Systematik zur Evolution technischer Anforderungen

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

genehmigte DISSERTATION

von
Christopher Lankeit
aus Hamm

Tag des Kolloqiums: 20. April 2021

Referent: Prof. Dr.-Ing. habil. Ansgar Trächtler Korreferent: Prof. Dr.-Ing. Roman Dumitrescu

Vorwort

Diese Dissertation entstand während meiner Zeit als wissenschaftlicher Mitarbeiter in der Fachgruppe Regelungstechnik und Mechatronik am Heinz Nixdorf Institut der Universität Paderborn und am Fraunhofer Institut für Entwurfstechnik Mechatronik (IEM). Sie ist das Ergebnis meiner wissenschaftlichen Arbeit im Rahmen von Industrie- und Forschungsprojekten. Hierzu gehörten die EU-Forschungsprojekte "ENTIME - Entwurfstechnik intelligente Mechatronik" und "SHT - Smart Headlamp Technology" sowie das Innovationsprojekt "InverSa - Intelligente vernetzte Systeme für automatisierte Geldkreisläufe" des BMBF-Spitzenclusters "it's OWL - Intelligente technische Systeme Ostwestfalen-Lippe".

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. habil. Ansgar Trächtler, meinem Doktorvater, der mir eine einmalige Chance gegeben hat und mich in unzähligen Gesprächen immer unterstützt hat und wesentlich zum Erfolg dieser Arbeit beigetragen hat. Ich danke ihm für sein Vertrauen und die Ermöglichung dieser Dissertation. Herrn Prof. Dr.-Ing. Roman Dumitrescu, Leiter der Fachgruppe Advanced Systems Engineering des Heinz Nixdorf Instituts, danke ich für die Übernahme des Korreferats und die konstruktiven Gespräche. Ich bedanke mich auch bei Herrn Prof. Dr. rer. nat. Thomas Tröster und Herrn Prof. Dr.-Ing. Detmar Zimmer für ihr Mitwirken in der Promotionskommission.

Meinen ehemaligen Kolleginnen und Kollegen danke ich für die erfolgreiche Zusammenarbeit und für die schöne Zeit am Lehrstuhl und am Fraunhofer IEM mit einem außergewöhnlich positiven Arbeitsklima. Ein besonderer Dank gebührt meinen ehemaligen Kollegen und Freunden Dr.-Ing. Matthias Lochbichler, Jan Michael, Simon Olma, Dr.-Ing. Felix Oestersötebier, Dr.-Ing. Viktor Just und Johannes Renninger für die fachlichen Diskussionen und für die inhaltliche Auseinandersetzung mit meiner Dissertation. Auch bei meinen ehemaligen studentischen Hilfskräften Peter Malsam, Fabian Ernst, Ke Xu, Christopher Lüke und Mohamed Abdelrahman bedanke ich mich.

Meine Eltern Birgit und Ralf sowie meine Schwester Kira mit Matthias und Lennart haben mir in meinem Leben immer zur Seite gestanden und haben mich wesentlich geprägt. Ich danke ihnen von ganzem Herzen. Ebenfalls danke ich Allyn und Leo für die tolle Unterstützung.

Abschließend geht mein besonders herzlicher Dank an Melisa, die mich auf unbeschreibliche Weise auf diesem Weg unterstützt hat.

Vorveröffentlichungen

- [1] LANKEIT, C.; MICHAEL, J.; HENKE, C.; TRÄCHTLER, A.: Holistic Requirements for Interdisciplinary Development Processes. In: Proceedings of 1st International Workshop on Learning from Other Disciplines for Requirements Engineering (D4RE), Banff, Canada, 20.-24. August 2018, IEEE
- [2] HENKE, C.; MICHAEL, J.; LANKEIT, C.; TRÄCHTLER, A.: Virtuelle Inbetriebnahme eines Fertigungszentrums. In: Proceedings of Tag des Systems Engineering 2017 (TDSE), Paderborn, 08.-10. November 2017, GfSE
- [3] Henke, C.; Michael, J.; Lankeit, C.; Trächtler, A.: Holistic Approach for Virtual Commissioning of Intelligent Systems: Model-Based Systems Engineering for the Development of a Turn-Milling Center. In: IEEE Systems Conference 2017 (SYSCON), Montreal, Canada, 24.-27. April 2017, IEEE
- [4] OESTERSÖTEBIER, F.; ABRISHAMCHIAN, F.; LANKEIT, C.; JUST, V.; TRÄCHTLER, A.: Approach for an Integrated Model-Based Design of Intelligent Dynamic Systems Using Solution and System Knowledge. In: 3rd International Conference on System-integrated Intelligence 2016 (SysInt), 13.-15. Juni 2016
- [5] LANKEIT, C.; JUST, V.; TRÄCHTLER, A.: Consistency Analysis for Requirements, Functions, and System Elements. In: IEEE Systems Conference 2016 (SYSCON), Orlando, USA, 18.-21. April 2016, IEEE
- [6] LANKEIT, C.; LOCHBICHLER, M; OESTERSÖTEBIER, F.; TRÄCHTLER, A.: From Goals to Systems via Technical Requirements and Physical Models. In: Proceedings of IEEE International Symposium on Systems Engineering 2015 (ISSE), Rom, Italien, 28. September 01. Oktober 2015, IEEE
- [7] LANKEIT, C.; LOCHBICHLER, M; TRÄCHTLER, A.: Challenges in Requirements Engineering for Mechatronic Systems Problem Analysis and first Approach. In: Jurnal Teknologi, 76(4), 2015
- [8] Lochbichler, M.; Lankeit, C.; Trächtler, A.; Landwehr, M.; Hoischen, L.: Systematische Wahl der Modellierungstiefe im Entwurfsprozess mechatronischer Systeme am Beispiel eines Geldautomaten. In: Tagungsband Mechatronik, Dortmund, 12.-13. März 2015, VDI Mechatronik
- [9] Shareef, Z.; Just, V.; Teichrieb, H.; Lankeit, C.; Trächtler, A.: Dynamical Model of Ball Juggling Delta Robots using Reflection Laws. In: 16th International Conference on Advanced Robotics (ICAR), Montevideo, Uruguay, 25.-29. November 2013, IEEE

Zusammenfassung

Der Umgang mit Komplexität, Schnittstellenvielfalt und Domänenvielfalt kann bei vernetzten und emergenten Systemen über eine übergeordnete Betrachtung und geführte Prozesse gelingen. Technischen Anforderungen kommt in allen Phasen der Entwicklung eine entscheidende Bedeutung zu. Sie eignen sich als Zieldefinition und Überprüfbarkeitskriterien sowohl in der Systemsynthese, als auch in der Systemanalyse. In dieser Arbeit wird ein Vier-Ebenen Modell von Anforderungen erläutert. In diesem werden technische Anforderungen sukzessive und systematisch detailliert. Ausgehend von Zielen werden funktionsorientierte Anforderungen, domänenübergreifende und domänenspezifische Anforderungen unterschieden. Diese Darstellungsform eignet sich für weitergehende Systemanalysen in Bezug auf die Durchgängigkeit und Vollständigkeit von Anforderungen. Die Vorgehensweise unterscheidet sich dabei von etablierten Prozessen durch die Konzentration auf die ingenieurwissenschaftliche Perspektive. Die formalen Ansätze des Software Engineerings, die bei kontinuierlichen Systemen schwierig einsetzbar sind, werden angepasst und für technische Systeme genutzt. Dabei wird auf eine konsistente Durchgängigkeit Wert gelegt. Die Systematik wird durch Analyseverfahren unterstützt, die zu verschiedenen Phasen des Entwurfs intelligenter technischer Systeme Rückschlüsse auf die Anforderungen zulassen und helfen, diese umzusetzen. Zur Evaluierung wird die Systematik an einem Beispiel aus der industriellen Praxis angewendet.

Abstract

Dealing with complexity, interface diversity and domain diversity in connected and emergent systems can succeed via a superordinate view and guided processes. Technical requirements are of crucial importance in every phase of development. They are suitable as a goal definition and test criteria during system synthesis, analysis and integration. In this thesis, a model, based on four levels of requirements, is provided. Therein, technical requirements are successively and systematically detailed. Starting with goals, a distinction is made between function-oriented requirements, domain-spanning and domain-specific requirements. This representation is suitable for further system analyzes with regard to the consistency and completeness of requirements. The approach is characterized by focusing on the engineering perspective and can therefore be distinguished from established processes. The concepts of formal approaches from the software engineering domain, which are difficult to implement in continuous systems, are adapted, and used for the holistic development of intelligent technical systems. Consistency is targeted by a systematology. This is supported by analytical methods that allow conclusions about the requirements at various stages of the development of intelligent technical systems and help to realize them. To evaluate the systematology, it is applied to an industrial practice example.

Inhaltsverzeichnis

Systematik zur Evolution technischer Anforderungen

Inhaltsverzeichnis

1	1.1	Motivation	1
	1.1	Motivation	3
	1.2	Zielsetzung	5
	1.3	Vorgehensweise	5
2	Eino	rdnung und Grundlagen	9
	2.1	Ausrichtung der Arbeit	9
	2.2	Intelligente mechatronische Systeme	10
		2.2.1 Struktur mechatronischer Systeme	12
		2.2.2 Klassifizierung mechatronischer Systeme	13
		2.2.3 Der Geldautomat als Beispiel eines mechatronischen Systems .	14
	2.3	Entwurfstechnik mechatronischer Systeme	15
		2.3.1 Konzept des Systems Engineering	16
		2.3.2 Entwurfstechnik: Kontext und Begriffe	18
		2.3.3 Modellbasierte Entwicklung: V-Modell	20
	2.4	Modellbildung und mechatronische Analyse	22
		2.4.1 Grundgedanken und allgemeine Einordnung	22
		2.4.2 Modellierungsziel	24
		2.4.3 Modellierungsarten dynamischer Systeme	26
	2.5	Anforderungen in konventionellen Entwurfsprozessen	29
		2.5.1 Anforderungen aus der Sicht klassischer Ingenieure	30
		2.5.2 Anforderungen aus der Sicht des Software Engineering	32
3	Prob	elemanalyse und Stand der Wissenschaft	37
	3.1	Allgemeine Problembeschreibung	37
	3.2	Problemorientierte Analyse	39
	0	3.2.1 Interdisziplinäre Entwicklungsumgebungen	41
		3.2.2 Diskrete Modelle vs. kontinuierliche Systeme	42
		3.2.3 Konsistenz in Entwicklungsprozessen	42
		3.2.4 Emergente Systeme	43
	3.3	Problemabgrenzung und Anforderungen an eine Systematik	44
	0.0	3.3.1 Problemabgrenzung	44
		3.3.2 Anforderungen an eine Systematik	46
	3.4	Stand der Wissenschaft	48
		3.4.1 Requirements Engineering Frameworks	48
			. •

II Inhaltsverzeichnis

		3.4.2 Verfahren zur Bewertung von Anforderungen3.4.3 Entwicklungsmethodiken und -umgebungen	. 56
		3.4.4 Normen und Standards	
		3.4.5 Anforderungsorientierte Methoden	
	3.5	3.4.6 Toolunterstützung	
	3.5	Handlungsbedarf	
		3.5.2 Grad der Übereinstimmung	
4	Svn	these technischer Anforderungen	. 75
	4.1	Anforderungsebenen im Entwurfsprozess	
		4.1.1 Abgrenzung unterschiedlicher Anforderungsebenen	
		4.1.2 Einordnung in das V-Modell	
	4.2	V-Modell: Systemkonzipierung	. 81
		4.2.1 Ziele	. 81
		4.2.2 Funktionsorientierte Anforderungen	
	4.3	V-Modell: Transition	
		4.3.1 Domänenübergreifende Anforderungen	
		4.3.2 Domänenspezifische Anforderungen	
	4.4	V-Modell: Systemintegration	
		4.4.1 Testverfahren	
	4 5	4.4.2 Testumgebungen: Prototypen und Prüfstände	
	4.5	Weitere Eigenschaften der Durchgängigkeit	
		4.5.1 Dokumentation von Anforderungen	
		4.5.2 Dei Alisatz IIII Lebeliszykius	. 90
5	Ana	lyseverfahren für Anforderungen	
	5.1	Problemunterscheidung	
		5.1.1 Herausforderungen an Anforderungen	
		5.1.2 Hypothesen und Annahmen	
	5.2	Analyseverfahren und Werkzeuge	
		5.2.1 Funktionsorientierte Analyse	
	- 0	5.2.2 Anforderungsanalyse mittels Modellbildung	
	5.3	Anforderungsindikator: N3 Matrix	
		5.3.1 λ-Rank5.3.2 N3 Singulärwertzerlegung	
		3.3.2 No Singular wertzeriegung	. 123
6		rendung und Bewertung	
	6.1	Probleme und Ziele eines robusten Bargeldhandlings	
	6.2	Synthese der Anforderungen	
		6.2.1 Ziele	
		6.2.2 Funktionen	
		6.2.3 Domänensbergreifende Anforderungen	
		6.2.4 Domänenspezifische Anforderungen	
	6.3	Analyse der Anforderungen am Beispiel einer industriellen Lösung.	
	0.0	6.3.1 Funktionsorientierte Analyse	

Inhaltsverzeichnis

	6.3.2 Modellbasierte Analyse	ŀ6 ŀ7
7	Zusammenfassung und Resümee157.1 Bewertung der Syntheseverfahren157.2 Bewertung der Analyseverfahren157.3 Bewertung der Systematik157.4 Nutzenevaluierung157.5 Problemlösungskapazität157.6 Erweiterbarkeit der Lösung167.7 Resümee16	53 54 55 56 58 50
Lite	aturverzeichnis	3
An	ang	
A 1	Begriffe und Zusammenhänge	'9
A2	Übersicht ausgewählter Methoden 18 A2.1 Methoden zur Ermittlung von Zielen 18 A2.2 Methode zur Darstellung des Lastenheftes 18	3
А3	Mathematische Grundlagen des Kompensationsreglers	15
A 4	Elemente der Bondgraphen	7
A 5	Ergänzende Bilder der Anwendung	1

Abkürzungsverzeichnis V

Abkürzungsverzeichnis

3D dreidimensional.

A Anforderung.

AMS Autonomes mechatronisches System.

AROR Approval Rate of Requirements.

ATM Automated Teller Machine.

bzgl. bezüglich.

CAD Computer Aided Design.

CMMI Capability Maturity Model Integration.

CSV Comma-separated values.
CTP Critical Testing Program.

DAE Differential algebraische Gleichung.

DGL Differentialgleichung.
DIN Deutsche Industrienorm.
DoC Density of Connections.

DoCoAC Degree of Coverage of Acceptance Criteria.

ECU Electronic Control Unit.

EMV Elektromagnetische Verträglichkeit.

EN Europäische Norm.

ENG Engineering.

ENTIME Entwurfstechnik intelligente Mechatronik.

etc. et cetera.

F Funktion.

FEM Finite Elemente Methode.

FMEA Fehlermöglichkeits- und -einflussanalyse.

ggf. gegebenenfalls.

GMA Gesellschaft Mess- und Automatisierungstechnik.

GPS Global Positioning System.

HiL Hardware-in-the-Loop. HIS Herstellerinitiative Software.

HoQ House of Quality.

HW Hardware.

ID Identifikation.

IEC International Electrotechnical Commission.

IEEE Institute of Electrical and Electronics Engineers.
INCOSE The International Council on Systems Engineering.
ISO International Organisation for Standardization.

IT Informationstechnik.

KFZ Kraftfahrzeug.

KPI Key Performance Indikator.

MBSE Modelbased Systems Engineering.
MFM Mechatronisches Funktionsmodul.

MiL Model-in-the-Loop.
MKS Mehrkörper System.

Mrd. Milliarden.

NASA National Aeronautics and Space Administration. NoACR Number of Approved Consistent Requirements.

NoDAC Number of Defined Acceptance Criteria.

NoR Number of Requirements.

OMG Object Management Group.

PAM Prozessassessmentmodell.

Prio Priorität.

PRM Prozessreferenzmodell.

PU Polyurethan.

QFD Quality Function Deployment.

R Requirement.

RAMI Referenzarchitekturmodell Industrie 4.0.

RE Requirements Engineering.

ReqIF Requirements Interchange Format.

Regt Requirement in der SysML.

RFLP Requirements - Functional - Logical - Physical.

S. Seite.

SE Systemelement.
SiL Software-in-the-Loop.

SoS Systems of Systems Engineering.

SPICE Software Process Improvement and Capability Determination.

SW Software.

SysML System Modeling Language.

u.a. unter anderem.

UML Unified Modeling Language. USA United States of America.

Abkürzungsverzeichnis VII

VDE Verband der Elektrotechnik, Elektronik und Informationstechnik.

VDI Verein Deutscher Ingenieure.

vgl. vergleiche.

VHDL Very High Speed Integrated Circuit Hardware Description Language.

VMS Vernetztes mechatronisches System.

XiL X-in-the-Loop.

XML Extensible Markup Language.

z.B. zum Beispiel.

z.T. zum Teil.

Symbol verzeichnis IX

Symbolverzeichnis

Name	Beschreibung	Einheit
\overline{D}	Variable der Polkonfiguration	[-]
$F_o(s)$	Übertragungsfunktion des offenen Regelkreises	[-]
F	Kraft	[N]
G(s)	Reglerübertragungsfunktion	[-]
L_{M}	Induktivität	[H]
M_A	Antriebsmoment der Abzugsrolle	[N m]
M_G	Antriebsmoment der Gegenlaufrolle	[N m]
$M_{\lambda_{max}}$	Eigenvektor zum maximalen Eigenwert	[-]
M_{λ}	Eigenvektor	[-]
M	Eigenvektoren	[-]
<i>N</i> 3	N3 Matrix	[-]
N_i	Nenner	[-]
P	Leistung	[W]
R_M	Elektrischer Widerstand	$[\Omega]$
U_{M}	elektrische Spannung	[V]
U	Matrix der Linkssingulärvektoren	[-]
V	Matrix der Rechtssingulärvektoren	[-]
Z_i	Zähler	[-]
Γ	Stabilitätsgebiet	[-]
Σ	Matrix der Singulärwerte	[-]
α	Variable der Polkonfiguration	[-]
$\ddot{\varphi}_A$	Winkelbeschleunigung der Abzugsrolle	[rads ⁻²]
$\ddot{\varphi}_G$	Winkelbeschleunigung der Gegenlaufrolle	[rads ⁻²]
$\ddot{x}_{N_{oben}}$	Beschleunigung der oberen Notenhälfte	$\left[\mathrm{ms^{-2}}\right]$
$\ddot{x}_{N_{unten}}$	Beschleunigung der unteren Notenhälfte	$\left[\mathrm{ms^{-2}}\right]$
∞	Unendlich	[-]
$ N3 _{F}$	Frobeniusnorm der N3 Matrix	[-]
ϕ	Magnetischer Fluss	[Wb]
$arphi_A$	Koordinate der Abzugsrolle	[rad]
c_V	Federstifigkeit	$\left\lfloor \frac{N}{m} \right\rfloor$
c	Federsteifigkeit	$\left[\frac{N}{m}\right]$
d	Dämpfungsfaktor	$\left[\frac{kg}{s}\right]$
e	Effort	[-]
f	Flow	[-]
i	Index	[-]
\dot{j}	Index	[-]
k_M	Maschinenkonstante	[-]
k	Parameter der Polkonfiguration	[-]
m_i	Masse	[kg]
m	Anzahl	[-]

X Kapitel 0

Name	Beschreibung	Einheit
n_{soll}	Solldrehzahl	rads ⁻¹
n	Anzahl	[-]
p(s)	Polynomfunktion	[-]
r_A	Radius der Abzugsrolle	[m]
r_G	Radius der Gegenlaufrolle	[m]
rng	Rang	[-]
r	Parameter der Polkonfiguration	[-]
t	Zeit	[s]
\mathcal{X}	Koordinate	[-]
У	Koordinate	[-]
${F}_{N_{AN}}$	spezifische Normalkraft	[N]
${F}_{N_{GN}}$	spezifische Normalkraft	[N]
${F}_{N_{NN}}$	spezifische Normalkraft	[N]
$F_{\mathit{R}_{\mathit{AN}}}$	spezifische Reibkraft	[N]
$F_{R_{GN}}$	spezifische Reibkraft	[N]
F_{R_M}	spezifische Reibkraft	[N]
$F_{R_{NN}}$	spezifische Reibkraft	[N]
$F_w(s)$	Führungsübertragungsfunktion	[-]
J_A	Trägheitsmoment der Abzugsrolle	[kgm ²]
J_B	Trägheitsmoment der Gegenlaufrolle	[kgm ²]
J_{M}	Trägheitsmoment des Motors	$\left[\mathrm{kgm}^{2}\right]$
$N3_{RF}$	N3 Matrix der Anforderungen und Funktionen	[-]
λ_i	Eigenwert	[-]
μ_{AN}	spezifischer Reibungskoeffizient	[-]
μ_{NN}	spezifischer Reibungskoeffizient	[-]
$arphi_G$	Koordinate der Gegenlaufrolle	[rad]
μ_{GN}	spezifischer Reibungskoeffizient	[-]

Einleitung

1 Einleitung

Anforderungen sind ein unverzichtbarer Bestandteil technischer Entwicklungen. Sie ermöglichen eine zielgerichtete Entwicklung. Sie sind integriert in alle Entwicklungsprozesse und bestimmen maßgeblich das Ergebnis einer Entwicklung. Daher sind sie entscheidend dafür, ob ein System erfolgreich ist oder nicht.

Die steigende Komplexität in modernen Entwicklungen führt zu einer Veränderung und Anpassung dieser Prozesse. Zwar gibt es wesentliche Fortschritte in der Interpretation von Anforderungen, vor allem im softwaretechnikorientierten Requirements Engineering, jedoch sind die Anwendungen von Anforderungsmethoden auf die traditionellen bzw. klassischen ingenieurwissenschaftlichen Disziplinen überschaubar. Die klassische disziplinspezifische¹ Vorgehensweise hat ihren Platz innerhalb übergeordneter Prozesse gefunden. Die traditionellen Anforderungsprozesse können sicherlich auch dort eingeordnet werden. Dabei wird jedoch die Verknüpfung der Anforderungen mit dem übergeordneten System bisher nicht hinreichend berücksichtigt. Die Prozesse des Ingenieurwesens sind längst in einem größeren Kontext etabliert, benötigen jedoch für eine zufriedenstellende Entwicklung klare Anforderungen. Die Herleitung der klaren Anforderungen muss folgerichtig ebenfalls in einem übergeordneten Systems Engineering Kontext eingebettet werden.

1.1 Motivation

Die Evolution mechatronischer Systeme hin zu stark vernetzten *intelligenten technischen Systemen* und *cyber physical systems* ist geprägt durch eine besonders hohe Komplexität und einen steigenden Softwareanteil. Im Vergleich zu Entwicklungen und Projekten vergangener Zeiten scheinen die Systeme von morgen unsagbar groß. Zum einen liegt dies an der enormen Funktionsvielfalt, welche in modernen Systemen vorliegt und zum anderen an dem spezifischen Wissen, welches für diese Systeme angewendet werden muss. Konnten früher einzelne Personen ein Produkt noch quasi vollständig verstehen und überblicken, so ist dies heutzutage bedeutend schwieriger. Dieser Herausforderung wird seit geraumer Zeit Rechnung getragen, sodass große Systeme mit Methodiken und Prozessen beherrschbar gemacht wurden. Die Komplexität in diesen Systemen ist bedingt und geprägt durch eine hohe Schnittstellenvielfalt und eine zunehmende Hierarchisierung der Systemelemente und Funktionalitäten. Zusätzlich gibt es nicht-offensichtliche Wechselwirkungen zwischen Systemelementen und auch zwischen Funktionen. Diese systematische Entwicklung brachte enorme Vorteile mit sich. Doch nicht alle Projekte, beziehungsweise Entwicklungen oder Produkte können vollends davon profitieren.

Berühmte Negativ-Beispiele finden sich interessanterweise in der stark durch konventionelles Systems Engineering geprägten Luftfahrtindustrie, was allerdings auch an der Öffentlichkeitswirksamkeit liegen dürfte. Der Airbus A400M sowie der Boeing 787 Dreamliner haben sehr lange Entwicklungszeiten durchgemacht, weisen dennoch unvorhergesehene Fehler auf. Sicherlich spielt die selbst für die Kategorie dieser Systeme sehr hohe

¹Im Verlauf dieser Arbeit werden die geläufigen Begriffe Disziplin und Domäne synonym verwendet.

Komplexität verstärkend mit hinein. Diese Beispiele scheinen zu zeigen, dass selbst konsequentes Systems Engineering nicht vor Fehlern schützt und weiterhin Handlungsbedarf besteht. Auch der Konsumelektronikmarkt bleibt nicht von unvorhergesehenen Fehlern verschont. Bei der Xbox 360 der Firma Microsoft beispielsweise kann es vorkommen, dass bestimmte Lötstellen durch die Hitze im Betrieb schmelzen und nicht mehr richtig leiten. Gegenstand kürzlicher Debatten ist auch das Gewehr G36, welches durch scheinbar mangelnde Präzision in die Kritik geriet. Das Verteidigungsministerium und die Firma Heckler und Koch bilden die beiden Parteien dieser Auseinandersetzung. In einem Prozess im Landgericht Koblenz wurde festgestellt, das Gewehr funktioniere anforderungsgerecht. Hieraus lassen sich jedoch zwei Gedanken ableiten: Zum einen liegt hier aus Sicht des Verteidigungsministeriums ein Versäumnis vor, die Anforderungen zu aktualisieren und zum anderen lieferte Heckler und Koch nicht entsprechend der eigentlichen, ggf. impliziten Anforderungen seines Kunden. Ohne eine Bewertung dieses Sachverhaltes vorzunehmen, wird durch den entstandenen Schaden und Imageverlust auf beiden Seiten die Bedeutung von Anforderungen deutlich. Nun scheint eine retrospektive Betrachtung solcher Ereignisse oft sehr belehrend, was an dieser Stelle nicht beabsichtigt ist. Eine gewisse Demut vor technischen Systemen und ihrer Komplexität schadet nicht. Das hindert jedoch nicht, daraus zu lernen und zu versuchen, Lösungen anzubieten, die mit solchen Problemen fertig werden.

In vergangenen Zeiten wurden großzügige Sicherheitsfaktoren berücksichtigt, um Risiken zu minimieren. Es ist zu vermuten, dass der Sicherheitsfaktor von 3 jedem Ingenieur² ein Begriff ist. In modernen Entwicklungen können Sicherheitsfaktoren hingegen hinderlich sein. Aus wirtschaftlicher Sicht sind unnötige Erweiterungen offensichtlich nachteilig. Aber auch aus technologischer Sicht kann sich ein Sicherheitsfaktor negativ auswirken – es können sich recht einfach Masse- und somit Trägheitsprobleme vorgestellt werden. Zur Abschätzung der Notwendigkeit von Sicherheiten, sollten tendenziell wichtige von unwichtigen Systemelementen abgegrenzt und entsprechend entwickelt und ausgelegt werden. Um möglichst frühzeitig Erkenntnisse über solche Fragestellungen zu erlangen, werden Simulationen an Modellen durchgeführt. Die immer frühere Integration von Modellen in den Entwurf der technischen Systeme steht hier in Konflikt mit dem Nutzen solcher Modelle. Frühe Modelle haben naturgemäß eine geringere Basis an Informationen als solche in ausgereiften Entwicklungsphasen, können als Abschätzungen aber von großer Bedeutung sein. Dabei setzen unterschiedliche wissenschaftliche Disziplinen den Fokus nicht immer gleich: Der Begriff Modell wird in den verschiedenen Fachdisziplinen im Speziellen durchaus unterschiedlich gesehen.

Im Sinne des Systems Engineering ist zu Beginn eines Entwicklungsprozesses das gemeinsame Verständnis der Ziele und des Systems ein entscheidender Faktor. Anforderungen können alle Voraussetzungen dafür erfüllen, werden jedoch ebenfalls nicht einheitlich betrachtet. Dadurch bleibt großes Potential zum einheitlichen Systemverständnis ungenutzt. Auf subtile Art und Weise wird die uneinheitliche Betrachtung von Anforderungen im Kontext *Industrie 4.0* deutlich. Im Glossar des VDI GMA³ Fachausschuss 7.21 *Industrie 4.0* werden Begriffe rund um Industrie 4.0 definiert und in ein gedankli-

²Beschreibungen, die sich auf Personengruppen beziehen und bei welchen nur die m\u00e4nnliche Form gew\u00e4hlt wurde, sind nicht geschlechtsspezifisch gemeint. Dies geschah ausschlie\u00ddlich aus Gr\u00fcnden der Lesbarkeit.

³VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik

Einleitung 3

ches Konstrukt gebracht (vgl. Anhang A1). Dabei steht gerade der Begriff *Anforderungen* auffällig isoliert am Rande des Schaubildes. Besonders die fehlenden Verknüpfungen zu anderen Elementen dieses gedanklichen Konstruktes ist markant.

1.2 Zielsetzung

Entwicklungen funktionieren. Das kann seit geraumer Zeit an der festzustellenden Innovationskraft deutlich erkannt werden. Technische Entwicklungen funktionieren mit den seit Jahren bekannten Hilfsmitteln der Ingenieure. Der Einzug der interdisziplinären Produkte in die Forschungslandschaft und Industrie förderte im Laufe der Zeit unzählige Forschungsarbeiten, z.B. für mechatronische Systeme, zu Tage. Die Konnektivität, Datenverfügbarkeit und fortschreitende Möglichkeiten der Kommunikation bringen immer komplexere Systeme mit sich. Dabei werden sowohl große Chancen offengelegt, als auch Probleme deutlich. Das Fundament der neuen Probleme bilden auch vererbte Schwierigkeiten, die aus dem in früherer Zeit üblichen und angebrachten monodisziplinären Vorgehen resultieren.

Die Systeme von morgen und auch von heute profitieren von der Symbiose der Disziplinen. Das Systems Engineering ist nicht gänzlich unbeteiligt daran. Das Zusammenwirken entfaltet sein volles Potential jedoch noch nicht. Einige Schwierigkeiten begleiten den an sich unaufhaltsamen Weg. Alleine die unterschiedlichen Sichtweisen der Ingenieure und der Informatiker zeigen eine Lücke auf dem Weg zum gemeinsamen Verständnis. Für Informatiker ist es selbstverständlich, dass ein Debugging Prozess durchlaufen wird, wohingegen die Ingenieure versuchen, Fehler von Anfang an auszuschließen. Das mag daran liegen, dass die Sichtweise im traditionellen Ingenieurwesen ein abstraktes Problem zu einer konkreten Lösung überführt, wohingegen aus der Sicht der Informatik ein reales Problem auf abstrakte Weise gelöst wird [Uys16].

Im Zuge dieser Arbeit wird ein Weg beschrieben, konsistent von den anfänglichen Anforderungen bis hin zum fertigen System zu gehen. Dabei wird es kontinuierlich zu einer Verfeinerung der Anforderungen kommen. Dies muss gewährleistet sein, um die richtigen Schlüsse aus den Anforderungen zu erlangen. Wenn ein komplexes System zusammengesetzt wird, werden Wechselwirkungen auftreten, welche nicht offensichtlich sind. Die Ergebnisse dieser Arbeit unterstützen dabei, solche Wechselwirkungen transparent zu machen. Die wesentlichen Fragestellungen sind diejenigen, die auf der Konsistenz in Entwicklungsprozessen aufbauen. Typische Entwicklungsmethodiken zeigen eine gewisse Durchgängigkeit auf, benötigen jedoch ständige Kontrolle, um erfolgreich in der Praxis angewendet werden zu können.

Zuerst sollte differenziert werden, dass sich je nach Perspektive die Fragen verändern. Wird primär die Erhebung von Anforderungen betrachtet, stellen sich andere Fragen, als wenn es beispielsweise um die Überprüfung bzw. den Test der Erfüllung der Anforderungen geht. In dieser Arbeit wird eine Systematik für technische Anforderungen (T-REQS) vorgestellt, welche durch den gezielten Einsatz von Methoden in einem geführten Prozess die Fragen ganzheitlich betrachtet. Im Vordergrund steht dabei die Frage nach einem passenden anforderungsgerechten Entwicklungsprozess, welcher auch als Verbindung dient, um die Durchgängigkeit von der Erhebung zu den Tests zu erhöhen.

In einer Umgebung, in welcher technische Systeme immer umfangreicher werden, müssen Entwickler also interdisziplinär zusammenarbeiten: Ein gemeinsames Verständnis des Vorhabens ist von elementarer Bedeutung für den Erfolg der Systementwicklung. Anforderungen sind vorherbestimmt, diese Aufgabe zu erfüllen. Die verbreiteten Entwicklungsmethoden berücksichtigen dies, haben jedoch Schwierigkeiten mit einer durchgängigen Berücksichtigung während der unterschiedlichen Entwicklungsphasen. Ein Mangel ist speziell die Gewährleistung des Grades der Konsistenz beim Übergang von den in frühen Entwicklungsphasen genutzten Systemmodellen zu dynamischen Simulationsmodellen. Ein weiteres nicht gelöstes Problem ist die Unterschiedlichkeit der Herangehensweisen der verschiedenen Disziplinen. Z.B. behandeln traditionelle Ingenieurwissenschaften dieses Konsistenzproblem anders als Softwareentwickler. Die formalen Methoden, welche in der Softwareentwicklung eingesetzt werden, funktionieren hervorragend für diskrete Modelle von Systemen. Methoden für kontinuierliche, ggf. echtzeitfähige Systeme, wie sie im Fokus traditioneller Ingenieure stehen, sind dagegen nicht so reif. Diese Arbeit zielt darauf ab, die Verbindung zwischen gezielten und speziellen Ingenieurstätigkeiten und diesen übergeordneten generischen Vorgehensmodellen herzustellen.

Klassische Ingenieurstätigkeiten finden Lösungen für gezielte Fragestellungen und technische Probleme. Wenn ein Problem klar definiert ist, werden Ingenieure immer in der Lage sein, dieses zu lösen. Dafür gibt es jahrelange Erfahrung aus unterschiedlichen Bereichen. Hier kann es sich zum einen um Mechanik handeln, jedoch auch um Thermodynamik, Fluiddynamik, Antriebstechnik, Fahrzeugtechnik, Regelungstechnik, Steuerungstechnik oder weitere. Werden die Problemstellungen allerdings komplexer bis hin zu sogenannten emergenten Systemen, werden die klassischen Methoden für sich allein stehend nicht unbedingt und zwangsläufig zum Erfolg führen.

Für die Verbindung dieser Disziplinen, Domänen und Branchen bieten sich natürlich auf den ersten Blick die Methoden des Systems Engineering an, welche sehr generisch und allgemein anwendbar sind (vgl. [KS11]). Dass die Anwendung der entsprechenden Methoden sehr sinnvoll ist, steht außer Frage. Es kann jedoch in Systemen zu Wechselwirkungen der Teilsysteme kommen, die mit den üblichen Systems Engineering Prozessen nicht gelöst werden können. Hierfür sind neuartige Ansätze notwendig. Wird bei einem solchen Prozess konsequent gedacht, sollte in dem entsprechenden Entwicklungsprozess möglichst früh damit begonnen werden. Im sogenannten V-Modell werden Anforderungen gleich zu Beginn berücksichtigt, gelten jedoch als bereits vollständig. Die Annahme der Vollständigkeit ist hingegen nicht zwangsläufig - zu Beginn einer Entwicklung ist es vielmehr eher unwahrscheinlich, dass Anforderungen vollständig sind. Der komplette fortführende Prozess im traditionellen Systems Engineering beruht genau auf diesen gegebenenfalls unvollständigen Anforderungen. Daher werden auch mit den generisch anwendbaren Methoden des Systems Engineering letztlich Fehler auftreten können, die eventuell hätten behoben werden können, führte man sich die Anforderungsproblematik detaillierter vor Augen.

Nicht nur die Systeme an sich, auch die Arbeit an diesen verändert sich. In einigen Jahren werden die angesprochenen Veränderungen der Technologien große Auswirkungen auch auf die Entwickler dieser Technologien haben. Es werden neue Jobs in Hybridindustrien entstehen, welche noch höhere Anforderungen an die Interdisziplinarität mit sich bringen. Die Nachfrage nach IT-Wissen wird in allen Bereichen, wie z.B. Sicherheit, Datenanalyse oder Cloud Computing steigen und der sich so verändernde Arbeitsmarkt benötigt neue

Einleitung 5

Formen der Kollaboration [Wor15]. Die einzelnen Disziplinen werden sich öffnen und zum Teil verschmelzen. Dies erfordert die Fähigkeit, sich zu verständigen: Schnittstellenfähigkeiten werden wichtiger, sodass hohe Fachkenntnisse auch auf anderen Fachgebieten erlangt werden müssen, um sich verständigen zu können. Auch erfordert diese Entwicklung eine erhöhte soziale und kommunikative Kompetenz sowie die Fähigkeit, die Zusammenarbeit verschiedener Disziplinen zu managen [Lan17] [Bet17]. Letztlich wird ein Produkt so nicht mehr bis in Perfektion entwickelt, sondern zu einem *minimal viable product*. Das erfordert jedoch eine erhöhte Berücksichtigung von Kunden- und Anwenderwünschen [Bet17]. Somit kann gerade die Integration von Anforderungsprozessen in interdisziplinäre Entwicklungsprozesse ein Hilfsmittel sein, um diesen Herausforderungen zu begegnen.

1.3 Vorgehensweise

Zuerst sei ein kurzer Einblick in die Wissenschaftstheorie gestattet, um das Vorgehen aus wissenschaftlicher Sicht darzulegen. Der wissenschaftliche Erkenntnisprozess sollte möglichst deduktiv gestaltet werden.

Deduktives Vorgehen zeichnet sich dadurch aus, dass Schlussfolgerungen als logische Konsequenz gezogen werden können (vgl. Bild 1-1). In der Mathematik ist ein klassisches Beispiel für ein deduktives Vorgehen der Beweis durch vollständige Induktion. Die Bezeichnung erscheint verwirrend, da Induktion eine andere wissenschaftliche Herangehensweise bedeutet. Nichtsdestotrotz helfen Formalisierungen bei deduktiven Prozessen – in der realen Welt sind Formalisierungen aber nicht immer geeignet. Dies stellt viele ingenieurwissenschaftliche Arbeiten vor ein Dilemma, befassen sie sich doch zu einem großen Teil mit Problemen der realen Welt. Selbstverständlich existieren aber auch hier Arbeiten mit deduktiven Ansätzen. In der Physik etablierten sich ebenfalls diese zwei Richtungen: Induktives Vorgehen, bei welchem aus der Experimentalphysik produziertes Wissen als Ausgangspunkt der Modellentwicklung verstanden wird, und deduktives Vorgehen, welches zunächst eine mathematisch formulierte Theorie sucht.

Die Rechtfertigung der Schritte der Induktion als wissenschaftliches Forschungsinstrument erfolgt über einige Argumente aus Reschers Buch *Induktion* [Res87]: Ein induktiver Schluss sei nötig, wenn die gegebene Informationslage unzureichend ist, um deduktiv schließen zu können. Demnach ist die induktiv angebrachte Antwort diejenige,

"die die verfügbaren Informationen auf plausibilistisch optimale Art und Weise ergänzt." [Res87, S. 19]

Die Induktion sucht somit nicht die beste Erklärung, sondern die beste Systematisierung [Res87]. Das bedeutet, das zur Verfügung stehende Wissen wird transzendiert. Dies geschieht nicht mit der bestmöglichen Antwort, sondern mit der besterreichbaren Antwort zum Allgemeinen durch das Besondere [Res87].

Vollständige Informationen werden in den meisten Fällen nicht verfügbar sein, jedoch sollen Entscheidungen fundiert getroffen werden. Dies ist eine Situation, welche in Bezug auf Anforderungen – gerade zu Beginn einer Entwicklung – oft auftritt. Entscheidungen werden in der Praxis nicht getroffen, weil Informationen nicht vollständig oder nicht gesichert sind. Das Ziel ist in manchen Fällen dennoch genau eine solche Entscheidung,

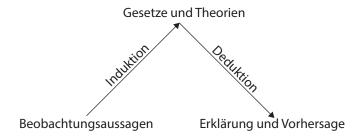


Bild 1-1: Induktion und Deduktion, abgewandelt nach [CAGB07]

auch wenn die Entscheidungen im weiteren Entwicklungsverlauf angezweifelt werden können.

Rescher gibt dazu allgemein an:

"Gegenstand allen Erkenntnistrachtens ist die Suche nach der Wahrheit (und nicht nur das Vermeiden von Irrtümern!)" [Res87, S. 23]

Auf Basis vorhandener Informationen muss ein Rahmenwerk geschaffen werden, um Entscheidungen so zu treffen, dass sie zu diesem Zeitpunkt die besterreichbaren sind, ohne die Entscheidungen unverhältnismäßig in die Länge zu ziehen. Es soll die Hypothese gelten, dass diesen Entwicklungsproblemen mit gesteigerter Konsistenz und Systematisierung von Anforderungen begegnet werden kann. Dabei bietet die Vernetzung immer weiterer Domänen und die Einbindung von Entwicklern aus vielen Disziplinen großes Potential. Um das Potential verfügbar zu machen, bedarf es jedoch moderner Methoden, die sich mit den neuen Problemen auseinander gesetzt haben und sie beherrschbar machen.

Diese Arbeit ist in sieben Kapitel gegliedert. In **Kapitel 2** wird die inhaltliche Ausrichtung der vorliegenden Arbeit spezifiziert und in den Kontext intelligenter mechatronischer Systeme gebracht. Es werden Begriffe sowohl der betrachteten Systeme als auch der Entwicklungswerkzeuge erläutert und grundsätzliche Zusammenhänge beschrieben. Dies sind Entwurfsmethodiken und Methoden, wie die Modellbildung. Abschließend wird eine Gegenüberstellung von Anforderungen einmal im Lauf der Zeit und einmal in unterschiedlichen Disziplinen vorgenommen.

Das **Kapitel 3** beginnt mit einer weitergehenden Problemanalyse, die sich auf die Herausforderungen der Entwicklung intelligenter Systeme konzentriert. Die erlangten Erkenntnisse werden abgegrenzt und aus ihnen werden Anforderungen an eine Systematik zur Evolution technischer Anforderungen abgeleitet. Der Stand der Wissenschaft wird in den Bereichen Requirments Engineering Frameworks, Verfahren zur Bewertung von Anforderungen, Entwicklungsmethodiken und -umgebungen, Normen und Standards, anforderungsorientierte Methoden und Toolunterstützung beleuchtet und mit den abgeleiteten Anforderungen verglichen.

Der erste Teil einer Systematik zur Evolution technischer Anforderungen wird in **Kapitel 4** beschrieben. Es werden vier unterschiedliche Anforderungsebenen abgegrenzt und in das V-Modell eingeordnet. Entlang des Entwicklungsprozesses gemäß des V-Modells werden Ziele, funktionsorientierte Anforderungen, domänenübergreifende Anforderun-

Einleitung 7

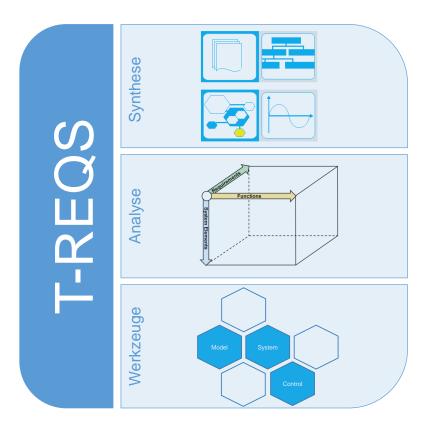


Bild 1-2: Die Elemente der Systematik T-REQS, Technical-REQuirementS

gen und domänenspezifische Anforderungen bis hin zu Testverfahren erläutert und mit Werkzeugen zur Steigerung der Durchgängigkeit angereichert.

Analyseverfahren für Anforderungen als wichtiger zweiter Bestandteil der Systematik werden in **Kapitel 5** dargelegt. Dafür werden mittels einer Problemunterscheidung entsprechend der Herausforderungen an Anforderungen Analyseziele ermittelt. Zur Bewältigung dieser werden einerseits Analyseverfahren und Werkzeuge genutzt, andererseits wird ein Indikator zur Bewertung und Priorisierung von Anforderungen vorgestellt.

Die Systematik wird in **Kapitel 6** anhand eines Anwendungsbeispiels evaluiert. Dafür werden Elemente der Entwicklung eines neuartigen Geldautomaten genutzt. Die Synthese der Anforderungen wird an einem akademischen Beispiel durchgeführt und die Analyse am Beispiel einer industriellen Lösung. Das **Kapitel 7** fasst die wesentlichen Inhalte dieser Arbeit zusammen und zieht ein Fazit. Im **Anhang** dieser Arbeit finden sich ergänzende Informationen zu Begriffen, Zusammenhängen, Methoden und Anwendungen dieser Arbeit.

2 Einordnung und Grundlagen

In diesem Kapitel wird der inhaltliche Kontext näher beleuchtet. Dazu werden die Ausrichtung der Arbeit konkretisiert und die wichtigsten Grundlagen vorgestellt. Zu Beginn wird die Ausrichtung der Arbeit in Beziehung zu angrenzenden Wissenschaftsbereichen gesetzt. Die darauf folgenden Abschnitte beinhalten einerseits wichtige Grundlagen und andererseits verdeutlichen sie die verschiedenen Themengebiete, mit denen sich diese Arbeit auseinandersetzt. Die Inhalte für dieses Kapitel basieren auf einem allgemein bewährten Stand der Wissenschaft. In Abschnitt 2.2 erfolgt eine Beschreibung genau der Systeme, die im Rahmen dieser Arbeit betrachtet werden. Entwicklungsprozesse dienen der methodischen Entwicklung solcher Systeme und werden in Abschnitt 2.3 beschrieben. Eine Ausführung zu Modellen und Modellbildung als Werkzeug innerhalb von Entwicklungsprozessen erfolgt in Abschnitt 2.4. Abschließend werden Anforderungen, als bedeutender Bestandteil dieser Arbeit, aus zwei verschiedenen Sichten betrachtet (Abschnitt 2.5). Neben den klassischen ingenieurwissenschaftlich geprägten Anforderungen, wird ebenfalls die aus dem Software Engineering geprägte Sicht auf Anforderungen erläutert.

2.1 Ausrichtung der Arbeit

Die wissenschaftlichen Disziplinen mit Berührungspunkten zu dieser Arbeit sind in übergeordneter Sicht die Mechatronik und wegen des steigenden Softwareanteils in mechatronischen Systemen auch die Softwaretechnik. Im Speziellen sind das Systems Engineering und das Software Engineering zu erwähnen, wobei das Systems Engineering übergeordnet steht. Das Requirements Engineering ist, aus wissenschaftlicher Perspektive betrachtet, mittlerweile softwareseitig einzugliedern und könnte als Teil des Systems Engineering interpretiert werden. Für die Anwendung und Umsetzung der Inhalte dieser Arbeit ist die Modellbildung dynamischer Prozesse relevant und die Regelungstechnik mit einigen Facetten. Es ist festzuhalten, dass in dieser Arbeit eine Systematik vorgestellt wird, welche ein bestehendes Vorgehensmodell anreichert und dafür neue Hilfsmittel bereitstellt. Eine ausgeprägte Durchgängigkeit innerhalb der Systematik wird mit Hilfe von Anforderungen erreicht. Dabei dienen Modelle explizit zur Ermittlung von passenden Anforderungen, aber auch zur Präzisierung und Überprüfung. Die dabei anfallenden Erkenntnisse über die Modellbildung werden der Systematik zurückgespielt und ergänzen diese.

Der Begriff *Entwicklung* lässt sich auf verschiedene Weisen interpretieren. In Bezug auf Systems Engineering kann Entwicklung wohl mit *Engineering* übersetzt werden. Eine für die vorliegende Arbeit besonders geeignete Definition wird von Kossiakoff und Sweet gegeben [KS11]. Entwicklung ist

"die Anwendung wissenschaftlicher Prinzipien für praktische Zwecke, wie den Entwurf, die Konstruktion oder die Bedienung effizienter und wirtschaftlicher Strukturen, Ausrüstungen und Systeme" [KS11, S. 3].

¹aus dem Englischen übersetzt

Zu beachten ist, dass sich eine Entwicklungssystematik durch ein Rahmenwerk und bestimmten Werkzeugen charakterisiert und sich von einer Entwicklungsmethodik unterscheidet. Eine Entwicklungsmethodik berücksichtigt zusätzlich denk-, arbeitspsychologische und organisatorische Aspekte [Dum10].

Ausgehend von der Detaillierung des *Systembegriffs* wird im Folgenden der Begriff *Entwurfstechnik* für diese Systeme beleuchtet. Anschließend folgt die grundlegende Beschreibung des hierfür wesentlichen Werkzeugs *Modellbildung* und eine Darstellung klassisch verwendeter *Anforderungen* im Entwurfsprozess. Die beiden für diese Arbeit schwergewichtigen Themen Anforderungen und Modellbildung werden im Rahmen des Systems Engineering und der Konsistenz in Entwicklungen beleuchtet.

2.2 Intelligente mechatronische Systeme

Hier handelt es sich um eine Unterart von Systemen. Nach der DIN 60050 [DIN14] ist ein System definiert als eine

"Menge miteinander in Beziehung stehender Elemente, die in einem bestimmten Zusammenhang als Ganzes gesehen und als von ihrer Umgebung abgegrenzt betrachtet werden." [DIN14, S. 21]

Für die Elemente gilt:

"Elemente eines Systems können natürliche oder künstliche Gegenstände ebenso wie Arten und Denkvorgängen und deren Ergebnisse (zum Beispiel Organisationsformen, mathematische Verfahren, Programmiersprachen) sein" [DIN14, S. 21].

Für die Abgrenzung gilt:

"Das System wird als von der Umgebung und von den anderen äußeren Systemen durch eine gedachte Hüllfläche abgegrenzt betrachtet, durch welche die Verbindung zwischen diesen Systemen und dem betrachteten System hindurchgehen" [DIN14, S. 21].

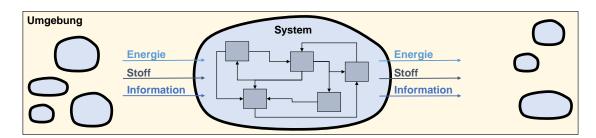


Bild 2-1: Schema eines Systems mit Ein- und Ausgängen, abgewandelt nach [Ehr09]

Ein System lässt sich auch grafisch wie in Bild 2-1 darstellen. Damit ist der Begriff soweit abstrahiert, dass selbst der ganze Planet Erde darunter fallen könnte. Erschwerend kommt hinzu, dass Systeme wiederum aus Systemen bestehen können. Das komplexe Zusammenspiel mehrerer Systeme wird im Forschungsgebiet des *System of Systems Engineering* untersucht [Jam09] [NLF+15]). Die Schwierigkeit des Forschungsgebietes *SoS*

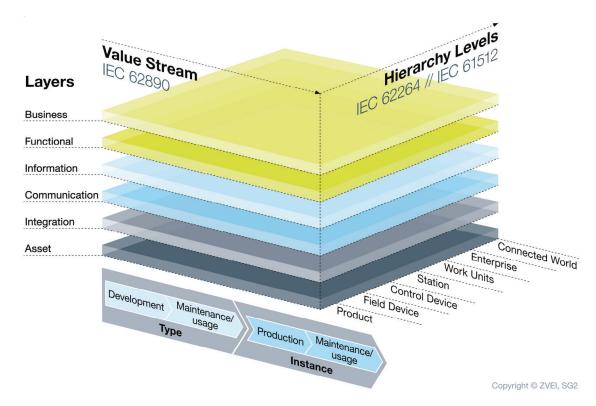


Bild 2-2: Referenzarchitekturmodell RAMI4.0 [ABD+15]

wird bereits an der Anzahl möglicher Definitionen deutlich. Nielsen et al. beschreiben dreißig Definitionen im Wandel der Zeit von 1956 bis 2015 [NLF+15]. Während in den Anfängen der Disziplin SoS noch vergleichsweise simpel als statische Strukturen offener Systeme aus unterschiedlichen Disziplinen² gesehen wurden [Bou56], entwickelt sich diese Definition aktuell zu multiplen heterogenen verteilten Systemen, die große und interdisziplinäre Probleme adressieren; sie ziehen mehrdeutige Anforderungen, unklare Systemgrenzen und unklare Schnittstellen nach sich³ [WRF+15].

Eine geeignete Sicht, um die praktische Relevanz der immer komplexer werdenden Systeme zu beschreiben bietet der Begriff *Industrie 4.0*. Industrie 4.0 beschreibt stark vereinfacht den Einzug der Digitalisierung in die industriellen Produkte und die Produktion. Durch die starke Vernetzung von Produkten entlang von Unternehmen, Organisationen, des Lebenszyklus oder der Entwicklung ist im Bereich der Mess- und Automatisierungstechnik ein Referenzarchitekturmodell eingeführt worden. *RAMI4.0* zeigt den Aufbau und die Arbeitsweise von Industrie 4.0 Komponenten (vgl. Bild 2-2). Dies geschieht mit Hilfe eines dreidimensionalen Gebildes, dem Industrie 4.0 Raum [ABD+15]. Wo früher die klassische Automatisierungspyramide zur Orientierung diente, benötigt es heutzutage also komplexere Beschreibungen, da die bekannten Methoden, Werkzeuge und Hilfsmittel nicht für alle Fälle leistungsfähig genug sind. Es bedarf jedoch nicht zwangsläufig einer revolutionären Neuentwicklung. *RAMI4.0* setzt an sinnvollen Stellen auf bestehenden Normen auf, um eine schnelle Handlungsfähigkeit zu erreichen [ABD+15]⁴. Die Bot-

²aus dem Englischen übersetzt

³aus dem Englischen übersetzt

⁴In einem anderen Kontext sagte Heinz Nixdorf: Technologischer Wandel erfolgt nicht durch Revolution, sondern durch Evolution, durch unendlich viele kleine Schritte, die man stetig tun muss.

schaft an dieser Stelle ist die sich verändernde Sicht auf Produkte und Systeme. Nicht nur das zu entwickelnde System bzw. die realen Gegenstände sollen demnach betrachtet werden. Es muss die Geschäftsschicht, die Funktionsschicht, die Informationsschicht, die Kommunikationsschicht, die Integrationsschicht und *ebenfalls* die Gegenstandsschicht berücksichtigt werden. Dies soll entlang des gesamten Lebenszyklus und innerhalb von Wertschöpfungsketten geschehen und alle Hierarchieebenen der funktionalen Zuordnungen des Produktes und seines Entstehungsprozesses berücksichtigen [ABD+15]. Der so aufgespannte Raum ist entsprechend groß und zeigt deutlich die steigende Komplexität und das Umfeld mit seinen Wechselwirkungen.

Zur genaueren Spezifikation der in dieser Arbeit betrachteten Systeme wird im Folgenden eine Klassifikation vorgenommen und die Struktur eines mechatronischen Systems erörtert. Beispiele verdeutlichen die Spezifikation des Systembegriffs für diese Arbeit. Eine weitreichende Beschreibung mechatronischer Systeme findet sich in der angegebenen Literatur [Bis02].

2.2.1 Struktur mechatronischer Systeme

Der Begriff Struktur steht für die Beziehungen zwischen den Elementen eines Systems. Die in Bild 2-3⁵ gezeigte Struktur eines mechatronischen Systems zeigt die Elemente Grundsystem, Aktuatorik, Sensorik und Informationsverarbeitung und die Beziehung unter ihnen. Es wird zwischen logischer und physikalischer Ebene unterschieden. Auf logischer Ebene werden Informationen ausgetauscht, auf physikalischer Ebene Energie und ggf. Stoffe. Über Kommunikationssysteme und Mensch-Maschine-Schnittstellen agiert das mechatronische System mit optionalen Elementen des Umfelds. Diese Darstellung ist nützlich, um die wesentlichen Bestandteile eines mechatronischen Systems zu erkennen.

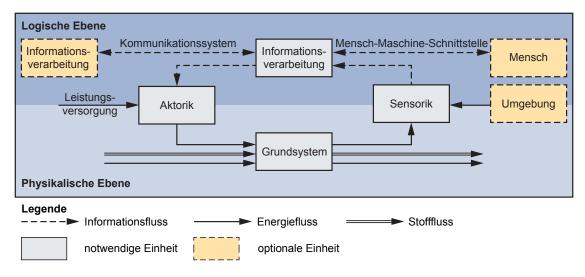


Bild 2-3: Struktur eines mechatronischen Systems [GTS14]

Dass beispielsweise der Mensch auch mit dem Grundsystem agieren kann, eine Leistungs-

⁵ursprünglich aus [VDI04a]

versorgung für die Informationsverarbeitung elementar wichtig ist oder die Aktorik⁶ Energie mit der Umgebung austauscht, sei an dieser Stelle hingenommen. Es kommt darauf an, dass ein Grundsystem mithilfe von Aktuatorik, Sensorik und einer Informationsverarbeitung so beeinflusst wird, dass es ein gewünschtes Verhalten zeigt. Weiterhin werden in dieser Darstellung die Digital/Analogumsetzung bei der Aktuatorik und die Analog/Digitalumsetzung bei der Sensorik vernachlässigt. Ebenfalls werden beispielsweise Vorverarbeitung und Verstärkung nicht dargestellt. Diese Bestandteile mechatronischer Systeme lassen sich natürlich noch weiter detaillieren. An dieser Stelle sei die Struktur auf die in Bild 2-3 dargestellte Art abstrahiert.

In der Literatur (z.B. [LKS00] [KFG07] [Hen13]) wird ausgehend von Systemen dieser Struktur, welche als *MFM (Mechatronisches Funktionsmodul)* bezeichnet werden, eine Vernetzung mehrerer dieser Systeme zu komplexeren Systemen beschrieben. Über *AMS (Autonome Mechatronische System)* bis zu *VMS (Vernetzten Mechatronischen Systemen)* steigert sich der Funktionsumfang. Eine Informationsverarbeitung übernimmt jeweils die regelnde Kontrolle. Diese Bezeichnungen haben sich im industriellen Kontext nicht vollständig durchgesetzt. Es wird eher in Kategorien wie Komponenten, Sub-Systemen, Systemen, und Systemen von Systemen gesprochen. Diese Nomenklatur scheint dahingehend besser geeignet zu sein, da eine breitere Anzahl unterschiedlicher Systeme darin enthalten sein kann. Gerade im Sinne intelligenter technischer Systeme und Industrie 4.0 Anwendungen, mit der Vernetzung unterschiedlichster Elemente auf verschiedenen Ebenen, scheint eine allgemeinere Bezeichnung passender. Das mechatronische System behält eine wesentliche Rolle, ist jedoch ggf. ein Teil eines übergeordneten Systems, welches nicht unbedingt einen Mechatronikfokus besitzen muss.

Es sei angemerkt, dass weitere Unterteilungen vorgeschlagen werden. So schlägt beispielsweise Rzevski eine Unterteilung in

- automatisierte mechatronische Systeme,
- intelligente mechatronische Systeme und
- intelligente mechatronische Netzwerke

vor [Rze03]. Dabei zeigt sich in jedem Fall, dass eine Entwicklung von mechanischen, über mechatronische, hin zu intelligenten technischen Systemen stattfindet, in welcher der Grad der Automatisierung und der Intelligenz der Systeme stetig zunimmt. Jedoch steigt auch die Komplexität solcher Systeme stark an, der mit geeigneten Methoden begegnet werden muss.

2.2.2 Klassifizierung mechatronischer Systeme

Während in der Mechatronik bisher im Wesentlichen zwei Unterscheidungen vorgenommen wurden [KFG07] – die räumliche Integration von Mechatronik und Elektronik und die Mehrkörpersysteme mit kontrolliertem Bewegungsverhalten– wird derzeit der Wan-

⁶Statt Aktorik sollten Stellglieder als Aktuatorik bezeichnet werden. Ein Zitat dazu aus dem Buch Systementwurf mechatronischer Systeme [Jan10]: mittelalterliches lateinisches Verb actuare = ausführen, davon Partizip Perfekt actuatus; mit actor (aus Wurzel ago) wird hingegen im klassischen Latein Verrichter, Schauspieler, Kläger gemeint.

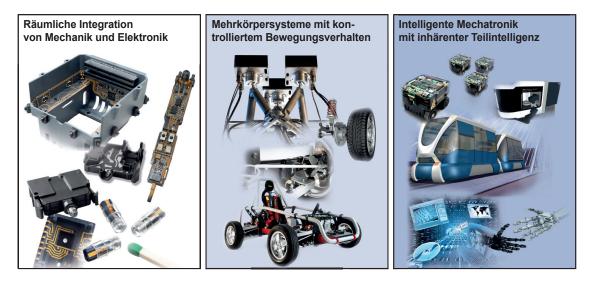


Bild 2-4: Klassen mechatronischer Systeme [GTS14]

del zu *intelligenten Systemen* vollzogen, die als dritte Klasse mechatronischer Systeme gesehen werden (vgl. Bild 2-4).

Eine weitere Klassifikation wird durch die Funktion beschrieben: Wird die Funktion durch Information bereitgestellt, handelt sich um eine andere Klasse an Systemen. Als Beispiel sei die GPS-Navigation genannt. Die Mechanik dient hier nur zur Ausführung von Befehlen und als Interface. Die intelligenten Systeme der Zukunft werden jedenfalls eine enge Interaktion der beteiligten Disziplinen benötigen. Diese Disziplinen werden jedoch nicht auf die Mechatronik bezogen sein, sondern auch weitere Wissenschaftsgebiete umfassen, wie z.B. adaptive Materialen. Die Interaktion mit den Benutzern wird sich vergrößern und daher müssen auch für z.B. psychologische Problemstellungen Lösungen gefunden werden, doch die domänenübergreifenden Herangehensweisen und Ansätze aus der Mechatronik eignen sich, den Herausforderungen der Entwicklung zukünftiger intelligenter Systeme entgegenzutreten.

2.2.3 Der Geldautomat als Beispiel eines mechatronischen Systems

Bild 2-5 zeigt ein komplexes mechatronisches System. Der Geldautomat integriert auf engstem Raum mechanische und elektronische Systeme und Komponenten. Die Interaktion von Feinwerkmechanik, Regelungstechnik und Elektronik macht diese Systeme zu einem geeigneten Anschauungsbeispiel im Kontext intelligenter Systeme. Es gibt verschiedene Elemente innerhalb eines Geldautomaten, wie verschiedene Geldwege, die u.a. den Tresor mit dem Bedienpanel verbinden und durch verschiedene Sensoren und Aktuatoren gesteuert werden. Die Funktionen eines Geldautomaten sind allgemein bekannt: Geldautomaten erlauben durch ihre ausgeklügelte Struktur die Aus- und Einzahlung von Bargeld, doch diese Sicht reicht nicht aus, um alle wichtigen Funktionen zu beschreiben. Schließlich zahlt ein Automat nicht nur Geld aus, sondern bucht den entsprechenden Betrag von einem bestimmten Konto ab, welches von Servern an anderer Stelle verwaltet wird. Erfahrungsgemäß kann Geld von einem einzigen Konto an völlig unterschiedlichen Geldautomaten in den meisten Ländern der Welt abgehoben werden. Eine isolierte



Bild 2-5: Ein handelsüblicher Geldautomat und die 955 mechanischen Einzelteile [Hei17]. Die beiden Darstellungen dienen der Veranschaulichung. Die Einzelteile gehören nicht zu dem links gezeigten Automaten.

