welt (Bild 4-42): Auf Basis der Shapes und der dazugehörigen Shape-Daten können über die internen Möglichkeiten des Werkzeuges („Visio Shape Report“) die Inhalte ausgelesen werden. Jedes Systemelement hat dabei unterschiedliche Zusatzinformationen im Shape gespeichert, u.a. den Namen des Systemelements, eine Shape-ID und die gewählte Elementart. Hiermit können bspw. über Makros und Tabellenkalkulationen die modellierten Inhalte in eine Stakeholder-Analyse überführt werden.

Für die in Kapitel 4.6.2.3 geplante IT-Infrastruktur wurde dagegen ein aufwändigeres Konzept prototypisch umgesetzt, das sowohl für die MBSE-Methoden CONSENS als auch für Sysmod genutzt werden kann. Es baut auf dem MBSE-Konzept 3 auf und ist in gewissem Rahmen auch nachhaltig in Projekten nutzbar. Konkret handelt es sich um eine Serverlösung inkl. installierter Modellierungs- und Anwendungssoftware. Die Anwendungssoftware entspricht hier der Umsetzung der Methodenfunktionen in Programmcode.¹⁹ Als Ergebnis können aufbauend auf der Systemmodellierung in dem Werkzeug Enterprise Architect über eine Schnittstelle die modellierten Inhalte in eine Datenbank ausgelesen werden. Konkret werden aus dem Systemmodell alle Elemente mit dem Attribut „Stakeholder“ extrahiert. Ebenso können die Ergebnisse aus der Methodenanwendung (z.B. hier die Bewertung der Stakeholder) über die Datenbank in das

¹⁹ Vgl. Anhang A7 zur gewählten IT-Infrastruktur. Auf die Umsetzung der Methodenfunktion in Programmcode wird hier nicht näher eingegangen, vgl. hierzu [GTH16].

Modell geschrieben werden – im Gegensatz zur Kopplung über das Werkzeug in MBSE-Konzept 2. Das Ergebnis wird dem Attribut Stakeholder zugeschrieben.

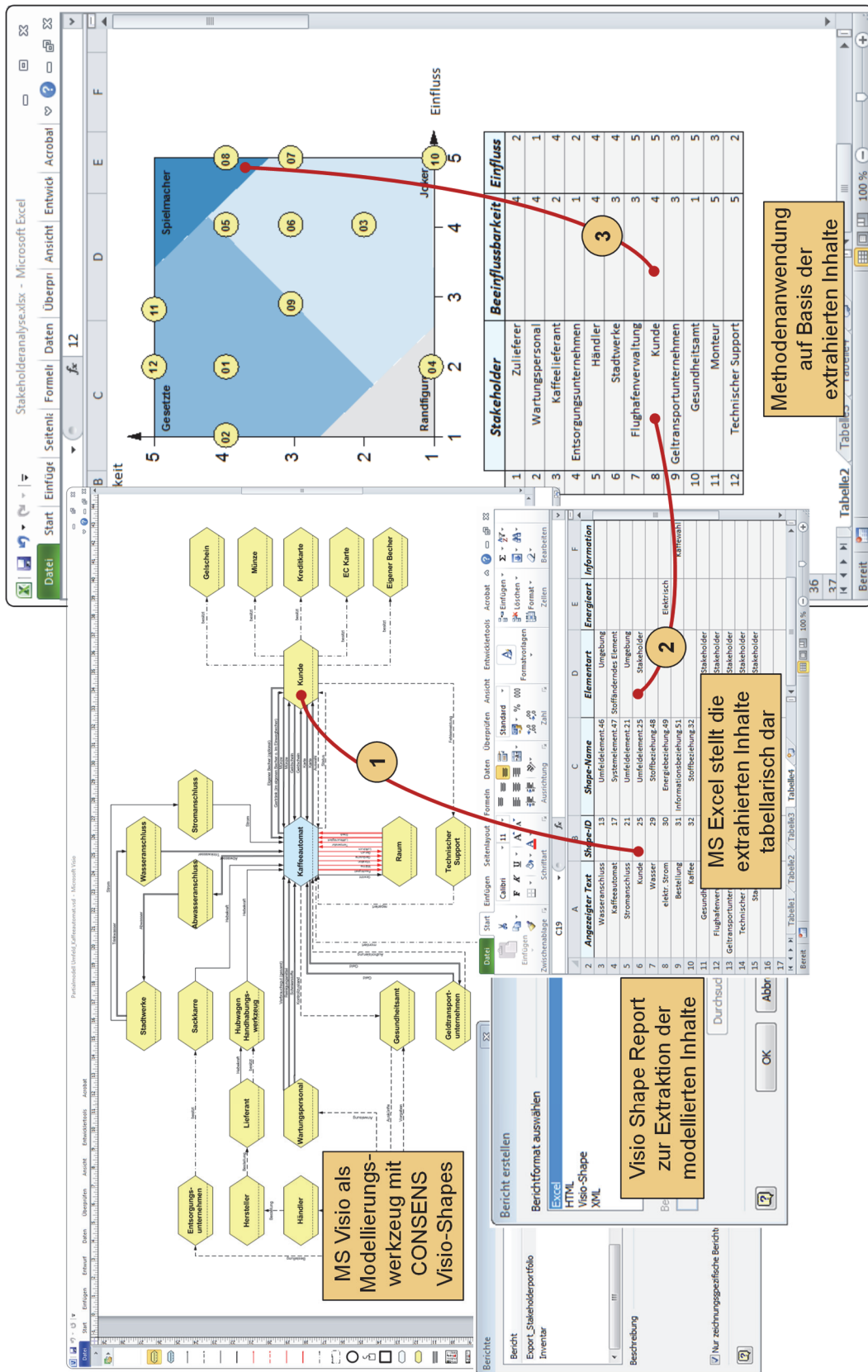


Bild 4-42 Stakeholderanalyse mit MBSE-Konzept 2, in Anlehnung an [TBD+15]

Die Stakeholderanalyse wurde dabei als php-Anwendung implementiert. Sie verbindet sich mit der Datenbank und sucht alle Umfeldelemente mit dem Attribut „Stakeholder“ aus dem Systemmodell. Bild 4-43 zeigt einen Mock-Up dieser prototypisch umgesetzten IT-Infrastruktur. So können hier durch die Anwender die unterschiedlichen Analysen ausgewählt werden und die weiterführenden Methodenschritte durchgeführt werden. Für die abgebildete Stakeholderanalyse können bspw. je Stakeholder die „Beeinflussbarkeit“ sowie der „Einfluss“ per Schieberegler mit einem Wert von 1 bis 5 bewertet werden – 1 bedeutet gering und 5 sehr hoch.

Die Ergebnisse werden entsprechend in einer Bewertungsmaske dargestellt. Ebenso können die Verbindungen zu den Datenbanken und dem Systemmodell aus dieser Darstellung direkt aufgerufen werden.

Durch die Positionierung der einzelnen Elemente innerhalb des Stakeholder-Relevanz-Portfolios können nun Maßnahmen für die weitere Bearbeitung abgeleitet werden. So ist der Kunde bspw. ein Joker – er sollte gemäß Normstrategie stärker an das Unternehmen gebunden werden, um ihn besser beeinflussen zu können. Aus Sicht der Modellierung kann nun der Systemanalyst detaillierte Use Cases erstellen – hierzu sollte ggf. der Kunde in weitere Untergruppen unterteilt werden.

Weitere Informationen über den Prozess und das Projekt komplettieren die Darstellung und koppeln prozessbezogene und modellbezogene Zusammenhänge²⁰. Hervorzuheben ist, dass bei diesem Ansatz die Ergebnisse der Stakeholderanalyse auch entsprechend in das Systemmodell zurückgespielt werden können – als Attribute eines Stakeholders.

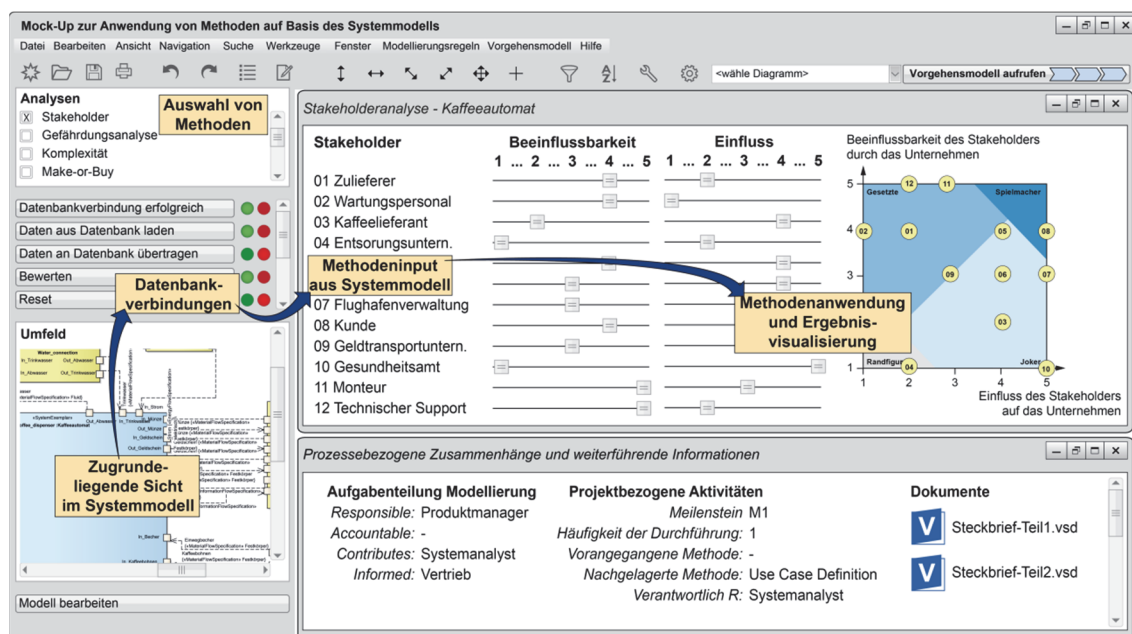


Bild 4-43 Mock-Up der prototypischen IT-Infrastruktur, in Anlehnung an [GTH16]

²⁰ Ein Snapshot der prototypischen Umsetzung ist in Anhang A6.2.1 dargestellt.

4.6.5 Control-Phase

Im Six Sigma geht es in der Control-Phase um die Verstetigung der Ergebnisse, d.h. die fortlaufende Anwendung, die Schulung der Mitarbeiter und den Projektabschluss. Ein Projektabschlussbericht wird an den Six Sigma Champion übergeben.

Die hier betrachteten Aktivitäten sind ähnlich ausgelegt: Zunächst geht es um die Anwendung der ausgewählten Methode auf Basis des Systemmodells, wobei die Anwendung der Stakeholderanalyse hier bereits im Kontext der Design-Phase beschrieben wurde. Zudem steht die erfolgreiche Einbindung der Änderungen in den Produktentstehungsprozess und die Schulung der zukünftigen Nutzer an. Bei der hier vorgestellten Modellierung zur Unterstützung einer konkreten Methode handelt es sich zwar zunächst um eine sehr spezielle Nutzung des Systemmodells und nur einen kleinen Ausschnitt der mit MBSE verbundenen Möglichkeiten. Durch diese schrittweise Integration der Modellierung und Nutzung der Modelle in den Produktentstehungsprozess und mit dem in Bild 4-17 vorgestellten Vorgehen inkl. der dazugehörigen vollständigen Dokumentation kann ein adäquater Veränderungsprozess sichergestellt werden. Nach Abschluss dieser Phase ist ein adäquater Einsatz der Methode auf Basis des Systemmodells im Arbeitsalltag gesichert.

5 Anwendung und Bewertung des Rahmenwerks

In Kapitel 5 wird zunächst eine Analyse verschiedener MBSE-Projekte durchgeführt. Das dient dem Zweck, einen Überblick über die aktuelle Verbreitung der methodisch hergeleiteten MBSE-Konzepte zu erlangen. Gleichzeitig soll dadurch auch die in Kapitel 2.6 aufgestellte These mehrerer existierender MBSE-Konzepte empirisch bestätigt werden. An die Analyse schließen sich ergänzend zu der Anwendung des flexiblen Vorgehensmodells in Kapitel 4 weiterführende Anwendungen an. Konkret sind das die Durchführung einer Gefährdungsanalyse in Anlehnung an NOHL [NTT88] und eine Komplexitätsanalyse in Anlehnung an MARTI [Mar07]. Damit wird einerseits gezeigt, dass weitere Methoden aus den technischen Managementprozessen auf Basis des Systemmodells anwendbar sind und nicht nur die aktuell im Mittelpunkt stehenden Methoden der Technischen Prozesse [ISO15288]. Andererseits wird so die Übertragbarkeit des Rahmenwerks auf neue Aufgabenstellungen dargestellt.

5.1 Konzepte des MBSE

In Kapitel 4.2 wurden vier verschiedene, konsistente MBSE-Konzepte hergeleitet. Diese sind als Prototypen des MBSE zu verstehen: Anwendungen sind typische Vertreter eines Konzepts, wenn sie eine gewisse Ähnlichkeit mit ihm aufweisen. Das ist dann der Fall, wenn ein gewisser Anteil der Charakteristika des betrachteten Projekts mit einer konsistenten Kombination der identifizierten Merkmalausprägungen übereinstimmt. In Kapitel 5.1.1 wird zunächst ein Überblick über die betrachteten Untersuchungseinheiten (UE) gegeben, gefolgt von ihrer Klassifizierung in Kapitel 5.1.2.

5.1.1 Überblick über die Untersuchungseinheiten

Insgesamt werden zur Klassifizierung 35 Untersuchungseinheiten herangezogen. Diese stammen aus selbst durchgeführten oder begleiteten Industrieprojekten aus dem Umfeld von Fraunhofer IEM²¹, verfügbaren wissenschaftlichen und industrienahen Veröffentlichungen sowie Erfahrungen und Erkenntnissen über weitere Projekte aus der Industrie, bspw. auf Basis von Expertengesprächen oder aus Arbeitskreisen unterschiedlicher Organisationen. Insgesamt ergeben sich 11 wissenschaftliche Untersuchungseinheiten und 24 Untersuchungseinheiten aus der Industrie. Jede der identifizierten Untersuchungseinheiten wird – wo möglich und sinnvoll – anhand der in Kapitel 4.2.1 extrahierten Merkmalausprägungen beurteilt und einem der MBSE-Konzepte zugeordnet. Tabelle 5-1 gibt einen Überblick über die Untersuchungseinheiten. Neben der Unterscheidung

²¹ Das sind einerseits reine Industriebeauftragungen, aber auch Projekte aus dem BMBF-geförderten Spitzencluster it's OWL – Intelligente Technische Systeme OstWestfalenLippe. Diese Projekte können sowohl Innovationsprojekte, als auch sog. Transferprojekte sein.

der Herkunft eines Projekts (Forschung vs. Industrie) wird zudem noch die Branche genannt, in der das Projekt durchgeführt wurde. Ebenso wird der adressierte Systemlevel genannt, wobei die in Kapitel 4 herausgearbeitete Abstufung maßgeblich ist. Aus Gründen der Übersicht wurde bereits die Zuordnung zu einem MBSE-Konzept in der Tabelle vorgenommen. Wo möglich wurden Referenzen genannt, wenngleich eine Großzahl der Untersuchungseinheiten aufgrund des Datenschutzes anonymisiert ist.

Tabelle 5-1 Überblick über die Untersuchungseinheiten

Nr.	Projektname	Projektart	Branche	Systemlevel	MBSE-Konzept	Referenz
1	INVIRTES	F	Luft- und Raumfahrt ²⁾	System	K3	[RSR+15]
2	MecPro2	F	Automobilbau	unbekannt	K3	[MCS+14]
3	SFB614	F	Maschinenbau, Fahrzeugtechnik	verschieden	K2	[GRS14]
4	Hehenberger	F	diverse	verschieden	K4	[Heh12]
5	Hochwallner	F	diverse	System	K3	[HHD+12]
6	SNL - Maestro	F	Embedded Systems	Modul	K4	[BSD+12]
7	VAMF	F	Automobilbau	Modul	K4	[BPC+13]
8	Micro Laser	F	Semiconductor	System	K4	[QDW09]
9	A3AO	F	diverse	verschieden	K1	[Bor10]
10	Zingel	F	Automobilbau	verschieden	K3	[Zin13]
11	Ramos	F	Personentransport	System	K3	[Ram09]
12	Audi	I	Automobilbau	Modul	K2	[SKD+13]
13	AVL	I	Automobilbau	System	K3	[SSM+12]
14	Daimler I	I	Automobilbau	Modul	K2	[Koe13]
15	Daimler II	I	Automobilbau	unbekannt	K3	[Haa16]
16	Hella	I	Automobilbau	Modul	K3	[WM14]
17	anonym	I	Informationstechnik ¹⁾	System	K2	*
18	anonym	I	Informationstechnik ¹⁾	System	K2	*
19	anonym	I	Informationstechnik ¹⁾	System	K3	*
20	anonym	I	Automobilbau ¹⁾	Modul	K1	*
21	anonym	I	Automobilbau ¹⁾	Modul	K2	*
22	anonym	I	Maschinen- und Anlagenbau ²⁾	System	K2	*
23	anonym	I	Maschinen- und Anlagenbau ²⁾	System	K2	*
24	anonym	I	Maschinen- und Anlagenbau ²⁾	System	K2	*
25	anonym	I	Maschinen- und Anlagenbau ²⁾	System	K2	*
26	anonym	I	Maschinen- und Anlagenbau ²⁾	System	K1	*
27	anonym	I	Maschinen- und Anlagenbau ²⁾	System	K2	*
28	anonym	I	Maschinen- und Anlagenbau ²⁾	System	K2	*
29	anonym	I	Lichttechnik ²⁾	S-o-S	K1	*
30	SAAB	I	Luft- und Raumfahrt	System	K3	[AHJ+09]
31	Kannegiesser A	I	Maschinen- und Anlagenbau ¹⁾	System	K1	-
32	Kannegiesser B	I	Maschinen- und Anlagenbau ¹⁾	System	K2	-
33	Karl Mayer	I	Maschinen- und Anlagenbau	System	K3	[KB14]
34	Miele	I	Elektrogeräte ¹⁾	System	K3	-
35	TKMS	I	Schiffbau	Modul	K3	[Hop15]

F - Forschung
I - Industrie

* aufgrund von Geheimhaltungsvereinbarungen nicht veröffentlichungsfähig
¹⁾ selbst durchgeführtes Projekt ²⁾ durch Fraunhofer IEM begleitetes Projekt

5.1.2 Klassifikation der Untersuchungseinheiten

Die 35 Untersuchungseinheiten wurden anhand der identifizierten MBSE- untersucht und entsprechend charakterisiert. Gemäß der gewählten Klassifikationsstrategie entscheidet eine gewisse Ähnlichkeit über die Zugehörigkeit zu einem MBSE-Konzept. Deshalb reicht es aus, wenn die Projekte bei einer Teilmenge der Merkmalausprägungen Übereinstimmungen mit einem idealisierten MBSE-Konzept aufweisen. Von Vorteil ist das insb. bei Übereinstimmung von eindeutigen oder dominanten Merkmalausprägungen. Geeignete Merkmale wurden identifiziert und retrospektiv auf Basis des Wissens über das Projekt bewertet. Bei der Bewertung kommt es darauf an, dass die Bewertungssumme je Merkmal 100% ergibt; zur Vereinfachung wurde nur eine grobe Unterteilung vorgenommen, die jedoch für die Darstellung der Ergebnisse ausreicht.

Die Beurteilung der Projekte wurde wie beschrieben vorgenommen und zunächst in einem Steckbrief in Kurzform dokumentiert. Das umfasst eine Kurzbeschreibung der Untersuchungseinheiten in Prosaform und die Darstellung der relevanten Merkmalausprägungen der Untersuchungseinheit. Die im Projekt identifizierten Merkmalausprägungen wurden prozentual bewertet, die Ausprägung des später zugeordneten MBSE-Konzepts zusätzlich analog zu Bild 4-7 farblich hervorgehoben. In den Bildern 5-1 bis Bild 5-7 sind einige Untersuchungseinheiten detailliert dargestellt. Die weiteren Untersuchungseinheiten sind – sofern keine öffentlich zugängliche Literatur existiert – stichpunktartig im Anhang dargestellt.

Untersuchungseinheit 12

Steckbrief – MBSE-Konzepte								
UE	Projektname	Projektart	Branche	Systemlevel	Referenz			
12	Audi	Industrie	Automobilbau	Modul	[SKD+13]			
Kurzbeschreibung			Klassifikation in Anlehnung an Kapitel 4.2					
<p>Das Unternehmen verfügt über ein pragmatisches Vorgehen, um die Systemanforderungen, die Systemchnittstellen und die Funktionsstrukturen („Wirkketten“) von einer High-Level-Sicht hin bis zu quantitativen Merkmalen herzuleiten. Diese sollen in Lastenheften oder für die Test- und Absicherungsspezifikationen genutzt werden können.</p> <p>Es wurde der Ansatz der Wirkkettenanalyse aufgegriffen und erweitert, s.d. über ein Zusammenspiel mehrerer Office-Werkzeuge und des Diagramming-Werkzeugs MS Visio die mechatronische Architektur im Sinne einer Mechatronic Sketch dargestellt werden kann. Für die Erstellung unterstützen einzelne ausgebildete Ingenieure laufende Projekte, wobei auch mit Zulieferern zusammengearbeitet wird.</p>			Formalis.grad	informal	semi-formal (100%)	formal		
			Visual Syntax	keine	Form/Farbe	Form und Farbe (100%)		
			Aspekte	Anforderungen (15%)	Struktur (80%)	Gestalt	Verhalten	Parameter (5%)
			Phasenablauf	formlos		formal (100%)		
			Phasendefin.	Top-Down (100%)		Bottom-Up	Zigzagging	
			Werkzeugart	Brown Paper	Diagr. Tool (100%)	COTS	Simulation	
			Datenmodell	ja		nein (100%)		
			Reichweite	projektintern	Unternehmen (70%)	Wertsch.kette (30%)		
			System-of-Int.	SoS	System	Modul (100%)		
			Aufwand	gering	mittel	hoch (100%)		
			Verantwortung	Workshop	Architekt (100%)	Fachspezialist		
			Anw.Bereich	Management	Technik (80%)	Enabler (20%)		
MBSE-Konzept	K2 – Mechatronic Sketch							

Bild 5-1 Untersuchungseinheit 12 – Audi, [SKD+13]

Die Untersuchungseinheit zeigt einen Ansatz mit dem Zweck einer verbesserten Kommunikation und Koordination der an einem Projekt beteiligten Parteien. Besonders auffällig ist, dass es sich um ein sehr großes Unternehmen handelt, das auf einfache Vorgehensweisen ohne Spezialsoftware setzt und dabei auch nicht die SysML als Sprache zugrunde legt, sondern einen eigenen Sprachansatz entwickelt hat. Der Ansatz scheint in besonderer Weise geeignet, die Grundzüge des Systems Engineering (vgl. Kapitel 2) in einer Organisation langsam, aber nachhaltig zu verankern.

Untersuchungseinheit 15

Ausgehend von einer Forschungsarbeit des Unternehmens (UE 14) mit inhaltlich weitgehend identischer Orientierung wie UE 15 ist zu beobachten, dass die Einführung des MBSE nun für die gesamte Organisation angegangen wird. Es werden zunächst der Bedarf, die Grundlagen und der Nutzen des MBSE analysiert. Die weitgehend öffentlich verfügbaren Informationen über die Projekte (z.B. [Koe13], [Haa16]) führen zu der Interpretation, dass es sich hierbei um eine Weiterentwicklung der Idee des MBSE über die Zeit handelt. Im Prinzip wird ausgehend von der Speziallösung eines Fachbereichs nun ein Rollout in große Teile des Unternehmens angestrebt. Ob hier tatsächlich ein Kausalzusammenhang besteht, kann nicht vollständig nachvollzogen werden, würde aber grundsätzlich einem typischen Lernprozess entsprechen. Somit ergibt sich die in Bild 5-6 durch den Pfeil dargestellte Entwicklung von der UE14 hin zu UE15.

Steckbrief – MBSE-Konzepte								
UE	Projektname	Projektart	Branche	Systemlevel	Referenz			
15	Daimler II	Industrie	Automobilbau	unbekannt	[Haa16]			
Kurzbeschreibung			Klassifikation in Anlehnung an Kapitel 4.2					
<p>Das Unternehmen stellt die Einführung des modellbasierten Systems Engineerings dar und informiert bspw. im Rahmen von Lieferantenveranstaltungen. Es verfolgt die Idee der durchgängig virtualisierten Produktentstehung.</p> <p>Die Virtualisierung beginnt bei den Anforderungen und wird über eine Systemmodellierung mit dem Ansatz SysML weitergeführt. Letztere steht hier bei der Betrachtung im Mittelpunkt. Im Prinzip kann das Programm als die logische Fortsetzung der Untersuchungseinheit 14 verstanden werden, wenngleich nicht bekannt ist, ob hier ein Zusammenhang besteht. Dennoch: Die Beispiele zeigen, dass sich beim Unternehmen über die Zeit eine Weiterentwicklung und Detaillierung der Ansätze ergeben hat, von K2 in Richtung K3.</p>			Formalis.grad	informal	semi-formal (80%)	formal (20%)		
			Visual Syntax	keine	Form/Farbe	Form und Farbe (100%)		
			Aspekte	Anforderungen (10%)	Struktur (30%)	Gestalt	Verhalten (30%)	Parameter (30%)
			Phasenablauf	formlos		formal (100%)		
			Phasendefin.	Top-Down (70%)		Bottom-Up	Zigzagging (30%)	
			Werkzeugart	Brown Paper	Diagr. Tool	COTS (100%)	Simulation	
			Datenmodell	ja (100%)		nein		
			Reichweite	projektintern	Unternehmen (100%)	Wertschöpfungskette		
			Projektart	Vorentwicklung	Kundenspez. Entw.	Serienentw. (100%)		
			Aufwand	gering	mittel	hoch (100%)		
			Verantwortung	Workshop	Architekt (100%)	Fachspezialist		
			Anw.Bereich	Management	Technik (80%)	Enabler (20%)		
MBSE-Konzept	K3 – Mechatronische Systemmodellierung							

Bild 5-2 Untersuchungseinheit 15 – Daimler, [Haa16]

Untersuchungseinheit 31

Das Innovationsprojekt *ReSerW* des BMBF-geförderten Spitzenclusters it's OWL – Intelligente Technische Systeme OstWestfalenLippe lässt sich hinsichtlich der Anwen-

derung von MBSE in zwei Phasen unterteilen: Zunächst stand die Analyse der Aufgabe an. Hierbei wurden Elemente der Spezifikationstechnik CONSENS genutzt und insb. auf die Partialmodelle Umfeld, Anwendungsszenarien, Funktionen und Anforderungen fokussiert. Ergänzt wurde das Vorgehen durch Beobachtungen der realen Prozesse und einfache Versuche mit Wäschestücken. Erkenntnisse wurden informal dokumentiert. Über das Umfeldmodell wurde zunächst die Systemgrenze des Roboters im Wäschereiprozess beschrieben und Anforderungen zur Integration des Roboters in eine bestehende Produktionslinie aufgenommen. Die Funktionen wurden u.a. aus der Beobachtung des realen Prozesses abgeleitet und hierfür auch Anwendungsszenarien beschrieben. Insb. die Bedeutung des Partialmodells Gestalt ist hervorzuheben, da hiermit auf einfache Weise Abläufe und Wirkprinzipien dargestellt werden konnten (vgl. Bild 5-4).