Betrachtung eines einzigen lokal physisch aufgebauten Automaten ist somit nicht ausreichend.

Für eine vollständige Analyse, müssen die Systemgrenzen vergrößert werden. Bild 2-6 zeigt verschiedene Geldautomaten, die über unterschiedliche Server mit einem Datenserver verbunden sind. Ganzheitlich ist diese Betrachtung damit noch nicht, es fehlen u.a. noch die Nutzer. Die erweiterte Sicht auf das System zeigt exemplarisch, dass Systemgrenzen nicht starr sind und erheblichen Einfluss auf die Betrachtung und die Komplexität haben. Dabei ist die Komplexität nicht allein auf die mechanischen Einzelteile zurückzuführen, was dieses Beispiel zeigt.

2.3 Entwurfstechnik mechatronischer Systeme

Der Begriff der Entwurfstechnik beschreibt eine Vielzahl an Methoden und Techniken, um ein technisches System systematisch zu entwerfen. In diesem Abschnitt werden ausgewählte Aspekte näher erläutert, ein vollständiger Überblick ist für diese Arbeit nicht zielführend, sodass das Konzept des Systems Engineering allgemein und Begriffe eines Entwicklungsprozesses speziell beschrieben werden. Diese Gedanken werden für die Beschreibung eines verbreiteten Entwicklungsprozesses, dem V-Modell für die Entwicklung mechatronischer Systeme [VDI04a], genutzt.

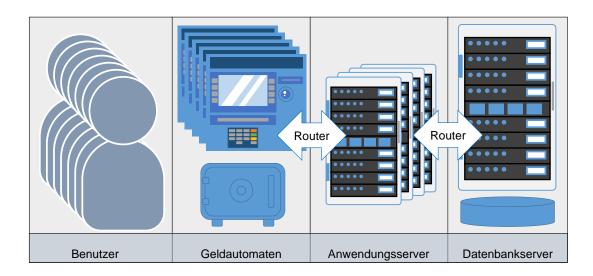


Bild 2-6: Geldautomaten als Teil eines Systems, abgewandelt nach [FT15]

2.3.1 Konzept des Systems Engineering

Viele der Prozesse, die dem Systems Engineering zugeordnet werden, sind auf eine neue Interpretation des Begriffs zurückzuführen. In den Ursprüngen war Systems Engineering auf Produkte ausgerichtet, die weitaus mehr als die ingenieurwissenschaftlichen Disziplinen beinhalteten. Es waren Produkte mit innovativer Technologie in einem hochkomplexen Umfeld, mit wenig Spielraum für Fehler. Digitale Vernetzung, hohe Komplexität und starke Interdisziplinarität stellen die Entwicklung technischer Produkte heutzutage vor neue Herausforderungen. Dies führte zu einer vermehrten Wahrnehmbarkeit von Systems Engineering in diesen Bereichen. Vor diesem Hintergrund gab es eine rasante Entwicklung der Methodiken und Prozesse, um komplexe Produkte systematisch entwickeln zu können. Dabei zeigen sich große Potentiale, aber auch Detailfragen, die noch geklärt werden müssen.

Es sollte demnach festgehalten werden, dass Systems Engineering bereits seit geraumer Zeit betrieben wird. Wird dies auf die Spitze getrieben, berichten sowohl die Bibel als auch der Koran und die Tora von der Entwicklung komplexer Systeme – z.B. der Arche. Und selbst, wenn es hier mit dem historischen Nachweis schwierig wird, wurden an anderer Stelle nachweislich sehr früh komplexe Gebilde entwickelt, beispielsweise die Pyramiden. Die explizite Anwendung der Disziplin *Systems Engineering* geht auf die 1940er, 1950er und 1960er Jahre zurück⁷. Der zweite Weltkrieg hatte eine technologische Weiterentwicklung zur Folge, durch welche die vermehrt aufkommenden interdisziplinären Systeme beherrschbar gemacht werden mussten. Der kalte Krieg, und das damit einhergehende Streben nach technologischer Dominanz verstärkte diese Entwicklung [KS11].

Zurzeit erlebt Systems Engineering eine Art Renaissance, für welche es im Wesentlichen zwei Gründe gibt. Mit der Zeit gibt es mehr technologische Möglichkeiten, die Konzepte

⁷Im weiteren Verlauf wird auch die Disziplin *Requirements Engineering* erwähnt, welche Ihre Ursprünge im Jahr 1977 hat [Par98].

des Systems Engineering adäquat und effizient einzusetzen. Der entscheidendere Grund für die neue und umfangreiche Forschungsaktivität ist sicherlich die steigende Komplexität durch die Vernetzung der Produkte, Produktionsanlagen und sonstiger Industrieerzeugnisse. Vor etwas mehr als einem Jahrzehnt waren auch angesehene Systems Engineering Experten teilweise der Meinung, es gäbe Produkte, für die dieses Konzept keine Vorteile bringe. Als Beispiel soll die Ansicht dienen, eine Waschmaschine sei kein Produkt, für welches Systems Engineering angewandt werden solle [KS11]. Wird die rasante Entwicklung in der Hausgerätetechnik der letzten Jahre betrachtet, sollte dies nun anders gesehen werden⁸. Zweifelsfrei gibt es weiterhin Produkte, für deren Entwicklung Systems Engineering keinen signifikanten Nutzen bringt, es bleibt jedoch zu erwarten, dass die Komplexität in den Produkten der Zukunft weiter steigen wird (vgl. Abschnitt 2.2). Hier sei besonders auf den Einzug von Industrie 4.0 verwiesen [ABD+15]. Die zunehmende Digitalisierung stellt mittlerweile eine erhebliche Anzahl technischer Anlagen an die Schwelle komplexer Systeme und macht sie damit zugänglich für einen effizienten Einsatz von Systems Engineering. Zwar plakativ und dennoch aussagekräftig titelten die VDI Nachrichten

"Mausklick statt Handschlag" [HB16].

Der Artikel der Ausgabe vom 8. September 2016 handelt von der Digitalisierung, welche Einzug in die Stahlindustrie erhält. Dabei werden Vertriebskanäle als Teil der Wertschöpfungskette verändert. Traditionelle Online-Shopping Ansätze entsprächen nicht den Anforderungen, sondern der Stahl müsse über Plattformen vertrieben werden, welche zusätzliche Tools und Services bereitstellen [HB16]. Vernetzung und Digitalisierung sorgen für eine Zunahme der Komplexität⁹. Für den Einsatz von Systems Engineering bedeutet das einen erwartbaren Zuwachs an Nutzen: Gleichzeitig muss die Disziplin sich an die neuen Herausforderungen anpassen. Es steht zwar außerhalb des Fokus dieser Arbeit, Systems Engineering wird mittlerweile jedoch auch in Bereichen wie Sozial- und Gesellschaftssystemen angewendet [MRGL13]. Das lässt erahnen, dass die Disziplin bereit ist, sich neuen Gegebenheiten anzupassen. Die Vielzahl an Open Source Anwendungen unterstreicht diese Annahme.

Ein Unterschied zwischen System Engineering und traditionellen Ingenieurwissenschaften ist durch die Art der getroffenen Entscheidungen gegeben. Traditionell verlassen sich Ingenieure auf quantitative Ergebnisse, um Entwicklungsentscheidungen zu treffen. Dies ist beim Systems Engineering in dieser Form nicht möglich, sodass auch Entscheidungen auf Basis qualitativer Einschätzungen getroffen werden [KS11]. Dies ist ein notwendiger Trade-off, da es sich hier um ein Spannungsfeld zwischen Pragmatismus und Sorgfalt handelt. Ein wesentlicher Bestandteil von Systems Engineering Aktivitäten ist daher das Risikomanagement [KS11]. Kritische Komponenten sollen möglichst frühzeitig erkannt werden, sodass die Sicherheit maximiert und das Risiko minimiert werden. Systems Engineering im Allgemeinen bedeutet nicht nur die Dekomposition von komplexen Aufgaben und der Sicherstellung der anschließenden Integration. Es umfasst alle Einflüsse, die auf

⁸Betrachtet werden sollte nicht ausschließlich Komplexität durch Funktionalität. Auch die Beteiligung eher unscheinbarer Disziplinen, wie z.B. die technische Dokumentation, haben Einfluss auf das System [Kot11].

⁹In der Ausgabe der VDI Nachrichten vom 15. September 2017 ist der Titel: *Tablet statt Tafel* gewählt. Hier geht es um den Einzug der Digitalisierung in die Bildung [Sch17].

und in einem System wirken können. Diese können auch erst allein durch das Zusammenspiel von unterschiedliche Systemelementen entstehen.

"Systems Engineering ist ein interdisziplinärer Ansatz und soll die Entwicklung von Systemen methodisch ermöglichen. SE fokussiert ein ganzheitliches und zusammenwirkendes Verständnis der Stakeholder Anforderungen, der Entdeckung von Lösungsmöglichkeiten und der Dokumentation von Anforderungen sowie das Synthetisieren, Verifizieren, Validieren und die Entwicklung von Lösungen. Das gesamte Problem wird währenddessen von der Konzeptentwicklung bis zur Systementwicklung betrachtet. Das Systems Engineering stellt hierfür geeignete Methoden, Prozesse und Best Practices bereit" [GCW+13, S. 20].

In der deutschstämmigen Industrie wird nach der ganzheitlichen Systembetrachtung am zweithäufigsten die Durchgängigkeit von den Anforderungen bis zur Systementwicklung mit dem Thema Systems Engineering verbunden (vgl. Bild 2-7).

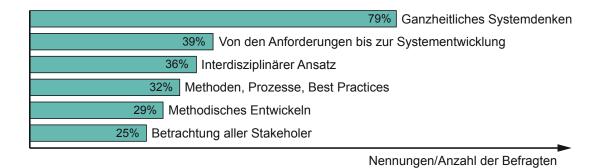


Bild 2-7: Interpretation der Bedeutung von Systems Engineering [GCW+13]

Einen wesentlichen Baustein für durchgängiges Systems Engineering stellt das Modelbased Systems Engineering (MBSE) dar. Es ist definiert als:

"die formalisierte Anwendung von Modellen zur Unterstützung von Aktivitäten zu Systemanforderungen, -design, -analyse, -verifikation und -validierung, beginnend in der konzeptionellen Designsphase und weiterführend in der Entwicklung und späteren Lebenszyklusphasen"¹⁰ [Fos13].

Systems Engineering benötigt diese Modelle: Vom klassischen Systemmodell bis hin zum Simulationsmodell. Die Entwicklung im MBSE geht dabei hin zu einem Systemmodell mit verbleibenden domänenspezifischen Modellen mit ihren charakteristischen Informationen [Fos13].

2.3.2 Entwurfstechnik: Kontext und Begriffe

Entwurfstechnik bietet eine methodische Unterstützung bei der Entwicklung technischer Systeme. Wertvolle Ergebnisse gehen aus dem Projekt ENTIME – Entwurfstechnik In-

¹⁰aus dem Englischen übersetzt

telligente Mechatronik hervor, so kann auf eine fachdiziplinübergreifende Entwicklungsmethodik zurückgegriffen werden, die einige Grundsätze zum methodischen Entwickeln festlegt und eine gewisse Nomenklatur bietet. Viele Begriffe, die auch in dieser Arbeit Verwendung finden, entstammen der Spezifikationstechnik *CONSENS*, welche in den frühen Phasen des Entwurfs angewendet wird. (vgl. Bild 2-8) [GTS14]. Multidiszipli-

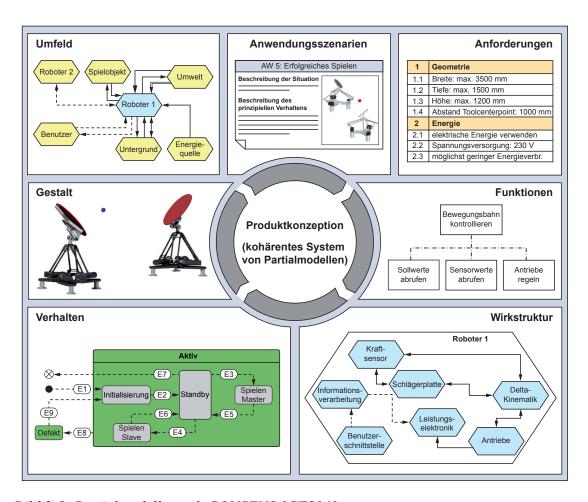


Bild 2-8: Partialmodelle nach CONSENS [GTS14]

näre Teams arbeiten in domänenübergreifenden konzeptionellen Phasen zusammen, um Zielvorstellungen und prinzipielle Lösungen zu entwerfen. Diese Erkenntnisse werden anschließend in die domänenspezifische Entwicklung übertragen, in welcher Spezialisten jeder involvierten Disziplin die Prinziplösung konkretisieren. In der domänenübergreifenden Phase kann systematisch auf Partialmodelle zurückgegriffen werden, wie sie von *CONSENS* vorgeschlagen werden. Diese beinhalten jeweils eine Sicht auf ein System, beispielsweise die *Funktionshierarchie* oder die *Wirkstruktur* und beschreiben somit die Funktionen des Systems oder die Wirkzusammenhänge der verschiedenen Systemelemente. Darüber hinaus können weitere Partialmodelle, wie z.B. die Anwendungsszenarien beschrieben werden. Gemeinsam bilden sie ein zusammenhängendes System, welches genutzt werden kann, um ein besseres Verständnis über das System in frühen Phasen zu erlangen. Es wird zur Konkretisierung des zu entwickelnden Systems in verschiedene Phasen unterteilt. Die frühe Phase des Entwurfs sucht eine Prinziplösung, in welcher Systemelemente noch als abstrakte Lösungsmuster angenommen werden. Eine solche Ab-

straktion könnte die Sicht eines Antriebs als Elektromotor sein. Dieser muss noch nicht zu einem konkreten Lösungselement ausgewählt werden, da dieser Schritt in der späteren disziplinspezifischen Entwicklungsphase durchgeführt wird. Es wird in diesem Beispiel jedoch bereits festgehalten, dass das Lösungsmuster ein Elektromotor ist. Eine detaillierte Beschreibung findet sich in der Literatur [GTS14].

2.3.3 Modellbasierte Entwicklung: V-Modell

In dieser Arbeit wird das bekannte V-Modell für den Entwurf mechatronischer Systeme (Bild 2-9) genutzt. Eine detaillierte Sicht auf dieses findet sich in Bild 2-10. Als eine kurze Beschreibung, wie eine generische Entwurfsmethodik durch einen Entwicklungsprozess führt, wird dieses im Folgenden in aller Kürze erläutert. Zu Beginn gehen Anforderungen in den Entwicklungsprozess ein. In der Systementwurfsphase, dem linken Ast, wird systematisch eine domänenübergreifende Lösung für das zu entwickelnde System bereitgestellt. Im fachdisziplinspezifischen Entwurf werden die benötigten Entwicklungsergebnisse zu der Lösung hinzugefügt, sodass das System mit den Resultaten aus dem Maschinenbau, der Elektrotechnik und der Informationstechnik angereichert und konkretisiert wird. In der Systemintegrationsphase werden die Teilergebnisse zu einem gemeinsamen System zusammengefügt. Dieser Prozess wird über weite Strecken durch Modellbildung und Modellanalyse unterstützt und die Eigenschaftsabsicherung ist elementarer Bestandteil des gesamten Vorgehens.

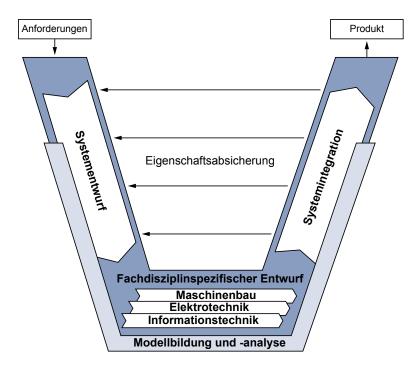


Bild 2-9: V-Modell entsprechend der VDI Richtlinie 2206 [VDI04a]

Im Folgenden werden die Konzepte der Modellbildung beschrieben, da sie im Sinne einer modellbasierten Entwicklung einen großen Anteil tragen.

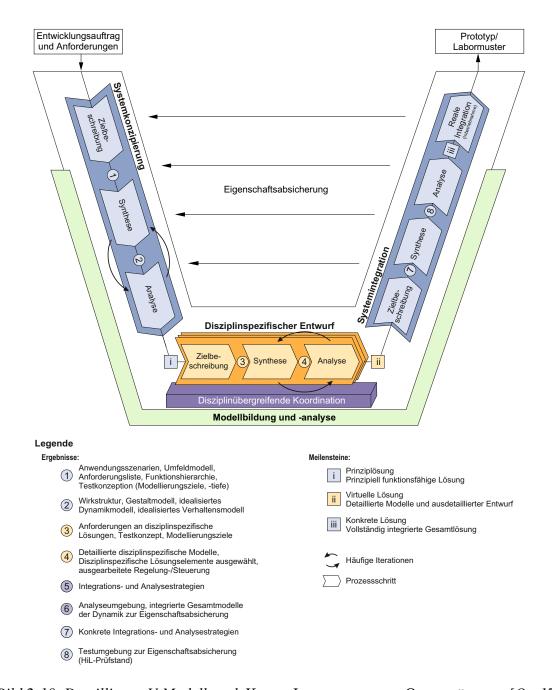


Bild 2-10: Detailliertes V-Modell nach Kruse, Lochbichler und Oestersötebier [Oes17]

2.4 Modellbildung und mechatronische Analyse

Modellbildung ist ein Teil der modellbasierten Entwicklung. Der Relevanz des Themas für diese Arbeit ist es geschuldet, dass dem Thema ein separater Abschnitt zugewiesen ist. Daher wird zu Beginn die grundlegende Frage gestellt: Besteht ein einheitliches Verständnis über den Begriff *Modell*? Sicher besteht es im Abstrakten. Schwieriger wird es über Domänengrenzen hinweg. In der interdisziplinären Kommunikation zwischen IT und Maschinenbau bestehen womöglich offensichtlich unterschiedliche Interpretationen, jedoch kann nicht davon ausgegangen werden, dass selbst in gleichen Fachgebieten ein einheitliches Verständnis herrscht. Da Modellbildung ein wirkungsvolles Werkzeug dieser Arbeit darstellt, wird im Folgenden eine Einordnung gegeben. Diese umfasst zwar nicht alle möglichen Ausprägungen, die relevanten Zusammenhänge werden jedoch dargestellt. Eine grundsätzlichere philosophische Auseinandersetzung findet sich in der Literatur [Sta83]. Eine mögliche Definition des Begriffs *Modell* lautet: Ein Modell ist eine

"mathematische oder physikalische Darstellung eines Systems oder Prozesses, die das System oder den Prozess aufgrund bekannter Gesetzmäßigkeiten, einer Identifikation oder getroffener Annahmen genügend genau abbildet" [DIN14, S. 29].

Ebenfalls könnte es sich auch um ein Abbild der Software oder um ein abstraktes Systemmodell handeln. Diese stellen ebenfalls Abbilder eines Systems dar; der Unterschied liegt allein in der mathematischen oder physikalischen Darstellung. Nun ist es in der anwendungsorientierten Industrie oftmals so, dass Modelle skeptisch betrachtet werden.

"Einem Modell traut niemand, außer dem, der es erstellt hat – einer Messung traut jeder, außer dem, der sie gemessen hat" [Jus14, S. 17].

Doch in jüngerer Vergangenheit wandelt sich diese Sicht, da die Bedarfe steigen, jedoch eine genaue Vorstellung von Leistungsfähigkeit und Inhalt fehlt. Die Frage nach der Bedeutung des Begriffes *Modell* stellt sich auch in anderen Bereichen. In der Medizin beispielsweise wird das Wachstum von Zellen ebenfalls mit mathematischen Zusammenhängen beschrieben. Die Bedeutung der Art der Gleichungen, der inneren Strukturen und des Ein-/Ausgangsverhaltens wird in diesem Kontext ebenfalls diskutiert [Eis79]. In der Medizin wird unterschieden nach *diskreten* und *kontinuierlichen* Modellen. Innerhalb der Klassifikationen wird wiederum unterschieden nach deterministischen und stochastischen Modellen [Eis79]. Es kann abgeleitet werden, dass der Modellierungszweck ebenfalls in der Medizin einen wesentlichen Einfluss auf das Modell hat. Ähnliche Gedanken sind für dynamische Systeme notwendig, um ein einheitliches Verständnis zu erlangen¹¹.

2.4.1 Grundgedanken und allgemeine Einordnung

Das Thema Modellbildung und Modelle ist in seiner Bedeutung entsprechend vielfältig. Verschiedene Disziplinen werden unterschiedliche Meinungen darüber haben, was mit

¹¹Eine interessante Eigenschaft der angegebenen Literaturquelle [Eis79] ist ein umfangreiches Kapitel über Regelungstechnik. Auf 56 Seiten werden Grundlagen zu relevanten regelungstechnischen Themen erläutert, mit denen das Verständnis über Wachstumsprozesse und Wechselwirkungen im Kontext maligner Neoplasien und entsprechender Therapie erhöht wird.

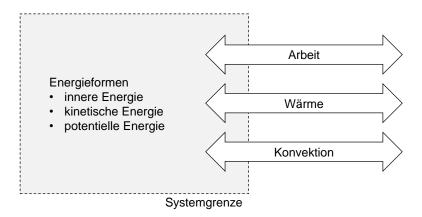


Bild 2-11: Die Formen des Energietransportes über die Systemgrenzen und die Energieformen, abgewandelt nach [HW12]

einem Modell oder mit einem Modellbildungsprozess gemeint ist. Informatiker denken bei Modellen an einen Code, Systemdesigner vermuten SysML hinter dem Begriff, und traditionelle Ingenieure stellen sich mathematische Gleichungen vor. Letztlich herrscht eine Einigkeit darüber, dass ein solches Modell eine Repräsentation eines Originals darstellt. Diese Definition ist jedoch sehr abstrakt und generisch. Vielmehr soll mit einem Modell eine spezielle Eigenschaft eines Originals untersucht werden, wenn es aus Sicht der Ingenieurswissenschaften betrachtet wird. So macht es nicht unbedingt Sinn, ein System immer in seiner vollen Komplexität zu modellieren. Es sollte sich auf die wichtigen Dinge konzentriert werden¹². Nun ist nicht nur das Problem an dieser Definition, dass es unterschiedliche Vorstellungen darüber geben kann, wie viel von diesem Original modelliert werden soll, sondern auch inwiefern diese Modellierung auf physikalischem Wissen beruht. Es kann sinnvoll sein, ein Modell so einfach zu belassen, dass es letztlich keine physikalisch motivierten Gleichungen beinhaltet, sondern zum Beispiel aus einfachen Kennlinien besteht. Auch sogenannte Blackbox-Modelle können eine Möglichkeit sein, relativ einfach ein Ein- und Ausgangsverhalten zu reproduzieren, ohne die genauen Abläufe zu kennen. Dies ist in vielen verschiedenen Anwendungsszenarien von Bedeutung. Oftmals ist das Ein- und Ausgangsverhalten das Entscheidende am Modell. Kommt es auf eine möglichst exakte Reproduktion eines Wirkzusammenhangs von Ein- zu Ausgang an, sind solche Modelle in vielen Fällen vorzuziehen. Kommt es jedoch auf die Analyse des Systems an, so sollten vielmehr konsequent physikalisch motivierte Modelle verwendet werden.

Bei der physikalischen Modellbildung sollte beachtet werden, dass typischerweise nicht alle Effekte modelliert werden. Ein häufig vernachlässigtes Phänomen ist *Reibung*. Diese Form des Energietransports ist in manchen Anwendungen jedoch von großer Bedeutung. Wenn eine große Menge Reibungsenergie über die Systemgrenze hinweg in Form von Wärme verloren geht, kann eine Nichtberücksichtigung im Modell zu Schwierigkeiten führen. Die möglichen Formen des Energietransports über Systemgrenzen hinweg sind in Bild 2-11 dargestellt. Auch innerhalb der physikalisch motivierten Modellbildung gibt es unterschiedliche Perspektiven. Mathematische Gleichungen welche auf physika-

¹²Das erste Prinzip des Artikels Five simple principles of modelling [Pid96] lautet model simple, think complicated. Der Artikel stammt zwar aus den Wirtschaftswissenschaften, bietet jedoch einige interessante Aspekte zur Modellierung.

lischen Prinzipien beruhen, können mit verschiedenen Formalismen wie Newton Euler oder Lagrange aufgestellt werden. Darüber hinaus kann auch eine andere Herangehensweise gewählt und die Methoden der Netzwerkmodellierung genutzt werden. Hier sind u.a. die Bondgraphen zu nennen. Zusätzlich zu den suggerierten virtuellen Modellen können auch reale Systeme die Eigenschaften eines Modells erfüllen. Prototypen, bzw. das Produkt dürfen daher im Kontext der Modelle nicht vernachlässigt werden. Aus dem gesamten Portfolio sollte ein geeignetes Modell ausgewählt werden, um einen bestimmten Zweck bestmöglich zu erfüllen. Das Modell dient als Werkzeug und daher sollte das richtige Werkzeug genutzt werden.

2.4.2 Modellierungsziel

Ein Modell kann beliebig komplex werden. Dies ist jedoch nicht immer sondern eher sehr selten sinnvoll. Erstens wird der Aufwand mit steigender Detaillierung größer (vgl. Bild 2-12) und zweitens wird die Analyse des Modells erschwert. Daher sollte sich der Modellierer mit dem jeweiligen Ziel der Modellbildung auseinandersetzen.

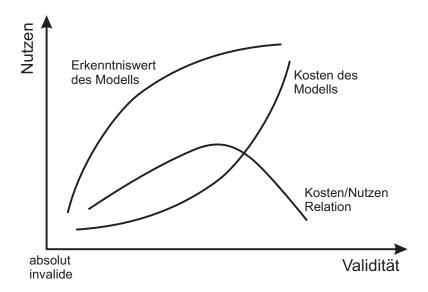


Bild 2-12: Kosten der Modellbildung [Jus14], nach [Sha75]

Das Modellierungsziel sollte immer im Kontext des zu modellierenden Systems stehen. Daher sollten Systemeigenschaften und Modelleigenschaften unterschieden werden können. In Bild 2-13 wird ein Ansatz einer solchen Unterscheidung verschiedener Eigenschaften dargestellt, welcher jedoch keinen Anspruch auf Vollständigkeit erhebt. Damit lassen sich vielfältige Modellziele auflisten, wie zum Beispiel [NC97]:

- Systemverständnis erhöhen,
- Reglerauslegung,
- Systemstrukturanalyse,
- Fehlersuche und Fehlernachbildung,
- Funktionsnachweis,
- Elementauslegung,
- Optimierung und
- modellbasierte Diagnose.

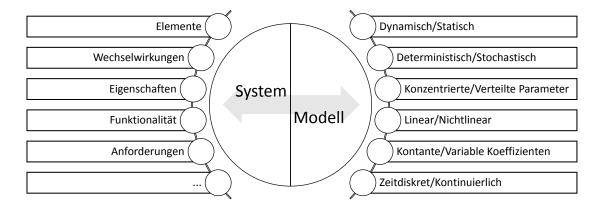


Bild 2-13: Systemeigenschaften und Eigenschaften mathematischer Modelle

In dieser Arbeit sei auch anhand der folgenden Kategorien unterschieden: Modelle zur *Generierung* von Wissen und Modelle zur *Repräsentation* von Wissen. Generierende Modelle werden zur Analyse bestehender Systeme, zur Synthese neuer Systeme oder zur Validierung bzw. zum Test genutzt. Repräsentative Modelle stellen eine Art der Dokumentation oder Spezifikation dar. Solche können auch als Anforderungen dienen, wenn sie gewisse Eigenschaften spezifizieren, die bei der Entwicklung umgesetzt werden. Eine Möglichkeit dafür wird in Abschnitt 5.2.1 vorgestellt. Je nach Modellierungsziel kann an unterschiedlichen Stellen des Modellbildungsprozesses gearbeitet werden (Bild 2-14). So kann der Fokus im Zusammenhang mit Anforderungen eher auf die im Bild blau markierten Prozessschritte *Abgrenzung zur Umwelt* und *Apriori Kenntnisse auswerten* gelegt werden.

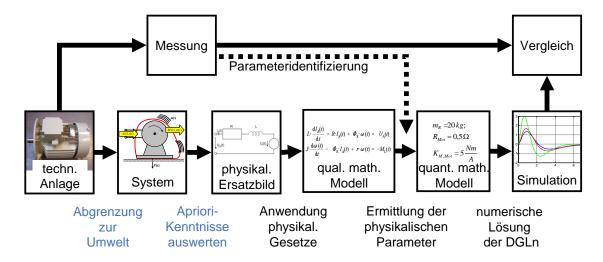


Bild 2-14: Grundsätzliches Vorgehen bei der Modellbildung dynamischer Systeme [GTS14].

2.4.3 Modellierungsarten dynamischer Systeme

Für die Repräsentation von Wissen gibt es einerseits die Modelle der SysML, aber auch Modelle diskreter Systeme, wie Zustandsautomaten, oder Modelle dynamischer Systeme. Im weiteren Verlauf wird näher auf die Modelle zur Generierung von Wissen eingegangen. Die Bilder 2-15 und 2-16 zeigen eine Übersicht einiger Modellierungen für unterschiedliche Arten von Systemen. Dabei wird zwischen kontinuierlichen und diskreten Systemen unterschieden. Innerhalb dieser Kategorien sind unterschiedliche Anwendungsbereiche aufgezeigt. Als zusätzliche Kategorisierung kann die Modellierungstiefe gesehen werden. So kann z.B. ein Mehrkörpersystem (MKS) Modell sehr detailliert oder eher abstrakt sein. Eine Quantifizierung der Modellierungstiefe wird in der Literatur vorgeschlagen [LOT14]. Modelle können also viele Gestalten annehmen und anhand mehrerer Kriterien unterschieden werden. Im weiteren Verlauf werden konkreter verschiedene Arten von Modellen erläutert.

Domäne	Strukturelle Beschreibung	Modellierungs- prinzip	Mathematische Modellform
Mechanik	MKS Modell	Mechanische Prinzipien: • d'Alembert • Euler-Lagrange	Deskriptorform: Differential- Algebraisches Gleichungssystem
Elektronik (analog)	Netzwerk	Kirchhoff'sche Sätze: • Modifizierte Knotenpkt Analyse • Sparse-Tableau	Deskriptorform: Differential- Algebraisches Gleichungssystem
Regelungstechnik	Blockschaltbild	Direkte Umsetzung	Erweit. Zustandsform: DGL + algebraische Hilfsgleichungen
Weitere Domänen			

Bild 2-15: Modellierung kontinuierlicher Systeme, abgewandelt nach [Rod13]

Domäne	Strukturelle	Beschreibung	Mathematische Modellform		
Elektronik (digitale Schaltkreise)	Logikmodelle	VHDL-Netzliste	Boole'sche Funktionen und Algorithmen		
Regelungstechnik (Abtastsyst.)	Blockschaltbild	-	Differenzen- gleichungen		
Binäre Steuerungstechnik	Zustandsgraphen	Petri-Netze	Zustands- automaten		
Softwaretechnik	Nassi-Shneiderman	Programm- ablaufpläne	Höhere Programmier- sprachen (Fortran, C++ etc.)		

Bild 2-16: Modellierung diskreter Systeme, abgewandelt nach [Rod13]

Gleichungsbasierte Modellierung

Letztlich basieren alle simulationsfähigen Modelle auf Gleichungen. Der Unterschied liegt darin, dass die Modellerstellung durch die explizite Formulierung von Gleichungen geschieht. Um den Kontext etwas einzugrenzen, wird die Beschreibung auf dynamische Modelle kontinuierlicher Systeme beschränkt. Gleichungsbasierte Modellierung findet hier einerseits zur Beschreibung der Systemdynamik Anwendung, andererseits können Zusammenhänge verdeutlicht werden. Diese Gleichungen können als Systeme sehr umfangreich werden, bieten aber stets exakte Analysemöglichkeiten. Gleichungen zeigen teils komplexe Zusammenhänge in einfachen mathematischen Termen. Dies ist ein besonderer Vorteil zur Anforderungsermittlung, was sich im weiteren Verlauf zeigt. Die Gleichung erlaubt dabei nicht nur durch ihre Zusammenhänge Einblicke, sondern ebenfalls durch ihre Struktur. Die Eigenschaften von Differentialgleichungen und Differentialalgebraischen Systemen sowie partiellen Differentialgleichungen und Integralen zeigen ohne detailliertere Berechnungen bereits Eigenschaften der Systeme auf, die sie beschreiben. Im ingenieurwissenschaftlichen Bereich ist die Modellierung mittels Blockschaltbildern am weitesten verbreitet. Die Systemgleichungen werden dabei als Blockschaltbilder mit gerichteten Verbindungen, beziehungsweise Größen, dargestellt. Die Modellierung basiert auf dem Ursache-Wirkungs-Prinzip und durch die rein mathematische Modellierung ergibt sich ein fachdisziplinunabhängiges Modell, wobei jedoch eine vorherige Generierung der physikalischen Gleichungen vorausgesetzt wird. Anwendung finden die signalflussorientierten Modellierungsarten vor allem im Zusammenhang mit der Informationsverarbeitung. Grundsätzlich hat ein mathematisches Modell vier Bestandteile [PW88]:

Systemvariablen: Dies sind Größen, die verschiedene Zustände des Systems spezifizieren. Sie können in einem geeigneten Bereich variieren.

Systemparameter: Dies sind Größen, die einen festen Wert erhalten. Sie sind abhängig von der Anwendung des Modells und weniger von den unterlagerten Phänomenen.

Systemkonstanten: Dies sind Größen, die einen festen Wert beibehalten, der von unterlagerten Phänomenen abhängt. Der Modellierer kann diese nicht verändern.

Mathematische Beziehungen: Dies sind Gleichungen und Ungleichungen, welche die Systemvariablen, -parameter und -konstanten in Verbindung zueinander setzen.

Im Zuge des Modellbildungsprozesses sind jedoch auch alle Schritte durchzuführen (vgl. Bild 2-14) und somit ist diese Modellierung gewiss auch umfangreich, da jeder Schritt gewisses Expertenwissen erfordert.

Netzwerkmodelle

Im Fall der Topologieorientierung erfolgt die Modellierung über Port-Schnittstellen (Mehrpole), die jeweils eine Fluss- und eine Potentialgröße aufweisen, wobei sowohl gerichtete (Signale), als auch ungerichtete Größen (z.B. Momente) verwendet werden. Dies ermöglicht eine Modellierung auf physikalischer Ebene und zusätzlich eine einfache Verkopplung von Teilsystemen. Eine Objektorientierung erleichtert die Wiederverwendung der

modellierten Komponenten und den Aufbau von umfangreichen, hierarchischen Bibliotheken. Durch disziplinübergreifende Analogien ist es weiterhin möglich, Modelle multidisziplinär zu koppeln. Es scheint, als sei es mithilfe von Netzwerkmodellen einfacher zu modellieren, jedoch sollte deshalb besonders auf die richtigen Annahmen geachtet werden. Der erfahrene Modellierer kann auf die Vorteile zurückgreifen, z.B. die Gleichungen nicht selbst zu ermitteln, sondern das System grafisch aufbauen zu können. Werden diese Schritte ohne Hintergrundwissen erledigt, können schwerwiegende Fehler eintreten.

Gestaltorientierte Modellbildung

Modelle im Maschinenbau lassen sich auch aus dreidimensionalen Gebilden zusammenstellen, was eine große Flexibilität ermöglicht. Diese Flexibilität kann am besten eingesetzt werden, wenn ein zu modellierendes System bereits real oder zumindest eine konkrete Vorstellung vorliegt. Dann können die Analysewerkzeuge sehr gut eingesetzt werden. Spezialisierte Softwarewerkzeuge stellen umfangreiche Bibliotheken zur Verfügung und sind intuitiv zu bedienen. Die Generierung der Dynamikgleichungen erfolgt automatisch und ist im Bereich der Mehrkörpersimulationen auf die mechanischen Eigenschaften fokussiert. Es existieren zwar Schnittstellen, beispielsweise zu Aktuatorik und Sensorik, der Fokus liegt jedoch auf der Mechanik. Durch diese disziplinspezifische Ausprägung ist eine Kopplung mit anderen Disziplinen in vielen Fällen nur eingeschränkt möglich. Eine detaillierte Beschreibung der Grundlagen der Mehrkörpersimulationstechnik findet sich in der angegebenen Literatur [RS10].

Mechatronische Analyse realer Systeme

Mechatronische Analysen können auf verschiedenste Arten erfolgen. Hier werden Untersuchungen an einem realen System im Kontext der Modellbildung beschrieben. Es bietet sich an, Simulationen durch Messungen zu ergänzen, gerade wenn detailliertere Informationen benötigt werden. Für diese mechatronischen Analysen realer Systeme werden im Allgemeinen geeignete Werkzeuge benötigt [Lan01]. Die Gründe, eine Simulation durch eine Messung zu ergänzen sind vielfältig. Ist der Modellierungsaufwand zu hoch, sollten reale Messungen erwogen werden. Hier kann es sich um schwer zu modellierende Effekte, wie z.B. Reibung, handeln oder um Wechselwirkungen, welche real untersucht werden sollen. Reale Untersuchung und Simulation gehen dabei symbiotisch vor und bereichern einander. Die Ergebnisse können z.B. durch Kennlinien ins Modell zurückgeführt werden. Weitere sehr bedeutende Anwendungen mechatronischer Analysen am realen System sind die *Validierung* von Modellen und die *Parameteridentifikation* [Ise92]. In diesem Kontext sind vielfältigen Zwecke denkbar, wie beispielsweise in der Literatur umgesetzt [Sch16]. Das genutzte reale System kann ein in der Entwicklung befindlicher Prototyp sein oder auch eine reale Komponente, die z.B. nicht modellbasiert analysiert werden kann.

Sind reale Komponenten in eine Simulationsumgebung eingebunden, handelt es sich um sogenannte *Hardware-in-the-Loop (HiL) Technologien*. Dies können traditionell Steuergeräte sein, welche gegen ein Modell getestet werden, aber auch reale Komponenten, die geeignete Schnittstellen zur Simulation benötigen. Die reale Komponente soll so beeinflusst werden, wie es im realen Umfeld wäre. Wertvolle Arbeit auf diesem Gebiet wird von Olma geleistet [OKT+16]. Verallgemeinert wird im Zusammenspiel von realen und

virtuellen Elementen von *X-in-the-Loop* Technologien gesprochen. Durch das *X* wird der vollständige und umfassende Entwicklungsprozess adressiert, also z.B. Software-in-the-Loop (SiL) und Model-in-the-Loop (MiL).

Modelle in Kombination mit Messungen können auch genutzt werden, um eine Diagnose von Systemen durchzuführen. Hier ist besonders das *Condition Monitoring* zu erwähnen. Das Ziel ist, die momentanen und zukünftigen Systemzustände zu deuten. Dies erfolgt z.B. in Hinblick auf einen Ausfall durch bekannte Schwachstellen, wie Verschleißteile. Eine Erweiterung des Condition Monitoring zu einer konkreten Empfehlung, wird *Predictive Maintenance* genannt. Diese vorausschauende Wartungsempfehlung und ggf. Reparatur verkürzt standardisierte Wartungsintervalle und ermöglicht eine effizientere Nutzung von Systemen. Detaillierte Informationen können der Literatur entnommen werden [NC97].

Einsatz von Modellen

An welchen Stellen können Modelle eingesetzt werden? Etabliert ist der Einsatz in der fachdisziplinspezifischen Entwicklungsphase¹³. Hier gilt es weiter zu differenzieren, da der Einsatz in weiteren Phasen des Produktlebenszyklus immer mehr in den Fokus gelangt. In der Produktion hat sich der Begriff *Digital Twin* durchgesetzt. Laut der Beschreibung der *Hannover Messe*, ist dieser ein virtuelles Abbild einer realen Maschine [Han17]. Die beiden Zwillinge tauschen kontinuierlich Daten aus. Das bedeutet, in der Nutzungsphase der Maschine, erlaubt der virtuelle Zwilling eine parallele Auswertung. Auch in der Entwicklung mache der digitale Zwilling den realen Prototypen überflüssig [Han17]. Es wird zwar angemerkt, dass im virtuellen Abbild sämtliche Eigenschaften des Originals hinterlegt sind, diese Aussage darf jedoch in dieser Form (noch) angezweifelt werden, da die heutigen Technologien noch nicht weit genug entwickelt sind. Die Einflüsse des Digital Twins reichen soweit, dass dieser auch Einfluss auf die Anforderungen und deren Nutzung haben soll:

"Ein digitales Pflichtenheft in Form eines Requirement Management Systems sammelt die Informationen und vergleicht sie mit den Anforderungen, die an das Produkt gestellt werden" [Han17].

So kann genau beurteilt werden, ob ein System den Anforderungen genügt. Wird es möglich, sämtliche relevante Daten zu extrahieren, so eröffnet dies sicher weitreichende Möglichkeiten für den expandierenden Einsatz von Modellen.

2.5 Anforderungen in konventionellen Entwurfsprozessen

Nahezu ausschließlich starten konventionelle Entwurfsprozesse mit Anforderungen. Dennoch ist die Definition dieses Begriffs im Detail nicht einheitlich. In diesem Abschnitt wird die unterschiedliche Bedeutung für die beiden diese Arbeit prägenden Disziplinen – das klassische Ingenieurwesen und das Software Engineering – erläutert. Im Allgemeinen lässt sich festhalten, dass den funktionalen Anforderungen viel Beachtung im Software

¹³Lochвichler zeigt, welche Modellierungstiefen für welche Aufgaben geeignet ist [Loc20].

Engineering geschenkt wird, während im Ingenieurwesen die technischen Anforderungen im Vordergrund der Betrachtung stehen. Beide Disziplinen vernachlässigen die anderen Typen von Anforderungen nicht, sodass an dieser Stelle eine Beschreibung der jeweiligen Unterschiede zielführend ist. Das Ziel dieses Abschnittes besteht darin, die grundlegenden Zusammenhänge zu verdeutlichen. In praktisch allen intelligenten technischen Systemen ist ein signifikanter Softwareanteil zu verzeichnen. Dies macht eine Betrachtung sowohl aus Ingenieursicht als auch aus Softwaresicht erforderlich. Beide Sichten haben gemeinsam, dass sie Gebrauch vom Systems Engineering machen.

2.5.1 Anforderungen aus der Sicht klassischer Ingenieure

In diesem Abschnitt werden zum einen Anforderungen in klassischen Entwicklungsmethodiken aufgegriffen. Hier ist vor allem die Anforderungsliste zu nennen. Zum anderen wird die bekannte Unterteilung in Lastenheft und Pflichtenheft vorgestellt. Gerade zum Thema Lastenheft wird ein pragmatischer Zugang beschrieben [Pfi17]. Anfangs wird die Anforderungsliste beschrieben, wie sie im Jahre 1977 vorgestellt wurde [PB77]. Hiermit beginnend, lässt sich ein stetiger Wandel der Sicht auf Anforderungen beobachten.

Anforderungen 1977 Unter dem Credo des Klärens der Aufgabenstellung werden Anforderungen charakterisiert als eine Konfrontation mit dem gestellten Problem. Im Allgemeinen wird die Aufgabenstellung in drei Formen an die Entwicklung getragen

- interner oder externer Entwicklungsauftrag,
- konkrete Bestellung,
- Anregungen.

Es wird bereits auf die Wichtigkeit der engen *Fühlungnahme* zwischen Initiator und Entwicklung hingewiesen. So hilft diese zur Konkretisierung

- des Problems,
- der nicht ausgesprochenen Erwartungen,
- der Bedingungen der Aufgabe,
- der Freiheitsgrade der Entwicklung.

Es wird darauf hingewiesen, dass eine möglichst lösungsneutrale Formulierung anzustreben sei. Diese beruhe auf Funktionen mit Ein- und Ausgangsgrößen und sei fokussiert auf den Zweck der Lösung, die geforderten Eigenschaften der Lösung und diejenigen Eigenschaften, die das Produkt nicht haben soll. Nach abgeschlossener Ermittlung dieser Informationen kann mit der Erstellung der Anforderungsliste begonnen werden, welche in dieser Hinsicht über das Lasten- und das Pflichtenheft hinausgehe. Die Anforderungsliste beinhaltet Forderungen und Wünsche. Bemerkenswert ist, dass bei der Beurteilung möglicher Lösungsvarianten darauf verwiesen ist, dass dafür lediglich die Wünsche relevant seien, da die Forderungen in jedem Fall erfüllt seien müssen. Es wird empfohlen, soweit möglich, Qualitäts- und Quantitätsangaben vorzunehmen und diese mit Zahlenangaben zu hinterlegen. Damit wird die Anforderungsliste definiert als:

				Ausg	abe:
Ber	nutzer	Anforderungsliste für Projekt, Produkt	Identifikation: Klassifizierung Blatt:	g: Seite	e:
ID	Prio.	Anforderungen	Verantw.	Änderung	Begründung

Bild 2-17: Aufbau einer Anforderungsliste, abgewandelt nach [PBFG13]

"Die Anforderungsliste ist somit ein internes Verzeichnis aller Forderungen und Wünsche in der Sprache der Abteilungen, die die Konstruktion durchzuführen haben. […] sie zwingt den auftraggebenden Partner zu einer klaren Stellungnahme" [PB77].

Der Aufbau einer klassischen Anforderungsliste unterliegt keiner zwingenden Struktur: Etabliert hat sich jedoch ein Aufbau gemäß Bild 2-17 [PBFG13]. Es wird als zweckmäßig erachtet, die entsprechenden Quellen, also die Ursprünge von wichtigen Anforderung anzugeben. Dies erleichtere es, ggf. anfallende Änderungen und Ergänzungen der Anforderungsliste möglichst schnell durchführen zu können. Die Anforderungsliste sollte stets aktuell gehalten werden, um ständig die aktuelle, möglicherweise veränderte Aufgabenstellung abzubilden. Zur Ermittlung der Anforderungen werden u.a. Checklisten genannt. Diese seien jedoch nur bedingt zweckmäßig. Indikationen zur Nutzung solcher Checklisten sind:

- Ein begrenzter Bereich wird erfasst,
- die Checkliste ist für längere Zeit aktuell,
- der Umfang der Checkliste ist begrenzt,
- der Aufwand zur Nutzung ist begrenzt.

Es werden des Weiteren vier methodische Schritte zur Erstellung einer Anforderungsliste vorgeschlagen:

- 1. Sammlung der Anforderungen,
- 2. Ordnung der Anforderungen,
- 3. Erstellung der Anforderungsliste,
- 4. Prüfung der Anforderungsliste.

Zusätzlich wird eine Leitlinie mit Hauptmerkmalen zum Aufstellen von Anforderungslisten bereitgestellt [PB77]. Durch die Anforderungsliste entsteht weiterer Nutzen bei der Verwendung für Änderungskonstruktionen, indem sie als Informationsspeicher für die geforderten und gewünschten Eigenschaften eines Produktes dient [PB77].

Anforderungen heute Anforderungen werden weiterhin als die Klärung und Präzisierung der Aufgabenstellung aufgefasst. Es werden nunmehr methodischere Werkzeuge an die Hand gegeben als in vergangenen Jahrzehnten. Die Unterscheidung zwischen Lastenheft, Pflichtenheft und Anforderungsliste bleibt bestehen. Ein *Lastenheft* wird gesehen als die

"schriftliche Anfrage vom Kunden, in dem die Anforderungen des Kunden dokumentiert sind" [PBFG13, S. 319].

Der Inhalt eines Lastenheftes sollte immer das System beschreiben. Es sollte beachtet werden, dass ein Lastenheft vom Kunden geschrieben wird und alle Anforderungen und Wünsche aus Sicht des Kunden enthält. Der Kunde muss dabei nicht extern, sondern kann auch z.B. ein interner Produktmanager sein. Neben dem Lastenheft können auch Normen vom Kunden vorgegeben werden. Das Lastenheft und die Normen bilden zusammen die Sicht auf das Problem. Eine Lösungssicht wird erst im Pflichtenheft integriert. Ein Lastenheft definiert den Zustand, der erreicht werden soll. Es handelt sich quasi um ein strategisches Dokument, welches die Ziele definiert [Pfi17]. Eine anschauliche grafische Darstellung der Inhalte kann in Anhang A2-2 gesehen werden. Nachdem dieses Lastenheft vorliegt, werden die Entwickler in eine *Klärungsphase* [PBFG13] eingebunden. In dieser Phase werden die Kundenanforderungen soweit angepasst und verändert, bis eine realisierbar scheinende Formulierung entsteht, welche als *Pflichtenheft* bezeichnet wird. Ein Pflichtenheft beinhaltet

"den vom Unternehmen erarbeiteten Realisierungsvorschlag und damit die Beschreibung der Umsetzung des vom Kunden vorgegebenen Lastenheftes" [PBFG13, S. 319].

Dieses Pflichtenheft dient in der Industrie als Grundlage für einen Vertrag zwischen Kunde und Entwickler. Der Entwickler löst nun zuerst die Aufgabe der Identifikation der maßgeblichen Anforderungen. Die Auflistung aller identifizierten Anforderungen wird als *Anforderungsliste* bezeichnet. Die Phasen dieser Interpretation einer Erstellung der Anforderungsliste werden anschaulich in Bild 2-18 gezeigt [PBFG13].

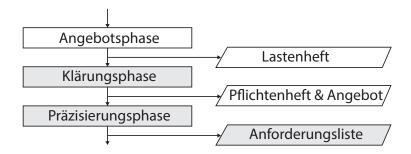


Bild 2-18: Die Phasen zur Erstellung einer Anforderungesliste, abgewandelt nach [PBFG13]

2.5.2 Anforderungen aus der Sicht des Software Engineering

Dieser Abschnitt zeigt eine Beschreibung von Anforderungen, wie sie im Software Engineering interpretiert werden. Dies stellt eine Zusammenfassung der wichtigsten Inhalte bezüglich dieser Arbeit dar und erhebt keinen Anspruch auf Vollständigkeit. Systems und Software Engineering sind dabei Disziplinen, die sich synergetisch unterstützen. Beide haben Errungenschaften hervorgebracht, von denen die andere Disziplin profitiert. Beispiele von Anregungen, welche das Software Engineering durch das Systems Engineering erhielt, sind [FT15]:

- Stakeholder Analyse,
- Requirements Engineering,
- funktionale Dekomposition,
- Schnittstellenspezifikation,
- Traceability und
- systematische Validierung und Verifikation.

Durch das Software Engineering wurden wiederum u.a. folgende Beiträge geliefert [FT15]:

- Kontextdiagramme,
- modellgetriebene Entwicklung,
- UML-SysML,
- Use Cases,
- objektorientierter Entwurf,
- inkrementale Validierung und Verifikation sowie
- Prozessmodellierung.

Diese Ansätze und Methoden bewähren sich längst auch für klassische Ingenieure. Es fällt auf, dass die Errungenschaften auf beiden Seiten mehrheitlich eher dem Systems Engineering zugeordnet werden können. Die Unterscheidung der Disziplinen kann auf einer anderen Ebene gezeigt werden: Eine abstraktere und reduzierte Form der unterschiedlichen Tätigkeiten von Systems und Software Engineering wird entsprechend in Bild 2-19 gezeigt. Die Entwicklung beginnt mit den Nutzeranforderungen, welche in Operationsanforderungen überführt werden. Aus diesen kann anschließend die Spezifikation der Systemanforderungen erfolgen und eine Systemarchitektur entworfen werden. Diese Schritte werden ständig überprüft und haben einen iterativen Charakter. Mit dem Entwurf der Systemarchitektur wird der erste explizite Anknüpfungspunkt des Software Engineering – die Klärung der Softwareanforderungen - deutlich. Der Softwareentwurf bildet dann die zentrale Aufgabe des Software Engineering. Die entwickelte Software wird anhand der Softwareanforderungen verifiziert und in das System integriert. Diese Integration verschiedener Systemkomponenten wird wiederum als Teil des Systems Engineering gesehen. Es sei angemerkt, dass es sich bei diesen Systemkomponenten um Komponenten unterschiedlicher Art handeln kann. Anschließend wird das System anhand der System-, Operationsund Nutzeranforderungen verifiziert und validiert. Diese Schritte werden auch dem Systems Engineering zugerechnet. Ist die Validierung und Verifizierung zufriedenstellend, kann der Prozess aus dieser Sicht beendet werden. Es wird deutlich, dass das Software Engineering eher disziplinspezifisch eingeordnet ist (vgl. [And97]). Es kann sich vorgestellt werden, dass die entsprechenden Schnittstellen zum Entwurf der Systemarchitektur und zur Integration der Systemkomponenten auch durch andere spezifische Disziplinen genutzt werden können. Dies entspricht der Entwicklung nach dem V-Modell (vgl. Abschnitt 2.3.3).

Bereits vor mehr als 20 Jahren waren Auseinandersetzungen mit den unterschiedlichen Betrachtungen zum Thema Systems und Software Engineering Teil von wissenschaftlichen Untersuchungen. Dabei stellte sich die Frage nach Gemeinsamkeiten und Unter-

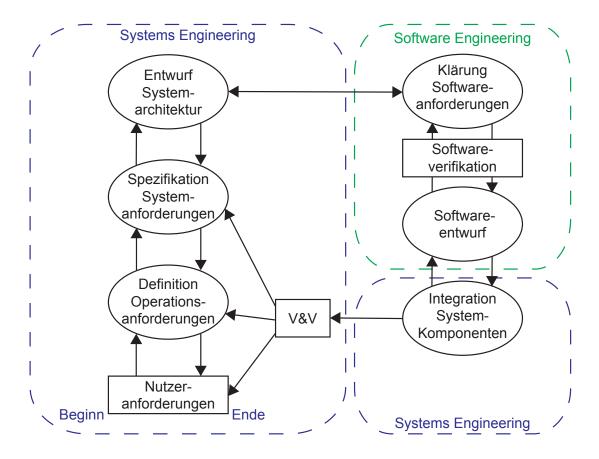


Bild 2-19: Die Sicht des Software Engineering auf das Systems Engineering, abgewandelt nach [FT15]

scheidungen der Disziplinen. Die wissenschaftliche Entwicklung zum Thema Anforderungen im Software Engineering wurde dabei kritisch thematisiert [And97]. Heute steht die Wissenschaft vor (mindestens) zwei praktisch separaten Welten zum Thema Anforderungen. Die wichtige Bedeutung der Anforderungen ist dagegen beiden Welten bewusst. Als einen von fünf prominenten Wegen, ein Softwareprojekt zum Misserfolg zu führen, gibt Lindstrom das ungenügende Anforderungsmanagement an [Lin93]. Brooks gibt in seinem martialisch anmutenden Artikel über silberne Kugeln gegen die *Dämonen der Software* die Detaillierung von Anforderungen als einen Hoffnungsschimmer an. Vielmehr geht er noch einen Schritt weiter und behauptet, die Präzisierung von Anforderungen sei die wichtigste Tätigkeit eines Entwicklers, da kein Kunde die vollständigen Anforderungen an ein modernes System überblicken könne [Bro87].

Zusammenfassung In diesem Kapitel wurde die Ausrichtung der Arbeit detailliert und wichtige Grundlagen gezeigt. Der Begriff der Anforderungen wurde im klassischen ingenieurwissenschaftlichen Kontext beleuchtet und im softwareorientierten Kontext. Die dabei auftretenden Herausforderungen werden im nächsten Kapitel erläutert und es werden Anforderungen an eine Systematik abgeleitet. Anschließend wir der Stand der Wissenschaft beleuchtet.

3 Problemanalyse und Stand der Wissenschaft

Dieses Kapitel dient der problemorientierten Analyse des Standes der Wissenschaft und der Technik. Zuerst werden Entwicklungsprozesse, -methoden und Werkzeuge im Allgemeinen untersucht und anschließend in Hinblick auf Anforderungen. Die Analyseergebnisse fließen in ein Bewertungsverfahren ein, aus dem ein Handlungsbedarf abgeleitet wird.

3.1 Allgemeine Problembeschreibung

Die Zielsetzung, die durch bessere Anforderungen erreicht werden soll, deckt sich mit der Zielsetzung von vielen Entwicklungsmethodiken. Diese sind oftmals zu generisch um alle Detailinformation zu vermitteln und verweisen auf spezialisierte Methoden. Doch gerade im Kontext der Anforderungen sind diese speziellen Methoden häufig zu speziell, um sie generisch anzuwenden. Ein weiteres Problem liegt in den unterschiedlichen Stadien: Entwicklungen mit weitgehend festgelegten technischen Lösungen erfordern andere Anforderungen, als solche mit einem wesentlich offenerem Lösungsraum. Wird ein Produkt im Detail verändert, können viele Bausteine übernommen werden und die Anforderungen sind sehr spezifisch und detailliert. Anders verhält es sich bei Neuentwicklungen; hier steht zu Beginn ein Problem im Vordergrund, für das erst eine Lösung gefunden werden muss.

Grundsätzlich ist die Bedeutung von Anforderungen bekannt, was auch im industriellen Kontext nachgewiesen werden kann [GCW+13]. Bild 3-1 zeigt, dass über mehrere Branchen hinweg, die Bedeutung von Anforderungsmanagement durchweg sehr hoch ist. Es zeigt sich auch, dass gemessen an der hohen Bedeutung, die derzeitige Position der Unternehmen nicht angemessen ist, also ein Defizit vorliegt. Daher ist eine Lösung für dieses Spannungsfeld zu finden. Zu dieser Problematik kommt hinzu, dass Anforderungen häufig mit aufgetretenen Fehlern im Betrieb korrelieren. Es stellt sich die Frage, ob die Systeme in Fehlerfällen nicht richtig funktionieren oder nicht auf eine bestimmte Situation hin entwickelt wurden. Außerdem gilt es zu klären, ob bei Anforderungen das Produkt oder der Prozess im Vordergrund steht: In den USA existiert ein Gerichtsentscheid, dass sich Anforderungen pro Jahr nur um 24 Prozent verändern dürfen. Gibt es Änderungen, die darüber hinaus gehen, ist ein neuer Vertrag zu verhandeln [SSB10].

Die Problematik der Anforderungen ist demnach keine rein akademische, sie ist längst in der Praxis angekommen. Es sollte aus Sicht des Autors festgehalten werden, dass zwei Strategien jedoch nicht zum erhofften Ziel führen:

- Überladung durch Tausende von Anforderungen zu einer Spezifikation mit dem scheinbar vordergründigen Ziel vertraglicher Sicherheit und
- Vernachlässigung von Anforderungen und Verweis auf Agilität.

Um eine Aufgabe hinreichend zu spezifizieren, müsste das Problem mit den Eigenschaften eines *Well Structured Problems* vorliegen: Es müsste eine klare Problembeschreibung, eine klare Zielvorgabe und ein realistischer Weg dorthin skizziert sein. In der Realität

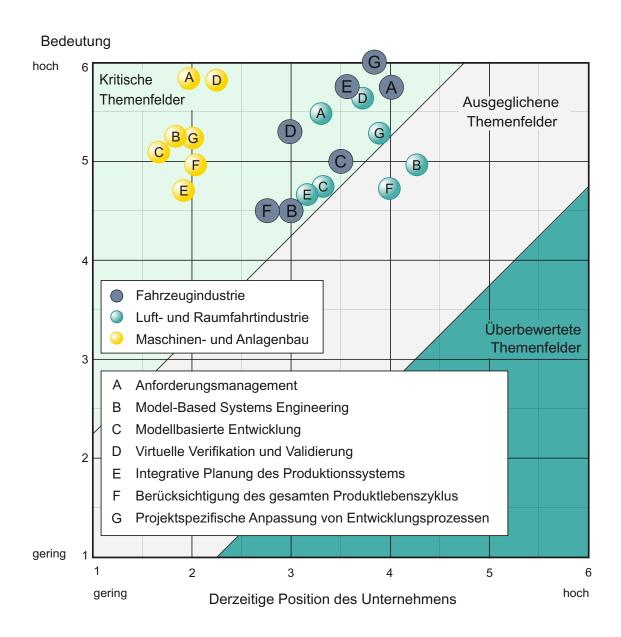


Bild 3-1: Branchenspezifische Einschätzung von unternehmenspezifischen Positionen zu verschiedenen Themenfeldern, abgewandelt nach [GCW+13]

gleicht dieser Versuch aber eher einer Illusion [Sim73]. Formalisierungen zeigen einen Ausweg, sind jedoch nicht perfekte Approximationen der realen Welt [CA07]. Kontinuierliche Prozesse werden in diskrete Analysen überführt, die jedoch entsprechende Nachteile haben. Eine Formalisierung der Anforderungen ist zwar wichtig für die Entwicklung: Das Verständnis für die Situation ist jedoch unbedingt erforderlich und sogar wichtiger als die formaliserte Beschreibung [YZ80]. Das bedeutendste Problem ist die Schaffung eines gemeinsamen Verständnisses aller betroffenen Personengruppen [Rau92]. Dass dies aus einer einzelnen Disziplin heraus – der Softwaretechnik – so gesehen wird, zeigt die Wichtigkeit für alle beteiligten Disziplinen.

3.2 Problemorientierte Analyse

Herausforderungen für Anforderungen werden von mehreren Autoren analysiert. Als kurzer Überblick seien im Folgenden einige genannt: Walia und Carver haben Fehler in Anforderungen im Software Engineering identifiziert, mit dem Ziel, eine Taxonomie dieser Fehler zu entwickeln [WC09]. Zave und Jackson haben in ihrer viel beachteten Arbeit die Herausforderungen an Anforderungen im Detail analysiert; eine dieser Herausforderungen ist adäquates Domänenwissen [ZJ97]: Ziele müssen mit zusätzlichen Informationen kombiniert werden. Die möglichen Einflüsse falsch angenommener Umweltbedingungen werden hervorgehoben. Darüber hinaus wird eine Lücke zwischen Anforderungen und Spezifikationen dargelegt, die immer mit Domänenwissen geschlossen werden muss [ZJ97]. Aus Sicht der Mechatronik kann in der Literatur eine starke Konzentration auf Software Engineering orientierte Ansätze beobachtet werden. Besonders die formalen Analysemethoden von Anforderungen sind dort verankert [FHP06]. Hull, Jackson und Dick legen dieses Ungleichgewicht explizit dar.

"Requirements Engineering ist ein Teil des Systems Engineering, nicht bloß des Software Engineerings" [HJD11, S. 8].