Steckbrief – MBSE-Konzepte								
UE	Projektname	Projektart	Branche	Systemlevel	Referenz			
31	Kannegiesser A	Industrie	Maschinen- und Anlagenbau	System	-			
Kurzbeschreibung			Klassifikation in Anlehnung an Kapitel 4.2					
Ein Ziel des Projekts ReSerW war die Entwicklung eines intelligenten Roboters für die Verarbeitung von Flachwäsche nach dem Waschprozess einer Großwäscherei. Aufgabe des Roboters ist das Auffinden, Greifen und Weiterreichen von Ecken des Wäschestücks. Das Projekt wurde in mehrere Phasen unterteilt. Phase A galt der Identifikation innovativer Ideen zur Gestaltung des Roboters. In Workshops wurden die Einbettung des Roboters in sein Umfeld und notwendige Funktionen und Lösungsprinzipien identifiziert. Hierauf aufbauend wurden Anforderungen an den intelligenten Greifroboter beschrieben. Dazu wurde u.a. CONSENS mit einem Kartenset genutzt, ebenso wurden Handskizzen des Prinzips und einfache Versuche gemacht.			Formalis.grad	informal (80%)	semi-formal (20%)	formal		
			Aspekte	Anforderungen (60%)	Struktur (10%)	Gestalt (15%)	Verhalten (15%)	Parameter
			Vorgehen	inkrementell		iterativ (100%)		
			Phasenablauf	formlos (100%)			formal	
			Phasendefin..	Top-Down (20%)		Bottom-Up	Zigzagging (80%)	
			Werkzeugart	Brown Paper (50%)	Diagr. Tool (50%)	COTS	Simulation	
			PLM	ja			nein (100%)	
			Projektart	Vorentwicklung (100%)		Kundenspez. Entw.	Serienentwicklung	
			Reichweite	projektintern (100%)		Unternehmen	Wertschöpfungskette	
			System-of-Int.	SoS		System (100%)	Modul	
			Verantwortung	Workshop (100%)		Architekt	Fachspezialist	
			Anw.Bereich	Management		Technik (20%)	Enabler (80%)	
MBSE-Konzept		K1 – Innovationswerkstatt						

Bild 5-3 Untersuchungseinheit 31 – Kannegiesser A

In Phase B wurden geeignete Wirkprinzipien zur Realisierung der Funktionen ausgewählt und gemäß CONSENS-Vorgehen zu einer Wirkstruktur auf verschiedenen Systemebenen detailliert – hier als „mechatronische Zeichnung“ aufbereitet und für die weiteren Phasen Entwurf und Ausarbeitung und zur Unterstützung der Kommunikation bereitgestellt. Durch die Unterteilung in zwei Phasen lässt sich ein zeitlicher Verlauf der Anwendung der MBSE-Konzepte in einem Projekt darstellen – der Pfeil in Bild 5-6 zwischen den Untersuchungseinheiten 31 und 32 bringt dies zum Ausdruck. Bild 5-4 zeigt verkürzt einzelne Ergebnisse dieses Vorgehens, Bild 5-5 skizziert Phase B des Projekts im Steckbrief.

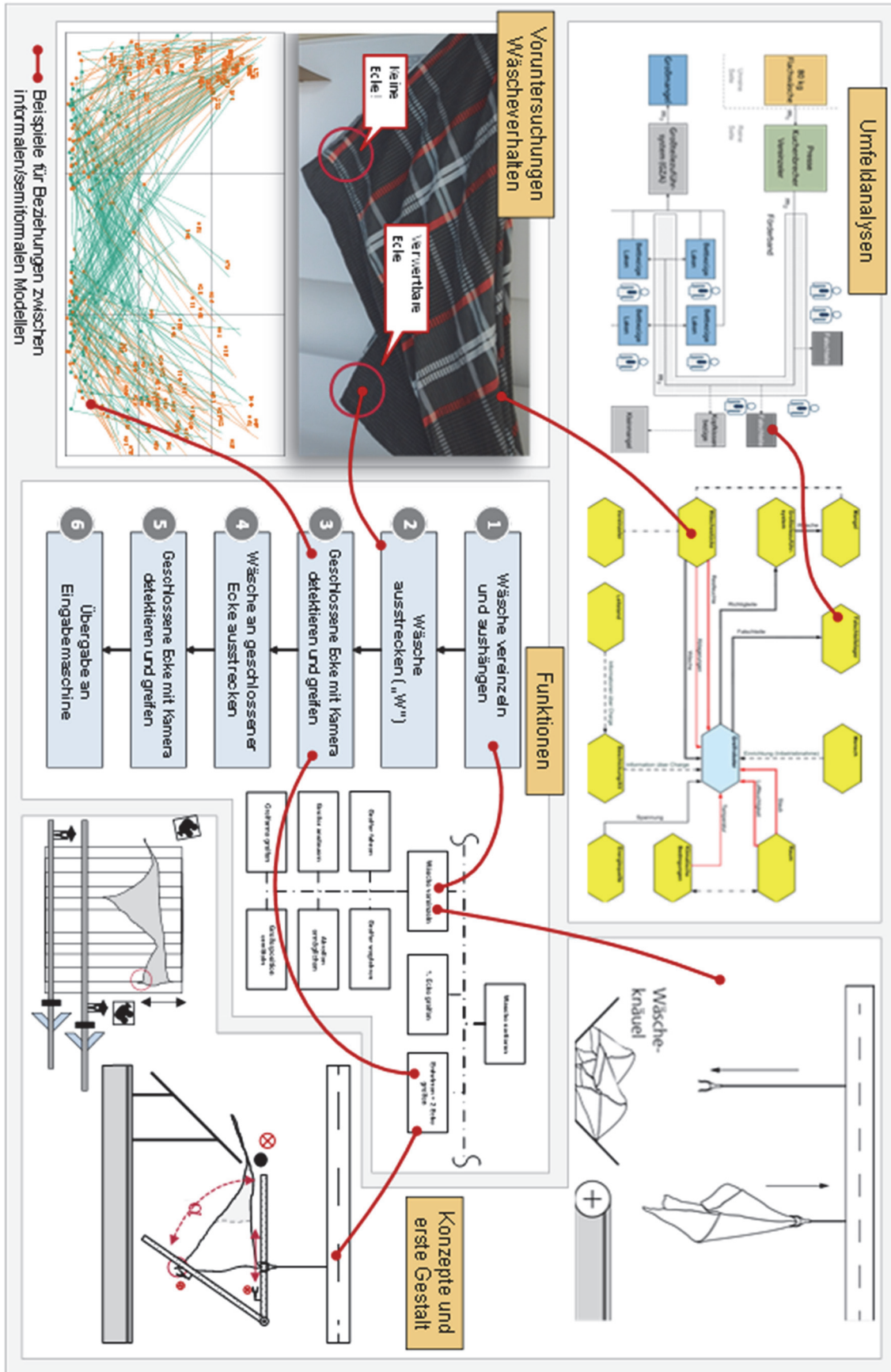


Bild 5-4 Vorgehen im Projekt ReSerW (vereinfacht, Auszug, UE 31)

Untersuchungseinheit 32

Steckbrief – MBSE-Konzepte								
UE	Projektname	Projektart	Branche	Systemlevel	Referenz			
32	Kannegiesser B	Industrie	Maschinen- und Anlagenbau	System	-			
Kurzbeschreibung			Klassifikation in Anlehnung an Kapitel 4.2					
<p>Ein Ziel des Projekts ReSerW war die Einbindung eines intelligenten Roboters für die Verarbeitung von Flachwäsche nach dem Waschprozess einer Großwäscherei. Aufgabe des Roboters ist das Auffinden, Greifen und Weiterreichen von Ecken des Wäschestücks.</p> <p>Das Projekt wurde in mehrere Phasen unterteilt. Phase B galt der Spezifikation auf Systemebene und erster Module. Ein Fraunhofer-Mitarbeiter hat als Systemarchitekt auf Basis vorangegangener Workshops detaillierte mechatronische Strukturbeschreibungen erstellt. Genutzt wurde das Werkzeug MS Visio mit Shapes der Spezifikationstechnik CONSENS. Die Systemspezifikation wurde dem Unternehmen zur Nutzung im Rahmen der weiteren Entwicklung zur Verfügung gestellt.</p>			Formalis.grad	informal	semi-formal (100%)	formal		
			Visual Syntax	keine	Form/Farbe	Form und Farbe (100%)		
			Aspekte	Anforderungen	Struktur (95%)	Gestalt	Verhalten	Parameter (5%)
			Phasenablauf	formlos		formal (100%)		
			Phasendefin.	Top-Down (100%)	Bottom-Up	Zigzagging		
			Werkzeugart	Brown Paper	Diagr. Tool (100%)	COTS	Simulation	
			PLM	ja		nein (100%)		
			Reichweite	projekintern (100%)	Unternehmen	Wertschöpfungskette		
			System-of-Int.	SoS	System (60%)	Modul (40%)		
			Aufwand	gering	mittel		hoch (100%)	
Verantwortung	Workshop (10%)	Architekt (90%)	Fachspezialist					
Anw.Bereich	Management	Technik (20%)	Enabler (80%)					
MBSE-Konzept		K2 – Mechatronic Sketch						

Bild 5-5 Untersuchungseinheit 32 – Kannegiesser B

5.1.3 Zusammenfassung und Bewertung

Aufbauend auf der Analyse sind in Bild 5-6 die vier MBSE-Konzepte und die 35 Untersuchungseinheiten abgetragen. Es stellt multidimensional skaliert die Positionierung der Untersuchungseinheiten in Relation zum MBSE-Konzept dar. Je dichter eine Untersuchungseinheit an einem Konzept liegt, desto größer ist die Ähnlichkeit mit ihm. Gleichzeitig deutet die Entfernung zwischen zwei Untersuchungseinheiten ihre Ähnlichkeit zueinander an.

Die 35 Untersuchungseinheiten spiegeln einen Ausschnitt der aktuellen Aktivitäten im MBSE wider. Dennoch zeigt die Analyse, dass es unterschiedliche Interpretationen und Herangehensweisen an das Themenfeld gibt. Jede der identifizierten Herangehensweisen hat ihre Berechtigung, da die Herleitung der Konzepte maßgeblich auf dem Grundverständnis des Systemdenkens und der Systemtheorie aufbaut und um die grundlegenden Merkmale des modellbasierten Systems Engineering erweitert wurde. Aus methodischer Sicht ergeben sich damit die vier konsistenten MBSE-Konzepte mit ihren unterschiedlichen Schwerpunkten.

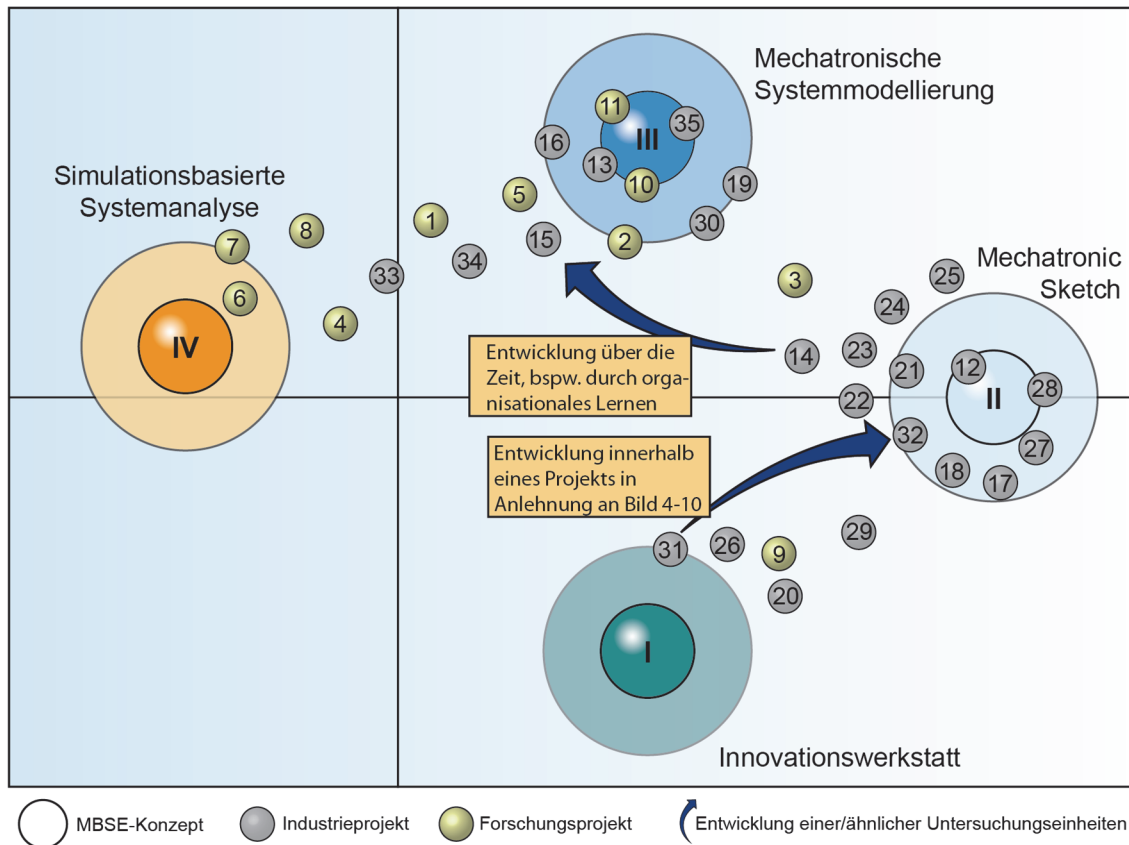


Bild 5-6 Projektlandkarte Modellbasiertes Systems Engineering

Insgesamt ergibt sich ein Bild, das aktuell eine Positionierung der Industrieansätze hin zu Konzept 2 darstellt, mit der Tendenz einer Weiterentwicklung in Richtung Konzept 3 (bspw. UE 14 und 15). Im Sinne eines Produktentstehungsprozesses, der vollkommen auf der Idee des modellbasierten Systems Engineering basiert, müssten sich Schwerpunkte in den einzelnen MBSE-Konzepten über die Projektphasen erkennen lassen, wie es sich bspw. bei den Untersuchungseinheiten UE 31 und UE 32 für die Konzepte K1 und K2 angedeutet. Für Konzept 4 konnten momentan nur Forschungsaktivitäten identifiziert werden, wengleich weitere Aktivitäten im Industrieumfeld anzunehmen sind. Der Grund: Konzept 4 weist eine inhaltliche Schnittmenge mit der Simulations-Community auf, die schon lange besteht. Hier werden bspw. Sprachen wie Modelica genutzt, die durchaus auch von vielen Praktikern eingesetzt werden. Bei der Identifikation möglicher Projekte wurde diese Community aber als disjunkt zur eigentlichen SE-Community betrachtet.

Somit ist festzuhalten: **Die Existenz unterschiedlicher MBSE-Konzepte konnte methodisch hergeleitet und auf Basis einer empirischen Analyse bestätigt werden.**

5.2 Anwendung von Methoden auf Basis des Systemmodells

5.2.1 Vorstellung des Praxisbeispiels


<p>Projektidee: Kaffeeautomat am Flughafenterminal</p> <p>Verfasser: sos/trs</p> <p>Termin: 22.12.2015</p> <p>Produktkizze:</p>		<p>Projektname: Tool Vendor Project GfSE</p> <p>Mission: Mehrere Kaffeesorten (mit unterschiedlichen Zutaten) nach Bezahlung zur Verfügung stellen</p>	<p>Umgebung:</p> <ul style="list-style-type: none"> • Maschine steht in einem Raum. • Die Temperatur beträgt mindestens 15°C. • Die Temperatur beträgt maximal 30°C. • Die Luftfeuchtigkeit ist zwischen 40 und 60 % rel. Luftfeuchte. • Der Zugang zur Montage ist sehr gut möglich, der schmalle Durchgang ist eine Flügeltür (Breite: 2m; Höhe 2,20m). • Es gibt Fenster. • Es gibt künstliche Lichtquellen. • Der Wasserdruck ist max. 3,5 bar. • Der Stromanschluss entspricht deutschem Standard. • Die Maschine steht immer neben einem Imbiss vom Flughafenbetreiber für Strom und Wasseranschluss. • Die Maschinen stehen vor und hinter der Security als auch in den Kantinen. • Die Betriebszeiten des Flughafens sind von 6 bis 23 Uhr. • Ein Zugang zu den Maschinen ist nur während der Betriebszeiten möglich. • Ein LAN Anschluss kann gelegt werden. 	<p>Potentielle Stakeholder:</p> <ul style="list-style-type: none"> • Angestellte und Mitarbeiter vom Flughafen/Security etc. • Passagiere • Wartungsbetriebe • Verwaltung vom Flughafen als Betreiber „Flyport“ (Wirtschaftsabteilung) • Gesundheitsamt • Mitarbeiter von Werttransport 	<p>Kundenwünsche:</p> <ul style="list-style-type: none"> • Ich will meinen eigenen Kaffeebecher (0,25l) mitbringen. • Ich möchte Frischmilch. • Die Stellfläche darf max. 1,5 qm sein. • Das Gerät muss den Hygiene Anforderungen entsprechen. • Die Geräte müssen nach 1 Jahr wirtschaftlich arbeiten. • Die Ausfallrate darf nicht höher sein als 1 mal im Monat für max. 1 Stunde. • Das Geld (EURO) muss sicher verwahrt werden. • Ich möchte mit EC Karte bezahlen können. • Ich möchte mit Kreditkarte bezahlen können. • Das Gerät muss Geldscheine bis 10 Euro annehmen können. • Die Bedienung muss für alle Reisenden und Mitarbeiter schnell und einfach zu bedienen sein. • ...
<p>Stückzahl: 10/anno</p>			<p>Das Tool Vendor Project (TVP) der Gesellschaft für Systems Engineering richtet sich an Softwarehersteller, die die Möglichkeiten ihrer Werkzeuge demonstrieren und vergleichen möchten. Als einheitliches Anwendungsbeispiel dient der Auftrag für einen Kaffeeautomaten in einem Flughafenterminal. Für das TVP existieren als Input für die Softwarehersteller die „vollständigen Kundenanforderungen“, die zu finden sind unter www.tdse.org/typ.</p>		

Bild 5-7 Praxisbeispiel: Kaffeeautomat des Tool-Vendor-Projects der GfSE

In Bild 5-7 ist in Form eines Steckbriefes ein Großteil der Informationen des sog. Tool-Vendor-Projects zusammengefasst, das jährlich im Rahmen des Tag des Systems Engineerings der Gesellschaft für Systems Engineering (GfSE) e.V. durch verschiedene Tool-Vendoren zur Demonstration der Leistungsfähigkeit ihrer Werkzeuge genutzt wird. Auf diesem Beispiel aufbauend wurde bereits die Stakeholderanalyse in Kapitel 4.6 vorgestellt. Hier soll die Anwendbarkeit weiterer Methoden und Hilfsmittel demonstriert werden. Das gelingt anhand der Gefährdungsanalyse in Anlehnung an NOHL [NT88] und einer Komplexitätsanalyse in Anlehnung an MARTI [Mar07]. Beide wurden über den Quick Check in Kapitel 4.6.1.1 als geeignet identifiziert und werden nun aufbauend auf der Modellierung des Kaffeeautomaten mit der Spezifikationstechnik CONSENS erarbeitet. Generell liegt der Fokus hier auf der Anwendung der Methodenkopplung und damit verbundenen Darstellung des Ergebnisses.

5.2.2 Gefährdungsanalyse in Anlehnung an NOHL (SysML4CONSENS)

Mit der Gefährdungsanalyse nach NOHL werden Elemente in einer Matrix in unterschiedliche Risikoklassen klassifiziert [NT88]. In Abhängigkeit der Eintrittswahrscheinlichkeit einer Gefährdung und der Schadenshöhe – die von dem Element und seinen Einflüssen ausgeht – ergeben sich i.d.R drei Risikoklassen: *geringes Risiko*, *signifikantes Risiko* und *hohes Risiko*. Eintrittswahrscheinlichkeit und Schadenshöhe werden kardinal skaliert von 1 bis 4 oder 0 bis 4. Wird ein Element in die Risikoklasse *gering* eingeordnet, ist meist keine Maßnahme zur Gefahrenreduzierung notwendig. Bei *signifikantem Risiko* müssen Schritte zur Risikoreduzierung durchgeführt werden, ein *hohes Risiko* fordert dringenden Handlungsbedarf. Die Grenzen der unterschiedlichen Klassen können dabei aufgabenspezifisch festgelegt werden.

Bei der Analyse können verschiedene Elemente betrachtet werden. Hier soll das Risiko der Einflüsse auf das System beurteilt werden – also sind Flüsse, die aus dem Umfeld mit dem Systemelement Kaffeeautomat wechselwirken entsprechende Risikoquellen, bzw. umgekehrt Risikofolgen. Im Prinzip ergibt sich nach Durchlauf des Vorgehensmodells die in Bild 5-8 dargestellte Essenz und das Sichten-Template (Auszug), auf dem aufbauend die Systemmodellierung starten kann: Hier wurden aus der Datenbank alle Flussobjekte gezogen, die das Black-Box-Element Kaffeemaschine mit den Umfeldelementen verbindet. Dazu muss überprüft werden, ob der Port an einem Ende des Flusses tatsächlich zu einem *UmfeldElement* gehört. Damit die Flüsse korrekt dargestellt werden können, müssen den Ports Flussobjekte zugewiesen werden, die den Namen des Flusses enthalten. Die entsprechenden Werte für die Eintrittswahrscheinlichkeit, die Schadenshöhe und das Risiko werden als Attribute in den entsprechenden Flussobjekten erzeugt. Schließlich ergibt sich über die php-Anwendung das im Mock-Up von Bild 5-9 dargestellte Ergebnis.

Analysephase – Informationsgewinnung: Essenz und Sichten-Template

Grundlage dieser Umsetzung ist die in Bild 5-8 dargestellte Methodenessenz. Die wesentlichen Schritte der Gefährdungsanalyse wurden entsprechend des Schemas aus Kapitel 4.6.2.2 analysiert und auf die Bereiche der Systemmodellierung, der Transformation und der Nutzung verteilt. Daraus lässt sich dann zusätzlich das Sichten-Template ableiten, das hier für die prozessbezogenen Zusammenhänge dargestellt ist.

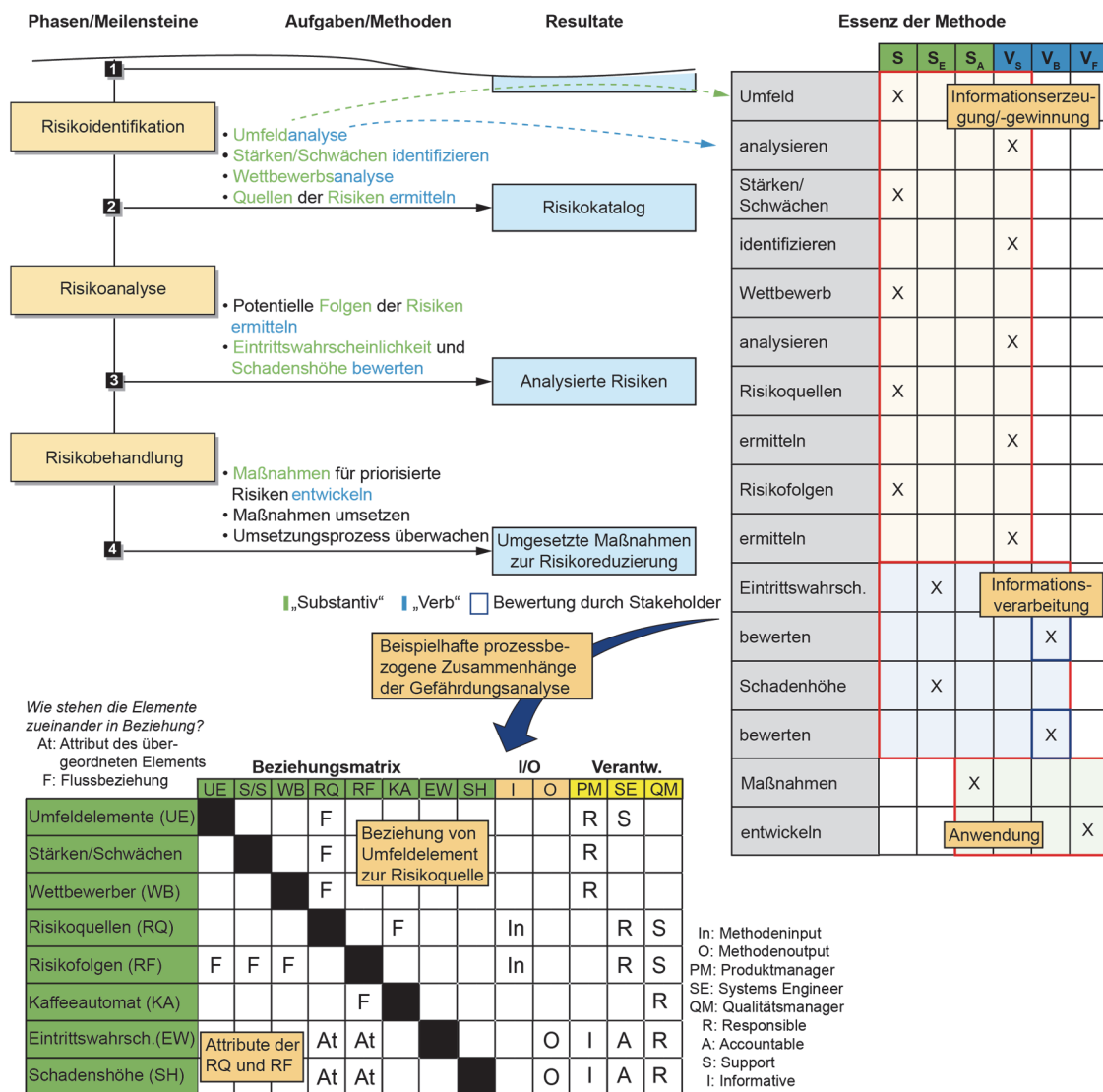


Bild 5-8 Methoden-Essenz und prozessbezogene Zusammenhänge des Sichten-Template der Gefährdungsanalyse nach NOHL

Der Teil der methodenspezifischen Zusammenhänge kann dann direkt genutzt werden, um die entsprechenden Modellierungsaktivitäten durchzuführen. Im Gegensatz zur Stakeholderanalyse aus Kapitel 4 wird hier die Bedeutung der Flüsse ergänzt, zudem sind grundsätzlich sämtliche Elemente aus dem Umfeld des Kaffeeautomaten möglich. Hier zeigt sich jedoch, dass eine genaue MBSE-Methodenkenntnis notwendig ist: Bei CONSENS sind bislang keine Elemente bekannt, die qualitative Aspekte des Entwicklungs-

umfelds beschreiben, bspw. Stärken oder Schwächen – die im Prinzip jedoch grundsätzlich modellierbar wären.

Control-Phase: Zusammenführung im Komplexitätsportfolio und Anwendung

Im Mock-Up sind nur einige Elemente dargestellt – insg. wurden 31 Elemente zur Bewertung identifiziert. Die Bewertungen werden schließlich wieder in das Systemmodell übermittelt. Im Mock-Up ist erkennbar, dass vom Geldschein ein signifikantes Risiko für das System ausgehen kann – bspw. aufgrund beschädigter Scheine, die nicht erkannt werden. Hierfür sind entsprechende Maßnahmen zu treffen, s.d. das Risiko reduziert wird. Das erfordert wiederum Spezialistenwissen und könnte im vorliegenden Fall bspw. den Abschluss von Geldscheinen mit hohem Wert bedeuten (also bspw. max 20 Euro-Schein).

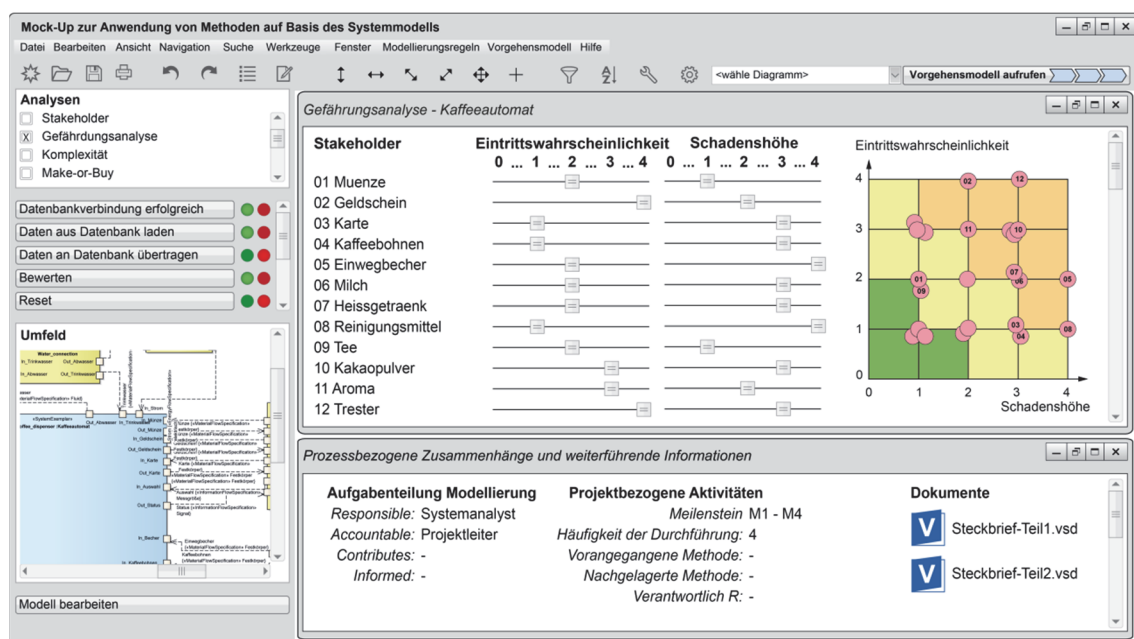


Bild 5-9 Mock-Up der prototypischen IT-Infrastruktur für die Gefährdungsanalyse, in Anlehnung an [GTH16]

Somit ist festzuhalten: **Das Vorgehen zur Kopplung der Stakeholderanalyse konnte erfolgreich auf die Gefährdungsanalyse übertragen werden.**

5.2.3 Komplexitätsanalyse des Kaffeeautomaten in Anlehnung an MARTI (SysML4CONSENS)

Häufig steht in Diskussionen das Thema „Komplexität von Mechatronik“ im Mittelpunkt. Durch das Systemmodell ergeben sich neue Möglichkeiten zur Bewertung der Systemkomplexität, da bspw. durch die Unterscheidung unterschiedlicher Flüsse in der Wirkstruktur weitaus mehr Wissen über das System verfügbar ist als bislang üblich – und auch früher verfügbar ist. Als Beispiel für eine solche Komplexitätsanalyse wird in Anlehnung an die Komplexitätsanalyse nach MARTI [Mar07] die mechatronische Komplexität des Kaffeeautomaten bestimmt. Die Methode von MARTI bewertet mechanische

Bauteile; die Methode wurde deshalb so angepasst, dass sie die Inhalte des Systemmodells nutzt und dadurch die mechatronische Komplexität bestimmt [GTH16]. Als Ergebnis wird in einem Portfolio der Nutzen einzelner Systemelemente aufgetragen, repräsentiert durch ihren Beitrag zu einzelnen Funktionen, über ihrer mechatronischen Komplexität²². Das resultierende Portfolio ist in vier Quadranten aufgeteilt. Jeder Quadrant bildet eine eigene Kategorie mit entsprechenden strategischen Handlungsmöglichkeiten. Aber allein durch die Visualisierung ergibt sich als Nutzen der Methode ein sehr gutes Verständnis über die Bedeutung eines einzelnen Elements im Gesamtsystem. Bild 5-10 gibt einen Überblick über das Konzept der Komplexitätsanalyse und die Berechnung der mechatronischen Komplexität. Zusätzlich zur Komplexitätsformel wurden die Namen der Quadranten neu definiert. Somit gilt: Hat eine Komponente einen hohen Funktionsbeitrag bei niedriger mechatronischer Komplexität C_i handelt es sich um einen Lucky Strike – d.h. die Komponente weist entsprechend der adaptierten Formel wenige Beziehungen zum Rest des Systems auf (ist also weitgehend entkoppelt), bietet aber viel Nutzen für den Kunden. Analog sind die weiteren Quadranten zu interpretieren.

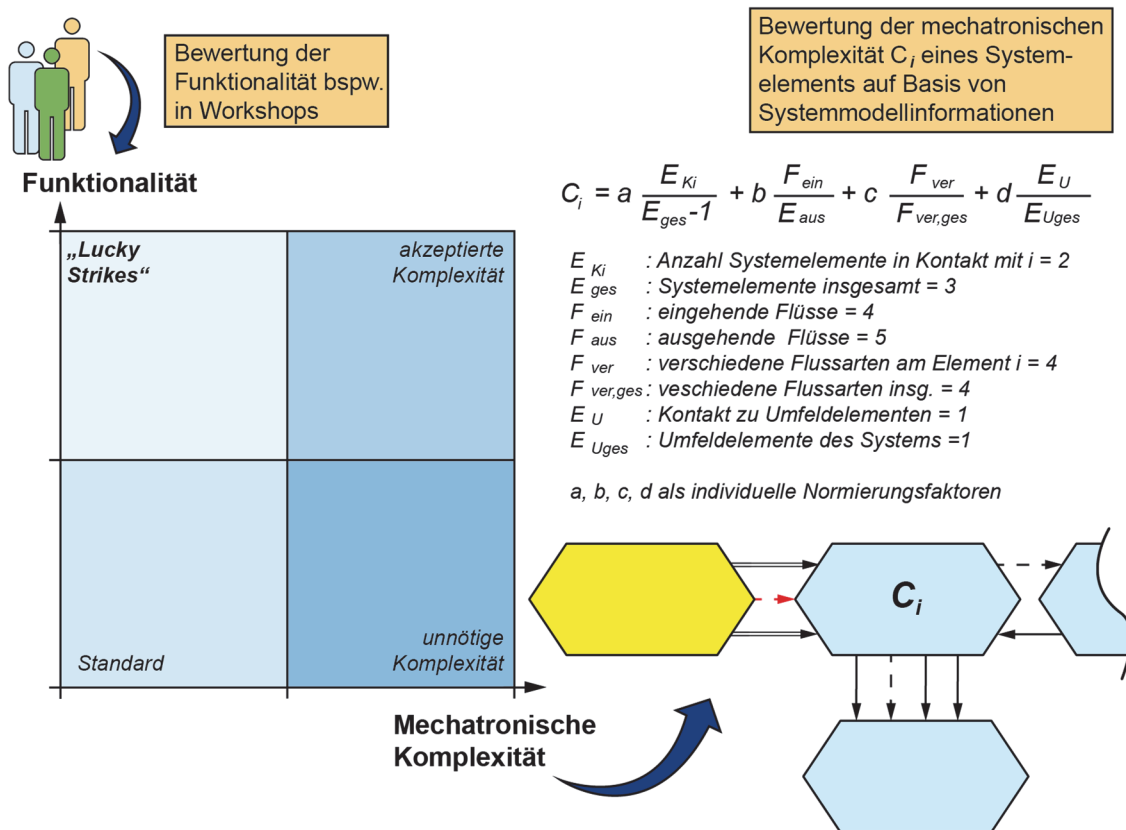


Bild 5-10 Konzept und Berechnung der mechatronischen Komplexität.

Aufgrund der schwierigen Informationsakquise konnten derartige Methoden bislang in der Produktentstehung nicht genutzt werden – erst recht nicht aufgrund der zahlreichen

²² Für Details: [Mar07] und Anhang A6.3.

Beziehungen mechatronischer Komponenten untereinander. Gleichzeitig zeigt die Adaption der Komplexitätsanalyse nun exemplarisch, dass derartige Methoden zukünftig nachhaltig und einfach auf Basis des Systemmodells anwendbar werden. Das unterstreicht die Potentiale, die sich durch die frühzeitige Systemmodellierung ergeben.

Analysephase – Informationsgewinnung: Essenz und Sichten-Template

MARTI stellt in seiner Methode die Tätigkeiten zur Erstellung der Complexity Matrix vor. Hieraus lässt sich die Essenz ausleiten und mit relevanten weiteren Inputs in das Sichten-Template überführen (vgl. Bild 5-11).

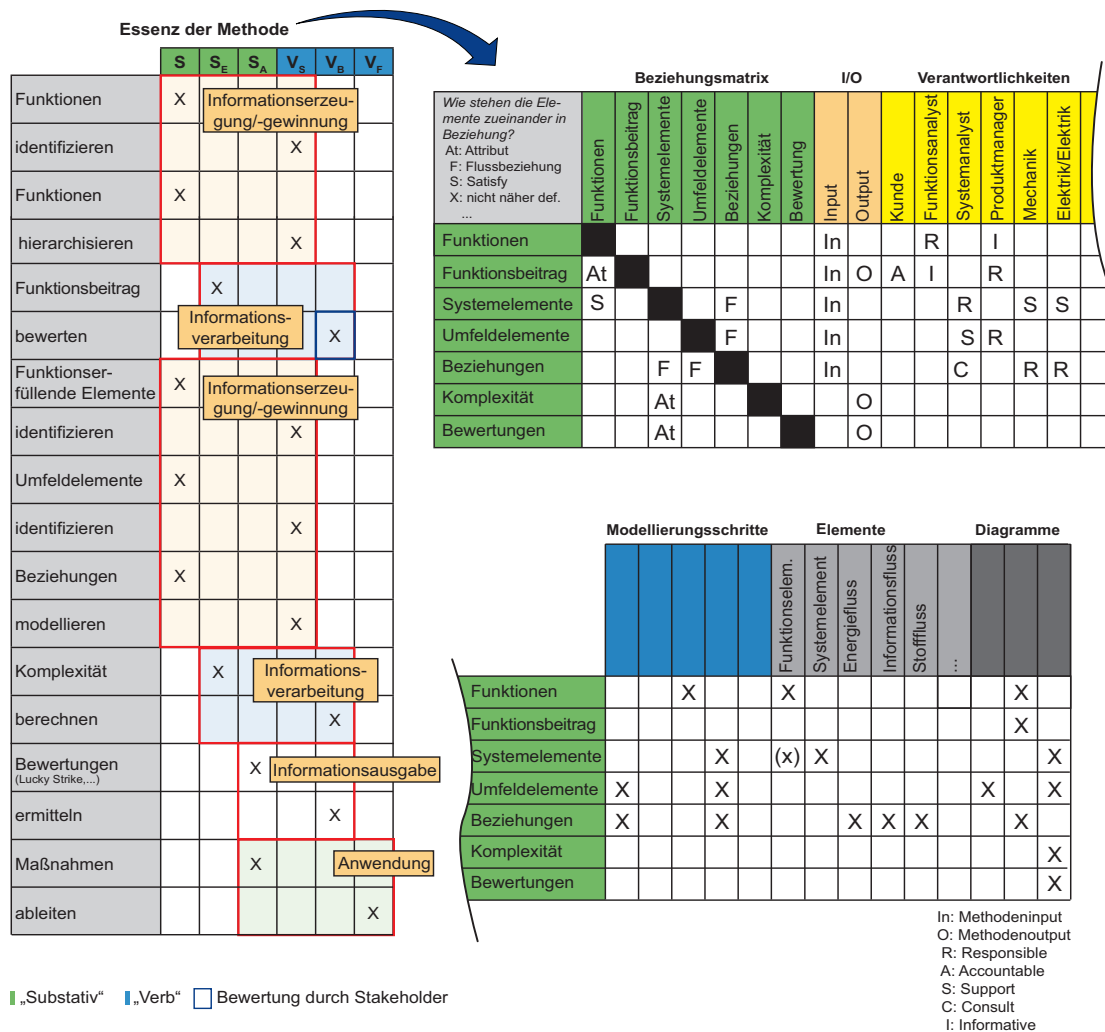


Bild 5-11 Sichten-Template

Informationserzeugung – Design und Control: Modellierung

Mit Unterstützung des Sichten-Templates kann die Modellierung organisiert werden. Neben der Funktionshierarchie müssen die erfüllenden Systemelemente definiert und hierarchisiert werden. Der Kaffeeautomat wurde dazu hierarchisch strukturiert und in acht übergeordnete Systemelemente gegliedert, die wiederum in 40 Systemelemente

unterteilt sind. Nach dem Mapping von Funktionen und Systemelementen können dann die einzelnen Elemente und ihre Beziehungen detailliert modelliert werden – dazu bietet es sich an, Sichten in einzelnen Diagrammen zu modellieren, bspw. die energie- oder die informationsspezifische Sicht. Durch das Repository sind alle modellierten Systemelemente und ihre Beziehungen auslesbar. Bild 5-12 zeigt einen Auszug aus der Funktionshierarchie – so wird die Funktion *Kaffeebohnen mahlen* durch das Element *Mahlwerk* realisiert, die Funktion *Mit Karte bezahlen* durch das Element *EPP*²³.

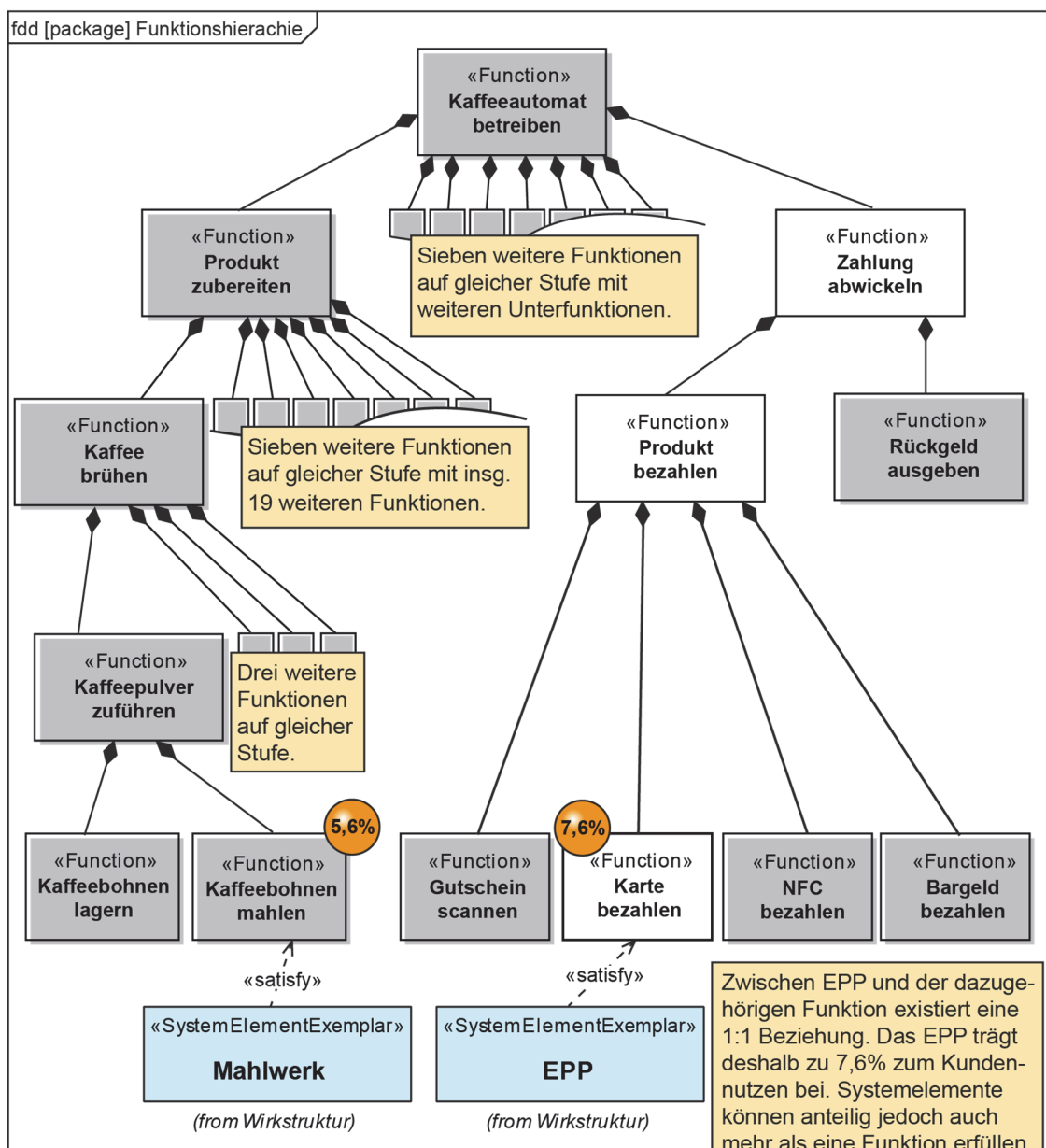


Bild 5-12 Auszug aus dem Systemmodell inkl. Beziehungen zwischen einzelnen Diagrammen, beispielhafte Darstellung „EPP – Encrypting Pinpad“

²³ Beim EPP – Encrypting Pinpad handelt es sich um ein Tastenfeld zur Eingabe einer Geheimzahl im Rahmen der bargeldlosen Bezahlung. Häufig umfasst es als Modul auch das Kartenlesegerät.

Wie im Ausschnitt des Blockdefinitionsdiagramms in Bild 5-13 zu sehen, ist das EPP Teil der zweiten Hierarchieebene des Kaffeeautomaten – neben acht weiteren Systemelementen. Das ist auch im kleinen Bildausschnitt des Project Browsers der verwendeten Software angedeutet. Auf dieser Ebene befinden sich die meisten Zuordnungen von Funktionen und Systemelementen. Ebenso sind die zahlreichen Ports des EPP sichtbar. Am EPP sind alle möglichen Flussarten anliegend. Den Ports müssen Flussobjekte zugewiesen werden, so dass die unterschiedlichen Flussarten später auch erfasst werden können. In einem nächsten Schritt erfolgt die Modellierung der Wirkstruktur – hierzu wurde das System in Anlehnung an [Kai14] in unterschiedlichen Sichten modelliert²⁴.

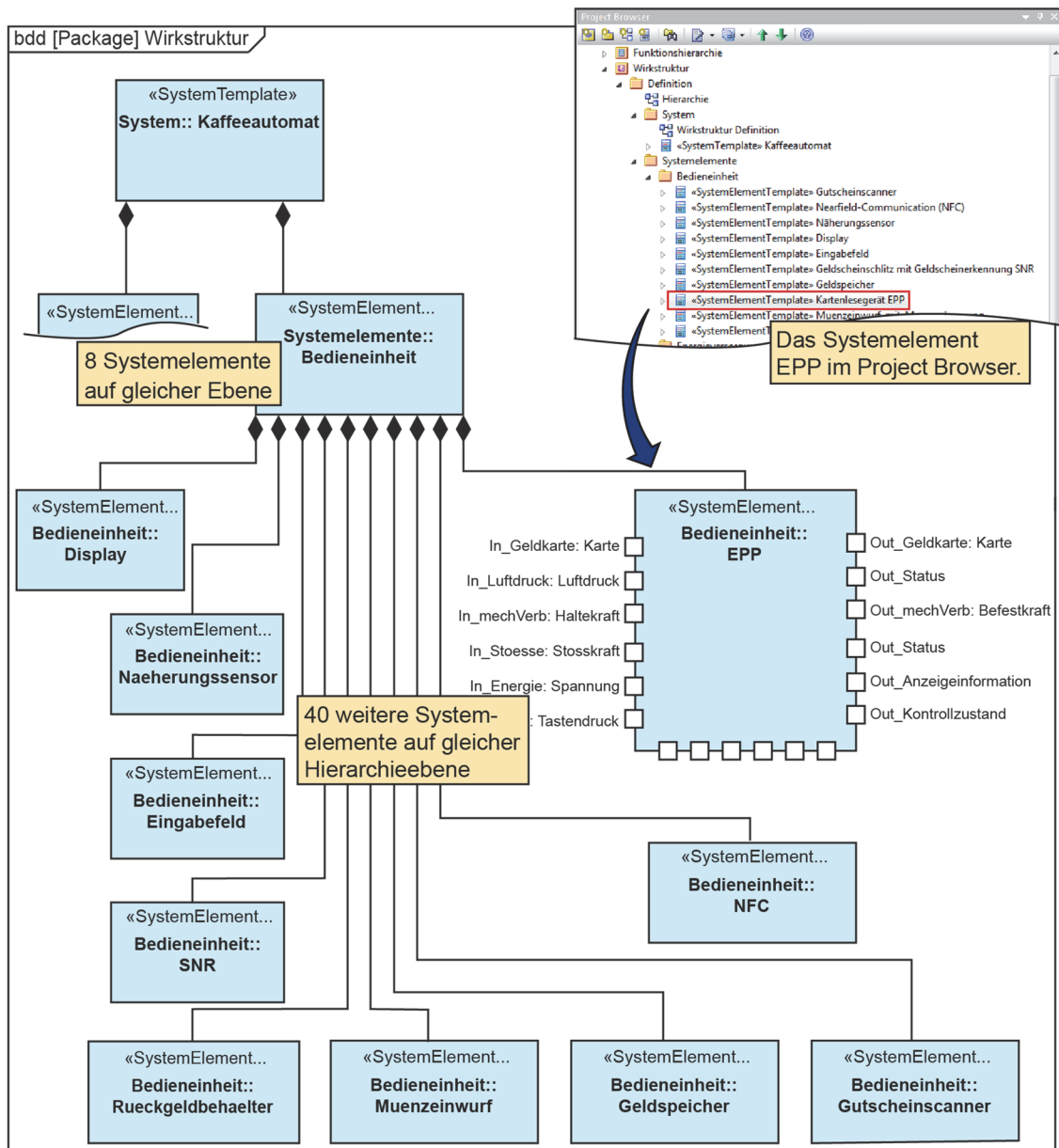


Bild 5-13 Blockdefinitionsdiagramm Kaffeemaschine (Auszug)

²⁴ Vgl. hierzu die Beispiele in Anhang A5.3

Design-Phase: Umsetzung der Methode zur Anwendung

Zur Bewertung der Funktionalität verweist MARTI auf das Verfahren von TANAKA [Tan89], das im vorangegangenen Text grob erklärt wurde: Die Gesamtfunktion des Systems wird in Unterfunktionen zerlegt und der durch die jeweilige Funktion entstehende Kundennutzen prozentual beurteilt – bspw. in Kundenworkshops. Da den Funktionen Systemelemente zugeordnet sind, addieren sich die bewerteten Funktionsanteile zum Funktionsbeitrag (interpretiert als Kundennutzen) des Systemelements. Wird eine Funktion durch mehrere Systemelemente realisiert, erfolgt eine Gleichverteilung.

Bei der Anwendung auf Basis von CONSENS wird zunächst die Funktionshierarchie aus dem Systemmodell übernommen und mit einer Bewertungsmaske versehen. Im Beispiel in Bild 5-14 ist das für die Hauptfunktion *Zahlung abwickeln* aufgezeigt: Sie trägt zu 10% zur Funktionserfüllung bei – die übrigen Funktionen dieser Hierarchie leisten also 90% zur Gesamtfunktion *Produkt zubereiten* bei (vgl. Funktionshierarchie im Anhang A5.1). *Zahlung abwickeln* teilt sich wieder in weitere Funktionen auf, bis schließlich die Funktion *Karte zahlen* zu 7,6% zum Ergebnis beiträgt. Dem SysML4CONSENS-Profil entsprechend, brauchen die Funktionen somit ein Attribut zur Speicherung des prozentualen Beitrags zur Gesamtfunktion.

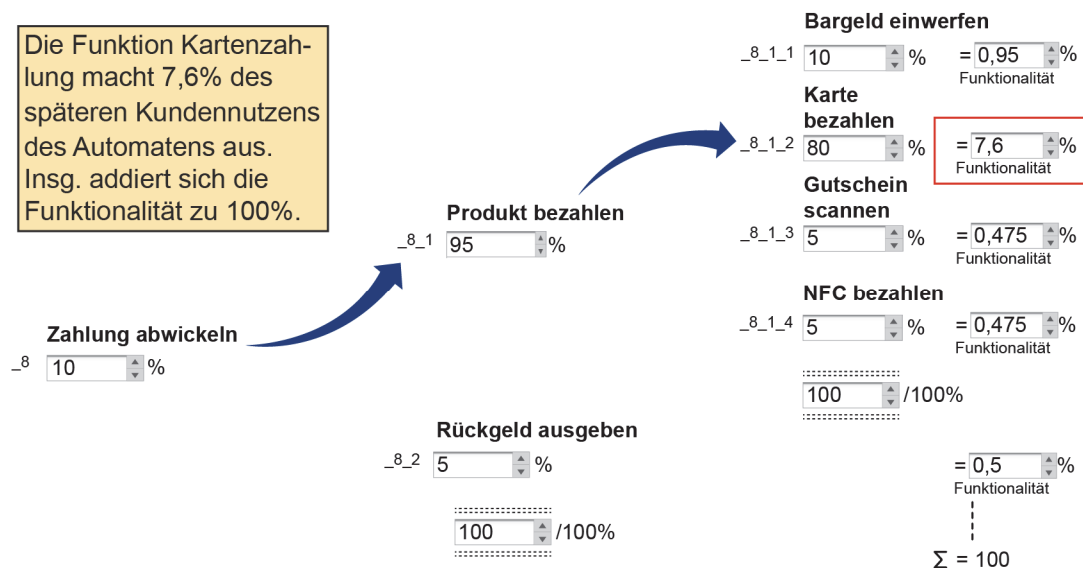


Bild 5-14 Von der Funktionshierarchie zur Funktionsbewertung – z.B. Kartenzahlung

Bewertung der Mechatronischen Komplexität

Die „mechatronische“ Komplexität wird über die in Bild 5-10 präsentierte Formel bestimmt, bei der sämtliche Informationen aus der Wirkstruktur stammen²⁵. Die Funktion *Karte bezahlen* wird wie in Bild 5-12 dargestellt durch das Systemelement EPP realisiert. Das Vorgehen zur Berechnung der mechatronischen Komplexität bleibt im Rah-

²⁵ Weitere Details zur Berechnung im Anhang A6.3.1.

men der adaptierten Methode grundsätzlich gleich zur Methode von MARTI: Die Berechnung erfolgt über die Addition von Verhältnissen verschiedener Element- und Systemparameter. Über alle Elemente hinweg wird dann die mechatronische Komplexität im System so normiert, dass das „komplexeste“ Element einer Komplexität von 1 entspricht. Generell wurden die Faktoren dabei aber so modifiziert, dass die Komplexität der Elemente rein aus dem Systemmodell abgeleitet werden kann. Das gelingt über die im Werkzeug modellierten Beziehungen zwischen Funktionen und den Systemelementen untereinander. Die jeweiligen Bewertungsparameter können für weitere Anwendungen selbstredend noch angepasst werden.

Control-Phase: Zusammenführung im Komplexitätsportfolio und Anwendung

Aus der Funktionshierarchie werden die im Workshop bewerteten Funktionen automatisch auf die y-Achse übertragen, wobei noch eine Normierung der y-Achse stattfindet – das Maximum des Funktionsbeitrags ist gleichzeitig auch das Maximum der y-Achse. Die Bewertung der mechatronischen Komplexität erfolgt auf Basis der in der Wirkstruktur modellierten Inhalte – die Bewertungen werden für Elemente zugeordnet, die mindestens eine Funktion der Funktionshierarchie erfüllen. Im resultierenden Portfolio ergibt sich durch dieses Vorgehen das in Bild 5-15 dargestellte beispielhafte Ergebnis. Die Diskussion der Ergebnisse steht hierbei nicht im Mittelpunkt. Dennoch ist unter anderem erkennbar, dass

- das EPP einen sehr hohen Funktionsbeitrag leistet im Gegensatz zu anderen Zahlungsmitteln wie z.B. Geldscheinzahlung (SNR)
- der Cappuccinatore, der im Prinzip eine „Komfortfunktion“ erfüllt (Erzeugung von Milchschaum), zu den Lucky Strikes gehört
- das Mahlwerk sehr aufwändig erscheint
- die Energieversorgung der Maschine eine Komplexität von 1 hat – bei gleichzeitig sehr geringem Nutzen für den Kunden.

Die Schlüsse, die auf Basis dieser Analyse gezogen werden können, sind vielfältig. Mit Bezug auf die in Kapitel 2.2 dargestellten Herausforderungen der Produktentstehung wird klar, dass auf Basis dieser frühzeitig durchgeführten Methode

- die Energieversorgung bspw. von einem externen Partner beschafft werden könnte und nicht selbst entwickelt werden sollte
- alternative Zahlungsmittel wie SNR (Geldschein) nicht den erhofften Nutzen bieten und ggf. deshalb nicht in das System integriert werden sollten
- zahlreiche Lucky Strikes möglich erscheinen – hier sollte selbst entwickelt werden und bspw. eine hohe Verfügbarkeit des Cappuccinatore angestrebt werden.

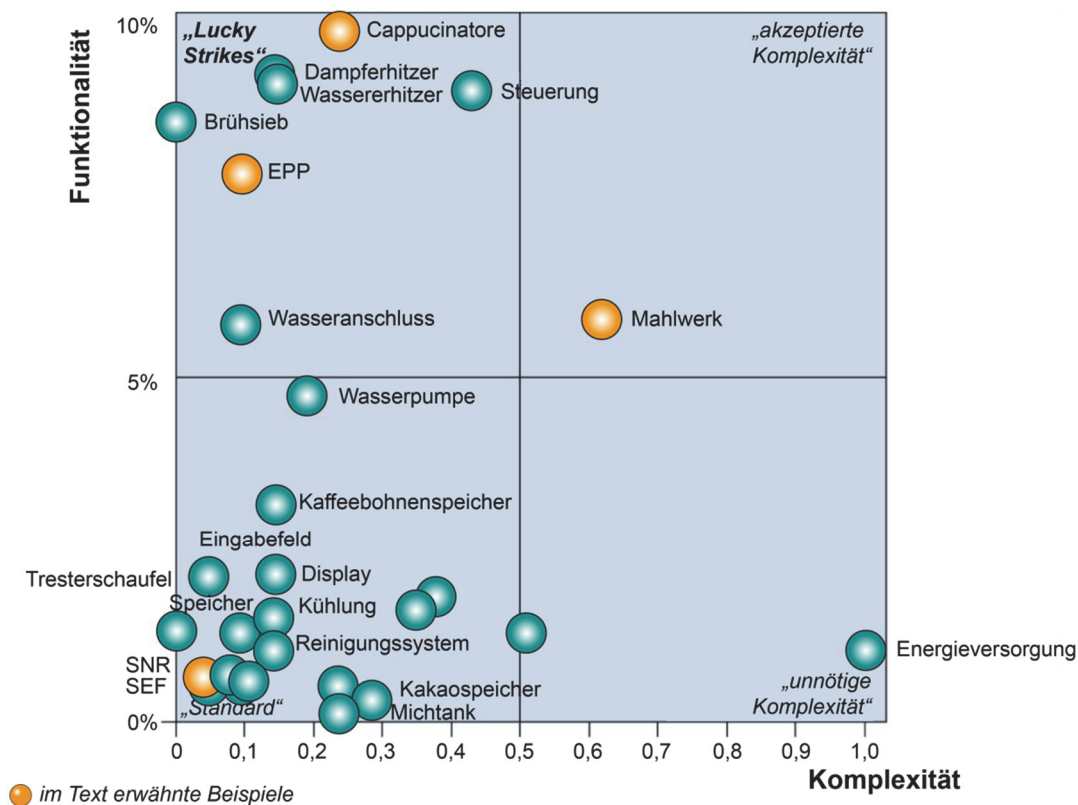


Bild 5-15 Automatische Erzeugung der Complexity Matrix aus dem Systemmodell, in Anlehnung an [Mar07], [GTH16]

Somit ist festzuhalten: **Das Vorgehen zur Kopplung der Stakeholderanalyse konnte erfolgreich auf die Komplexitätsanalyse übertragen werden.**

5.3 Bewertung der Arbeit anhand der Anforderungen

In Kapitel 4 wurde das *Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme* entwickelt und gleichzeitig anhand der durchgängigen Anwendung einer Stakeholderanalyse beispielhaft erläutert. Fundament des Rahmenwerks sind die vier MBSE-Konzepte – ihre Existenz wurde in Kapitel 5.1 anhand einer empirischen Studie bestätigt. Darüber hinaus wurde in Kapitel 5.2.2 und Kapitel 5.2.3 gezeigt, dass die Anwendung weiterer Methoden aus dem Umfeld des Technical Managements im Sinne der ISO15288 mit Elementen des Rahmenwerks unterstützt werden können.