Eine Untersuchung dieser und weiterer Forschungsfragen wird von Cheng und Atlee bereitgestellt [CA07]. Es wird festgehalten:

"Die Umgebung oder der Kontext in welchen ein Software System läuft, ist oftmals der am wenigsten verstandene und unsicherste Aspekt eines beabsichtigten Systems; und RE (Requirements Engineering) Technologien und Werkzeuge zur Integration physikalischer Umgebungen, menschlichen Verhaltens, Schnittstellengeräten und Software Systemen sind unter den am wenigsten fortgeschrittenen" [CA07].

Umfangreiche Analysen zu Problemen mechatronischer Systementwicklung im Allgemeinen sind nicht im Fokus dieser Arbeit, hier sei auf die Literatur verwiesen [AFT⁺10]. In diesem Abschnitt stehen die Herausforderungen in Bezug auf Anforderungen im Vordergrund, welche im Folgenden aufgegriffen werden.

Interdisziplinarität Moderne Systeme sind geprägt durch einen großen Anteil unterschiedlicher beteiligter Disziplinen. Verschiedene Komponenten oder Systemelemente

¹aus dem Englischen übersetzt

²aus dem Englischen übersetzt

werden separat entwickelt, dürfen aber nicht unabhängig voneinander entwickelt werden, daher sollten Ansätze aus dem Systems Engineering angewendet werden, um dieser Herausforderung entgegenzutreten.

Die Integration mehrerer Disziplinen führt aber zu immer größerem Einfluss von Domänenwissen. Jung analysiert in diesem Kontext verschiedene Entwicklungsmethodiken. Dabei seien viele der untersuchten Ansätze für die Anwendung von unterstützenden Anforderungen nicht alleine passend für zukünftige Entwicklungsprojekte, welche komplexer und interdisziplinärer werden [Jun06]. Es werden dementsprechend Experten benötigt; jedoch besteht so die Gefahr von impliziten Annahmen, welche nicht in den Anforderungen berücksichtigt werden. Niknafs und Barry bestätigen einen Einfluss, den eine unabhängige Person ohne Expertenwissen auf ein Team von Experten bei der Anforderungsermittlung hat [NB12].

Vernetzung und Digitalisierung Daten sind die wichtigste Ressource, die in Zukunft die Wertschöpfung maßgeblich beeinflussen wird. Die Gewinnung und Nutzung dieser Daten ist vielfältig und sie werden Produkte und deren Entwicklung begleiten. Das bedeutet, dass die Weichen idealerweise schon in frühen Phasen gestellt werden. Automobilzulieferer werden in Zukunft nicht allein Komponenten herstellen, sondern auch Informationen liefern und bekommen wollen. Mit genug Daten und der Möglichkeit der Auswertung kann quasi jede Herausforderung der Menschheit gelöst werden [SRE14]. Dies ist eine Beschreibung aus IT-Sicht, die bei genauer Betrachtung jedoch scheinbar wenig Gegenargumente zulässt. Daten nehmen einen immer größeren Anteil an Systemen ein und dies sollte berücksichtigt werden. Das in dieser Arbeit verwendete Beispiel eines Geldautomaten zeigt die Problematik auf. Die Zahl der Geldautomaten ist seit dem Jahr 2015 rückläufig. Waren es 2015 noch 61.000 Geräte in Deutschland, liegt diese Zahl aktuell bei 58.000. Das Konträre an dieser Tatsache ist, dass es hier um einen Automaten geht³. Jeder Prozess wird derzeit versucht zu automatisieren und dennoch verliert ein überaus bekannter Automat an wirtschaftlicher Bedeutung [Spi18]. Bargeldlose Technologien auf Datenbasis sägen hier an einem ganzen Geschäftsfeld. Die Frage danach, welche Geldautomaten angesichts dieser Situation zukünftig entwickelt werden sollten, sollte spätestens in den Anforderungen an eine Neuentwicklung geklärt sein.

Produktlebenszyklus Ein System steht heute nicht nur für seine Funktion, sondern auch für seinen vollständigen Produktlebenszyklus. Hierzu gehören Markteinführung, Releaseplanung und Beseitigung, aber auch die Erfüllung der Bedarfe der Nutzer. Es geht nicht mehr um das reine technische Entwickeln, sondern um externe Effekte, die das System grundlegend beeinflussen können. Somit muss dies eine Berücksichtigung finden.

Technologischer Fortschritt Ingenieure können nicht mehr ausschließlich bekannte Technologien anwenden, um Produkte zu entwickeln. Neue, noch nicht vollständig durchdrungene Technologien erreichen den Markt. Hier können Wechselwirkungen entstehen, welche unvorhersehbare Einflüsse auf das System haben und in vielen Bereichen teure Änderungen oder Verzögerungen nach sich ziehen können [KS11].

³Die Kosten, einen Geldautomaten zu betreiben, belaufen sich jährlich auf etwa 20.000 bis 25.000 € [Spi18].

3.2.1 Interdisziplinäre Entwicklungsumgebungen

Truxal formuliert diese Problematik im Jahre 1960 wie folgt:

"In der Tat ist es für den Entwicklungsingenieur vielleicht am wichtigsten, daß [sic] er bei jedem Problem die verschiedenen Methoden kombinieren kann. Er muß [sic] nicht nur eine Anzahl verschiedener technischer Gebiete kennen (Mechanik, Hydraulik, Elektrotechnik, usw.), sondern muß [sic] außerdem imstande sein, die verschiedenen Methoden des Systementwurfs zu kombinieren, um ein endgültiges System realisieren zu können, das Angaben befriedigt, die in der Form eine ungeheure Mannigfaltigkeit annehmen können" [Tru60, S. 331]

Der Entwicklungsingenieur wird es knapp sechzig Jahre später nicht einfacher haben, was im Folgenden deutlich wird.

Komplexitätszuwachs durch Beeinflussung Die steigende Komplexität technischer Systeme ist hinreichend bekannt, doch lauern auch Risiken in der gegenseitigen Beeinflussung von Systemen und Wechselwirkungen.

Elektromagnetische Beeinflussungen dienen im Kontext des Maschinenbaus als wirksames Beispiel für ungewollte Vernetzung, da Feldgrößen in dieser Disziplin eine gewisse Vernachlässigung erfahren. Einerseits können diese Beeinflussungen naturgegeben sein, beispielsweise in der Raumfahrt, oder die Beeinflussungen sind dem System selbst geschuldet. Schwaab unterscheidet dabei zwischen *funktionalen* und *nichtfunktionalen* Quellen für *EMV*. Unter die Gruppe der funktionalen Störquellen fallen in erster Linie Kommunikationssender. Nichtfunktionale Störquellen können KFZ-Zündanlagen, Schweißeinrichtungen oder Leuchtstofflampen sein. Störquellen können schmalbandig oder breitbandig sein und so an mehreren Systemelementen ungewollte Beeinflussungen hervorrufen [Sch96]. Es ist nicht trivial, eine Lösung für solche Beeinflussungen zu finden. Oft tritt dieser Effekt erst im Betrieb in Erscheinung, aber es existieren einige Richtlinien zur Vermeidung solcher Störungen. Es lassen sich weitere dieser ungewollten Effekte in komplexen Systemen aufzählen. Akustik stellt dabei eine weitere bekannte Problematik dar [Par88]. Sobald ein System Schall emittiert, sollte gewährleistet sein, dass dieser nicht in ungünstigen Frequenzbereichen liegt.

Je komplexer ein System wird, desto intensiver treten auch parasitäre Effekte auf. Ein elektrischer Widerstand hat als reales Bauteil eben auch eine Masse, welche auf den ersten Blick nicht zu seinen primären Eigenschaften gehört. Ein anschauliches Beispiel für Komplexität ist das Flugzeug *Boeing 747*. Die insgesamt mehr als 250 km Kabel sind u.a. für die Stromversorgung des gesamten Systems zuständig. Das bedeutet auch, dass die an Bord befindlichen Kaffeemaschinen entsprechend in den Untersuchungen und Tests berücksichtigt werden müssen. Diese dürfen nicht mehr verbrauchen, als geplant. Ansonsten geht dieser Mehrverbrauch zu Lasten wichtigerer Systemelemente. Wird an das Alter des Flugzeugs gedacht und realisiert, dass neuere Systeme einen deutlichen Funktionszuwachs erhalten, wird deutlich, dass ein Bedarf an geeigneten Lösungen existiert. Dabei ist es nicht so, dass keine Lösungen für Einzelprobleme vorliegen. Ein komplexes System wird aus Sicht der Reglungstechnik u.a. durch Zerlegung des Systems beherrsch-

bar gemacht. Solche disziplinspezifischen Ansätze, wie etwa die ϵ -Dekomposition [ZS10] sind jedoch noch nicht systematisch in einen Entwicklungsprozess eingebunden.

Auch der Einfluss informationstechnischer Änderungen auf physikalische Funktionen sind von Wichtigkeit. Diese gegenseitigen Wechselwirkungen in technischen Systemen kann das Fehlen von Informationen oder schlicht die zu langsame Kommunikation innerhalb der gegebenen Strukturen sein. Auch Rückkopplungen durch geschlossene Regelkreise in vorher gesteuerten Systemen können eine Instabilität nach sich ziehen. Oftmals sind diese scheinbar unkritischen Probleme in der Praxis ein sehr großer Faktor bei Systementwicklungen. Zu technisch gedachte Lösungen bedürfen einer Korrektur bezüglich der einsetzbaren und verfügbaren Technologien: Dies sollte bereits in den Anforderungen klar sein bzw. werden. An dieser Stelle schließt sich der Kreis zu den interdisziplinären Systemen. Diese müssen ganzheitlich interdisziplinär entwickelt werden und ganzheitlich bedeutet von den Anforderungen bis letztlich zum Ende des Lebenszyklus.

3.2.2 Diskrete Modelle vs. kontinuierliche Systeme

In der Literatur über künstliche Intelligenz [RN04] wird zur Bewegungsplanung von Robotern darauf hingewiesen, dass im Allgemeinen auf die Betrachtung des dynamischen Zustands (also auf Differentialgleichungen) verzichtet wird, obwohl dies zu einer überlegenen Leistung führen würde, wenn entsprechend geplant werden könnte. Dies wäre jedoch nur für die allereinfachsten Roboter praktikabel [RN04]. Aus Sicht der Computerwissenschaften ist das eine möglicherweise geeignete Annahme. Dies führt jedoch aus ganzheitlicher Sicht zu Problemen: Nicht nur, dass dynamische Einflüsse vernachlässigt werden, oft wird physikalisches Verhalten als diskret angenommen⁴. Dies ist in einigen Disziplinen zielführend – für die ganzheitliche Betrachtung von Anforderungen im Kontext intelligenter technischer Systeme scheint es jedoch nicht die ideale Herangehensweise zu sein. Es wird ein Verständnis für alle Problemklassen benötigt. Über diskrete und kontinuierliche Betrachtungsweisen sollte sich ein Entwickler zumindest bewusst sein. Letztlich ist eine kontinuierliche Betrachtung realitätsnäher. Manchmal wird ein Problem jedoch einfacher durch eine diskrete Betrachtung, wie eine Formalisierung. Solche Formalisierungen sind jedoch nicht perfekte Approximationen kontinuierlicher Effekte, daher werden auch aus Sicht des Software Engineerings bessere Abstraktionen benötigt, um das Verhalten physikalischer und menschlicher Entitäten und deren Schnittstellen mit Software zu modellieren [CA07].

3.2.3 Konsistenz in Entwicklungsprozessen

Die MBSE Initiative sieht Modelle als immer wichtiger werdende Ergebnisse von Systems Engineering Prozessen [FGS07]. Dennoch fällt der Übergang zwischen klassischen Systemmodellen und Modellen des dynamischen Verhaltens schwer. Besonders deutlich wird dies an dem Begriff des Digital Twin. Ein Digital Twin ist eben nicht ein einziges Modell, wie in Bild 3-2 gezeigt. Dieses stellt lediglich eine mögliche Sichtweise auf das System dar. Der Digital Twin sollte vielmehr die komplette Bandbreite an Modellen sein,

⁴Das es auch im Kontext der Robotik anders geht, zeigt u.a. Neugebauer [Neu06].

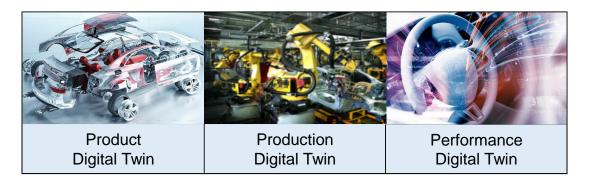


Bild 3-2: Der Digital Twin ist nicht ein einziges virtuelles Abbild, sondern kann aus unterschiedlichen Sichten betrachtet werden, abgewandelt nach [Lud17]

die vorstellbar sind. Also Systemmodelle, CAD- und FEM-Modelle, dynamische Modelle und viele weitere. Alle diese sind in unterschiedlichen Phasen der Entwicklung und der Nutzung ein Teil des Digital Twin. Diese Denkweise ist essentiell, um die vollen Potentiale zu heben. Die virtuelle Entwicklung ist sicher ein wesentlicher Baustein zur Konsistenz in Entwicklungen, dafür benötigt es jedoch ein gemeinsames Ziel der Modelle. Es werden geeignete Prozesse benötigt und die richtigen Schnittstellen müssen verfügbar sein. Auf diesem Gebiet ist sicher viel Forschungsarbeit nötig, bevor es zu einer anwendbaren Lösung kommt.

3.2.4 Emergente Systeme

Der Schritt zu einer durchgängigen Entwicklung wird maßgeblich durch eine Eigenschaft behindert: Emergenz. Emergente Systeme zeichnen sich durch ihre Systemeigenschaften aus, welche über die einzelnen Eigenschaften der Bestandteile hinausgehen. Auftretende Effekte können nur über das gemeinsame Zusammenspiel erklärt werden, daher sind isolierte Betrachtungen nicht zielführend, sodass die klassische Dekomposition von Entwicklungsprozessen hier nicht angewendet werden sollte. Eine beschränkende Annahme, ein System könne durch die Zerlegung in die Einzelbestandteile verstanden werden, wird in manchen Fällen für eine erfolgreiche Systementwicklung nicht ausreichend sein. Diese Eigenschaft kann natürlich auch positiv gedeutet werden. So kann der berühmte Ausdruck, etwas sei größer als die Summe seiner Teile hierauf angewendet werden.

Ein einfaches Beispiel eines emergenten Systems stellt der ungedämpfte Ein-Massen-Schwinger dar. Dieser besteht aus einer Masse und einer Feder. Für sich genommen, werden die dynamischen Eigenschaften der Masse durch ihren Trägheitstensor und ihre Masse charakterisiert. Die dynamische Beschreibung der Feder erfolgt über die Federsteifigkeit und ihre Länge. Beim Zusammenspiel dieser beiden Elemente tritt jedoch ein Effekt auf, welcher keinem der beiden Elemente zugeordnet werden kann. Das System der beiden Elemente ist schwingungsfähig und hat eine Resonanzfrequenz. Die implizite Annahme über die Existenz eines solchen Effektes funktioniert bei kleineren Systementwicklungen zweifelsfrei. Komplizierter wird es jedoch u.a. schon bei Regelungen. Diese können durch das Schließen des Regelkreises ein für sich genommenes stabiles System gar instabil werden lassen. Einerseits birgt diese Eigenschaft eine große Gefahr für die

erfolgreiche Funktion von Systemen. Positiv betrachtet scheint hier andererseits auch ein großes Potential zu liegen, neue Funktionen zu realisieren. Diesen Herausforderungen sollte nicht mit realen Prototypen begegnet werden und das macht die (modellbasierte) Analyse der Wechselwirkungen innerhalb eines Systems umso bedeutender.

3.3 Problemabgrenzung und Anforderungen an eine Systematik

Um diesen Problemen zu begegnen, müssen Kriterien gefunden werden, anhand welcher eine Bewertung der Eignung unterschiedlicher Ansätze vorgenommen werden kann. Zuerst werden daher Anforderungen noch einmal beleuchtet, um ein eindeutiges Verständnis zu erlangen und anschließend werden Anforderungen an eine Systematik aufgenommen.

3.3.1 Problemabgrenzung

Dieser Abschnitt soll auf die unterschiedlichen Interpretationsmöglichkeiten von Anforderungen hinweisen. Oftmals werden Anforderungen zu Beginn eines Entwicklungsprozesses benötigt. Anlehnend an die geforderten Eigenschaften für Anforderungen aus der ISO/IEC/IEEE 29148:2011 ist dies jedoch nicht immer umsetzbar. Eine Anforderung kann mitunter nicht am Anfang einer Entwicklung korrekt, eindeutig oder überprüfbar sein. Anforderungen können ggf. auch nicht vollständig sein, weder im Sinne der Anzahl, noch in Bezug auf ihre Eigenschaften, wie eine Identifikationsnummer, eine Beschreibung oder weitere.

Zu Beginn sollten daher eher Ziele formuliert werden. Diese Ziele beschreiben Aufgaben, die ein System erfüllen soll [van01]. Es kann eine Lücke zwischen solchen Zielen und detaillierten technischen Anforderungen gesehen werden. Ziele können bereits Funktionen repräsentieren, welche realisiert werden sollen [Ale77]. Von diesem Gesichtspunkt aus werden funktionale Anforderungen relevant: Also welche Funktionen benötigt werden, um die Ziele des Systems zu erreichen [Nat07]. Leistungsanforderungen definieren, wie gut eine Funktion erfüllt werden muss. Schnittstellenanforderungen beschreiben die internen und externen Schnittstellen des Systems. Solche können u.a. sein: Beschleunigungen, Vibrationen, Akustik, Wärme und elektromagnetische Kompatibilität. All diese beeinflussen technische Aspekte des Systems. Dementsprechend beziehen sich technische Anforderungen eben auf alle diese Bereiche. Ein gemeinsames, domänenübergreifendes Verständnis von Anforderungen ist, wie dargelegt, im Allgemeinen nicht gegeben. Im Wesentlichen lassen sich für technische Systeme sechs übergeordnete Unterscheidungsmerkmale von Anforderungen identifizieren (vgl. Bild 3-3). Eine Unterscheidung nach Disziplinen bedeutet, dass eine Anforderung einer bestimmten Disziplin zugeordnet werden kann. Softwaretechniker können andere Anforderungen als Ingenieure haben. Ebenfalls können sich die betrachteten Anforderungen von Maschinenbauern und Elektrotechnikern unterscheiden. Ein weiteres Unterscheidungsmerkmal ist die Detailliertheit. Anforderungen können eher abstrakt oder bereits sehr konkret sein. In einem üblichen Entwicklungsprozess werden sich Anforderungen im Laufe der Zeit zu immer detaillierteren Formen entwickeln. Ebenfalls üblich ist eine Unterscheidung nach der Quelle der Anfor-

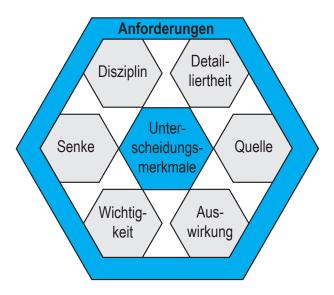


Bild 3-3: Übergeordnete Unterscheidungsmerkmale für Anforderungen

derung. Dies betrifft alle Stakeholder⁵. Für dieses Unterscheidungsmerkmal der Quelle sind Kundenanforderungen das bekannteste Beispiel. Es können sich jedoch Anforderungen aus jedem Stakeholder ergeben, sodass auch von einer Unterscheidung nach Stakeholdern gesprochen werden könnte. Die Auswirkung einer Anforderung kann ebenfalls als Unterscheidungsmerkmal genutzt werden. Die Frage, die hier beantwortet wird ist: Welchen Effekt hat eine Anforderung auf ein Produkt oder einen Prozess? Dieses Unterscheidungsmerkmal stellt das Bindeglied zwischen der Unterscheidung nach der Quelle und der Unterscheidung nach der Wichtigkeit einer Anforderung dar. Die Wichtigkeit als Unterscheidungskriterium beruht vor allem auf der Priorisierung von Anforderungen. Es kann sich bei einer Anforderung um eine kritische fest zu erfüllende Anforderung handeln oder z.B. um einen Wunsch. Die Unterscheidung nach dem Merkmal Senke beruht auf bekannten Kategorien, wie z.B. Sicherheits-, Funktional-, Schnittstellen-, Leistungsoder Umfeldanforderungen. Die entscheidende Frage an dieser Stelle ist: Für was ist die Anforderung gedacht? Eine Anforderung kann nach mehreren der vorgestellten Merkmale unterschieden werden. Je nach betrachtetem Wissenschaftsgebiet wird eine der sechs Unterscheidungsmerkmale bevorzugt. Dadurch wird eine disziplinübergreifende Betrachtung des Themas Anforderungen erschwert.

Auf weitere mögliche Unterscheidungsmerkmale aus entfernteren Gebieten wird nicht eingegangen. Der Begriff wird ebenfalls verwendet, beispielsweise in Sozialsystemen – z.B. Anforderungen eines Berufes –, in juristischen Gegebenheiten – z.B. in der Strafprozessordnung –, in der Pädagogik – z.B. an Bildungsstandards – oder in der Medizin, Biologie, Chemie und Physik – z.B. die Anforderungen an Experimente.

⁵Eine interessante Diskussion zu Stakeholdern findet sich in der Literatur [MAW97].

3.3.2 Anforderungen an eine Systematik

Die Anforderungen die an eine Systematik gestellt werden, welche Anforderungen entwickeln soll, liegen einerseits in der Betrachtung der Anforderungen und deren Berücksichtigung im Entwurfsprozess, andererseits liegen sie in der interdisziplinären Behandlung und der Analysemöglichkeiten. Um den Herausforderungen zu begegnen, die in diesem Kapitel erläutert sind, müssen konkret die folgenden Anforderungen erfüllt sein:

A1 Systematische Synthese von Anforderungen:

Ein systematischer Prozess zur Synthese von Anforderungen sollte eine Systematisierung berücksichtigen und Zielrichtungen innerhalb der Systematik vorgeben. Anforderungen sollten mittels einer Art Schablone ermittelt und zugeordnet werden können. Dabei wird vor allem der Prozess der Detaillierung der Anforderungen adressiert. Anforderungen aus Checklisten zu sammeln ist zwar systematisch, jedoch wird die Synthese vernachlässigt. Anforderungen können auf verschiedene Ebenen bezogen sein und ein **systematischer Prozess** zur Synthese sollte dies berücksichtigen.

A2 Mathematisch interpretierbare Analyse von Anforderungen:

Die Berechnung einer Kennzahl zur Bewertung von Anforderungen sollte nicht nur auf eine konkreten singuläre Erkenntnis abzielen, sondern Analysen über die Struktur der Anforderungen, die Vollständigkeit und die Priorisierung ermöglichen. Dabei sollte als Kontext das System mit seinen Elementen dienen und die Vernetzung der Anforderungen mit diesem System Berücksichtigung finden. Eine Systematik sollte **eine Analyse enthalten**, die Anforderungen kardinal auf Vollständigkeit und Priorisierung überprüft.

A3 Darstellen von Anforderungen und modellbasierte Übertragbarkeit:

Die Darstellung von Anforderungen ist nicht allein im Sinne einer Dokumentation zu sehen. Ebenfalls sollten die Anforderungen Verknüpfungen enthalten können und einfach zu interpretieren sein, damit die Handhabung der Anforderungen erleichtert wird. Sie haben eine tragende Rolle in der Entwicklung und bilden Anknüpfungspunkte in den Entwicklungsprozess. Anforderungen in einer Liste sind z.T. schwer nachzuvollziehen. Eine grafisch einfach zu interpretierende Darstellung ist wünschenswert. Diese sollte anhand später zu erstellender Modelle eine gewisse Struktur besitzen.

A4 Einbindung in einen Entwurfsprozess (V-Modell):

Anforderungen werden für die Entwicklung eines Systems erstellt. Für diese Entwicklung ist eine Entwicklungsmethodik hilfreich, um z.B. die Komplexität zu beherrschen. Die Systematik der Anforderungen sollte demnach auch **in eine Entwicklungsmethodik eingebettet** werden. Sie müssen den Entwicklungsprozess unterstützen und an sinnvollen Stellen eingeordnet sein.

A5 Ganzheitliche Betrachtung des Entwurfsprozesses:

Anforderungen sollten nicht ausschließlich zu Beginn berücksichtigt werden. Sie entwickeln sich, genau wie das System selbst. Eine Allokation und Definition von Anforderungen zu einem ganzheitlichen Entwurfsprozess ist erforderlich.

Durch die ganzheitliche Sicht auf den Prozess müssen die Anforderungen selbst ganzheitliche Eigenschaften besitzen, um den Herausforderungen interdisziplinärer Systementwicklungen entgegentreten zu können.

A6 Durchgängige Berücksichtigung von Anforderungen:

Im Vergleich zur vorherigen Forderung nach ganzheitlicher Betrachtung wird hier eine durchgängige Berücksichtigung gefordert. Maßnahmen zur Detaillierung und **Transition von einer Anforderungsstufe zu einer nächsten** werden gefordert. Das bedeutet, dass die Anforderungen nicht an einem Punkt der Entwicklung abgeschlossen sind, sondern von Beginn an bis zum Test Berücksichtigung finden sollten. Dabei sollen sie die Übergänge der Systementwicklung unterstützen und sich selbst ebenfalls weiterentwickeln.

A7 Erhöhung der Konsistenz des gesamten Entwurfs:

Die Systematik sollte die Konsistenz des Entwurfs vergrößern. Anforderungen stellen ein wirkungsvolles Hilfsmittel bei der Entwicklung von Systemen dar. Daher sollten Anforderungen eben jener Entwicklung nicht im Weg stehen, sondern diese sinnvoll und **zielgerichtet unterstützen**. Sie bilden damit keinen Selbstzweck, sondern dienen der Systementwicklung und sind einer erfolgreichen Systementwicklung sinnvoll unterzuordnen.

A8 Bereitstellung einer Klassifikation von Anforderungen:

Eine **Kategorisierung** von Anforderungen sollte für alle beteiligten Disziplinen beim Entwurf intelligenter technischer Systeme eine Unterstützung bieten und die Entwicklung unterstützen, nicht bloß gewisse Eigenschaften von Anforderungen unterscheiden. Daher sollte eine Kategorisierung von Anforderungen in Hinblick auf Kriterien vorgenommen werden, welche die hier gestellten Anforderungen an eine Systematik erfüllen.

Die aufgezählten Anforderungen bilden den objektiven Bewertungsrahmen für unterschiedliche Ansätze. Aus wissenschaftstheoretischer Sicht gelten einige weitere Parameter der Systematizität, also der Eigenschaft strukturierten Wissens und somit Grundlage guter Systematiken, wie [Res87]:

- Vollständigkeit,
- Zusammenhang,
- Konsistenz,
- funktionelle Sicherheit,
- funktionelle Einfachheit.
- funktionelle Wirksamkeit.

Diese sollten möglichst auf jedes zu konstruierende wissenschaftliche System zutreffen. Während die ersten drei Stichpunkte eindeutig zu interpretieren sind, bedarf es bei den unteren drei Stichpunkten ggf. einer Erläuterung. Funktionelle Sicherheit bedeutet, ein System folgt gewohnten Gesetzmäßigkeiten. Funktionelle Einfachheit bezieht sich auf die Ökonomie des Systems. Die Systemelemente sollten aufeinander abgestimmt sein und möglichst die größtmögliche Eleganz aufweisen. Die funktionelle Wirksamkeit bezieht sich auf die Effizienz des Systems bzw. seiner Beschreibung. Effektivität und Angepasstheit an den Zweck stehen im Vordergrund [Res87].

3.4 Stand der Wissenschaft

Die Analyse des Standes der Wissenschaft erfolgt anhand übergeordneter Kategorien. Es werden Ansätze des softwareorientierten Requirements Engineering betrachtet und Entwicklungsmethodiken aus ingenieurwissenschaftlicher Sicht, dabei wird der Kontext zum Systems Engineering behalten. Es werden jedoch auch spezielle Ansätze für Anforderungen untersucht. In Richtlinien und Normen wird die Berücksichtigung von Anforderungen ermittelt. Darüber hinaus werden einige Softwarewerkzeuge betrachtet.

3.4.1 Requirements Engineering Frameworks

Ihrer Bekanntheit und Relevanz für das Requirements Engineering geschuldet, werden in diesem Abschnitt einige ausgewählte Frameworks beschrieben. Sehr softwarelastige Ansätze werden nicht analysiert, da sie zu wenig Anknüpfungspunkte bieten. Nähere Informationen finden sich in der Literatur [Gra98b].

Requirements Engineering nach EBERT

In seinem vierhundertfünfzigseitigem Buch thematisiert Ebert umfangreich verschiedene Bereiche des Requirements Engineerings. Das praxisnahe Buch startet mit dem Zitat

"Wer sein Ziel nicht kennt, kann jeden Weg gehen."

Dies ist zwar aus dem fachfremden Buch *Alice im Wunderland*, es repräsentiert jedoch die Notwendigkeit von Zielen und Anforderungen [Ebe10]. Es wird von drei wichtigen Risiken im Requirements Engineering ausgegangen:

- Fehlende Anforderungen,
- falsche Anforderungen,
- sich ändernde Anforderungen.

Von der Erkenntnis, dass diese drei Risiken nicht softwarespezifisch sind, wird die Anwendbarkeit des Requirements Engineering auf jegliche Produktentwicklung abgeleitet, explizit auch für den Maschinenbau. Anforderungen werden aus unterschiedlichen Sichten betrachtet. Diese sind *Marktanforderungen*, *Produktanforderungen* und *Komponentenanforderungen*. Es handelt sich um eine rekursive Interpretation, da Komponentenanforderungen bei einem Zulieferer ggf. Marktanforderungen sein können [Ebe10]. Zusätzlich wird zwischen unterschiedlichen Arten unterschieden. *Funktionale Anforderungen* beschreiben eine vom System bereitzustellende Funktion. *Qualitätsanforderungen* beschreiben qualitative Eigenschaften des Systems. *Randbedingungen* sind organisatorischer oder technischer Natur und beschränken die Realisierungsmöglichkeiten eines Systems.

Requirements Engineering wird definiert als das

"disziplinierte und systematische Vorgehen zur Ermittlung, Spezifikation, Analyse, Vereinbarung, Validierung und Verwaltung von Anforderungen, um Bedürfnisse und Ziele in ein Produkt umzusetzen" [Ebe10, S. 32].

Der dafür vorgesehene methodische Ansatz unterscheidet *stringentes*, *iteratives* und *agiles* Requirements Engieering sowie Methoden für *extern beschaffte Komponenten* und *Dienste*. Die Ermittlung der Anforderungen startet mit dem Kunden und führt zu den drei Anforderungsarten. Hierfür werden Schritt-für-Schritt Prozesse bereitgestellt, die praxisnah orientiert sind. Als Spezifikation wird die geeignete Dokumentation der Anforderungen beschrieben und es werden verschiedene Templates für unterschiedliche Arbeitsergebnisse bereitgestellt. Daraufhin wird die Analyse der Anforderungen diskutiert, mit dem Ziel, eine Lösung zu synthetisieren [Ebe10]. Dabei wird eine modellbasierte Unterstützung vorgeschlagen. Die Modelle beruhen auf der UML. Es wird eine sequentielle Vorgehensweise mithilfe der Modelle beschrieben. Gestartet wird mit Kontextdiagrammen, mit konsolidierter Ergebnisdokumention wird geendet. Es werden auch weiterführende Methoden, wie das Quality Function Deployment zur Analyse von Anforderungen vorgeschlagen.

Die Messung der Qualität von Anforderungen ist für Ebert eine Möglichkeit, die Qualität des zu entwickelnden Produkts zu bewerten. Es wird vorgeschlagen, die folgenden Metriken zu verwenden [SSB10]. Die Zahl der Anforderungen in einem Projekt als quantitatives Maß, dient der Beurteilung des späteren Umfangs der Software. Je mehr Anforderungen, desto mehr muss die Software leisten und daher kann angenommen werden, dass der Umfang steigen wird. Der Fertigstellungsgrad der Anforderungen zeigt an, wie viele Anforderungen bereits abgenommen, also geprüft und bestätigt sind. Darüber hinaus wird eine Unterteilung der nicht abgenommen Anforderungen vorgenommen: In vereinbarte Anforderungen, welche ausführlich beschrieben sind und in noch zu liefernden Anforderungen, welche lediglich als Überschrift existieren. An dieser Stelle sei angemerkt, dass diese Unterteilung der Fertigstellung den Entwicklungsprozess der Anforderungen verdeutlicht. Eine Anforderung durchläuft dementsprechend eine Entwicklung, bevor sie als fertige Anforderung auftaucht.

Die Änderungsrate der Anforderungen deutet auf eine gewisse Stabilität des Projektes hin [SSB10]. Es wird der relative Anteil der geänderten und im Nachhinein hinzugefügten Anforderungen berechnet. Außerdem wird die Zahl der Änderungsursachen aufgenommen, da es verschiedene Gründe für Änderungen geben kann, die erfasst werden sollen. Etwas komplexer ist die Erfassung der Vollständigkeit des Anforderungsmodells. Als Beispiel sei angenommen, Anforderungen werden als Anwendungsfall spezifiziert [SSB10]. Es sollte entsprechend deutlich werden, welche Anforderungen zu welchem Anwendungsfall gehören. Ziel sollte es sein, alle Anforderungen in den Anwendungsfällen wiederzufinden. Letztlich sollte der prozentuale Anteil, der nicht wiederzufindenden Anforderungen möglichst gering sein. Allgegenwärtig beim Thema Anforderungsanalyse sind Prüfungen der bekannten Eigenschaften, die Anforderungen besitzen sollten. Auch darüber hinausgehende Eigenschaften werden berücksichtigt. Typischerweise werden Testbarkeit, Verfolgbarkeit, etc. abgeprüft. Zusätzlich werden andere Kriterien geprüft, wie die Formulierung. Abweichungen von diesen Sollvorgaben werden als Mangel bezeichnet. Die Anzahl der Anforderungsmängel lässt Rückschlüsse auf die Qualität der Anforderungsspezifikation zu. Hier wird ebenfalls die Anzahl der Mängelarten als bedeutende Erkenntnis vorgeschlagen, welche Rückschlüsse auf die Ursachen der Mängel erlaube [SSB10]. Die Zuweisung eines Nutzwertes jeder Anforderung führt laut Ebert dazu, dass Anforderungen entsprechend priorisiert werden können. Dieser Nutzwert kann finanziellen Charakter haben. Der Nutzwert der Gesamtentwicklung entspricht dann der

Summe der Nutzwerte jeder Anforderung: Dies erlaube eine Fortschrittskontrolle und bringe demnach einen weiteren Nutzen.

Requirements Engineering nach Hull et al.

In diesem Ansatz wird Requirements Engineering als elementarer Bestandteil aller Entwicklungsschritte betrachtet. Es werden Anforderungen auf Kundenebene, auf Systemebene, auf Subsystemebene und auf Komponentenebene gesehen. Als Testebenen werden der Komponententest, der Integrationstest, der Systemtest und der Akzeptanztest genannt [HJD11]. Dabei wird Traceability als Verbindung zwischen diesen Ebenen definiert, also ist der Entwicklungspfad einer Anforderung durch die Detaillierung gegeben. Bei der konkreten Dokumentation von Anforderungen werden zwei Forderungen gestellt:

- Das Anforderungsdokument muss lesbar sein.
- Jede Anforderung muss handhabbar sein.

Der Fokus liegt dabei einerseits auf der Struktur des Dokumentes und andererseits auf der Klarheit und Präzision der einzelnen Anforderungen. Bei den Anforderungen wird zwischen Problembereich und Lösungsbereich unterschieden. Die Übergänge erfolgen mithilfe von Modellen aus der UML. Dies unterstreicht die softwareorientierte Herangehensweise, obwohl insgesamt ein ganzheitlicher Ansatz überwiegt. Das systematische Vorgehen von der Problemdomäne an, bei der auf Stakeholderebene agiert wird, unterstützt die Erhebung von Anforderungen stark. Die Strukturierung der Stakeholderwünsche und -szenarien führt zu den Stakeholderanforderungen. Diese sollten nicht technischer Natur sein. Beim Übergang in die Lösungsdomäne werden geführte Prozesse vorgeschlagen. Als Anforderungsfokus dienen [HJD11]:

- Interne Funktionalität: Die Beschreibung dessen, was das System tun soll.
- Schnittstellenfunktionalität: Die Interaktion des Systems mit anderen Systemen.
- Menschliche Interaktionsfunktionalität: Die Beschreibung der benötigten Interaktionsmöglichkeiten mit dem Menschen.
- Sicherheitsfunktionalität: Die sicherheitsrelevanten Einflüsse des Umfelds.
- Systemtransaktionen: Die Szenarien der Benutzung des Systems.
- Operationsmodi: Die Modi, die sich von den typischen Anwendungsszenarien unterscheiden.
- Zusätzliche Beschränkungen: Sonstige Anforderungen.

Die Verfeinerung zu Subsystemanforderungen erfolgt mithilfe der Systemarchitektur und der damit einhergehenden Identifikation der Subsysteme. Die Komponentenanforderungen werden dagegen nicht gesondert behandelt.

Requirements Engineering nach Partsch

In der ersten Auflage des Buches Requirements Engineering systematisch von 1998 gibt Partsch an, die Zielsetzung des Requirements Engineering sei "das Finden eines theoretisch fundierten Beschreibungsmittels, das methodisch sauber eingebettet ist und durch geeignete Werkzeuge unterstützt wird" [Par98, S. 15].

Der Inhalt bezieht sich daher stark auf die entsprechenden *Beschreibungsmittel*, geht jedoch auch auf Methodik und Werkzeuge ein. Es wird davon ausgegangen, dass Tätigkeiten zu Beginn von Projekten im Vordergrund stehen, dennoch wird darauf hingewiesen, dass bei einer schrittweisen Entwicklung, das Requirements Engineering ggf. erst am Ende des Projektes beendet ist. Anforderungen werden in diesem Kontext als Aussagen über zu erbringende Leistungen definiert [Par98] und traditionell in *funktionale* und *nichtfunktionale Anforderungen* unterteilt. Zu der Zeit galt laut Partsch, dass kein Ansatz alle Aspekte des Requirements Engineering erfülle⁶. Dementsprechend wird ein ganzheitlicher Ansatz angestrebt und durchgehend gefordert. So ist z.B. die Verifikation und Validierung der Anforderungsbeschreibung Teil des vorgeschlagenen Vorgehens. *Testbarkeit* wird als wichtige Eigenschaft einer Anforderung gefordert. Die zu der Zeit bekannten *Methoden* setzten in erster Linie gesunden Menschenverstand voraus [Par98]. Es werden Kategorien von Sichtweisen auf ein System vorgestellt. Diese sind:

- Funktions-orientiert.
- Ablauf-orientiert,
- Ereignis-orientiert,
- Datenfluss-orientiert,
- Datenstruktur-orientiert,
- Aufgabenbereich-orientiert,
- Anlagen-/Geräte-orientiert.

Als Aufgabe von *Werkzeugen* im Requirements Engineering wird die Verwendung formaler und semiformaler Sprachen angegeben. Es wird unterschieden zwischen:

- Dokumentationswerkzeugen,
- Analysewerkzeugen,
- Simulationswerkzeugen,
- Codeerzeuger.

Das Ziel dieser Werkzeugkategorien ist das Erstellen von Code. Im Weiteren untersucht Partsch mehrere Ansätze. Einerseits wird die Kategorie *Formalismen und Konzepte*, wie z.B. Diagramme, Datenflusspläne, Automaten und Modelle beschrieben. Diese sind stark auf die Softwaresicht bezogen. Darüber hinaus wird die Kategorie *Strukturierte Methoden* genutzt. Hier werden Methoden, wie *Structured Analysis* beschrieben. *Objektorientierte Methoden* stellen eine weitere Kategorie dar. *Object oriented Analysis* oder UML sind Beispiel dieser Kategorie. In der nachfolgenden Auflage aus dem Jahr 2010 [Par10] modernisiert Partsch sein Werk und fügt z.B. Beschreibungen der SysML hinzu. Es bleibt ein Werk für die Entwicklung softwaregestützter Systeme.

⁶Es wird davon ausgegangen, dass die wesentlichen Ansätze von Рактэсн betrachtet wurden.

Requirements Engineering nach Ронь

Als Motivation für die Beschäftigung mit Anforderungen sieht Рон u.a. den steigenden Anteil an Software in Systemen [Poh10]. Anforderungen dienen dabei als Schnittstelle zwischen Qualität, Projektmanagement, Entwicklung und Wartung. Es werden drei Typen von Anforderungen unterschieden: Funktionale Anforderungen, Qualitätsanforderungen und Randbedingungen. Unter die Kategorie Qualitätsanforderungen fallen die typischen nicht-funktionalen Anforderungen. Diese seien zu wenig spezifiziert und manchmal eben funktionaler Natur. Verschiedene Methoden zur Repräsentation von Anforderungen stehen zur Verfügung, wie z.B.:

- i*,
- KAOS,
- AND/OR.

Entscheidend werden dabei Szenarien gesehen, welche den Kontext für Anforderungen bereiten. Grundsätzlich kann dabei zwischen verschiedenen Formen unterschieden werden. Dies können Momentanszenarien und Wunschszenarien sein, aber auch positive und negative Szenarien und viele mehr. Die Ermittlung von Anforderungen erfolgt durch Interviews, Checklisten, Workshops und weiteren Methoden. Als Dokumentation wird einerseits strukturierter Text genannt, andererseits auch

- Entity Relationship Modelle,
- Data-Flow Modelle,
- Verhaltensmodelle (Finite Automaten und Zustandsautomaten),
- SysML.

Der Ansatz behandelt weitestgehend die frühen Phasen des Entwurfs [Poh10]. Gerade bei den Verhaltensmodellen fällt der Fokus auf diskrete Systembetrachtung auf. Die Priorisierung von Anforderungen erfolgt entsprechend der Stakeholder. Vorgeschlagene Methoden hierfür sind das KANO Modell, der Cost Value Approach oder die Wiegers Priotisation Matrix, welche eine Rangfolge entsprechend Kosten und Nutzen der Anforderungen produziert. Der eigens entwickelte Ansatz nennt sich COSMOD-RE [PS07]. Dieser basiert auf vier Abstraktionsebenen: System Layer, Functional Components Layer, HW/SW Components Layer, Software Deployment Layer [PS07]. Der Ansatz dient der parallelen Entwicklung von Anforderungen und Architektur und basiert auf einer softwareorientierten Vorgehensweise.

Requirements Engineering nach Rupp

Als wohl bekannteste Vertreterin des Requirements Engineering in Deutschland gilt die Gruppe der *Sophisten* um Rupp. Die umfangreiche Beschreibung lässt sich im Zuge dieser Arbeit auf die Bereiche

- Anforderungen ermitteln und
- Anforderungen validieren

fokussieren. Für die vielen weiteren Erkenntnisse sei auf die Literatur verwiesen [Rup07] [Rup09]. Es wird zwischen folgenden Anforderungsarten unterschieden:

- Funktionale Anforderungen,
- technische Anforderungen,
- Anforderungen an die Benutzerschnittstelle,
- Qualitätsanforderungen,
- Anforderungen an sonstige Lieferbestandteile,
- Anforderungen an durchzuführende Tätigkeiten,
- rechtlich-vertragliche Anforderungen.

Diese können in unterschiedlichen Detaillierungsgraden vorliegen. Zur Entwicklung von Systemen werden mehrere Vorgehensmethoden vorgestellt. Einerseits sind dies Vorgehensmodelle, wie das V-Modell, andererseits Werkzeuge, wie das Simulationsmodell. Dieses wird bei zunehmender Komplexität des Anforderungsdokumentes empfohlen. Es kann dazu dienen, Abläufe zu dokumentieren und Abstimmungen vorzunehmen. Die bessere Vorstellung von Anforderungen kann so erreicht werden, was die Änderungstendenz von Anforderungen um 10-40 % reduzieren kann [Rup07]. Das Simulationsmodell wird einem Prototypen gleichgesetzt, welcher anhand von drei Kriterien unterschieden werden kann [Rup07]:

- Die Teilmenge der abgebildeten Anforderungen (horizontal/vertikal),
- Die Wiederverwendung des Prototyps (Wegwerfprototyp/Evolutionärer Prototyp),
- Das zur Erstellung verwendete Medium (Papier/Elektronisch).

Zur Erstellung dieser Prototypen sind unterschiedliche Aufwände nötig. Als repräsentative Beispiele sei der schnell erstellte *Papierprototyp* für frühe Analysen und der aufwändige *elektronische Prototyp*, beispielsweise im *rapid prototyping* für detaillierte Diskussionen, zu nennen. Zum Ermitteln von Anforderungen schlägt Rupp zu Beginn einen stakeholderorientierten Ansatz vor. Nach der Auseinandersetzung mit den Stakeholdern wird auf die Klassifikation von Zielen eingegangen. Diese lassen sich anhand der Kriterien [Rup07]

- Reichweite (strategisch/operativ),
- Bedeutung (Hauptziel/Nebenziel),
- Zeithorizont (lang-/kurzfristig),
- Inhalt (Branche, Vertriebsziele, ...),
- Messbarkeit (qualitativ/quantitativ),
- Verantwortungsbereich,
- Grad der juristischen Verbindlichkeit

einordnen. Für die Ermittlung von Kundenwünschen werden vielfältige Techniken vorgestellt und anhand von Rahmenbedingungen bewertet. Diese sind im Wesentlichen in die Bereiche *Kreativität*, *Beobachtung*, *Befragung* und *Vergangenheitsorientiert* unterteilt. Von der Ermittlung der Kundenwünsche bis zur Erhebung von Anforderungen wird

das *SOPHIST-Regelwerk* vorgeschlagen. Es handelt sich um eine Vereinigung von verschiedenen Methoden aus den Disziplinen Linguistik, Informatik, Psychologie und Psychotherapie [Rup07]. Es handelt sich um einen Ansatz,

"die gesamte Komplexität eines Systems durch einen vollständigen Satz von qualitativ hochwertigen Anforderungen zu finden" [Rup07, S. 175].

Für die Dokumentation der Anforderungen unterschiedlicher Art werden verschiedene Techniken vorgeschlagen, beispielsweise Modelle aus der UML, wie das Zustandsdiagramm. Für die Formulierung von Anforderungen wird ein schablonenorientierter Ansatz vorgeschlagen. Im Kontext der Anforderungsbewertung werden u.a. die klaren Regeln für die Formulierung von Anforderungen aufgegriffen. Diese Regeln sollen die Eigenschaften, wie Eindeutigkeit, Klassifizierbarkeit, Lesbarkeit, etc. berücksichtigen. Daran anschließend werden Maße für die Qualität von Anforderungen bereitgestellt [SSB10]. Die Eindeutigkeit wird angesehen als die relative Anzahl der Anforderungen ohne Regelverletzung in Bezug auf Eindeutigkeit. Es soll gewährleistet werden, dass jede Person das gleiche unter den Anforderungen versteht. Eine gute Anforderungsdokumentation enthalte keine Passivform und Sätze sollten aktiv formuliert werden, daher wird ein Ausschluss der Passivform gefordert. Eine weitere wichtige Eigenschaft ist die Klassifizierbarkeit. Diese ist definiert als der relative Anteil der Anforderungen mit erkennbaren Schlüsselworten an den Gesamtanforderungen. Hier gilt die Annahme, dass Schlüsselworte vorher definiert sind und eine Zuordnung zulassen. Identifizierbarkeit wird z.B. durch eine eindeutige Nummerierung der Anforderungen erreicht. Lesbarkeit ist eine Eigenschaft, die ein Anforderungsdokument besitzt, in dem möglichst wenige Querverweise enthalten sind, sodass jede Anforderung für sich verstanden werden kann. Eine einfach nachvollziehbare Eigenschaft ist die Selektierbarkeit. Es wird ein Attribut benötigt, anhand dessen die Anforderungen ausgesucht werden können. Beispiele sind das Datum, die Personen oder die Priorität [SSB10].

Weitere Ansätze aus dem Requirements Engineering Die große Vielzahl von Ansätzen lässt sich in dieser Arbeit allein aus Gründen der Übersichtlichkeit nicht vollständig darstellen. Es sei jedoch angemerkt, dass auf dem Gebiet innovative Forschung betrieben wird. Dabei sollten u.a. und exemplarisch die Arbeiten von Cheng et al. [CLG⁺09] oder Finkelstein et al. [SBW⁺10] sowie vielen weiteren, wie Robertson und Robertson [RR13] sowie Hood et al. [HFP08] nicht unerwähnt bleiben. Aus der Perspektive dieser Arbeit liegt deren Fokus auf der softwareorientierten Sicht.

3.4.2 Verfahren zur Bewertung von Anforderungen

Neben den bereits in den Frameworks enthaltenen Bewertungsverfahren, werden im Zuge dieses Abschnitts einige Vorgehensweisen zum Thema Anforderungsbehandlung im softwaregeprägten Requirements Engineering vorgestellt.

Ein Pionier auf dem Gebiet der Bewertung von Anforderungen ist GILB [SSB10]. Er schlägt eine Unterteilung in funktionale und nicht-funktionale Maße vor. Dabei sind *funktionale* Maße etwa Maße für den Umfang und die Komplexität. Die Anzahl der Knoten in einem hierarchischen Anforderungsbaum, die Ein- und Ausgabeobjekte oder die

Anzahl definierter Zustände gelten als Maß für den Umfang. Die Komplexität lässt sich z.B. anhand der Breite und Tiefe der Funktionsbäume ermitteln. Die Qualität ist ein Maß der Kategorie *nicht-funktional*. Diese kann beispielsweise anhand der Portierbarkeit, der Wartbarkeit oder der Performanz ermittelt werden [GB06].

Boehm [BBH94] stellt vier Kriterien für Anforderungen vor. Die *Vollständigkeit* wird anhand der Anzahl offen gelassener Anforderungen, fehlender Pflichtangaben oder fehlender Funktionen bestimmt. *Konsistenz* lässt sich über die Anzahl der internen, sich widersprechenden Aussagen und der Anzahl der externen Dokumenten widersprechenden Aussagen messen. Die Anzahl an Anforderungen, welche nur mit sehr hohen Kosten realisierbar sind, ist ein Maß für die *Machbarkeit*. Die Anzahl an testbaren Anforderungen im Vergleich zu nicht-testbaren Anforderungen und das Verhältnis von quantifizierbaren zu nicht quantifizierbaren Anforderungen ist ein Maß für die *Testbarkeit* [SSB10].

Parnas und Weiss schlagen eine Anforderungsüberprüfung anhand von Reviews nach einem bestimmten Muster vor [PW85]. Dabei werden unterschiedliche Sichtweisen eingenommen – Versäumnis und Kommision⁷ *Versäumnisse* werden z.B. an fehlenden Funktionalitäten oder Schnittstellen festgemacht. Uneindeutige Aussagen und falsche Aussagen gehören zur Gruppe *Kommission*. Dieses Vorgehen wird mit Checklisten unterstützt [SSB10].

Fraser und Vaishnavi beschäftigen sich mit formalen Anforderungen [FV97]. Sie schlagen jedoch auch ein pragmatisches Vorgehen vor. Eine Anforderungsspezifikation soll zwei Mal von unterschiedlichen Personen erstellt werden. Anschließend werden die Ergebnisse verglichen. Erst ab einer Übereinstimmung von 80 Prozent wird die Spezifikation freigegeben und die Entwicklung kann beginnen [SSB10].

HAYES stellt die *Traceability* als wesentliches Kriterium zur Bewertung von Anforderungen dar [HDS+07]. Es wird eine textbasierte Suche nach Wortpaaren vorgeschlagen. Dabei können Subjekt, Prädikat und Objekt jeweils vom Tester kombiniert werden. Die Trefferrate wird als

getroffene Anforderungen
getroffene + verpasste Anforderungen

definiert. Die Präzision ist durch die

Anzahl der Suchaktionen
Anzahl der Suchaktionen + Anzahl der falschen Ergebnisse

gegeben. Die Verfolgbarkeit wird bei diesem Ansatz als das Produkt aus *Trefferrate* und *Präzision* interpretiert [SSB10].

GLINZ wählt einen betriebswirtschaftlichen Ansatz zur Bewertung der Anforderungen. Es werden harte und weiche Anforderungen unterschieden: Harte haben einen feststehenden Nutzen und weiche steigern ihren Nutzwert mit dem Grad der Erfüllung [Gli05]. So ist der Nutzwert jeder einzelnen Anforderung zu ermitteln, um eine Priorisierung vorzunehmen. Ein Beispiel ist die verlorene Zeit, wenn die Performanz langsamer ist, als in einer Anforderung gefordert. Wird die verlorene Zeit in Geld – einen Wert, der Vergleiche ermöglicht – umgerechnet ergibt sich sukzessive ein Nutzwert für jede Anforderung. Diese können ggf. detaillierter gewichtet werden, um eine Bewertung zu erhalten, wie teuer das Nichterfüllen einer Anforderung ist [SSB10].

⁷aus dem Englischen übersetzt: Original *omission* und *commission*

3.4.3 Entwicklungsmethodiken und -umgebungen

Estefan gibt einen Überblick über einige international genutzte modellbasierte Entwicklungsprozesse [Est07]. Diese scheinen wie ein definierter Pfad für die Entwicklung von Anforderungen zu sein. Innerhalb einer Disziplin werden Anforderungen von Experten identifiziert werden können. Die generische Berücksichtigung von mehreren Disziplinen ist jedoch weiterhin problematisch [Bir11]. Die Aufführung der tendenziell generischen Methodiken dient der Vollständigkeit, da in allen eben auch Anforderungen eine gewisse Rolle spielen. Dabei ist es nicht zwangsläufig Anspruch der Methodiken, primär Anforderungen zu bedienen. Dies sollte allerdings auch bei Methodiken zu Anforderungen selbst nicht im Vordergrund stehen, da diese im Kontext dieser Arbeit stets der erfolgreichen Entwicklung von Systemen dienen und keinen Selbstzweck darstellen. Es gibt sicher zahlreiche weitere Methodiken als die hier untersuchten. So beschreiben beispielsweise HABERFELLNER et al. in ihrem grundlegenden Ansatz zum Systems Engineering zwar Situationsanalysen, Problembetrachtungen und Zielformulierung, bleiben jedoch sehr abstrakt und gehen nicht detaillierter auf technische Anforderungen ein [HWF+12]. Eine nähere Analyse aller Methodiken entspricht auch daher keiner ergebnisorientierten Vorgehensweise. Vielmehr sollten die hier ermittelten Erkenntnisse extrapoliert werden können und somit repräsentativ für weitere Methodiken gelten.

SysML

Die explizite Aufführung von SysML dient der Veranschaulichung der Werkzeuge, die für Anforderungen von Bedeutung sind. Es bauen mehrere Methodiken auf die SysML auf. Zuerst sollte festgehalten werden, dass SysML eine Sprache ist. Mit dieser Sprache lassen sich Systeme beschreiben. Die verschiedenen Diagramme sind in Bild 3-4 dargestellt. Von besonderer Wichtigkeit ist hier das Anforderungsdiagramm. Die Definition von



Bild 3-4: Aufbau der Sprache SysML, abgewandelt nach [Wei14]

Anforderungen wird von Weilkins wie folgt vorgenommen:

"Die Anforderung (engl. requirement) beschreibt eine oder mehrere Eigenschaften oder Verhaltensweisen eines Systems, die stets erfüllt sein müssen" [Wei14].

Der vertragliche Charakter wird in den Vordergrund gestellt. Die Verbindung funktionaler Anforderungen mit weiteren Elementen, wie z.B. *Anwendungsfällen* wird über Schnittstellen dokumentiert, nicht jedoch in den Anforderungen selbst. Darüber hinaus ermöglicht die SysML Notation, dass die Anforderungen weitere Beziehungen mit anderen Elementen haben können. Diese Beziehungen und wichtigsten Elemente sind [Wei14]:

- Ableitungsbeziehung: Anforderungen können sich aus anderen Anforderungen ableiten und solche Beziehungen können mit *deriveReqt* deutlich gemacht werden.
- Enthältbeziehung: In SysML können Anforderungen andere Anforderungen enthalten.
- Erfüllungsbeziehung: Elemente, die eine Anforderung erfüllen, können mittels *satisfy* mit der entsprechenden Anforderung verknüpft werden.
- Kopiebeziehung: Da SysML Verbindungen einer Anforderung zu mehr als einer anderen Anforderung unterbindet, ist im Sinne der Wiederverwendbarkeit *copy* eine nützliche Beziehung.
- Prüfbeziehung: Die Prüfung der Umsetzung einer Anforderung durch ein anderes Element wird mit der Beziehung *verify* deutlich gemacht.
- Verfeinerungsbeziehung: Ein Element kann eine Anforderung detaillieren. Diese Beziehung wird mit *refine* dargestellt.
- Verfolgungsbeziehung: Das Thema Verfolgbarkeit wird in SysML mittels *trace* gelöst. Diese Verbindungen können sehr allgemein genutzt werden, sodass Kombinationen zu beliebigen Elementen dargestellt werden können.
- Testfall: Der *testCase* ist keine Verbindung, sondern ist ein konkreter Prüfablauf. Üblicherweise sollte die Beziehung *refine* genutzt werden, um Testfälle mit Anforderungen zu verknüpfen.
- Tabellennotation: In einigen Softwarewerkzeugen lassen sich die Anforderungen als Tabelle exportieren, sodass sie in traditionellem Format vorliegen.

Diese weitreichenden und formalen Dokumentationsmöglichkeiten werden ergänzt durch Export- und Importfunktionalitäten bis hin zu der Möglichkeit von ausführbaren Modellen. Diese Simulationsfähigkeit ist jedoch nicht der primäre Zweck und sehr aufwändig [Wei14]. Entsprechende Werkzeugketten können toolspezifische Simulationsmodelle mit dem SysML Modell koppeln (vgl. Abschnitt 3.4.6).

Entwicklungsmethodik nach Ehrlenspiel

Ehrlenspiel liefert in seinem Buch *Integrierte Produktentwicklung* [Ehr09] ein detailliertes Vorgehen zur Identifikation von Anforderungen. Der Problemlösungszyklus unterteilt die drei Arbeitsabschnitte *Zielsuche*, *Lösungssuche* und *Lösungsauswahl*. In der Phase der Zielsuche gilt es, die Aufgabe zu analysieren, zu formulieren und zu strukturieren. Anforderungen werden im Wesentlichen als Aufgabenklärung betrachtet. Diese Phase wird jedoch als äußerst bedeutend erachtet.

"In den Anforderungen liegt der Kern des zukünftigen Produkts" [Ehr09, S. 374].

Es wird zwischen Ziel und Anforderung unterschieden. Ziele seien Soll-Vorstellungen des Auftraggebers und Anforderungen Soll-Eigenschaften des Systems. Hier wird ein Prozess aus Fragebögen, Checklisten und Gesprächen vorgeschlagen, um letztlich die Anforderungsliste zu erhalten. Als Quellen für Anforderungen werden verschiedene Stakeholder vorgeschlagen, wie Normen, Marketing und der Kunde. Eine Klassifizierung erfolgt entsprechend der Kategorien:

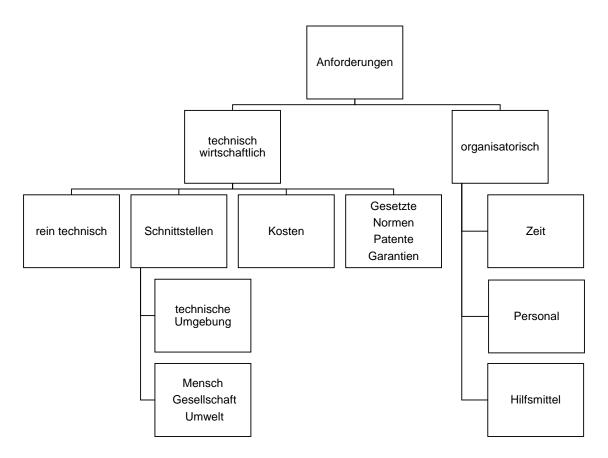


Bild 3-5: Anforderungsarten nach Ehrlenspiel, abgewandelt nach [Ehr09]

- explizit/implizit,
- Forderung/Wunsch.

Es wird u.a. unterschieden zwischen Festforderungen, Intervallforderungen, Maximalforderungen und Minimalforderungen, wobei sich die Attribute auf den entsprechenden Zahlenwert der Eigenschaft der Anforderung beziehen. Ein Beispiel im Bereich der Fahrzeugtechnik ist die Mindestforderung einer zu erreichenden Geschwindigkeit von mindestens 100 km/h. Eine kurze Thematisierung möglicher Zielkonflikte von Anforderungen wird dargestellt, jedoch nicht weiter detailliert. Auf Pahl/Bertz [PBFG13] verweisend wird eine Leitlinie mit Hauptmerkmalen zum Aufstellen von Anforderungslisten bereitgestellt. Am Beispiel des Hauptmerkmals *Geometrie* werden u.a. *Gröβe, Durchmesser, etc.* angegeben. Verschiedene Anforderungsarten werden identifiziert und in einer Darstellung festgehalten (vgl. Bild 3-5).

Automotive SPICE

Als Beispiel eines stark aus der Informatik geprägten Prozesses wird an dieser Stelle das Reifegradmodell *SPICE* der *ISO/IEC 15504* beschrieben [ISO12]. Dies wird in der Ausprägung *Automotive SPICE* getan, einer Entwicklung ab dem Jahr 2001 (initiiert durch die Herstellerinitiative Software) speziell für die Automobilindustrie zur Einordnung der Leistungsfähigkeit der Entwicklungsprozesse von Steuergerät-Lieferanten. Ein

Reifegradmodell dient der Bewertung der Prozessfähigkeit, gibt jedoch auch eine Art Sollzustand vor. Der Wandel der Denkweise zu disziplinübergreifenden Ansätzen wird am Beispiel *Automotive SPICE* sehr gut deutlich. Waren vor einigen Jahren noch quasi alle Domänen in einem Prozess zusammengeführt, so wurde 2015 mit der Version *SPICE* 3.0 der Schritt zu domänenübergreifenderen Ansätzen getan. Im Folgenden wird der alte Entwicklungsprozess beschrieben, die Neuerungen werden in Kapitel 4 aufgegriffen. *Automotive SPICE* hat einen starken Fokus auf Prozesse [MHDZ07]. In der Automobilindustrie wird ein Einsatz des Prozesses oftmals von Lieferanten verlangt, um eine gewisse Qualität in den Prozessen zu gewährleisten. Grundsätzlich besteht SPICE aus zwei Komponenten.

- Prozessreferenzmodell PRM,
- Prozessassessmentmodell PAM.