In diesem Abschnitt wird das Rahmenwerk nun anhand der in Kapitel 2.7 formulierten Anforderungen bewertet. Hierzu wird zu jeder Anforderung eine Übersicht gegeben, inwieweit das Rahmenwerk zur Erfüllung der Anforderung beiträgt.

A1) Systems Engineering als Basis: Bei der Entwicklung des Rahmenwerks wurden die wesentlichen Konzepte des Systems Engineerings konsequent berücksichtigt. So stehen im Beschreibungsmodell zur Nutzung des Systemmodells die zwei

Bereiche technische Anwendungen und Managementanwendungen im Mittelpunkt, was im Einklang mit dem Konzept von HABERFELLNER ist und darüber hinaus insb. die Technical Management Processes der ISO15288 adressiert. Die MBSE-Konzepte erfüllen diese Anforderung ebenfalls, da sie für unterschiedlichste Anwendungsbereiche in der Produktentstehung ausgelegt sind – hier wurde insb. die Idee des lebenszyklusbegleitenden Ansatzes hervorgehoben. Mit dem flexiblen Vorgehensmodell und seiner Aufteilung in einen Makro-Prozess und einen Mikro-Prozess wurde zudem der Grundsatz „Vom Abstrakten zum Konkreten“ erfüllt. Darüber hinaus wurde mit der konzeptionellen Erweiterung der ISO42010 eine richtungsweisende SE-Norm berücksichtigt. Somit nutzt das Rahmenwerk die wesentlichen Normen und Konzepte des SE als Ansatzpunkte.

- A2) Systemmodell als Mittelpunkt sämtlicher Aktivitäten:** Mit der Anwendung von Methoden auf Basis des Systemmodells wurde dieser Anforderung Rechnung getragen – dadurch wurde zudem die methodische Durchdringung der Produktentstehung gestärkt, was wiederum eine stärkere Fokussierung auf das Systemmodell mit sich zieht. Darüber hinaus wurden mit den beispielhaft angewendeten Methoden die frühen Phasen der Produktentstehung adressiert, weshalb von Projektbeginn an das Systemmodell existieren muss, mit dem das Geschehen in der Produktentstehung auch zu einem späteren Zeitpunkt orchestriert werden kann.
- A3) Klares MBSE-Konzept:** Diese Anforderung wird insb. durch die methodisch hergeleiteten vier MBSE-Konzepte erfüllt. Aus einer Analyse der Handlungsfelder der Produktentstehung und Systems Engineering wurden 17 Merkmale mit insg. 50 Merkmalausprägungen identifiziert, die durch eine Konsistenzanalyse und anschließender multi-dimensionaler Skalierung zu vier konsistenten, eindeutigen MBSE-Konzepten zusammengefasst werden konnten. Dass es sich hierbei nicht nur um theoretische Konzepte handelt, wurde anhand der Untersuchung von 35 Projekten und Anwendungsbeispielen aus Wissenschaft und Industrie belegt.
- A4) Sprachen- und Methodenunabhängigkeit:** Die Sprachen- und Methodenunabhängigkeit des Rahmenwerks wurde konsequent über alle Elemente des Rahmenwerks verfolgt: Die MBSE-Konzepte berücksichtigen zwar Merkmale, die sich auf MBSE-Sprachen und Modellierungsmethoden beziehen, jedoch unabhängig von einer konkreten Sprache oder Methode sind. Die Entscheidungsunterstützung fokussiert Aspekte der Anwendung, wie z.B. Lebenszyklus oder Art der Methode – jedoch nicht sprachen- oder methodenspezifische Kriterien. Gleiches gilt für das Konzept zur Nutzung des Systemmodells in Kapitel 4.4. Das flexible Vorgehensmodell nutzt Hilfsmittel, die bei der Auswahl einer Modellierungsmethode unterstützen – die Anwendung der weiteren Elemente ist aber vollkommen unabhängig von der gewählten Sprache und Methode. Letzteres wurde am Beispiel der Stakeholderanalyse dargestellt; sie wurde sowohl mit der Methode CONSENS als auch der Methode SysMod unterstützt.

- A5) Integration von MBSE in die Organisation und Prozesse:** Das Konzept zur Nutzung des Systemmodells unterscheidet die vier Aufgabenbereiche Modellplanung, Modellierung, Methodenkopplung und Anwendung des Systemmodells. Hierauf aufbauend wurde hergeleitet, dass die Anwendung des Systemmodells gut mit dem Konzept der Trennung von Modell und Sicht unterstützt werden kann. Damit dies auch im Produktlebenszyklus gelingen kann, wurde eine Erweiterung der ISO42010 vorgeschlagen – womit eine Grundlage für die Integration in die Organisation und die Prozesse der Produktentstehung gelegt ist. Durch die weiteren Elemente des Rahmenwerks wurde die Integration dann unterstützt, so z.B. die Entscheidungsunterstützung und Methoden und Hilfsmittel des flexiblen Vorgehensmodells – hierzu gehören bspw. der Quick Check und die Möglichkeiten zur Organisation der Modellierungsaufgabe.
- A6) Adaption von Erfolgsprinzipien bekannter Managementansätze:** Ansätze wie Six Sigma zeigen eine Vision auf und bestehen aus vielen einfachen Hilfsmitteln, die zusammen und allein eingesetzt werden können und beliebig erweiterbar sind. Diese drei Eigenschaften konnten ebenfalls im Rahmenwerk verankert werden. Durch das Konzept zur Nutzung des Systemmodells konnte die grundlegende Marschrichtung für das modellbasierte Systems Engineering aufgezeigt werden. Insb. das flexible Vorgehensmodell bietet den richtigen Rahmen für die Integration zahlreicher Methoden und Hilfsmittel, die die Anwendung von MBSE unterstützen sollen. Dabei wurde das flexible Vorgehensmodell bewusst an den DMAIC-Circle des Six Sigma angelehnt.
- A7) Rationalitätssicherungsfunktion:** Grundlage zur Sicherung der Rationalität bei der Integration des modellbasierten Systems Engineering in die Produktentstehung ist die Definition der MBSE-Konzepte. Der eigentliche Beitrag zur Rationalitätssicherung wurde durch die verschiedenen Entscheidungshilfen in Kapitel 4.3 geleistet. Dabei wird die Entscheidung für ein MBSE-Konzept aus vier unterschiedlichen Blickwinkeln unterstützt – jedoch dem Anwender nicht abgenommen. Einen weiteren Beitrag zur Sicherung der Rationalität liefert das flexible Vorgehensmodell mit seinen idealtypischen Vorgehensweisen.
- A8) Stakeholderorientierte Modellierung und Nutzung des Systemmodells:** Es wird grundsätzlich zwischen technischen Anwendungen und Management-Anwendungen auf Basis des Systemmodells unterschieden. Das geschieht in Anlehnung an die ISO15288 und die dort definierten Prozessbereiche. Durch die Erweiterung der ISO42010 und ihre zentrale Position im Konzept zur Nutzung des Systemmodells wird der *Concern* eines *Stakeholders* in den Mittelpunkt der Modellierung gestellt. Der Fokus liegt in der Arbeit auf managementnahen Anwendungen und insb. auf Methoden der Technical Management Processes. Insgesamt kann die Anforderung deshalb als erfüllt betrachtet werden.

- A9) Wohlfundierte Modellplanung:** Die Modellplanung ist Ausgangspunkt für die Anwendung des modellbasierten Systems Engineerings – das wurde im Konzept zur Nutzung des Systemmodells verankert. Das flexible Vorgehensmodell greift dies auf und verankert diesen Hauptaufgabenbereich in den Phasen *Define* und *Measure/Analyse* und bietet ein erstes Set an entsprechenden Hilfsmitteln. Durch die Dokumentation der einzelnen Schritte in der Modellplanung wurde zudem die Nachhaltigkeit in der Anwendung sichergestellt.

Das erarbeitete *Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme* erfüllt damit alle gestellten Anforderungen. In der Anwendung wurde darüber hinaus gezeigt, dass Bestandteile des Rahmenwerks aufeinander aufbauen, jedoch auch einzeln eingesetzt werden können.

Mit dem Rahmenwerk wurden somit ebenfalls die in Kapitel 2.6 formulierten Forschungsfragen beantwortet und dadurch ein Beitrag zur Lösung der dargestellten Herausforderungen des modellbasierten Systems Engineerings geleistet.

6 Zusammenfassung und Ausblick

Die Vision von intelligenten technischen Systemen (ITS) ist faszinierend. Diese Systeme sind in der Lage, intelligent und flexibel auf veränderte Umgebungsbedingungen zu reagieren. Sie sind adaptiv, robust, vorausschauend und benutzungsfreundlich. Begriffe wie Industrie 4.0, Internet-of-Things, Cyber-Physical Systems geben dieser Vision markante Namen. In Initiativen, wie dem vom BMBF-geförderten Spitzencluster it's OWL, werden derartige Systeme bereits in Innovations- und Transferprojekten entwickelt. Dennoch: Die flächendeckende Entwicklung von ITS ist mehr kühne Vision als nahe Realität. Schon bei mechatronischen Systemen haben viele Unternehmen Probleme, das anspruchsvolle Gesamtsystem aus Produkt und Organisation erfolgreich zu gestalten – verspätete Markteinführungen und zahlreiche Rückrufaktionen belegen dies.

Die **Herausforderungen** bei der Entwicklung mechatronischer Systeme, aber insb. auch der Innovationsdruck im Wettbewerb bedingt durch ITS haben zu einem wachsenden Interesse am Konzept des Systems Engineerings und des Model-Based Systems Engineerings geführt. Unternehmen erhoffen sich durch die ganzheitliche Herangehensweise eine Leistungssteigerung ihrer Prozesse mit dem Ziel einer „1st time quality“. Gerade das MBSE regt dabei die Phantasie an und schürt Hoffnungen auf eine baldige durchgängige Virtualisierung der Produktentstehung. Dabei wird allerdings häufig übersehen, dass das MBSE noch in seinen Kinderschuhen steckt und bislang viele offene Fragen existieren. Das betrifft zunächst natürlich zahlreiche technische Aspekte, an denen mit Hochdruck gearbeitet wird. Vernachlässigt werden aber die zahlreichen sozio-technischen Aspekte des MBSE – das Konzept und seine Reichweite an sich sind unklar. Dadurch kennen zahlreiche potentielle Stakeholder die Nutzenpotentiale von (MB)SE noch nicht, da ihre Arbeiten und Aufgaben bislang nicht ausreichend adressiert werden. Als Folge wird auch die Integration des MBSE in die Produktentstehungsprozesse, die dazugehörige Organisation und die Qualifikation der Anwenderinnen und Anwender nicht angemessen erforscht. Letztlich wird dadurch die Etablierung des Ansatzes als zentrales Paradigma der Produktentstehung ungewollt behindert.

Vor diesem Hintergrund wurden aufbauend auf einer intensiven Analyse der Produktentstehung, der Veränderungen in der Organisation und Arbeitswelt, des Systems Engineering und des modellbasierten Systems Engineerings in Kapitel 2.6 die folgenden drei **Forschungsfragen** formuliert:

- Wie sieht ein Konzept des MBSE aus, das die vielfältigen Facetten der Produktentstehung unterstützt?
- Wie kann MBSE im Sinne des klassischen Systems Engineering in die Produktentstehungsarbeit integriert werden?
- Wie können existierende Methoden und Aufgaben aus der Produktentstehung systematisch mit MBSE unterstützt werden?

Die vorliegende Arbeit beantwortet diese Forschungsfragen durch Bereitstellung eines *Rahmenwerks zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme*. Es umfasst im Kern die folgenden **Elemente**:

- **Beschreibungsmodell des MBSE:** MBSE und seine meist genutzte Sprache SysML werden häufig fälschlicherweise synonym verwendet. Zudem wird das MBSE oft auf die Erstellung der Systemarchitektur reduziert. Deshalb wurden grundlegende Merkmale des MBSE identifiziert und hierauf aufbauend vier MBSE-Konzepte entwickelt. Jedes Konzept ist unterschiedlich ausgestaltet und leistet einen anderen Beitrag zur Unterstützung von Innovation, verbesserter Kommunikation und Koordination in Projekten sowie zu einer virtualisierten Produktentstehung als zentrale Motivation für das Arbeiten mit MBSE. Gerade durch diese Unterschiedlichkeit wird dem Anspruch Rechnung getragen, MBSE als Weiterentwicklung des klassischen Systems Engineering zu betrachten und als weithin akzeptiertes Paradigma der Produktentstehung zu etablieren.
- **Entscheidungsunterstützung zur Auswahl geeigneter MBSE-Konzepte:** Mit den MBSE-Konzepten wurde die Grundlage geschaffen, MBSE in der Produktentstehung einzusetzen. Damit dies auch rational gelingt, wird ihr Einsatz aus vier Sichten geplant. So kann der Einsatz der MBSE-Konzepte in Abhängigkeit des Produktlebenszyklus, des Modellierungszwecks, der konkreten Aufgabe und anhand der Kompetenz der potentiellen Anwenderinnen und Anwender beurteilt werden. Damit wird die Reichweite des Ansatzes dargestellt und die individuelle sowie organisatorische Akzeptanz sichergestellt.
- **Konzept zur Nutzung des Systemmodells in der Produktentstehungsarbeit:** Um das MBSE in der Produktentstehung zu verankern, wurde das Beschreibungsmodell zur Nutzung des Systemmodells entwickelt. In Anlehnung an das Verständnis des Systems Engineerings von HABERFELLNER wurde die Nutzung für technische und für managementnahe Aufgaben forciert. Zudem wurden vier Hauptaufgabenbereiche unterschieden: Modellplanung, Modellierung und Pflege, Methodenkopplung und Anwendung. Das präzisiert die textuelle Definition des MBSE von [INC07]; über die technischen Prozesse hinaus verankert es das MBSE als Weiterentwicklung des Systems Engineerings und erhält gleichzeitig eine organisatorische Komponente. Ebenso wurde die Bedeutung der Trennung von Modell und Sicht für die Anwendung des Systemmodells herausgearbeitet.
- **Werkzeugkasten:** Mit dem flexiblen Vorgehensmodell wurde ein Ansatz geschaffen, mit dem Modellierungsprojekte und Anwendungen auf Basis des Systemmodells geplant und umgesetzt werden können. Kern ist eine 4x4-Matrix, die in die vier Hauptaufgabenbereiche des Konzepts zur Nutzung des Systemmodells unterteilt ist. Für jeden Bereich wurden die relevanten Aufgaben in einem Vorgehensmodell beschrieben. Für den Modellierungszweck der Methodenanwendung wurden zudem erste Methoden und Hilfsmitteln bereitgestellt.

Durch die offene Struktur können weitere Methoden und Hilfsmittel problemlos ergänzt werden. So kann der Werkzeugkasten für das modellbasierte Systems Engineering stetig wachsen.

Das Rahmenwerk wurde in Kapitel 4 vorgestellt. Die initiale und durchgängige **Anwendung** erfolgte im gleichen Kapitel anhand einer Stakeholderanalyse auf Basis des Systemmodells. Den Rahmen für das Anwendungsbeispiel lieferte das Projekt Flyport – ein fiktiver Entwicklungsauftrag für einen Kaffeevollautomaten an einem Flughafen; das Beispiel stammt aus dem „Tool Vendor Project“ der Gesellschaft für Systems Engineering e.V. (GfSE) und wurde im Rahmen der Tätigkeit als wissenschaftlicher Mitarbeiter bei Fraunhofer IEM zusammen mit dem Vorstand der GfSE ins Leben gerufen. Die Anwendung wurde anhand von zwei Modellierungsmethoden demonstriert: CONSENS und SysMod. Zudem wurde sie auf Basis der zwei MBSE-Konzepte *Mechatronic Sketch* und *Mechatronische Systemmodellierung* dargestellt. Für letztere existiert eine prototypische Werkzeugumgebung, die ein COTS-Werkzeug über eine Datenbank mit einer php-Anwendung koppelt. Auf dieser Basis wurden zwei weitere Methoden in Kapitel 5 angewendet – eine Gefährdungsanalyse in Anlehnung an NOHL und eine Komplexitätsanalyse in Anlehnung an MARTI. Die Existenz der MBSE-Konzepte als Ausgangspunkt des Rahmenwerks konnte durch die Analyse von 35 Projekten aus Wissenschaft und Industrie dargestellt werden. In der Bewertung der Arbeit konnte aufgezeigt werden, dass die aus der Problemanalyse abgeleiteten Anforderungen erfüllt wurden.

Die Erfüllung der Anforderungen sollte jedoch nicht darüber hinwegtäuschen, dass auch zukünftiger **Forschungsbedarf** hinsichtlich der Integration des modellbasierten Systems Engineerings in die Produktentstehung mechatronischer Systeme besteht. Insgesamt ordnet sich die Arbeit in die vom Heinz Nixdorf Institut formulierte Vision einer *Neuen Schule des Entwurfs für Intelligente Technische Systeme* ein, die von Fraunhofer IEM als *Advanced Systems Engineering* aufgegriffen wurde. Im Rahmen des *Advanced Systems Engineering* stehen dabei nicht nur rein entwicklungsmethodische Aspekte im Mittelpunkt, vielmehr werden auch die sozio-technischen Fragestellungen der Produktentstehung stärker adressiert. Diese ist, dass nicht nur die technischen Systeme intelligent werden, sondern vielmehr auch die dazugehörigen Organisationsformen, Prozesse und Werkzeuge. Dazu ist das im Rahmen der Arbeit geschaffene Rahmenwerk ein geeigneter Ausgangspunkt. Begriffe wie Engineering 4.0 und Arbeit 4.0 charakterisieren diesen Blickwinkel auf intelligente Systeme.

Konkret lassen sich vor diesem Hintergrund verschiedene organisatorisch-methodische und technisch-methodische Forschungs Herausforderungen darstellen. Aus organisatorischer Sicht sind das bspw.:

- Mit dem Rahmenwerk wurde ein erster Schritt zur Integration des MBSE in die Produktentstehung angeboten. Wie die tatsächliche Einführung für das MBSE ausgestaltet ist, sollte noch erforscht werden. SE und MBSE verändern die Arbeits- und Denkweisen – ohne entsprechende **Einführungsstrategien**, die auch

die besonderen Herausforderungen kleiner und mittelgroßer Unternehmen berücksichtigen, wird der Ansatz sich aber nicht durchsetzen können.

- Die Veränderung der Arbeitsweise im MBSE sollte durch geeignete **Weiterbildungs- und Lehrkonzepte** erreicht werden. Die Gründe hierfür sind einerseits die sehr abstrakten Herangehensweisen des MBSE, aber auch die Notwendigkeit der Unternehmen, Ingenieure zu Systems Engineers fortzubilden – womit sie sich allerdings nicht leicht tun. Erleichtert werden könnte die Veränderung durch Nutzung neuartiger Technologien im Kontext der Weiterbildung und Lehre – die Begriffe Blended Learning und MOOC charakterisieren diese Entwicklung.
- Ein weiterer wichtiger Forschungsaspekt ist die Anwendung von SE und **MBSE in Entwicklungsnetzwerken**, wie sie in Kapitel 2 anhand der Archetypes beschrieben sind. Hier müssen bspw. Konzepte und Prozesse entwickelt werden, die das Zusammenspiel der Akteure im Prozess gestalten.

Aus technischer Sicht sollten die folgenden Herausforderungen angegangen werden:

- Das Rahmenwerk zeigt ein grundsätzliches Vorgehen zur Anwendung des MBSE im Produktentstehungsalltag. Insgesamt wird sich das Vorgehen aber nur durchsetzen können, wenn es mit einem entsprechenden **Werkzeug** vollständig unterstützt werden kann. Dabei geht es jedoch nicht um die reine Implementierung, sondern um die Möglichkeit, die anwendbaren Methoden mit einfachen Hilfsmitteln sukzessive und im Sinne eines Baukastens ergänzen zu können. Dabei spielen gerade Aspekte der Usability eine besondere Rolle
- Der **Zusammenhang zwischen MBSE und PLM** muss adressiert werden. Einerseits ist unklar, wie die im Rahmen des MBSE adressierten Themenstellungen methodisch in eine PLM-Umgebung integriert werden können, da diese meist nur auf hierarchischen Strukturen aufbauen, die entstehenden Systemmodelle i.d.R. aber Netzstrukturen aufweisen. Andererseits sind die beiden Konzepte je nach Standpunkt auch begrifflich sauber voneinander abzugrenzen.
- Im Rahmen der Arbeit wurde häufig die Methode CONSENS verwendet. Sie ist sehr gut für das MBSE-Konzept 2 Mechatronic Sketch geeignet. Ebenso existieren für sie SysML-Profile, wie bspw. für den Enterprise Architect. Bei der **Anwendung von CONSENS** hat sich herausgestellt, dass sowohl Sprache als auch Methode stärker formal beschrieben werden müssen, um eine gute Anwendbarkeit im MBSE-Konzept 3 zu erreichen. Es sind weitaus mehr Stereotypen zu unterscheiden und gleichzeitig die Schritte des Vorgehensmodells sehr viel detailliert zu beschreiben, bspw. durch die Definition dedizierter Diagramme. Ein Beispiel: Das Umfeldmodell könnte bspw. in Diagramme für die Stakeholder, die Nutzungsphase oder Risiken strukturiert werden. Das würde die Zusammenarbeit im Rahmen der Methode erleichtern und flexibler gestalten. Mit den Modellierungsregeln nach [Kai14] wurde hier ein erster Schritt gegangen.

7 Abkürzungsverzeichnis

AADL	Architecture Analysis and Design Language
ADL	Architecture Description Language
ABS	Anti-Blockier-System
AF	Architecture Frameworks
API	Application Programming Interface
A3AO	A3 Architecture Overview
bdd	Blockdefinitionsdiagramm
BMBF	Bundesministerium für Bildung und Forschung
BPMN	Business Process Modeling Notation
bspw.	beispielsweise
bzw.	beziehungsweise
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CAN	Controller Area Network
CaX/CAX	Computer-Aided X
CONSENS	Conceptual Design Specification Technique for the Engineering of Complex Systems
COTS	Commercial-off-the-Shelf
CPS	Cyber-Physical Systems
DDR	Deutsche Demokratische Republik
DfSS	Design for Six Sigma
d.h.	das heißt
DIN	Deutsches Institut für Normung
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
DMADV	Define – Measure – Analyze – Design – Verify

DMAIC	Define – Measure – Analyze – Improve – Control
DNS	Domain Name System
DSML	domänenspezifische Modellierungssprache
EA	Enterprise Architect
E/E	Elektrik/Elektronik
EMF	Eclipse Modeling Framework
EPP	Encrypting Pinpad
e.V.	eingetragener Verein
EW	Eintrittswahrscheinlichkeit
FEM	Fenite Elemente Methode
GfSE	Gesellschaft für Systems Engineering
GRAPHITE	Graphical Tool for Stakeholders‘ interaction
GG	Gefährdungsgrad
GUI	Graphical User Interface
HIL	Hardware in the Loop
IA	Informationsausgabe
IBM	International Business Machines Corporation
i.d.R.	in der Regel
IE	Informationserzeugung
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IEM	Institut Entwurfstechnik Mechatronik
IG	Informationsgewinnung
INCOSE	International Council on Systems Engineering
inkl.	inklusive
insb.	insbesondere
insg.	insgesamt
iPeM	integriertes Produktentstehungsmodell

ISO	International Organization for Standardization
IT	Informationstechnik
ITS	Intelligente Technische Systeme
IV	Informationsverarbeitung
KA	Kaffeeautomat
kg	Kilogramm
KMU	kleine und mittlere Unternehmen
L	Logisches Lösungselement
LITHE	Eigenname
LML	Lifecycle Modeling Language
lt.	laut
MA	Mitarbeiter
MBCD	Model-Based Concept Development
MBD	Model-Based Design
MBSA	Model-Based Systems Architecting
MBSE	Model-Based Systems Engineering
MDM	Multi-Domain-Matrix
MDS	multidimensionale Skalierung
MOEs	Measures of Effectiveness
MOOC	Massive Open Online Courses
NFC	Near Field Communication
OEM	Original Equipment Manufacturer
o.g.	oben genannter
OMG	Object Management Group
OOSEM	Object-Oriented Systems Engineering Method
OPCAT	Object-Process CASE Tool
OPD	Object-Process Diagram
OPL	Object-Process Language

OPM	Object-Process Methodology
OWL	OstWestfalen-Lippe
PDM	Produktdatenmanagement
Php/PHP	Hypertext Preprocessor
PLM	Product Lifecycle Management
PLZ	Produktlebenszyklus
PM	Produktmanager
PoMM	Prozessorientiertes Methodenmodell
QFD	Quality Function Deployment
QM	Qualitätsmanagement
R-A-S-C-I	responsible – accountable – supportive – consulted – informed
RF	Risikofolgen
RFLP	Requirement – Function – Logic – Physics
RoI	Return on Investment
RUP	Rational Unified Process
RQ	Risikoquellen
R&D	Research & Development
SE	Systems Engineering
SE	Systems Engineer
SEBoK	Systems Engineering Body of Knowledge
SECF	Systems Engineering Competencies Framework
SEF	Short Edge First
SEP	Systems Engineering Process
S-I-P-O-C	Supplier-Input-Process-Output-Customer
SM	Systemmodell
SNR	Single Note Recycler / Geldscheineinzahlung
SoS	System-of-Systems
sog.	sogenannt

SoI	System-of-Interest
S/S	Stärken/Schwächen
SW	Software
SQL	Structured Query Language
SysML	Systems Modeling Language
SysMod	Systems Modeling Method
TVP	Tool Vendor Project
u.a.	unter anderem
UE	Untersuchungseinheiten
UML	Unified Modeling Language
USA	United States of America
V _S	Verben des Sammelns
V _B	Verben des Bewertens
V _{SF}	Verben des Schlussfolgerns
v.a.	vor allem
VDI	Verein Deutscher Ingenieure
vgl.	vergleiche
VHDL	Very high speed integrated circuit Hardware Description Language
vs.	versus
VSEs	Very Small Entities
VW	Volkswagen
WB	Wettbewerber
XMI	XML Metadata Interchange
XML	Extensible Markup Language
z.B.	zum Beispiel