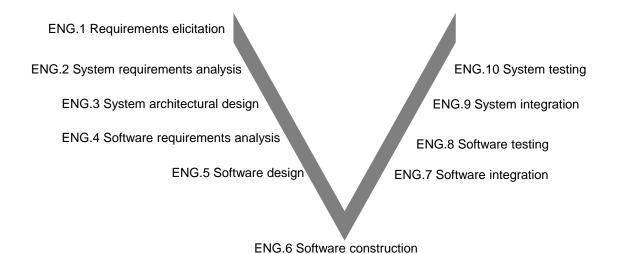


Bild 3-6: Einordnung der SPICE Einzelprozesse in ein einfaches V-Modell

Die Betrachtung des PAM ist für den Praxiseinsatz ausreichend [MHDZ07]. Dieses hat zwei Dimensionen – die Prozessdimension und die Reifegraddimension. Der Reifegrad wird auf einer Skala von θ (unvollständig) bis θ (optimierend) beurteilt. Die Prozessdimension ist im Kontext dieser Arbeit von größerer Bedeutung, sie enthält neben Indikatoren ein Prozessergebnis und den Zweck des Prozesses. Der Engineeringprozess besteht aus 10 Schritten [MHDZ07]:

- ENG.1 Requirements elicitation,
- ENG.2 System requirements analysis,
- ENG.3 System architectural design,
- ENG.4 Software requirements analysis,
- ENG.5 Software design,
- ENG.6 Software construction,
- ENG.7 Software integration,
- ENG.8 Software testing,

- ENG.9 System integration,
- ENG.10 System testing.

Diese können vereinfacht in ein V-Modell eingeordnet werden (vgl. Bild 3-6). Der Bezug der Prozessschritte ENG.1, ENG.2 und ENG.4 zu den Anforderungen wird sofort ersichtlich. Der Zweck dieser Prozessschritte ist die Erfassung, Verarbeitung und Verfolgung von Kundenanforderungen. Diese beziehen sich auf den gesamten Produktlebenszyklus und dienen der Definition der benötigten Arbeitsprodukte. Weiterhin werden Anforderungen auf Systemebene und auf Softwareebene analysiert [MHDZ07].

NASA Systems Engineering

Einen bedeutenden Beitrag zum Thema Systems Engineering liefert zweifelsohne die NASA. Die Raumfahrttechnik mit ihren komplexen Systemen ist ein Vorzeigebeispiel für ein geeignetes Gebiet, in welchem Systems Engineering betrieben werden sollte. Das NASA Systems Engineering Handbook beinhaltet wertvolle Darlegungen und Methoden [Nat07]. Eine besondere Errungenschaft in diesem Kontext ist die Requirements Specification Language [FWFC92]. Diese formale Sprache nach strikten Prinzipien erlaubt eine Spezifikation von Anforderungen. Zusammenhänge und Zugehörigkeiten können systematisch ausgedrückt werden und bilden somit eine Basis für die spätere Analyse. Im weiteren Verlauf werden die unterschiedlichen Methoden analysiert. Zur Definition und Handhabung technischer Anforderungen werden mehrere, sich ergänzende Verfahren vorgeschlagen. Die Spezifikation wird dabei als iterativer Prozess verstanden, an deren Ende die technischen Anforderungen stehen. Es wird zwischen Stakeholdererwartungen, Problemdefinition und technischen Anforderungen unterschieden. Technische Anforderungen beinhalten funktionale Anforderungen, Leistungsanforderungen und Schnittstellenanforderungen. Diese technischen Anforderungen stehen dabei auf der gleichen Ebene, wie bspw. Sicherheitsanforderungen und Verlässlichkeitsanforderungen.

Die Dekomposition von abstrakten zu detaillierteren Anforderungen findet systematisch statt. Dabei wird immer ein Abgleich gegenüber abstrakteren Anforderungen vorgenommen, um den Prozess zu validieren. Es wird explizit gefordert, dass die Anforderungen verifiziert werden müssen. Darüber hinaus werden gewisse Eigenschaften verlangt, die eine Anforderung besitzen muss. Diese sind einerseits inhaltlicher Natur und andererseits formeller. So werden Metadaten, wie eine ID oder eine Begründung zu jeder Anforderung verlangt. Eine allgemeine Übersicht des Prozesses zur Spezifikation findet sich in Bild 3-7. Dabei werden Anforderungen auf Kunden- und Organisationsebene unterschieden. Grundsätzlich werden Missionsanforderungen (Mission Requirements) in funktionale Anforderungen (System Functional Requirements) übersetzt, um anschließend in Leistungsanforderungen (System Performance Requirements) überführt zu werden. Diese werden zu den Subsystemen hin weiter detailliert. Der Ansatz bedient sich der Hierarchisierung nach Systembestandteilen. Die Handhabung von Anforderungen geschieht ebenfalls hierarchisch. Besonders für die *Traceability*, die *Validierung* und das Änderungsmanagement werden Vorgehen vorgeschlagen.

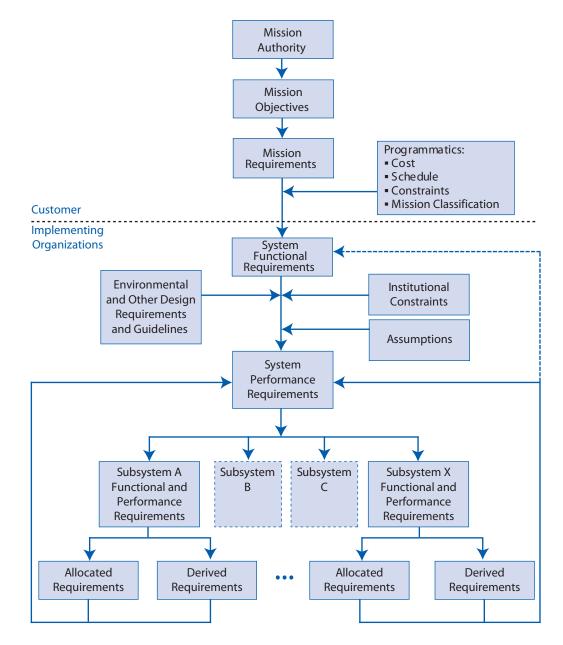


Bild 3-7: Flussdarstellung von Anforderungen im NASA Systems Engineering Handbook [Nat07]

3.4.4 Normen und Standards

"Die existierenden Normen unterstützen die Anwendung von Systems Engineering in der Breite nicht ausreichend genug. Das Zusammenwirken der Normen ist nur schwer zu überschauen und hilft kaum, dem Anwender Systems Engineering näher zu bringen" [GCW+13, S. 25].

Dieses Zitat beschreibt die Situation im Systems Engineering. Das gleiche gilt sicherlich für Normen im Bereich der Anforderungen. Letztendlich ist dies keine sehr gewagte These, da die Mehrheit der Systems Engineering Normen auch Anforderungen berücksichtigt. Eine kurze Analyse einiger bedeutender Normen für Anforderungen folgt dennoch. Eine ausführliche Analyse in einem anderen Kontext kann in der Literatur gefunden werden [SB13].

ISO/IEC 15288

Die Norm ist eine relevante Norm im Kontext *Systems Engineering* und behandelt Systeme allgemein. Anforderungen werden im Bereich technischer Prozesse gesehen (vgl. Bild 3-8) [ISO08]. Der *Stakeholder Requirements Definition Process* dient dem Zweck, die Anforderungen an das System zu definieren. Dafür werden Teilprozesse vorgeschlagen:

- Die Stakeholderanforderungen werden erhoben,
- die Stakeholderanforderungen werden definiert,
- die Stakeholderanforderungen werden analysiert.

Anschließend wird der *Requirements Analysis Process* gestartet, um die Stakeholderanforderungen in Systemanforderungen zu überführen. Dabei werden die Angaben sehr allgemein gehalten, um für möglichst viele Systementwicklungen zu gelten.

VDI 2206

Die Richtlinie Entwicklungsmethodik für mechatronische Systeme ist analog zur allgemeingültigen VDI 2221 zu verstehen. Sie fokussiert dabei speziell mechatronische Systeme [VDI04a]. Es werden systematisch mechatronische Systeme erläutert und Wege aufgezeigt, die Komplexität zu meistern. Das Vorgehensmodell beruht dabei auf dem V-Modell (vgl. Bild 2-9). Ausgangspunkt ist ein konkreter Entwicklungsauftrag. Die Aufgabenstellung soll an dieser Stelle, also am Anfang, präzisiert und ausgearbeitet sein. Dann folgen eine domänenübergreifende Entwicklung, in der das System entworfen wird. Anschließend wird domänenspezifisch das gemeinsam entwickelte Konzept konkretisiert. Letztlich werden die Ergebnisse zu einem Gesamtsystem integriert. Dieser Zyklus kann dabei mehrfach durchlaufen werden. Es werden für jede Phase des V-Modells unterstützende Methoden empfohlen, wie z.B. der modellbasierte Entwurf oder HiL-Tests. Insgesamt ist die Richtlinie für die Entwicklung mechatronischer Systeme sehr gut geeignet, behandelt jedoch das Thema Anforderungen fast nicht.

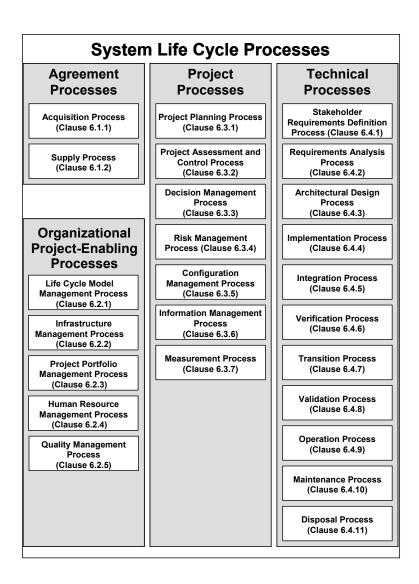


Bild 3-8: Die Prozesse für ein System im Lebenszyklus [ISO08]

VDI 2221

Die Richtlinie aus dem Jahre 1993 beschreibt eine *Methodik zum Entwicklen und Konstruieren technischer Systeme und Produkte* [VDI93]. Anforderungen werden im ersten Schritt gesehen und mit der Anforderungsliste abgeschlossen (vgl. Bild 3-9). Anschließend wird die Ermittlung von Funktionen vorgeschlagen, zu denen Lösungsprinzipien gesucht werden. Nach weiteren Schritten wird ein Gesamtentwurf erreicht, welcher mit einer Produktdokumentation abschließt. Es wird gefordert, dass die Anforderungen im gesamten Prozess erfüllt und angepasst werden. Dabei sind iterative Sprünge zu vorherigen Arbeitsschritten ggf. erforderlich. Die Berücksichtigung der Anforderungen wird mit vorgeschlagenen Methoden, wie

- Wettbewerberanalyse,
- Wertanalyse,
- Systemtechnik

unterstützt.

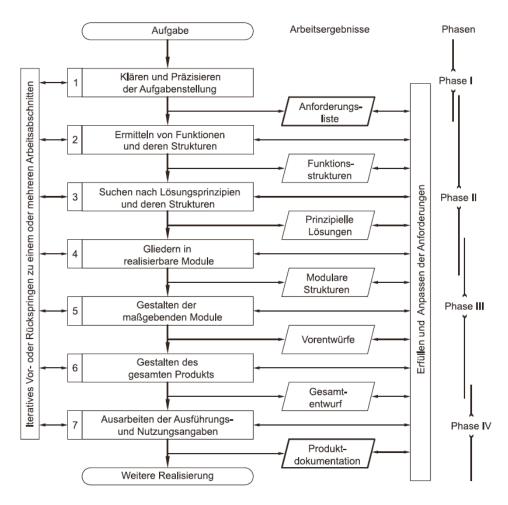


Bild 3-9: Methodisches Entwickeln nach der VDI Richtlinie 2221 [VDI93]

VDI 2223

Die Richtlinie gliedert sich in die VDI 2221 ein und spezifiziert den Prozess des Entwerfens [VDI04b]. Dabei fokussiert sie die Phase drei (vgl. Bild 3-9). Unter Schritt vier, Gliedern in realisierbare Module, wird der erste Schritt als Erkennen gestaltungsbeeinflussender Anforderungen und Bedingungen definiert. Eine Priorisierung der Anforderungen wird gefordert. Darüber hinaus sollen die allgemeinen Anforderungen in spezielle Anforderungen überführt werden. Dafür werden auch erste Berechnungen vorgeschlagen. Weiterhin werden dominierende Anforderungen thematisiert, welche mit dem Gestalten von außen nach innen gehandhabt werden. Zur Bewertung von Vorentwürfen wird explizit gefordert, in regelmäßigen Abständen, die Zwischenlösungen anhand der Anforderungsliste zu bewerten. Dies kann mithilfe von

- Funktionsberechungen,
- Nachweis der Funktionsfähigkeit,
- Dimensionsberechnungen,
- Kostenberechnungen,
- Störgrößenabschätzungen,
- Vorversuchen

geschehen. Die Richtlinie geht dementsprechend auf die Berücksichtigung von Anforderungen ein, konzentriert sich jedoch eher auf die Tätigkeiten des Entwurfes und konkretisiert die Eigenschaften der Anforderungen nicht ausreichend.

ISO 25030

Die Norm aus dem Software Engineering ist in die *ISO 25000* Familie eingeordnet und fordert ein bestimmtes Schema, um Anforderungen zu dokumentieren [ISO11b]. Dabei werden vier Elemente vorgeschlagen:

- Qualitätsmerkmal,
- Messeinheit,
- Messmethode,
- Zielwert.

Dieser Vorschlag eignet sich sehr gut, um nicht-funktionale Anforderungen zu beschreiben. Die entsprechende Nutzung für funktionale Anforderungen ist eine Herausforderung, da diese erst quantifiziert werden müssen [SSB10].

IEEE 29148

Der Standard aus dem Jahr 2011 zum Systems und Software Engineering und Requirements Engineering spezifiziert die Prozesse zur Entwicklung von Anforderungen für Systeme und Softwareprodukte [ISO11a]. Besonders aussagekräftig in dem betrachteten Kontext sind die geforderten Eigenschaften von Anforderungen:

- notwendig,
- lösungsneutral,
- eindeutig,
- konsistent,
- vollständig,
- einzeln,
- möglich,
- verfolgbar,
- verifizierbar.

Es werden grundsätzliche Hinweise zum Umgang und zur Verwendung von Anforderungen vorgeschlagen. Dabei werden die Prozesse in Bild 3-10 übersichtlich dargestellt. Die Norm gibt eine gute Anleitung, um grundsätzlich mit Anforderungen umzugehen. Dabei bleibt sie jedoch eher generisch.

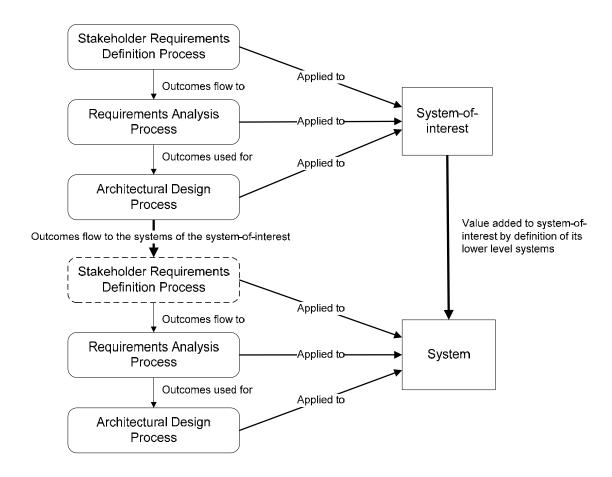


Bild 3-10: Anwendung der Prozesse der IEEE Norm 29148 [IEE11]

3.4.5 Anforderungsorientierte Methoden

Quality Function Deployment nach Akao Im QFD wird das sogenannte House of Quality genutzt, um Kundenanforderungen mit technischen Konstruktionsmerkmalen zu verbinden [Aka17]. Stark vereinfacht gesprochen, soll mit dem HoQ das Lastenheft in das Pflichtenheft überführt werden. Der Aufbau ist in Bild 3-11 zu sehen. Auf der linken Seite werden die Kundenwünsche aufgelistet und entsprechend ihrer Bedeutung gewichtet und auf der rechten Seite in Bezug auf den Wettbewerb bewertet. In dem oberen Bereich werden die technischen Konstruktionsmerkmale einbezogen, deren Beziehung untereinander im sogenannten Dach des Hauses berücksichtigt werden. Ein einfaches Beispiel soll dies verdeutlichen. Es kann angenommen werden, es gäbe zwei Kundenwünsche. Der erste Kundenwunsch lautet schnelles Fahrzeug und der zweite Kundenwunsch wenig Verbrauch. Zugehörige technische Konstruktionsmerkmale sind in diesem Fall u.a. Hubraum und Gesamtmasse. Diese beiden Merkmale sind gegenläufig. Diese Beziehung kann im Dach des Hauses dargestellt werden. Letztlich sollten alle gewünschten Kundenmerkmale in einer Beziehung mit mindestens einem technischen Konstruktionsmerkmal stehen. Der Keller des Hauses wird genutzt, um den technischen Konstruktionsmerkmalen konkrete Werte zuzuweisen.

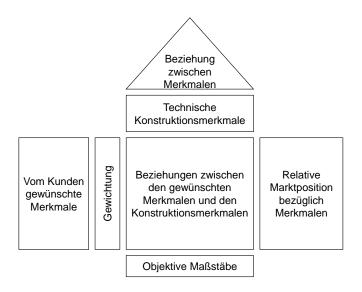


Bild 3-11: Aufbau des House of Quality, abgewandelt nach [HV03]

Szenariobasierte Spezifikation Plock und Zuck schlagen vor, sogenannte *Live Sequence Charts* zu nutzen, um von Anforderungen zu Spezifikationen zu gelangen [PZ05]. Diese bilden einen visuellen Formalismus für eine szenariobasierte Spezifikation. Es handelt sich um einen Ansatz aus den Computerwissenschaften, welcher für diskrete Betrachtungen im Sinne von Automaten (Timed Automata) formal abläuft, kontinuierliche Prozesse jedoch vernachlässigt.

3.4.6 Toolunterstützung

Beispielhaft sei an dieser Stelle ein Überblick über verschiedene Tools gegeben, welche Anforderungen berücksichtigen.

Dassault 3D Experience Der RFLP ((Requirements - Functional - Logical - Physical) Ansatz von Dassault ist die tooltechnische Umsetzung der Verbindung der Anforderungen mit Modellen. Es werden Teile aus SysML umgesetzt und mit konkreten Entwicklungsmodellen verbunden. Ein Element aus der Plattform ist Dymola. Hier gibt es ebenfalls die Möglichkeit, im Tool selbst formale Anforderungsanalysen mittels der Dymola Requirements Formalization durchzuführen [OTB+15].

Doors Doors setzt den Fokus auf die Erfassung und Verwaltung von Anforderungen und bietet vielfältige Funktionalitäten. Dabei kann auf verschiedene Standards und Prozessmodelle, wie z.B. SPICE, aufgebaut werden. Anforderungen werden in einer hierarchischen Dokumentstruktur hinterlegt [Ebe10]. Doors erlaubt die Verlinkung unterschiedlicher Anforderungsdokumente, auch mit Testfällen. Die Analyse der Zusammenhänge wird über Funktionalitäten wie die *Traceability View* ermöglicht [Ebe10]. Von der Modellierung bis zum Test sind Verknüpfungen möglich, sodass der Grad der Realisierung sehr transparent ist. Dies ist eine wertvolle Funktion, gerade in Bezug auf Aufwandsabschätzung bei Änderungswünschen von Anforderungen.

Matlab Requirements Die Erweiterung der Matlabumgebung dient zum einen der Verbindung von Modellelementen mit dem Anforderungsdokument und zum anderen der formalen Überprüfung von Spezifikationen. Dabei wird auf Methoden der Untersuchung diskreter Zustandsautomaten zurückgegriffen.

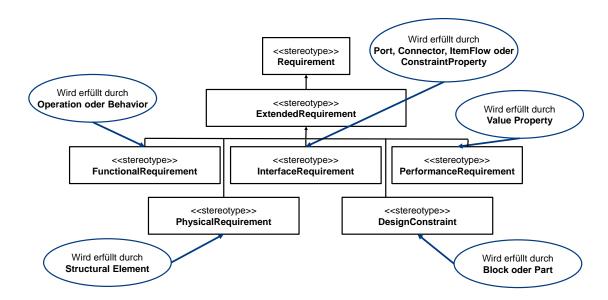


Bild 3-12: Requirements Diagram, abgewandelt nach [Wei14]

SysML Requirements Diagram Als SysML-Umgebungen dienen diverse Tools der ganzheitlichen Systembetrachtung mithilfe von Systemmodellen. Die jeweiligen Funktionalitäten gehen teilweise über klassische SysML hinaus. Das *Requirements Diagram* (vgl. Bild 3-12) ist die typische Darstellung von Anforderungen in der SysML, welches jedoch weiter verknüpft sein kann. Deutlich wird der Sinn der Systemmodelle als übergeordnete Sichtweise anhand von Bild 3-13. Die Artefakte der weiterführenden Entwicklungsschritte können mit dem Systemmodell verbunden werden, was eine deutliche Erhöhung der Durchgängigkeit verspricht. Die sehr generisch veranlagten Tools müssen jedoch spezifisch in einen konkreten Entwicklungsfall eingebunden werden, da ihre Möglichkeiten weitreichend sind.



Bild 3-13: Toolanbindung in Cameo [No 18]

Weitere Tools Die vorgestellten Tools sind lediglich ein repräsentativer Ausschnitt aus der Vielzahl an Softwarewerkzeugen, welche sich mit Anforderungen beschäftigen. Ein weit verbreitetes weiteres Tool ist z.B. *PTC Integrity* [PTC17]. Das Werkzeug *Stimulus* verspricht durch seinen modellbasierten Ansatz unter Berücksichtigung der praxisorientierten Testfälle vielfältige Möglichkeiten [Sti17]. Auch neue Toolboxen in den Simulationstools, wie z.B. das Testing Tool in Dymola, zeigen, dass sich aktuell die Softwarebranche bewegt.

3.5 Handlungsbedarf

Handlungsbedarf entsteht durch die Kombination der gestellten Anforderungen an diese Arbeit und die nicht-Erfüllung dieser durch die analysierten Ansätze. Darüber hinaus wird dieser Bedarf zunehmend dringend, da die Weiterentwicklung hin zu komplexeren Systemen zunimmt. Durch die Analyse des Stands der Wissenschaft kann gezeigt werden, dass manche der beschriebenen Probleme noch von keinem Lösungsansatz gelöst wurden, jedoch ein kollektives Problembewusstsein in manchen Wissenschaftszweigen herrscht.

3.5.1 Bewertung der untersuchten Ansätze

Requirements Engineering Frameworks stellen formale Methoden bereit, die jedoch nicht ohne Weiteres für kontinuierliche Systeme angewandt werden können. Sie bieten dabei eine sehr systematische Vorgehensweise. Ebert bewertet die Vollständigkeit anhand von Anwendungsszenarien ohne eine Gewichtung vorzunehmen. Der große Vorteil des Ansatzes ist die starke Praxisnähe. Dies führt jedoch auch zu einer Verringerung der detaillierten Betrachtungen. Quantifizierbare Merkmale werden bereitgestellt und dienen der Priorisierung von Anforderungen. Der Ansatz bezieht sich jedoch auf den frühen Teil der Systementwicklung und nimmt eine stark softwarelastige Sicht ein. Die trotz oder wegen des praxisnahen Ansatzes vorliegende Vernachlässigung kontinuierlicher Phänomene bremst die Konsistenz im Entwurf. Hull et al. fordern eine ganzheitliche Berücksichtigung aller Schritte der Entwicklung bis hin zum Test. Trotz des ganzheitlichen Ansatzes liegt der Fokus eindeutig auf den frühen Entwicklungsphasen, was auch im Umfang deutlich wird. Auch dieser Ansatz nimmt eine softwareorientierte Perspektive ein. In den drei Auflagen des Buches kann jedoch trotz gleich bleibenden Inhaltsverzeichnisses ein stetiger Prozess hin zu ganzheitlicheren Ansätzen erkannt werden [HJD11] [HJD05] [HJD02]. Die Anlehnung an Volere⁸ zeigt jedoch die praxisnahe Herangehensweise. Insgesamt handelt es sich um einen geeigneten Ansatz der Detaillierung von Anforderungen in Multidomänenentwicklungen, jedoch ist die vorgeschlagene Detaillierung nicht ausreichend. In dem Ansatz von Partsch werden explizit die frühen Phasen des Entwurfs behandelt. Die Ganzheitlichkeit wird zwar gefordert, eine konkrete Lösung wird nicht beschrieben. Der Fokus auf softwareintensive Systeme wird deutlich. Ein Vorgehen und Priorisierungsmöglichkeiten in den Arbeiten von Pohl zeigen den systematischen Ansatz. Die Konzentration auf Software wird anhand der Design Layer deutlich. Zwar wird die Notwendigkeit von verschiedenen Anforderungsebenen erkannt, jedoch entwickeln sich diese Ebenen von der Systemebene ausgehend hin zur Software. Im Ansatz von Rupp steht das Anforderungsdokument im Vordergrund. Der Ansatz ist zwar sehr systematisch und ganzheitlich gedacht – doch detaillierte technische Anforderungen werden nicht betrachtet. Das Anforderungsdokument wird allerdings ganzheitlich beschrieben, von unternehmensstrategischen Zielen bis hin zu juristischen Fragestellungen. Diese Fokussierung auf die Ermittlung und Dokumentation in frühen Phasen zeichnet den Ansatz aus. Die Softwareorientierung schlägt nicht so stark zu Buche, sodass sich dieser Ansatz zur Ermittlung und Dokumentation anwenden lässt. Die Verfahren zur Bewertung von Anforderungen wer-

⁸Volere (italienisches Verb für wollen) beinhaltet Unterstützung für Anforderungen u.a. in Form von Templates [RR13].

den eingesetzt, um sowohl Anforderungen als auch das System an sich zu quantifizieren. Dadurch wird gemessen, ob Anforderungen gewisse Kriterien erfüllen, bzw. gewisse Eigenschaften besitzen. Dabei bleiben diese Bewertungen sozusagen auf Metaebene, ohne den Inhalt der Anforderungen explizit zu berücksichtigen. Das Ziel dieser Ansätze liegt nicht in der ganzheitlichen Betrachtung. Die Kategorie der Entwicklungsmethodiken und -umgebungen beschränkt sich auf sehr generische Ansätze. Einerseits ermöglicht dies die bedarfsgerechte Einbindung konkreter Werkzeuge, andererseits werden diese nicht bereitgestellt. SysML stellt nützliche Tools im Sinne der Durchgängigkeit bereit, hat jedoch nicht den Anspruch einen Prozess mitzuliefern. Selbstverständlich gibt es Ansätze, die die SysML genau dafür nutzen. Diese konzentrieren sich jedoch nicht auf Anforderungen und detaillieren diese nicht hinreichend. Dennoch werden Werkzeuge von der Erfassung bis zur Dokumentation bereitgestellt. Entwicklungsmethodiken, wie in den Beispielen VON PAHL/BEITZ und EHRLENSPIEL, erkennen die Bedeutung von Anforderungen und deren Diversität. Auch eine Unterscheidung zwischen Zielen und Anforderungen wird erkannt, jedoch nicht konsequent beibehalten. Im Sinne der Konsistenz müsste sich beispielsweise die Bezeichnung der Quellen von Anforderungen auf Quellen für Ziele erweitern. Die entsprechende Detaillierung wird nicht explizit behandelt, da die Entwicklungsmethoden nicht darauf abzielen. Tendentiell weit in der Berücksichtigung von Anforderungen ist das Reifegradmodell SPICE. Anforderungen werden auf Systemebene und auf Softwareebene gesehen. Doch genau an dieser Stelle wird die Ganzheitlichkeit des Ansatzes verletzt, denn eine Aufteilung sollte bis in alle beteiligten Disziplinen erfolgen. Das Vorgehen der NASA ist dagegen sehr ganzheitlich angelegt und berücksichtigt auch eine Detaillierung von Anforderungen auf Prozessebene. Daher stellt der Ansatz unter den generischen Methodiken den Prozess mit den größten Überdeckungen mit den Zielen dieser Arbeit dar. Wenn es um fachspezifische Werkzeuge und mathematische Interpretierbarkeit geht, sinkt die Leistungsfähigkeit jedoch. Es bleibt festzuhalten, dass elementare Anforderungen an eine Systematik nur in geringem Maße erfüllt sind. Gerade die Verbindung mit der realen Entwicklungsarbeit geht in den Entwicklungsmethodiken insgesamt etwas unter.

Die untersuchten *Normen und Standards* bieten jeweils Prozesse an, um Anforderungen zu berücksichtigen. Sie bleiben sehr allgemein und sehen Anforderungen zum großen Teil vor einem Entwicklungsprozess und nicht begleitend. Eine Durchgängigkeit wird vermisst. Lediglich die *VDI 2221* Normenfamilie behandelt Anforderungen entwicklungsbegleitend. Dafür fehlt dieser Richtlinie die interdisziplinäre Sicht in Bezug auf Anforderungen. Außerdem wird keine quantifizierbare Auswertung bereitgestellt, obwohl eine Priorisierung gefordert wird. Metadaten von Anforderungen, wie übergeordnete Eigenschaften werden vorgeschlagen. Die Normen bleiben trotz ihrer unterschiedlichen Ausrichtungen auf abstrakter Prozessebene und detaillieren die notwendigen Schritte nicht hinreichend, sodass sie eher als generisch anzusehen sind.

Das *QFD* verbindet Kundenanforderungen mit technischen Eigenschaften. Der Ansatz zeigt einen pragmatischen Weg, um abstrakte Anforderungen zu detaillieren. Die Konkretisierung erfolgt dabei, wenn möglich, bis zu technischen Anforderungen mit quantifizierbaren Eigenschaften. Dabei wird auf Zwischenschritte verzichtet und eine weitere Unterteilung von Anforderungen ist nicht gefordert. Der Ansatz ist in frühen Phasen des Entwurfes einzugliedern. Die untersuchten *Tools* stellen jeweils bestimmte Werkzeuge dar, und bleiben an sich nahezu ohne Methodik. Sie stellen somit wertvolle Hilfsmittel im Sinne der Konsistenz und Durchgängigkeit, sind jedoch entweder nicht für kontinuierli-

che Systembetrachtungen ausgelegt oder unterstützen keinen ganzheitlichen Prozess.

Die weiteren untersuchten Methoden erfüllen jeweils nur Teile der gestellten Anforderungen an eine Systematik. Eine triviale Kombination der Ansätze führt nicht zum Ziel, da kein Ansatz vorsieht, kontinuierliche Phänomene zu berücksichtigen. Darüber hinaus sind die sehr generischen Ansätze ungeeignet, auf Anforderungen einzugehen, da dies mit mangelnder Durchgängigkeit einher ginge. Auch die Nutzung von Analysen funktioniert bei den betrachteten Ansätzen eher auf Metaebene. Die Interdisziplinarität ist bei keinem Ansatz explizit realisiert, obwohl sie teilweise gefordert wird. Die untersuchten Ansätze erfüllen die an sie gesetzten, jedoch nicht die an diese Arbeit gesetzten Anforderungen.

3.5.2 Grad der Übereinstimmung

Anhand des Stands der Wissenschaft wird deutlich, dass kein Ansatz, noch eine triviale Kombination, die gestellten Anforderungen vollends erfüllt (vgl. Bild 3-14).

A1 Systematische Synthese von Anforderungen:

In manchen Ansätzen wird auf die Detaillierung von Anforderungen eingegangen. Dies erfüllt die Forderung jedoch nur teilweise, da die Systematik entweder fehlt oder disziplinspezifisch ist.

A2 Mathematisch interpretierbare Analyse von Anforderungen:

Grundsätzlich werden mehrere mathematisch interpretierbare Analysemethoden für Anforderungen bereitgestellt.

A3 Darstellen von Anforderungen und modellbasierte Übertragbarkeit:

Es existieren geeignete Ansätze der Anforderungsdokumentation, die auch für die in dieser Arbeit vorgestellte Systematik verwendet werden können.

A4 Einbindung in einen Entwurfsprozess (V-Modell):

Die kontinuierliche Einbindung in einen Entwurfsprozess ist nicht vollständig erfüllt.

A5 Ganzheitliche Betrachtung des Entwurfsprozesses:

Ganzheitliche Ansätze finden sich vor allem in den Entwurfsmethodiken wieder. Die disziplinspezifischen Ansätze schaffen keine generische Anwendbarkeit.

A6 Durchgängige Berücksichtigung von Anforderungen:

Mehrere Ansätze fordern die durchgängige Berücksichtigung von Anforderungen, sind dabei allerdings eher generisch oder eher speziell.

A7 Erhöhung der Konsistenz des gesamten Entwurfs:

Konsistenz ist auch eine Frage der Nachvollziehbarkeit. Verkopplungen müssen nachvollzogen werden können. Das Themenfeld der Verkopplung von Gesamtfunktionen und Teilfunktionen wurde bereits in der Literatur aufgegriffen. Ebenfalls zeigen die untersuchten Tools Möglichkeiten einer Verkopplungsanalyse, allerdings auf einem nicht vollständigen Niveau.

A8 Bereitstellung einer Klassifikation von Anforderungen:

Es existieren Ansätze zur Klassifikation von Anforderungen, jedoch ohne Berücksichtigung des Entwicklungsprozesses. Es wirkt eher wie eine Ermittlungsunterstützung, in mehreren gleichberechtigten Kategorien zu unterscheiden. Die Richtung entlang des Entwicklungsprozesses ist in keinem der untersuchten Ansätze ausreichend berücksichtigt.

Zusammenfassung Nachdem eine Problemanalyse zu Anforderungen an eine Systematik zur Evolution von technischen Anforderungen geführt hat, wurden verschiedene Bereiche der Wissenschaft dahingehend untersucht. Es zeigt sich, dass keiner der untersuchten Ansätze und auch keine triviale Kombination die gestellten Anforderungen vollständig erfüllen. Die untersuchten Methoden sind entweder sehr generisch gehalten oder zu speziell mit einem zu engen Fokus. Es mangelt insbesondere an einem Ansatz, welcher Anforderungen einerseits im Kontext konzeptioneller Entwicklungsphasen nutzt und diese systematisch in die disziplinspezifischen Phasen überführt. In den folgenden beiden Kapiteln wird daher eine Systematik zur Evolution technischer Anforderungen dargestellt.

Bewertung der untersuchten Ansätze gemäß der gestellten Anforderungen		Anforderungen							
Bewertungsskala		Systematische Synthese von Anforderungen	Math. interpretierbare Analyse von Anforderungen	Darstellung von Anforderungen	Einbindung in einen Entwurfsprozess	Ganzheitliche Betrachtung des Entwurfsprozesses	Durchgängige Anforderungs- berücksichtigung	Erhöhung der Konsistenz des gesamten Entwurfs	Bereitstellung einer Klassifikation von Anforderungen
RE: Requirements Engineering SE: Systems Engineering		A1	A2	А3	A4	A5	A6	A7	A8
RE aus Softwaresicht	RE nach EBERT		\bigcirc		\bigcirc	\bigcirc			\bigcirc
	RE nach HULL	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc		\bigcirc	
	RE nach PARTSCH	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc		\bigcirc	
	RE nach POHL								
	RE nach RUPP					\bigcirc			
	Bewertung von Anforderungen				\bigcirc				
RE im SE	SysML		\bigcirc			\bigcirc		\bigcirc	
	Entwicklungsmethodik nach EHRLENSPIEL		\bigcirc				\bigcirc	\bigcirc	\bigcirc
	Entwicklungsmethodik nach PAHL BEITZ		\bigcirc			\bigcirc		\bigcirc	
	Automotive SPICE		\bigcirc	\bigcirc	\bigcirc	\bigcirc		\bigcirc	
	NASA SE		\bigcirc	\bigcirc		\bigcirc		\bigcirc	
RE in Normen	ISO 15288							\bigcirc	
	VDI 2206								
	VDI 2221								
	VDI 2223								
	ISO 25030					\bigcirc		\bigcirc	
	IEEE 29148							\bigcirc	
	QFD nach AKAO			\bigcirc		\bigcirc		\bigcirc	

Bild 3-14: Bewertung des Standes der Wissenschaft in Bezug auf die Anforderungen an eine Systematik

4 Synthese technischer Anforderungen

Zur Erfüllung der in Abschnitt 3.3.2 gestellten Anforderungen an eine Systematik wird in diesem Kapitel der Teil der Synthese genauer beleuchtet. Dabei werden Anforderungen kategorisiert, einem Entwicklungsprozess zugeordnet und zur Spezifikation geeignete Werkzeuge werden erläutert. Es wird im Anschluss ein Überblick zum Testen gegeben, um Anforderungen in einen entsprechenden Kontext zu setzen.

Ein Gerüst der Synthese von Anforderungen steht demnach im Fokus dieses Kapitels. Es stellt ein wichtiges Element der Systematik T-REQS dar (vgl. Bild 4-1). Das Gerüst soll helfen, Anforderungen zu ermitteln, zu analysieren, zu validieren und zu priorisieren: Die Evolution der Anforderungen soll unterstützt werden. Das ist der Anspruch und somit stellt sich die Frage, ob Anforderungen so miteinander verbunden sein können, dass sie den Pfad der Entwicklung prägen?

Es gibt zwei Arten der Zusammenhänge von Anforderungen untereinander: Die hierarchische Verbindung mit abstrakteren bzw. detaillierteren Anforderungen und die systembedingte Verbindung mit gleich detaillierten Anforderungen. Aus der Forderung nach einer bestimmten Höchstgeschwindigkeit folgt z.B. hierarchisch die Forderung nach einer Motorleistung, welche dann auf einer Stufe und in Zusammenhang steht mit dem Windwiderstand der Karosserie, dem Rollwiderstand der Reifen, und der Verluste im Antriebsstrang. Entwicklungen sind typischerweise ressourcenbeschränkt und können nicht in Perfektion enden. Die Diskussionen über minimal valuable Products bekräftigen diese These. Eine Gesamtheit an Anforderungen übersteigt oftmals die erfassbare Menge an Informationen und erschwert eine Priorisierung. Komplexe Zusammenhänge von Anforderungen sind ohne Systematisierung nicht zu überschauen. In diesem Kapitel wird daher die Frage beantwortet, wie Anforderungen hierarchisch gegliedert werden können und die Zusammenhänge der Anforderungen werden beschrieben. In der Praxis werden Anforderungen aus unterschiedlichsten Gründen oft ausschweifend formuliert. Auch in vorgegebenen Strukturen, werden z.B. im Automotive-Sektor Anforderungen auch auf Systemebene schon mit Umsetzungen auf Komponentenebene verbunden. Dies verletzt

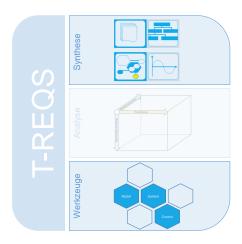


Bild 4-1: Die Synthese in T-REQS

zwar den ursprünglichen Gedanken einer reinen Problembeschreibung, andererseits wird so jedoch bereits ein Teilproblem gelöst. Dies stellt erstmal einen positiven Effekt dar, zwingt jedoch nachfolgende Entwicklungsschritte in uneindeutige Muster.

Anforderungserfassung steht im Grenzbereich zwischen Pragmatismus und Genauigkeit. Ein guter Entwickler wird immer versuchen, die Anforderungen zu erfüllen und das Beste aus dem System zu holen. Dennoch besteht die Möglichkeit, dass nicht alle Anforderungen vom Entwickler erkannt werden. Anforderungen sollten dementsprechend vollständig sein. Die Erreichung der Vollständigkeit sollte also nicht von den (Interpretations-) Fähigkeiten des Entwicklers abhängen, sondern das Resultat einer systematischen Synthese sein. Systematische Synthese kann nur entsprechend einer Systematisierung vorgenommen werden, sodass eine Klassifizierung benötigt wird. Als Einstieg seien einige Beispiele von Klassifikationen für Anforderungen genannt:

- Funktion,
- Design,
- Leistung,
- Schnittstellen.
- Sicherheit,
- Material.
- Prozesse,
- Teile,

- Organisation,
- Produktion,
- Operation,
- Zuverlässigkeit,
- Wartbarkeit,
- Umwelt.
- Verpackung.

Diese Liste ist nicht vollständig sowie ebenfalls nicht zwangsläufig hilfreich. Typische Klassifikationen von Anforderungen gehen nicht explizit auf die Zusammenhänge ein, sondern eher auf die Unterschiede. Diese Einschränkungen zeigen, dass eine Kategorisierung per se nicht die Lösung für die Entwicklung intelligenter technischer Systeme darstellt. Bei genauerer Analyse kann festgestellt werden, dass Anforderungen eine Systematisierung benötigen, die ihre inhärente Imperfektion berücksichtigt. Zurückgreifend auf die Erkenntnisse des induktiven Schließens aus Kapitel 1, kann in Bezug auf Anforderungen eine Gemeinsamkeit identifiziert werden. Ein induktiver Schluss ist laut Rescher [Res87]

"ein anspruchsvoller deduktiver Schluß [sic] mit einem Manko […] ein Enthymem, ein Argument, bei dessen Formulierung eine notwendige Bedingung fehlt, so daß [sic] die Konklusion Aussagen enthält, die nicht Bestandteil der Prämissen sind." [Res87, S. 24]

Implizite Annahmen sind ein elementares Problem bei der Ermittlung von vollständigen Anforderungen. Die Gemeinsamkeit ist demnach eine fehlende Aussage. Als Beispiel diene zum einen die Aussage, ein gegebenes mathematisches Modell berechne die Bahn der Erde um die Sonne und zum anderen, es berechne die Bahn des Jupiters um die Sonne. Ein induktiver Schluss wäre, dass dieses mathematische Modell die Bahnen aller Planeten des Sonnensystems berechnen könne. Die fehlende Prämisse ist in diesem Fall, dass das Modell die Bahnen der übrigen Planeten¹ auch berechne. Ein deduktiver Schluss wäre dagegen, wenn die Prämisse vorliege, das Modell berechne die Bahnen aller Planeten

¹Merkur, Venus, Mars, Saturn, Uranus, Neptun und ehemals Pluto

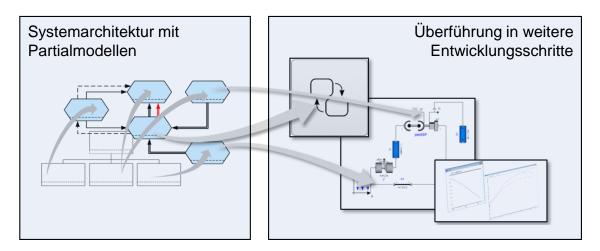


Bild 4-2: Konsistentes Vorgehen bei der Entwicklung, abgewandelt nach [HMLT17]

des Sonnensystems. Dann kann geschlossen werden, die Bahn eines bestimmten Planeten könne berechnet werden. Deduktives Schließen ist gelegentlich jedoch nicht möglich, sodass induktives Schließen die besterreichbare Antwort verspricht. Die Einführung von Prämissen, welche die Unvollkommenheit vorhandener Informationen überwindet, ist eine geeignete Lösung dieses Dilemmas [Res87]. Das sollte im Weitesten auch für Anforderungen berücksichtigt werden. Die Induktion lässt sich auch als kognitive Systematisierung deuten [Res87]. Lässt sich ein Schluss in eine Systematik eingliedern, die gewisse Charakteristika erfüllt, so ist die Wahrscheinlichkeit hoch, einen korrekten Schluss gezogen zu haben. Ist eine Entscheidung wegen mangelnder Information nicht sicher, wird die induktive Akzeptierbarkeit zu einer Sache der systematischen Angepasstheit [Res87]. Bradley interpretiert die Systematizität als Schiedsrichter der Akzeptierbarkeit [Bra12]². Als Schlussfolgerung dieser Einleitung über die Wissenschaftstheorie steht, dass in einer nicht perfekten Umgebung eine Systematisierung zur Detaillierung Klarheit verspricht.

Um ein Entwicklungsvorgehen mithilfe von Anforderungen zu realisieren, etwa durch ein *Design-by-Requirements*, wird zu Beginn eben eine solche Systematisierung von Anforderungen vorgestellt. Verschiedene Anforderungen werden in Ebenen bzw. Level systematisiert und dem Entwurfsprozess zugeordnet. Diese Einordnung stellt einen wesentlichen Schritt zur sinnvollen Systematisierung dar. Ein Überblick über die Synthese der Anforderungen wird anschließend gegeben. Vom linken Teil bis zum rechten Teil des V-Modells werden Syntheseschritte erläutert und im Anschluss bewertet.

Einerseits wird dabei die Konsistenz betrachtet, wie in Bild 4-2 zu sehen ist. Die Informationen aus einem vorherigen Schritt werden in einem nächsten Schritt berücksichtigt. Andererseits soll ein systematisches Vorgehen dabei unterstützen, die richtigen Schritte zu vollziehen (vgl. Bild 4-3), sodass auf einen Prozess eingegangen wird, um Anforderungen zu berücksichtigen.

Detaillierungen von Entwicklungen werden üblicherweise von verschiedenen Perspektiven aus betrachtet. Selbst in sehr flachen Hierarchien in der industriellen Praxis werden nicht immer alle Details kommuniziert, sodass eine Hierarchisierung der Anforderungen auch dahingehend sinnvoll erscheint.

²ursprünglich aus dem Jahre 1914

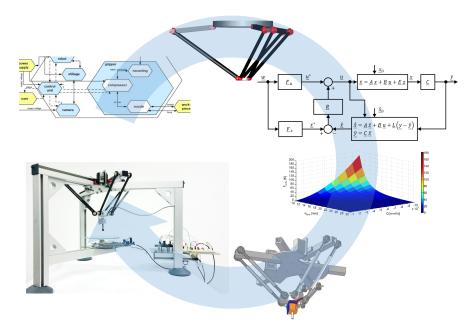


Bild 4-3: Ausschnitt von Ergebnissen aus einem systematischen Entwicklungsprozess

Das Ziel ist ein ganzheitliches systematisches Vorgehen zur Lösung der Syntheseaufgabe. Dies stellt eine direkte systematische Vorgehensweise dar. Sie beruht nicht auf indirektem Vorgehen, welches eher auf systematischem Probieren beruht, z.B. der iterativen Wiederholung elementarer Schritte. Somit ist dieses Vorgehen stärker von der Erfahrung des Anwenders entkoppelt.

4.1 Anforderungsebenen im Entwurfsprozess

Mehrere Klassifikationen von Anforderungen sind von vielen Autoren vorgeschlagen worden. Die bekannteste Klassifikation geht wohl zurück auf *Volere* [RR13]. Die Anforderungen werden dort anhand dessen unterschieden, was sie beschreiben. Dieser Abschnitt zeigt eine Zuordnung von verschiedenen Ebenen von Anforderungen zu einem Entwicklungsprozess. Die Ebenen geben einen Rahmen, welcher gewisse Rückschlüsse über einen angemessenen Detaillierungsgrad der Anforderungen in bestimmten Entwicklungsphasen zulässt. Innerhalb der Ebenen können die bekannten Klassifikationen beibehalten werden, falls es dem Anwender sinnvoll erscheint.

Entwicklung bedeutet oftmals, dass ein Problem zuerst beschrieben werden sollte, bevor nach einer Lösung gesucht wird. Es ist in jedem Fall wichtig, die Problembeschreibung von der Lösungsbeschreibung zu trennen. Die Grenze zwischen diesen Beschreibungen kann jedoch verschwimmen – Entwickler sollten dies zumindest beachten, wenn ein anforderungsorientierter Entwicklungsprozess angestrebt wird. Mit diesen Gedanken strebt die Entwicklung normalerweise von einem abstrakten zu einem detaillierten Level. Das gleiche Konzept wird für Anforderungen in dieser Arbeit verwendet.

4.1.1 Abgrenzung unterschiedlicher Anforderungsebenen

Da es bei der Betrachtung von Anforderungen im Weitesten um eine Repräsentation von Wissen geht, kann festgehalten werden, dass Kategorien die wichtigsten Elemente darstellen, das Wissen zu handhaben. Man benötigt flexible und allgemeine Repräsentationen, wenn das zu untersuchende Wissen, wie in diesem Fall angenommen, hinreichend komplex ist [RN04]. Eine Kategorisierung wird in Bild 4-4 gezeigt.

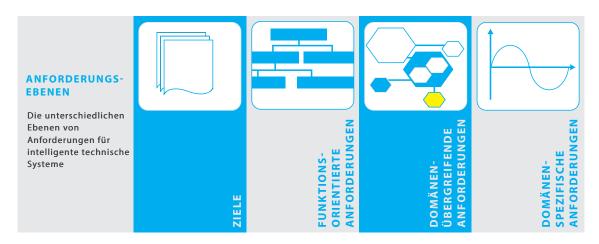


Bild 4-4: Anforderungsebenen im Entwurf

Anstatt mit eigentlichen Anforderungen in einen Entwicklungsprozess zu starten, wird eine Entwicklung oftmals mit Zielen beginnen³, welche sukzessiv in ein System überführt werden. Dieser Prozess kann systematisch mit Anforderungen unterstützt werden. Während der Systementwurfsphase werden bestimmte Funktionen definiert, welche die übergeordneten Ziele erfüllen. Dies bedeutet einen Bedarf an *funktionsorientierten Anforderungen*. Am Ende der Systementwurfsphase, vor dem Übergang in die domänenübergreifende Phase, sollten alle *domänenübergreifenden Anforderungen* verfügbar sein. Diese beinhalten alle externen Schnittstellen des Systems. Probleme während der Entwicklung entstehen üblicherweise an genau diesen Übergängen der Entwicklungsphasen. Innerhalb des disziplinspezifischen Entwurfs sind die meisten Entwickler in der Lage, die gestellten technischen Herausforderungen anzunehmen. Dementsprechend sind Fehler insbesondere bei unklaren oder missverständlichen Aufgabenbeschreibungen problematisch. Dies sollte mit den domänenübergreifenden Anforderungen vermieden werden.

Da der Lösungsraum noch teilweise offen sein sollte, um die Entwickler nicht zu stark einzuschränken, ist zu viel Detail nicht immer förderlich. Letztlich müssen die *domänenspezifischen Anforderungen* erstellt werden, die den geeigneten Detaillierungsgrad aufweisen, um Komponenten und Teilsysteme auszuarbeiten oder auszuwählen. Durch den ganzen Prozess hindurch sollten entsprechende Tests gegen die Anforderungen immer beachtet werden. Die verschiedenen Ebenen bieten daher einerseits die Möglichkeit zur

³Neben diesen Zielen stehen noch Randbedingungen, wie ein Budget, organisatorische Anforderungen oder ein Zeitplan, die sich im Laufe der Entwicklung zwar anpassen können, jedoch nur sekundären Einfluss auf das technische System haben. Es wird angenommen, dass diese organisatorischen Randbedingungen als gegeben anzusehen sind. In der Praxis kommt es hier dagegen häufig zu Problemen. Die Lösung zu diesem Problem steht hier nicht im Zentrum, sondern wird dem Projektmanagement zugeordnet.

Dekomposition und andererseits bilden sie die Basis für eine systematische Strukturierung von Testszenarien für aggregierte Systeme.

4.1.2 Einordnung in das V-Modell

Die Definition verschiedener Ebenen alleine reicht für einen effizienten Einsatz nicht aus. Es bedarf eines Prozesses, an den die Ermittlung und Konkretisierung der Anforderungen angelehnt ist. Dafür eignet sich das bekannte V-Modell, da es alle wesentlichen Entwicklungsphasen für mechatronische Systeme berücksichtigt. Der erste Unterschied zum klassischen Vorgehen nach der VDI Richtlinie 2206 ist der fehlende konkrete Entwicklungsauftrag. Ist dieser nicht bis ins Detail gegeben, benötigt es also eine entwicklungsbegleitende Detaillierung der Anforderungen: Dadurch kann gewährleistet werden, dass die Anforderungen die benötigte Detaillierung aufweisen. Dies fördert ebenso einen agileren Prozess.

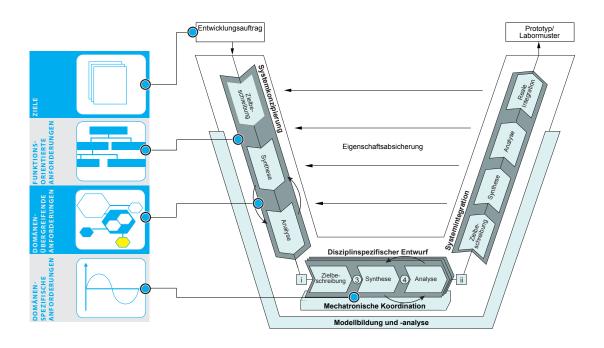


Bild 4-5: Anforderungsebenen in einem systematischen Entwurfsprozess

Die zu erreichenden Ziele sind die Eingangssituation in das V-Modell, wie in Bild 4-5 zu erkennen ist. Diese werden durch die Phase, die *Zielbeschreibung* genannt wird, in funktionsorientierte Anforderungen überführt. Im Wechselspiel aus Synthese und Analyse in der Entwicklung werden die domänenübergreifenden Anforderungen ermittelt. In der disziplinspezifischen Entwicklungsphase liegen nach der dortigen Zielbeschreibung die domänenspezifischen Anforderungen vor.

4.2 V-Modell: Systemkonzipierung

Die Systemkonzipierungsphase dient der Schaffung einer domänenübergreifenden Prinziplösung, die in den weiteren Entwicklungsphasen ausgearbeitet werden soll.

4.2.1 Ziele

"Etwas, worauf jemandes Handeln, Tun o. Ä. ganz bewusst gerichtet ist, was jemand als Sinn und Zweck, angestrebtes Ergebnis seines Handelns, Tuns zu erreichen sucht" [Dud17].

Wenn nicht gerade ein konkreter Entwicklungsauftrag vorliegt, dann startet eine Entwicklung typischerweise mit Zielen. Diese sollten im weiteren Verlauf spezifiziert werden. Nun fallen solche Ziele nicht vom Himmel, sondern ergeben sich aus verschiedenen Quellen. Die wichtigste Bezugsquelle sind die Stakeholder. Alle Stakeholder, also neben Auftraggebern und Kunden auch Normen und Gesetze, sollten ihre Erwartungen explizit kundtun. Es ist an dieser Stelle selbstverständlich nicht verboten, auch bereits konkrete und detaillierte Anforderungen aufzunehmen, diese sollten sich jedoch unbedingt in die Systematisierung eingliedern. Nun existieren neben Erfahrungen und Recherchen eine Reihe an Techniken, die zur Identifikation von Zielen dienen können (vgl. Anhang A2). Im Besonderen für eine technikorientierte Herangehensweise können Anwendungsszenarien sehr gut geeignet sein. Ein Anwendungsszenario ist eine Beschreibung einer Situation der Benutzung des Systems. Dies hilft in vielerlei Hinsicht, da die Entwicklung in einem späteren Schritt gegen diese Anwendungsszenarien getestet werden kann.

Die weiteren Nutzeranforderungen und Stakeholderanalysen vervollständigen also die Ermittlung der Ziele. Diese sind eine Beschreibung einer gewünschten Leistung oder Eigenschaft, die das System bietet. Darunter fallen z.B. die Entwicklung eines umweltfreundlichen Systems oder die Entwicklung einer Vorrichtung zum Entfalten von geknickten Banknoten. Andererseits können dies auch messbare Ziele sein, wie z.B. der Transport eines Werkstücks in weniger als einer Minute. Solch ein konkreteres Ziel könnte in diesem Fall direkt in einige Funktionsanforderungen überführt werden.

4.2.2 Funktionsorientierte Anforderungen

"Funktionen beschreiben das Verhalten von Produkten, oder Teilen des Produktes, vorzugsweise in Form eines Zusammenhangs zwischen Eingangs- und Ausgangsgrößen; häufig erst nur gewollt oder gewünscht" [VDI97, S. 6].

Der Übergang von den Zielen zu den funktionsorientierten Anforderungen kann fließend sein. Funktionsanforderungen sollten jedoch weitergehend aufgeschlüsselt sein und das System detaillierter beschreiben. Im besten Fall reicht die Aufschlüsselung bis es nicht mehr weiter geht, also bis zu konkreten zu erfüllenden Funktionen auf Komponentenebene. So kann von den Funktionen ausgehend direkt auf Lösungselemente geschlossen werden. Der Syntheseschritt wäre demnach inklusive. In vielen Fällen wird dies jedoch nicht vollständig möglich sein. Vielmehr sind die funktionsorientierten Anforderungen

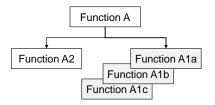


Bild 4-6: Schematische Darstellung einer morphologischen Funktionshierarchie [LJT16]

am nächsten mit den typisch genutzten Anforderungen verwandt. Somit wäre eine Detaillierung in etwa soweit vorzunehmen, bis das verbleibende Entwicklungsrisiko von allen Stakeholdern akzeptiert werden könnte. Sollte ein System alle an dieser Stelle genannten Funktionen erfüllen, jedoch keine weitere, wäre es dennoch eine erfolgreiche Entwicklung. Die funktionsorientierten Anforderungen überschreiten diese Grenze insoweit, als dass Lösungen und Alternativen mittels einer morphologischen Funktionshierarchie Berücksichtigung finden dürfen. Oft verbauen zahlreiche Entwicklungsoptionen eine gezielte Prozessführung und bremsen unnötig aus. Die Integration von Varianten in die Funktionshierarchie hilft dabei, dieses Problem zu lösen. Das ist zwar ein Randproblem, jedoch vereinfacht dieses Vorgehen einen optionsreichen Entwicklungsprozess.

Eine Funktionshierarchie hilft demnach bei der Verfeinerung von Zielen in funktionsorientierte Anforderungen. Abstrakte Funktionen werden auf einen Detaillierungsgrad gebracht, welcher spezifisch ausdrückt, was das zu entwickelnde System tun soll. Für gewisse Probleme können mehrere Funktionen existieren, die eine Lösung bieten. Oftmals führt dies, wie bereits angemerkt, zu einer Verlangsamung des Prozesses, weil die Entwickler nicht zu früh Lösungsoptionen fallen lassen wollen. Eine detaillierte Analyse ist der sinnvollste Weg zu einer Entscheidung. Jedoch befinden sich andere Entwicklungsschritte ggf. in einer Abhängigkeit einer solchen Entscheidung. So eine Verlangsamung kann ein signifikantes Hindernis sein. Eine morphologische Funktionshierarchie stellt eine potentielle Lösung dar. Die Abhängigkeiten zu nachfolgenden Schritten können so ermittelt werden und entsprechend frühzeitig berücksichtigt werden. Die Entscheidung über die zu realisierende Option wird ressourcengerecht nachgeholt.

Bild 4-6 zeigt schematisch die bewusste Berücksichtigung von Entwicklungsoptionen, um den Entwicklungsprozess nicht unnötig zu verlangsamen. Bereits in so frühen Phasen der Entwicklung können Simulationen zur Verfeinerung der Funktionen empfehlenswert sein [LLT15]. Die Beschränkung der Dekomposition durch Domänenwissen wird so umgangen [LLO+15]. Das bedeutet, dass ein Mangel an Domänenwissen nicht zu einer zu abstrakten Beschreibung führt. Gerade in Hinblick auf die angestrebte disziplinübergreifende Entwicklung, in welcher Experten verfügbar sind, wird so eine nicht hinreichend detaillierte Funktion offensichtlich. Sobald alle funktionsorientierten Anforderungen definiert sind, werden Anforderungen auf Systemlevel⁴ bestimmt. Hinweise auf diese sind bereits vorhanden und ggf. ist eine Systemarchitektur weitestgehend festgelegt. Sollte eben das nicht der Fall sein, bedarf es einer Lösung zur Erfüllung der geforderten Funktionen. Ein wichtiger Aspekt dabei sind die Schnitstellen im System. Diese werden im folgenden Abschnitt weiter erläutert.

⁴Aus dem Englischen: system architecture requirements

4.3 V-Modell: Transition

Die erfolgreiche Transition von der Systembeschreibung zur Systemausarbeitung stellt einen wesentlichen Schritt hin zu einer erfolgreichen durchgängigen Entwicklung dar. In der Praxis ist dies jedoch oftmals eine Schwierigkeit. Die Systembeschreibung ist häufig zu abstrakt für die Ausarbeitung. Es bedarf weiterer Analysen, bevor die eigentliche Ausarbeitung beginnt. Durch einen personellen Verantwortungsübergang können Inkonsistenzen auftreten. Es bedarf einerseits geeigneter Schnittstellen und möglichst durchgängiger IT-Lösungen, andererseits ist dies ein allzu selten betrachteter Bereich der Forschung, der jedoch wesentliches und elementares Potential zur Verbesserung mit sich bringt. Systematisches Vorgehen ist hier in einem ersten Schritt notwendig. Die digitale Durchgängigkeit ist dann eine Frage von Toolkapazitäten.

4.3.1 Domänenübergreifende Anforderungen

Es zeigt sich im vorangegangenen Abschnitt, dass Synthese zu einem Teil funktional erfolgen kann. Funktionen gehorchen jedoch noch keinen physikalischen Gesetzen und die Realisierbarkeit dieser muss gewährleistet werden. Es gilt, Elemente zu identifizieren oder zu entwickeln, welche die Funktionen bestmöglich erfüllen. Dieser bedeutende Schritt einer Entwicklung endet mit einer Art von Architektur. Die Elemente werden geplant und ihnen wird eine Rolle zugewiesen. Die Rolle ergibt sich dabei aus den funktionsorientierten Anforderungen, die Elemente selber müssen ausgewählt oder entwickelt werden. Natürlich stehen diese Elemente in Relation zueinander, was anhand einer Wirkstruktur verdeutlicht werden kann. Diese zeigt außerdem die Schnittstellen der Systemelemente, denen eine wichtige Bedeutung zukommt. Eine Schnittstelle ist die

"Grenze zwischen zwei Funktionseinheiten, die durch funktionelle Merkmale, Signalkenngrößen oder andere Merkmale als geeignet festgelegt ist" [DIN14, S.28].

Bei Systemelementen aus unterschiedlichen Disziplinen ist eine solche Übersicht besonders hilfreich. Die Schnittstellen können durch eine Wirkstruktur also übersichtlich dargestellt werden. Dennoch bedarf es einer geeigneten Berücksichtigung in den Anforderungen. Eine Schnittstelle erfüllt die Verbindung zwischen mindestens zwei Systemelementen und genau das sollte als Anforderung formuliert werden. Ggf. wird eine bestimmte Spannung benötigt oder ein gewisser Datentyp. Gerade bei Schnittstellen gibt es Vorgaben zu den entsprechenden Angaben. Die Dokumentation von Schnittstellen in einer Anforderungsliste ist z.B. für die Raumfahrttechnik in der DIN EN 16603-10-24 genormt [DIN17]. Mit den Schnittstellen und den Systemelementen steht eine Prinziplösung.

""Prinzipielle Lösungen"beschreiben unscharfe bzw. grobe aber funktionsbestimmende Vorstellungen zur Realisierung von Produkten und sind durch die Einbeziehung von Effekten gekennzeichnet" [VDI97, S. 6].