8 Literaturverzeichnis

- [Abu12] ABULAWI, J.: Ansatz zur Beherrschung der Komplexität von vernetzten 3D-CAD-Modellen. Dissertation, Fakultät für Maschinenbau, Helmut Schmidt Universität - Universität der Bundeswehr, Hamburg, 2012
- [acatech11] ACATECH – Deutsche Akademie der Technikwissenschaften (Hrsg.): Cyber-Physical Systems – Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion (acatech POSITION). Springer-Verlag, Berlin, 2011
- [Alb10] ALBERS, A.: Five Hypotheses about Engineering Processes and their Consequences – In: Proceedings of the TMCE 2010, Ancona, 2010
- [AB11] ALBERS, A.; BRAUN, A.: A Generalized Framework to Compass and to Support Complex Product Engineering Processes. International Journal of Product Development, Bd. 15 (2011), Nr. 1/2/3, S. 6-25, Genève, 2011
- [AG12] ALBERS, A.; GAUSEMEIER, J.: Von der fachdisziplinorientierten Produktentwicklung zur vorausschauenden und systemorientierten Produktentstehung. In: ANDERL, R.; EIGNER, M.; SENDLER, U.; STARK, R. (Hrsg.): Smart Engineering. Interdisziplinäre Produktentstehung (acatech DISSKUSION), Springer-Verlag, Heidelberg, 2012
- [ALR12] ALBERS, A.; LOHMEYER, Q.; RADIMERSKY, A.: Individuelle und organisatorische Akzeptanz von Methoden des Systems Engineerings. In: MAURER, M.; SCHULZE, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2012
- [Alt12] ALT, O.: Modellbasierte Systementwicklung mit SysML – In der Praxis. Carl Hanser Verlag, München, 2012
- [And15] ANDERL, R.: Strategische Innovationen durch Industrie 4.0 – Vernetzte intelligente Systeme der Zukunft. Keynote. Wissenschaftsforum Intelligente Technische Systeme 2015 - WinTeSys, Paderborn, 2015
- [AES+12] ANDERL, R.; EIGNER, M.; SENDLER, U.; STARK, R.: Smart Engineering. Interdisziplinäre Produktentstehung. Springer-Verlag, Berlin, 2012
- [AHJ+09] ANDERSSON, H.; HERZOG, E.; JOHANSSON, G.; JOHANSSON, O.: Experience from introducing unified modeling language/systems modeling language at Saab Aerosystems. Systems Engineering, Wiley Online Library 2009
- [BDR11] BADKE-SCHAUB, P.; DAALHUIZEN, J.; ROOZENBURG, N.: Towards a Designer-Centred Methodology: Descriptive Considerations and Prescriptive Reflections. In: BIRKHOFFER, H. (Hrsg.): The future of design methodology, Springer-Verlag, London, 2011
- [BG98] BAHILL, A. T.; GISSING, B.: Re-evaluating systems engineering concepts using systems thinking. In: IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), 28/4, 1998, S. 516–527
- [BSD+12] BAJAJ, M.; SCOTT, A.; DEMING, D.; WICKSTROM, G.; SPAIN, M. D.; ZWEMER, D.; PEAK, R.: Maestro - A model-based systems engineering environment for complex electronic systems. INCOSE International Symposium, Rom, 1/2012, S. 1999–2015
- [Bal99] BALZER, H.: Lehrbuch Grundlagen der Informatik. Spektrum-Verlag, 1999.
- [Bät04] BÄTZEL, D.: Methode zur Ermittlung und Bewertung von Strategiealternativen im Kontext Fertigungstechnik. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 141, Paderborn, 2004
- [BPV12] BECKER, J.; PROBANDT, W.; VERING, O.: Grundsätze ordnungsmäßiger Modellierung – Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement. Springer-Verlag, Berlin, 2012

- [Bei70] BEITZ, W.: Systemtechnik im Ingenieurbereich. VDI-Verlag, VDI-Berichte Nr. 174, Düsseldorf, 1970
- [Ber32] BERTALANFFY, L. V.: Theoretische Biologie. Borntraeger Verlag, Berlin, 1932
- [Bin11] BINNER, H.-F.: Business Analyst und Business Process Professional – Neue Berufsbilder in prozessorientierten Organisationen. zfo Zeitschrift Führung + Organisation, 02/2011, Schäffer-Poeschel, Stuttgart, 2011
- [Bir11] BIRKHOFFER, H. (Hrsg.): The future of design methodology. Springer-Verlag, London, 2011
- [BKB+02] BIRKHOFFER, H.; KLOBDERDANZ, H.; BERGER, B.; SAUER, T.: Cleaning up Design Methods – Describing Methods completely and standardised. Proceedings of the International Design Conference – DESIGN 2002, Dubrovnik, 2002
- [BC10] BONE, M.; CLOUTIER, R. J.: The Current State of Model Based Systems Engineering – Results from the OMG Sys'ML Request for Information 2009. In: Proceedings of the 8th Conference on Systems Engineering Research, Hoboken, 2010
- [Bon14] BONNEMA, M. G.: Communication in multidisciplinary systems architecting. 24th CIRP Design Conference 2014, Procedia CIRP, Volume 21, S. 27-33, 2014
- [Bor10] BORCHES, P. D.: A3 architecture overviews – A Tool for Effective Communication in Product Evolution. Dissertation, Department of Engineering Technology (CTW), Universität von Twente, Enschede, 2010
- [BPC+13] BRANSCOMB, J. M.; PAREDIS, C. J.; CHE, J.; JENNINGS, M. J.: Supporting Multidisciplinary Vehicle Analysis Using a Vehicle Reference Architecture Model in SysML. Procedia Computer Science, 2013, S. 79-88
- [BM14] BREISIG, T.; MEYER-TRUELSEN, C.: Organisation und Veränderungsmanagement – Berufsbegleitender internetgestützter Bachelor-Studiengang Business Administration (BA) in kleinen und mittleren Unternehmen. Carl von Ossietzky Universität Oldenburg, Oldenburg, 2014
- [Bri98] BRILL, J. H.: Systems engineering? – A retrospective view. Systems Engineering, 4/1998, S. 258–266
- [Bro11] BROWN, B.: Model-based systems engineering: Revolution or evolution? IBM, Somers, 2011
- [Bro09] BROWNING, T. R.: The many views of a process: Toward a process architecture framework for product development processes. Systems Engineering, 1/2009, S. 69–90
- [BEJ06] BUSCHERMÖHLE, R.; EEKHOOF, H.; JOSKO, B.: SUCCESS Erfolgs- und Misserfolgskriterien bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland, Report VSEK/55/D, BIS-Verlag, Oldenburg, 2006
- [Chr09] CHRISTIANSEN, S.-K.: Methode zur Klassifikation und Entwicklung reifegradbasierter Leistungsbewertungs- und Leistungssteigerungsmodelle. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 264, Paderborn, 2009
- [Clo09] CLOUTIER, R.: Introduction to this Special Edition on Model-based Systems Engineering. INCOSE INSIGHT, 4/2009, S. 7–8
- [Clo13] CLOUTIER, R.: Why is MBSE so Hard? – Perspectives on moving to an MBSA/MBSE Environment, 2013
- [CR06] COHEN, S.; ROUSSEL, J.: Strategisches supply chain management. Springer-Verlag, Berlin, 2006
- [DoDAF11] Department of Defense: DoDAF – Department of Defence Architecture Framework, Version 2.02., 2011

- [DGQ10] Deutsche Gesellschaft für Qualität e.V.: Prozessorientiertes Qualitätsmanagement I – Grundlagen, Frankfurt am Main, 5. Auflage, 2010
- [Dij06] DIJKMAN, R.: Consistency in Multi-Viewpoint Architectural Design. Dissertation, Telematica Institut Fundamental Research, Universität Twente, Enschede, Series Number TI/FRS/17, 2006
- [Dil78] DILLING, H.-J.: Methodisches Rationalisieren von Fertigungsprozessen am Beispiel montagegerechter Produktgestaltung. Dissertation, TU Darmstadt, 1978
- [Dor02] DORI, D.: Object-process methodology – A holistics Systems Paradigm. Springer-Verlag, New York, 2002
- [DR03] DORI, D.; REINHARTZ-BERGER, I.: An OPM Based Metamodel of System Development Process. Israel Institute of Technology, Haifa, 2003
- [Dre07] DRESSLER, S.: Shared Services, Business-process-Outsourcing und offshoring – Die moderne Ausgestaltung des Back-Office – Wege zu Kostensenkung und mehr Effizienz im Unternehmen. Gabler, Wiesbaden, 1. Auflage, 2007
- [Dum11] DUMITRESCU, R.: Eine Entwicklungssystematik zur Integration kognitiver Funktionen in fortgeschrittene mechatronische Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 286, Paderborn, 2011
- [DYY+13] DUNFORD, C. N.; YEARWORTH, M.; YORK, D. M.; GODFREY, P.: A view of Systems Practice: Enabling quality in design. *Systems Engineering*, 2/2013, S. 134-151
- [ERZ14] EIGNER, M.; ROUBANOV, D.; ZAFIROV, R.: Modellbasierte virtuelle Produktentwicklung. Springer-Verlag, Berlin, 2014
- [EG12] ELM, J. P.; GOLDENSON, D.: The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey. Software Engineering Institute, Carnegie Mellon, 2012
- [Est08] ESTEFAN, J. A.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. In: INCOSE MBSE Initiative, California, 2008
- [Fra06] FRANK, U.: Spezifikationstechnik zur Beschreibung der Prinzipiellösung selbstoptimierender Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 175, Paderborn, 2006
- [Fra95] FRANTZ, W. F.: The Impact of Systems Engineering on Quality and Schedule. INCOSE International Symposium, 5, 1995
- [Fre84] FREEMAN, E.: Strategic Management: A Stakeholder Approach. Pitman, Boston, 1984
- [FMS08] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A Practical Guide to SysML – The Systems Modeling Language. Elsevier, Amsterdam, 2008
- [FMS11] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A Practical Guide to SysML – The Systems Modeling Language. Morgan Kaufmann, Waltham, 2011
- [FMS12] FRIEDENTHAL, S.; MOORE, A.; STEINER, R.: A Practical Guide to SysML – The systems Modeling Language. Morgan Kaufmann, Waltham, 2. Auflage, 2012
- [Fro04] FROST, X.: Aufbau- und Ablauforganisation. In: SCHREYÖGG, G. (Hrsg.): Handwörterbuch Unternehmensführung und Organisation, Schäffer-Poeschel, ; Band 2, Stuttgart, 2004
- [Gaa10] GAASBEEK VAN, J. R.: Model-Based Systems Engineering (MBSE). INCOSE LA Mini-Conferencne, Loyola Marymoint University, 2010
- [Gau10a] GAUSEMEIER, J. (Hrsg.): Frühzeitige Zuverlässigkeitsanalyse mechatronischer Systeme. Carl Hanser Verlag, München, Wien, 2010

- [Gau10b] GAUSEMEIER, J.: Maschinenbau braucht Systems Engineering – In: Konstruktion, Ausgabe 12/2010, Springer VDI-Verlag, Düsseldorf, 2010
- [GAC+13] GAUSEMEIER, J.; ANACKER, H.; CZAJA, A.; WABMANN, H.; DUMITRESCU, R.: Auf dem Weg zu intelligenten technischen Systemen. In: GAUSEMEIER, J.; DUMITRESCU, R.; RAMMIG, F.-J.; SCHÄFER, W.; TRÄCHTLER, A. (Hrsg.): 9. Paderborner Workshop Entwurf mechatronischer Systeme, 18.-19. April., HNI-Verlagsschriftenreihe, Band 310, Paderborn, 2013
- [GDS+13a] GAUSEMEIER, J.; DUMITRESCU, R.; STEFFEN, D.; TSCHIRNER, C.; CZAJA, A.; WIEDERKEHR, O.: Systems Engineering in der industriellen Praxis. In: GAUSEMEIER, J.; DUMITRESCU, R.; RAMMIG, F.-J.; SCHÄFER, W.; TRÄCHTLER, A. (Hrsg.): 9. Paderborner Workshop Entwurf mechatronischer Systeme, 18.-19. April., HNI-Verlagsschriftenreihe, Band 310, Paderborn, 2013
- [GDS+13b] GAUSEMEIER, J.; DUMITRESCU, R.; STEFFEN, D.; TSCHIRNER, C.; CZAJA, A.; WIEDERKEHR, O.: Systems Engineering in der industriellen Praxis, Paderborn, 2013
- [GFD+08] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus. In: Konstruktion, Ausgabe 8/2008 und 9/2008, Springer VDI-Verlag, Düsseldorf, 2008
- [GFD+09] GAUSEMEIER, J.; FRANK, U.; DONOTH, J.; KAHL, S.: Specification technique for the description of self-optimizing mechatronic systems. In: Research in Engineering Design, Vol. 20, Number 4, November 2009, Springer-Verlag, London, 2009
- [GHK+06] GAUSEMEIER, J.; HAHN, A.; KESPOHL, H. D.; SEIFERT, L.: Vernetzte Produktentwicklung – Der erfolgreiche Weg zum Global Engineering Networking. Carl Hanser Verlag, München, 2006
- [GKP+10] GAUSEMEIER, J.; KAISER, L.; POOK, S.; NYBEN, A.; TERFLOTH, A.: Rechnerunterstützte Modellierung der Prinziplösung mechatronischer Systeme. In: GAUSEMEIER, J.; RAMMIG, F.-J.; SCHÄFER, W.; TRÄCHTLER, A. (Hrsg.): 7. Paderborner Workshop Entwurf mechatronischer Systeme, HNI-Verlagsschriftenreihe, Band 272, Paderborn, 2010
- [GLL12] GAUSEMEIER, J.; LANZA, G.; LINDEMANN, U.: Produkte und Produktionssysteme integrativ konzipieren – Modellbildung und Analyse in der frühen Phase der Produktentstehung. Carl Hanser Verlag, München, 2012
- [GLR+00] GAUSEMEIER, J.; LINDEMANN, U.; REINHART, G.; WIENDAHL, H.: Kooperatives Produktengineering – Ein neues Selbstverständnis des ingenieurmässigen Wirkens. HNI-Verlagsschriftenreihe, Band 79, Paderborn, 2000
- [GPW14] GAUSEMEIER, J.; PLASS, C.; WENZELMANN, C.: Zukunftsorientierte Unternehmensgestaltung – Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. Carl Hanser Verlag, München, 2. Auflage, 2014
- [GTD13] GAUSEMEIER, J.; TSCHIRNER, C.; DUMITRESCU, R.: Der Weg zu Intelligenten Technischen Systemen – Spitzencluster it's OWL – Mit Intelligenten Technischen Systemen an die Spitze. Industriemanagement, 29/2013, S. 49–52
- [GAP+93] GRABOWSKI, H.; ANDERL, R.; POLLY, A.; WARNECKE, H.J.: Integriertes Produktmodell. Beuth, Berlin, 1993
- [Gre03] GREIFFENBERG, S.: Methoden als Theorien der Wirtschaftsinformatik. In: UHR, W.; ESSWEIN, W.; SCHOOP, E. (Hrsg.): Wirtschaftsinformatik 2003/Band II, Physica-Verlag HD, Heidelberg, 2003
- [GTH16] GREINERT, M.; TSCHIRNER, C.; HOLTMANN, J.: Anwendung von Methoden der Produktentstehung auf Basis des Systemmodells mechatronischer Systeme. In: SCHULZE, S.-O.; TSCHIRNER, C.; KAFFENBERGER, R.; ACKVA, S. (Hrsg.): Tag des Systems Engineering 2016, Carl Hanser Verlag, München, 2016
- [GD11] GROBSHTEIN, Y.; DORI, D.: Generating SysML views from an OPM model – Design and evaluation. Systems Engineering, 3/2011, S. 327–340

- [Haa16] HAASIS, S.: Systems Engineering for future mobility. REConf, 2016
- [HWF+12] HABERFELLNER, R.; WECK, O. L. DE; FRICKE, E.; VÖSSNER, S.: Systems Engineering – Grundlagen und Anwendungen. Orell Füssli, Zürich, 12. Auflage, 2012
- [Han56] HANSEN, F.: Konstruktionssystematik. VEB-Verlag Technik, Berlin, 1956
- [Han65] HANSEN, F.: Konstruktionssystematik Berlin. VEB-Verlag Technik, Berlin, 7. Auflage, 1965
- [Hap-o.J.] HAPERSCHIEDT, S.: Seminar Modellierung und Metamodellierung: Syntax und Semantik. Heinz Nixdorf Institut, o.J.
- [HTF96] HARASHIMA, F.; TOMIZUKA, M.; FUKUDA, T.: Mechatronics – "What Is It, Why and How?" An Editorial. IEEE/ASME Transactions on Mechatronics, Volume 1, Nr. 1, 1996
- [Hau11] HAUBE, M.: "Are we there yet?" Assessing Quality in Model Based Systems Engineering. INCOSE, San Diego, USA, 2011
- [HB15] HAVEMAN, S. P.; BONNEMA, M. G., 2015: A conceptual model to support communication of systems modeling and simulation activities. In: IEEE Systems Council (Hrsg.): 2015 IEEE International Systems Conference (SysCon 2015) Proceedings. Vancouver, British Columbia, Canada, April, 13-16, 2015
- [Heh12] HEHENBERGER, P.: Advances in Model-Based Mechatronic Design. Habilitationsschrift, Johannes Kepler Universität (JKU) Linz Trauner Verlag, Linz 2012
- [Hit07] HITCHINS, D. K.: Systems engineering – A 21st Century Systems Methodology. John Wiley, West Sussex, England, 2007
- [HHD+12] HOCHWALLNER, M.; HÖRL, M.; DIERNEDER, S.; SCHEIDL, R.: Some Aspects of SysML Application in the Reverse Engineering of Mechatronic Systems, 2012
- [Hof08] HOFFMANN, H.-P.: Harmony SE: A SysML Based Systems Engineering Process. Innovation 2008, 2008
- [Hon04] HONOUR, E.: Understanding the Value of Systems Engineering. In: International Council on Systems Engineering (INCOSE) (Hrsg.): Proceedings of the 14th Annual INCOSE International Symposium, 2004
- [Hon05] HONOUR, E. C.: A Practical Program of Research to Measure Systems Engineering Return on Investment (SE-ROI). 2005
- [Hon13] HONOUR, E. C.: Systems engineering return on investment, Dissertation, Defence and Systems Institute, School of Electrical and Information Engineering, University of South Australia, 2013
- [Hop15] HOPPE, M.: Schon wieder Requirements Management oder wie spezifiziere ich das Automationssystem eines U-Bootes? In: Schulze, S.-O. (Hrsg.): Tag des Systems Engineering 2015, Carl Hanser Verlag, München, 2015
- [IDC13] IDC Business Media: Schlechtes Dokumentenmanagement frisst bis zu 15 Prozent des Umsatzes, 2013
- [INC10] INCOSE UK: Systems Engineering Competency Framework.
- [IEEE1471] Institute of Electrical and Electronics Engineers (IEEE): IEEE1471 - Recommended practice for architectural description for software-intensive systems, New York, 2000
- [INC07] International Council on Systems Engineering (INCOSE): Systems Engineering Vision 2020, unter: http://www.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf
- [INC08] International Council on Systems Engineering (INCOSE) (Hrsg.): INCOSE MBSE Initiative. INCOSE MBSE Focus Group, California, 2008

- [INC12] International Council on Systems Engineering (INCOSE) (Hrsg.): Systems Engineering Handbuch – Deutsche Übersetzung. Carl Hanser Verlag, 2012
- [INC15] International Council on Systems Engineering (INCOSE): International Council of Systems Engineering – Systems Engineering Vision 2025, <http://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf?sfvrsn=4>, 2015
- [ISO15288] ISO/IEC 15288:2008: Systems and software engineering — System life cycle processes, 2008
- [ISO29110] ISO/IEC 29110-4-1:2011: Software engineering — Lifecycle profiles for Very Small Entities (VSEs) – Part 4-1: Profile specifications: Generic profile group, 2011
- [ISO42010] ISO/IEC/IEEE 42010:2011: Systems and software engineering — Architecture Description, 2011
- [IGB+15] IWANEK, P.; GAUSEMEIER, J.; BANSMANN, M.; DUMITRESCU, R.: Integration of Intelligent Features by Model-Based Systems Engineering. In: Proceedings of 18th ISERD International Conference. Tokyo, Japan, Nov. 2015
- [IKD+13] IWANEK, P.; KAISER, L.; DUMITRESCU, R.; NYBEN, A.: Fachdisziplinübergreifende Systemmodellierung mechatronischer Systeme mit SysML und CONSENS. In: MAURER, M.; SCHULZE, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2013
- [Jän07] JÄNSCH, J.: Akzeptanz und Anwendung von Konstruktionsmethoden im industriellen Einsatz. Fortschritt-Berichte. VDI Verlag, Nr. 369, Düsseldorf, 2007
- [Kac13] KACHLER, J.: Erfolgreiches Management grüner Entwicklungsprojekte. Dissertation, Technische Universität Berlin, Berlin, 2013
- [Kai14] KAISER, L.: Rahmenwerk zur Modellierung einer plausiblen Systemstruktur mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 327, Paderborn, 2014
- [Kal98] KALLMEYER, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 42, Paderborn, 1998
- [Kas10] KASSER, J. E.: Seven systems engineering myths and the corresponding realities – Proceedings of the Systems Engineering Test and Evaluation Conference, Adelaide, Australia, 2010
- [Kle03] KLEINER, S.: Föderatives Informationsmodell zur Systemintegration für die Entwicklung mechatronischer Produkte. Shaker, , Band 15, Aachen, 2003
- [KB14] KLEINER, S.; BLUMÖR, A.: Model Based Systems Engineering - Einführung und Anwendung der modellbasierten Arbeitsweise in der Maschinenentwicklung. In: SCHULZE, S.-O.; MAURER, M. (Hrsg.): Tag des Systems Engineering 2014, Carl Hanser Verlag, München, 2014
- [Kno14] KNOKE, M.: Handlungsbedarf für Systems Engineering aus Anwendersicht. Fachgruppe Systems Engineering, Vlotho, 2014
- [Koe13] KOENIGS, S.: Konzeption und Realisierung einer Methode zur templategestützten Systementwicklung. Dissertation, Fakultät V – Verkehrs- und Maschinensysteme, Technische Universität Berlin, Fraunhofer-Verlag, Berlin, 2013
- [Kol85] KOLLER, R.: Konstruktionslehre für den Maschinenbau – Grundlagen, Arbeitsschritte, Prinziplösungen. Springer-Verlag, Berlin, 1985
- [Kru95] KRUCHTEN, P.: Architectural Blueprints – The “4+1” View Model of Software Architecture. IEEE Software, 6/1995, S. 42–50
- [Lau10] LAUER, W.: Gezielte Integration von Produktmodellen in den Entwicklungsprozess. In: MEERKAMM, H.; HENRICH, A.; JABLONSKI, S.; KRUMAR, H.; LINDEMANN, U.; RIEG, F. (Hrsg.): Flexible Prozessunterstützung in der Produktentwicklung: Prozesse – Daten – Na-

- vigation, Bayerischer Forschungsverbund FORFLOW für Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung, 2010
- [Leh14] LEHNER, M.: Verfahren zur Entwicklung geschäftsmodell-orientierter Diversifikationsstrategien. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe; Band 324, Paderborn 2014
- [LML13] LIFECYCLEMODELINGLANGUAGE: Lifecycle Modeling Language (LML) Specification, http://www.lifecyclemodeling.org/spec/LML_Specification_1_0.pdf, 2013
- [Lin09] LINDEMANN, U.: Methodische Entwicklung technischer Produkte – Methoden flexibel und situationsgerecht anwenden. Springer-Verlag, Berlin, 3. Auflage, 2009
- [LMB09] LINDEMANN, U.; MAURER, M.; BRAUN, T.: Structural complexity management – An approach for the field of product design. Springer-Verlag, Berlin, 2009
- [LG04] LIU, H.; GLUCH, P. D.: Conceptual modeling with the object process methodology in software architecture. Journal of Computing Sciences in Colleges, Januar 2004, 2004
- [MR09] MAIER, M. W.; RECHTIN, E.: The art of systems architecting. CRC Press, Boca Raton, 3rd ed, 2009
- [Mar15] MARKGRAF, D.: Gabler Wirtschaftslexikon – Stichwort: Produktlebenszyklus, <http://wirtschaftslexikon.gabler.de/Definition/produktlebenszyklus.html#definition>, abgerufen: 17. März 2016
- [Mar07] MARTI, M.: Complexity Management – Optimizing Product Architecture of Industrial Products. PhD Thesis, St. Gallen, CH, DUV, [S.I.], 2007
- [Mar04] MARTIN, J. N.: The Seven Samurai of Systems Engineering: Dealing with the Complexity of 7 Interrelated Systems. INCOSE - Proceedings of the 14th Annual International Symposium, 2004
- [MSL+15] MATTHIESEN, S.; SCHMIDT, S.; LUDWIG, J.; JOHMANN, S.: Iteratives Vorgehen in räumlich getrennten mechatronischen Entwicklungsteams – Das Wechselspiel von Synthese und testbasierter Analyse. In: VDI Mechatroniktagung 2015, 2015
- [MHJ+10] MEERKAMM, H.; HENRICH, A.; JABLONSKI, S.; KRCDMAR, H.; LINDEMANN, U.; RIEG, F. (Hrsg.): Flexible Prozessunterstützung in der Produktentwicklung: Prozesse – Daten – Navigation, Bayerischer Forschungsverbund FORFLOW für Prozess- und Workflowunterstützung zur Planung und Steuerung der Abläufe in der Produktentwicklung, 2010
- [MCS+14] MEISSNER, H.; CADET, M.; STEPHAN, N.; BOHR, C.: Model-Based Development Process of Cybertronic Products and Production Systems. Advanced Materials Research Vol. 1018 – Proceedings of the WGP Congress 2014 – Progress in Production Engineering, (2014): S. 539-546
- [Möh12] MÖHRINGER, S.: Systems Engineering im Mittelstand – ein flexibles Modell zur Rollverteilung – Praxiserfahrungen aus Entwicklungsprojekten. In: MAURER, M.; SCHULZE, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2012
- [MHM10] MOODY, D. L.; HEYMANS, P.; MATULEVIČIUS, R.: Visual Syntax Does Matter – Improving the Cognitive Effectiveness of the i* Visual Notation. Requirements Engineering, 2/2010, S. 141–175
- [Mül90] MÜLLER, J.: Arbeitsmethoden der Technikwissenschaften – Systematik, Heuristik, Kreativität. Springer-Verlag, Berlin, 1990
- [ML11] MÜLLER-STEWENS, G.; LECHNER, C.: Strategisches Management – Wie strategische Initiativen zum Wandel führen. Schäffer-Poeschel Verlag, Stuttgart, 4. Auflage, 2011
- [Mur12] MURRAY, J.: Model Based Systems Engineering (MBSE) Media Study, 2012

- [Nat14] NATTERMANN, R.: Integrierte, virtuelle Entwicklung adaptronischer Systeme. Dissertation, Fachbereich Maschinenbau, Technische Universität Darmstadt, Shaker-Verlag, Aachen, 2014
- [NA13] Nattermann, R.; Anderl, R.: The W-Model – Using Systems Engineering for Adaptronics. In: Conference on Systems Engineering Research (CSER 13), Elsevier B.V., Atlanta, GA, 2013
- [NT88] NOHL, J.; THIEMEKE, H.: Systematik zur Durchführung von Gefährdungsanalysen. Verlag für neue Wissenschaft, 1988
- [NBB+08] NOLAN, B.; BROWN, B.; BALMELL, L. BOHN, T.; WAHLI, U.: Model Driven Systems Development with Rational Products. Redbooks, 2008
- [OMG12] OBJECT MANAGEMENT GROUP: OMG Systems Modeling Language (OMG SysML™) – Version 1.3, <http://www.omg.org/spec/SysML/1.3/>, 2012
- [oV91] oV: Die Einführung von CAD/CAM verlangt aktives Management – Gründe für das Scheitern von CAD-Projekte. Computerwoche, 22.03.1991, 1991
- [oV15] oV: Model-Based Systems Engineering Overview – MBSE Synopsis. online: <http://mbse.works/mbse-overview/>, 2015
- [PBF+05] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung. Springer-Verlag, Berlin, 6. Auflage, 2005
- [PBF+07] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Konstruktionslehre – Grundlagen erfolgreicher Produktentwicklung Methoden und Anwendung. Springer-Verlag, Berlin, 7. Auflage, 2007
- [PBF+13] PAHL, G.; BEITZ, W.; FELDHUSEN, J.; GROTE, K.-H.: Konstruktionslehre – Methoden und Anwendung erfolgreicher Produktentwicklung. Springer-Verlag, Heidelberg, 8., vollst. überarb. Auflage, 2013
- [PDH11] PARNELL, G.S.; DRISCOLL, P.J.; HENDERSON, D.L. (Hrsg.): Decision Making in Systems Engineering and Management. John Wiley, New York, 2nd ed., 2011
- [PR14] PATZAK, G.; RATTAY, G.: Projektmanagement – Projekte, Projektportfolios, Programme und projektorientierte Unternehmen. Linde Verlag Ges.m.b.H, Wien, 6. wesentlich erw. und aktualisierte Auflage, 2014
- [Pau05] PAULUKUHN, L.: Typologisierung von Entwicklungsprojekten im Maschinenbau. Dissertation, Fakultät für Maschinenwesen, Rheinisch-Westfälische Technische Hochschule Aachen, Shaker Verlag, Aachen, 2005
- [PM06] PAWELLEK, G.; MARTENS, I.: Methodenapplication forcieren. Optimierung der Instandhaltung und Ersatzteillogistik. In: Der Betriebsleiter, 7-8/2006, S. 26-27, 2006
- [PH12] PEARCE, P.; Hause M.: ISO-15288, OOSEM and Model-Based Submarine Design. Deep Blue Tech Pty, 2012
- [Pet13] PETRE, M.: UML in practice. In: ICSE 2013
- [Pon07] PONN, J.: Situative Unterstützung der methodischen Konzeptentwicklung technischer Produkte. Dissertation, Fakultät für Maschinenwesen, Technische Universität München, Hut, München, 1. Auflage, 2007
- [PMI13] PROJECT MANAGEMENT INSTITUTE: A Guide to the Project Management Body of Knowledge (PMBOK Guide) (PMBOK Guide). Project Management Institute, 2013
- [Pul04] PULM, U.: Eine systemtheoretische Betrachtung der Produktentwicklung. Dissertation, Produktentwicklung München, Technische Universität München, r, 2004
- [Qui04] QUINT, W.: Modellierung und Management von Workflows. Shaker, Aachen, 2004

- [Ram08] RAMOS, A. L.: Model-based Systems Engineering: A System for Traffic & Environment. Dissertation, Institut für Wirtschaft, Management, und industrielles Engineering, Universität Aveiro, Aveiro, 2008
- [Ram11] RAMOS, A. L.: Model-Based Systems Engineering: A System for Traffic & Environment. Dissertation, Universität Aveiro, Aveiro, 2011
- [RFB12a] RAMOS, A. L.; FERREIRA, J. V.; BARCELO, J.: Model-Based Systems Engineering: An Emerging Approach for Modern Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 1/2012, S. 101-111
- [RFB12b] RAMOS, A. L.; FERREIRA, J. V.; BARCELO, J.: Modeling & Simulation for Intelligent Transportation Systems. International Journal of Modeling and Optimization, 3/2012, S. 274-279
- [RFB13] RAMOS, A. L.; FERREIRA, J. V.; BARCELO, J.: LITHE: An Agile Methodology for Human-Centric Model-Based Systems Engineering. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 3/2013, S. 504-521
- [RFB10] RAMOS, A. L.; FERREIRA, J. V.; BARCELÓ, J.: Revisiting the Similar process to engineer the contemporary systems. Journal of Systems Science and Systems Engineering, 3/2010, S. 321-350
- [RR09] RHODES, D.; ROSS, A. M.: Concept Design and Tradespace Exploration. Systems Engineering Advancement Research Initiative (SEArI), 2009
- [RWD14] ROBINSON, K.; WAITE, M.; DO, Q.: Introduction to the Model-Based Conceptual Design Special Issue – INSIGHT Special Feature. INCOSE INSIGHT, 2014
- [Roe11] ROELOFSEN, J. M. K.: Situationspezifische Planung von Produktentwicklungsprozessen. Verlag Dr. Hut, München, 1. Auflage, 2011
- [Rop75] ROPOHL, G.: Systemtechnik – Grundlagen und Anwendung. Carl Hanser Verlag, München, 1975
- [Rop09] ROPOHL, G.: Allgemeine Technologie – Eine Systemtheorie der Technik. Universitätsverlag Karlsruhe, Karlsruhe, 3. Auflage, 2009
- [Rot82] ROTH, K.: Konstruieren mit Konstruktionskatalogen. Springer-Verlag, Berlin, Heidelberg, 1982
- [Rot12] ROTH, S.: Architekturrahmenwerk in der modellbasierten Systementwicklung. In: MAURER, M.; SCHULZE, S.-O. (Hrsg.): Tag des Systems Engineering. Carl Hanser Verlag, München, 2012
- [SF14] SAMPSON, M.; FRIEDENTHAL, S.: Role of MBCD in the INCOSE MBSE Initiative. INCOSE INSIGHT, 2014
- [SBB02] SAUER, T.; BERGER, B.; BIRKHOFFER, H.: Beschreibung Design Methods According to the Specific Needs of Users. 2002
- [Sau12] SAUTER, C.: Ein Beitrag zur Integration von Wikis und Social Tagging in die Produktentstehung. Dissertation, IPEK - Institut für Produktentwicklung (Karlsruhe), Forschungsberichte, Band. 54, Karlsruhe, 2012
- [SRC10] SCHALLES, C.; REBSTOCK, M.; CREAGH, J.: Ein generischer Ansatz zur Messung der Benutzerfreundlichkeit von Modellierungssprachen. In: ENGELS, G. (Hrsg.): Modellierung 2010, GI, Bonn, 2010
- [Sch80] SCHEIN, E. H.: Organisationspsychologie. Gabler Verlag, Wiesbaden, 1980
- [Sch10] SCHLICK, C.: Arbeitswissenschaft. Springer-Verlag, Berlin, Heidelberg, 3. vollst. überarb. und erw. Auflage, 2010
- [SKD+13] SCHMITT, N.; KAISER, L.; DUMITRESCU, R.; HOFMANN, M.: Von der Anforderungserfassung bis zur Funktionsstruktur – Ein Systems Engineering-Vorgehen für die industrielle Praxis.