Funktionen werden durch Systemelemente und deren Interaktionen erfüllt. Dies kann, wie bereits dargelegt, in einer Wirkstruktur dargestellt werden [LLO+15]. In einem einfachen



Bild 4-7: Schematische Darstellung einer Wirkstruktur

Beispiel kann angenommen werden, es gäbe drei Systemelemente SEn, welche miteinander in Beziehung stehen (Bild 4-7). Zwischen SE1 und SE3 werden Informationen ausgetauscht und zwischen SE1 und SE2 wirkt eine Kraft. Zur besseren Vorstellung kann SE1 als Greifer, SE2 als Werkstück und SE3 als Steuerung gedacht werden. Die Wirkstruktur entspricht so einem ersten physikalisch motivierten Modell des Systems. Für die technischen Anforderungen bedeutet dies einen weiteren Übergang zu domänenübergreifenden Anforderungen, wie externe Schnittstellen und Verbindungen. Die Traceability zu den funktionsorientierten Anforderungen sollte selbstverständlich gewahrt bleiben. Dies kann entweder mit den Werkzeugen in der SysML getan werden oder durch eine hierarchisch gegliederte Anforderungsliste. Die Wirkstruktur eignet sich ebenfalls zur ersten Analyse impliziter Annahmen. Die Wirkstruktur gibt einen Überblick über die Wechselwirkung von Systemelementen. So können Abhängigkeiten von Systemelementen aufgedeckt werden. Prinzipiell eignen sich dafür auch Methoden, wie die Fehlerbaum-analyse [DIN90]. Diese würde jedoch wegen ihres Fokus auf ungewünschte Zustände sehr umfangreich werden und eine Vermischung von Funktionen, Zuständen und Systemelementen nach sich ziehen. Auch die Fehlermöglichkeits- und -einflussanalyse (FMEA) stellt eine Methode dar, die durchaus Nutzen stiften kann [DIN06]. Jedoch wird auch bei ihrer Anwendung im Zweifel ein sehr umfangreiches Ergebnis erhalten werden.

Die Wirkstruktur kann mit ihren Systemelementen auch einen Überblick über die Detaillierung der Entwicklung geben. Lösungsmuster und konkrete Lösungselemente können in der Wirkstruktur aufgenommen werden. Für Lösungsmuster bedarf es einer weitergehenden Entwicklung. Diese kann z.B. mit semantischen Technologien unterstützt werden [Oes17].

4.3.2 Domänenspezifische Anforderungen

Letztlich liegt der Sinn eines jeglichen Anforderungsprozesses in der Erreichung einer finalen und vollständigen Spezifikation. Die Detaillierung sollte soweit vorgenommen werden, dass alle Unklarheiten beseitigt sind und eine Entwicklung eindeutig weitergeführt werden kann. Für die Systeme dieser Arbeit wird solch ein Satz an vollkommenen Anforderungen durch die Gesamtheit aller Ebenen der Anforderungen angenommen, in denen letztlich die domänenspezifischen Anforderungen die detaillierteste Beschreibung darstellen. Die Forderung nach Vollkommenheit scheint an dieser Stelle ein edles und utopisches Ziel zugleich zu sein. Nichtsdestotrotz sollte es der Anspruch einer jeden Entwicklung sein, so nah wie eben möglich und sinnvoll an diesen Zustand heranzureichen.

Eine Detaillierung von Anforderungen ist für Ingenieure keine Neuigkeit. Dabei wird zwar ein methodisches Vorgehen gewählt, allerdings ist dies oftmals problemspezifisch und nicht immer systematisch. ILLG zeigt, dass für die Auslegung von Kettenfahrwerken drei Hauptbereiche von Anforderungen existieren. Sie lauten *Beweglichkeit*, *Fahrkomfort*

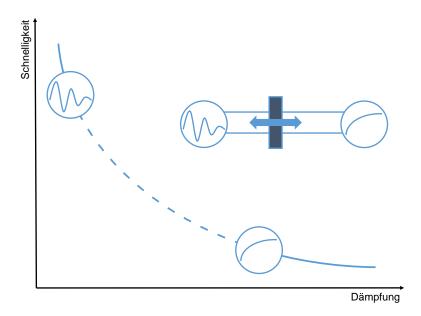


Bild 4-8: Gegenläufige Anforderungen am Beispiel des Überschwingverhaltens einer Regelung

und Fahrsicherheit [III14]. Diese können weiter untergliedert werden. Dafür können entsprechend auch Berechnungen herangezogen werden, um z.B. die maximale Vortriebskraft frühzeitig zu ermitteln. Auch können Bewegungsfrequenzen klassifiziert werden, um die Bewegungsübertragung auf den Fahrer zu analysieren. Auf diese Weise können Zielkonflikte aufgedeckt werden und erste Abschätzungen über die Einflüsse von Entwurfsfreiheitsgraden vorgenommen werden [III14]. Wielenberg nennt Auslegungsziele, welche Grenzen und Optimierungsgrößen beinhalten und stellt somit einen vorgegebenen Rahmen inklusive Bewertungskriterien der Entwicklung bereit, in welchem Anforderungen ebenfalls detailliert werden können [Wie14].

Die systematische Ermittlung von domänenspezifischen Anforderungen ist, wie der Name vermuten lässt, spezifisch. Oftmals kann jedoch eine konkrete Größe gefunden werden, die die Anforderungen spezifiziert. Da solche Angaben in komplexen Systemen oft gegenläufige Anforderungen darstellen, ist ein systematischer Ansatz notwendig. Ein Beispiel gegenläufiger Ziele kann in Bild 4-8 gesehen werden. Soll eine Regelung schneller werden, neigt sie eher zu überschwingendem Verhalten. Es wird jedoch oftmals gefordert, dass eben ein solches Überschwingen nicht stattfinden darf. Steht ein Entwickler nun vor der Aufgabe, eine optimale Lösung für ein Problem zu finden, kann er auf ein umfangreiches Angebot an geeigneten mathematischen Methoden zurückgreifen [PW88]. Oftmals werden jedoch solch konkrete Designvorgaben gescheut. Dennoch kann eine mathematische Beschreibung von Designzusammenhängen viele Potentiale aufdecken und Designvorgaben fundieren. Doch existieren weit mehr Methoden, als dass ohne einen systematischen Einsatz dieser optimale Ergebnisse zu erwarten wären. Allein die Auswahl der richtigen Modellierungsart ist individuell und abhängig von den benötigten Erkenntnissen, den geplanten Operationen oder der Weiterverwendung. Für eine geeignete Durchgängigkeit sollte ein systematischer Modellbildungsprozess dienen. Der aus SysML extrahierbare Aufbau der Wirkstruktur kann bereits in die Modellstruktur überführt werden.

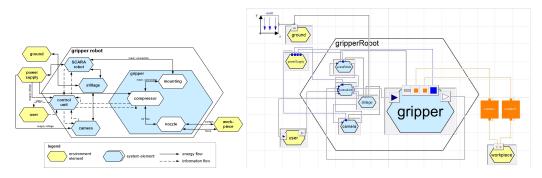


Bild 4-9: Von der Wirkstruktur im Systemmodell zum dynamischen Simulationsmodell

Bild 4-9 zeigt exemplarisch den Übergang der Wirkstruktur in eine Modellstruktur. Es ist zu berücksichtigen, dass Ansätze zur Kopplung von Systemmodellen und dynamischen Modellen die Durchgängigkeit erhöhen können. Bidirektionaler Austausch von Modelica und SysML oder Ansätze wie SysML4Modelica ermöglichen eine solche Kopplung durch ihre objektorientierten Gemeinsamkeiten [JJPB07] [RPC+12]. Die Aufgaben, die in diesem Abschnitt gelöst werden sollen, bestehen also in der Klassifikation und Identifikation von domänenspezifischen Anforderungen und der durchgängigen Berücksichtigung. Dies wird im Folgenden am Beispiel regelungstechnischer Anforderungen verdeutlicht

Exemplarisch: Regelungstechnische Anforderungen Domänenspezifische Anforderungen sind eng verbunden mit entsprechendem Domänenwissen. Das ist ein wichtiger Grund, die Anforderungssynthese systematisch anzugehen. Selbst auf disziplinspezifischer Ebene der Regelungstechnik existieren generische Anforderungen, die für alle Systeme gelten sollten. Solche generischen Anforderungen helfen ebenfalls, die disziplinspezifische Anforderungssynthese zu systematisieren. Bei genauer Betrachtung gelten in mechatronischen Systemen die folgenden grundlegenden regelungstechnischen Anforderungen [FKL+13]:

Anforderung 1: Stabilität

Das mechatronische System muss stabil sein, d. h. die Führungs- und die Störsprungantwort müssen für ∞ gegen feste Werte streben.

Anforderung 2: Genauigkeit

Das mechatronische System muss ausreichende stationäre Genauigkeit aufweisen. D.h. bei einem Eingangssprung muss die Reaktion des Systems nach dem Einschwingen in einem gewissen Abstand zum Sollwert liegen. Der tolerierbare Abstand hängt dabei vom gegebenen Problem ab.

Die Aufgabe besteht darin, dass die Forderungen nach Stabilität und stationärer Genauigkeit beide erfüllt werden, obwohl sie gegensätzlich sind. Wenn jedoch beispielsweise auf einen Führungssprung die Regelabweichung nur sehr langsam abklingt, ist der Regelkreis zwar theoretisch stabil, praktisch aber fast unbrauchbar. Er muss außerdem also eine gewisse Dämpfung aufweisen. Andererseits aber nicht zu langsam sein. Daher gibt es zwei zusätzliche qualitative Anforderungen an die Dynamik⁵ [FKL⁺13]:

⁵vgl. Bild 4-8

Anforderung 3: Dämpfung

Die Antwort auf einen Führungs- bzw. Störsprung soll nicht zu stark über- bzw. unterschwingen und nicht zu sehr oszillieren.

Anforderung 4: Schnelligkeit

Die Antwort auf einen Führungs- bzw. Störgrößensprung soll in nicht zu langer Zeit dem stationären Wert genügend nahe kommen.

Die Forderungen drei und vier sind ebenfalls gegenläufig. Zu viel Dämpfung wird den Regelkreis verlangsamen. Prinzipiell sind diese Anforderungen wenig präzise. Spezifizierungen der Anforderungen können jedoch über Quantifizierung geeigneter Größen geschehen. Diese können dann auch getestet werden. Der Test dieser Forderungen kann prinzipiell auf zwei Arten geschehen. Dies sind z.B.

- Testfunktionen,
- Inhärente Systemeigenschaften.

Als Testfunktionen eignen sich z.B. der Einheitssprung, der Dirac-Stoß oder eine harmonische Anregung zur Ermittlung des Frequenzganges. Aus diesen Anregungen können Werte, wie z.B. ein Bode-Diagramm, die Überschwingweite oder die 10 %-Anregelzeit, also die Zeit, in der 90% des Endwertes des Ausgangs erreicht werden, ermittelt werden. Dies ermöglicht eine systematische Analyse des Systems schon auf Anforderungsebene und erleichtert den Übergang zum Test, da dem System quantifizierbare Eigenschaften zugeordnet werden. Die inhärenten Systemeigenschaften sind durch die Pole und Eigenwerte des Systems gegeben und liegen demnach ebenfalls vor. Hierdurch können grundlegende Stabilitätseigenschaften abgelesen werden. Diese Eigenschaften ermöglichen eine systematische Analyse und spezifizieren ein System Schritt für Schritt. Darüber hinaus existieren weitere Werte, die eine Regelung spezifizieren. Eine Möglichkeit sind die sogenannten Gütemaße, wie beispielsweise die aufzuwendende Energie zur Ausregelung einer Störung. Auf Basis dieser Gütemaße können optimale Regelungen entworfen werden [FKL+13] und das System so optimal den Anforderungen entsprechend ausgelegt werden. Werden die spezifischen Methoden sukzessive angewendet, ergibt sich nach und nach ein detaillierter Satz an Anforderungen. Es ist offensichtlich, dass sich die Anforderungen in der disziplinspezifischen Phase deutlich von den Anforderungen früherer Phasen unterscheiden, wie sie üblicherweise Bestandteil von anforderungsspezifischen Herangehensweisen sind. Genau dies soll beim Einsatz der vorliegenden Systematik berücksichtigt werden, sodass sich eine Möglichkeit des Übergangs hin zu den disziplinspezifischen Anforderungen ergibt.

4.4 V-Modell: Systemintegration

Eine Entwicklung muss getestet werden. Ohne Test werden keine Rückschlüsse auf die Leistungsfähigkeit eines Systems möglich sein, zumindest nicht in zufriedenstellender Ausprägung⁶. Die schrittweise Realisierung und der damit verbundene schrittweise Test ist in der Praxis zumindest theoretisch etabliert. Ziel des rechten Astes des V-Modells

⁶Als letzter Test wird sich schließlich der Erfolg des Systems am Markt erweisen oder die Erfüllung vertraglicher Vereinbarungen. Tests helfen auch dabei, die Risiken des Misserfolgs zu reduzieren.

ist es also, die disziplinspezifisch entwickelten Komponenten und Subsysteme zu einem Gesamtsystem zusammenzufügen. Hier wird deutlich, dass die disziplinübergreifenden Entwicklungsschritte die Basis für die anstehende Integration bilden. Der Wert der domänenübergreifenden Anforderungen wird ebenso an dieser Stelle gesehen. Sie ermöglichen eben eine systematische Integration der Einzelteile, nicht bloß hierarchisch, sondern bezogen auf die domänenübergreifenden Bedürfnisse der Integration. Im Sinne des modellbasierten Entwurfs wird das System im rechten Ast entsprechend von seiner virtuellen Existenz Schritt für Schritt in die Realität überführt. Dabei unterstützen die X-in-the-Loop Technologien und ein methodisches Vorgehen, das unter Berücksichtigung der Anforderungen, zu einem funktionierenden Gesamtsystem führt. Systemintegration und Testen sind Gegenstand aktueller Forschung und Verfahren sind oft individuell. Durch einen bestandenen Test schafft es eine Entwicklung in die nächste Integrationsphase zu gelangen. Das Dilemma des Testens ist, dass es dabei oftmals nicht ausreicht, entsprechend der Ebenen der Anforderungen auf unterster Ebene zu testen und davon auszugehen, dass mit steigender Integration funktionierender Einzelbausteine auch das zusammengesetzte Element funktioniert. In Abschnitt 3.2 finden sich die Schwierigkeiten emergenter Systeme, die diese Annahme verhindern. Aus Sicht der Regelungstechnik ist dies anschaulich mit dem Schließen des Regelkreises darzulegen. Ein System für sich kann stabil sein und durch eine Regelung instabil werden. Es bedarf u.a. einer systematischen Integration, um solche Schwierigkeiten zu umgehen.

Systemintegration ist daher oftmals mit dem Test der Integration verbunden. Speziell bei komplexen Systemen sind solche Tests aufwändig. Ein System zu testen beschränkt sich wiederum offensichtlich nicht auf die Integration. Es können beliebig komplexe Szenarien gedacht werden. Die Reaktion des Systems auf Umwelteinflüsse, wie Temperatur, Luftfeuchtigkeit oder der Salzgehalt der Luft müssen ggf. getestet werden [Rey96]. Es kann unterschieden werden zwischen natürlichen und induzierten Umwelteinflüssen, wie Schwingungen, Wärmeeinbringung und Akustik. Offensichtlich gibt es weitere Bereiche, die relevant für Tests sind. So muss die Sicherheit getestet werden, was allerdings nicht primär im Fokus dieser Arbeit steht. Wie bereits angedeutet, ist der Test der Interoperabilität elementar. Die Kompatibilität von Systemelementen hat einen signifikanten Einfluss auf das System [Rey96]. Schnittstellen sind geprägt durch Mechanik, Elektrik und Information. Darunter fallen Datenaustausch, Versorgung und mechanische Verbindung von Systemelementen. Diese sind in der Wirkstruktur hinterlegt und sollten in den domänen-übergreifenden Anforderungen aufgegriffen werden, um in Tests berücksichtigt werden zu können.

Die aktuelle anwendungsorientierte Forschung hat jedoch noch keine hinreichenden Antworten auf die Frage nach systematischen Tests. Besonders wird dies derzeit am Beispiel der Automobilindustrie deutlich. Im Zuge der Automatisierung des Fahrens⁷ treten schlicht zu viele mögliche Testszenarien auf, welche mit aktuellen Methoden und realen Tests nicht zielführend abgedeckt werden können (Bild 4-10). Auf Umsetzungsebene bedeutet dies, dass neue Ansätze in die Denkweisen der Entwickler und Tester gelangen sollten. Reale Testfahrten zeigen sich als zu aufwändig, um alle Teile des Systems zu testen. Hinzu kommt die starke Vernetzung der Systeme untereinander und die neuen Funktio-

⁷Automatisiertes Fahren wird in Stufen realisiert werden. Doch bereits auf den mittleren Ebenen treten Herausforderungen hervor, die der Komplexität geschuldet sind und die Entwicklung dieser Systeme zumindest nicht leichter machen.

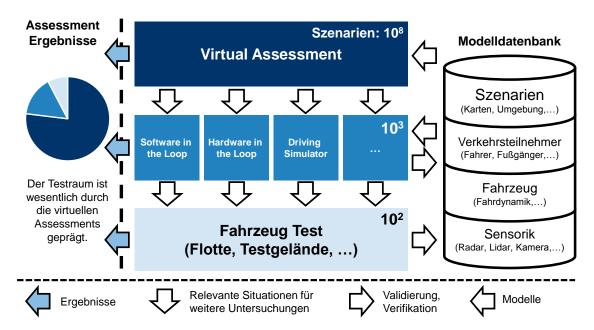


Bild 4-10: Der Anteil virtueller Tests im Zuge des automatisierten Fahrens, abgewandelt nach [SSS15]

nen. Bild 4-10 zeigt, dass die bekannten Testmethoden, wie *Hardware-in-the-Loop*, einen kleinen Teil im Vergleich zu virtuellen Tests (Virtual Assessment) einnehmen. In Zahlen ausgedrückt, werden 10⁸ Szenarien virtuell ablaufen. Diese Anzahl ist wohl schwierig in der Realität umzusetzen. Es bleibt außerdem zu klären, wie sichergestellt werden kann, dass der kleine Anteil realer Tests trotzdem entsprechend aussagekräftig ist. Das bedeutet, dass nicht nur die Anzahl der realen Tests wichtig ist, sondern auch die Qualität der Erkenntnisse. Außerdem gilt es zu identifizieren, wie bei komplexen und vernetzten Systemen das Testen optimiert werden kann. Eine Charakterisierung entsprechend der Anforderungslevel könnte auch für Testfälle eine sinnvolle Herangehensweise sein.

Die in der Automobilbranche übliche Verknüpfung von einer Anforderung mit einem Test ist dementsprechend unter Umständen nicht ausreichend. Die Kombination von verschiedenen Testfällen könnte ein Testresultat erzeugen, welches nicht auftreten darf. Ist hier die Anforderungsspezifikation in der Pflicht, solche Zusammenhänge aufzudecken? Zielführender ist ein ganzheitlicher Ansatz, in welchem Anforderungen und Tests zusammen die Systementwicklung unterstützen. Methoden, die einschränkend wirken, wie z.T. Formalisierungen können zwar unterstützen, haben aber den Nachteil der Approximation realer Gegebenheiten in einer formalen Welt. Teillösungen bringen demnach wohl nicht den erhofften Erfolg, sondern ganzheitliche Lösungen müssen gefunden werden. Es bietet sich trotz aller Herausforderungen an, die verfügbaren Methoden zum Testen sinnvoll einzusetzen, um das Risiko unvollständiger Tests zumindest zu verringern. Systematisches Vorgehen, z.B. entsprechend des V-Modells ist dafür zumindest erforderlich. Bereits die aktuellen Systeme im Automobil umfassen tausende von Anforderungen. Angenommen, diese seien in keinem systematischen Zustand – das nachfolgende Testing stände vor einer fast unlösbaren Herausforderung. In den folgenden Abschnitten werden die Herangehensweisen aus dem Testen beleuchtet.

4.4.1 Testverfahren

Testen ist

"das Messen der Performance eines Systems und die Verwendung der Ergebnisse für einen oder mehrere Zwecke" [Rey96, S. 2].

Es sollte explizit aufgeführt werden, das Testen einem Ziel dient. In erster Linie dient es der Verifizierung oder Validierung. Verifizierung ist definiert als:

"Überprüfung, ob eine Realisierung (z.B. ein Software-Programm) mit der Spezifikation (in diesem Fall mit der Algorithmenbeschreibung) übereinstimmt" [VDI04a].

Validierung ist die

"Prüfung, ob das Produkt für seinen Einsatzzweck geeignet ist bzw. den gewünschten Wert erzielt" [VDI04a].

In der Praxis finden sich unterstützende Methoden für die Verifizierung und Validierung, wie z.B. das Vorgehen im Entwicklungsprozess aus Automotive Spice 3.1 (Bild 4-11).

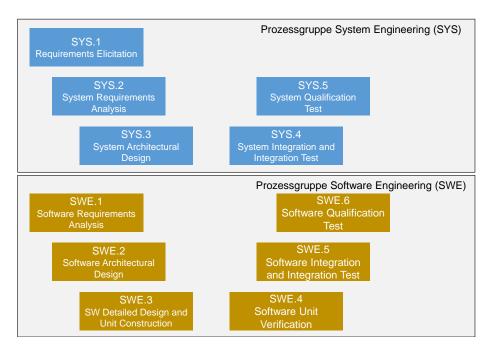


Bild 4-11: Die Entwicklungsschritte nach Automotive SPICE 3.1, abgewandelt nach [VDA17]

Hier sollte diskutiert werden, wie im Automotive SPICE Ansatz die Übergänge von der System Engineering Prozessgruppe zur Software Engineering Prozessgruppe und zurück gelingen sollen.

Es scheint inhärent davon ausgegangen zu werden, dass von der Systemarchitektur die Domänen ihre eigenen Anforderungen erstellen können. Schnittstellen stellen dabei sicherlich ein bedeutendes Element dar, aber eben nicht das einzige. Gerade die domänenübergreifenden Anforderungen sollten hier sowohl die Schnittstellen berücksichtigen, als

⁸aus dem Englischen übersetzt

auch technische Zusammenhänge. An dieser Stelle wird eine Analyse etwa mittels Bondgraphen (vgl. Abschnitt 5.2.2) einen Vorteil in der Konsistenz bei der Verifizierung und Validierung bringen.

Grundsätzlich existieren vier Optionen um zu validieren bzw. zu verifizieren: *testen, analysieren, inspizieren und demonstrieren* [Gra98a]. Diese nicht gleichberechtigten Kategorien können sich je nach Aufgabe ergänzen. Das aufwändige, jedoch zuverlässige Testen kann auch methodisch betrachtet werden.

Die Verbindung von Anforderungen und Tests wird in Testmethoden, wie Agile Acceptance Testing, Acceptance Test Driven Development, Story Testing und Example Driven Development berücksichtigt. Dies sind alles ähnliche Methoden mit verschiedenen Bezeichnungen [Adz11]. Üblicherweise ist das Problem ein striktes Festhalten an statischen Anforderungen. Somit wäre das eigentliche Problem nicht getestet, sondern lediglich die Anforderung. Dies verdeutlicht die Vorteile der hier beschriebenen Systematik der Anforderungsebenen auch für das Testen. Zur Einordnung seien einige Beispiele für Tests angeführt:

- Funktionstest,
- Lebensdauertest,
- Akzeptanztest,
- Bauteiltest,
- Komponententest,

- Materialtest,
- Systemtest,
- Betatest,
- Verträglichkeitstest.

Zahlreiche weitere Tests können sicherlich ergänzt werden. Daher sollte das Testen zuerst in einen allgemeineren und abstrakteren Kontext gebracht werden. Als typische Aufgaben des Testens gelten u.a. folgende Beispiele [Rey96]:

- Bewertung eines Konzeptes,
- Garantie der Sicherheit,
- Sicherstellung der Erfüllung der Anforderungen,
- Vermeidung von Servicefehlern,
- Bereitstellung von Feedback für den Entwickler,
- Validierung von Modellen und Simulationen,
- Vergleich von Systemen.

Testen dient letztlich in all diesen Aufgaben zur Minimierung des Risikos [Rey96]. Traditionell werden Tests dabei für hardwareintensive Systeme in die Kategorien *Materialtest, Teiletest, Komponententest, Subsystemtest*, und *Systemtest* eingeordnet [Rey96]. Es wird jedoch deutlich, dass in übergeordneten Kategorien gedacht werden muss, sodass der gesamte Produktlebenszyklus betrachtet wird, nicht allein das Produkt. So wird die Entwicklung, die Produktion, das Marketing und der Service auch in den Testprozess eingebunden [Rey96]. Im Allgemeinen wird der Testprozess in vier Schritte aufgeteilt (Bild 4-12). Diese Schritte können weiter detailliert werden, sodass ein standardisierter Prozess von der Planung bis zur Dokumentation eingehalten werden kann [IST20].

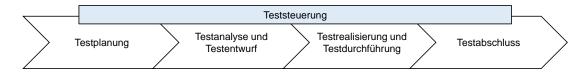


Bild 4-12: Der Testprozess nach dem International Software Testing Qualifications Board (ISTQB), abgewandelt nach [IST20]

Wird das Testen in der Entwicklung betrachtet, beinhaltet es u.a. Experimente und Tests zur Charakterisierung der Performance, zur Bewertung von Änderungen, zur Eignung für den Einsatzzweck, zur Lebensdauerermittlung, zur Steigerung des Vertrauens in ein System und viele weitere. Fünf Schritte charakterisieren dabei die Entwicklung eines Testfalls [Rey96]:

- Identifizierung der Objekte,
- Festlegung der Messgrößen,
- Ermittlung der Anforderungen an die Daten,
- Bestimmung des Testablaufs,
- Erstellung eines Analyseplans.

Die einzelnen Schritte werden entsprechend der Gegebenheiten detailliert, sodass geeignete Testfälle abgeleitet werden können. Geeignete Testfälle alleine garantieren jedoch keine Qualität, weder bei Software noch bei Hardware. Die besten Tests sorgen nicht für eine gute Qualität, wenn die Entwicklung nicht hinreichend qualitativ ist. Daher müssen Tests immer in Verbindung mit der Entwicklung und dem Entwicklungsprozess betrachtet werden [WAC12]. Anforderungen können einen wesentlichen Teil dieser Aufgabe übernehmen. Die Erfüllung der Anforderungen ist in jedem Fall bereits Teil des Testens. Prinzipiell sollte die Erfüllung jeder Anforderung durch Tests abgesichert sein. Da gerade bei komplexen Systemen eine häufige Veränderung von Anforderungen zu erwarten ist, müssen sich die Testfälle auch flexibel gestalten. Eine Nachvollziehbarkeit bei den Anforderungen unterstützt diesen Prozess. In komplexen Systemen ist darüber hinaus die Testautomatisierung von besonderer Bedeutung.

Testautomatisierung

"Testautomatisierung ist die Durchführung von ansonsten manuellen Testtätigkeiten durch Automaten" [SBB12, S. 7].

Das weite Themenfeld lässt sich zum einen zur Sicherstellung einer gewissen Qualität (meistens der Software) nutzen und auf der anderen Seite als Teil eines Entwicklungsprozesses sehen – hier dienen als Beispielprozesse SPICE, CMMI und auch das V-Modell. Es gibt eine Reihe an Tools, die bei der Testautomatisierung unterstützen. Dies sind einerseits proprietäre Tools, wie HP Quicktest Professional oder IBM Rational Functional Tester und andererseits Open Source Tools, wie Selenium oder Canoo Webtest (vgl. [SBB12]). Diese Tools sind sehr softwareorientiert. Im Sinne eines ganzheitlichen Vorgehens sind u.a. die Tools dSPACE Veos und NI Veristand zu nennen. Testautomatisierung ist sehr an

die zu Verfügung stehenden Werkzeuge gebunden. Ein einheitlicher Leitfaden existiert bisher nicht [SBB12]. Es lässt sich jedoch ein *fundamentaler Testprozess* (vgl. Bild 4-12) darstellen [SBB12] [SRWL08]:

- Planung und Steuerung: Erstellung des Testkonzepts, inklusive der Dokumentation, z.B. mithilfe der Norm IEEE 29119 [ISO13].
- Analyse und Design: Anknüpfung an Anforderungen und Beginn der Umsetzung der Testautomatisierung.
- Realisierung und Durchführung: Testfälle werden konkretisiert, implementiert und durchgeführt.
- Auswertung und Bericht: Prüfung der Testergebnisse gegen Kriterien.
- Abschluss: Evaluierung und Archivierung.

Laut der ISO/IEC 25000 [ISO11b] sollen mehrere Qualitätskriterien die Software bewerten. Um Mängel an der Softwarequalität zu finden werden Tests eingesetzt. Dies ist – stark vereinfacht – die Sicht des Softwareengineerings auf Testautomatisierung. Im Bereich funktionaler Sicherheit im Automobil hat sich die ISO Norm 26262 etabliert, die eine entsprechende Richtlinie darstellt [ISO11c].

Requirements-Based Testing

Requirements-Based Testing ist eine Methode, in welcher dem Dilemma entgegengetreten wird, dass umfangreiche und vollständige Tests nicht immer mit einem angemessenen Budget durchgeführt werden können. Daher werden Verfahren benötigt, Tests zu priorisieren und entsprechend auszuwählen. Ein wesentlicher Bestandteil ist dabei das Risiko, die Gefahr der Beschädigung oder des Verlusts [PvJV04]. Es gilt zwischen Projektund Produktrisiko zu unterscheiden. Das Risikomanagement umfasst dabei vier Schritte [PvJV04]:

- Identifikation des Risikos,
- Klassifizierung des Risikos,
- Zuordnung und Auswahl des Risikos,
- Beobachtung des Risikos.

Das Risikomanagement in Bezug zum ressourcenbeschränkten Testen findet unter dem Credo statt, dass mit den höchsten Risiken gestartet wird. Eine bekannte Methode zur Risikoabschätzung ist die Fehlermöglichkeits- und -einflussanalyse. Auch die MoSCoW Methode ist hier zu nennen [PvJV04]⁹. Hier werden Tests zu verschiedenen Risikoprioritätsstufen zugeordnet, von denen die wichtigsten unbedingt durchgeführt werden. Tests mit niedrigerer Risikoprioritätsstufe werden je nach Ressourcen behandelt. Allein die Risikobetrachtung reicht jedoch nicht aus. Die Anforderungen müssen einbezogen werden. Oftmals werden Tester verleitet sein, eher die Entwicklung zu testen, als die Anforderung. In einer guten Entwicklung gehen die Anforderungen in entsprechende Eigenschaften über, was diesem Vorgehen die Problematik nimmt [Gra98a], systematischer ist jedoch ein Test entsprechend der Anforderungen.

⁹Must, Should, Could, Won't test

Existiert ein Risiko ohne Anforderung, muss entweder die Anforderung hinzugefügt werden oder das Risiko bewusst vernachlässigt werden. Andernfalls, wenn eine Anforderung existiert, die ohne Risiko angenommen ist, bedarf es einer neuerlichen Betrachtung des möglichen Risikos. Gegebenenfalls bleibt auch die Option, die Anforderung zu streichen. Gerade bei voluminösen und überfüllten Anforderungslisten ist dies eine denkbare Option [PvJV04]. Eine weitere Möglichkeit besteht in der *Budgeting, Flowdown, Allocation Methode* [Gra98a]. Hier werden Anforderungen auf Systemebene mit einem Budget ausgestattet. Alle zugehörigen und untergeordneten Anforderungen müssen mit dem Budget erfüllt und getestet werden, sodass die Allokation von Ressourcen sukzessive erfolgt.

4.4.2 Testumgebungen: Prototypen und Prüfstände

Zur Durchführung von Tests werden neben Prozessen und Werkzeugen auch Testumgebungen benötigt. Diese können modellbasiert, als realer Prüfstand oder durch das reale System erfolgen. Die Auswahl geeigneter Umgebungen erfordert eine Auseinandersetzung mit der zu testenden Eigenschaft. Einfache physikalische Gegebenheiten können modellbasiert getestet werden. Kompliziertere Sachverhalte, wie etwa der Einfluss von Reibung wird nur schwierig ausschließlich modellbasiert erfolgen können. Das Vertrauen in den Test muss gewährleistet sein. Liegen also keine validierten Testmodelle vor, sollte prinzipiell eher der reale Test vorgezogen werden.

Virtual Prototyping und XiL Trotz der Vorteile realer Tests, wird das Virtual Prototyping ein immer bedeutenderes Werkzeug. Tests können am Modell virtuell durchgeführt werden, was neben einer Kostenminimierung eine Zeiteinsparung und eine Risikominimierung bedeutet. Letztlich kann eine Verlagerung des Testaufwands in frühere Entwicklungsphasen vorgenommen werden. Die genutzten Modelle (vgl. Abschnitt 2.4) müssen dafür eine geeignete Güte aufweisen. Ein wesentlicher Treiber modellbasierter Tests kann die modellbasierte Entwicklung sein, wofür die Anforderungen, die getestet werden sollen, mit Modellen verknüpft werden müssen. Dafür stellt die hier beschriebene Systematik einen Ansatz bereit. In industrienahen Anwendungen werden die Tests an Modellen in quasi identischen Umgebungen bereitgestellt. Vorreiter dieser Entwicklung ist die Automobilindustrie, in welcher modellbasierte Tests bereits seit geraumer Zeit ein Standardvorgehen darstellen. Neben dem rein virtuellen Testen ergeben sich gerade im Kontext domänenübergreifender Anforderungen besonders geeignete Möglichkeiten aus dem Kontext XiL. Wie in Bild 4-10 deutlich wird, können X-in-the-Loop Methoden wirksam sein zum Testen in einem komplexen und interdisziplinären Umfeld. Eine Branche, die ebenfalls komplexe, dynamische und intelligente Systeme bereitstellt, findet sich in der Hausgerätetechnik. Den Herausforderungen des engen Zusammenwirkens verschiedener Disziplinen in einem technologisch anspruchsvollen Umfeld wird seit geraumer Zeit mit fortgeschrittenen Methoden aus dem Bereich X-in-the-Loop entgegengetreten. Bild 4-13 zeigt die systematische Integration von Model-, Software-, Hardware-in-the-Loop oder Rapid-Control-Prototyping Ansätzen in den Entwicklungsprozess¹⁰.

¹⁰Dieses Vorgehen wurde von Michael et al. im Fraunhofer Instituts für Entwurfstechnik Mechatronik mitentwickelt.

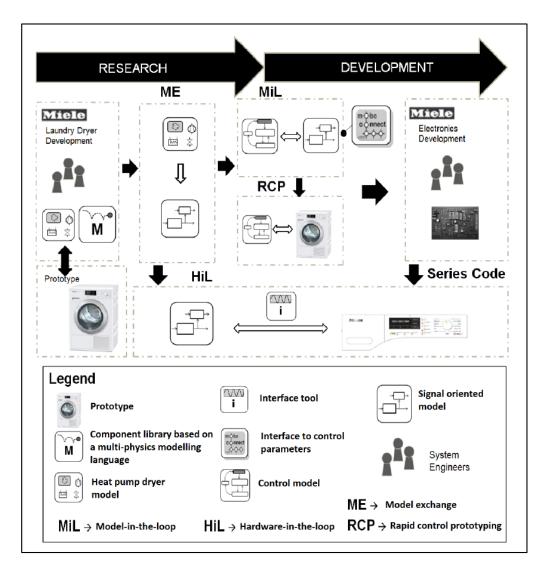


Bild 4-13: Ein strukturiertes Vorgehen bei der Entwicklung von Hausgeräten [HMH+18]

4.5 Weitere Eigenschaften der Durchgängigkeit

Die Durchgängigkeit anhand der Anforderungen sollte bei Anwendung der Systematik eine erhebliche Steigerung erfahren. Zur vollständigen digitalen Durchgängigkeit sind weitere Schritte zu berücksichtigen. Einerseits stellt die Dokumentation mit den Entwicklungsartefakten eine kritische Stelle dar, da der Prozess ohne die richtigen Dokumentationsmethoden einen Teil seiner Leistungsfähigkeit einbüßt. Andererseits sollte der durchgängige Prozess im ganzheitlichen Kontext des Lebenszyklus betrachtet werden, um den Herausforderungen moderner Systeme gerecht zu werden. Die Wertschöpfungsverschiebung, die z.B. derzeit in der Automobilbranche zu Daten hin vollzogen wird, zeigt den Bedarf an einer ganzheitlichen Betrachtung. Daten während der Nutzungsphase von Systemen werden immer bedeutender, daher müssen sich auch Entwicklungsprozesse dieser Herausforderung stellen.

4.5.1 Dokumentation von Anforderungen

Neben den traditionellen dokumentzentrierten Vorgehensweisen mit den Artefakten *Lastenheft*, *Pflichtenheft* und *Anforderungsliste* können moderne Methoden eingesetzt werden. Eine modellbasierte Vorgehensweise bietet dabei die größten Möglichkeiten der Durchgängigkeit. Dafür wird das Requirements Diagram aus der SysML vorgestellt und als ein generisches Austauschformat die ReqIF Spezifikation. Darüber hinaus sollte nicht vernachlässigt werden, dass gewünschte Eigenschaften eines Systems auch in dynamischen Modellen dokumentiert werden können.

Requirements Diagram

Im Systemmodell der SysML können konsistent Informationen hinterlegt werden. Als besonders geeignet für die Dokumentation von Anforderungen ist das Requirements Diagram. Hier können alle Anforderungen in einer übersichtlichen Form hinterlegt werden. Beispielhaft kann in Bild 4-14 die Einbindung der in dieser Arbeit vorgeschlagenen vier Anforderungslevel betrachtet werden. Einerseits sind die Anforderungsebenen dargestellt und mit den entsprechenden unterstützenden Werkzeugen (Funktionshierarchie, Modellbildung, etc.) verbunden. Andererseits ist die Dokumentation entsprechend dieser Ebenen in Form eines SysML Requirements Diagram zu sehen. Die unterschiedlichen Verknüpfungen der Anforderungen untereinander können modular und integral sein und somit auch komplexe Zusammenhänge darstellen. Gerade, wenn mehrere Systemelemente zur Erfüllung einer Funktion dienen, können diese Zusammenhänge kenntlich gemacht werden. Dieses Diagramm kann als Teil des Systemmodells im Verlauf der Entwicklung gepflegt werden und bietet Verknüpfungsmöglichkeiten zu anderen Modellen. Es gewährleistet damit ganz im Sinne der SysML eine geeignete Möglichkeit zur Dokumentation von Anforderungen.

ReqIF

Die Möglichkeiten über das Requirements Diagram und auch über Tools, wie Doors sind weitreichend. Die Annahme, dass alle Beteiligten einer Entwicklung auf die gleichen Werkzeuge zugreifen, ist in der Praxis oftmals nicht gegeben. Im Sinne einer übergreifenden Betrachtung ist daher ein Austauschformat für Anforderungen, wie ReqIF notwendig [OMG13]. Doch nicht alle Tools unterstützen den Export von ReqIF-Daten. Das Austauschformat wurde 2011 von der Object Management Group als formale Spezifikation akzeptiert. ReqIF dient zum Anforderungsmanagement, dem Anforderungsaustausch und weiteren Prozessen. Es ist ein generisches XML basiertes Format, welches mit den meisten Requirements Tools interagieren kann. Alle Informationen in einer ReqIF-Datei sind eindeutigen Positionen zugeordnet. Diese global geltenden Zuordnungen sorgen für die einfachen Austauschmöglichkeiten. Anforderungen selbst haben keine konkreten Werte, sondern greifen auf Attribute zurück. Diese können durch die Anwendung definiert werden.

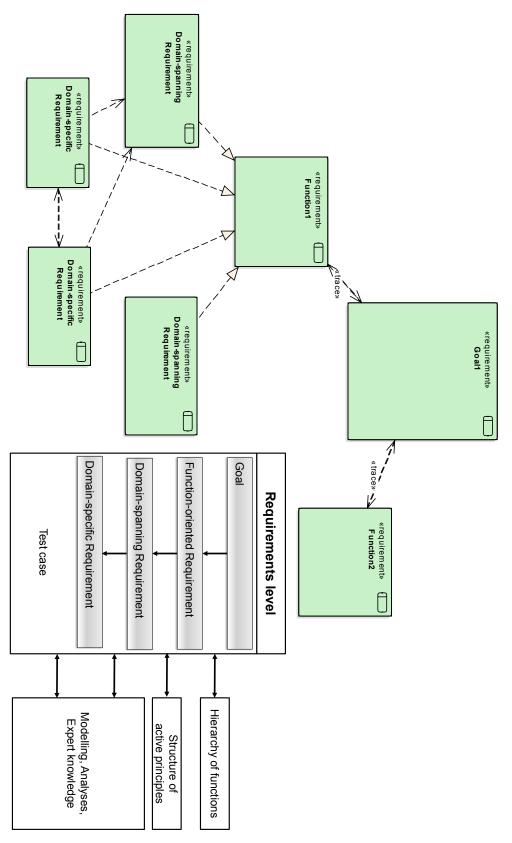


Bild 4-14: Das Requirements Diagram aus SysML, hier als Enterprise Architect Modell [Spa17], im Kontext der Anforderungslevel in englischer Sprache

Verhaltensmodelle als Anforderungsdokumentation

Ein Modell eines Systems, das mit allen Stakeholdern vereinbart ist und z.B. das Verhalten eines zu entwickelnden Systems exakt widerspiegelt, stellt nichts anderes als einen Satz dokumentierter Anforderungen dar. Schaffte eine Entwicklung es, ein Verhalten soweit in einem Modell zu spezifizieren, würden fortgeschrittene Möglichkeiten der Durchgängigkeit erreichbar sein. Die Modelle dienten als Ausgangspunkt der realen Detaillierung und sämtliche Entwicklungsschritte könnten modellbasiert getestet werden. Dieses Vorgehen würde die Absicherung der Eigenschaften auf ein systematischeres Niveau heben, denn zu jeder Zeit wären sowohl logische als auch kontinuierliche Vorgänge spezifiziert und mit testbaren Simulationsergebnissen verfügbar. Hierfür benötigt es jedoch noch weitreichende Forschung und Standardisierung. Die Limitierung auf konkrete Szenarien durch Festlegung der Modelleingänge ist hier ein wesentlicher Punkt, der zwar im Kontext der Modellbildung immer beachtet werden sollte, die Grenzen eines Modells müssen jedoch eindeutig sein, wenn es als Anforderungsartefakt genutzt werden soll.

4.5.2 Der Ansatz im Lebenszyklus

Durch den Wandel der Systeme hin zu digitalisierten Systemen wird eine ganzheitliche Betrachtung nicht nur entlang der Disziplinen notwendig, sondern auch entlang der Wertschöpfung und des Lebenszyklus. Wichtig für die Durchgängigkeit ist daher, bereits während der Entwicklung spätere Phasen zu berücksichtigen. Für z.B. die Produktion kann dies durch die *Virtuelle Inbetriebnahme* (VIBN) geschehen [HMLT17]. Die Nutzungsphase kann durch gezielte Ausstattung des Systems mit Sensorik berücksichtigt werden. Im Sinne veränderter Wertschöpfung können entsprechende Daten des Systems von großer Bedeutung sein, obwohl sie wenig mit der eigentlichen Funktion gemein haben. Diese späteren Phasen müssen in die Entwicklungsphase und die Anforderungen einfließen. Die Geschäftsmodelle zielen immer öfter auf die Nutzung von Daten ab und dies sollte frühzeitig berücksichtigt werden. Bild 4-15) zeigt eine exemplarische Einordnung in einen Lebenszyklus. Idealerweise fließen die Erkenntnisse aus Produktionsphase und Nutzungsphase mit in die Entwicklungsphase ein. Die vorangegangenen Gedanken zum *Digital Twin* aus Kapitel 3 beschreiben diese ganzheitliche Sicht.

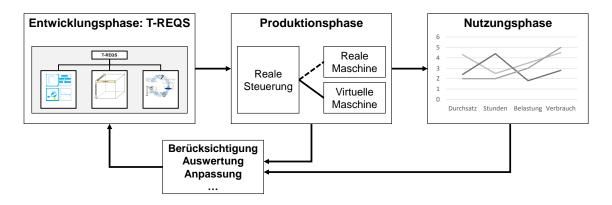


Bild 4-15: Die Einordnung des Ansatzes in den Lebenszyklus

Zusammenfassung In diesem Kapitel wurden vier Ebenen zur Kategorisierung technischer Anforderungen beschrieben. Diesen wurden verschiedene Werkzeuge zugeordnet, um konsistente Übergänge in einem Entwurfsprozess zu ermöglichen. Ausgehend von Zielen können funktionsorientierte Anforderungen, domänenübergreifende und domänenspezifische Anforderungen erlangt werden. Es wurden Möglichkeiten des Testens diskutiert. Somit ist die Systematik mit allen Phasen des Entwicklungsprozesses, von der Konzeption über die disziplinspezifische Entwicklung, bis zur Integration in den Prozess eingebettet. T-REQS stellt über die Synthese, die in diesem Kapitel erläutert wurde, hinaus auch Möglichkeiten der Analyse von Anforderungen bereit, die im folgenden Kapitel beschrieben werden.

5 Analyseverfahren für Anforderungen

John Aaron, Mitarbeiter der NASA während der Apollo 13 Tragödie und Beteiligter der Rückholaktion der Besatzung zur Erde, hält in einem Interview fest, dass er das Problem mit dem Raumschiff u.a. deshalb lösen konnte, weil er exakt wusste, welches Equipment welche Funktionen erfüllte [FH17]. Die Zusammenhänge zwischen Systemelementen und Funktionen können offensichtlich hilfreich sein. Werden Anforderungen direkt in diese Zusammenhänge mit einbezogen, ergeben sich diese Vorteile frühzeitig in einer Entwicklung.

In Kapitel vier sind Anforderungen systematisch in einen Entwicklungsprozess eingeordnet worden. Weitere Analysen benötigen fortgeschrittene Analysewerkzeuge. Ein erstes Konzept ist durch die N3 Matrix gegeben [LLO⁺15]. Darüber hinaus stehen weitere Methoden zur Analyse von Anforderungen zur Verfügung, welche im weiteren Verlauf des Kapitels erläutert werden. Die Analyseverfahren gliedern sich in die T-REQS Systematik ein (vgl. Bild 5-1) und unterstützen die Spezifikation systematisch und mithilfe von Werkzeugen.

5.1 Problemunterscheidung

Analysen im Kontext von Anforderungen können an verschiedenen Stellen durchgeführt werden. In diesem Abschnitt werden nicht berücksichtigte technische Anforderungen als ein Problem identifiziert. Dieses Problem kann auf verschiedene Weisen analysiert und gelöst werden, wie im weiteren Verlauf deutlich wird. Grundsätzlich kann in drei Kategorien gedacht werden, um unberücksichtigte Anforderungen zu klassifizieren. Traditionell wird in der Kategorie gedacht, es *fehlen Anforderungen*, weil nicht alle Quellen berücksichtigt werden. Darüber hinaus sollte bedacht werden, dass es eine gewisse Menge an Anforderungen gibt, welche von stark *untergeordneter Ordnung* sind. Diese gehören streng genommen zu einem vollständigen Satz an Anforderungen. In der Realität wird es jedoch eine Grenze geben. Es sollte weiterhin gelten, dass Anforderungen ein System

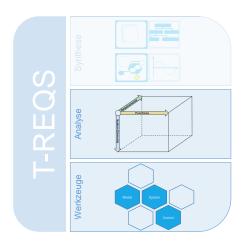


Bild 5-1: Die Analyse in T-REQS

soweit beschreiben, bis das verbleibende Risiko von allen Stakeholdern akzeptiert wird. Die dritte Kategorie unberücksichtigter Anforderungen ist durch eine *mangelnde Durchgängigkeit* gegeben. Deren Ursache sind letztlich nicht ausreichende Detaillierungen.

5.1.1 Herausforderungen an Anforderungen

Der folgende Abschnitt beschreibt unterschiedliche Herausforderungen an Anforderungen. Es existieren deutlich mehr Herausforderungen, als hier aufgeführt sind. Die Konzentration auf drei von ihnen stellt eine zielgerichtete Stoßrichtung sicher.

Quellen unbekannter Anforderungen

Die offensichtlichsten Quellen unbekannter Anforderungen sind die Stakeholder. Sind nicht alle Stakeholder bekannt, so können nicht alle Anforderungen erfasst werden. Weniger ersichtlich sind die impliziten Annahmen, die ein Stakeholder trifft. Hier lauern besonders im Kontext interdisziplinärer Systeme Gefahren. Die Voraussetzungen der beteiligten Personen sind sehr unterschiedlich und dies kann zu Missverständnissen führen, welche tunlichst vermieden werden sollten und durch eine gewisse Sorgfalt und Systematik vermieden werden können. Ein weiterer Aspekt unbekannter Quellen sind Schnittstellen. Die Einzelkomponenten können erfahrungsgemäß gut entwickelt bzw. bezogen werden. Deren Zusammenspiel stellt jedoch eine Möglichkeit unbekannter Anforderungen dar. Parasitäre Effekte und Emergenz sind weitere Einflüsse für unbekannte Anforderungen. Diese nicht vollständige Aufzählung zeigt, dass es zahlreiche Einflussfaktoren für diese Thematik gibt. Eine Beschränkung auf die Stakeholder ist angesichts dessen nicht ausreichend und geeignete Methoden müssen für eine im weitesten Sinne vollständige Analyse gefunden werden.

Priorisieren von Anforderungen

Selbst bei theoretisch vollständigen Anforderungen, wäre eine Priorisierung dieser vorzunehmen. Eine Entwicklung muss mit Ressourcen haushalten, daher kann unter praktischen Gesichtspunkten nicht jede Anforderung mit dem gleichen Budget erfüllt werden. Für die Entwicklung bedeutet dies, ein System bestmöglich zu entwickeln, also die wichtigsten Anforderungen müssen erfüllt sein und die weniger wichtigen sollten es. Diese muss und kann Unterscheidung findet sich in der klassischen Anforderungsliste (vgl. Kapitel 2). Eine feinere Auflösung zur Unterscheidung mit einer Quantifizierung der Prioritäten ist wünschenswert. Nun existieren einige Einflussfaktoren auf die Priorität von Anforderungen. Der quantifizierte Beitrag zum gewünschten System ist ein abstrakter Kompromiss, um alle Einflüsse zu berücksichtigen. Somit steht ein Bewertungskriterium fest: Etwas pragmatisch formuliert, ist dieses der Zusammenhang einer Anforderung zu den Systemelementen und den Funktionen eines Systems. Im weiteren Verlauf des Kapitels folgt die Erläuterung einer Kennzahl zur quantifizierten Beschreibung eben dieser Zusammenhänge. Auf diese Weise kann entschieden werden, welche Anforderungen relevant sind und ob ggf. weitere Anforderungen benötigt werden.

Verfolgbarkeit und Überprüfbarkeit

Letztlich kann es trotz größter Sorgfalt bei der Ermittlung der Anforderungen zu Problemen kommen. Dies hängt auch mit der digitalen Durchgängigkeit zusammen: Schlichte Fehler bei der Übertragung, z.B. aus dem Lastenheft ins Pflichtenheft, sind bei Tausenden von Anforderungen nicht auszuschließen. Hier wäre argumentativ sicherlich ebenfalls die Sorgfalt als Lösung anzubringen, jedoch scheitert dieses Argument an der Vielschichtigkeit von Entwicklungsprozessen. Übergänge von Anforderungen erfolgen an unterschiedlichen Stellen in der Praxis. Die unterschiedlichen Pfade, die Anforderungen dabei annehmen können, sind nur schwer zu überblicken. Dies erklärt, warum selbst bei bewährten Systemen nicht alle Anforderungen erfasst werden, obwohl die Voraussetzungen in diesen Fällen durch die Bekanntheit von Entwicklungsschritten nahezu optimal sind. Diese Rückverfolgbarkeit ist auch im Bereich von Anforderungsänderungen von großer Bedeutung. Eine kleine Änderung einer Spezifikation betrifft möglicherweise einen großen Teil des Systems. Die Verfolgbarkeit sollte ebenfalls für die Verbindung mit Testfällen gelten. Eine der großen Fragen in der Automobilindustrie dieser Zeit ist beispielsweise die nach dem Testen. Ein Testfall pro Anforderung kann ggf. diese nicht verifizieren, was gerade angesichts der Herausforderungen komplexer autonomer Systeme Folgen für die Entwicklung hat¹.

5.1.2 Hypothesen und Annahmen

Um die Methoden dieser Arbeit zielgerichtet anzuwenden, sind gewisse Annahmen und Hypothesen darzustellen. Diese umfassen Eigenschaften, wie die Wichtigkeit und die Priorisierung von Anforderungen. Auch die Erkenntnis, dass eine Vollständigkeit nicht durch eine Unendlichkeit an Anforderungen definiert sein muss, gehört hierzu. Es wird im Folgenden unterschieden zwischen den Zusammenhängen von Anforderungen mit der Entwicklung und der Frage nach der Hierarchisierbarkeit.

Anforderungen, Funktionen und Systemelemente

Die Wichtigkeit von Anforderungen hat Einfluss auf die Priorisierung. Doch wie kann wichtig im Kontext von Anforderungen definiert werden? Es stellt eine große Bedeutung einer einzelnen Anforderung dar; gewissermaßen eine Bewertung. Einerseits im Vergleich zu anderen Anforderungen, jedoch kann dies nicht der alleinige Maßstab sein. Es bedarf ebenfalls die Berücksichtigung der Zusammenhänge einer Anforderung mit Funktionen und Systemelementen. Denn dies ist letztlich der Zweck der Anforderungen und anhand dessen lassen sich geeignete Kriterien ableiten. Doch in all der Unterschiedlichkeit von Systemen und ihren Ausprägungen gilt es Gemeinsamkeiten zu identifizieren, welche generisch anwendbar sind. Das können beispielsweise die Verknüpfungen der Anforderungen mit dem System sein.

¹Ob sich hier andere Herangehensweisen hinsichtlich der Anforderungen oder des Testens ergeben müssen, ist eine Frage in diesem Kontext. Jedenfalls scheint eine ganzheitliche Herangehensweise sinnvoll.

Auf detaillierterer Ebene können physikalisch motivierte Bewertungskriterien gefunden werden. Die Energie etwa, die benötigt wird, um eine Anforderung zu erfüllen. Oder die Verluste, die durch entsprechende Anforderungen indiziert werden. Allgemein sind Verluste in technischen Prozessen die irreversible Veränderung der Entropie [HW12]. Verluste bedeuten immer eine Entwertung von Energie. Die bei Reibung erzeugte Wärme ist nicht so wertvoll, wie die kinetische Energie. Dies führt zu einer Verminderung der Arbeitsfähigkeit der Energie. Es ist also von Vorteil, diese Verluste zu analysieren und Gegenmaßnahmen einzuleiten. Dies erfolgt idealerweise im Zusammenspiel von Anforderungen, Funktionen und Systemelementen. Gerade dissipative Systeme bergen Gefahren nicht berücksichtigter Effekte. Bei Kenntnis dieser Effekte können entsprechende Methoden angewendet werden [LBEM00]. Oftmals können simple modellbasierte Vorgehen diese Effekte offenlegen und somit einen Rahmen für detailliertere Anforderungen schaffen. Die ganzheitliche Sicht auf Anforderungen ermöglicht so, den Herausforderungen entgegenzutreten. Die Physik trifft weder implizite Annahmen, noch vernachlässigt sie parasitäre Effekte. Die Anforderungen auf physikalischer Basis zu überprüfen ist daher empfehlenswert.

Hierarchischer Nachweis

Als ein Ausweg aus dem Zwang der Komplexität wird die Hierarchisierung gesehen. Kann eine übergeordnete Anforderung nicht durch einen einzelnen Testfall abgedeckt werden, so sollte das Kriterium des hierarchischen Nachweises gelten. Das bedeutet, dass gewissermaßen vereinfachend angenommen wird, dass eine Anforderung erfüllt ist, wenn alle untergeordneten Anforderungen erfüllt sind. In der Realität wird dies nicht für jedes System gelten. Diese Einschränkung kann jedoch auch als Klassifizierung von Systemen betrachtet werden. Ist es also möglich, über eine geschickte Anforderungsspezifikation ein System soweit zu beschreiben, dass trotz emergenter und komplexer Eigenschaften ein hierarchischer Testablauf durchgeführt werden kann, so lässt sich das Systemverhalten vollständig validieren. Der Testablauf muss in dem Fall explizit alle Schnittstellen umfassen, um der Herausforderung der Emergenz entgegenzutreten.

5.2 Analyseverfahren und Werkzeuge

Es bedarf nun der Beschreibung von geeigneten Methoden, um die Wichtigkeit von Anforderungen zu bewerten und unberücksichtigte Anforderungen zu identifizieren. Mögliche Herangehensweisen sind im Zeitalter immer schnellerer Rechenleistung auch statistische (neuronale Netze, Bayessche Netze, also Lernen als Form unsicheren Schließens aus Beobachtungen) [RN04] und stochastische Herangehensweisen, wie z.B. die Maximum-Likelihood-Methode [Bei95]. Auch der Einsatz von Wissensrepräsentation, wie der Prädikatenlogik, ließe sich diskutieren. Jedoch bietet selbst ein ausgearbeitetes System die Möglichkeit von Unsicherheit: Sensoren liefern zwar lokale Informationen, ein Großteil der Systeminformationen bleibt dennoch unbekannt. Daher fällt es schwer, mit diesem Ansatz die Komplexität intelligenter technischer Systeme abzudecken. Theoretisch ist es denkbar, alle möglichen Konsequenzen in einer Abhängigkeit darzustellen. In der Praxis ist dies jedoch noch nicht zweckmäßig. Russel nennt drei Gründe für das Scheitern der Prädikatenlogik, eine gesamte Domäne abzudecken [RN04]:

- Faulheit.
- theoretisches Unwissen,
- praktisches Unwissen.

Theoretisches Unwissen bezieht sich drauf, dass in einer Domäne wie der Ingenieurwissenschaft, nicht alles Wissen vorhanden ist. Praktisches Unwissen bedeutet, dass selbst im Falle einer vollständigen Regelabdeckung der Domäne im Einzelfall nicht alle benötigten Informationen vorliegen könnten. Es bleibt dementsprechend immer eine Unsicherheit, welche z.B. in Wahrscheinlichkeiten möglicher Zusammenhänge ausgedrückt werden kann. Dementsprechend werden im weiteren Verlauf mathematisch interpretierbare Methoden für unterschiedliche Detaillierungen von Anforderungen dargelegt, mit dem Ziel die Wichtigkeit von Anforderungen zu bewerten und zu identifizieren.

5.2.1 Funktionsorientierte Analyse

Lässt sich eine Funktionshierarchie nicht weiter aufschlüsseln, bedarf es eindeutig Expertenwissen. Oftmals geht es beim Lösen von Entwicklungsaufgaben aber darum, eine komplexe Gesamtaufgabe in handhabbare Teilaufgaben zu zerlegen: Das Expertenwissen also bewusst nicht in eine Anforderung aufzunehmen. Für diese Teilaufgaben lassen sich dann Lösungen finden, die anschließend zusammengesetzt werden können. In Abschnitt 4.1 wird die *morphologische Funktionshierarchie* vorgestellt. Angenommen, es würde eine Dekomposition von Funktionen stattfinden, bis zur eindeutigen disziplinspezifischen Funktion. Diese müsste anhand zweier Eigenschaften bewertet werden. Sie müsste entweder eindeutig kategorisierbar und lösbar sein oder eine entsprechende Handhabung durch geeignete Methoden erfahren. Im Folgenden werden beispielhaft Methoden aus der Regelungstechnik beschrieben und eine mögliche Herangehensweise zur Kategorisierung.

Ansätze zur Kategorisierung disziplinspezifischer Methoden

Eine grundsätzliche Kategorisierung über alle beteiligten Disziplinen der Entwicklung ist zwar anzustreben, jedoch nicht ohne weiteres umsetzbar. Ein erster Schritt im Sinne einer systematischen Herangehensweise der funktionalen Dekomposition ist die Katalogisierung der Disziplinen. Grundlagenwerke [CH08] schlagen die Disziplinen folgendermaßen vor:

- Mathematik und Stochastik,
- Physik,
- Chemie,
- Werkstoffe,
- Technische Mechanik,
- Technische Thermodynamik,
- Elektrotechnik,
- Messtechnik.

- Regelungs- und Steuerungstechnik,
- Technische Informatik,
- Entwicklung und Konstruktion,
- Produktion,
- Betriebswirtschaft,
- Management,
- Normung,
- Recht,
- Patente.

Die Problematik ist, dass diese Kategorien nicht gleichwertig sind. Sowohl im Umfang als auch im Detaillierungsgrad gehen die Disziplinen auseinander. Auch die Vollständigkeit ist nicht gegeben. Dennoch eignet sich die Kategorisierung zur Auswahl weiterer Detaillierungen. Einige dieser Disziplinen bieten weitere Unterteilungen nach Lösungsmustern geordnet an, andere eher nach Teildisziplinen. Ein Beispiel lösungsmusterorientierter Unterteilung ist die Entwicklung und Konstruktion. Konstruktionskataloge haben eine lange Tradition. Hier sind Einteilung in Kategorien, wie u.a.

- Klebverbindungen,
- Schraubenverbindungen,
- Federn,
- Achsen, Wellen, und Zapfen,
- Kupplungen und Bremsen,
- Wälzlager,
- Gleitlager,
- Riementriebe,
- Elemente zur Führung von Fluiden,
- Dichtungen,
- Zahnräder und Zahngetriebe

üblich [MWB⁺03]. In der Regelungstechnik sind dagegen selbst in Formelsammlungen heterogene Unterteilungen gebräuchlich [Ste11]. Einerseits nach Systemen (Beispiel: Linearität) andererseits nach Tätigkeiten (Beispiel: Modellbildung) und Teildisziplinen (Beispiel: Reglerentwurf). Lösungsmuster, wie *P-Regler*, *Zustandsregler* oder *Zweipunktregler* sind nicht als Kategorisierung üblich. Es existieren jedoch ebenfalls spezifische Ausarbeitungen, wie z.B. für adaptive Regelungen, die eher die lösungsmusterorientierte Sicht einnehmen [Web71b] [Web71a]. Grundsätzlich scheinen solche Kategorisierungen sinnvoll, sind jedoch durch ihre unterschiedliche Granularität nicht ideal geeignet.

Methoden zur funktionsorientierten Lösung

Wenn eine Funktion festgelegt ist, stellt sich die Frage, wie diese zu realisieren ist. Das System sollte in der Lage sein, diese zu erfüllen. In einer perfekten Welt, ohne Störungen von außen, ist eine Funktion sicher einfacher zu realisieren als in der Wirklichkeit. Damit Systeme mit solchen Störungen umgehen können, werden Regelungen eingesetzt². Diese Erkenntnis allein reicht jedoch nicht aus, um Funktionen wie gewünscht umzusetzen. Daher wird aufgezeigt, was es bedeutet, robust zu regeln und wie ein Streckenverhalten kompensiert werden kann.