- In: MAURER, M.; SCHULZE, S.-O. (Hrsg.): Tag des Systems Engineering, Carl Hanser Verlag, München, 2013
- [Sch01] SCHMITT, R.: Unternehmensübergreifender Engineering Workflow – Verteilte Produktentwicklung auf der Grundlage eines parameterbasierten Daten- und Prozessmanagements. Papierflieger, Clausthal-Zellerfeld, 1. Auflage, 2001
- [Sch03] SCHREYÖGG, G.: Organisation – Grundlagen moderner Organisationsgestaltung ; mit Fallstudien. Gabler, Wiesbaden, 4., vollst. überarb. und erw. Auflage, 2003
- [Sch05] SCHUH, G.: Produktkomplexität managen – Strategien – Methoden – Tools. Carl Hanser Verlag, München, Wien, 2., überarb. und erw. Auflage, 2005
- [SEBoK15] SEBOK: Systems Engineering Body of Knowledge – <http://www.sebokwiki.org>, 2015
- [Sen13] SENDLER, U.: Industrie 4.0 – Beherrschung der industriellen Komplexität mit SysLM. Springer-Verlag, Berlin, Heidelberg, 2013
- [Ser94] SERVATIUS, H.-G.: Reengineering-Programme umsetzen – Von erstarrten Strukturen zu fließenden Prozessen. Schäffer-Poeschel Verlag, Stuttgart, 1994
- [She96] SHEARD, S. A.: Twelve Systems Engineering Roles. In: International Council on Systems Engineering (INCOSE) (Hrsg.): Proceedings of the 6th INCOSE Annual International Symposium, Boston, Massachusetts, USA, 1996
- [SSG09] SMERLINSKI, M.; STEPHAN, M.; GUNDLACH, C.: Innovationsmanagement in hessischen Unternehmen: Eine empirische Untersuchung zur Praxis in klein- und mittelständischen Unternehmen. Discussion Papers on Strategy and Innovation, No. 09-01, 2009
- [SER04] SOSA, M. E.; EPPINGER, S. D.; ROWLES, C. M.: The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. Management Science, 12/2004, S. 1674-1689
- [SD16] SPATH, D.; DANGELMAIER, M.: Produktentwicklung Quo Vadis. In: LINDEMANN, U. (Hrsg.): Handbuch Produktentwicklung, Carl Hanser Verlag, München, 2016
- [Sta73] STACHOWIAK, H.: Allgemeine Modelltheorie. Springer-Verlag, Wien, 1973
- [SVE+07] STAHL, T.; VÖLTER, M.; EFFTINGE, S.; HAASE, A.: Modellgetriebene Softwareentwicklung – Techniken, Engineering, Management. Dpunkt-Verlag, Heidelberg, 2., aktualisierte und erw. Auflage, 2007
- [SSG14] STEFFEN, D.; SCHULZE, S.-O.; GAUPP, F.: Opportunity: Systems Engineering – Produktentwicklung erfindet sich neu, UNITY, 2014
- [SB09] STETTER, R.; BLUM, T.: Verschläft der Deutsche Maschinenbau seine Chancen? – Mechatronische Möglichkeiten im internationalen Wettbewerb. 2009
- [SL05] STETTER, R.; LINDEMANN, U.: The Transfer of Methods into Industry. In: CLARKSON, J.; ECKERT, C. (Hrsg.): Design Process Improvement. A Review of Current Practice. Springer-Verlag, London, 2005
- [SMM+12] STIEGLER, A.; MALETZ, M.; MROTZEK, M.; WECK, T.: Generierung eines multiperspektiven Systemmodells in der automobilen Antriebsstrangentwicklung- Herausforderungen und Erfahrungen. In: MAURER, M.; SCHULZE, S.-O. (Hrsg.): Tag des Systems Engineering 2012, Carl Hanser Verlag, München, 2012
- [Sto12a] STOEWER, H.: Model based Engineering (MBSE) and Model based Systems Engineering (MBSE) – Are we approaching the Golden Age of Simulation?. SECESA Conference, Lisbon, 2012
- [Sto12b] STOEWER, H.: Model based Systems Engineering (MBSE) – Hip or Hype?. CESUN Conference, Delft, 2012

- [Taf06] TAFRESCHI, A.: Zur Benennung und Kategorisierung alltäglicher Gegenstände – Onomasiologie, Semasiologie und kognitive Semantik. Dissertation, Universität Kassel, Kassel Univ. Press, Kassel, 2006
- [Tan89] TANAKA, M.: Cost planning and control systems in the design phase of a new product. In: MONDEN, Y.; SAKURAI, M. (Hrsg.): Japanese management accounting: a world class approach to profit management, Productivity Press, Portland, 1989
- [Taz10] TAZIR, N.: Systems Engineering – Viel diskutiert, selten wirklich angewendet. ProduktDatenJournal, Nr. 2/2010
- [TW10] THOM, N.; WENGER, A. P.: Die optimale Organisationsform – Grundlagen und Handlungsanleitung. Gabler, Wiesbaden, 1. Auflage, 2010
- [Tie10] TIEMEYER, E.: Handbuch IT-Projektmanagement – Vorgehensmodelle, Managementinstrumente, Good Practices. Carl Hanser Verlag, München, 1. Auflage, 2010
- [TK08] TOUTENBURG, H.; KNÖFEL, P.: Six Sigma – Methoden und Statistik für die Praxis. Springer-Verlag, Berlin, Heidelberg, 2008
- [Tro10] TROLL, A.: Systematische Einbindung der CAE-Werkzeuge in den Entwicklungsprozess. In: MEERKAMM, H.; HENRICH, A.; JABLOWSKI, S.; KRCCMAR, H.; LINDEMANN, U.; RIEG, F. (Hrsg.): Flexible Prozessunterstützung in der Produktentwicklung: Prozesse – Daten – Navigation, Abschlussbericht, 2010
- [TA15a] TSCHIRNER, C.; ACKVA, S.: ISO/IEC 15288 und ISO/IEC 29110 – Prozesse für das Systems Engineering. develop3 systems engineering, KONRADIN, 02/2015, 2015
- [TA15b] TSCHIRNER, C.; ACKVA, S.: Model-Based Systems Engineering. develop3 systems engineering, KONRADIN, 03/2015, 2015
- [TA16] TSCHIRNER, C.; ACKVA, S.: Systems Engineering Return on Investment - oder: Wie viel SE ist notwendig? develop3 systems engineering, KONRADIN, 01/2016, 2016
- [TBD+15] TSCHIRNER, C.; BRETZ, L.; DUMITRESCU, R.; GAUSEMEIER, J.: Applying Model-Based Systems Engineering for Product Engineering Management – Concepts for Industrial Application. In: IEEE SYSTEMS COUNCIL (Hrsg.): 2015 IEEE International Systems Conference (SysCon 2015) Proceedings. Vancouver, British Columbia, Canada, April, 13-16, 2015
- [TKD+14] TSCHIRNER, C.; KAISER, L.; DUMITRESCU, R.; GAUSEMEIER, J.: Collaboration in Model-Based Systems Engineering based on Application Scenarios. In: Proceedings of NordDesign 2014 Conference, Aalto, 2014
- [TS15] TSCHIRNER, C.; SCHULZE, S.-O.: Systems Engineering. develop3 systems engineering, KONRADIN, 01/2015, 2015
- [VDI2206] VDI-RICHTLINIE: VDI 2206, Entwicklungsmethodik für mechatronische Systeme, 2004
- [VDI2211] VDI-RICHTLINIE: VDI 2211, Blatt 2, Informationsverarbeitung in der Produktentwicklung, Berechnungen in der Konstruktion, 2003
- [VDI2219] VDI-RICHTLINIE: VDI 2219, Informationsverarbeitung in der Produktentwicklung - Einführung und Wirtschaftlichkeit von EDM/PDM-Systemen, 2002
- [VDI2221] VDI-RICHTLINIE: VDI 2221, Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte, 1993
- [War13] WARSCHAT, J. (Hrsg.): Transfer von Forschungsergebnissen in die industrielle Praxis – Konzepte, Beispiele, Handlungsempfehlungen; Zusammenfassung der Ergebnisse des Projektes Entwicklung von Transfermechanismen für die effiziente und nachhaltige Verbreitung von Forschungsergebnissen in die industrielle Praxis am Beispiel Mechatronik. Fraunhofer-Verlag, Stuttgart, 2013
- [Wei06] WEILKIENS, T.: Systems Engineering mit SysML/UML – Modellierung, Analyse, Design. Dpunkt-Verlag, Heidelberg, 2006

-
- [Wei08] WEILKIENS, T.: Systems Engineering mit SysML/UML – Modellierung, Analyse, Design. Dpunkt-Verlag, Heidelberg, 2. Auflage, 2008
- [Wei15] WEILKIENS, T.: SYSMOD – The Systems Modeling Toolbox – Pragmatic MBSE with SysML, 2015
- [Wit07] WITTENSTEIN, A.-K.: Bedarfssynchrone Leistungsverfügbarkeit in der kundenspezifischen Produktentwicklung. Dissertation, Institut für industrielle Fertigung und Fabrikbetrieb, Universität Stuttgart, Fraunhofer Institut für Produktionstechnik und Automatisierung, Stuttgart, Jost-Jetter Verlag, Heimsheim, 2007
- [Zac87] ZACHMAN, J. A.: A Framework for Information Systems Architecture. IBM System Journal, 26,(3), 1987, S. 276–297
- [Zap10] ZAPF, J.: Kopplung und prozessorientierte Verknüpfung der Methoden und Werkzeuge. In: MEERKAMM, H.; HENRICH, A.; JABLOWSKI, S.; KRCCMAR, H.; LINDEMANN, U.; RIEG, F. (Hrsg.): Flexible Prozessunterstützung in der Produktentwicklung: Prozesse – Daten – Navigation, Abschlussbericht, 2010
- [Zin13] ZINGEL, C.: 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, Fakultät für Maschinenbau, Universität des Landes Baden-Württemberg und nationales Forschungszentrum in der Helmholtz-Gemeinschaft, Forschungsberichte: IPEK, Band 70, Karlsruhe, 2013

Anhang

Inhaltsverzeichnis	Seite
A1	Ergänzungen zur Problemanalyse..... A-1
A1.1	Prozessbereiche der ISO15288-2015 A-1
A1.2	Zuordnung von SE-Rollen zu einem Unternehmen des Mittelstand . A-2
A2	Ergänzungen zu den erarbeiteten MBSE-Konzepten A-2
A2.1	Visualisierung der MBSE-Konzepte..... A-2
A2.2	Vollständige Ausprägungsliste der MBSE-Konzepte A-3
A2.3	Kompetenzorientierte Auswahl von MSBE-Konzepten A-5
A2.4	Ergänzung zu Beispielprojekten ohne Quellenangabe A-5
A3	Ergänzungen zum flexiblen Vorgehensmodell A-7
A3.1	Kurzüberblick und Hinweise zu Six Sigma..... A-7
A3.1.1	DMAIC-Circle A-7
A3.1.2	Six Sigma-Rollen A-8
A3.1.3	Reifegradmodell der Einführung von Six Sigma A-9
A3.2	Übersicht über entwickelte, erweiterte und eingesetzte Methoden und Hilfsmittel im flexiblen Vorgehensmodell A-10
A4	Merkmale Morphologie der Modellierungsmethoden..... A-10
A5	Ergänzungen zu den Modellen des Kaffee-Automaten A-14
A5.1	Umfeldmodell mit SysML4CONSENS A-14
A5.2	Auszug Partialmodell Funktionshierarchie (CONSENS)..... A-15
A5.3	Auszüge aus dem Partialmodell Wirkstruktur (CONSENS) A-16
A5.4	Auszug aus der Modellierung mit Sysmod..... A-17
A6	Ergänzungen zu den gekoppelten Methoden A-17
A6.1	Stakeholderanalyse A-17
A6.1.1	Stakeholderanalyse nach MÜLLER-STEWENS und LECHNER A-17
A6.1.2	Web-Applikation Stakeholderanalyse A-18
A6.2	Gefährdungsanalyse A-19
A6.2.1	Web-Applikation Gefährdungsanalyse..... A-19
A6.3	Mechatronische Komplexitätsanalyse, angelehnt an MARTI A-19
A6.3.1	Formel zur Errechnung der Komplexität nach MARTI A-20

A7	Ergänzende Erläuterungen zur IT-Infrastruktur	A-21
A8	Auszüge aus dem Datenmodell.....	A-22

A1 Ergänzungen zur Problemanalyse

A1.1 Prozessbereiche der ISO15288-2015

Die führende Richtlinie des Systems Engineering ist der International Standard ISO/IEC/IEEE15288: Systems and software engineering – System lifecycle processes, der in seiner aktuellen Ausgabe vom 15. Mai 2015 vorliegt. Hier werden neben den bislang stark adressierten technischen Prozessen zusätzlich neben anderen auch insb. die Technischen Managementprozesse als essentiell angesehen. Aufgrund der engen Verzahnung von Systems Engineering und Projektmanagement finden sich hier auch Prozesse wie z.B. Projektplanung und Entscheidungsmanagement. Insgesamt umfasst die Richtlinie 30 unterschiedliche Prozesse, die für ein erfolgreiches Systems Engineering durchgeführt und beherrscht werden müssen. Das unterstreicht den ganzheitlichen Charakter von Systems Engineering und die Notwendigkeit, das modellbasierte Systems Engineering auch auf diese Prozessbereiche auszuweiten, die insb. über die Architekturgestaltung hinausgehen.

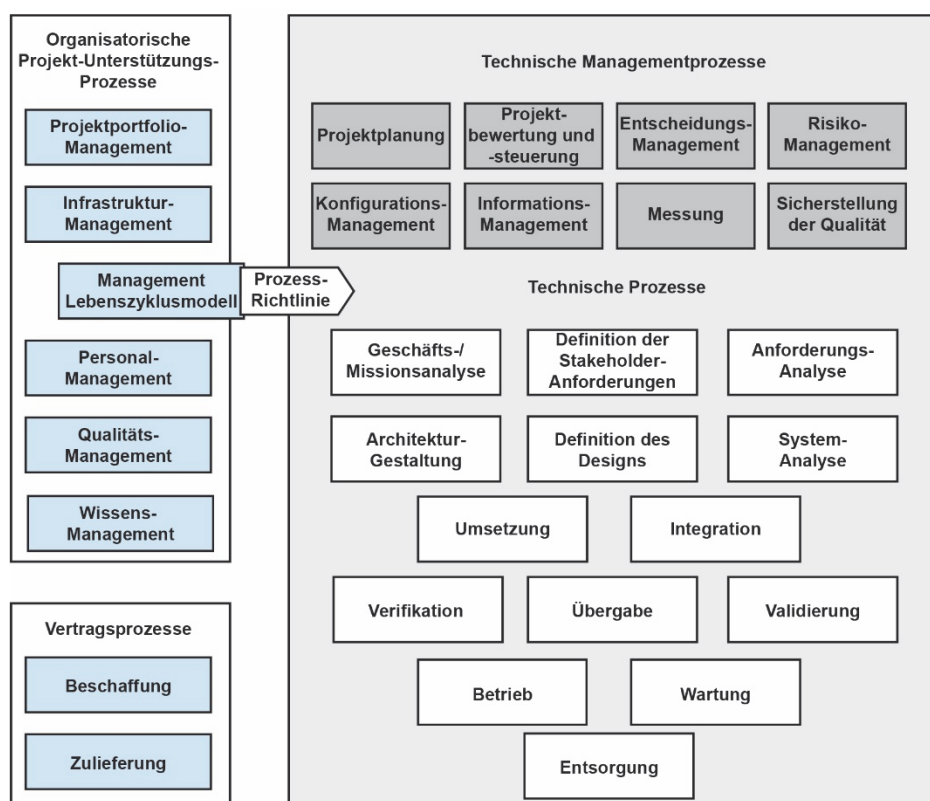


Bild 8-1 Prozessklassen und Prozesse der ISO15288-2015, Darstellung angelehnt an das dt. SE-Handbuch von INCOSE [INC12], Übersetzung ins Deutsche durch den Autor

A1.2 Zuordnung von SE-Rollen zu einem Unternehmen des Mittelstand

MOEHRINGER analysiert die Bedeutung der verschiedenen SE-Rollen nach SHEARD [She96] und überträgt sie auf sein eigenes Unternehmen, einem Betrieb mit etwa 130 Mitarbeitern/-innen. Durch diese Zuordnung wird deutlich, dass die Idee des Systems Engineering auch in kleinen Betrieben gelebt werden kann und dass eine flexible Rollenverteilung dazu notwendig ist [Möh12].

	RO	SD	SA	VV	LO	G	CI	TM	IM	PE	CO
Vertriebsingenieur	Red						Red				
Projektleiter	Blue					Red	Blue	Red		Red	Red
Entwicklungsleiter		Grey		Grey							
Entwicklungsingenieure (nach Produktgruppen)		Blue	Red	Blue	Red						
Serviceleiter					Grey						
QS-Leiter										Grey	
Geschäftsleitung											Grey

Rollenzuordnung zu Funktionsträgern nach „Projektgröße“

RO: Requirements Owner LO: Logistics and Operations IM: Information Manager
 SD: System Designer G: Glue PE: Process Engineer
 SA: System Analyst CI: Customer Interface CO: Coordinator
 VV: Validation and Verification TM: Technical Manager

■ Kleines Projekt ■ Mittleres Projekt ■ Großprojekt

Bild 8-2 Übertragung von SE-Rollen auf Rollen eines kleinen Betriebes

A2 Ergänzungen zu den erarbeiteten MBSE-Konzepten

A2.1 Visualisierung der MBSE-Konzepte

In Kapitel 4.2 wurden mit Hilfe der Szenario-Technik vier verschiedene „MBSE-Konzepte“ entwickelt, die als idealtypische Repräsentanten des MBSE seinen Einsatz unterstützen sollen. In der Szenario-Technik liegen als Ergebnis sog. „Szenarien“ vor – hier MBSE-Konzepte. Die Szenarien werden zugunsten der schnelleren Verständlichkeit häufig visualisiert – z.B. mittels Collagen in sog. Zukunftsbildern [GPW14]. In Anlehnung an dieses Vorgehen werden die MBSE-Konzepte hier ebenfalls visualisiert; dabei werden gerade die dominanten und die eindeutigen Merkmalausprägungen berücksichtigt. Auf diese Weise erhalten die Konzepte zusätzlich zur textuellen Beschreibung noch einen visuellen Charakter.

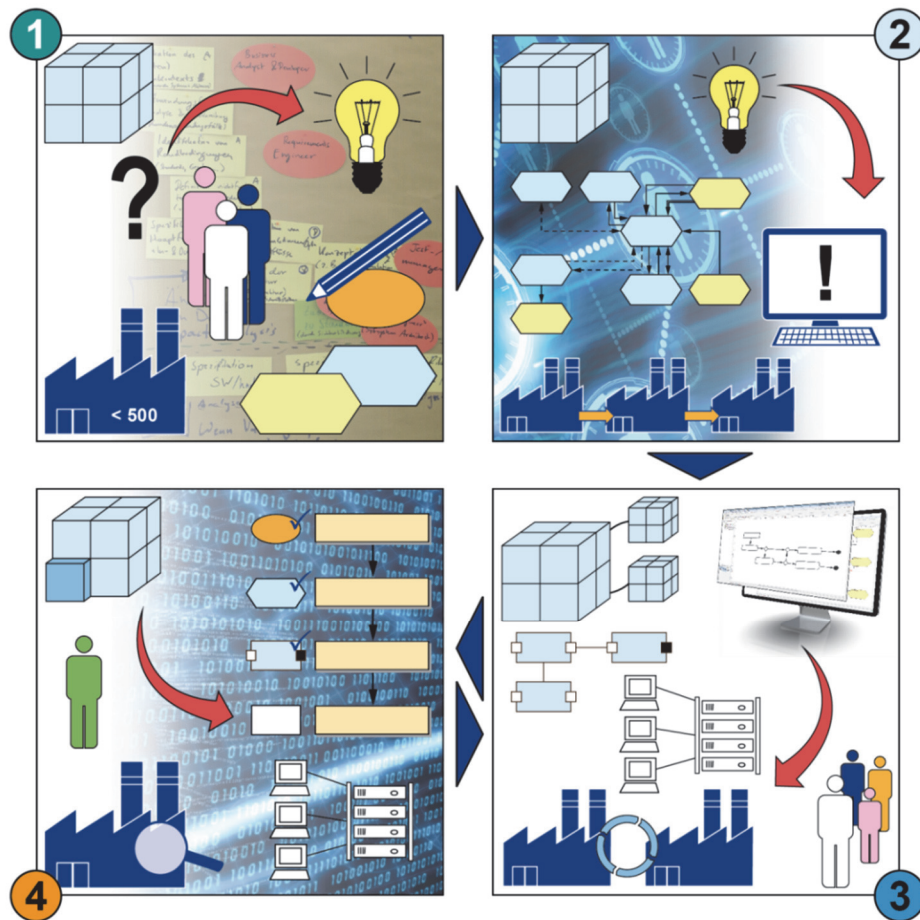


Bild A-1 Visualisierung der MBSE-Konzepte, angelehnt an die sog. "Zukunftsbilder"

A2.2 Vollständige Ausprägungsliste der MBSE-Konzepte

Die folgende Abbildung zeigt als Ergänzung zu Bild 4-7 die vollständige Merkmalliste mit sämtlichen Ausprägungen und den Gewichten der einzelnen MBSE-Konzepte.

Merkmal	Ausprägungen		Klasse I	Klasse II	Klasse III	Klasse IV
Formalisierungsgrad	1A	Informal	70	0	0	0
	1B	semi-formal	30	100	64	4
	1C	formal	0	0	35	95
Anzahl Sprachkonstrukte	2A	Sehr wenige	20	40	0	0
	2B	Wenige	80	60	87	4
	2C	Viele	0	0	12	95
Konkrete Syntax („Visual Syntax“)	3A	Keine	40	0	0	0
	3B	Form/Farbe	60	0	3	50
	3C	Farbe und Farbe	0	100	96	50
Aspekte	4A	Anforderungen	100	20	8	0
	4B	Struktur	0	80	62	6
	4C	Gestalt	0	0	0	16
	4D	Verhalten	0	0	27	62
	4E	Parameter	0	0	1	16
Vorgehensweise	5A	Inkrementell	0	20	83	95
	5B	Iterativ	100	80	16	4
Phasenablauf	6A	formlos	100	0	12	0
	6B	formal	0	100	87	100
Phasendefinition	7A	Top-Down	80	100	93	0
	7B	Bottom-Up	0	0	0	100
	7C	Zigzagging	20	0	6	0
Werkzeugart	8A	Brown Paper	70	0	8	0
	8B	Diagramming Tools	30	60	32	0
	8C	COTS/Dedicated Tools	0	40	59	20
	8D	Simulationswerkzeug	0	0	0	79
Datenmodell	9A	ja	0	0	74	100
	9B	nein	100	100	25	0
Unternehmensgröße	10A	Klein	100	100	3	0
	10B	Groß	0	0	96	100
Projektgröße	11A	Klein	90	0	0	16
	11B	Mittel	10	80	12	45
	11C	Groß	0	20	87	37
Projektart	12A	Vorentwicklung	70	40	0	33
	12B	Kundenspez. Entwicklung	30	60	32	0
	12C	Serienentwicklung	0	0	67	66
Reichweite Wertschöpfungskette	13A	Projektintern	100	0	0	100
	13B	Unternehmensintern	0	0	30	0
	13C	Wertschöpfungskette	0	100	69	0
System-of-Interest	14A	SoS	0	0	24	0
	14B	System	100	100	75	0
	14C	Modul	0	0	0	100
Modellierungsaufwand	15A	Gering	0	0	0	0
	15B	Mittel	100	40	22	0
	15C	Hoch	0	60	77	100
Modellierungsverantwortung	16A	Workshop	100	20	6	0
	16B	Architekt	0	80	96	0
	16C	Fachspezialisten	0	0	0	100
Stakeholder	17A	Management	0	0	0	0
	17B	Technik	0	0	0	100
	17C	Enabler	100	100	100	0

 eindeutige Ausprägung
 dominante Ausprägung

 alternative Ausprägung
 schwache Ausprägung/ Ausprägung tritt nicht auf

Bild A-2 Vollständige Ausprägungsliste MBSE-Konzepte

A2.3 Kompetenzorientierte Auswahl von MSBE-Konzepten

Die Minimalanforderungen zur Anwendung der MBSE-Konzepte orientieren sich an dem vom INCOSE UK Advisory Board erarbeiteten Systems Engineering Competencies Framework (SECF) [INC10]. Es umfasst die drei Beurteilungsklassen *Systems Thinking*, *Holistic Lifecycle View* und *Systems Engineering Management*. In der Originalversion enthalten die Beurteilungsklassen die folgenden Inhalte, die im Rahmen der Arbeit adäquat ins Deutsche übersetzt wurden:

- **Systems Thinking:** systems concepts, super-system capability issues, enterprise and technology environment
- **Holistic Lifecycle view:** Determine and manage stakeholder requirements, system design (architectural design, concept generation, design for ..., functional analysis, interface management, maintaining design integrity, modelling and simulation, select preferred solution, system robustness), integration & verification, transition to operation
- **Systems Engineering Management:** concurrent engineering, enterprise integration, lifecycle process definition, planning, monitoring and controlling

Die einzelnen Inhalte der Beurteilungsklassen sind weitgehend konsistent mit der ISO15288. Damit gehen sie auch über die rein technischen Aspekte des Systems Engineerings hinaus und sind deshalb eine gute Grundlage zur Unterstützung der kompetenzorientierten Auswahl eines MBSE-Konzepts.

Die gewählten Fähigkeitsklassen *verstehen – anwenden – beherrschen – Fachexperte* orientieren sich an den Originalklassen *awareness – supervised practitioner – practitioner – expert*. Hier wurde das gleiche Beurteilungsschema zugrunde gelegt.

Für weiterführende Informationen sei auf die Arbeiten von INCOSE UK verwiesen.

A2.4 Ergänzung zu Beispielprojekten ohne Quellenangabe

Die in Kapitel 5.1.1 aufgeführten Untersuchungseinheiten konnten entweder in der Literatur identifiziert werden oder stammen aus Erfahrungen im Projektumfeld des Fraunhofer IEM. Letztere unterliegen häufig einer Geheimhaltungsvereinbarung, weshalb i.d.R. nicht auf veröffentlichte Quellen verwiesen werden kann. Zum besseren Verständnis werden hier anonymisierte Zusatzinformationen zu Untersuchungseinheiten aufgeführt, die in Tabelle 5-1 oder den Bildern 5–2 bis 5–7 näher erläutert werden.

UE17: Das Unternehmen (> 9.000 Mitarbeiter, > 2 Mrd. Euro Umsatz) entwickelt und produziert Systeme und Module für Unternehmen unterschiedlicher Branchen mit Fokus auf den Handel. Die Systeme verarbeiten physische Objekte und Informationen gleichermaßen. Im betrachteten Projekt wurde ein neuartiges System für einen Spezial-

markt vorentwickelt, das nun im Rahmen einer Kundenanfrage spezifisch auf die Anforderungen des Kunden hin angepasst wurde. Der Systemarchitekt wurde mit der Methode CONSENS unterstützt.

UE18: Das gleiche Unternehmen und Vorgehen wie UE17, jedoch ein anderes System

UE19: Analog UE18, nun jedoch mit typischen COTS-Werkzeugen zu einem späteren Lebenszykluszeitpunkt.

UE20: Das Unternehmen (> 6.000 Mitarbeiter, > 1 Mrd. Euro Umsatz) ist Automobilzulieferer. Ursprünglich mit großer Mechanik-Kompetenz hat sich das Unternehmen über die Zeit zu einem Innovator im Bereich mechatronischer Module für den gesamten Kfz-Markt entwickelt. Im Projekt wurden in einem Innovationsworkshop mit Mitarbeitern aus Mechanik und Elektronik ein aktuelles System mit der Methode CONSENS retrospektiv modelliert und hierauf aufbauend Verbesserungspotentiale identifiziert.

UD21: Das Unternehmen (1.000 Mitarbeiter, 200 Mio. Euro Umsatz) ist als Zulieferer in der Automobilbranche bekannt. Eine innovative Idee hat zu einem Großauftrag eines OEM geführt. Das Projekt wurde begleitet, im Mittelpunkt der Aufbau der Systemstruktur, teilweise auch erste Verhaltensbeschreibungen. Genutzt wurde die Spezifikationstechnik CONSENS; für das Unternehmen wurde zeitweise die Rolle des Systemarchitekten übernommen.

UE22 – UE25: Die Untersuchungseinheiten UE22 bis UE25 sind sich alle weitgehend ähnlich, sowohl was die Art der Unternehmen, der Projekte und die Ausgestaltung der Projekte mit Methoden des MBSE angeht. Alle Unternehmen stammen aus dem Umfeld des Maschinen- und Anlagenbau und sind mittelständisch geprägt. In Innovations- und Kundenprojekten ging es um das Erstellen einer Systemarchitektur, die Projekte wurden durch einen externen Systemarchitekten mit der Methode CONSENS unterstützt.

UE26: Der Sondermaschinenbauer (40 Mitarbeiter, Umsatz unbekannt) einer großen Unternehmensgruppe hat Methoden des MBSE als Vertriebsunterstützung genutzt. Mit dem externen Kunden wurden in sehr kurzen Workshops mit der Spezifikationstechnik CONSENS erste Modelle aufgebaut – ähnlich zu UE31.

UE27: Das Unternehmen (> 3.000 Mitarbeiter) als Teil eines weltweit agierenden Konzerns nutzt CONSENS in einem Projekt über einen längeren Zeitraum. Dabei stehen immer wieder unterschiedliche Aspekte im Mittelpunkt, meist geht es um die Nutzung der Methode für die Weiterentwicklung der eigenen Produkte.

UE28: Das Unternehmen (> 7.000 Mitarbeiter, > 1 Mrd. Euro Umsatz) hat ein Innovationsprojekt mit Methoden von CONSENS unterstützt. Dabei ging es zunächst um die Identifikation von Anforderungen und Use Cases, dann in einem weiteren Schritt das Erstellen von Funktionshierarchie und Wirkstruktur. Änderungen an Anforderungslisten und den anderen Partialmodellen wurden mittels MS Office-Produkten über einen längeren Zeitraum konsistent gepflegt.

UE29: Für ein Unternehmen aus der Branche Lichttechnik (< 30.000 Mitarbeiter, > 5 Mrd. Euro Umsatz) wurden in Ideenworkshops neue Anwendungsbereiche und Ansätze für neuartige Technologien identifiziert. Die Ergebnisse wurden auf unterschiedlichen Wegen dokumentiert, wobei die Methode CONSENS eine wichtige Rolle spielte.

UE34: Im Rahmen eines Messeprojekts haben das Unternehmen Miele und Fraunhofer IEM für das Unternehmen Dassault Systèmes (DS) einen Messedemonstrator erstellt. Genutzt wurde die spezifische Software von DS. Darüber hinaus wurde durch Fraunhofer die Methode CONSENS genutzt, um ein System von Miele darzustellen.

A3 Ergänzungen zum flexiblen Vorgehensmodell

A3.1 Kurzüberblick und Hinweise zu Six Sigma

Das gesamte Rahmenwerk orientiert sich an den Grundpfeilern etablierter Ansätze, wie z.B. dem Six Sigma. Die Inhalte und Informationen zu Six Sigma stammen weitgehend aus dem vom Autor absolvierten Six Sigma-Zertifizierungskurs zum Black Belt nach ASQ-Standard, teilweise ergänzt um Inhalte aus der Literatur, bpsw. aus [TK08].

A3.1.1 DMAIC-Circle

Grundlage des Six Sigma ist der sog. DMAIC-Circle, der den Makroprozess des Gesamtansatzes darstellt. Die Bedeutung der einzelnen Phasen wurde bereits in Kapitel 4 vorgestellt und für das vorliegende Rahmenwerk auf vier Phasen reduziert. Mit dem Durchlaufen dieses Kreises soll die Basisidee des Ansatzes strukturiert erreicht werden: Das Bestreben, Qualität mittels einer Kennzahl messbar zu machen. Die Standardabweichung einer Grundgesamtheit Sigma ist hier das entsprechende Qualitätsmaß. Wörtlich genommen müsste beim Ziel „Six Sigma“ zu 99,99966% fehlerfrei gearbeitet werden. Pragmatisch gilt allerdings, dass das Qualitätsniveau so hoch sein sollte, dass es vom Kunden akzeptiert wird. Aus Gründen der Vollständigkeit wird hier noch einmal der klassische DMAIC-Circle vorgestellt.



Bild 8-3 DMAIC-Circle

A3.1.2 Six Sigma-Rollen

Six Sigma lebt von seinen stringent definierten Rollen („belts“). Im Systems Engineering existieren mit den sog. „SEPs“ (A-SEP, C-SEP, E-SEP) und dem dt. Pendant den SE-Zerts ähnliche, jedoch bei weitem nicht so verbreitete Konzepte. Der Vollständigkeit halber werden hier noch einmal die Six Sigma-Rollenkonzepte präsentiert.

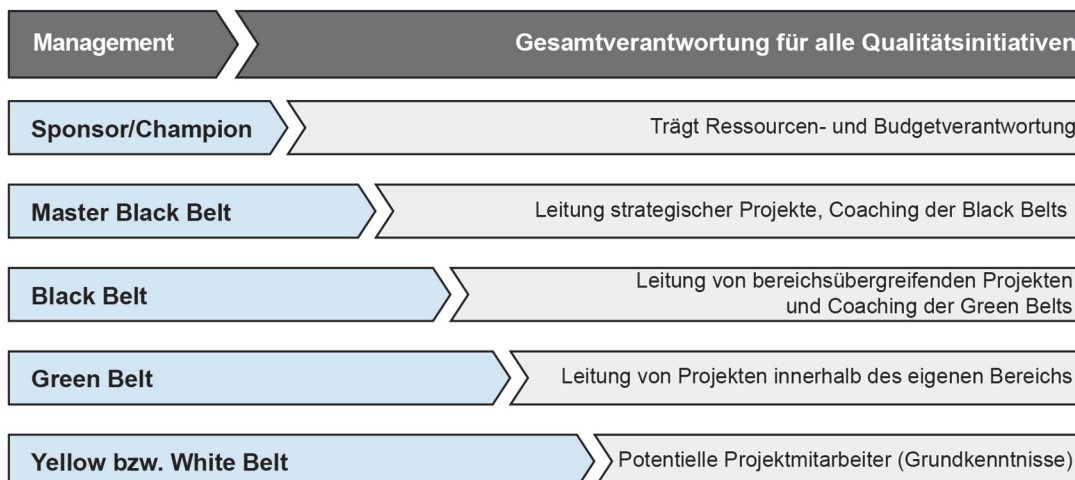


Bild 8-4 Six Sigma-Rollen

A3.1.3 Reifegradmodell der Einführung von Six Sigma

Die Erfolgsgeschichte von Six Sigma begann mit der konsequenten Anwendung des Ansatzes durch General Electric. Selbstredend kann ein solcher Ansatz nicht von jetzt auf gleich als Unternehmensparadigma eingesetzt werden. Aus organisatorischer Sicht kann ein Reifegradmodell für die Einführung von Six Sigma idealtypisch wie im Bild dargestellt werden. Ähnlich kann sich auch die Einführung von SE und MBSE gestalten. Die MBSE-Konzepte folgen einem ähnlichen Ansatz.

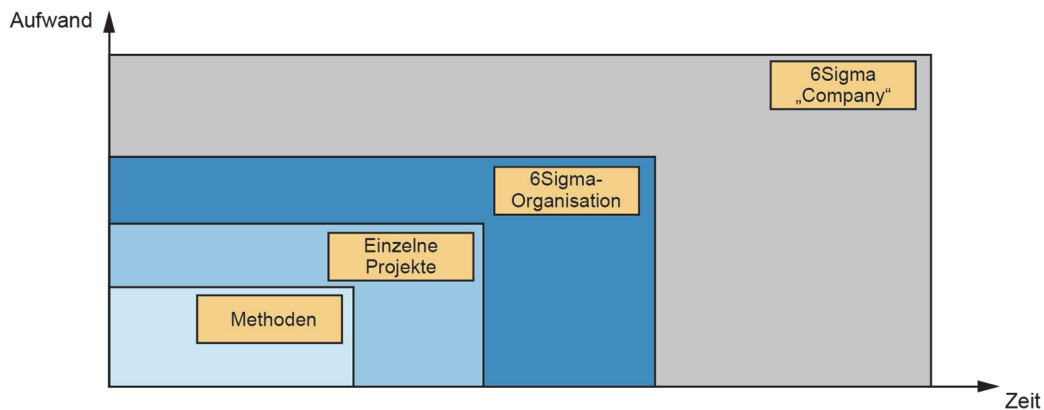


Bild 8-5 Reifegradmodelle für die Einführung von Six Sigma

A3.2 Übersicht über entwickelte, erweiterte und eingesetzte Methoden und Hilfsmittel im flexiblen Vorgehensmodell

Tabelle 8-1 Überblick über entwickelte oder adaptierte Methoden und Hilfsmittel

Nr.	Name	Phase	Aktivität	Art	Kapitel
1	Flexibles Vorgehensmodell zur Planung des Modellierungsprojekts			E	Kap. 4.5.1
2	Idealtypische Vorgehensweisen im flexiblen Vorgehensmodell			E	Kap. 4.5.2
3	MBSE-Konzepte	Analyse	IE	E	Kap. 4.2.3
4	Entscheidungshilfe zur lebenszyklusorientierten Auswahl der MBSE-Konzepte	Analyse	IE	E	Kap. 4.3.1
5	Entscheidungshilfe zur modellierungszweckorientierten Auswahl der Konzepte	Analyse	IE	E	Kap. 4.3.2
6	Entscheidungshilfe zur aufgabenorientierten Auswahl der MBSE-Konzepte	Analyse	IE	E	Kap. 4.3.3
7	Entscheidungshilfe zur kompetenzorientierten Auswahl der MBSE-Konzepte	Analyse	IE	E	Kap. 4.3.4
8	Steckbrief „Planung d. Modellierungsprojekts“	Define	IE	E	Kap. 4.6.1.1
9	Steckbrief „Methodenanalyse“	Define	IG/IV/IA	E	Kap. 4.6.1.2
10	Steckbrief „Control“	Control	IG/IV/IA	E	Kap. 4.6.5
11	Quick Check zur Methodenauswahl	Define	IE	E	Kap. 4.6.1.1
12	S-I-P-O-C	Define	IG	A	Kap. 4.6.1.2
13	Stakeholdernetz	Define	IG	A	Kap. 4.6.1.2
14	RASCI-Matrix	Define	IG	A	Kap. 4.6.1.2
15	Grundkonzepte Methodenkopplung	Define	IV	A	Kap. 4.6.1.2
16	Vorlage „Informationsausgabearten“	Define	IA	E	Kap. 4.6.1.2
17	MBSE-Konzepte Merkmalsliste	Analyse	IE	E	Kap. 4.6.2.1
18	MBSE-Methoden-Steckbrief (Morphologie)	Analyse	IE	E	Kap. 4.6.2.1
19	Organisationskonzepte zur Modellierung	Analyse	IE	E	Kap. 4.6.2.1
20	Methoden-Essenz	Analyse	IG	E	Kap. 4.6.2.2
21	Sichten-Template	Analyse	IG	E	Kap. 4.6.2.2
22	GUI-Design-Vorlagen	Analyse	IA	E	Kap. 4.6.2.3

E: Eigenentwicklung - A: Adaption bestehender Ansätze

IE: Informationserzeugung; IG: Informationsgewinnung; IV: Informationsverarbeitung; IA: Informationsausgabe

A4 Merkmale Morphologie der Modellierungsmethoden

Die im Methoden-Steckbrief (Morphologie) beschriebenen Merkmale und ihre Ausprägungen werden hier der Vollständigkeit halber kurz beschrieben.

Methodeninhärente Merkmale

- Phasenablauf: Unter Phasenablauf wird die Vorgabe zur Abfolge der einzelnen Arbeitsschritte einer Methode verstanden. Im Entwicklungsprozess können die einzelnen Phasen schrittweise durchlaufen werden und/oder in regelmäßigen

Abständen wiederkehren – also inkrementell oder iterativ. In diesem Kontext werden Methoden als iterativ betrachtet, wenn die Modellierungsmethode keine abgeschlossenen Phasen definiert. Streng inkrementelle Methoden sind einfacher in der Anwendung. Iterative Methoden hingegen ermöglichen den Anwender neue Erfahrungswerte in vorhergegangene Entwicklungsphasen mit einfließen zu lassen. In der modernen Entwicklung gibt es Methoden mit sowohl inkrementellen als auch iterativen Phasenablauf [Ram12].

- **Phasendefinition:** Unter der Phasendefinition wird die Genauigkeit der Vorgaben zu den einzelnen Arbeitsschritten einer Methode verstanden. Die Definition der Phasen hat Einfluss auf die Anwendung einer Methode. Methodisches Vorgehen ist nach BALZER „eine Gradwanderung zwischen Formalismus und Formlosigkeit“. Methoden die eher formlos sind begünstigen die Kreativität auf Kosten eines strukturierten Vorgehens. Zu formale Methoden dämmen zwar einen chaotischen und undurchsichtigen Ablauf ein, hemmen jedoch zugleich eigene Ideenansätze [Bal99]. Unterschieden wird in formlos oder formal.
- **Sprachenkompatibilität:** Unter Sprachenkompatibilität wird die Umsetzbarkeit einer Methode mit Sprachen verstanden. Die Anzahl an kompatiblen Sprachen erhöht sich signifikant durch die industrielle Verbreitung einer Methode. Je größer die Auswahl an kompatiblen Sprachen zu einer Methode ist, desto größer ist die Chance für Unternehmen, eine passende Sprache für den Bedarf zu finden – und desto attraktiver ist die Methode für die praktische Anwendung. Unterschieden wird in Kompatibilität mit einer Sprache oder mit mehreren Sprachen.
- **Sprachausrichtung:** Unter der Sprachausrichtung wird das Maß der Inanspruchnahme des Sprachumfangs durch die Modellierungsmethode verstanden. Diese gibt Aufschluss, wie effektiv die Methode die Sprache – mit ihren Vielfältigen Darstellungen und Möglichkeiten – tatsächlich ausnutzt: „Verwertet die Methode die Stärken einer Sprache?“ Die Ausrichtung der Methode zu der Sprache ist schwach oder stark. Methoden, die nur mit einer Sprache effektiv umsetzbar sind, weisen somit eine hohe Abhängigkeit zur Sprache auf. Ist die Methode mit mehreren Sprachen kompatibel, wird an dieser Stelle die größtmögliche Sprachausrichtung aufgeführt. Die Sprachausrichtung kann aus der Beschreibung der Methode entnommen werden – z.B. die Einbindung konkreter Diagrammartentypen. Gerade bei Methoden, die nicht von demselben Herausgeber wie die zugehörige Modellierungssprache stammen, gibt dieses Merkmal Aufschluss wie effektiv eine Modellierungsmethode die Sprache umsetzt.
- **Skalierbarkeit:** Die Leitfrage hierbei ist, ob die Methode in Teilen genutzt werden kann oder nur als Ganzes sinnvoll einsetzbar ist. Das ist gerade vor dem Hintergrund unterschiedlicher Modellierungszwecke von Bedeutung. Wichtig für die Skalierbarkeit ist, dass bei gegebener Skalierbarkeit die einzelnen Ele-

mente bereits einen Mehrwert generieren. Grundsätzlich wird die Skalierbarkeit unterschieden in *gegeben* und *nicht gegeben*.

Anwendungsbezogene Merkmale

- Modellierungszweck: vgl. Kapitel 4.3.2
- Anforderungsermittlung: Hiermit ist die Herangehensweise zur Erfassungen der ersten Anforderungen mit Hilfe der Methode gemeint. Die Ermittlung der Anforderungen kann in einer Modellierungsmethode durch drei unterschiedliche Arten gelingen: *request-driven*, *feature-driven* oder *usage-driven*. Stehen bei der Anforderungserfassung die Kundenaufträge an erster Stelle („request-driven approach“), dann wird weniger Marktforschungs-Knowhow in der Entwicklung gefordert – die Anforderungen werden extern an das Produkt gestellt. Erfolgt die Ermittlung der Anforderungen intern, so ist die Reihenfolge der Erfassung von Bedeutung: „Werden zunächst die Produkteigenschaften bzw. Funktionen eines Systems erfasst („feature-driven approach“)“ oder „Werden als erstes die möglichen Anwendungsfälle eines Systems ermittelt („usage-driven approach“)“ Dies beeinflusst die möglichen Adressierungen einer Methode: Nach NOLAN ET AL. sind „usage-driven“ Methoden im Allgemeinen besser auf die Kundenwünsche ausgerichtet, da nur Anforderungen berücksichtigt werden die im Anwendungsfall relevant sind [NBB+08]. In der Praxis eignen sich „usage-driven“ Methoden zur Entwicklung neuer Produkte, während „feature-driven“ Methoden schneller und besser für Variantenentwicklungen oder für die Erschließung neuer Anwendungsbereiche geeignet sind.
- Systemvalidierung: Unter der Validierung des Systems wird die Leistungsbeurteilung des Systems im Ablauf der Modellierungsmethode verstanden. Da die Eigenbewertung nach MÜLLER Bestandteil einer Methode ist [Mül90] und LOHMEYER aufführt, dass Methoden u.a. aufgrund ihrer Leistung akzeptiert werden [ALR12], ist es interessant zu überprüfen, bei welchen Methoden die Systemvalidierung konkret vorgeschrieben ist. So schreiben manche Methoden die Modellierung von Leistungskriterien („Measures Of Effectiveness) oder konkrete Testabläufe vor. Wenn die Validierung des Systems in der Methode nicht explizit vorgeschrieben wird, bedeutet dies natürlich nicht, dass keine Validierung stattfindet. Die Validierung eines Systems kann bei einer Modellierungsmethode demnach die Ausprägungen annehmen: *nicht Bestandteil der Methode – Leistungskriterien – Testvorgaben*.
- Variantenmodellierung: Auch wenn bislang keine der in Kapitel 3 analysierten Methoden die Variantenmodellierung unterstützt, ist sie ein wichtiges Kriterium bei der Auswahl einer Modellierungsmethode. Da selbst die ersten Versionen der Sprache SysML die Variantenmodellierung noch vollkommen ausser Acht gelassen haben, wird die Integration von Variantenmodellierung in Methoden

noch einige Zeit dauern. Dennoch wird sie als Merkmal mit den Ausprägungen *ja* und *nein* in die Morphologie aufgenommen.

- Hilfestellungen: Unter Hilfestellungen werden Hinweise und Ratschläge bezüglich der Methodenanwendung verstanden. Konkrete Vorschläge – wie z.B. der Verweis zum Einsatz anderer Methoden oder Modellierungsregeln – helfen generell bei der Methodenanwendung und sind vor allem in Beschreibungen zu praxisorientierten Methoden zu finden. Dadurch können in Methodenbeschreibungen auftreten: *keine – Hinweise – dedizierte Vorgaben*.

Organisation

- Methodeneigner: In Erweiterung zu den identifizierten MBSE-Konzepten wird hier das Kriterium Methodeneigner aufgenommen, mit den Unterscheidungen *nicht definiert – Systemarchitekt – Systemmodellverantwortlicher*.
- Rollenprofile: Für die tatsächliche Nutzung der Methode ist auch die Unterscheidung von Rollen wichtig, wie bspw. in Kapitel 4.6.2 mit der R-A-S-C-I Klassifikation aufgezeigt. Das kann bspw. bei der Modellierung einzelner Diagramme helfen. Unterschieden wird hier in *keine Vorgaben – idealisierte Vorschläge – klar definiert*.

Werkzeug

- Werkzeugabhängigkeit: Unter Werkzeugabhängigkeit wird die Notwendigkeit der Unterstützung einer Methode mit Hilfe der Werkzeuge verstanden. Eine hohe Abhängigkeit liegt vor, wenn signifikante Abläufe der Methode nur mit Hilfe des Werkzeuges möglich sind. Die Werkzeugabhängigkeit einer Methode kann zwischen *niedrig* und *hoch* unterschieden werden. Die Werkzeugabhängigkeit kann durch die Liste an Werkzeugfeatures unter Abgleich mit den Methodenvorgaben ermittelt werden
- Softwarezugänglichkeit: Unter der Softwarezugänglichkeit wird der Aufwand zum Erwerb rechnerunterstützter Werkzeuge verstanden. Der Werkzeugzugang wird durch die Anzahl an verfügbaren Werkzeugen und die Kosten für deren Erwerb beeinflusst. Der Zugang zu Softwareunterstützung kann sich als *einfach* oder *schwierig* herausstellen. Die Angaben zum Werkzeugzugang sind aus der Sprachbeschreibung entnommen.

A5 Ergänzungen zu den Modellen des Kaffee-Automaten

A5.1 Umfeldmodell mit SysML4CONSENS

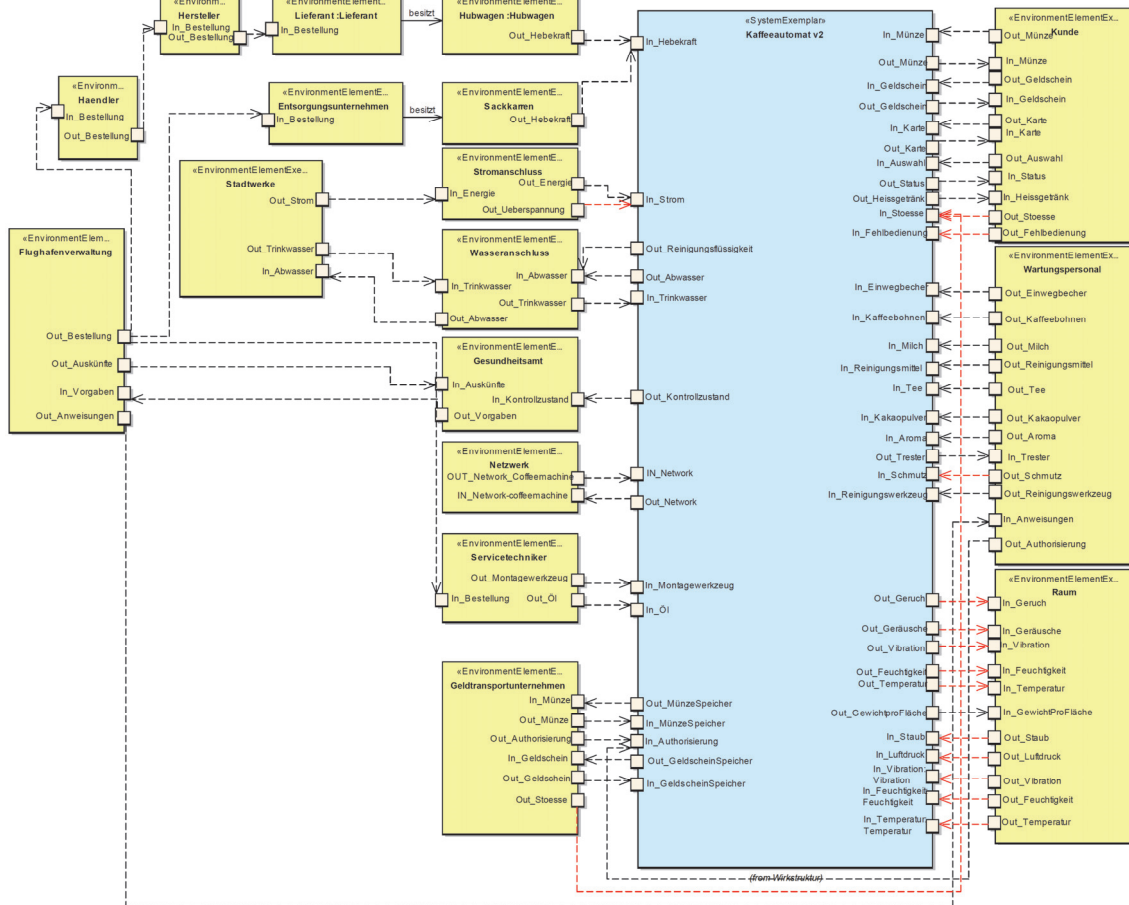


Bild 8-6 Stakeholder im Umfeldmodell mit SysML4CONSENS – MBSE-Konzept 3

A5.2 Auszug Partialmodell Funktionshierarchie (CONSENS)

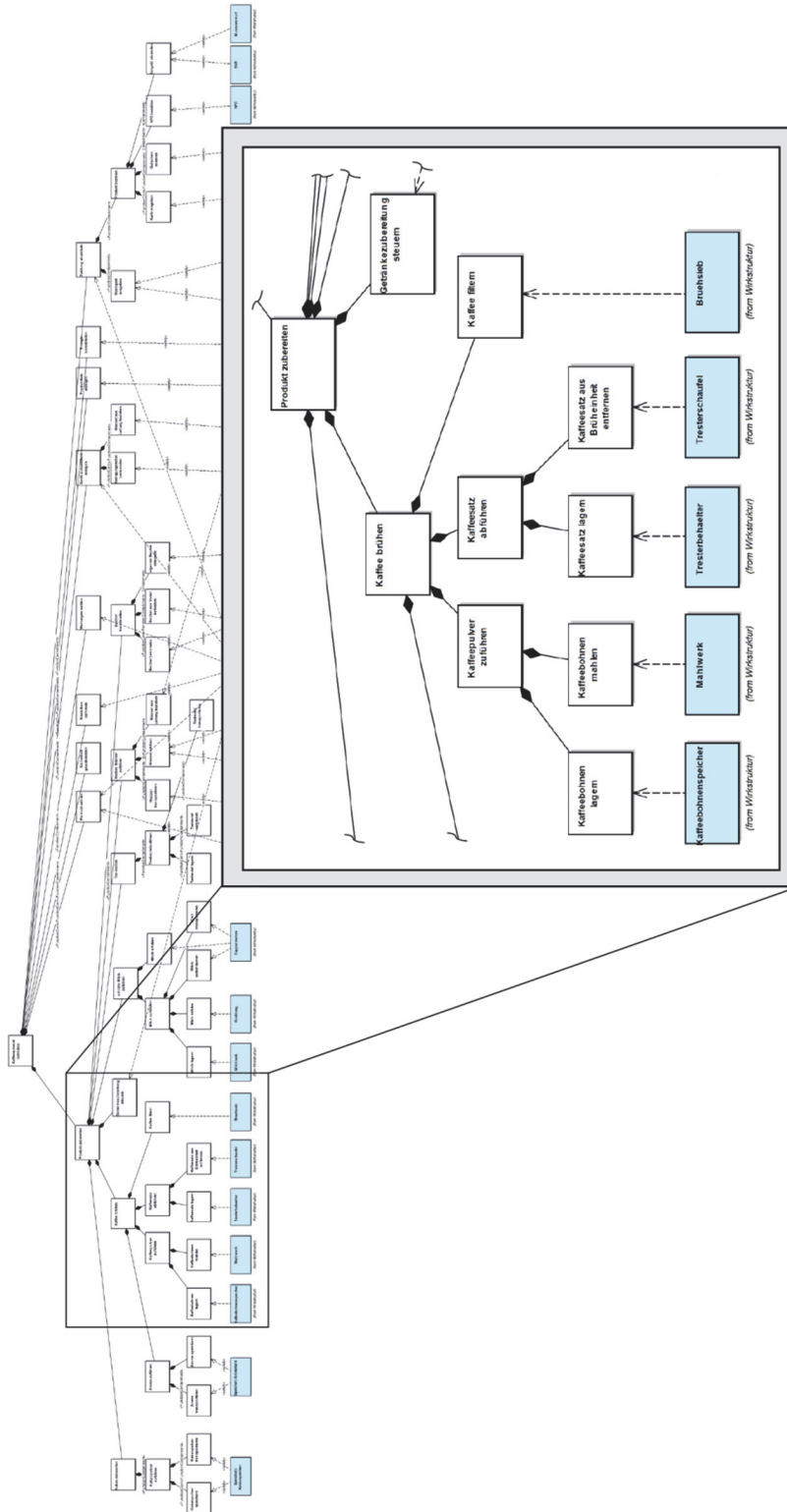


Bild 8-7 Funktionshierarchie (Überblick und Auszug Funktion Produkt zubereiten)

A5.3 Auszüge aus dem Partialmodell Wirkstruktur (CONSENS)

Im Folgenden einige Auszüge der Wirkstruktur der Kaffeemaschine.