Robuste Regelungen In Kapitel 2 wurden Aspekte der Modellbildung beleuchtet. Auf dieser Basis bleibt festzuhalten, dass bei Abweichungen zwischen Modell und Wirklichkeit, ein Regler in der Realität nicht mehr zwangsläufig die gleiche Wirkung erzeugt, wie bei der modellbasierten Betrachtung. Dies betrifft einerseits Ungenauigkeiten des Modells, anderseits auch strukturelle Abweichungen [FKL⁺16], z.B. durch Vernachlässigung bestimmter Effekte oder den Ausfall bestimmter Signale im Betrieb. Abhilfe schaffen hier robuste Regler. So, z.B. mithilfe des H_{∞} -Entwurfs, können Funktionen resilient ermöglicht werden. Dafür muss mathematisch einiges sichergestellt werden, z.B. dass die Systemeigenwerte im Stabilitätsgebiet liegen. Mathematisch formuliert gilt [Ack93]:

Ein Polynom $p(s) = (s - s_1)(s - s_2)...(s - s_n)$ wird als Γ -Stabil bezeichnet, wenn alle $s_i \in \Gamma, i = 1, ..., n$.

Ackermann definiert drei Regeln der robusten Regelung [Ack93]:

- 1. Die Robustheit darf nur für physikalisch motivierte Parameterwerte und nicht für beliebige Unsicherheiten des Modelles gefordert werden.
- 2. Wenn der Regelkreis geschlossen wird, soll ein langsames System langsam, ein schnelles schnell bleiben.
- 3. Eine pessimistische Stabilitätsgebietsanalyse ermöglicht einen optimistischen Entwurf.

Robuste Regelungen können helfen, Anforderungen quasi gegen Unwägbarkeiten durchzusetzen. Ein zu diesem Zweck offensichtlicheres Verfahren ist das des Kompensationsreglers.

Kompensationsregler Dieser Reglerentwurf hat die Zielsetzung, dem geschlossenen Regelkreis ein vorgegebenes dynamisches Verhalten zu verleihen [Lun10]. Es geht entsprechend darum, quantitative dynamische Forderungen zu erfüllen, was schwieriger ist, als die Erfüllung qualitativer Forderungen. Dafür wird ein streng systematisches Vorgehen bereitgestellt [FKL+16]. Damit eignet er sich prinzipiell sehr gut, um eine Anforderung nach einem bestimmten Verhalten, also einer Funktion, zu erfüllen. Das Verfahren wurde von Weber beschrieben [FKL+16] und hat seine Ursprünge in den Arbeiten von Truxal [Tru50]. Die Inhalte dieses Abschnitts sind an der einschlägigen Literatur angelehnt [FKL+16] [Lun10] [Unb08] [Tru60]. Zuerst wird der einfache Kompensationsregler

²Regelungstechnik als Disziplin ist selbstverständlich vielfältiger. Für eine geeignete Einführung sei auf die Literatur verwiesen. [Tru58] [Pal86] [FKL+16].

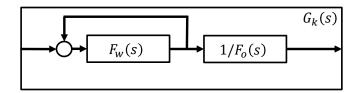


Bild 5-2: Idee des einfachen Kompensationsreglers

beschrieben, um das Konzept darzulegen. Anschließend folgt eine Beschreibung weiterer Regelungen, um quantitative Forderungen zu erfüllen.

Das Ziel des einfachen Kompensationsreglers besteht darin, dem geschlossenen Regelkreis durch Festlegung eines geeigneten Reglers ein vorgegebenes dynamisches Verhalten zu geben. Dafür wird der Kreis an ein vorgegebenes Modell angepasst. Im englischen spricht man daher auch von *model matching*. Die Übertragungseigenschaften sind durch das Modell vollständig fixiert, was gerade in Bezug auf Anforderungen eine sehr nützliche Eigenschaft ist. In diesem Fall wird von einer modellbasierten Regelung gesprochen, da das Regelstreckenmodell nicht nur zur Ermittlung der Reglerparameter genutzt wird, sondern auch im Regler selbst als inverses Modell enthalten ist (vgl. Bild 5-2). Drei Forderungen müssen dafür erfüllt sein:

- Die Regelstrecke muss stabil sein.
- Das inverse Modell der Regelstrecke muss stabil sein.
- Die gewünschte Führungsübertragungsfunktion muss stabil sein.

So (vgl. Bild 5-2) ist der Regler jedoch nicht zu realisieren, da der Term $\frac{1}{F_o(s)}$ einen Nullstellenüberschuss aufweist. Deshalb wird in einem zweistufigen Verfahren zuerst die gewünschte Führungsübertragungsfunktion $F_w(s)$ festgelegt. Anschließend wird aus $F_w(s)$ und $F_o(s)$, also der Strecke, inklusive Sensorik und Aktuatorik, der Regler $G_k(s)$ bestimmt. Dabei gilt, dass das *gewünschte Verhalten* mindestens die Verzögerung der Strecke haben muss. Für Systeme höherer Ordnung werden bestimmte Standardformen genutzt (vgl. [Unb08]), um das Verhalten vorzugeben:

$$F_w(s) = \frac{Z_w(s)}{N_w(s)}$$

Dabei steht Z für Zähler und N für Nenner.

Für die Strecke gilt

$$F_o(s) = \frac{Z_o(s)}{N_o(s)}$$
 mit Grad $Z_o(s) = m < \text{und Grad } N_o(s) = n.$

Dann ist

$$F_w(s) = \frac{G_k(s)F_o}{1 + G_k(s)F_o(s)}.$$

Wenn gilt, dass

$$F_w(s) = \frac{1}{N_w(s)}$$

ist

$$\frac{G_k(s)F_o(s)}{1+G_k(s)F_o(s)}=\frac{1}{N_w(s)}.$$

Also gilt für den Regler

$$G_k(s) = \frac{F_o(s)^{-1}}{N_w(s) - 1} = \frac{N_o(s)}{Z_o(s)[N_w(s) - 1]} = \frac{Z_k(s)}{N_k(s)}$$
 mit Grad $Z_k \leq \text{Grad } N_k$.

Für die gewünschte Übertragungsfunktion können unterschiedliche Verläufe durch mathematische Terme mit geschickter Parameterauswahl gewählt werden. Ist Überschwingen nicht gewünscht, werde etwa

$$F_w(s) = \frac{1}{N_w(s)} = \frac{1}{[1 + \frac{s}{D}]^r}$$

gewählt. Steht ein schnellerer Verlauf im Vordergrund und ist Überschwingen erlaubt, sei

$$F_w = \frac{1}{N_w(s)} = \frac{1}{\frac{1}{(1+\alpha^2)D_2} [1 + \frac{s}{5D}]^k [s^2 + 2Ds + (1+\alpha^2)D^2]} = \frac{Z_w(s)}{N_w(s)}.$$

Durch entsprechende Auswahl der Parameter kann ein vorgegebenes Verhalten z.B. hinsichtlich Abklingzeit optimiert werden. So kann z.B. je nach Auswahl der Parameter ein k-facher Pol mit komplexem Polpaar vorgegeben werden und durch Wahl der anderen Parameter ein entsprechendes Verhalten eingestellt werden. Das Verfahren ist jedoch nicht anwendbar bei nicht-minimalphasigen und bei instabilen Systemen. Darüber hinaus ist der Regler aufwändig und es besteht keine Sicherheit bezüglich der inneren Stabilität bei Störverhalten. Außerdem gelten Beschränkungen z.B. hinsichtlich Stellgrößen, Parameterfehlern und Modellungenauigkeiten.

Letztlich sind die Stellgrößenbeschränkungen jedoch kein Problem des Verfahrens. Das gewünschte Verhalten sollte als eine Anforderung interpretiert werden. Etwaige Probleme würden dementsprechend eher darauf hindeuten, dass die Anforderungen in der Entwicklung nicht konsistent berücksichtigt wurden – so sollte das gewünschte Verhalten schließlich auch realisiert werden können. Im Kontext dieser Arbeit lässt dieses sehr systematische Verfahren direkte Rückschlüsse auf mögliche Konflikte zu. Im Anhang dieser Arbeit findet sich eine Beschreibung des Vorgehens des modifizierten Kompensationsreglers.

Für Anforderungen sollte geprüft werden, ob nicht die Regelungstechnik entsprechende Methoden bereithält, um die Erfüllung einer Anforderung zu ermöglichen, jedoch unterliegt diese Aufgabe womöglich einigen Einschränkungen: Der Kompensationsregler ist gut geeignet, um eine Wunschdynamik in einem linearen System vorzugeben. Obwohl es sichtbar werden kann, dass ein Motor ggf. zu klein dimensioniert ist, stellt er sich jedoch nicht vollständig den Herausforderungen in komplexen Systemen. Könnte sichergestellt werden, das zu regelnde System sei jederzeit linear, könnte hier eine wirksame Möglichkeit zur gezielten Umsetzung von Anforderungen gefunden sein. Diese Forderung ist in der Praxis jedoch nicht gegeben. Aus gleichem Grund sind auch besonders robuste Verfahren, wie etwa die H_{∞} -Regelung [FKL⁺13] nicht geeignet, um ein Wunschverhalten des dynamischen Gesamtsystems zu garantieren.

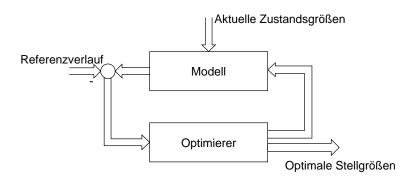


Bild 5-3: Grundsätzliche Struktur einer modellprädiktiven Regelung, abgewandelt nach [Ada18]

Besser für nichtlinearen Einsatzgebiete geeignet ist z.B. die modellprädiktive Regelung (vgl. [Ada18]). In ihr werden nicht nur Probleme, wie Stellgrößenbeschränkungen und nichtlineares Verhalten berücksichtigt, sondern dieser Regler ist besonders für den Einsatz in einem komplexen Umfeld geeignet. Als Gedankenexperiment sei es erlaubt, ein Fahrzeug auf einer Trajektorie mit einer gewissen Geschwindigkeit halten zu wollen. Wird angenommen, es gäbe für verschiedene Abschnitte unterschiedlich maximal erlaubte Geschwindigkeiten und das Fahrzeug müsste sich entsprechend anpassen, so wird ein klassischer Regler erst reagieren, sobald eine Abweichung zwischen Soll- und Ist-Geschwindigkeit entsteht. Ohne zu detailliert auf mögliche Auswege aus der Situation einzugehen, ist es offenkundig, dass in solchen Fahrsituation ein vorrausschauendes Fahren vorteilhaft ist. Ein vorrausschauendes Verhalten kann von einem modellprädiktiven Regler realisiert werden, unter Berücksichtigung von Randbedingungen und eben Stellgrößenbeschränken.

Dies wird durch einen Regler ermöglicht, der mittels eines ggf. nichtlinearen Modells und vergangener Daten das zukünftige Verhalten optimal vorausberechnet. Optimal ist hier bezogen auf eine Gütefunktion, die z.B. die Abweichungen von der Wunschtrajektorie oder Stellgrößeneingriffe berücksichtigen kann. Es wird also konsequent ein optimales Verhalten für einen gewissen Zeithorizont berechnet. Nun liegt der Gedanke darin, diese Berechnungen für einen gewissen Zeithorizont in jedem Zeitschritt neu durchzuführen, um in jedem neuen Zeitschritt die optimalen Stellgrößen vorzugeben. Lässt man die Herausforderung der Echtzeitfähigkeit außen vor, ist dies aus Sicht der Anforderungen ein geeigneter Ansatz, um ein gewünschtes Verhalten in komplexen Systemen zu erhalten. Allerdings sollte die konkrete Auswahl des modellprädiktiven Reglers und die Auslegung mit genügend Expertise erfolgen, um z.B. unlösbare Optimierungsprobleme zu vermeiden. Bild 5-3 zeigt die grundsätzliche Struktur einer modellprädiktiven Regelung zur Veranschaulichung der Bestandteile.

Weitere Informationen u.a. zur Auslegung finden sich in der Literatur [Ada18]. Als ebenfalls geeignet im Umgang mit nichtlinearen Systemen zeigt sich, unter Voraussetzung eines guten Modells, die Methode der exakten Zustandslinearisierung [Kno17].

Ein Forschungsfeld, welches sich derzeit intensiv mit der grundsätzlichen Steigerung der Intelligenz in technischen Systemen auseinandersetzt, ist das der datengetriebenen Verfahren, wie dem Machine Learning. Hier sei auf die Literatur und die derzeitigen Forschungsaktivitäten verwiesen [TG18]. In der Kombination mit geeigneten Ansätzen aus

der Regelungstechnik liegt hier womöglich noch Potential, das es in der Zukunft zu heben gilt.

5.2.2 Anforderungsanalyse mittels Modellbildung

Nachdem in den vorangehenden Abschnitten eine Kategorisierung entlang verschiedener Disziplinen und regelungstechnische Methoden diskutiert wurden, zielt dieser Abschnitt darauf ab, eine physikalisch motivierte Herangehensweise zur Anforderungsüberprüfung zu beschreiben. Ein wertvolles Werkzeug zur Analyse sind Modelle, wie in Abschnitt 2.4 gezeigt. In einzelnen Disziplinen sind Modellierungsansätze längst bewährt und weit verbreitet. Modelle für Simulationen über Domänengrenzen hinweg sind noch forschungslastiger und ihr Einsatz in der Industrie nicht so sehr verbreitet. Es existieren jedoch Ansätze, die eine Multidomänenmodellierung erlauben und sich daher bei interdisziplinären Fragestellungen besser eignen können. Die Grundlagen lassen sich exemplarisch anhand der Bondgraphen veranschaulichen. Modelle abstrahieren ein System, sie kategorisieren gewisse Eigenschaften. Die reduzierten Lösungselemente, die über die Größen aus Tabelle 5-1 beschrieben werden, sind ein Beispiel. Gleiche Modellierungsansätze werden auf unterschiedliche Disziplinen und dementsprechend unterschiedliche Problemstellungen angewendet. Die Gemeinsamkeit hier ist die zu Grunde liegende Mathematik. Kategorisierungen können auch durch grafische Elemente abstrahiert werden, wie z.B. in der Darstellung der SysML. Eine grafische Kategorisierung ist trotz aller Vorteile nicht so leistungsfähig, wie eine mathematisch interpretierbare Kategorisierung, welche von großer Bedeutung für Modelle ist³:

"A model based on equations generally provides much more insight into the nature of the optimum design" [PW88].

Der Grad der Information, die aus einem Modell extrahiert werden kann, steigt mit der Möglichkeit mathematischer Untersuchungen rapide an. Grafische Beschreibungssprachen bieten zwar eine Übersichtlichkeit, ihre Analyse ist jedoch beschränkt. Bondgraphen vereinen gewissermaßen eine grafische Sicht und mathematische Interpretierbarkeit. Hinzu kommt die Anwendungsmöglichkeit in verschiedenen Disziplinen. Diese geht auf die Betrachtung der Energie zurück. An dieser Stelle sei ein kurzer Exkurs in die Physik erlaubt. Emely Noether formulierte 1918 das sogenannte Noether-Theorem. Es besagt:

"Zu jeder kontinuierlichen Symmetrie eines physikalischen Systems gehört eine Erhaltungsgröße" [Noe18].

So lassen sich u.a. der Energieerhaltungssatz, der Impulserhaltungssatz und der Drehimpulserhaltungssatz erklären. Aufbauend darauf können mathematische Beschreibungen physikalischer Systeme interpretiert werden.

Bondgraphen Der Grundstein für die heutige Methodik der Modellierung mit Bondgraphen⁴ wurde von Paynter gelegt. Während seiner Zeit am *Massachussets Institute*

³Wegen der einprägsamen Formulierung wird auf eine Übersetzung verzichtet.

⁴Die hier beschriebenen Grundlagen über Bondgraphen wurden mit Unterstützung von Еккноff aufgearbeitet [EL15]

50	*	• •
Domäne	Effort	Flow
Elektrotechnik	Potential (Spannung)	elektrischer Strom
Mechanik (Translatorisch)	Kraft	Geschwindigkeit
Mechanik (Rotatorisch)	Drehmoment	Winkelgeschwindigkeit
Fluidtechnik	Druck	Volumenstrom
Chemie	chemisches Potential	molarer Fluss
Thermodynamik	Temperatur	Entropiefluss

Tabelle 5-1: Effort und Flow aus verschiedenen Disziplinen [Rod13]

of Technology arbeitete er in erster Linie in interdisziplinären Projekten. Im Jahre 1959 konnte er das Bondgraphen-System vervollständigen [Bor00] [GB07]. Die Idee hinter den Bondgraphen ist der zeitkontinuierliche Energieaustausch zwischen Teilen eines Systems. Es gilt zu jeder Zeit der Energieerhaltungssatz [Rod13] [Bor00]. Energie ist gespeicherte Arbeit und damit die Fähigkeit eines Körpers, Arbeit zu verrichten⁵. Bondgraphen bieten eine graphische Darstellungsmöglichkeit des Systems durch unterschiedliche integrierte Elemente, zwischen denen der Energieaustausch stattfindet. Der Energiefluss, die Verbindung zwischen den Teilsystemen wird durch sogenannte (Leistungs-) Bonds dargestellt. Die Leistung P beschreibt die pro Zeit geleistete Arbeit⁶. Leistung ist wie Energie eine universelle Größe und damit domänenunabhängig. Sie kann als das Produkt zweier charakteristischer Größen, effort (e) und flow (f) dargestellt werden [Bor00]. Somit gilt:

$$P = e \cdot f$$
.

Die Effort- und Flowgrößen aus den verschiedenen Disziplinen sind in Tabelle 5-1 dargestellt⁷. Durch den Zusammenhang zwischen Leistung und Effort, bzw. Flow, ergibt sich
die Darstellung der Bonds, die den Leistungsfluss zwischen den einzelnen Subsystemen
darstellen. Dabei wird Energie weder gespeichert, erzeugt oder verbraucht [Bor00]. Unter Ports versteht man im Zusammenhang mit Bondgraphen die Schnittstellen oder auch
Anschlusspunkte der jeweiligen Subsysteme. Je nach Portanzahl *n* ist dabei die Rede von
einem *n*-Port. Die Leistung wird nun mit den *generalisierten Variablen* erweitert, da in
dynamischen Prozessen die physikalischen Größen zeitabhängig sind. Die Bezeichnung *generalisierte Variablen* leitet sich hierbei aus der Domänenunabhängigkeit der Größen
Effort und Flow ab [Bor00] [Rod13]:

$$P(t) = e(t) \cdot f(t)$$
.

Im Zusammenhang mit dem Verhalten dynamischer Systeme sind die Art und Anzahl der enthaltenen Speicherelemente, in die Leistung hinein oder hinaus fließt, maßgeblich. Wird im System weder Leistung verbraucht noch erzeugt, lassen sich resultierende mathematische Modelle aufstellen. Durch Anwendung des folgenden Zusammenhangs ist die im Speicher verfügbare Leistung als Differenz aus Eingangs- und Ausgangsleistung

⁵Die Einheit der Energie ist $[J = Nm = \frac{kg \cdot m^2}{s^2}]$.

⁶Die Einheit der Leistung ist $[W = \frac{J}{a}]$.

⁷Die Gültigkeiten der Analogien sollten nicht einfach hingenommen, sondern physikalisch überprüft werden. Eine kurze Analyse dieser Thematik findet sich in der Literatur [HB99].

(Transport über die Systemgrenze) definiert. Die Bilanzgleichung zur Erstellung eines mathematischen Modells lautet somit [Rod13]:

Ein Kennzeichen der Bondgraphen ist, dass die ersten Schritte der Modellierung ohne Rechnerunterstützung stattfinden und das Modell schon auf dem Papier nützliche Informationen über die Struktur des Systems liefert. Systeme werden zwar immer komplexer, aber solche agileren Ansätze sollten nicht vernachlässigt werden. Dadurch ist es möglich, potentielle Probleme, wie zum Beispiel algebraische Schleifen oder potentiell vernachlässigbare Elemente des Modells frühzeitig zu erkennen. Wenn das Modell vervollständigt ist, ergeben sich die Gleichungen für die Simulation. Darauf aufbauend ist es möglich, aus dem Modell eine Zustandsraumdarstellung abzuleiten. Zu Beginn sind dabei nur qualitative Aussagen über die physikalische Struktur des Systems möglich, die jedoch hilfreiche Informationen liefern können [GB07]. Die Kausalitäten von solchen Modellen sollen im Folgenden beschrieben werden. Dabei wird auf verschiedene Elemente der Bondgraphen zurückgegriffen. Eine Erläuterung dieser findet sich im Anhang dieser Arbeit (vgl. Anhang A4).

Berechnungskausalitäten Für die Ermittlung der mathematischen Gleichungen und eine erste Analyse eines Bondgraphen ist es wichtig, das Konzept der Kausalität⁸ zu erläutern. In diesem Kontext existiert eine unabhängige Systemvariable und eine von den Eigenschaften der Systeme abhängige Variable. Kausalität bedeutet in der Bondgraphenmethodik deshalb, die Entscheidung, welche Leistungsgröße an einem Port die unabhängige ist. Die jeweils konjugierte Variable wird infolgedessen berechnet. Das System erhält demzufolge die Information der einen Größe und liefert die entsprechend konjugierte Größe [Bor00]. Wirkt beispielsweise eine Kraft, realisiert über eine Effortquelle, auf eine Masse, erhält das System eine Effortinformation. Dort wird die Flowgröße bestimmt und die zugehörige Information zurück an die Effortquelle gesendet. Bild 5-4 zeigt den Bondgraphen eines Einmassenschwingers. Die Effortquelle ist hier die Kraft F. Jedes reale Element ist dabei nur in der Lage, eine der Größen zu steuern und erhält die andere Information von außen. Die Richtungen der Größen sind somit entgegengesetzt [Bor00]. Die Wahl der Kausalität wird aus Übersichtlichkeitsgründen durch eine zusätzliche Information an den Bondgraphenkanten verdeutlicht. Hierzu wird ein Kausalitätsquerstrich verwendet. Dieser gibt an, in welche Richtung die Effortinformation fließt, beziehungsweise auf welcher Seite die Flowvariable berechnet wird.

Kausalität von Energiespeichern: Im Fall der Speicherelemente werden zwei Formen der Kausalität unterschieden. Bei der integralen Kausalität wird der aktuelle Zustand durch die Integration von Ursachen aus der Vergangenheit erreicht. Formaler kann dies durch den folgenden Zusammenhang ausgedrückt werden [Bor00]:

$$Gr\ddot{o}eta e (aktuell) = \int Ursache (Vergangenheit) dt$$

⁸Hiermit ist nicht die zeitliche Kausalität gemeint.

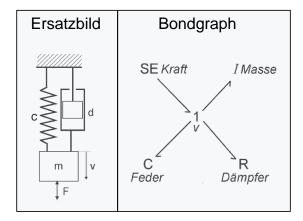


Bild 5-4: Bondgraph eines Einmassenschwingers, abgewandelt nach [Rod13]

Im Gegensatz dazu wird bei der differentiellen Kausalität der zukünftige Zustand über die Ableitung der momentanen Ursache bestimmt:

$$\frac{d(Ursache\ (aktuell))}{dt} = Gr\"{o}\beta e\ (zuk\"{u}nftig)$$

Bei der Anwendung der Bondgraphenmethodik sollte darauf geachtet werden, soweit wie möglich, integrale Kausalitäten zu verwenden. Die differentielle Kausalität macht das System von der Zukunft abhängig. In vielen Fällen kann eine differentielle Kausalität ein Indikator dafür sein, dass in einem Modell falsche Annahmen getroffen wurden.

Kausalität von I-Elementen: Im Falle der I-Elemente entspricht die Flowgröße dem integrierten Effort. Eine integrale Kausalität bedeutet somit, dass das I-Element vom System die Effortinformation erhält und die sich ergebende Flowinformation zurückliefert. Der Kausalitätsquerstrich ist entsprechend an der Bondseite, die am I-Element liegt, angeordnet. Bei einer differentiellen Kausalität wird der Querstrich im Gegenzug der anderen Bondseite zugeordnet [Bor00].

Kausalität von C-Elementen: Bei C-Elementen bildet sich der Effort aus dem integrierten Flow. Das Erhalten der Flow- und Zurückliefern der Effortinformation stellt infolgedessen die bevorzugte kausale Struktur der C-Elemente dar. Dementsprechend ergibt sich die den I-Elementen entgegengesetzte Anordnung der Kausalitätsquerstriche [Bor00].

Kausalität von Energieverbrauchern: Beim R-Element liegen keine zeitlichen Präferenzen zwischen den Größen vor. Demzufolge lässt sich keine allgemeine Regel ableiten. Die Eingangsgröße kann sowohl in Form von Flow, als auch in Form von Effort vorliegen. Als Ausgangsgröße ergibt sich die jeweilige konjugierte Größe [Bor00].

Kausalität von Transformer und Gyrator: Die Effort- und Flowgrößen der Ein- und Ausgangsseite stehen bei einem Transformer in einem bestimmten Verhältnis. Es ergeben sich zwei mögliche Kausalstrukturen, da nicht festgelegt ist, auf welcher Seite des Transformers die Ein- oder Ausgangsseite liegt. Der Kausalquerstrich ist jeweils auf der gleichen Seite der Bonds angeordnet [Bor00]. Beim Gyrator stehen die Eingangsgrößen in einem bestimmten Verhältnis zu den konjugierten Größen am Ausgang. Infolgedessen ergibt sich eine entgegengesetzte Anordnung der Querstriche [Bor00].

Kausalität von Junctions: Im Falle der 0-Junctions erhält ein Bond den Kausalitätsquerstrich an der zur Junction liegenden Seite. Dieser Bond führt somit die Effortinformation

zu. Da im Falle der 0-Junction die Efforts identisch sind, führen die restlichen Bonds die Effortinformation ab. Demnach sind die Kausalitätsquerstriche der Bonds am wegführenden Ende angeordnet [Bor00]. Durch die identischen Flowgrößen einer 1-Junction ergibt sich für die Kausalitätsquerstriche das Gegenteil gegenüber der 0-Junction. Hier führt ein Bond die Flowinformation zu und erhält somit einen Querstrich an der von der Junction abgewandten Seite. Die Bonds, welche die Flowinformation abführen, erhalten dementsprechend einen Querstrich am Bondende, das der Junction zugeordnet ist. Bei der Kausalität der Junctions besteht im Gegensatz zu den anderen Elementen keine Alternativmöglichkeit. Die Struktur ist somit verpflichtend einzuhalten [Bor00] [Rod13].

Fazit Der Vorteil einer solchen Modellierungssprache sind die Ähnlichkeiten zur Wirkstruktur. Es lässt sich also ein simulierbares und analysierbares Modell analog zum Systemmodell erstellen. Diese Eigenschaft sorgt ebenfalls dafür, dass schwer zu identifizierende Einflüsse, wie Feldgrößen⁹, also z.B. der Schall oder die Strahlung, sichtbar werden können. Die Modelle können systematisch und vollständig¹⁰ auf solche Einflüsse untersucht werden. Rein gleichungsbasierte Modellierung erschwert diese Analyse, da Wechselwirkungen ohne direkte Systemfunktion in vielen Fällen nur durch Expertenwissen berücksichtigt werden. Gesamtsystemmodelle erleichtern diese Analyse. Insgesamt hält diese Modellierung detailliertere Analysemöglichkeiten bereit, als eine Wirkstruktur. Die Bondgraphen und die Wirkstruktur sollten symbiotisch genutzt werden, sodass sie ihre jeweiligen Stärken für die Identifikation konkreter domänenübergreifender Anforderungen ausspielen können.

Strukturmaße Die fortgeschrittene Analyse von Modellen kann auch Rückschlüsse auf die initiale Wichtigkeit von Anforderungen liefern. Dies soll am Beispiel von Strukturmaßen erläutert werden. Strukturmaße in der Regelungstechnik werden zur Quantifizierung von Struktureigenschaften angewendet. Dies geschieht aus drei Gründen [Rie98]:

- Die Ordnung von Systemen sollte so gering wie möglich sein.
- Die Erkenntnis, welche Systemteile wie gut beeinflusst werden können.
- Die Ermittlung der Störempfindlichkeit verschiedener Teile komplexer Systeme.

Diese Gründe weisen gewisse Analogien zu der Beurteilung von Anforderungen auf. Die Fragen, welche Systemelemente wichtig sind und welche Beeinflussungen stattfinden können, helfen bei der Bewertung von Anforderungen. Bei der Modellbildung von Systemen werden typischerweise implizit Effekte vernachlässigt. Reibung beispielsweise wird oft nicht modelliert. Das Expertenwissen bzgl. des Systems und der Modellbildung sind die wesentlichen Kriterien für geeignete Vereinfachungen. Systematischer ist dieses Thema im Bereich der Ordnungsreduktionsverfahren aufgearbeitet [Har02]. Bild 5-5 zeigt eine vorgeschlagene Klassifizierung von Ordnungsreduktionsverfahren. Es sind jeweils Kategorien gebildet bis sich konkrete Verfahren zuordnen lassen. Harrer weist darauf hin, dass die eigenwertspezifischen Verfahren prinzipiell den physikalischen Verfahren zuzuordnen sind, wegen ihrer großen Bedeutung jedoch gesondert betrachtet werden.

⁹Hierfür sollten agf. Multiportfelder (vgl. [Rod13]) eingesetzt werden.

¹⁰Vollständig, gemessen an der Wirkstruktur

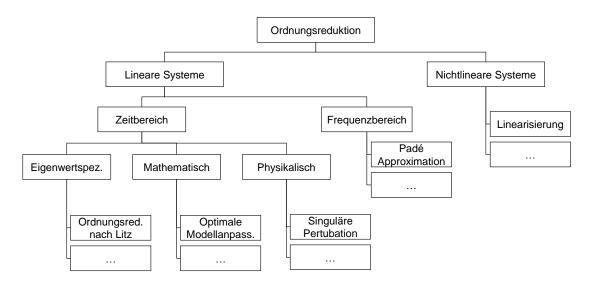


Bild 5-5: Auszug aus der Klassifizierung von Ordnungsreduktionsverfahren nach Harrer [Har02]

Es zeigt sich jedenfalls, dass unter gewissen Annahmen eine Reduzierung der Modellordnung streng systematisch erfolgen kann. Weitere Anwendungen der Strukturmaße mit
konkretem Bezug auf Bondgraphen finden sich in der Literatur [Orb10]. Dort werden speziell leistungs- und energiebasierte Methoden zur Ordnungsreduktion thematisiert. Diese physikalischen Herangehensweisen erlauben eine Bewertung von Systemen und ihren
Elementen. Diese kann mit einer initial vorgenommenen Bewertung von Anforderungen
verglichen werden. Sollen Anforderungen zu einem Zeitpunkt der Entwicklung analysiert
werden, in welchem noch keine physikalisch-mathematischen Modelle vorliegen, bedarf
es anderer Analyseformen. Auch bei grundlegenden Analysen zur Verknüpfung und Priorisierung von Anforderungen werden zusätzliche Verfahren benötigt. Diese weisen einen
anderen Abstraktionsgrad auf und werden im folgenden Abschnitt erläutert.

5.3 Anforderungsindikator: N3 Matrix

Typische Key Performance Indikatoren (KPI) im Bereich der Anforderungen sind auf Fertigstellungsgrade und Vollständigkeit ausgerichtet. So ist beispielsweise der Freigabegrad als

$$ARoR = \frac{NoACR}{NoR} \times 100\%$$

definiert¹¹, also als der Anteil an freigegebenen widerspruchsfreien Anforderungen an der Gesamtanzahl an Anforderungen. Der Abdeckungsgrad durch Abnahmekriterien wird zu

$$DoCoAC = \frac{NoDAC}{NoR} \times 100\%$$

¹¹ARoR = Approval Rate of Requirements (Freigabegrad von Anforderungen)

NoACR = Number of Approved Consistent Requirements (Anzahl aller freigegebener widerspruchsfreier Anforderungen)

NoR = Number of Requirements (Anzahl der Anforderungen)

berechnet¹² [DDN⁺13]. Diese KPI sind sehr einfach zu berechnen, bieten einen Überblick über den Prozess und haben ihre Berechtigung zur Bewertung der Prozessperformance. Sie bieten jedoch eher wenig Analysetiefe und eignen sich daher nur bedingt für detailliertere Analysen. Es wird ein formaler Ansatz gesucht, um die Zusammenhänge von Anforderungen auf einem weiter reichenden und tiefer gehenden Detaillierungsgrad aufzudecken.

Metrischer Raum

Grundsätzlich sollte bei der Frage, welche Zusammenhänge zwischen Anforderungen, Systemelementen und Funktionen bestehen, ein möglichst formaler Ansatz gefunden werden. Wird die Verbindung zwischen den drei genannten Elementen im weitesten Sinne etwa als Abstand betrachtet, so stellt die Mathematik die sogenannten *metrischen Räume* zur Verfügung. Der Gedanke ist, dass in einem gewissen Raum eine gedachte Verbindung zwischen den drei Elementen umso intensiver ist, je mehr diese Elemente miteinander in Berührung kommen. Dafür wird an dieser Stelle fiktiv angenommen, dass es sich um einen linearen Raum handelt. Welche Eigenschaften muss nun ein Abstand haben, der die Verbindung dieser drei Elemente charakterisieren soll? Zuerst müssen Anforderungen, Systemelemente und Funktionen in diesem gedachten Raum vorliegen und dieser Raum sollte gewisse Eigenschaften mit sich bringen. So sollte gelten (entsprechend [Heu06]):

- Der Abstand von x zu x ist 0.
- Der Abstand von x zu y ist gleich dem Abstand von y zu x (Symmetrie).
- Die gedachte Verbindung über einen Umweg ist nicht kürzer als die direkte Verbindung.

Eine Metrik besitzt dann folgende Eigenschaften:

- d(x, y) >= 0 mit d(x, y) = 0, bei x = y,
- d(x,y) = d(y,x),
- $d(x, y) \le d(x, z) + d(z, y)$.

Mit der Einschränkung, dass es Elemente geben kann, deren Abstand gleich 0 ist, die jedoch verschieden sind, wird von einer Pseudometrik gesprochen. In der Statistik wird die Frage nach Zusammenhängen durch Zusammenhangsmaße quantifiziert, welche auf einer Skala abgebildet werden können. Die Aussagefähigkeit einer solchen Skala wird durch das Skalenniveau gekennzeichnet. Bei *Nominal*- und *Ordinalskalen* wird von kategorialen Skalen gesprochen, bei den sogennanten *Kardinalskalen* von metrischen Skalen. Eine exakte Einordnung ist nicht immer möglich, jedoch ist das Ziel, ein Zusammenhangsmaß von Anforderungen, Funktionen und Systemelementen kardinal zu skalieren. Unter diesen Randbedingungen kann nun ein Vorgehen gesucht werden, welches auf formalen Kriterien beruht und auf die Problematik angepasst ist. Dabei werden die Eigenschaften als Suchgrößen genutzt und nicht als feststehende, unbedingt zu erfüllende Kriterien. Es wird ein *möglichst* formaler Ansatz gesucht.

¹²DoCoAC = Degree of Coverage of Acceptance Criteria (Abdeckungsgrad durch Abnahmekriterien)
NoDAC = Number of Defined Acceptance Criteria (Anzahl der definierten Abnahmekriterien)

Anmerkung: Metriken werden nicht nur als Bezeichnung mathematischer Gebilde genutzt. In der Softwaretechnik hat sich dieser Begriff so etabliert, eine Messung von Software zu bedeuten. Es wird zwischen

- Quantitätsmetrik,
- Komplexitätsmetrik,
- Qualitätsmetrik

unterschieden [SSB10]. Es können dementsprechend Mengen, Verhältnisse und Beziehungen sowie die Güte von Software mithilfe von Metriken erfasst werden. In Methoden wie CMMI (vgl. Abschnitt 3.4) wird die kontinuierliche Überprüfung von Software mit Metriken bereits gefordert.

Ein Vorgehen zur Analyse technischer Anforderungen

Das Ziel ist es nun, ein Werkzeug zur Hand zu haben, um die Zusammenhänge der Anforderungen mit einem System zu identifizieren. Angesichts der gesetzten Anforderung an eine Systematik, einen kardinal skalierten Wert zu ermitteln, sollten die ermittelten Zusammenhänge nicht lediglich gekennzeichnet sein, sondern entsprechend ihrer Bedeutung gewichtet sein. Für eine solche formale Analyse müssen die Eigenschaften des geometrischen Abstandes aus dem vorherigen Abschnitt nun gedanklich auf einen mehr oder weniger *engen* Zusammenhang abstrahiert werden. Der Zusammenhang zwischen den Anforderungen, Funktionen und Systemelementen wird also gesucht. Es ist vorstellbar, dass diese drei Entitäten auf drei Achsen in unterschiedlichen Raumrichtungen einen Raum aufspannen (Bild 5-6). Der Einfachheit halber sei jede Richtung jeweils orthogonal zu den anderen beiden, sodass von einem Quader ausgegangen werden kann. Dies ist ein diskret unterteilter Raum, der Einträge zulässt, welche die drei Entitäten in Beziehung setzen.

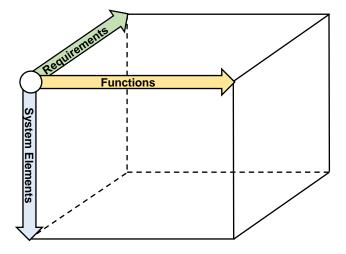


Bild 5-6: N3 Cube

An dieser Stelle wird nun eine Unterscheidung vorgenommen. Anfänglich wird angenommen, sobald eine Verbindung zwischen allen dreien bestehe, stehe an der entsprechenden Stelle ein Eintrag mit dem Wert eins. Ebenfalls unabhängig vom Grad der Verbindung

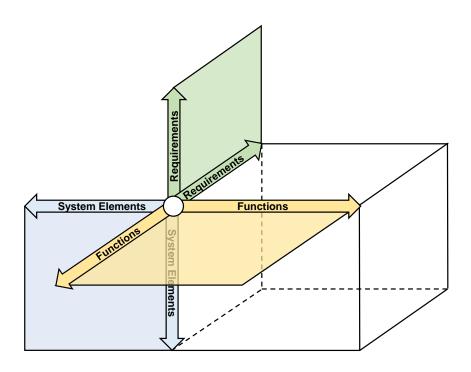


Bild 5-7: Die drei Ebenen für Beziehungen der gleichen Entitäten im N3 Diagramm

seien auf den Würfelseiten die Einträge an den Stellen 1, an denen die entsprechenden zwei Entitäten eine Verbindung haben. In Zukunft seien gewichtete Einträge an den jeweiligen Stellen zugelassen (vgl. Abschnitt 7.2), doch vorerst wird auf diese Eigenschaft verzichtet.

Nun wird angenommen, die jeweiligen Achsen, erlauben eine Spiegelung (Bild 5-7) und die so aufgespannten Ebenen haben die gleichen Einträge an den gleichen Stellen. Die Informationen bleiben also vollständig erhalten. Hieraus folgen erhebliche Vereinfachungen im mathematischen Umgang mit den Informationen des dreidimensionalen N3-Cubes. Die Inhalte lassen sich nun als eine zweidimensionale Matrixdarstellung entsprechend Bild 5-8 interpretieren. Dazu werden lediglich die Ebenen mit den entsprechenden Informationen der Verknüpfung anders angeordnet, ohne jedoch ihre primäre Bedeutung zu verlieren. Es bleibt jedoch festzuhalten, dass die Zusammenhänge, die sich dadurch kennzeichnen lassen, sowohl eine Funktion, als auch ein Systemelement in Relation zu setzen, nicht mehr offensichtlich sind. Dennoch bietet diese Darstellung nun Analysemöglichkeiten. Der symmetrischen zweidimensionalen Matrix fehlen noch Einträge auf den diagonalen Feldern. Offensichtlich sind hierfür Informationen über die Verbindungen innerhalb einer jeden Entität notwendig. Die zusätzlichen Flächen enthalten nun genau die benötigten Informationen, um in der zweidimensionalen Ansicht die N3¹³ Matrix zu vervollständigen.

¹³Der Name N3 leitet sich von der N2 Darstellung ab, in welcher die beiden Dimensionen *Funktionen* und *Systemelemente* gegenübergestellt werden [Nat07].

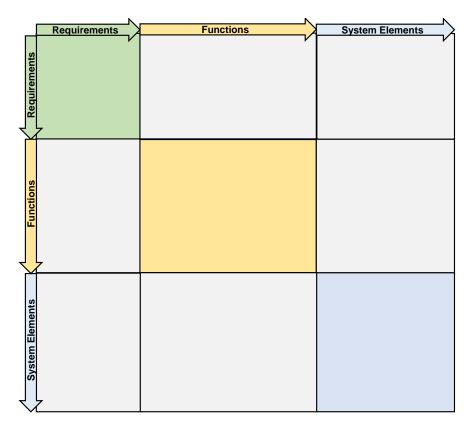


Bild 5-8: Schematische Darstellung der N3 Matrix

5.3.1 λ -Rank

Die N3 Matrix verbindet also Anforderungen mit Funktionen und Systemelementen und all diese auch untereinander. Die Ermittlung eines Index hieraus wird anhand eines Beispiels erläutert. Es wird angenommen, es gäbe ein Systemelement (*Greifer*), welches mit einer Anforderung (F>20N) und einer Funktion (*Kraft kontinuierlich aufbringen*) verbunden sei. Dies lässt sich im N3 Cube gemäß Bild 5-9 darstellen.

Jede Komponente¹⁴ sollte mit mindestens einer anderen verbunden sein, kann jedoch mit mehreren verbunden sein. Komponenten, welche sehr viele Verbindungen haben sind tendenziell von größerer Bedeutung – unabhängig davon, ob es sich um ein Systemelement, eine Funktion oder eine Anforderung handelt. Stark vernetzte Komponenten innerhalb des N3 Cubes haben einen großen Einfluss auf das System. Die offensichtliche Frage an dieser Stelle ist, wie diese verschiedenen Verbindungen analysiert werden können. Der erste Schritt ist, wie bereits dargelegt, die Überführung in eine mathematisch leichter zu interpretierende Struktur. Wenn Anforderungen, Funktionen und Systemelemente unterschieden werden und ihre Verbindungen als ein Graph dargestellt werden, wie in Bild 5-10 gezeigt.

Der Graph ist ungerichtet und die Extraktion der Adjazenzmatrix stellt kein Problem dar. Anforderungen, Funktionen und Systemelemente sind dabei nebeneinander angeordnet. Der Eintrag ist 1 wenn eine Verbindung vorliegt und andernfalls ist der Eintrag 0. Es sollte

¹⁴Komponente im mathematischen Sinne: Hier stellvertretend für jede Anforderung, jede Funktion und jedes Systemelement.

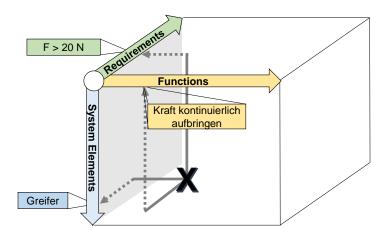


Bild 5-9: Schematische Darstellung des N3 Cubes

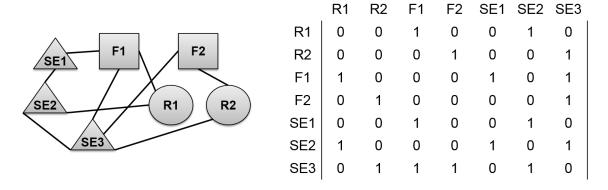


Bild 5-10: Graph des N3 Cubes mit Adjazenzmatrix als Beispiel

	Eigenwerte	$M_{\lambda_{max}}$ Eigenvektor zum Eigenwert 3.7321
R1	-1.5616	0.3053
R2	0.0000	0.3053
F1	0.2679	0.4170
F2	1.0000	0.3053
SE1	1.0000	0.3053
SE2	2.5616	0.4170
SE3	3.7321	0.5287

Tabelle 5-2: M_{λ} *Analyse der N3 Matrix*

angemerkt werden, dass die Elemente auf der Diagonalen für die Berechnung gleich 1 sein sollten. Dies stellt eine Einschränkung bezogen auf die Zielsetzung dar, entsprechend einer Metrik vorzugehen. Die Berechnung erfolgt also streng genommen auf Basis der Addition aus der *N3 Matrix* und der Einheitsmatrix.

Dies erlaubt jedoch die Verwendung einiger Matrix Charakteristika zur Analyse. Da die N3 Matrix immer symmetrisch ist, ist sie diagonalisierbar. Es existiert eine invertierbare Matrix M und eine diagonale Matrix $N3^*$, sodass gilt

$$M^{-1}N3M = N3^* = diag[\lambda_1, \lambda_2, ..., \lambda_n].$$

Die Eigenvektoren sind diejenigen Vektoren, die bei Multiplikation mit dem jeweiligen Eigenwert den Vektor ergeben, welcher Ursprünglich mit N3 multipliziert wurde. M enthält die Eigenvektoren und $N3^*$ enthält die zugehörigen Eigenwerte auf der Diagonalen. Der Eigenvektor M_{λ} , welcher zu dem größten Eigenwert gehört, kann für eine weitere Analyse herangezogen werden¹⁵. Die Werte von $M_{\lambda_{max}}$ korrelieren mit der Anzahl der Verbindungen innerhalb der N3 Adjazenzmatrix (vgl. Bild 5-10)¹⁶. Diese Werte in Tabelle 5-2 repräsentieren die relative Anzahl der Verbindungen zwischen Systemelementen, Funktionen und Anforderungen. Es wird deutlich, dass der Eintrag für Systemelement SE3 den größten Wert hat (0.5287). Durch einfaches Nachzählen der Verbindungen des Graphen (vgl. Bild 5-10) bestätigen sich die Berechnungsresultate aus M_{λ} . Innerhalb eines Systems hilft dieser Wert, Komponenten mit hoher Bedeutung zu identifizieren. Darüber hinaus werden mögliche Vernachlässigungen deutlich. Potentiell unwichtige Elemente mit einem hohen M_{λ} Wert sollten einer weiteren Betrachtung unterzogen werden.

Die Erkenntnisse, dieser Betrachtung beziehen sich jeweils auf ein konkretes System und M_{λ} kann nicht über Systeme hinweg verglichen werden. Dieser Wert kann aber in Relation zu der gesamten Anzahl der Verbindungen innerhalb des Systems gesetzt werden. Eine Möglichkeit zur Erlangung von Informationen zur Dichte der Verbindungen für das gesamte System ist die Anwendung der Frobenius Norm¹⁷. In dem hier aufgeführten Beispiel (vgl. Bild 5-10¹⁸) ist

¹⁵Eine Erläuterung und mathematische Herleitung dieser Aussage kann in der Literatur zum *Analytical Hierarchy Process* nachgelesen werden [Saa90].

¹⁶Die N3 Matrix wird aus der Addition der N3 Adjazenzmatrix mit der Einheitsmatrix gebildet.

¹⁷Die Frobenius Norm ist keine natürliche Matrixnorm.

¹⁸Es ist zu beachten, dass N3 Einträge auf der Hauptdiagonalen enthält.

$$||N3||_F = \sqrt{(\Sigma\Sigma(N3_{ij})^2)} = \sqrt{(1^2 + 0^2 + \dots + 1^2)} = \sqrt{25} = 5.$$

Für diesen Ansatz, der eine systemübergreifende Betrachtung erlaubt, muss $||N3||_F$ auch in Relation zur Systemgröße gebracht werden. Dies kann mittels Division durch die Anzahl aller Systemelemente, Funktionen und Anforderungen erreicht werden. Die Dichte der Verbindungen, oder in englischer Sprache die *Density of Connections (DoC)* ist dementsprechend

$$DoC = ||N3||_F/(\Sigma SE + \Sigma F + \Sigma R).$$

In diesem Fall ist *DoC* gleich 5/7. Je näher dieser Wert gegen 1 strebt, desto höher ist die Dichte der Verbindungen. Damit ist eine Grundlage gefunden, um die Zusammenhänge von Anforderungen mit dem System systematisch zu quantifizieren.

5.3.2 N3 Singulärwertzerlegung

Wie im vorherigen Abschnitt beschrieben, gehen bei einer zweidimensionalen Betrachtung des dreidimensionalen Zusammenhangs einige Informationen verloren. Die Anwendung der Singulärwertzerlegung bringt gegenüber des λ -Ranks Vorteile bei der Auswertung von inneren Zusammenhängen mit sich. Bei der Auswertung mittels Eigenwerten gehen Informationen verloren, die gegenseitige Beeinflussung ist nicht mehr vorhanden. Wird stattdessen die Singulärwertzerlegung angewendet, so ist nicht nur die relative Wichtigkeit von Elementen bezogen auf das gesamte System deutlich, sondern auch die Zusammenhänge zwischen Anforderungen und Funktionen sowie zwischen Anforderungen und Systemelementen und ebenso zwischen Funktionen und Systemelementen und jeweils untereinander. Damit sind weitergehende Rückschlüsse möglich. Zur mathematischen Betrachtung sei festgestellt, dass jede Matrix eine Singulärwertzerlegung besitzt. Für N3 gilt dementsprechend

$$N3 = U\Sigma V^T$$
$$U(m \times m),$$

 $\Sigma(m \times n),$ $V(n \times n).$

U und V sind die Links- bzw. Rechts-Singulärvektoren und Σ enthält die Singulärwerte. Die Anzahl der Zeilen und Spalten von N3 wird durch m bzw. n beschrieben. Es muss sich jedoch nicht zwangsläufig um die Zerlegung von N3 handeln. Vielmehr interessieren Teile von N3. Somit kennzeichnen m und n die Anzahl der betrachteten Komponenten von N3. Die Funktionen in Relation zu den Systemelementen oder die Anforderungen in Relation zu den Funktionen bergen bedeutende Informationen. Eine zu extrahierende Information sind Gruppen von Anforderungen. Diese Gruppen zeigen zusammengehörige Anforderungen, die beispielsweise alle einer Funktion zugeordnet sind. Dafür muss

mit

die klassische Singulärwertzerlegung etwas angepasst werden und zwar abhängig vom Rang (rng) der Matrix. Die Zusammenhänge mit einer *Dimensionsreduktion* werden so offensichtlich [RLU14]. Die neuen Eigenschaften der Matrizen und Vektoren sollen im Folgenden gelten und lauten:

 $U(m \times rng)$,

 $\Sigma(rng \times rng)$,

 $V(n \times rng)$.

Es gibt eine Anzahl an Anforderungen und eine Anzahl an Funktionen. Ist eine Funktion nun mit einer Anforderung verbunden, wird ein positiver Eintrag gesetzt – analog zum Vorgehen beim λ -Rank. Liegt diese Matrix vor, kann sie zerlegt werden. Daraus ergibt sich die Möglichkeit, weitergehende Zusammenhänge aufzudecken. So können Funktionen miteinander zusammenhängen oder auch Anforderungen. Sind beispielsweise mehrere Anforderungen mit der gleichen Funktion verknüpft, besteht offensichtlich ein Zusammenhang. Sind diese Anforderungen jedoch auch mit weiteren Funktionen verknüpft, bleibt dies bei großen Datensätzen möglicherweise verborgen. An genau dieser Stelle kann die Singulärwertzerlegung unterstützen, solche Zusammenhänge aufzudecken. So können gewisse Funktionen etwa der Sicherheit dienen und entsprechend auch gewisse Anforderungen explizit für die Sicherheit spezifiziert sein. Die Singulärwertzerlegung erlaubt es, solche Gruppen zu identifizieren.

Die weiterführende Analyse bleibt anschließend Aufgabe der Entwicklung. Eine Quantifizierung erfolgt jedoch anhand dieser Methode. Die Spalten von U quantifizieren die Stärke der Zugehörigkeit zu einer solchen Gruppe. Die betragsmäßige Größe der Singulärwerte zeigt, wie deutlich eine Gruppe ausgeprägt ist. Sind die Singulärwerte zu klein, sollten diese ggf. nicht als Gruppe identifiziert werden. Die Zeilen von V entsprechen den Zugehörigkeiten der Funktionen zu den Gruppen

- *U*: Anforderungen zu Gruppen,
- V Funktionen zu Gruppen,
- Σ Stärke der Gruppen.

Die Möglichkeit, die sich daran ableitet, ist, dass ggf. Anforderungen gefunden werden können, die einer Gruppe zugeordnet sein sollten, es jedoch noch nicht sind. Das ist der Fall, wenn z.B. zwei Anforderungen völlig unterschiedlichen Funktionen zugeordnet sind, beide jedoch der Gruppe Sicherheit zugeordnet sein sollten. In der ursprünglichen Betrachtung in *N*3 bleiben solche Zusammenhänge verborgen. Werden die Zeilen der Anforderungen dann aber mit der Matrix *V* multipliziert, eröffnen sich doch Ähnlichkeiten, falls sie vorhanden sind. Als Beispiel sei ein System mit 11 Anforderungen und 5

Funktionen gegeben. $N3_{RF}^{19}$ sei

Diese Teilmatrix aus N3 beschreibt die Zusammenhänge zwischen Anforderungen $(R)^{20}$ und Funktionen (F). Der Rang dieser Matrix ist $rang(N3_{RF}) = 4$. Somit ist U dann

[-0.3909]	0.1422	0.0182	-0.1309
-0.2679	0.1018	-0.4438	0.8491
-0.3909	0.1422	0.0182	-0.1309
-0.0737	-0.5052	-0.0057	0.0344
-0.0737	-0.5052	-0.0057	0.0344
-0.3909	0.1422	0.0182	-0.1309
-0.4646	-0.3630	0.0125	-0.0965
-0.3909	0.1422	0.0182	-0.1309
-0.0737	-0.5052	-0.0057	0.0344
-0.1304	0.0478	-0.6239	-0.3729
-0.2605	0.0944	0.6421	0.2421

 Σ ist

$$\begin{bmatrix} 4.3131 & 0 & 0 & 0 \\ 0 & 2.7271 & 0 & 0 \\ 0 & 0 & 1.2548 & 0 \\ 0 & 0 & 0 & 0.6210 \end{bmatrix}$$

und V ist

$$\begin{bmatrix} -0.5626 & -0.5306 & -0.5927 & -0.1590 & -0.1590 \\ 0.1303 & 0.1101 & 0.1474 & -0.6889 & -0.6889 \\ -0.7829 & 0.5797 & 0.2260 & -0.0036 & -0.0036 \\ -0.2316 & -0.0685 & 7588 & 0.0107 & 0.7071 \end{bmatrix}.$$

Es wird anhand der Singulärwerte deutlich, dass zwei Gruppen von größerer Bedeutung sind (4,3131 und 2,7271). Anhand von *U* kann ermittelt werden, dass die Anforderungen

¹⁹Die Indizes weisen darauf hin, welcher Teil der N3 Matrix betrachtet wird. In diesem Fall die beiden Dimensionen: *Requirements* und *Functions*.

²⁰R für Requirements

1, 2, 3, 6, 7, 8, 10 und 11 zur ersten Gruppe gezählt werden können und Anforderungen 4, 5 und 9 zur zweiten Gruppe. Dies kann anhand der ersten bzw. zweiten Spalte abgelesen werden. Die kleinen Singulärwerte der anderen Gruppen in Σ erlauben es, diese Gruppen zu vernachlässigen.

Aus V wird wiederum deutlich, dass die Funktionen 1, 2 und 3 zur ersten Gruppe zugeordnet sind und die Funktionen 4 und 5 zur zweiten. Dies lässt sich anhand der ersten
beiden Zeilen zuordnen. Auch zeigt sich, dass Anforderung 2, 10 und 11 in $N3_{RF}$ nicht
mit den Funktionen ihrer vermeintlich zugehörigen Gruppe verbunden sind. Hier wäre eine Prüfung der Anforderungen sinnvoll, um die Zuordnungen ggf. zu korrigieren. Die N3
Singulärwertzerlegung kann demnach als weiterführende Analyse eingesetzt werden, um
detailliertere Erkenntnisse über die Zusammenhänge von Anforderungen mit dem System
zu erhalten. Dies hilft auf einer struktureller Ebene die Anforderungen zu bewerten. Es
werden demnach nicht die Eigenschaften der einzelnen Anforderungen betrachtet, sondern der Zusammenhang dieser mit dem System.

Zusammenfassung Dieses Kapitel behandelte drei übergeordnete Themengebiete. Zu Beginn wurde eine Problemunterscheidung vorgenommen, um die hier beschriebenen Teile der Systematik T-REQS zielgerichtet einzusetzen. Anschließend wurden Verfahren und Werkzeuge, etwa aus der Regelungstechnik und der Modellbildung beschrieben, um die Analyse von Anforderungen zu unterstützen. Abschließend wurden zwei Anforderungsindikatoren, der λ-Rank und die N3 Singulärwertzerlegung vorgestellt, um über die Verbindungen der Anforderungen mit dem System Rückschlüsse zu erlauben. Standen in Abschnitt 5.2 die modellbasierte Analyse der Anforderungen im Fokus, geben die in Abschnitt 5.3 vorgestellten Indikatoren Einblicke in strukturelle Eigenschaften der Anforderungen mit dem System. Im folgenden Kapitel werden die Ergebnisse der in Kapitel 4 und 5 beschriebenen Systematik an einem Beispiel angewendet.

6 Anwendung und Bewertung

Im Sinne eines ganzheitlichen Ansatzes wird in diesem Kapitel ein Beispiel mit einem Bezug zu allen Entwicklungsschritten genutzt und die Evolution der Anforderungen beleuchtet. Diese Anwendung verdeutlicht den Nutzen der Systematik und zeigt die Anwendbarkeit in einem industriell geprägten Umfeld. Es wird sowohl auf akademische Beispiele und an geeigneten Stellen auf industrielle Systeme zurückgegriffen.

Das hier verwendete Beispiel stammt aus einem Forschungsprojekt, welches im Rahmen des Spitzenclusters *it's OWL intelligente technische Systeme OstWestfalenLippe* durchgeführt wurde. Mehrere Forschungspartner erarbeiteten gemeinsam mit der Firma Wincor Nixdorf International GmbH im Projekt *it's OWL InverSa - Intelligente vernetzte Systeme für automatisierte Geldkreisläufe* Konzepte und Lösungen für moderne Geldautomaten. Zu Beginn wird die Problematik beschrieben, mit der sich herkömmliche Geldautomaten auseinandergesetzt sehen. Anschließend wird eine Lösungsidee beschrieben. Die Umsetzung dieser zu einer technischen Lösung wird an einzelnen Teilsystemen gezeigt. Hierbei wird anhand der in Kapitel 4 beschriebenen Systematik und in Kapitel 5 vorgestellten Analyseverfahren vorgegangen.

6.1 Probleme und Ziele eines robusten Bargeldhandlings

Die meisten mechatronischen Produkte werden umso besser funktionieren, je exakter die Sensorik und je leistungsfähiger die Aktuatorik ist. Dies führt zu erhöhter Performance, was am Beispiel der Geldautomatenentwicklung gesehen werden kann. Bild 6-1 zeigt einen Geldautomaten aus dem Jahre 1978. Ein moderner Geldautomat ist in Bild 6-2 dargestellt. Diese Entwicklung wird im Spannungsfeld aus Qualität, Zeit und Kosten durch den Kostendruck begrenzt. Nicht erst seit der Banken- bzw. Finanzkrise im Jahre 2008 gilt dies insbesondere für das Handling von Bargeld. Zwei unterschiedliche Strategien zur Lösung dieses Problems können beobachtet werden. Einerseits werden Produkte auf dem Markt etabliert, welche immer günstigere Komponenten verwenden, um Basisfunktionen anzubieten. Andererseits werden teurere Produkte entwickelt, welche einen



Bild 6-1: Ein Geldautomat aus dem Jahr 1978. Ein Bildschirm ist zu dieser Zeit nicht vorhanden [Hei17].



Bild 6-2: Ein moderner Geldautomat von der Seite mit Blick ins Innere [Hei17].

– im Vergleich zu herkömmlichen Geldautomaten – wesentlich höheren Funktionsumfang bieten. Gerade der letzte Ansatz verlangt nach einem systematischen Vorgehen, da das Produkt naturgemäß komplexer wird und deutlich mehr innere und äußere Interaktionen beinhaltet. Gleichzeitig erfordert dieser Ansatz auch technische Innovationen, um die Prozesssicherheit zu steigern, einen flächendeckenden Einsatz zu gewährleisten und die Automatisierung der Geldkreisläufe zu fördern.

Als besondere technische Herausforderung zeigt sich dabei das *unfitte* Bargeld. Dies sind Banknoten mit erhöhter Verschmutzung, mechanischen Beschädigungen und Verschleißerscheinungen. Ebenfalls sind die unterschiedlichen Eigenschaften von Banknoten zu beachten. Es treten Unterschiede durch Denominationen und unterschiedliche Währungen auf. Dies betrifft sowohl die Geometrie als auch die Oberflächeneigenschaften. Zusätzlich erschweren Manipulationsversuche am Geldautomaten einen sicheren Prozess und führen zu hohen Schäden. Etwa 90 Prozent aller Bezahltransaktionen weltweit werden mittels Bargeld getätigt. Es fallen dafür Kosten in Höhe von 300 Mrd. € an – davon 50 Mrd. € in Europa. Die Anzahl der sich im Umlauf befindenden Euro-Noten steigt pro Jahr um etwa 9 Prozent¹. Trotz der voranschreitenden Entwicklung zum bargeldlosen Zahlen zeigen diese Daten eine zur Zeit beobachtbare Relevanz für Innovationen. Die Prozesse und Aktionen, die diese Kosten mitverursachen sind bedingt durch Abrechnungen bei jedem Verantwortungsübergang des Bargeldes. Oft wird nach dem Vieraugenprinzip das Geld gezählt. Die Automatisierung dieser Prozesse ist abhängig vom fehlerfreien Zustand der Banknoten. Werden wegen mechanischer Verarbeitungsfehler einzelne Banknoten nicht erkannt, entsteht sofort ein lokaler Schaden. Gleichzeitig ergibt sich hierdurch eine zu-

¹Diese Informationen entstammen einer Aussage eines Industrieunternehmens im Jahre 2012.

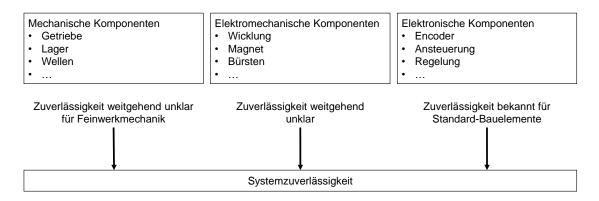


Bild 6-3: Auszug der Komponenten von Feinwerksystemen und deren Einfluss auf die Zuverlässigkeit, abgewandelt nach [BGJ09]

nehmende Unsicherheit und Unzuverlässigkeit.