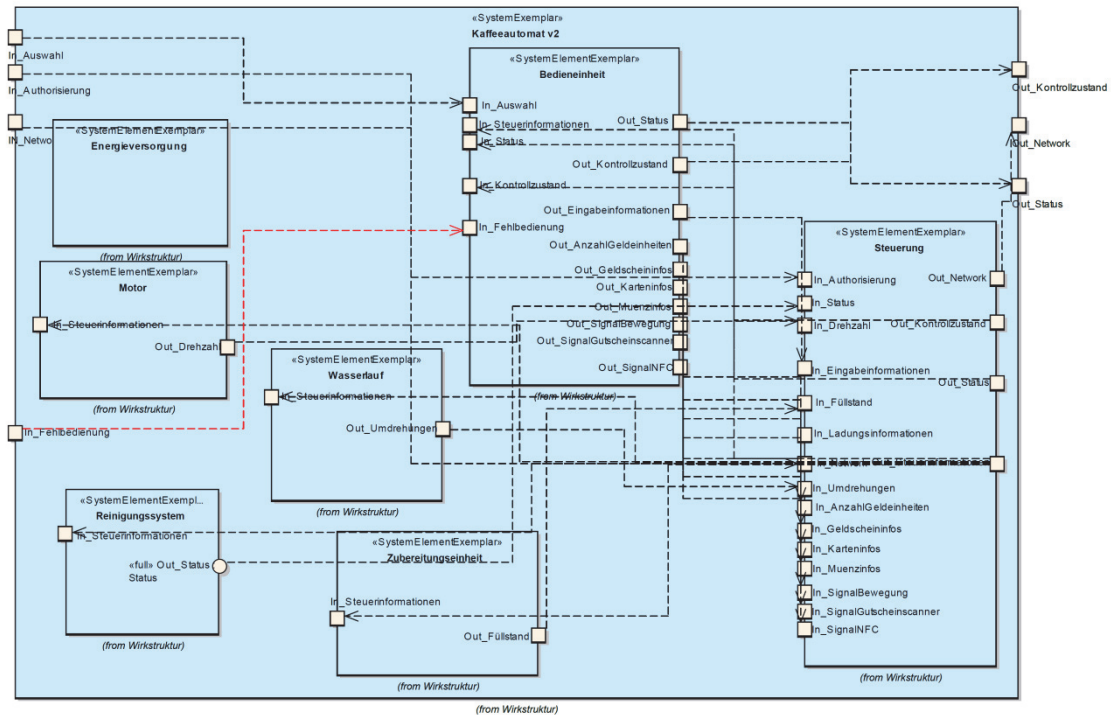


Bild 8-8 Wirkstruktur, Ebene 1 – Informationssicht

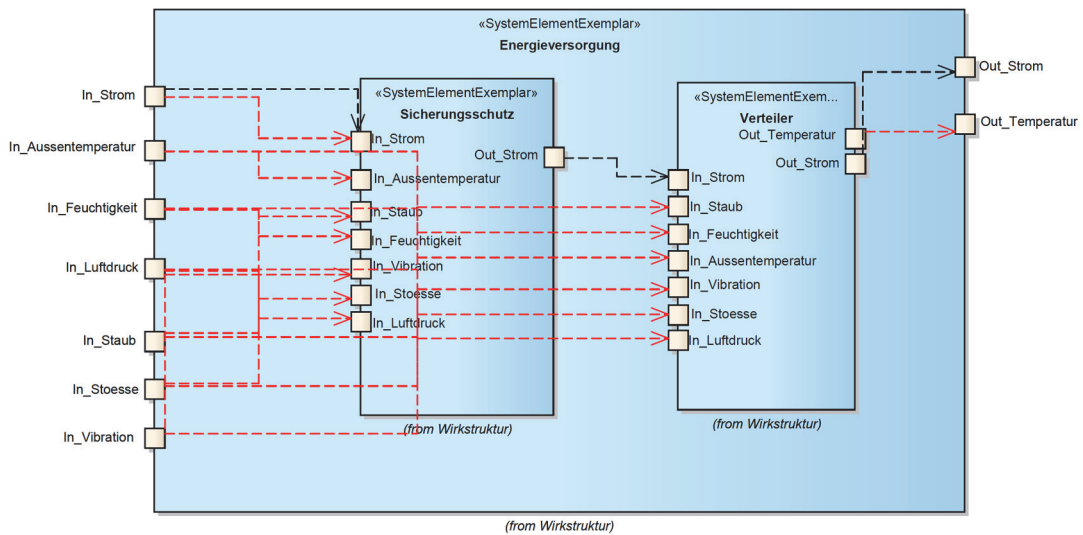


Bild 8-9 Wirkstruktur, Ebene 2 – Energieversorgung

A5.4 Auszug aus der Modellierung mit SysMod

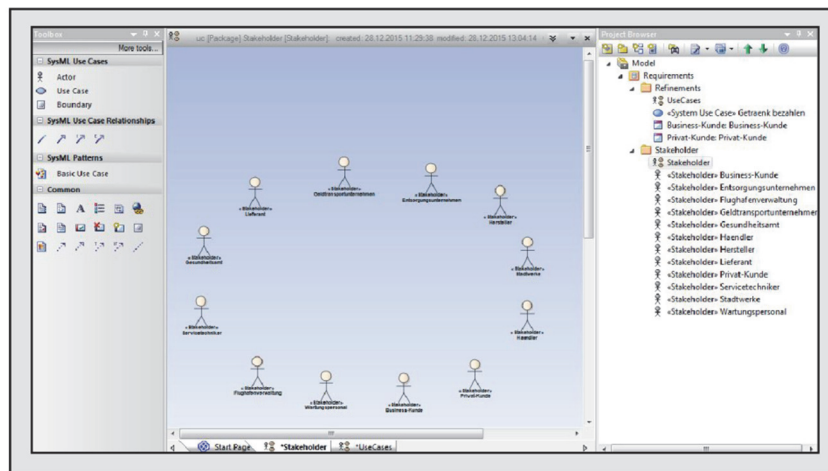


Bild 8-10 Stakeholder-Diagramm mit SysMod

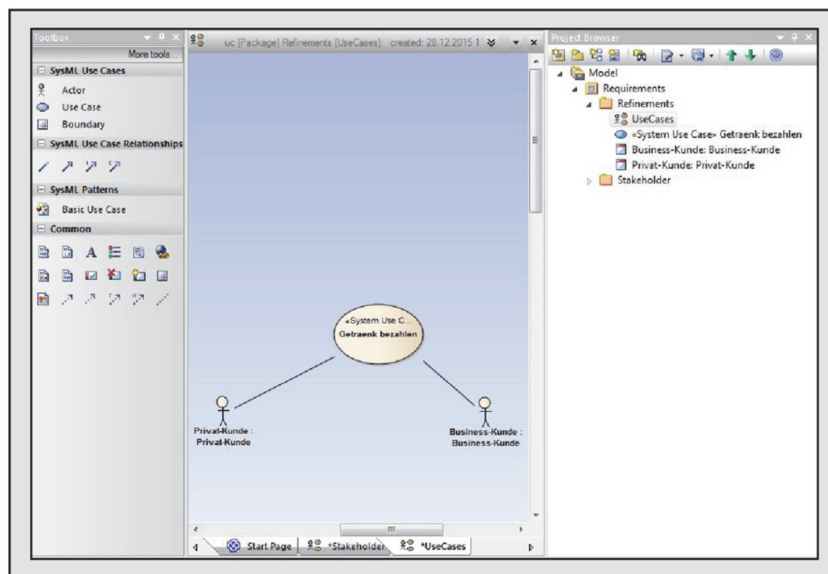


Bild 8-11 Use Case „Getraenk bezahlen“ und beteiligte Stakeholder

A6 Ergänzungen zu den gekoppelten Methoden

A6.1 Stakeholderanalyse

A6.1.1 Stakeholderanalyse nach MÜLLER-STEWENS und LECHNER

In Kapitel 4 wurde die Stakeholderanalyse nach MÜLLER-STEWENS und LECHNER genutzt, um die Methoden und Hilfsmittel des flexiblen Vorgehensmodells darzulegen. In Ergänzung zu den Darstellungen in Kapitel 4 folgen hier zur besseren Nachvollziehbarkeit eine kurze Zusammenfassung der Methode und ergänzende Informationen [ML11].

Ziel des Ansatzes von MÜLLER-STEWENS und LECHNER sind Hinweise für Strategien und Maßnahmen im Umgang mit Stakeholdern, die aus der Analyse von Stakeholdern abgeleitet werden können. Das Vorgehen gliedert sich in vier Schritte, die im Rahmen von Kapitel 4 dargestellt wurden. Nach der eher intuitiven Bewertung der Stakeholder lassen sich diese in vier Klassen einteilen, die eine unterschiedliche Relevanz für das weitere Agieren haben:

Spielmacher: Sie besitzen einen hohen Einfluss auf das Unternehmen, werden aber gleichzeitig stark vom Unternehmen beeinflusst. Zu diesen Stakeholdern sollten dauerhafte Kommunikationskanäle aufgebaut werden, da sie höchste Priorität für ein Unternehmen haben sollten.

Joker: Stakeholder können einen hohen Einfluss ausüben, sind allerdings kaum zu beeinflussen. Ein Unternehmen sollte versuchen, seinen Einfluss auf die Joker zu steigern, bspw. durch Aufbau von Kooperationen im Falle eines Wettbewerbers.

Gesetzte: Hier hat das Unternehmen eine Machtposition gegenüber dem Stakeholder. Diese Stakeholder könnten allerdings versuchen, indirekt über „Verbündete“ ihren Einfluss zu erhöhen, so z.B. im Rahmen von Abnehmer-Lieferanten-Beziehungen.

Randfiguren: Momentan sind die Randfiguren noch ohne Bedeutung für das Unternehmen. Der Aufwand sollte ihnen ggü. gering gehalten werden, bspw. durch einfachen Informationsaustausch.

A6.1.2 Web-Applikation Stakeholderanalyse

Als ein Beispiel der implementierten Web-Applikationen zur Durchführung von Analysen auf Basis des Systemmodells dient hier die Web-Applikation der Stakeholderanalyse nach MÜLLER-STEWENS, die in [TBD+15] bereits dargestellt wurde.

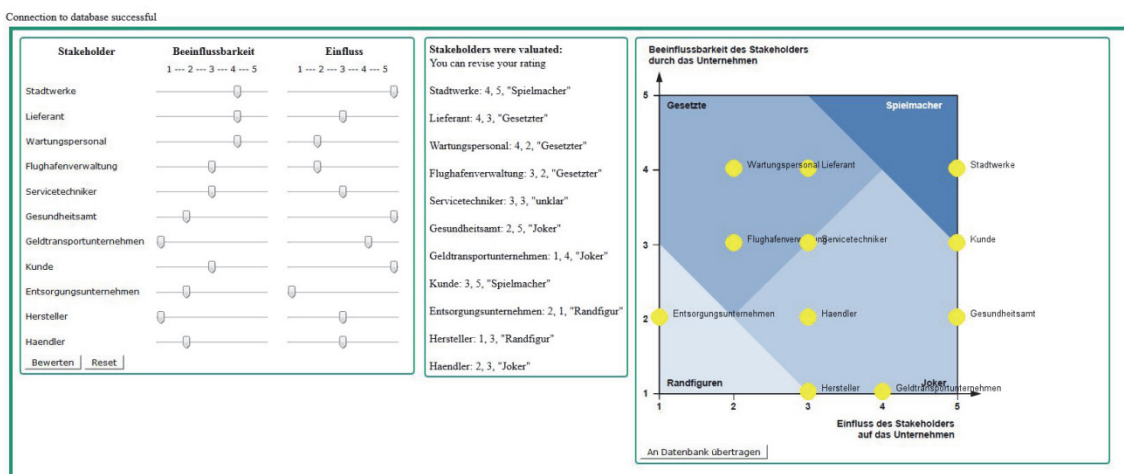


Bild 8-12 Web-Applikation Stakeholderanalyse auf Basis des Systemmodells

A6.2 Gefährdungsanalyse

A6.2.1 Web-Applikation Gefährdungsanalyse

Als ein Beispiel der implementierten Web-Applikationen zur Durchführung von Analysen auf Basis des Systemmodells dient hier die Web-Applikation der Gefährdungsanalyse nach NOHL.

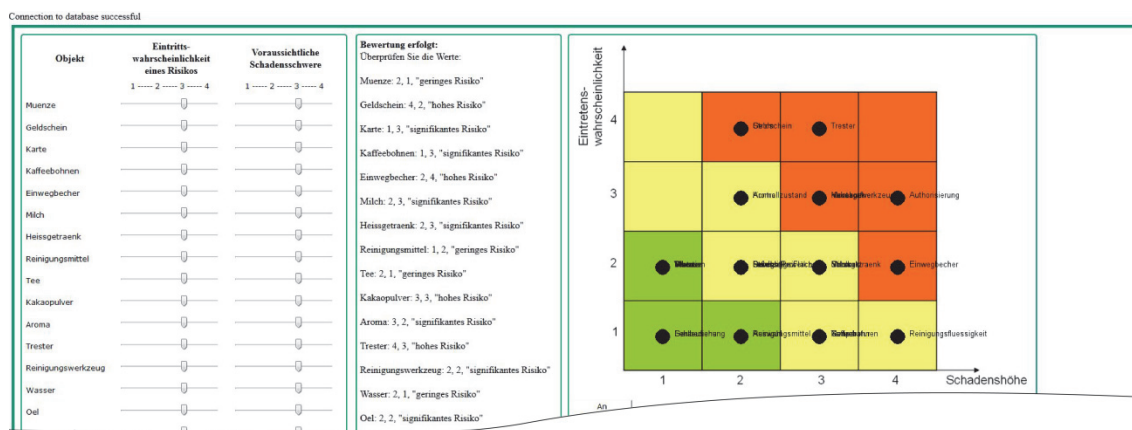


Bild 8-13 Web-Applikation Gefährdungsanalyse (Ausschnitt)

A6.3 Mechatronische Komplexitätsanalyse, angelehnt an MARTI

MARTI entwickelte 2007 eine Methode zur Beurteilung der Komplexität technischer Systeme [Mar07]. Basis war die Bewertung der Komplexitätstreiber Vielfalt und Vielzahl als Repräsentant für die interne Komplexität eines Produkts. Diese resultiert maßgeblich aus der Vielzahl an verbauten Elementen und ihrer Verschiedenartigkeit (= Varianten). Diese wird in einer Matrix der sog. externen Komplexität gegenübergestellt, repräsentiert durch den empfundenen Nutzen einer Produktfunktion. Die interne Komplexität lässt sich anhand verschiedener Formeln berechnen, die externe Komplexität wird bspw. in Workshops ermittelt. Hierzu wird die Gesamtfunktion des Produkts so lange heruntergebrochen, bis eine Funktion in eine Komponente übersetzt werden kann. Das entspricht dem klassischen entwicklungsmethodischen Vorgehen nach PAHL/BEITZ [PBF+07]. Im resultierenden Portfolio ergeben sich in Anlehnung an die BCG-Matrix vier unterschiedliche Bereiche, anhand der die Komplexität einer Komponente beurteilt werden kann und etwaige Produktänderungen geplant werden können:

- *Lucky Strikes* sind ideale Elemente. Sie steuern ein hohes Maß an Funktionalität zu dem Produkt bei und stiften damit Kundennutzen. Gleichzeitig weisen sie eine geringe Komplexität auf.
- *Stars* sind genau wie Lucky Strikes sehr wichtig aus Kundensicht. Sie weisen allerdings eine hohe Komplexität auf

- *Standards* weisen geringen Funktionsnutzen bei gleichzeitig geringer Komplexität auf – meist handelt es sich um standardisierte Komponenten.
- *Money Burners* weisen wenig Funktionalität bei hoher Komplexität auf. Die hohen Kosten, die durch ihre Komplexität entstehen, werden vom Kunden nicht honoriert. Elemente sollen mit Normstrategien wie bspw. einer besseren Modularisierung oder je nach Situation einem stärkeren Fokus auf Integralbauweise aus diesem Quadranten verschoben werden.

Die Methode von MARTI ist für eine variantenorientierte Analyse geeignet. Im Rahmen dieser Arbeit wurde sie zur Veranschaulichung der Nutzung der Informationen aus dem Systemmodell für die Charakteristika mechatronischer Systeme weiterentwickelt. Dadurch ergab sich die Möglichkeit bislang aufgrund fehlender Informationen neuartige Aspekte bei der Analyse der Komplexität zu berücksichtigen. In diesem Zuge wurden auch die o.g. Bereiche neu benannt.

A6.3.1 Formel zur Errechnung der Komplexität nach MARTI

MARTI setzt auf dem klassischen Verständnis der Produktarchitektur auf und bringt die „externe Komplexität“ (repräsentiert durch die Funktionen) und die „interne Komplexität“ (repräsentiert durch die Komplexitätstreiber Vielfalt und Vielzahl) ins Verhältnis. Dazu benutzt er für die physikalische Komplexität C der Komponente i :

$$C_i = \alpha \frac{N_{e,j}}{N_{e,max}} + \beta \frac{V_{e,j}}{V_{e,max}} + \gamma \frac{N_{r,i}}{N_{r,max}} + \delta \frac{V_{r,avg,i}}{V_{r,avg,max}}$$

$N_{e,i}$: Anzahl Elemente der Komponente i

$N_{e,max}$: höchste Anzahl an Elementen einer Komponente im betrachteten System

$V_{e,i}$: Anzahl an Varianten einer Komponente i

$V_{e,max}$: höchste Anzahl an Varianten einer Komponente im betrachteten System

$N_{r,i}$: Anzahl an Schnittstellen einer Komponente i

$N_{r,max}$: höchste Anzahl an Schnittstellen einer Komponente im System

$V_{r,avg,i}$: durchschnittliche Anzahl an Varianten einer Schnittstelle

$V_{r,avg,max}$: maximale durchschnittliche Anzahl an Varianten einer Schnittstelle

Berechnung der Durchschnittswerte jedes Komplexitätstreibers:

$$F_{1,avg} = \frac{1}{n} \sum_{i=1}^n \frac{N_{e,j}}{N_{e,max}}$$

$$F_{2,avg} = \frac{1}{n} \sum_{i=1}^n \frac{V_{e,j}}{V_{e,max}}$$

$$F_{3,avg} = \frac{1}{n} \sum_{i=1}^n \frac{N_{r,i}}{N_{r,max}}$$

$$F_{4,avg} = \frac{1}{n} \sum_{i=1}^n \frac{V_{r,avg,i}}{V_{r,avg,max}}$$

Jeder Komplexitätstreiber hat das gleiche Gewicht:

$$w_{CD} = \frac{1}{n_{CD}}$$

n_{CD} : Anzahl der Komplexitätstreiber, hier vier

Berechnung der Skalierungsfaktoren:

$$\alpha' = \frac{w_{CD}}{F_{1,avg}} \quad \beta' = \frac{w_{CD}}{F_{2,avg}} \quad \gamma' = \frac{w_{CD}}{F_{3,avg}} \quad \delta' = \frac{w_{CD}}{F_{4,avg}}$$

Die Komplexität muss zwischen 0 und 1 liegen, daher gilt dies auch für die Summe der Koeffizienten:

$$\alpha + \beta + \gamma + \delta = 1$$

Daher erfolgt einer Skalierung bezüglich ihrer Summe:

$$\alpha = \frac{\alpha'}{\alpha' + \beta' + \gamma' + \delta'}$$

$$\beta = \frac{\beta'}{\alpha' + \beta' + \gamma' + \delta'}$$

$$\gamma = \frac{\gamma'}{\alpha' + \beta' + \gamma' + \delta'}$$

$$\delta = \frac{\delta'}{\alpha' + \beta' + \gamma' + \delta'}$$

Die Berechnung der Komplexität auf Basis des Systemmodells erfolgt weitgehend analog und wurde in [GTH16] ebenso wie die Gefährdungsanalyse umgesetzt. Die Formel zur Berechnung inkl. der vorgenommenen Anpassungen ist in Bild 5-10 dargestellt und erfolgt damit weitgehend analog.

A7 Ergänzende Erläuterungen zur IT-Infrastruktur

Für die vorliegende Anwendung von Methoden auf Basis des Systemmodells wurde eine einfache Infrastruktur eingerichtet, die grundsätzlich auch bspw. für die Anwendung in einem kleinen Unternehmen des Maschinen- und Anlagenbau geeignet ist. Dies betrifft das Datenbankmanagementsystem MySQL, einen Apache-Server auf dem die PHP-Anwendungen liegen, die Installation der Anwendungssoftware Enterprise Architect (EA) und die Installation von Java. Das Einrichten der Server geschieht hier bei-

spielhaft auf Basis eines XAMPP in der Version 3.2.1. Dieses Paket beinhaltet einige Serverlösungen unter anderem einen Apache-Server, MySQL und PHPmyAdmin zur Verwaltung der Datenbank.

Datenbank und Repository werden mit einem von Enterprise Architect bereitgestellten SQL-Befehl kompatibel gemacht. Für eine Verbindung zwischen Datenbank und Repository ist zudem noch eine ODBC-Schnittstelle nötig, in der die Verbindungsdaten eingetragen sind. So kann nun ein Systemmodell in die Datenbank geschrieben werden und umgekehrt auch aus der Datenbank in das Systemmodell eingelesen werden.

A8 Auszüge aus dem Datenmodell

Das mit Enterprise Architect unter Verwendung der SysML-Profilen SysML4CONSENS und SysMod erstellte Datenmodell kann über verschiedene Wege ausgelesen werden, bspw. über eine SQL-Datenbank. Die enthaltenen Elemente sind in zahlreichen Tabellenspalten dokumentiert. Im Folgenden sind die für die Methodenanwendungen wichtigsten Spalten der relevanten Tabellen zusammengefasst. Die Aufzählung erhebt keinen Anspruch auf Vollständigkeit, sondern spiegelt den Stand wider, der für eine erfolgreiche Methodenkopplung genutzt wurde. Je nach SysML-Profil und Vorgehensweise gestalten sich die Inhalte der Datenbank unterschiedlich. Zur detaillierten Erläuterung der Begriffe, wie z.B. Primärschlüssel, sei auf die etablierte Standardliteratur verwiesen.

t_attribut: Beinhaltet alle Attribute, denen Objekte zugewiesen wurden außer tagged values (vgl. t_objectproperties und t_connectortag)

- Object_ID zugeordnetes Objekt
- Name Bezeichnung des Attributs
- ID Primärschlüssel
- Default Standardwert
- Type Parametertyp (int, char, etc.)
- ea_guid Fungiert wie Primärschlüssel

t_connector: Beinhaltet sämtliche Verbinder z.B. Flüsse, satisfy oder FunctionContainments

- Connector_ID Primärschlüssel
- Name Bezeichnung des Verbinders
- Connector_Type Typ des Verbinders (Aggregation, InformationFlow, etc.)
- Start_Object_ID Objekt am Anfang des Verbinders (Ports, Function, SystemElement, etc.)

- End_Object_ID Objekt am Ende des Verbinders (Ports, Function, SystemElement, etc.)
- Stereotype Spezifikation des Verbinders (satisfy, FunctionContainment, Flow, etc.)
- ea_guid Fungiert wie Primärschlüssel

t_connectortag: beinhaltet die sog. Tagged Values der Verbinder

- PropertyID Primärschlüssel
- ElementID zugeordneter Verbinder
- Property Bezeichnung des tagged values
- Value Wert des tagged values
- ea_guid Fungiert wie Primärschlüssel

t_object: Beinhaltet sämtliche Objekte außer den Verbindern z.B. Systemelemente, Ports, Funktionen etc.

- Object_ID Primärschlüssel
- Object_Type Art des Objekts (Port, Part, Class, etc.)
- Name Bezeichnung des Objekts
- Stereotype Spezifikation der Objektart (EnvironmentElementTemplate, MaterialFlowSpezifikation etc.)
- PDATA1 ea_guid der Klasse des Objekts
- ea_guid Fungiert wie Primärschlüssel
- ParentID Object_ID des übergeordneten Objekts z.B. zugehöriges Systemelement eines Ports oder das Systemelement dem ein anderes ein gegliedert ist

t_objectproperties: beinhaltet die sog. Tagged Values der Objekte der Tabelle t_object

- PropertyID Primärschlüssel
- Object_ID zugeordneter Verbinder
- Property Bezeichnung des tagged values
- Value Wert des tagged values
- ea_guid Fungiert wie Primärschlüssel

Lebenslauf

Aus Gründen des Datenschutzes enthält diese online-Version keinen Lebenslauf.

Das Heinz Nixdorf Institut – Interdisziplinäres Forschungszentrum für Informatik und Technik

Das Heinz Nixdorf Institut ist ein Forschungszentrum der Universität Paderborn. Es entstand 1987 aus der Initiative und mit Förderung von Heinz Nixdorf. Damit wollte er Ingenieurwissenschaften und Informatik zusammenzuführen, um wesentliche Impulse für neue Produkte und Dienstleistungen zu erzeugen. Dies schließt auch die Wechselwirkungen mit dem gesellschaftlichen Umfeld ein.

Die Forschungsarbeit orientiert sich an dem Programm „Dynamik, Mobilität, Vernetzung: Auf dem Weg zu den technischen Systemen von morgen“. In der Lehre engagiert sich das Heinz Nixdorf Institut in vielen Studiengängen der Universität. Hier ist das übergeordnete Ziel, den Studierenden die Kompetenzen zu vermitteln, auf die es in der Wirtschaft morgen ankommt.

Heute wirken am Heinz Nixdorf Institut sieben Professoren mit insgesamt 200 Mitarbeiterinnen und Mitarbeitern. Etwa ein Viertel der Forschungsprojekte der Universität Paderborn entfallen auf das Heinz Nixdorf Institut und pro Jahr promovieren hier etwa 30 Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler.

Heinz Nixdorf Institute – Interdisciplinary Research Centre for Computer Science and Technology

The Heinz Nixdorf Institute is a research centre within the University of Paderborn. It was founded in 1987 initiated and supported by Heinz Nixdorf. By doing so he wanted to create a symbiosis of computer science and engineering in order to provide critical impetus for new products and services. This includes interactions with the social environment.

Our research is aligned with the program “Dynamics, Mobility, Integration: En-route to the technical systems of tomorrow.” In training and education the Heinz Nixdorf Institute is involved in many programs of study at the University of Paderborn. The superior goal in education and training is to communicate competencies that are critical in tomorrow's economy.

Today seven Professors and 200 researchers work at the Heinz Nixdorf Institute. The Heinz Nixdorf Institute accounts for approximately a quarter of the research projects of the University of Paderborn and per year approximately 30 young researchers receive a doctorate.

Bände der Verlagsschriftenreihe

{Liste der Bände erhältlich im HNI Sekretariat}