Es werden robuste und intelligente mechatronische Lösungen benötigt, die die Varianz der Banknoteneigenschaften berücksichtigen. Dies kann nicht durch individuelle Einstellungen erreicht werden, die Vielfalt der Einstellparameter wäre an herkömmlichen Geräten zu hoch. Stattdessen muss das System an sich robuster gegen Parameterschwankungen und externe Einflüsse werden. Erschwert wird dieses Ziel durch die hohen Kundenanforderungen, die z.B. Verarbeitungsgeschwindigkeiten von etwa zehn Banknoten pro Sekunde fordern. Dies stellt hohe Ansprüche an die verschiedenen Komponenten, welche dennoch robust funktionieren müssen. Gerade im Bereich der in Geldautomaten verbreiteten Feinwerktechnik gibt es dabei viele Herausforderungen. Eine dieser Herausforderungen ist die Zuverlässigkeit (vgl. Bild 6-3). Es existieren praktisch nur wenige gesicherte Erkenntnisse, auf denen aufgebaut werden kann. Eine systematische Herangehensweise ist daher umso wichtiger.

Die Entwicklung eines intelligenten Automaten zur flexiblen Handhabung von Bargeld steht im Vordergrund der Betrachtung. Es sollen Faktoren wie Fehlbedienung (z.B. durch geknickte Geldnoten), mechanische Veränderungen und qualitative Unterschiede an den Geldnoten sowie weitere Umwelteinflüsse berücksichtigt werden. Die mechatronische Lösung soll adaptiv und robust gegenüber diesen Faktoren und dabei wirtschaftlich sein. Erhöhte Funktionalität und geringere Kosten sind natürlich gegenläufige Ziele. Im Sinne des Systems Engineering ergeben sich Konsequenzen nicht nur für den einzelnen Automaten, sondern ebenfalls für ein übergeordnetes System. Die Geldkreisläufe können weiter automatisiert und manuelle Tätigkeiten innerhalb dieser Prozesse reduziert werden: Dies führt letztlich zu geringeren Betriebskosten. Als primäre Einflüsse auf die Funktionalität können die Medien und die Bediener gesehen werden, mit längerem Umlauf des Bargeldes steigt der Verschleiß. Hier ist allerdings kein trivialer Zusammenhang erkennbar, sondern dieser variiert stark mit z.B. länder- und klimaspezifischen Gegebenheiten. Verschmutzte oder beschädigte sowie falsch eingelegte Banknoten können mit existierenden Systemen nur schwer gehandhabt werden, und werden beim Unterschreiten gewisser Qualitätseigenschaften aus dem Verkehr gezogen. Dies führt zu teuren Unterbrechungen geschlossener Geldkreisläufe.

Für den Markterfolg ist neben der Funktionalität vor allem die Zuverlässigkeit ein entscheidender Faktor. Dies betrifft einerseits die physikalischen Geldflüsse und anderseits

die Sicherheit der Transaktionen: In den vergangen Jahren ist ein starker Anstieg der organisierten Kriminalität zu beobachten. Bisher wurden Lösungen durch Redundanzen gefunden, was wiederum wirtschaftlich nachteilig ist. Ein Zuwachs an Funktionalität macht die Systeme allerdings auch komplexer und schwieriger beherrschbar. Durch den zu erwartenden Zuwachs an notwendigen Komponenten treten außerdem weitere wirtschaftliche Aspekte hervor. Ein effizienter technischer Ressourcenbedarf stellt demnach ein weiteres Ziel dar.

6.2 Synthese der Anforderungen

Dieser Abschnitt setzt die vier Anforderungsebenen aus der T-REQS Systematik an einem Beispiel um. Zur Veranschaulichung wird eine vereinfachende Darstellung des Systems verwendet. Das originale System wird auf eine Kinematik reduziert, welche Banknoten transportieren kann. Auf diese Weise können gerade die detaillierteren domänenspezifischen Anforderungen anschaulicher skizziert werden. Wird dies in Relation zum Geldnotentransport gesetzt, zeigt sich die Plausibilität dieser Vereinfachung, da wesentliche Merkmale eines Geldautomaten erhalten bleiben (vgl. Anhang A5-1).

6.2.1 Ziele

Ein Ziel des zu entwickelnden Systems stellt demnach der Transport von Banknoten dar, welches jedoch gerade im Gesamtkontext nicht das einzige Ziel ist. Ein Auszug möglicher Ziele für das zu entwickelnde System zeigt, dass es vielfältige Möglichkeiten gibt, auf diesem Level zu spezifizieren (vgl. [PvJV04]).

- Der Kunde kann eine Transaktion durchführen.
- Der Kunde erhält eine Bargeldauszahlung am Ende der Transaktion.
- Der Kunde bekommt eine Quittung am Ende der Transaktion.
- Der Kunde erhält eine Nachricht auf dem Display.
- Das System ist maximal 15 Minuten nicht verfügbar.
- Der Neustart des Systems dauert weniger als 2 Stunden.
- Der Kunde versteht auf Anhieb die Bedienung des Gerätes.

Es wird deutlich, dass es sich um eindeutige Ziele handelt, welche noch weiter spezifiziert werden müssen. Letztlich lassen sich die technischen Anforderungen auf die Ziele zurückführen. Daher ist die Ermittlung aller relevanten Ziele die erste Aufgabe im Kontext von Anforderungen und sollte systematisch erfolgen. Auf Basis der formulierten Ziele erfolgt die weitere Spezifikation. Geeignete Methoden zur Extraktion von Zielen finden sich in der Literatur [Rup07], darüber hinaus sind besonders *Anwendungsszenarien* sowohl zur Ermittlung von Zielen als auch zur Bestimmung deren Tests geeignet. Ein Anwendungsszenario beschreibt eine Situation der Nutzung des Systems und dient daher als Ansatz zur Verknüpfung von Anforderungen mit Tests.

Test von Zielen Die abstrakteren Ziele geben einen ersten Überblick über eine geeignete Testabdeckung. Konkrete Tests der Ziele sind z.T. schwierig zu realisieren, da Szenarien nicht unbedingt mit den Zielen korrelieren. Als Beispiel wird im folgenden MoSCoW angewendet, um Risiken für die Nichterfüllung der Ziele zu bestimmen. Dafür wird eine Zuordnung vorgenommen und die Auswirkungen möglicher Fehler in Betracht gezogen. Anschließend werden die Tests entsprechend der Methode zu *Must, Should, Could* und *Won't Tests* zugeordnet (Bild 6-4). Damit können die Ergebnisse festgehalten und entsprechend die Ziele konkreter formuliert werden (Tabelle 6-1), um für die Anforderungsdokumentation geeignet zu sein.

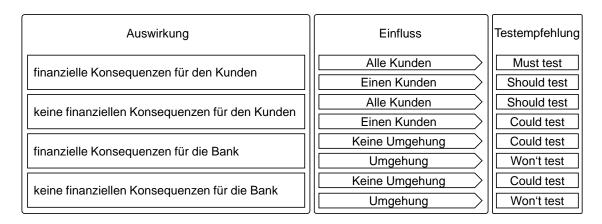


Bild 6-4: MoSCoW Testauswahl für einen Geldautomaten, abgewandelt nach [PvJV04]

6.2.2 Funktionen

Aus dem Ziel, dass dem Kunden die gewünschte Menge Bargeld ausgezahlt wird, lassen sich hierarchisch einige notwendige Funktionen ableiten. Die Verarbeitung von Bargeld wird im Wesentlichen durch die Hauptfunktionen aus Bild 6-5 erfüllt, dem Vereinzeln, Entfalten, Ausrichten, Transportieren, Zuordnen und Detektieren. Als Besonderheit ist

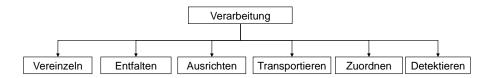


Bild 6-5: Die Funktionshierarchie der Verarbeitung von Banknoten

das Entfalten von Bargeld zu nennen, welche einen deutlich Funktionszuwachs in Geldautomaten darstellt². Der nächste Schritt ist die Detaillierung der aufgeführten Funktionen entsprechend einer Funktionshierarchie (Bild 6-5). Beispielhaft wird die Funktion des Entfaltens beschrieben (Bild 6-6) und eine erste Besonderheit ersichtlich: Offensichtlich gibt es mehrere Sichten auf eine Funktion. Einerseits ist dies eine prozessorientierte

²Die hier vorgestellte Methode zur Entfaltung ist nicht zwangsläufig im Geldautomaten umgesetzt. Innerhalb des Forschungsprojekts wurden verschiedene Methoden zur Entfaltung von Bargeld erarbeitet. Daraus sind weltweit mehrere Patente entstanden, z.B. [LLLH15].

Tabelle 6-1: Die Verbindung von Risiken, Testabdeckung und konkretisierten Zielen, abgewandelt nach [PvJV04]

Beschreibung der aufeinanderfolgenden Schritte und andererseits eine physikalische Beschreibung. Letztendlich wird ein Entfalten möglich, indem eine Haltekraft auf einen Teil der Note und eine Trennkraft auf den anderen Teil wirkt: Dafür ist allerdings ein technischer Prozess notwendig. An dieser Stelle sind verschiedene Prozesse denkbar, daher wird durch eine Festlegung der Übergang in den Lösungsbereich vollzogen. Dies sollte für die Anforderungen auf der Ebene der Funktionen berücksichtigt werden. Der Übergang in den Lösungsbereich ist das Schicksal einer jeden Anforderung, was deutlicher anhand der Funktionshierarchie einer detaillierteren Teilfunktion wird. Der Charakter der lösungsneutralen Beschreibung wird verändert, in dem notwendige Funktionen soweit detailliert werden, dass Lösungen berücksichtigt werden. Bild 6-7 zeigt exemplarisch die Funktionshierarchie eines Kraftaufnehmers mit lösungsorientierter Charakteristik. Die daraus abzuleitenden Anforderungen ergeben sich aus den jeweiligen zu erfüllenden Funktionen.

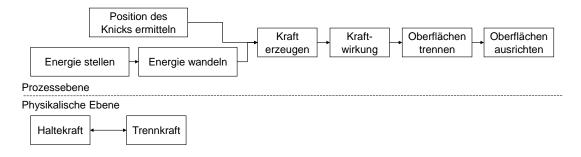


Bild 6-6: Die unterschiedlichen Sichten auf die Funktion des Entfaltens

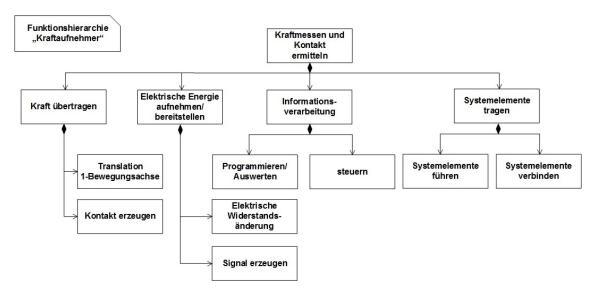


Bild 6-7: Auszug der Funktionshierarchie für einen Kraftaufnehmer in SysML Notation [EL16]

Diese sollten vollständig aufgenommen werden, um die Spezifikation der Anforderungen zu optimieren. Als Beispiel lautet eine offensichtliche Anforderung: Der Kraftaufnehmer muss eine Kraft übertragen. Für die Antwort auf die Frage, wie groß diese Kraft sein muss, ist der Einsatz erster Modelle sinnvoll (Bild 6-8). Diese können gleichungsbasiert sein und per Hand gerechnet werden, aber es ist sinnvoll, die so ermittelten Werte nachvollziehbar zu hinterlegen. Aus diesen Werten können entsprechende Kriterien für Funktionsnachweise erfolgen.

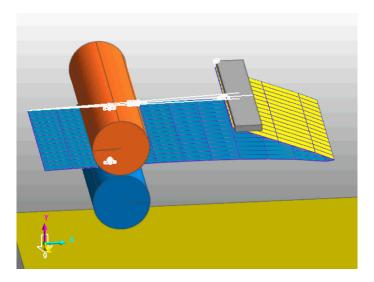


Bild 6-8: Frühes Modell eines Entfaltungsvorgangs für einen Funktionsnachweis

Test von Funktionen Ein solcher Funktionsnachweis kann als ein modellbasierter Test von Anforderungen in frühen Phasen der Entwicklung sehr hilfreich sein. Er ist jedoch nicht mit einem Test einer Anforderung zu verwechseln, da dieser in der späteren Integrationsphase der Entwicklung durchgeführt wird. Funktionsnachweise sind dennoch eine

sehr wertvolle Fähigkeit von modellbasierten Analysen in frühen Phasen und können zu Konkretisierung und ggf. Korrektur von Anforderungen eingesetzt werden. Sie leisten damit einen Beitrag zur Synthese der Anforderungen und bieten eine frühe Möglichkeit zur detaillierteren Spezifikation.

6.2.3 Domänenübergreifende Anforderungen

Als wirkungsvolles Instrument der Identifikation von domänenübergreifenden Anforderungen kann die Wirkstruktur genutzt werden. Die Wirkstruktur zeigt den Zusammenhang von Systemelementen auf unterschiedlichen Ebenen. Dabei werden vor allem die Schnittstellen deutlich. Externe Schnittstellen aus dem Umfeld des Systems und interne Schnittstellen im System selbst können dargestellt werden. Gerade die internen Schnittstellen sind von Interesse, da bei diesen die Schwierigkeiten der Abstimmung für Entwickler üblicherweise größer sind, als bei den prominenteren externen [Pfi17]. Auch zeigen sich in der Darstellung die Wechselwirkungen zwischen den Systemelementen. Wird in der Notation der Verbindungen unterschieden zwischen der Art der energetischen Verbindung, z.B. mechanisch oder elektrisch, werden beteiligte Domänen offensichtlich und eine Berücksichtigung in den Anforderungen kann erfolgen. In Bild 6-9 ist beispielsweise der Entfaltungsmechanismus u.a. verbunden mit den Medien, der Steuerung und weiteren Systemelementen. Hier sollte festgelegt werden, wie die Interaktion dieser Elemente erfolgen soll und welche Disziplinen beteiligt sind.

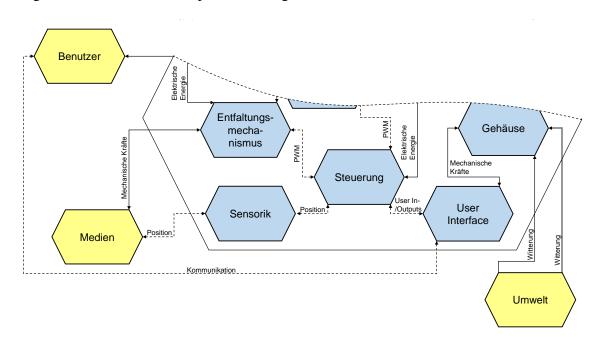


Bild 6-9: Ausschnitt einer Wirkstruktur eines Geldautomaten mit Entfaltungsmechanismus und Umfeldelementen

Eine Analyse der Wirkstruktur hilft auch dabei, nicht ausreichend spezifizierte Anforderungen aus vorherigen Schritten zu konkretisieren. Als Beispiel sei die Abweichung einer Note von ihrem Normzustand genannt, die durchaus Potential einer Quantifizierung besitzt und den Satz an Anforderungen eindeutiger macht. Im Anhang findet sich ex-

emplarisch ein Teil einer systematischen Kategorisierung von Noteneigenschaften (vgl. Anhang A5-2). Um bzgl. der Anforderungen eine korrekte Formulierung zu erlangen, reicht es nicht zu erwähnen, der Mechanismus solle eine Entfaltung von geknickten Noten realisieren. Eine präzise Formulierung lautet:

Das Entfaltungsmodul soll eine Trennkraft von 30 N auf die Note aufprägen.

Somit lassen sich konkrete Tests ableiten und der Interpretationsspielraum ist minimiert. Wird also ein Test des Entfaltungsmechanismus vorgenommen, muss dieser Test positiv abgeschlossen werden. Ansonsten erfüllt der Mechanismus nicht die an ihn gestellten Anforderungen.

Für eine weitere Detaillierung auf domänenübergreifender Ebene wird angenommen, ein Labormuster des Entfaltungsmechanismus soll mithilfe einer DELTA Kinematik³ realisiert werden. Die entsprechende Wirkstruktur ist in Bild 6-10 zu sehen, aus welchem sich beispielhaft domänenübergreifende Anforderungen an ein Sensorelement ergeben:

- Die Verbindung des Kraftsensors zu dem Aluminiumwinkelprofil soll über eine Klebeverbindung hergestellt werden.
- Der Austausch elektrischer Energie soll mit einer Messumformerschaltung realisiert werden.
- Die Krafteinleitung soll über einen PU-Zylinder erfolgen.

Die Lösungsneutralität ist weitestgehend aufgehoben, die Schnittstellen werden sichtbar, die beteiligten Disziplinen werden deutlich und der Grad der Detaillierung ist geeignet, um die domänenspezifische Ausarbeitung zu beginnen. Die Wirkstruktur bietet Einblicke in die Verbindungen und Schnittstellen von Systemelementen und enthält viele Informationen. Sie bleibt jedoch erstmal limitiert auf graphische Substanz, obwohl im Sinne des Systems Engineering schon weitere Analysen möglich sind. Diese Möglichkeiten sind jedoch nicht ausreichend, um den beschriebenen Herausforderungen in technischen Entwicklungen zu begegnen. Der konsistente Übergang zu domänenspezifischen Entwicklungsschritten erfordert systematische Detaillierungen.

Dies kann auch mithilfe einer physikalisch motivierten Wirkstruktur erfolgen, welche in Form der Modellstruktur in Bild 4-9 zu sehen ist. Die Systemstruktur wird beibehalten und die graphischen Eigenschaften bleiben vertraut. Ein solches physikalisch-basiertes Modell bietet überlegene Analysemöglichkeiten der Eigenschaften der Systemverbindungen: Das Risiko ungenügend spezifizierter Schnittstellen wird reduziert. Die Subsysteme sind über physikalisch sinnvolle Verbindungen miteinander verbunden und Annahmen über Gelenke, Massen oder Dimensionen können mit reduzierter Gefahr für implizite Annahmen ermittelt werden.

Eine einfache Regel lautet daher, die Systemstruktur in der Modellstruktur beizubehalten und alle Schnittstellen zu berücksichtigen.

Test von domänenübergreifenden Anforderungen Die Notwendigkeit der Tests wird auch an dieser Stelle gesehen. Im Zuge der domänenübergreifenden Anforderungen sollten Umwelteinflüsse, wie Umgebungstemperatur und Luftfeuchtigkeit systematisch

³Eine DELTA Kinematik ist eine von Clavel erfundene Roboterstruktur mit drei Freiheitsgraden [VC98].

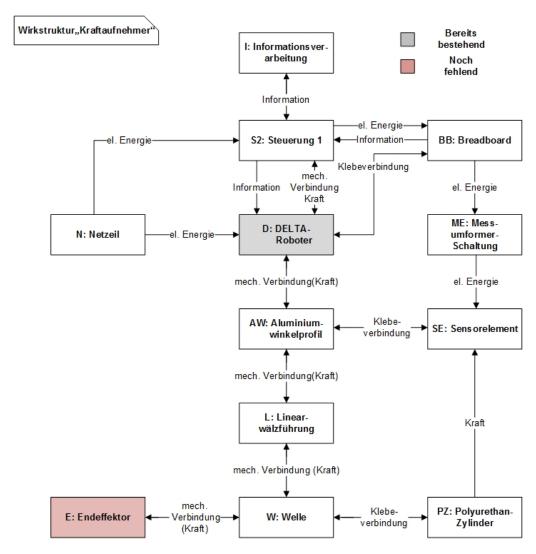


Bild 6-10: Wirkstruktur eines Kraftaufnehmers in SysML Notation [EL16]

überprüft werden. Die Schnittstellen des Systems sind hinsichtlich der Systemintegration zu überprüfen. Darüber hinaus sollten – entsprechend des Umfeldmodells – Tests durchgeführt werden, auch um z.B. Benutzerfehler oder mutwillige Manipulation zu testen.

6.2.4 Domänenspezifische Anforderungen

In Kapitel 4 werden die Herausforderungen gegenläufiger Anforderungen beschrieben. Diese Art von Anforderungen bietet auch Potential für eine detaillierte Spezifikation. Als Einstieg dient die Dimensionierung der Tresorwand des Geldautomaten. Unter den Gesichtspunkten des Explosionsschutzes durch mutwillige Manipulation und damit der Sicherheit des Bargeldes sollte die Wandstärke möglichst groß gewählt werden. Eine solche Optimierung würde auf eine unendliche und unsinnige Dicke hinauslaufen, welche der Vorgabe nach möglichst geringem Gewicht widersprüchlich ist. Die Wandstärke wird außerdem durch die maximalen Außenmaße begrenzt. Dennoch wird ein konkretes Volumen im Inneren des Tresors benötigt. Innerhalb dieser Randbedingungen lassen sich sehr konkrete Anforderungen an die Wandstärke aufstellen, ohne dass ein komplexes Problem

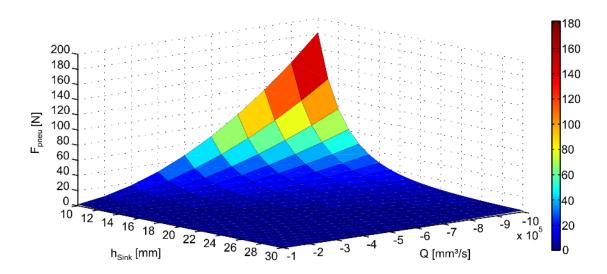


Bild 6-11: Simulationsergebnisse eines Saugmodells eines Greifers [LLT15]

gelöst werden muss. Dieses simple Beispiel zeigt, dass auch domänenspezifische Anforderungen in einem ganzheitlichen Kontext betrachtet werden sollten. Eine singuläre Optimierung einzelner Elemente führt nicht zwangsläufig zum Ziel.

Doch nicht für alle Systemelemente kann ein solches Vorgehen angewandt werden. Gerade für Elemente mit weitreichender Vernetzung ist dies schwieriger. Hier bietet sich das Werkzeug der Modellbildung und Simulation an. Es kann gedacht werden, dass für die Kraftaufbringung auf einer Note beispielsweise ein pneumatischer Greifer verwendet wird. Tritt dabei das Problem auf, die benötigten Luftströme des Greifers zu bestimmen, welcher als Endeffektor zur Aufbringung der Kräfte auf die Banknoten dienen soll (vgl. Bild 6-10), können über Pneumatikmodelle Relationen des Abstandes des Greifers zur Note, der aufgebrachten Kraft und des Luftstroms ermittelt werden. Somit können quantifizierte Anforderungen an den benötigten Luftstrom gestellt werden (vgl. Bild 6-11).

Das Modell lässt sich einfach adaptieren und idealerweise liegen parametrierbare Modelle in einer Art Modellbibliothek, um sie ohne großen Aufwand nutzen zu können. Bei vorangegangenen Entwicklungen entsteht womöglich eine breite Auswahl an Modellen, die entsprechend aufbereitet, zukünftige Entwicklungen vereinfachen. Solche Modelle systematisch zur Ermittlung der Anforderungen einzusetzen, unterstützt die Spezifikation hin zu korrekten und verifizierbaren, also quantifizierten Kriterien, bzw. Sollvorgaben. Die Größe der Kraft, die ein pneumatischer Greifer in diesem Beispiel erbringen muss, kann hier festgelegt werden. Im Laufe der Entwicklung kommt es zwar zu weiteren Herausforderungen, wie der Realisierung des dichten Kontaktes von Greifer und Banknote: Dies erfolgt jedoch unter der Anforderung, eine gewisse Kraft aufzubringen. Somit ist ein messbares Kriterium gefunden. Der Vorteil wird offenkundig, wenn die Evolution dieser Anforderung betrachtet wird. Zu Beginn steht ein (Teil-) Ziel, eine gewisse Note soll entfaltet werden. An dieser Stelle entspringt die Frage, wie dies überprüft werden soll. Sicher ist es möglich, schnell einen Testfall zu konzipieren, in welchem beispielsweise zwei Noten übereinanderliegen und das System soll nun eine Note von der anderen trennen. In diesen Gedanken sind mehrere Annahmen enthalten, zu einem großen Teil implizit. Die Nachteile einer solchen Herangehensweise sind klar: Es zeigt sich, dass diese uneindeu-

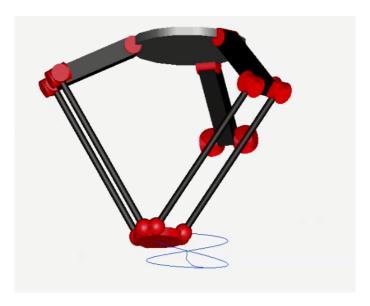


Bild 6-12: Modell einer DELTA Kinematik zur Bestimmung des Arbeitsraumes und der benötigten Aktuatordynamik

tigen Anforderungen ein Projekt torpedieren können.

Werden die Anforderungen detailliert und quantifiziert, treten eindeutige Anforderungen auf und ein Testfall kann konkret zugeordnet werden. Die Funktion, eine konkrete Halte-kraft auf die Note aufzubringen, lässt sich zwar auf mehrere Arten testen, aber sie lässt sich eben eindeutig testen. Jedes weitere Ziel sollte auf diese Weise detailliert werden, in dem es analysiert wird und ggf. durch Modelle unterstützt wird. Schritt für Schritt entstehen demnach konkretere Anforderungen.

Ein weiteres Beispiel ist durch die Anforderungen an die notwendigen Gelenkauslenkungen für einen definierten Arbeitsraum gegeben, welche ebenfalls modellbasiert ermittelt werden können (Bild 6-12). Mithilfe dieses Modells ergeben sich ebenfalls die benötigten dynamischen Eigenschaften der Antriebe, sodass konkrete Anforderungen an diese spezifiziert werden können. Das Spannungsfeld aus benötigter Geschwindigkeit, Dimensionierung der Antriebe und Größe des Arbeitsraumes kann auf diese Weise aufgelöst spezifiziert werden. Diese Modelle sind früh realisierbar und bieten dennoch einen Einblick in das Systemverhalten, der sonst viel später ermöglicht würde und somit ggf. Fehler durch nicht ausreichend spezifizierte Anforderungen, wie oftmals bei gegenläufigen Forderungen der Fall, entstünden.

Test von Systemen im Geldautomaten

Als Beispiel für Test domänenspezifischer Anforderungen dient ein Test einer Steuerung der Kinematik. Dafür kann angenommen werden, dass das gewünschte Verhalten durch das dynamische Modell exakt spezifiziert ist. Eine solche modellbasierte Spezifikation auf Basis dynamischer Modelle durchzuführen, kann die klassischen Anforderungslisten ergänzen.

Bild 6-13 zeigt eine Ansteuerung, mit welcher die Aktuatoren der Kinematik im Modell angesteuert werden. Auf der linken Seite sind verschiedene Testabläufe modellbasiert hin-

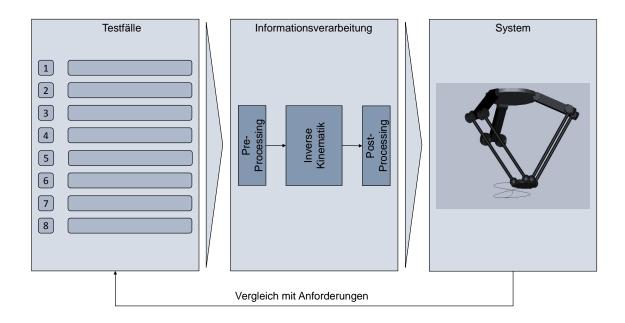


Bild 6-13: Modell zum Testen der Ansteuerung

terlegt, welche sukzessiv durchgeführt werden. Die entsprechenden Sollvorgaben werden über ein Filter (Pre-Processing), eine Koordinatentransformation und eine Begrenzung (Post-Processing) an das System geleitet. Hier setzen Elektromotoren die Signale in eine Bewegung um. Die erhaltenen Bewegungen sollen beim Übergang in die Realität ebenso beim realen System erfolgen, dies kann über den Einsatz von HiL Tests sichergestellt werden. Darüber hinaus sind die dafür verantwortlichen elektrischen Größen zu vergleichen. Ansonsten kann es zwar vorkommen, dass das System sich anforderungsgerecht bewegt, dafür aber die Aktuatoren unter Überlast laufen: Die dafür geeigneten Testfälle können analog dazu definiert werden, um anschließend systematisch einen Vergleich der Ergebnisse aus Simulation und realem Test durchzuführen.

Bei der Auswahl und Durchführung von Testfällen unterstützen weitere Methoden, wie in Abschnitt 4.4.1 beschrieben. Besonders die systematische Unterstützung bei der Testausführung bis zur Testautomatisierung erleichtert die Tests. Wird entsprechend der Anforderungen ein Testplan erstellt und dieser systematisch abgetestet, ergibt sich eine durchgängige Lösung. Zusätzlich können auch sehr pragmatische Ansätze, wie die FMEA unterstützend eingesetzt werden. So können beispielsweise Fehler, wie ein Sensorausfall, bewusst injiziert werden, um das Verhalten zu überprüfen. Das Ziel ist in jedem Fall eine Durchgängigkeit von den Zielen bis zum Test zu gewährleisten.

Dokumentation mit dem RegIF Creator

Um die systematischen Ansätze zur Anwendung zu bringen, sind an geeigneten Stellen unterstützende Tools einzusetzen. Zur besseren Handhabung und Dokumentation der Anforderungen wurde ein Tool zur Erstellung von ReqIF Dateien entwickelt. Dies ist gerade im Sinne eines pragmatischen Arbeitens vorteilhaft, um sich nicht zu früh auf die Möglichkeiten und Einschränkungen durch die typischen, sehr umfangreichen Tools festzulegen. Außerdem erlaubt die Software eine Durchgängigkeit von konzeptionellen Ar-

beitsergebnissen in standardisierte proprietäre Software. Die realisierten Funktionen sind rudimentär, ermöglichen jedoch die Erstellung von ReqIF Dateien aus csv-Dateien. Somit können alle Tools genutzt werden, die einen csv-Export anbieten, um daraus eine ReqIF Anforderungsliste zu generieren. Die Grammatik der Anforderungsdokumentation muss vom Anwender persönlich beachtet werden, dennoch erlaubt das Tool eine durchgängige Dokumentation von Anforderungen im richtigen Format: Es ermöglicht einen schnellen Zugang zur Dokumentation von *Requirements*. Bild 6-14, als exemplarische Anwendung, zeigt die Umwandlung einer Funktionshierarchie in ein ReqIF Dokument. Festzuhalten ist, dass das eingesetzte Tool zur Dokumentation sich in den Prozess einordnen sollte. Dementsprechend sollte ein Tool zur Anforderungsdokumentation systemspezifisch bewertet und ausgewählt werden. Mögliche Kriterien zur Auswahl können der Umfang der Umfang der Anforderungen, die Vernetzung oder die Änderungsrate sein.

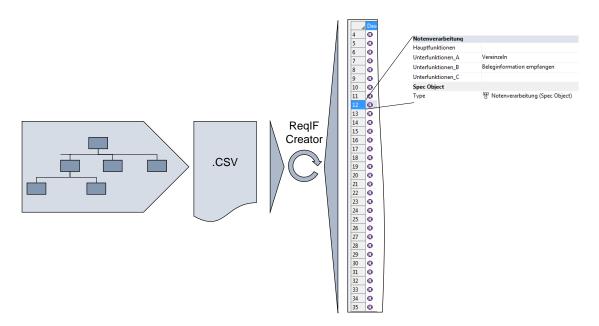


Bild 6-14: Schnelle Requirements-Dokumentation mit einem ReqIF Creator

6.2.5 Konsistente Anforderungen

Zum Abschluss der Synthese der Anforderung nach T-REQS soll eine Übersicht der Anforderungsebenen stehen. Aus einem Anwendungsszenario des Kunden mit dem Automaten seien beispielhaft zwei Ziele dargestellt:

- Der Kunde soll gefaltetes Bargeld einzahlen können.
- Der Kunde soll entfaltetes Bargeld ausbezahlt bekommen.

Aus dem Anwendungsszenario und den Zielen können funktionsorientierte Anforderungen abgeleitet werden:

- Der Entfaltungsmechanismus soll eine gefaltete Banknote entfalten.
- Die Position eines Knicks soll detektiert werden.

An dieser Stelle sei darauf hingewiesen, dass es bei der zu Beginn dieser Arbeit beschriebenen softwareorientierten Sicht oftmals zu Unterscheidungen zwischen funktionalen Anforderungen und nicht-funktionalen Anforderungen kommt. Das hier genannte Beispiel zeigt, dass diese Herangehensweise bei interdisziplinären Systemen an ihre Grenzen stößt. So könnte die untenstehende Anforderung, es sollen 20 Noten pro Minute entfaltet werden eben als Leistungsanforderung unzureichend bei der Auslegung berücksichtigt werden. Es ist aus mechatronischer Sicht jedoch eine bedeutende Anforderung, um z.B. die Mechanik auszulegen oder den Motor auszuwählen, sodass solch eine Anforderung explizit zu den domänenübergreifenden Anforderungen gezählt werden sollte. Diese lassen sich aus den funktionsorientierten Anforderungen bspw. durch einfache Modellbildung ableiten:

- Es sollen 20 Noten pro Minute entfaltet werden.
- Es soll eine Haltekraft von 30 N auf die Note aufgeprägt werden.
- Es soll eine Trennkraft 30 N auf die Note aufgeprägt werden.
- Die Abzugsrolle soll eine Maximalgeschwindigkeit von 1000 Umdrehungen pro Minute erreichen.
- Die Kommunikation zwischen Motor und Steuerung soll über einen PWM Ausgang erfolgen.

An dieser Stelle sei angemerkt, dass in diesem Schritt der domänenübergreifenden Anforderungen alle Anforderungen berücksichtigt werden sollten, die nicht explizit einer Domäne oder Disziplin zugeordnet werden können. Oftmals in Entwicklungen kommt es zu dem Trugschluss, dass ein interdisziplinäres System damit schon ausreichend spezifiziert ist für die disziplinspezifische Ausarbeitung. Hier gilt es jedoch behutsam vorzugehen. Bild 2-10 zeigt den Bedarf an disziplinübergreifender Koordination in der Phase der disziplinspezifischen Ausarbeitung im V-Modell. Die hier beschriebenen domänenübergreifenden Anforderungen eigenen sich, diese Koordination zu systematisieren. Sind die Anforderungen letztlich soweit spezifiziert, dass sie einer Disziplin zugeordnet werden können, und damit so detailliert beschrieben, dass eine weitere Betrachtung anderer Disziplinen nicht erforderlich ist, um die Anforderung zweifelsfrei umzusetzen, ist die Ebene der disziplinspezifischen Anforderungen erreicht:

- Der Reibwert zwischen Note und Walze soll 0,8 μ betragen. (Mechanik)
- Der Kraftaufnehmer soll Kräfte bis 60 N aufnehmen. (Mechanik)
- Der Elektromotor für die Abzugswalze soll 10 Nm stellen bei 1000 Umdrehungen pro Minute. (Elektrotechnik)
- Wenn die Funktion "Entfaltung" einen Befehl "Note vereinzeln" empfängt, soll die Abzugsrolle aktiviert werden. (Softwaretechnik)
- Wenn die Funktion "Entfaltung" von der "Kamera" das Signal erhält, dass die Note keinen Knick enthält, sollen die Abzugsrolle und die Gegenlaufrolle mit gleicher Geschwindigkeit in Transportrichtung beschleunigt werden. (Softwaretechnik)
- Die Kamera soll eine Auflösung von 1920x1080 Pixeln haben. (Elektrotechnik)

Gerade die hier aufgeführten disziplinspezifischen Anforderungen aus der Softwaretechnik scheinen eine gewisse Ähnlichkeit zu den funktionsorientierten Anforderungen aufzuweisen. Es sei jedoch darauf hingewiesen, dass es sich um eine konkrete Anforderung an eine Softwarefunktion handelt, wofür es u.a. eine Softwarearchitektur benötigt. Die weiteren genannten Anforderungen aus der Mechanik und der Elektrotechnik sind soweit detailliert, dass ein konsistenter Übergang in die disziplinspezifische Ausarbeitung gegeben ist.

Als Fazit aus dieser beispielhaften Darstellung der Anforderungsebenen ist die erhöhte Konsistenz zu ziehen. Die systematische Verfeinerung der Anforderungen zeigt gerade auf der domänenübergreifenden Anforderungsebene die Notwendigkeit, eine weitere Dekomposition vorzunehmen, um Anforderungen z.B. mit konkreten Systemelementen in Zusammenhang zu bringen. Zusätzlich bieten die domänenübergreifenden Anforderungen die Möglichkeit einer systematischen Integration im rechten Teil des V-Modells. Auf diese Art kann der Entwicklungsprozess systematisch unterstützt und die Durchgängigkeit erhöht werden.

6.3 Analyse der Anforderungen am Beispiel einer industriellen Lösung

Bisher wurde in diesem Kapitel ein akademisches Beispiel des DELTA Roboters verwendet. Zur Veranschaulichung der Anwendbarkeit von T-REQS wird im Folgenden auf die Umsetzung in einem industriellen Gerät eingegangen.

Eine Systematisierung des Analyseprozesses von Anforderungen bietet einen entscheidenden Mehrwert im Sinne des konsistenten Entwickelns: Die eingesetzte Systematik wird während der Entwicklung bereits reflektiert. Feedback und Reviews sind mächtige Werkzeuge zur Analyse von Anforderungen. Diese müssen jedoch systematisch verfügbar sein, um nicht durch iterative Schleifen ausgebremst zu werden. Dabei sind die eingesetzten Methoden, wie etwa die N3 Analyse, nicht als Pflicht zu sehen, sondern als mögliche Hilfestellungen. Die Analyse der Anforderungen als Prozessschritt kann sehr gut in den Entwicklungsprozess eingebettet werden, wenn vor jedem Verantwortungsübergang ein Analyseschritt eingeführt wird. Eine Auswahl der dafür eingesetzten Methoden ist letztlich zweitrangig und abhängig vom Entwicklungsfokus. Entscheidend ist die Integration der Analyse der technischen Anforderungen in den Entwicklungsprozess selbst.

6.3.1 Funktionsorientierte Analyse

Am Beispiel der funktionsorientierten Anforderungen sollte eine simple Analyse der Spezifikation Aufschluss über die geeignete Überprüfbarkeit geben. Alle Anforderungen sollten überprüfbar sein, daher kann bereits sehr früh ermittelt werden, ob die Anforderungen quantifizierbare Eigenschaften besitzen:

- Aufbringen zweier Kräfte als Halte und Trennkraft auf eine oder mehrere Geldnote/n
 - \rightarrow in der Einheit [N] möglich

- Positionsgetreues Aufbringen der Kräfte auf der oder den Note/n
 - → in der Einheit [mm] möglich bei Verwendung kartesischer Koordinaten
- Orientierung der Note ändern (rotatorische Genauigkeit der Positionierung um Achsen)
 - → in der Einheit [°] möglich

An dieser Stelle steht der Hinweis im Vordergrund, dass selbst einfache Analysen auf der Ebene der Ziele oder der funktionsorientierten Anforderungen einen Vorteil bieten. Die weiteren Aufgaben in der wären die Dekomposition der Anforderungen in die unterschiedlichen Disziplinen. Ein geeignetes Werkzeug zur begleitenden Analyse dafür ist z.B. die Modellbildung.

6.3.2 Modellbasierte Analyse

Das Vorgehen einer modellbasierten Analyse soll mittels eines Mechanismus gezeigt werden, welcher eine geknickte Banknote entfaltet. Die Synthese des Systems mit einer Analyse der Anforderungen Hand in Hand zu realisieren, ist ein erstrebenswertes Ziel. Es muss jedoch unbedingt beachtet werden, dass die getroffenen Annahmen nicht sowohl in das Modell einfließen, als auch in die Anforderungen selbst. Spätestens bei der Realisierung des Systems wird dies zu Unstimmigkeiten führen, die tunlichst vermieden werden sollten. Wird ein Modell jedoch als eine Form der Dokumentation von Anforderungen gesehen, wie bereits beschrieben, so ergeben sich Synergieeffekte, die die Effizienz der Entwicklungsarbeit steigern. Auf der Ebene der domänenübergreifenden Anforderungen ergeben sich auch strukturelle Fragestellungen bei der Analyse. Wie in Abschnitt 4.3 gezeigt, können z.B. aus der Wirkstruktur innere Zusammenhänge abgeleitet werden. Im Zuge einer Analyse sollten die getroffenen Annahmen für die Erstellung der Wirkstruktur hinterfragt werden, Schnittstellen und physikalische Effekte könnten sonst unberücksichtigt bleiben. Beispielhaft sei eine Abzugsrolle betrachtet, die von einem Elektromotor angetrieben wird. Bild 6-15 zeigt das physikalische Ersatzbild und den zugehörigen Bondgraphen.

Das Ersatzbild besteht aus einem elektrischen und einem mechanischen Teil. Mit entsprechendem Domänenwissen lassen sich die Bestandteile gut identifizieren vom Elektromotor bis zur Abzugsrolle. Wird der Zusammenhang aus Schnittstellensicht betrachtet, eignet sich die Bondgraphendarstellung aus dem oberen Teil von Bild 6-15. Der Gyrator (GY) kennzeichnet den Übergang der Disziplinen und bei der rechts hiervon betrachteten 1-Junction fällt auf, dass eine Reibung beachtet werden sollte. Diese ist aus Gründen der Darstellungskonsistenz zwar auch unten eingezeichnet, im Bondgraphen kann ein solch dissipatives Element jedoch an jeder 1-Junction systematisch überprüft werden. Letztlich schützt eine solche Betrachtung in gewissem Maße vor falschen Annahmen. Dies lässt sich sicher auch mit Domänenwissen realisieren, erfordert jedoch eben auch explizit dieses. Insgesamt bietet der Bondgraph also eine systematische Analysemöglichkeit für die in einer Wirkstruktur angenommenen Schnittstellen.

Zur weiteren Analyse soll ein Modell aus Gleichungen aufgebaut werden, um konkrete Vorgaben für die zu realisierende Lösung zu erhalten. Das freigeschnittene⁴ System be-

⁴Der Freischnitt ist eine elementare Technik der mechanischen Analyse von Systemen.

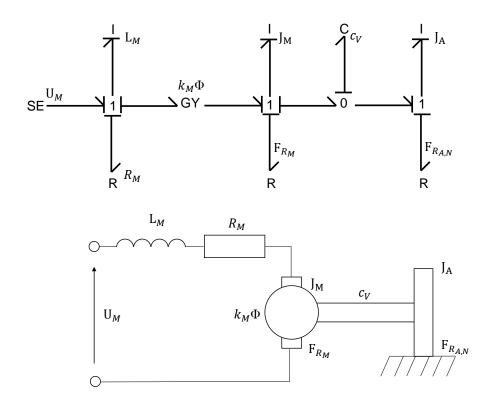


Bild 6-15: Bondgraph (oben) und physikalisches Ersatzbild (unten) einer von einem Elektromotor angetriebenen Abzugsrolle

steht aus zwei Rollen und zwei Banknotenhälften. Die Bewegung der Rollen wird über das von den Motoren bereitgestellte Drehmoment bestimmt. Die Bewegung der Papierhälften wird maßgeblich über die wirkenden Reibkräfte bestimmt, die zwischen der jeweiligen Rolle und der Banknote sowie zwischen den beiden Papierhälften wirken. Die wirkenden Normalkräfte variieren in Abhängigkeit der Anzahl der Banknoten, die sich zwischen dem Walzenpaar befinden. Solche Modelle sind prinzipiell für die Systemauslegung anzuwenden, bringen jedoch auch für die systematische Anforderungsanalyse Vorteile mit sich, da durch die mathematische Beschreibung eine Eindeutigkeit erzielt wird. Die Kraftgleichungen für das freigeschnittene System in Bild 6-16 lauten wie folgt [ML16]:

$$F_{R_{AN}} = \mu_{AN} \cdot F_{N_{AN}},$$

$$F_{R_{NN}} = \mu_{NN} \cdot F_{N_{NN}},$$

$$F_{R_{GN}} = \mu_{GN} \cdot F_{N_{GN}}.$$

Für das erstellte System werden zwei Drallsätze aufgestellt:

$$J_A \cdot \ddot{\varphi_A} = M_A - F_{R_{AN}} \cdot r_A,$$

$$J_G \cdot \ddot{\varphi_G} = M_G - F_{R_{GN}} \cdot r_G.$$

Hieraus folgen die Differentialgleichungen:

$$\ddot{\varphi}_A = \frac{1}{J_A} (M_A - F_{R_{AN}} \cdot r_A),$$

$$\ddot{\varphi}_G = \frac{1}{J_G} (M_G - F_{R_{GN}} \cdot r_G),$$

$$\ddot{x}_{N_{oben}} = \frac{1}{m_{oben}} (F_{R_{AN}} - F_{R_{NN}}),$$

$$\ddot{x}_{N_{unten}} = \frac{1}{m_{unten}} (F_{R_{GN}} - F_{R_{NN}}).$$

Diese Differentialgleichungen können nun in das Gesamtmodell eingebunden werden, um sie für weitere Analysen zu nutzen. Im weiteren Verlauf ergibt sich eine konkrete Beschreibung der zu erfüllenden Funktionen des Entfaltungsmoduls. Der Aufwand, der in die modellbasierte Betrachtung fließt zahlt sich üblicherweise aus. In diesem Beispiel eignen sich die hergeleiteten Differentialgleichungen zur Reglerauslegung in der domänenspezifischen Ausarbeitung. An dieser Stelle kann demnach ein zusätzlicher Nutzen für die folgenden Schritte der Entwicklung gesehen werden.

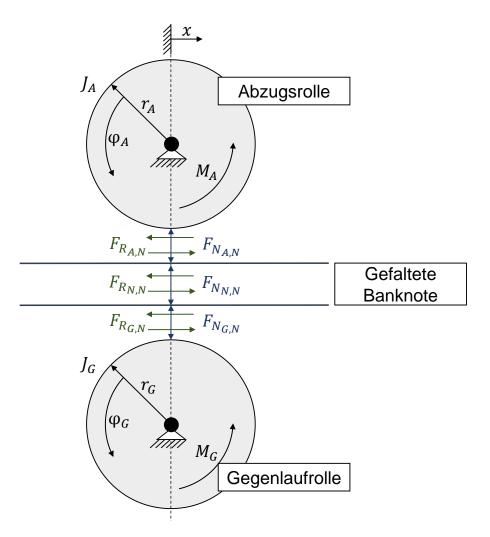


Bild 6-16: Einfacher Freischnitt eines Entfaltungsmoduls zur Darstellung der prinzipiellen Wirkung von Kräften und Momenten

6.3.3 Anforderungen für einen Entfaltungsmechanismus

In der laufenden Entwicklung sollten Entwicklungsergebnisse und Analyseergebnisse in die Anforderungen zurückgespielt werden. In diesem Abschnitt wird ein Auszug gezeigt, wie eine konsistente und eindeutige Beschreibung aussehen kann. Aus den verschiedenen Bestandteilen einer Anforderung (vgl. Abschnitt 3.3.1) wird die textuelle Beschreibung herausgegriffen, welche eine Möglichkeit darstellt, Eindeutigkeit zu unterstreichen⁵. Es werden beispielhaft einzelne zu realisierende Zustände spezifiziert, die für einen Entfaltungsvorgang benötigt werden und jeweils von einer Kamera initiiert werden. Dabei handelt es sich nicht um einen vollständigen Satz an Anforderungen, sondern um domänenübergreifende Anforderungen, die aus funktionsorientierten Anforderungen abgeleitet sind.

Ausgeschaltete Antriebsrollen:

- Die Drehzahl der Abzugsrolle soll $n_{soll} = 0$ [rad/s] betragen.
- Die Drehzahl der Gegenlaufrolle soll $n_{soll} = 0$ [rad/s] betragen.

Antreiben:

- Erkennt die Kamera eine Note, wird die Abzugsrolle auf die Drehzahl $n_{soll} = 62, 8$ [rad/s] beschleunigt.
- Erkennt die Kamera eine Note, wird die Gegenlaufrolle auf die Drehzahl $n_{soll} = 62, 8 [rad/s]$ beschleunigt.

Rolle bremsen:

- Nachdem der Knick in der Banknote die Abzugsrolle und die Gegenlaufrolle passiert hat, soll die Abzugsrolle mit maximalem Gegenmoment gebremst werden.
- Nachdem der Knick in der Banknote die Abzugsrolle und die Gegenlaufrolle passiert hat, soll die Gegenlaufrolle mit maximalem Gegenmoment gebremst werden.
- Information: Hierbei besteht immer noch ein Kontakt zwischen Rolle und Banknotenoberfläche. Es wird ein maximales Gegenmoment eingeleitet, um die Note anzuhalten und das Nachlaufen der Rolle wegen ihrer Trägheit gering zu halten.

Papierentfaltung:

- Nachdem die Banknote zwischen der Abzugsrolle und der Gegenlaufrolle fixiert ist, soll die Abzugsrolle ein rechtsdrehendes maximales Drehmoment einleiten.
- Nachdem die Banknote zwischen der Abzugsrolle und der Gegenlaufrolle fixiert ist, soll die Gegenlaufrolle auf ein adaptives rechtsdrehendes Drehmoment geregelt werden, um die Banknotenhälfte zu fixieren.

Walze bremsen:

- Nachdem die Banknote entfaltet ist, soll die Abzugsrolle auf die Drehzahl $(n_{soll} = 0 \ [rad/s])$ gebremst werden.
- Nachdem die Banknote entfaltet ist, soll die Abzugsrolle auf die Drehzahl $(n_{soll} = 0 [rad/s])$ gebremst werden.

⁵Die weiteren Bestandteile, wie die Identifikationsnummer werden an dieser Stelle nicht berücksichtigt.

Abtransport:

- Die Abzugsrolle wird auf die gewünschte Transportdrehzahl von $n_{soll} = 62, 8 [rad/s]$ in Transportrichtung geregelt.
- Die Gegenlaufrolle wird auf die gewünschte Transportdrehzahl von $n_{soll} = 62, 8$ [rad/s] in Transportrichtung geregelt.

Anhalten:

- Die Abzugsrolle soll auf die Drehzahl $n_{soll} = 0$ [rad/s] geregelt werden.
- Die Gegenlaufrolle soll auf die Drehzahl $n_{soll} = 0$ [rad/s] geregelt werden.
- Nachdem die Abzugsrolle und die Gegenlaufrolle zum Stillstand gekommen sind, soll der Automat in den Zustand *Ausgeschaltete Antriebsrollen* wechseln.

Diese Beschreibungen verdeutlichen demnach die Anforderungen an die Systemelemente der Rollen eines Entfaltungsmechanismus von Banknoten. Die vollständige Spezifikation umfasst weitergehende Anforderungen, wie die Schnittstellen oder die Architektur des Systems. Der Aufwand, welcher in eine gute Spezifikation geflossen ist, zahlt sich in der Regel aus.

6.3.4 N3 Analyse des Geldautomaten

In der funktionsorientierten und modellbasierten Analyse stehen die einzelnen Anforderungen im Fokus der Betrachtung. Im folgenden Abschnitt wird der Zusammenhang der Anforderungen mit dem System in Beziehung gesetzt und analysiert. Während der Entwicklung des Geldautomaten wird die N3 Matrix (Bild 6-17) aufgestellt und die entsprechenden Analyseschritte, wie die DoC Analyse durchgeführt. Das Vorgehen zeigt den Vorteil: Durch die Analyse entsprechend des in Abschnitt 5.3 beschriebenen Vorgehens, ergibt sich für die verschiedenen Anforderungen, Funktionen und Systemelemente eine relative Wichtigkeit, somit kann die Entwicklung auf die wesentlichen Anforderungen konzentriert werden. Hier soll gedanklich daran anschließend eine weitergehende Analyse exemplarisch gezeigt werden.

Aus dem Ausschnitt der N3 Matrix (Bild 6-17) kann augenscheinlich entnommen werden, dass die oberen beiden Ziele mit vielen Systemelementen und mit vielen Funktionen zusammenhängen. Dies ist so zu erwarten und gilt als Indikator für die Anforderungsebene der Ziele. Weiter unten in der Darstellung ist zu sehen, dass sich die Zusammenhänge reduzieren, es also weniger Einträge mit 1 gibt. Nichtsdestotrotz sollte festgehalten werden, dass es nicht das Ziel ist, die disziplinspezifischen Anforderungen in einen Zustand zu bringen, in welchem ihnen ein Systemelement oder eine Funktion eindeutig zugeordnet ist. Vielmehr sollten die Anforderungen soweit detailliert sein, dass eine Disziplin sie beherrscht. Ausdrücklich erwünscht ist der Austausch mit den anderen Disziplinen, bspw. bei Änderungen der Anforderung. Hier kann ebenfalls aus der N3 Matrix ein Zusammenhang gezogen werden, welchen Einfluss eine Änderung der Anforderung mit sich bringt. Als Anschauungsbeispiel sei die Anforderung herausgegriffen, der Reibwert solle 0,8 ubetragen. Es kann sich leicht vorgestellt werden, dass diese Anforderung im Zuge der Entwicklung nicht zu erreichen sei. Nun wäre es offensichtlich vorteilhaft, zu identifizieren, welche Systemelemente davon betroffen sind. In diesem Beispiel wäre eine Rückverfolgbarkeit z.B. über die Funktionen Vereinzeln und Entfalten möglich, da die Einträge

	unc	tions						Sys	tem	Elem	ents				
	/	//	/	//	/)	/)		Ser.		Elem Cogo,	//				
	\ \	//	//	//	\ \	\)		THOM	Yor.		',	//			
		Trans, Christians	\ \		A	(E)	Maurion (Se	1/2	10/A	6	Tolks	The hone			
Cres.	(S)/2	(87). S	0/2		Contraction of the second	1200	18/1/0	EL S	E SON	12/09	Wally.		(by	(F)	
Cotoling.	Entralia.	Usrichie 20	3/2	Tior one	Alexinore Property	Gederation of the second	%\\\(^{\infty}		o Ma		\$ \\ \[\gamma_{\text{!}}	6 13/2	LUB NO 1	tame,	\$
Der Kunde soll gefaltetes Bargeld	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
einzahlen können. Der Kunde soll entfaltetes Bargeld	1	0	1	1	0	0	1	1	1	1	1	1	1	1	0
ausbezahlt bekommen.	'	U	'	'	U	U	'	'	'	1	'	1	ı	ı	٥
Der Entfaltungsmechanismus soll eine gefaltete Banknote entfalten.	0	1	1	0	0	1	1	1	1	0	1	1	1	1	1
Die Position eines Knicks soll detektiert	0	0	0	0	0	1	_	0	1	0	0	0	0	0	1
werden.	U	U	U	U	U	'	0	U	,	U	U	U	U	U	'
Die Note soll mit 2 Metern pro Sekunde transportiert werden.	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0
Es sollen 20 Noten pro Minute entfaltet	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
werden.	'	•	'	'	'	'	l '	'	'	U	'	'	'	'	١
Es soll eine Transportkraft von 20 N auf die Note aufgeprägt werden.	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0
Es soll eine Haltekraft von 30 N auf die	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0
Note aufgeprägt werden.	'	•	'	U	U	U	l '	'	'	U	U	U	'	•	ŭ
Es soll eine Trennkraft 30 N auf die Note aufgeprägt werden.	1	1	1	0	0	0	1	1	1	0	0	0	1	1	0
Die Abzugsrolle soll eine															
Maximalgeschwindigkeit von 1000	1	1	0	0	0	0	1	0	1	0	1	0	0	0	0
Umdrehungen pro Minute erreichen.															
Die Kommunikation zwischen den Rollen- Elektromotoren und der Steuerung soll	1	1	0	0	0	0	0	0	1	0	1	1	0	0	0
über einen PWM Ausgang erfolgen.	•	•	O	Ü	O	O		Ü		O			Ü	O	ĭ
Der Reibwert zwischen Note und															
Abzugsrolle soll 0,8 μ betragen.	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
(Mechanik) Der Kraftaufnehmer soll Kräfte bis 60 N															
aufnehmen. (Mechanik)	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
Der Elektromotor für die Abzugswalze soll															
10 Nm stellen bei 1000 Umdrehungen pro	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
Minute. (Elektrotechnik)															
Wenn die Funktion "Entfaltung" einen															
Befehl "Note vereinzeln" empfängt, soll die Abzugsrolle aktiviert werden.	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0
(Softwaretechnik)															
Wenn die Funktion "Entfaltung" von der															
"Kamera" das Signal erhält, dass die Note keinen Knick enthält, sollen die															
Abzugsrolle und die Gegenlaufrolle mit	0	1	0	1	0	0	1	1	1	0	1	1	0	0	1
gleicher Geschwindigkeit in															
Transportrichtung beschleunigt werden. (Softwaretechnik)															
Die Kamera soll eine Auflösung von															
1920x1080 Pixeln haben (Elektrotechnik)	1	1	0	0	1	1	0	0	0	0	0	0	0	0	1

Bild 6-17: Ein Ausschnitt aus der N3 Matrix

in der *N*3 Matrix an diesen Stellen den Wert 1 haben. Darüber hinaus lässt sich augenscheinlich feststellen, dass die Funktion *Ausrichten* durch keine domänenspezifische Anforderung abgedeckt ist. Diese einfachen Analysen können systematisch vorgenommen werden. Ebenfalls lohnt sich bei komplexen und interdisziplinären Systemen der Einsatz erweiterter Analysemethoden, z.B. der Singulärwertzerlegung.

Wie in Kapitel 5 beschrieben, kann die Singulärwertzerlegung dazu genutzt werden, Zusammenhänge zwischen einzelnen Teilen der N3 Matrix zu identifizieren. Beispielhaft sei hier der Zusammenhang zwischen den domänenspezifischen Anforderungen und den Funktionen gezeigt. Die Matrix $N3_{RF_{DS}}$ lautet also

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Die Singulärwertzerlegung ergibt U=

[-0.4347]	0.0668	0.3599	-0.1022	0.8041	-0.1420
-0.4347	0.0668	0.3599	-0.1022	-0.5250	-0.6253
-0.4347	0.0668	0.3599	-0.1022	-0.2790	0.7673
-0.2439	0.3101	-0.0926	0.9142	0	0
-0.2719	0.6924	-0.5604	-0.3642	0	0
[-0.5475]	-0.6411	-0.5375	0.0170	0	0

 Σ ist

3.1154	0	0	0	0	[0
0	1.3459	0	0	0	0
0	0	1.0945	0	0	0
0	0	0	0.5337	0	0
0	0	0	0	0	0
0	0	0	0	0	0]

und V ist gleich

[-0.5943	-0.3275	0.4952	-0.5425	0	0]
-0.7599	0.4174	-0.1014	0.4880	0	0
0	0	0	0	-1.0000	0
-0.0873	0.5145	-0.5120	-0.6823	0	0
-0.1757	-0.4764	-0.4911	0.0318	0	0.7071
-0.1757	-0.4764	-0.4911	0.0318	0	-0.7071

Neben der oben erwähnten Erkenntnis, dass keine Anforderung für die Funktion *Ausrichten* vorhanden ist, können mithilfe der Singulärwerte weitere Analysen erfolgen: Der erste Singulärwert ist verhältnismäßig groß und deutet in diesem Beispiel an, dass viele Anforderungen auf eine Gruppe von Funktionen wirken. Die erste Spalte von U zeigt

den entsprechenden Einfluss der Anforderungen. Die ersten drei domänenspezifischen Anforderungen, sowie die zuletzt genannte, wirken am intensivsten auf diese Gruppe an Funktionen ein. In diesem Beispiel ist diese Gruppe durch die Funktionen *Entfalten* und *Vereinzeln* gekennzeichnet, was eine plausible Erkenntnis ist. Als weitere Erkenntnis lässt sich festhalten, dass die weiteren Anforderungen den anderen Funktionen scheinbar nicht so systematisch zugeordnet sind. In einem solchen Fall sollte die Spezifikation dahingehend analysiert werden.

Ohne eine detaillierte Analyse vorzunehmen, zeigt dieses Beispiel, dass praktische Erkenntnisse gewonnen werden können. Es kann ebenso gesehen werden, dass die Singulärwertzerlegung zwar einen praktischen Beitrag zur Analyse leistet, sie jedoch nicht vollständig durchführen kann, was auch nicht der Anspruch an sie ist. Sie ist ein unterstützendes Werkzeug aus der T-REQS Systematik. Eine wichtige Erkenntnis ist jedoch, dass durch die Analysen an der *N*3 Matrix sowohl der Zusammenhang der Anforderungen, als auch die Durchgängigkeit und Konsistenz betrachtet werden können.

6.4 Projektbezogene Evaluierung

Systematik zeigt, wie technische Anforderungen identifiziert werden können und in den Entwicklungsprozess integriert werden. Auf diese Weise kann die Entwicklungsaufgabe sehr systemnah spezifiziert werden und sich auf entscheidende Fragestellungen für den Entwickler konzentriert werden, ohne die übergeordneten Ziele zu vernachlässigen. Das systematische Vorgehen ermöglicht eine durchgängige Entwicklung des Geldautomaten. Die Entwicklung hat sich den Herausforderungen der Praxis zu stellen, und hat sich so im Laufe des Projektes adaptiert: Durch die konsistente Synthesephase in T-REQS ist ein Leerlauf der Entwicklungsziele vermieden worden. Konsequent können Systemelemente spezifiziert und damit einhergehend entwickelt werden. Dies resultierte in vier Patenten aus dem Forschungsprojekt heraus [LLLH15].

Durch die Analyse der Anforderungen und die entsprechende Extraktion von Kriterien ist eine systematische Bewertung unterschiedlicher Designoptionen möglich. Ein Einblick in die Auswertung wird in Bild 6-18 gegeben. Mithilfe von z.B. Spinnendiagrammen können die Eigenschaften eines Systems mit den Zielen visuell abgeglichen werden. Die modellbasierte Vorgehensweise erlaubt tiefere Einblicke in das Systemverhalten und das frühzeitige Absichern von Entwicklungsentscheidungen. Dabei werden diese Entscheidungen genau an den Anforderungen abgesichert, sodass eine zielgerichtete Entwicklung ermöglicht wird.

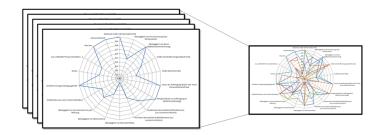


Bild 6-18: Mehrdimensionale Bewertung von Entwicklungsoptionen

Bei sehr vielen Entwicklungsoptionen ist ein Test gegen die Anforderungen notwendig, um die Entscheidungen für und gegen eine Option zu objektivieren. Dabei sind übergeordnete Anforderungen, wie etwa frühzeitige Ziele nur bedingt geeignet für solche Tests. Eine Synthese der unterschiedlichen Anforderungsebenen hilft dabei, die richtigen Entwicklungsschritte durchzuführen und geeignete Tests zu definieren. Diese Systematisierung bedeutet eine Verlagerung des Entwicklungsaufwandes in frühere Phasen des Entwurfs. Dies ist im Sinne niedriger Fehlerkosten sinnvoll, um Fehler entweder gleich zu vermeiden oder früher zu entdecken.

Zusammenfassung In diesem Kapitel wurde die Systematik (T-REQS) an einem Beispiel angewendet und evaluiert. Eine Quantifizierung von Projekterfolg lässt sich im wissenschaftlichen Sinn nur mit größerem Aufwand darstellen. Es bleibt jedoch festzuhalten, dass das Projekt erfolgreich abgeschlossen wurde und eine überdurchschnittlich hohe Anzahl an Patenten hervorbrachte. Dies sind zumindest Indikatoren für einen erfolgreichen Einsatz der Systematik. Es konnte gezeigt werden, wie konsistente Anforderungen erarbeitet werden und wie diese analysiert werden. Einerseits wurden dafür Modelle eingesetzt, andererseits strukturelle Untersuchungswerkzeuge, wie die *N*3 Matrix.

7 Zusammenfassung und Resümee

Nach der Evaluation an einem industriell geprägten Beispiel blickt dieses Kapitel in einem weiteren Sichtfeld auf die T-REQS Systematik. Dafür wird zuerst die Synthese bewertet. Anschließend folgen Bewertungen der Analyse und der Systematik insgesamt, um sie dann mit den in dieser Arbeit gestellten Anforderungen an eine Systematik zu vergleichen. Die Erweiterbarkeit der Systematik wird im anschließenden Abschnitt thematisiert, um das Kapitel mit einem Fazit zu schließen.

7.1 Bewertung der Syntheseverfahren

Die beschriebenen vier Ebenen von Anforderungen stellen eine Richtlinie zur Verfeinerung und Detaillierung von Anforderungen in einem Entwicklungsprozess dar. Die Systematik ermöglicht systematisches Spezifizieren und Analysieren von Anforderungen. Eine Unterscheidung zwischen Ebenen von Anforderungen ist nicht grundsätzlich neuartig, jedoch sind diese Unterscheidungen bisher nicht systematisch in einen Entwicklungsprozess für traditionell ingenieurwissenschaftliche Entwicklungsprozesse eingegliedert. Die verschiedenen Ebenen zeigen eine Möglichkeit einer ganzheitlichen systematischen Anforderungsermittlung auf. Dennoch kann es mitunter problematisch werden, die softwareorientierten Anforderungen einzuordnen. Die formalen Testmethoden sollen nicht ersetzt werden, sondern geeignet eingebettet bzw. ergänzt werden. Dafür wird vorgeschlagen, Softwareanforderungen, welche Funktionen quantitativ beschreiben, in das Level der disziplinspezifischen Anforderungen einzusortieren. Qualitative Funktionen sind auf einer übergeordneten Ebene und sollten in die Ebene der Funktionen eingegliedert werden.

Aus betriebswirtschaftlicher Sicht kann die Entwicklung eines Produktes auch im Sinne einer Push- oder einer Pullstrategie erfolgen. Bei der Pushstrategie wird versucht, ein Produkt zu platzieren, dessen Beschaffenheit eine Nachfrage nach sich zieht. Eine Pullstrategie entwirft dieses Produkt entsprechend im Vorfeld genau analysierter Anforderungen [MBK15]. Schon seit geraumer Zeit wird eine dual getriebene Strategie diskutiert [HV03]. Für diese Strategien benötigt es einen entsprechenden unterstützten Entwurfsprozess, welcher durch die hier vorliegende Systematik gegeben ist. Juristische Fragen werden nicht fokussiert, denn letztendlich steht aus wissenschaftlicher Sicht die erfolgreiche Entwicklung eines Produktes im Vordergrund. Das Ziel einer Anforderungserfassung und -detaillierung ist somit die Ermittlung einer Menge an Anforderungen, welche die geforderten Eigenschaften aufweisen.

Anforderungen werden benötigt, wenn es zu Übergabe- oder Übergangsprozessen kommt. Sie ähneln einem Protokoll, welches nach geeigneten Regeln definiert, was von einem Prozess zu erwarten ist. Ein solcher Übergang tritt beispielsweise ein, wenn ein Kunde einem Entwickler einen Auftrag gibt. Dies wird in bekannter Form mittels Lasten- und Pflichtenheft definiert. In einem komplexen Entwicklungsprozess treten solche Übergaben jedoch häufiger auf – gerade, wenn es sich um interdisziplinäre Projekte handelt. Die vielen verfügbaren Methoden zur Entwicklung interdisziplinärer Systeme werden durch die vorgestellte Systematik ergänzt. Die Schnittstellen werden systematisch berücksich-

tigt, sodass feste Strukturen für die Informationsweitergabe festgelegt sind. Typischerweise ändern sich gerade im Laufe interdisziplinärer Systementwicklungen die geforderten Eigenschaften der Stakeholder in unterschiedlichen Stufen des Entwurfs. Je weiter die Entwicklung voran schreitet, desto detaillierter werden die geforderten Eigenschaften sein. In einem System aus Anforderungen, das auf einem einzigen Level beruht, kann dies schnell zu Traceability Problemen führen. Im hier vorgestellten Fall wird diesen Herausforderungen strukturell begegnet.

Diese strukturelle Sicht auf Anforderungen hilft auch bei der Entwicklung stark vernetzter und komplexer Systeme. Durch den hierarchisierenden Charakter wird eine Strategie genutzt, die häufig zur Komplexitätsbeherrschung verwendet wird. Die Klassifikation entlang des Entwurfsprozesses enthält demnach die Eigenschaften:

- Strukturelle Einordnung
- Prozessunterstützung
- Hierarchie

Diese Eigenschaften sind grundlegend zur Erreichung der geforderten Eigenschaften an eine Systematik. Zurückblickend auf die ergänzenden Forderungen nach Systematizität (vgl. Abschnitt 3.3.2) können die Eigenschaften der Vollständigkeit, des Zusammenhangs und der Konsistenz positiv beurteilt werden. Die Fragen nach der Anwendbarkeit und Einfachheit können auf Grundlage der bisherigen Anwendungserfahrungen ebenfalls positiv betrachtet werden, jedoch ist hier weitere Evaluierung notwendig. Insgesamt bilden die vier Ebenen ein geeignetes Gerüst zur Synthese der Anforderungen.

7.2 Bewertung der Analyseverfahren

Der Analyseteil von T-REQS zielt auf die Bewertung von Verbindungen von Anforderungen mit dem System ab. Gängige SysML Softwaretools bieten viele Optionen, um Informationen und Verbindungen zwischen verschiedenen Arten von Elementen zu sichern, zu definieren und darzustellen. Im Speziellen das Requirement Diagram bietet die Möglichkeit, Systemelemente, Funktionen und Anforderungen zu verbinden [Wei14]. Nichtsdestotrotz gibt es keine Systembeschreibung, die die Forderung nach mathematisch interpretierbarer Darstellung zur Analyse dieser Verbindungen vollständig erfüllt. Solch eine Darstellung wird durch die N3 Matrix ermöglicht. Dabei können alle Ebenen der Anforderungen darin berücksichtigt werden.

Die Nutzung der Einträge *Null* oder *Eins* in der N3 Matrix stellen jedoch eine unbedeutende Verbindung zwischen zwei ihrer Bestandteile als genauso wichtig dar, wie eine bedeutende. Möglicherweise stellt ein Ansatz wie der Page Rank Algortihmus eine Inspiration zur Lösung der besseren Bewertung dar, indem die Gewichtung der Matrixelemente Berücksichtigung findet [PB98]. In dem in Kapitel 5 dargestellten Beispiel werden sieben Elemente analysiert. In einem solch kleinen Anwendungsfall wird offensichtlich, dass eine simple Zählung der entsprechenden Einträge ebenfalls funktionieren würde. Die vorgeschlagene Methode ist jedoch dahingehend nachhaltiger, dass die Einführung einer

Gewichtung realisiert werden kann. Daraus würden weitere Vorteile von M_{λ} und DoC entstehen.

Eine Eigenschaft eines generischen Ansatzes ist die eingeschränkte Anwendbarkeit auf bestimmte Anwendungsfälle, die es sicherlich auch für die vorgestellte Systematik gibt. Zielgerichtete Anpassungen sollten daher bei Bedarf unternommen werden, um die Benutzbarkeit zu erhöhen. Die N3 Analyse umfasst nicht zwangsläufig physikalische Parameter, da ihr eine strukturelle Analyseaufgabe übertragen ist. Diesen Schwachpunkt gilt es mit einer konsequent physikalisch motivierten Entwicklung zu beheben. Dafür erlaubt die N3 Matrix flexible Anpassungen. Ohne Weiteres sind weitere Dimensionen dem N3 Cube hinzuzufügen. Dies wäre zwar graphisch nicht mehr darstellbar, ändert jedoch die mathematische Herangehensweise nicht. Somit ist eine skalierbare Analysefunktion gegeben. Außerdem bietet sie selbst für komplexe Systeme eine Strukturierung und bietet Einblicke völlig unabhängig von der Systemgröße. Des Weiteren sie ist unabhängig von Expertenwissen interpretierbar. Für die Singulärwertzerlegung von N3 ergeben sich ähnliche Eigenschaften. Potentielle Schwierigkeiten sind dennoch nicht von der Hand zu weisen. Die Auswertung und Interpretation der Ergebnisse bleiben mehr eine Kunst als eine Wissenschaft. Vor allem, wenn sich keine Gruppen ausprägen, ist die Interpretation der Singulärwertzerlegung schwierig. Doch dies allein könnte bereits eine lohnenswerte Erkenntnis sein. Die weiteren dargestellten Analysewerkzeuge unterstützen einen Entwicklungsprozess im Kontext von Anforderungen. In den diskutierten Gegebenheiten ist ein Einsatz durchaus Johnenswert.

Auch um den Fortschritt von Anforderungen beurteilen zu können, bieten sich Performance Indikatoren, wie z.B. der λ -Rank an. Weitere Erkenntnisse für die Analyse von Anforderungen können aus der Merkmalshäufigkeitsanalyse gezogen werden. Hierfür ist die Zuordnung zu den unterschiedlichen Ebenen der Anforderungen zwingend erforderlich. Wird nun die prozentuale Verteilung betrachtet, zeigt sich, in welchem Stadium ein Set an Anforderungen ist und worauf sich eine Entwicklung primär konzentriert: Konzeptionelle Arbeiten sind anzustreben, wenn tendentiell viele Ziele vorhanden sind. Implementierungsprojekte sind durchzuführen, wenn der Anteil domänenspezifischer Anforderungen besonders hoch ist. Diese Erkenntnisse können sich ebenfalls positiv auf die Konsistenz auswirken und vor allem auf die Vollständigkeit. Das Ziel sollte ein ausgewogener Satz an Anforderungen sein. Die Analyse in T-REQS erlaubt die Bewertung der Ausgewogenheit von Anforderungen.

7.3 Bewertung der Systematik

Neben der Bewertung der Einzelteile der Systematik, lohnt eine Betrachtung der Zusammensetzung anhand ganzheitlicher Kriterien. Das gezeigte Vorgehen ist u.a. eine systematische und mathematische Möglichkeit der Identifizierung von integraler oder modularer Zuordnung von Funktionen, Systemelementen und Anforderungen. Eine modulare Zuordnung beschreibt die direkte Verknüpfung jeder Funktion mit jeweils einem Systemelement. Sind mehrere Systemelemente an der Erfüllung einer Funktion beteiligt, handelt es sich um eine integrale Zuordnung [Ulr92]. Eine Darstellung der Verbindung von Funktionen und Systemelementen wird in Bild 7-1 gezeigt. Diese Sichtweise lässt sich auch auf Anforderungen erweitern. Modulare Systeme vereinfachen das Komplexitätsmanage-

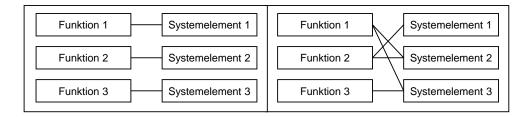


Bild 7-1: Modularer (links) und integraler Zusammenhang zwischen Funktionen und Systemelementen, abgewandelt nach [Ulr92]

ment: Analog helfen modulare Anforderungen, die Komplexität zu beherrschen. Dafür sind jedoch geeignete Maßzahlen notwendig, um eine Aussage treffen zu können, ob Anforderungen modular oder integral sind – die N3 Matrix erlaubt solche Erkenntnisse.

Zur objektiveren Bewertung der Systematik lassen sich weitere Kriterien finden. Diese können aus der Bewertung von Software abgeleitet werden [Bay14]. Für die Bewertung der Systematik müssen diese individuell angepasst werden, doch die Konzepte dahinter bleiben bei der Bewertung erhalten. Es kann die perzeptive Unterscheidbarkeit untersucht werden, die zwar üblicherweise zur Bewertung von Diagrammen Anwendung findet, hier aber unterstützend mitgenutzt werden kann: Ist also eine einfache und genaue Unterscheidung von verschiedenen Symbolen, bzw. in diesem Fall von Anforderungsebenen, möglich? Daher gilt es für die Anforderungsebenen eine möglichst große Unterscheidbarkeit herzustellen. Grundsätzlich ist dies erfüllt, jedoch ist durch fließende Übergänge eine Aufweichung dieses Bewertungskriteriums gegeben. Die Bewertung der visuellen Ausdruckskraft anhand der N3 Analyse ist kritisch zu sehen. Die Anzahl verwendeter visueller Variablen und die Kapazität ist alleine durch Ziffern und Buchstaben gegeben. Hier wäre eine Erweiterung um grafisch zu interpretierende Elemente denkbar. Dagegen ist das Konzept der semantischen Transparenz eher positiv zu beurteilen. Sicher sind weitergehende Analysen notwendig, um diese Annahme zu stützen, doch die Exisitenz von vier verschieden Ebenen von Anforderungen ist für den Nutzer erst einmal genauso plausibel, wie die Verknüpfungen innerhalb der N3 Matrix. Ein Anwender kann diese Elemente der Systematik schnell begreifen.

Eine weitere Eigenschaft, die bisher eher randständig integriert ist, ist das Konzept der dualen Codierung. Auf eine Systematik umgedeutet, bedeutet dies, dass sie nicht ausschließlich textuell beschrieben ist: Die schriftlich beschriebenen Elemente der Systematik sind rudimentär bildlich dargestellt (vgl. Bild 1-2) und hier liegt eine Chance, das Verständnis der Systematik weiter zu erhöhen. Es zeigen sich Verbesserungspotentiale der Systematik, die in der Umsetzung liegen. Zur grundsätzlichen Bewertung des Nutzens nimmt der folgende Abschnitt Stellung.

7.4 Nutzenevaluierung

Eine Frage, die sich immer stellt, ist die nach dem messbaren Vorteil bei Anwendung neuartiger Systematiken und modellbasierter Vorgehen. Am Ende steht eine Investition von Aufwand zum Einsatz, es fehlt jedoch der Vergleich, ob es ohne diese Investition ein vergleichbares Ergebnis gegeben hätte. Zur positiven Beantwortung dieser Frage sei auf die Literatur verwiesen [Kir11]. Es wird die Effizienzsteigerungen modellbasierter Entwicklung bestätigt. In der Softwareentwicklung der Automobilbranche sind durch modellbasierte Entwicklung bis zu 30 % Zeitersparnis möglich. Es zeigt sich, dass mittels einer Anforderungsvalidierung 10 % der Gesamtfehler gefunden werden. Die Durchgängigkeit von Softwaredesign und Implementierung spart tendentiell 50 % der Kosten und der Zeit. Außerdem wird der positive Effekt aufs Testen dargestellt [Kir11]. Die grundsätzlichen Vorteile werden dadurch bereits deutlich. Der Nutzen von Anforderungen lässt sich auch anhand zahlreicher weiterer Studien belegen. Repräsentativ seien zwei Zusammenhänge angegeben [Ebe10]

- 80 % der Fehler in Tests und 43 % der Feldfehler resultieren aus fehlerhaftem Requirements Engineering.
- Es werden etwa 3-6 % des Aufwands in das Requirements Engineering investiert. Wird dieser Aufwand verdoppelt, reduzieren sich die Lebenszykluskosten in einem Bereich von 20-40%.

Diese Zahlen belegen die Bedeutung einer sinnvollen Beschäftigung mit Anforderungen. Letztlich deuten Anforderungen darauf hin, ob jemand weiß, was er will oder nicht. Die Bereitschaft zur detaillierten Auseinandersetzung sollte also auch in diesem Kontext gesehen werden.

Verwertungsstrategie Der Nutzen kann demnach grundlegend quantifiziert werden. Im Sinne eines durchgängigen Entwurfs gibt es bisher viele sehr abstrakte Methodiken, welche gerade in technologieintensiven Entwicklungen oft nicht nachhaltig eingesetzt werden. Sehr formale Ansätze aus dem Systems Engineering bedeuten immer auch einen hohen Aufwand für die Dokumentation. Die digitale Durchgängigkeit wird ebenfalls nicht inhärent geliefert, sondern ist toolabhängig. In technologieintensiven Branchen, in denen Entwicklungsprozesse neben der gelebten Entwicklung laufen und nicht Hand in Hand gehen, bietet der in dieser Arbeit vorgestellte Ansatz ein großes Potential für einen echten Nutzen. Der Entwickler wird geleitet und die Werkzeuge werden gezielt eingesetzt.

Es können neue Erkenntnisse in den Bereichen Systementwicklung und Modellbildung für den Entwurf intelligenter technischer Systeme gewonnen werden. Die Ergebnisse erhalten somit eine umfangreiche Multiplikation, hinausgehend auf alle relevanten Bereiche, die der Entwicklung und Konzeption intelligenter technischer Systeme dienen. Gerade im Hinblick auf die Vernetzung und Komplexität und die verstärkte Nutzung und Gewinnung von Daten als neue Geschäftsmodelle wird ein solcher Ansatz benötigt.

Die Eigenschaft von T-REQS zur Steigerung der Zielorientierung und die Abschwächung des Kausalismus in unterschiedlichen Entwicklungsschritten bergen Potentiale zu einem Einsatz in sehr vielen verschiedenen Entwicklungsprojekten. Prinzipiell sind einem unterstützendem Einsatz der Systematik wenig Grenzen gesetzt. Die Skalierbarkeit des Ansatzes ist durch die klar getrennten Bausteine der Synthese, der Analyse und der Werkzeuge gegeben. Da nicht alle Werkzeuge genutzt werden müssen, um Anforderungen entsprechend der Systematik zu klassifizieren, stehen einem bedarfsgerechten Einsatz wenige Hindernisse im Weg. Diese Eigenschaft erlaubt es, die Systematik sukzessive in Projekten einzusetzen und die Erkenntnisse zurückzuspielen.

Durch den offenen Gedanken der Systematik können Einzelteile dieser sich ebenfalls in etablierten Methodiken wiederfinden. Darüber hinaus zeigt sich die Möglichkeit der In-

tegration in bestehende Softwarelösungen. Einerseits zur Klassifikation von Anforderungen, andererseits für die Auswertung dieser, z.B. mittels N3 Analyse. Außerdem zeigen sich Einsatzmöglichkeiten in der Lehre. Ausgehend von steigenden Bedarf an interdisziplinär denkenden Ingenieuren und den daraus abzuleitenden Herausforderungen an die Ausbildung [Wor15], ist es sicher nicht schädlich, auch das Thema der Anforderungen als Brücke in interdisziplinären Entwicklungen zu begreifen und zukünftigen Entwicklern dieses näher zu bringen.

7.5 Problemlösungskapazität

Der systematische Kreislauf, der mit T-REQS erreicht werden kann, ist in Bild 7-2 dargestellt. Die Elemente können mit den Werkzeugen, die in dieser Arbeit beschrieben sind, systematisch detailliert werden.

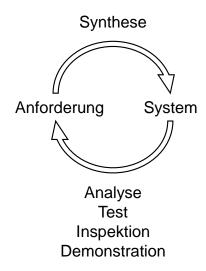


Bild 7-2: Systematische Evaluation der Entwicklungsschritte

Zurückblickend auf deduktive und induktive Herangehensweisen zu Beginn dieser Arbeit soll ein Exkurs in die Physik getätigt werden. Laplace wagte zu Beginn des 19. Jahrhunderts eine Formulierung:

"Eine Intelligenz, welche für einen gegebenen Augenblick alle in der Natur wirkenden Kräfte sowie die gegenseitige Lage der sie zusammensetzenden Elemente kennte und überdies umfassend genug wäre, um diese gegebenen Größen der Analysis zu unterwerfen, würde in derselben Formel die Bewegungen der größten Weltkörper wie des leichtesten Atoms umschließen; nichts würde ihr ungewiß sein und Zukunft und Vergangenheit würde ihr offen vor Augen liegen" [GK06, S. 394].

In einer solchen Umgebung wären die Möglichkeiten deduktiver Herangehensweisen womöglich größer. Seit der Entdeckung der Quantenmechanik ist diese Formulierung, die auf den Newtonschen Bewegungsgleichungen [New10] fußt, jedoch weitestgehend entkräftet [Haw88]. In den Ingenieurwissenschaften wird zwar aus offensichtlichen Gründen auch mit Modellen, die auf diesen Newtonschen Bewegungsgleichungen beruhen, gearbeitet. Trotzdem sollte nicht vergessen werden, dass wohl nicht anders als in der Physik, ein deduktives Gebilde (vorerst) nicht zwangsläufig im Fokus steht. Eine detaillierte Beschreibung zu diesem Thema kann in der angegebenen Literatur nachgelesen werden [Fra88]. Die bestmögliche Schlussfolgerung begrenzt jedenfalls die Möglichkeiten der Herangehensweisen.

Wird diese Darlegung auf die geschilderten Probleme angewandt, ist die Zielsetzung dieser Arbeit durch die systematische Unterstützung bei der Entwicklung technischer Systeme abstrahiert. Dabei sind die Probleme die durch die Interdisziplinarität, die Vernetzung sowie den technologischen Fortschritt bedingt sind, durch die systematische Herangehensweise der unterschiedlichen Anforderungslevel adressiert. Somit kann von einer gesteigerten Konsistenz durch den zielgerichteten Prozess ausgegangen werden. Hierbei ist das typische Problem unterschiedlicher Detaillierungsgrade von Systembeschreibungen vor allem durch die Auseinandersetzung mit diskreten und kontinuierlichen Systembeschreibungen angegangen. Hierdurch ist ein Beitrag zum besseren Verständnis von Anforderungen aus unterschiedlichen Sichten gegeben. Die strikte Einbindung von Modellen sorgt darüber hinaus für eine gesteigerte Fähigkeit der Auseinandersetzung mit Emergenz. Die gestellten Anforderungen an eine Systematik, die sich aus den beschriebenen Problemen ergeben, können mithilfe der vorgestellten Systematik positiv bewertet werden:

- A1 Systematische Synthese von Anforderungen: Durch die verschiedenen Ebenen der Anforderungen und die Bereitstellung von Werkzeugen zur Überführung von Anforderungen von einer in die nächste Ebene ist ein systematischer Prozess zur Synthese von Anforderungen vorhanden.
- A2 Mathematisch interpretierbare Analyse von Anforderungen: Die vorgeschlagenen Werkzeuge zur Analyse beruhen weitestgehend auf mathematisch interpretierbaren Vorgehen. Die Modellierungsansätze und die regelungstechnischen Methoden bauen natürlich auf der Mathematik auf. Doch auch die N3 Analyse ist auf einem soliden mathematischen Fundament konstruiert, um eine einfache, jedoch plausible Analyse von Anforderungen zu erhalten.
- A3 Darstellen von Anforderungen und modellbasierte Übertragbarkeit: Die Dokumentation von Anforderungen ist keine kritische Forderung und kann auf viele Arten realisiert werden. Es ist jedoch von Bedeutung, dass die jeweiligen Eigenschaften der Dokumentationsmöglichkeiten optimal eingesetzt werden. Durch den in dieser Arbeit genutzten Ansatz existiert eine Möglichkeit, schnell formal korrekte Anforderungen zu erzeugen. Auch die Diskussion über die Eigenschaft dynamischer Modelle als Anforderungsdokumentation eröffnen bisher nicht umgesetzte Denkanstöße.
- A4 Einbindung in einen Entwurfsprozess (V-Modell): Die Reflektion der Anforderungen entlang eines Entwurfsprozesses stellt eine bedeutende Eigenschaft des hier vorgestellten Ansatzes dar.
- A5 Ganzheitliche Betrachtung des Entwurfsprozesses: Da die Anforderungen nicht primär thematisch unterschieden werden, sondern durch ihren Detaillierungsgrad, ergeben sich Synergieeffekte zwischen den Disziplinen. Die Auseinandersetzung mit Emergenz und domänenübergreifenden Anforderungen zeigt die Fähigkeit, einen

Entwurf ganzheitlich zu spezifizieren und zwar nicht allein durch Abstraktion zu übergeordneten Zielen.

- A6 Durchgängige Berücksichtigung von Anforderungen: Die enge Kopplung der Anforderungen an die verschiedenen Entwicklungsziele der unterschiedlichen Phasen eines Entwicklungsprozesses stellt eine durchgängige Berücksichtigung von Anforderungen sicher.
- A7 Erhöhung der Konsistenz des gesamten Entwurfs: Die Übergänge der verschiedenen Phasen des Entwurfs werden eben genau mit den Anforderungen geschafft. Der besonders kritische Übergang vom Systementwurf zur spezifischen Ausarbeitung wird durch den Übergang von domänenübergreifenden zu domänenspezifischen Anforderungen unterstützt.
- A8 Bereitstellung einer Klassifikation von Anforderungen: Komplexen Problemen kann mit einer Hierarchisierung und Klassifizierung begegnet werden. Die vorgestellte Systematik beinhaltet elementare Eigenschaften der Systematizität und erfüllt daher diese Anforderung.

Insgesamt kann daher ein positives Fazit bezüglich der gestellten Anforderungen an eine Systematik gezogen werden.

7.6 Erweiterbarkeit der Lösung

Gerade bei den Analyseverfahren kann durch weitere Forschungsarbeit wertvolles Potential gehoben werden. Unwichtige Systemelemente in Bezug auf die Funktionen könnten systematisch identifiziert werden. Wenn der Anteil der erfüllten Funktionen bei Elimination bestimmter Systemelemente hinreichend hoch ist, kann hieraus ein System entstehen, welches die geforderten Funktionen maximal erfüllt. Gerade, wenn die N3 Matrizen mit gewichteten Werten arbeiten, bringen der λ-Rank und die Singulärwertzerlegung ihre Stärken ein und erlauben weitreichende Analysen. Die automatisierte Auswertung von Entwicklungsprojekten ist ein weiteres Ziel. Hier sind Korrelationsanalysen denkbar. Stellt man den Zusammenhang zwischen zwei Systemelementen einer vergangenen Entwicklung her, kann in einer zukünftigen Entwicklung diese Information nutzbringend eingesetzt werden. Wird entsprechend ein Systemelement mit einer konkreten Anforderung verknüpft, ist eine Rückmeldung denkbar, dass diese Komponente in der Vergangenheit in einer ganz bestimmten Art und Weise eingesetzt wurde, oder, dass diese Komponente oft in Kombination mit bestimmten anderen Elementen eingesetzt wurde. Solche Zusammenhänge sind nicht immer offensichtlich. In Anlehnung an die Singulärwertzerlegung kann jedoch davon ausgegangen werden, dass solche Einflüsse offengelegt werden können. Diese latenten Zusammenhänge können als Wissensmanagement gewinnbringend sein. Semantische Technologien sind in diesem Kontext besonders hilfreich [Oes17]. Für die skizzierten Szenarien sind einfach anzuwendende Toolunterstützungen zu realisieren, welche den Nutzen und die Automatisierung erhöhen.

Der Vorschlag der Aufteilung von Anforderungen in unterschiedlich abstrakte Ebenen dient der Entwicklung mechatronischer Systeme. Für jegliche Art der Systementwicklung kann das Vorgehen adaptiert werden. Die Ebenen sollten dann an typische Dekompositionsebenen angepasst werden, die in der spezifischen Entwicklung angewandt werden.

Angenommen, es würde gelingen ein streng deterministisches Umfeld zu definieren, in welchem alle Systeme kausal mit den Anforderungen zusammenhängen würden. Also ein Systemelement wird genau zu dem Systemelement, was spezifiziert ist. Dies wäre sicher die Idealvorstellung. Ist nun so ein Umfeld denkbar? Darüber lässt sich streiten. Allerdings ist die Schwierigkeit, dass ein bis auf den letzten Entwicklungsfreiheitsgrad spezifiziertes System eben keine Entwicklung mehr benötigt. Zur pragmatischen Herangehensweise kann gefordert werden, dass die Anforderungen mit dem Fortschreiten der Entwicklung immer weiter spezifiziert werden. Sodass die vollkommene Spezifikation dennoch vorliegt – nur eben nicht vor der Entwicklung. Dieser Satz an Spezifikationen könnte entsprechend der Möglichkeiten der automatisierten Auswertung wiederverwendet werden. Die Frage nach vollständiger Formalisierung der Anforderungserhebung bis hin zum Test könnte nur positiv beantwortet werden, falls es eine abgeschlossene Menge an Anforderungen gäbe. Im Sinne der Innovation und der Kreativität bleibt zu hoffen, dass dieser Schritt nicht zu schnell real wird.

Über die genannten Erweiterungsmöglichkeiten hinausgehend, liegen einige pragmatischere Ansätze der Erweiterung vor. Die Umsetzung in einer Softwarelösung erhöht die Benutzbarkeit sowohl der Synthese- als auch der Analysemöglichkeiten. Doch auch ein begleitender Einsatz in sowohl Forschungsprojekten als auch industriell geprägten Projekten wird wertvolle Erkenntnisse zu T-REQS liefern und ist in Zukunft angedacht. Die Systematik muss in der Lage sein, sich den Herausforderungen der Zeit anzupassen und daher ist sie unbedingt als wandlungsfähig anzusehen.

7.7 Resümee

Die Thematik der Anforderungen ist so umfangreich, dass eine gewisse Demut vor den Arbeiten anderer Forscher mehr als angebracht ist. Bereiche, wie die korrekte Formulierung wurden in dieser Arbeit nicht explizit behandelt und Methoden zum Test nicht allumfassend aufbereitet. Außerdem findet die Unterstützung durch Softwaretools einen lediglich geringen Anteil. Der große Anteil grundlegender Forschung zu Anforderungen im Kontext interdisziplinärer Systeme ist die Ursache dafür.

So wurden zu Beginn die grundsätzlichen Probleme bei Systementwicklungen betrachtet, die mithilfe von Anforderungen angegangen werden können. Diese umfassten Systemeigenschaften, wie Komplexität und Interdisziplinarität, aber auch prozessorientierte Eigenschaften, wie die Konsistenzprobleme in Entwicklungen. Darauf aufbauend wurden Anforderungen aus mehreren Sichten beleuchtet. Dabei wurden im Speziellen die Anwendungsbereiche Requirements Engineering und Systems Engineering untersucht und grundsätzliche Schwierigkeiten innerhalb dieser Beziehungen erarbeitet. Im weiteren Verlauf wurden aus diesen Erkenntnissen eine Anzahl von acht Anforderungen an eine Systematik extrahiert, um die Probleme systematisch zu lösen.

Der aktuelle Stand der Wissenschaft wurde daraufhin analysiert und ein Handlungsbedarf wurde ermittelt. Im Folgenden wurde eine Systematik zur Synthese und Analyse technischer Anforderungen vorgestellt, welche auf vier Anforderungsebenen beruht, die in einen Entwicklungsprozess integriert wurden. Es wurden mehrere Werkzeuge und Analysemethoden, wie die N3 Analyse vorgestellt, um Anforderungen im Kontext der Systementwicklung zu überprüfen. Die Systematik wurde anschließend mit einem Praxisbeispiel

evaluiert und die Erkenntnisse wurden in die Systematik zurückgespielt. Eine Bewertung anhand objektivierter Kriterien wurde am Ende gegeben, um Potentiale der Erweiterung der Systematik aufzuzeigen.

Das theoretische Gerüst steht damit fest, um den Herausforderungen der Entwicklung intelligenter technischer Systeme mit der Unterstützung von Anforderungen entgegenzutreten. Es bleibt zu hoffen, dass sich die T-REQS Systematik durch kontinuierliche Anwendung in der Praxis weiterentwickeln kann, um die Bandbreite der Fähigkeiten zu erhöhen. Sie soll flexibel auf sich ändernde Herausforderungen reagieren und die Entwickler nachhaltig unterstützen, sie nicht in eine Systematik zwingen. Abschließend sei angemerkt, dass die Systeme von morgen weitreichende Möglichkeiten mit sich bringen. Die Chancen dieser zukünftiger Entwicklungen sollten daher gegenüber den Herausforderungen überwiegen.

Literaturverzeichnis

[ABD+15] Adolphs, D.; Bedenbender, H.; Dirzus, D.; Ehlich, M.; Epple, U.; Hankel, M.; Heidel, R.; Hoffmeister, M.; Huhle, H.; Kärcher, B.; Koziolek, H.; Pichler, R.; Pollmeier, S.; Schewe, F.; Walter, A.; Waser, B.; Wollschlaeger, M.; Verein Deutscher Ingenieure e.V., ZVEI – Zentralverband Elektrotechnik und Elektroindustrie e.V. (Hrsg.): Referenzarchtiekturmodell Industrie 4.0 (RAMI4.0). 2015

- [Ack93] Ackermann, J.: Robuste Regelung: Analyse und Entwurf von linearen Regelungssystemen mit unsicheren physikalischen Parametern. Berlin: Springer, 1993
- [Ada18] Adamy, J.: Nichtlineare Systeme und Regelungen. Springer Vieweg, 2018
- [Adz11] Addr., G.: Specification by example: How successful teams deliver the right software. Shelter Island, N.Y: Manning, 2011
- [AFT+10] Alvarez Cabrera, A. A.; Foeken, M. J.; Tekin, O. A.; Woestenenk, K.; Erden, M. S.; Schutter, B. de; van Tooren, M.J.L.; Babuška, R.; van Houten, F.J.A.M.; Tomiyama, T.: Towards automation of control software: A review of challenges in mechatronic design. In: *Mechatronics* 20 (2010), Nr. 8, S. 876–886
- [Aka17] Akao, Y.: Quality Function Deployment. http://www.qfdi.org/. Version: 2017
- [Ale77] Alexander, C.: Center for Environmental Structure <Berkeley, Calif.>: Center for Environmental Structure series. Bd. 2: A pattern language: Towns, buildings, constructions. 19. print. New York: Oxford Univ. Press, 1977
- [And97] Andelfinger, U.: Europäische Hochschulschriften: Reihe 41, Informatik. Bd. 25: Diskursive Anforderungsanalyse: Ein Beitrag zum Reduktionsproblem bei Systementwicklungen in der Informatik. Frankfurt am Main: Lang, 1997
- [Bay14] BAYRAK, G.: Vergleich und Evaluation von Beschreibungsmitteln für die Automatisierung hybrider Prozesse, TU München, Dissertation, 2014
- [BBH94] Boehm, B.; Bose, P.; Horowitz, E. Min, J.: Software Requirements Negotiation and Renegotiation Aids:. 1994
- [Bei95] Beichelt, F.: Stochastik für Ingenieure: Eine Einführung in die Wahrscheinlichkeitstheorie und mathematische Statistik; mit zahlreichen Beispielen und Übungsaufgaben. Stuttgart: Teubner, 1995
- [Bet17] Bettenhausen, K.: Wir Ingenieure brauchen in der Digitalisierung mehr Hybridkompetenzen, https://blog.vdi.de/2017/01/wir-ingenieure-brauchen-in-der-digitalisierung-mehr-hybridkompetenzen/: VDI Verein Deutscher Ingenieure. 2017
- [BGJ09] Bertsche, B.; Göhner, P.; Jensen, U.: Zuverlässigkeit mechatronischer Systeme: Grundlagen und Bewertung in frühen Entwicklungsphasen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009

[Bir11] Birkhofer, H.: The future of design methodology. London: Springer, 2011

- [Bis02] BISHOP, R. H.: The mechatronics handbook. Boca Raton, Fla, 2002
- [Bor00] Borutzky, W.: Bondgraphen: Eine Methodologie zur Modellierung multidisziplinärer dynamischer Systeme. SCS, 2000
- [Bou56] Boulding, K.: General Systems Theory-The Skeleton of Science. In: *Management Science* 2 (1956), Nr. 3, S. 197–208
- [Bra12] Bradley, F. H.: *Essays on truth and reality*. Oxford : Cambridge Univ. Press, 2012
- [Bro87] Brooks, F.: No Silver Bullet Essence and Accidents of Software Engineering. In: *Computer* 20 (1987), Nr. 4, S. 10–19
- [CA07] Cheng, B.H.C.; Atlee, J. M.: Research Directions in Requirements Engineering. 2007 (Future of Software Engineering)
- [CAGB07] Chalmers, A. F.; Altstötter-Gleich, C.; Bergemann, N.: Wege der Wissenschaft: Einführung in die Wissenschaftstheorie. Sechste, verbesserte Auflage. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007
- [CH08] CZICHOS, H. (Hrsg.); HENNECKE, M. (Hrsg.): *Hütte das Ingenieurwissen: Mit 340 Tabellen.* 33., aktualisierte Aufl., Jubiläumsausg. 150 Jahre Hütte. Berlin : Springer [u.a.], 2008
- [CLG+09] Cheng, B. H. C.; Lemos, R. de; Giese, H.; Invardi, P.; Magee, J.; Andersson, J.; Becker, B.; Bencomo, N.; Brun, Y.; Cukic, B.; Serugendo, G.; Dustdar, S.; Finkelstein, A.; Gacek, C.; Geihs, K.; Grassi, V.; Karsai, G.; Kienle, H.; Kramer, J.; Lituio, M.; Malek, S.; Mirandola, R.; Müller, H.; Park, S.; Shaw, M.; Tichy, M.; Tivoli, M.; Weyns, D.; Whittle, J.: Software Engineering for Self-Adaptive Systems: A Research Road Map. 2009
- [DDN⁺13] Drewinski, R.; Dondit, S.; Nick, D.; Wallkamm, H.; Rybak, Heinz; Leitner, R.; Schabaker, M.; Simon, A.: *Process indicators for product engineering (PIPE): Engineering-Prozesse einheitlich bewerten.* Frankfurt am Main: VDMA-Verl., 2013
- [DIN90] DIN Deutsche Industrie Norm: DIN 25424-2:1990-04, Fehlerbaumanalyse; Handrechenverfahren zur Auswertung eines Fehlerbaumes. 1990
- [DIN06] DIN Deutsche Industrie Norm: DIN EN 60812:2006-11, Analysetechniken für die Funktionsfähigkeit von Systemen Verfahren für die Fehlzustandsart- und -auswirkungsanalyse (FMEA). 2006
- [DIN14] DIN DEUTSCHE INDUSTRIE NORM: DIN IEC 60050-351, Internationales Elektrotechnisches Wörterbuch. Berlin, September 2014
- [DIN17] DIN Deutsche Industrie Norm: DIN EN 16603-10-24:2017-11, Raumfahrttechnik Schnittstellenmanagement. 2017
- [Dud17] Duden: Ziel: Bedeutungsübersicht. https://www.duden.de/rechtschreibung/Ziel. Version: 2017
- [Dum10] Dumitrescu, R.: Entwicklungssystematik zur Integration kognitiver Funktionen in fortgeschrittene mechatronische Systeme. Paderborn, Universität Paderborn, Dissertation, 2010

[Ebe10] EBERT, C.: Systematisches requirements engineering: Anforderungen ermitteln, spezifizieren, analysieren und verwalten. 3., aktualisierte und erw. Aufl. Heidelberg: dpunkt-Verl., 2010

- [Ehr09] Ehrlenspiel, K.: Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit. 4., aktualisierte Aufl. München: Hanser, 2009
- [Eis79] EISEN, M.: LECTURE NOTES IN BIOMATHEMATICS. Bd. 30: Mathematical Models in Cell Biology and Cancer Chemotherapy. Berlin, Heidelberg, New York: Springer, 1979
- [EL15] EICKHOFF, S.; LANKEIT, C.(Betreuer): Analyse von Modellierungsansätzen mittels verschiedener Robotermodelle. Paderborn, Universität Paderborn, unveröffentlichte Bachelorarbeit, 2015
- [EL16] Ernst, F.; Lankeit, C.(Betreuer): Entwurf und Aufbau eines Sensorkonzepts für einen Greifmechanismus in einem DELTA Roboter. Paderborn, Universität Paderborn, unveröffentlichte Bachelorarbeit, 2016
- [Est07] ESTEFAN, J. A.: Survey of model-based systems engineering (MBSE) methodologies. In: *Incose MBSE Focus Group* 25 (2007), Nr. 8
- [FGS07] FRIEDENTHAL, S. (Hrsg.); GRIEGO, R. (Hrsg.); SAMPSON, M. (Hrsg.): INCOSE model based systems engineering (MBSE) initiative. 2007
- [FH17] FAIRHEAD, D.; HAVILAND, K.: Mission Control: The Unsung Heros of Apollo: Dokumentation. 2017 (Haviland Digital)
- [FHP06] Frappier, M.; Habrias, H.; Poizat, P.: A Comparison of the Specification Methods. In: Habrias, H. (Hrsg.); Frappier, M. (Hrsg.): Software Specification Methods. London, UK: ISTE, 2006, S. 351–363
- [FKL⁺13] FÖLLINGER, O.; KONIGORSKI, U.; LOHMANN, B.; ROPPENECKER, G.; TRÄCHTLER, A.: Regelungstechnik: Einführung in die Methoden und ihre Anwendung. 11. Auflage. Berlin and Offenbach: VDE Verlag GmbH, 2013
- [FKL⁺16] FÖLLINGER, O.; KONIGORSKI, U.; LOHMANN, B.; ROPPENECKER, G.; TRÄCHTLER, A.: Regelungstechnik: Einführung in die Methoden und ihre Anwendung. 12., überarbeitete Auflage. Berlin and Offenbach: VDE Verlag GmbH, 2016
- [Fos13] Fosse, E.: *Model-based Systems Engineering: Workshop MBSE 101*. Jackson-ville, USA, 26.-29.01.2013 (International Workshop)
- [Fra88] Frank, P.: Suhrkamp-Taschenbuch Wissenschaft. Bd. 734: Das Kausalgesetz und seine Grenzen. 1. Aufl. [Frankfurt am Main]: Suhrkamp, 1988
- [FT15] Fairley, R.; Towhidnejad, M.: Software engineering for systems engineers: IEEE International Symposium on Systems Engineering Rome: Tutorial. 2015
- [FV97] Fraser, M.: ; Vaishnavi, V.: A formal specifications maturity model. In: *Communications of the ACM* 40 (1997), Nr. 12, S. 95–103
- [FWFC92] Frincke, D.; Wolber, D.; Fisher, G.; Cohen, G. C.; National Aeronautics and Space Administration (Hrsg.): *Requirements Specification Language* (RSL) and Supporting Tools. 1992

[GB06] GILB, T.; BRODIE, L.: Competitive engineering: A handbook for systems engineering, requirements engineering, and software engineering using Planguage. Amsterdam, 2006

- [GB07] GAWTHROP, P. J.; BEVAN, G. P.: Bond-graph modeling: A Tutorial For Control Engineers. In: *IEEE control systems* 27 (2007), Nr. 2, S. 24–45
- [GCW⁺13] Gausemeier, J.; Czaja, A. M.; Wiederkehr, O.; Dumitrescu, R.; Tschirner, C.; Steffen, D.: Studie: Systems Engineering in der industriellen Praxis. In: 9. Paderborner Workshop: Entwurf mechatronischer Systeme (2013)
- [GK06] Greene, B.; Kober, H.: Goldmann. Bd. 15374: Das elegante Universum: Superstrings, verborgene Dimensionen und die Suche nach der Weltformel. 8. Aufl., Taschenbuchausg. München: Goldmann, 2006
- [Gli05] GLINZ, M.: Rethinking the notion of non-functional requirements. In: *Proc. Third World Congress for Software Quality* Bd. 2, 2005, S. 55–64
- [Gra98a] Grady, J.: System validation and verification. Boca Raton: CRC Press, 1998 (Systems engineering series)
- [Gra98b] Graham, I.: Requirements engineering and rapid development: An object oriented approach. Harlow: Addison-wesley, 1998
- [GTS14] Gausemeier, J.; Trächtler, A.; Schäfer, W.: Semantische Technologien im Entwurf mechatronischer Systeme: Effektiver Austausch von Lösungswissen in Branchenwertschöpfungsketten. München: Hanser, 2014
- [Han17] Hannover Messe: Digital Twin: Auf ewig verbunden. http://www.hannovermesse.de/de/news/top-themen/digital-twin/.
 Version: 2017
- [Har02] HARRER, H.: Ordnungsreduktion: Vom komplexen Strukturmodell zur vereinfachten Beschreibung technischer Systeme. München: Pflaum, 2002 (Elektround Informationstechnik)
- [Haw88] Hawking, S.: Eine kurze Geschichte der Zeit: D. Suche nach d. Urkraft d. Universums. 1. Aufl., 13. 22. Tsd. Reinbek bei Hamburg: Rowohlt, 1988
- [HB99] Hogan, N. J.; Breedveld, P. C.: The physical basis of analogies in network models of physical system dynamics, Western Multi Conference, Simulation Series Vol. 31, No.1, Proceedings 1999 International Conference on Bond Graph Modeling and, 1999, S. 96–104
- [HB16] Hartbrich, I.; Böckmann, C.: Mausklick statt Handschlag: Die Traditionsbranche steht vor der Digitalisierung. In: *VDI Nachrichten* 2016 (08.09.2016), Nr. 36. http://www.vdi-nachrichten.com/Gesellschaft/Mausklick-statt-Handschlag
- [HDS+07] Hayes, Jane H.; Dekhtyar, Alex; Sundaram, Senthil K.; Holbrook, E. A.; Vadlamudi, Sravanthi; April, Alain: REquirements Tracing On target (RETRO): improving software maintenance through traceability recovery. In: *Innovations in Systems and Software Engineering* 3 (2007), Nr. 3, S. 193–202
- [Hei17] Heinz Nixdorf Museumsforum: Ganz schön zerlegt: Die Kunst, Dinge neu zu ordnen: Ausstellung. 2017

[Hen13] Henke, C.: Betriebs- und Regelstrategien für den autonomen Fahrbetrieb von Schienenfahrzeugen mit Linearantrieb. Paderborn, Universität Paderborn, Dissertation, 2013

- [Heu06] Heuser, H.: Funktionalanalysis: Theorie und Anwendung. 4., durchges. Aufl. Wiesbaden: Teubner, 2006 (Mathematische Leitfäden)
- [HFP08] Hood, C.; Fichtinger, S.; Pautz, U.: Requirements Management: The Interface Between Requirements Development and All Other Systems Engineering Processes. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2008
- [HJD02] HULL, E.; JACKSON, K.; DICK, J.: Requirements engineering: [the latest techniques from practising requirements engineers; a flexible process for a variety of system development contexts; explains the important new concept of rich traceability]. London: Springer, 2002 (Practitioner series)
- [HJD05] HULL, E.; JACKSON, K.; DICK, J.: *Requirements Engineering*. Second Edition. London: Elizabeth Hull Ken Jackson Jeremy Dick, 2005
- [HJD11] HULL, E.; JACKSON, K.; DICK, J.: Requirements Engineering. Third Edition. London: Springer London, 2011
- [HMH⁺18] Holtkötter, J.; Michael, J.; Henke, C.; Trächtler, A.; Bockholt, M.; Möhlenkamp, A.; Katter, M: Rapid-Control-Prototyping as part of Model-Based Development of Heat Punp Dryers. In: *4th International Conference on System-Integrated Intelligence*, 2018
- [HMLT17] HENKE, C.; MICHAEL, J.; LANKEIT, C.; TRÄCHTLER, A.: Virtuelle Inbetriebnahme eines Fertigungszentrums. Paderborn, 08.-10. November 2017
- [HV03] Herstatt, C.; Verworn, B.: Management der frühen Innovationsphasen: Grundlagen Methoden neue Ansätze. 1. Aufl. Wiesbaden, 2003
- [HW12] Herwig, H.; Wenterodt, T.: Entropie für Ingenieure: Erfolgreich das Entropie-Konzept bei energietechnischen Fragestellungen anwenden. Wiesbaden: Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, 2012
- [HWF+12] Haberfellner, R.; Weck, O. de; Fricke, E.; Vössner, S.; Büchel, A.; Nagel, P.; Massow, H. von; Becker, M.: Systems Engineering: Grundlagen und Anwendung. 12., völlig neu bearb. und erw. Aufl. Zürich: Orell Füssli, 2012
- [IEE11] IEEE: ISO/IEC/IEEE 29148:2011(E), Systems and software engineering Life cycle processes Requirements engineering. (2011)
- [Ill14] ILLG, I.: Mechatronischer Entwurf und Erprobung einer regelbaren Federung für ein leichtes geländegängiges Kettenfahrzeug, Universität Paderborn, Dissertation, 2014
- [Ise92] ISERMANN, R.: *Identifikation dynamischer Systeme 1: Grundlegende Methoden.* Zweite neubearbeitete und erweiterte Auflage. Berlin and Heidelberg: Springer, 1992 (Springer-Lehrbuch)
- [ISO08] ISO International Organization for Standardization: ISO 15288, Systems and software engineering —System life cycle processes. 2008. 2008

[ISO11a] ISO International Organization for Standardization: *IEEE 29148: 2011-*Systems and software engineering-Requirements engineering. 01.01.2011

- [ISO11b] ISO International Organization for Standardization: ISO 25000, Software engineering Software product Quality Requirements and Evaluation (SQua-RE) Guide to SQuaRE. 2011
- [ISO11c] ISO International Organization for Standardization: ISO 26262, Road vehicles Functional safety. 2011
- [ISO12] ISO International Organization for Standardization: ISO 15504, Software Process Improvement and Capability Determination. 2012
- [ISO13] ISO International Organization for Standardization: ISO 29119 Software Testing. 2013
- [IST20] ISTQB International Software Testing Qualifications Board: ISTQB. https://www.istqb.org/. Version: 2020
- [Jam09] Jamshidi, M.: Systems of systems engineering: Principles and applications. Boca Raton, FL: CRC Press/Taylor & Francis Group, 2009
- [Jan10] Janschek, K.: Systementwurf mechatronischer Systeme: Methoden, Modelle, Konzepte. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010
- [JJPB07] JOHNSON, T.; JOBE, J.; PAREDIS, C.; BURKHART, R.: Modeling Continuous System Dynamics in SysML. 2007 (ASME International Mechanical Engineering Congress and Exposition)
- [Jun06] Jung, C.: Anforderungsklärung in interdisziplinärer Entwicklungsumgebung. München, Technische Universität München, Dissertation, 2006
- [Jus14] Just, V.: Modellbasierte Analyse und regelungstechnische Maßnahmen zur Optimierung des Vereinzelungsvorgangs in Bankautomaten. Paderborn, Universität Paderborn, Dissertation, 2014
- [KFG07] Krause, F.; Franke, H.; Gausemeier, J.: Innovationspotenziale in der Produktentwicklung. München: Hanser, 2007
- [Kir11] Kirstan, S.: Kosten und Nutzen modellbasierter Entwicklung eingebetteter Softwaresysteme im Automobil. München, TU München, Dissertation, 2011
- [Kno17] Knoop, S.: Flachheitsbasierte Positionsregelungen für Parallelkinematiken am Beispiel eines hochdynamischen hydraulischen Hexapoden. Paderborn, Universität Paderborn, Dissertation, 2017
- [Kot11] Kothes, L.: Grundlagen der Technischen Dokumentation: Anleitungen verständlich und normgerecht erstellen. Berlin, Heidelberg : Springer-Verlag Berlin Heidelberg, 2011 (VDI-Buch)
- [KS11] Kossiakoff, A.; Sweet, W. N.: Wiley series in systems engineering and management. Bd. 67: Systems engineering: Principles and practice. 2nd ed. Hoboken, N.J: Wiley-Interscience, 2011
- [Lan01] Landwehr, M.: Fortschritt-Berichte VDI: Reihe 8, Meß-, Steuerungs- und Regelungstechnik. Bd. 894: Labor- und Simulationswerkzeuge für die Analyse mechatronischer Systeme in der Feinwerktechnik. Als Ms. gedr. Düsseldorf: VDI-Verl., 2001

[Lan17] Lander, A.: Der Ingenieur in der digitalen Transformation, das unbekannte Wesen? VDI Blog. https://blog.vdi.de/2017/01/ingenieur-in-der-digitalen-transformation/. Version: 2017

- [LBEM00] Lozano, R.; Brogliato, B.; Egeland, O.; Maschke, B.: Dissipative systems analysis and control: Theory and applications. London: Springer, 2000 (Communications and control engineering)
- [Lin93] LINDSTROM, D. R.: Five ways to destroy a development project (software development). In: *IEEE Software* 10 (1993), Nr. 5, S. 55–58
- [LJT16] LANKEIT, C.; JUST, V.; TRÄCHTLER, A.: Consistency Analysis for Requirements, Functions, and System Elements. In: *IEEE Systems Conference*. Orlando, USA, 2016
- [LKS00] Lückel, J.; Koch, T.; Schmitz, J.: *Mechatronik als integrative Basis fur innovative Produkte*. Nichtred. Ms.-Dr. Düsseldorf, 2000 (VDI-Berichte)
- [LLLH15] Lochbichler, M.; Lankeit, C.; Landwehr, M.; Hoischen, L.: Device for Separating Sheet Material: WO 2016/001011 A1. 2015 (Patent)
- [LLO+15] Lankeit, C.; Lochbichler, M.; Oestersötebier, F.; Trächtler, A.; Landwehr, M.: From goals to systems via technical requirements and physical models: design of an intelligent cash handling system. 2015
- [LLT15] Lankeit, C.; Lochbichler, M.; Trächtler, A.: Challenges in Requirements Engineering for Mechatronic Systems Problem Analysis and First Approach. In: *Jurnal Teknologi* 76 (2015), Nr. 4
- [Loc20] Lochbichler, M.: Systematische Wahl einer Modellierungstiefe im Entwurfsprozess mechatronischer Systeme. Paderborn, Universität Paderborn, Dissertation, 2020
- [LOT14] Lochbichler, M.; Oestersötebier, F.; Trächtler, A.: *Dynamic Behavior Models and their Modeling Depth in the Design Process of Mechatronic Systems*. 2014 (ASME 2014 Proceedings of the ASME 2014)
- [Lud17] Ludwig, H.: Joint Development Projects with Bentley Systems. Singapur, 10.-12.10.2017
- [Lun10] Lunze, J.: Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010 (Springer-Lehrbuch)
- [MAW97] MITCHELL, R.; AGLE, B.; WOOD, D.: Toward a Theory of Stakeholder Identification and Salience: Defining the Principle of Who and What Really Counts. 1997
- [MBK15] Meffert, H.; Burmann, C.; Kirchgeorg, M.: Marketing: Grundlagen marktorientierter Unternehmensführung Konzepte - Instrumente - Praxisbeispiele. 12., überarb. u. aktualisierte Aufl. 2014. Wiesbaden: Springer Fachmedien Wiesbaden, 2015
- [MHDZ07] MÜLLER, M.; HÖRMANN, K.; DITTMANN, L.; ZIMMER, J.: Automotive SPI-CE in der Praxis: Interpretationshilfe für Anwender und Assessoren. 1. Aufl. Heidelberg: dpunkt. Verl., 2007

[ML16] Malsam, P.; Lankeit, C.(Betreuer): Modellbasierter Entwurf eines Regelungskonzepts für ein Abzugswalzenpaar mit Drehmomentregelung. Paderborn, Universität Paderborn, unveröffentlichte Studienarbeit, 2016

- [MRGL13] McDermott, T.; Rouse, W.; Goodman, S.; Loper, M.: Multi-level Modeling of Complex Socio-Technical Systems. In: *Procedia Computer Science* 16 (2013), S. 1132–1141
- [MWB+03] Muhs, D.; Wittel, H.; Becker, M.; Jannasch, D.; Vossiek, J.: *Roloff/Matek Maschinenelemente Formelsammlung*. 7., verb. Aufl. 2003
- [Nat07] National Aeronautics and Space Administration: *NASA Systems Engineering Handbook*. Washington, D.C, 2007
- [NB12] Niknafs, A.; Berry, D. M.: The impact of domain knowledge on the effectiveness of requirements idea generation during requirements elicitation. 2012
- [NC97] NATKE, H.; CEMPEL, C.: Model aided diagnosis of mechanical systems: Fundamentals, detection, localization, assessment. Berlin: Springer, 1997
- [Neu06] Neugebauer, R.: Parallelkinematische Maschinen: Entwurf, Konstruktion, Anwendung. Springer Berlin Heidelberg, 2006
- [New10] Newton, I.: *Philosophiæ naturalis principia mathematica*. [Nachdruck]. [Seaside]: Watchmaker, 2010
- [NLF⁺15] Nielsen, C. B.; Larsen, P. G.; Fitzgerald, J.; Woodcock, J.; Peleska, J.: Systems of Systems Engineering. In: *ACM Computing Surveys* 48 (2015), Nr. 2, S. 1–41
- [No 18] No Magic: Cameo Systems Modeler. https://www.nomagic.com/products/cameo-systems-modeler. Version: 2018
- [Noe18] Noether, E.: Invarianten beliebiger Differentialausdrücke. In: *Nachr. Ges. Wiss. Göttingen, Math.-Phys. Kl.* 1918 (1918), S. 37–44
- [Oes17] Oestersötebier, F.: Modellbasierter Entwurf intelligenter mechatronischer Systeme mithilfe semantischer Technologien. Paderborn, Universität Paderborn, Dissertation, 2017
- [OKT+16] Olma, S.; Kohlstedt, A.; Traphöner, P.; Jäker, K.; Trächtler, A.: Substructuring and Control Strategies for Hardware-in-the-Loop Simulations of Multiaxial Suspension Test Rigs. 2016 (Proceedings of the 7th IFAC 2016)
- [OMG13] OMG OBJECT MANAGEMENT GROUP: Requirements Interchange Format (ReqIF). 2013
- [Orb10] Orbak, A.: *Physical model reduction: A bond graph approach for engineering systems*. Saarbrücken: Lambert Academic Publ, 2010
- [OTB+15] Otter, M.; Thuy, N.; Bouskela, D.; Buffoni, L.; Elmqvist, H.; Fritzson, P.; Garro, A.; Jardin, A.; Olsson, H.; Payelleville, M.; Schamai, W.; Thomas, E.; Tundis, A.: Formal Requirements Modeling for Simulation-Based Verification. In: *The 11th International Modelica Conference*, Linköping University Electronic Press, 2015 (Linköping Electronic Conference Proceedings), S. 625–635

- [Pal86] PALM, W.: Control systems engineering. New York: Wiley, 1986
- [Par88] PARKER, S.: Acoustics source-book. 1. [Dr.]. New York: McGraw-Hill, 1988 (The McGraw-Hill science reference series)
- [Par98] Partsch, H.: Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme. Berlin: Springer, 1998
- [Par10] Partsch, H.: Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010
- [PB77] PAHL, G.; BEITZ, W.: Konstruktionslehre: Handbuch für Studium und Praxis. Berlin: Springer, 1977
- [PB98] Page, L.; Brin, S.: The Anatomy of a Large-Scale Hypertextual Web Search Engine. 1998
- [PBFG13] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.: *PahlBeitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung*. 8., vollst. überarb. Aufl. Berlin: Springer Vieweg, 2013
- [Pfi17] Pfingsten, M.: Zukunftsarchitekten Podcast. http://zukunftsarchitekten-podcast.de/. Version: 2017
- [Pid96] Pid9. M.: Five Simple Principles of Modelling: WSC '96 Proceedings of the 28th conference on Winter simulation. 1996
- [Poh10] Pohl, K.: Requirements engineering: Fundamentals, principles, and techniques. Berlin: Springer, 2010
- [PS07] POHL, K.; SIKORA, E.: COSMOD-RE: Supporting the Co-Design of Requirements and Architectural Artifacts: 15th IEEE International Requirements Engineering Conference. 2007
- [PTC17] PTC Integrity: PTC Integrity. http://www.ptc-de.com/application-lifecycle-management/integrity. Version: 2017
- [PvJV04] Pinkster, I.; van de Burgt, B.; Janssen, D.; Veenendaal, E.: Successful test management: An integral approach; with 44 tables. Berlin: Springer, 2004
- [PW85] Parnas, D.; Weiss, D.: Active design reviews: principles and practices. In: *Proceedings of the 8th international conference on Software engineering*, 1985, S. 132–136
- [PW88] PAPALAMBROS, P.; WILDE, D.: Principles of optimal design: Modeling and computation. Cambridge: Cambridge University Press, 1988
- [PZ05] PLOCK, C.; ZUCK, L.: From Requirements to Specifications: Proceedings of the 12th IEEE International Conference and Workshops. 2005
- [Rau92] RAUTERBERG, M.: Partizipative Modellbildung zur Optimierung der Softwareentwicklung. In: Brauer, W. (Hrsg.); Studer, R. (Hrsg.): *Informationssysteme und Künstliche Intelligenz: Modellierung* Bd. 303. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, S. 113–128
- [Res87] Rescher, N.: *Induktion: Zur Rechtfertigung induktiven Schlieβens*. München : Philosophia-Verlag, 1987 (Introductiones)

[Rey96] REYNOLDY, M.: *Test and evaluation of complex systems*. Chichester: Wiley, 1996 (Wiley series in measurement science and technology)

- [Rie98] RIEGE, B.: Strukturmaße für dynamische Systeme, Gerhard-Mercator-Universität Duisburg, Forschungsbericht, 1998
- [RLU14] RAJAMAN, A.; LESKOVEC, J.; ULLMAN, J.: Mining of Massive Datasets. Palo Alto, USA, 2014
- [RN04] Russel, S.; Norvig, P.: Künstliche Intelligenz: Ein moderner Ansatz. 2. Aufl. München: Pearson Studium, 2004 (Informatik: Künstliche Intelligenz)
- [Rod13] Roddeck, W.: Grundprinzipien der Mechatronik: Modellbildung und Simulation mit Bondgraphen. Wiesbaden: Springer Fachmedien Wiesbaden, 2013
- [RPC+12] Reichwein, A.; Paredis, C.; Canedo, A.; Witschel, P.; Stelzig, P.; Votintseva, A.; Wasgint, R.: Maintaining consistency between system architecture and dynamic system models with SysML4Modelica. 2012
- [RR13] ROBERTSON, S.; ROBERTSON, J.: *Mastering the requirements process: Getting requirements right.* 3rd ed. Upper Saddle River, N.J.: Addison-wesley, 2013
- [RS10] RILL, G.; Schaeffer, T.: Grundlagen und Methodik der Mehrkörpersimulation. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, 2010
- [Rup07] Rupp, C.: Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis. 4., aktualisierte und erw. Aufl. München: Hanser, 2007
- [Rup09] Rupp, C.: Requirements-Engineering und -Management: Professionelle, iterative Anforderungsanalyse für die Praxis. 5., aktualisierte und erw. Aufl. München: Hanser, 2009
- [Rze03] Rzevski, G.: On Conceptual Design of Intelligent Mechatronic Systems: Mechatronics (13), Elsevier Ltd. 2003
- [Saa90] SAATY, T. L.: How to make a decision: the analytic hierarchy process. In: European journal of operational research 48 (1990), Nr. 1, S. 9–26
- [SB13] Schneider, F.; Berenbach, B.: A Literature Survey on International Standards for Systems Requirements Engineering. In: *Procedia Computer Science* 16 (2013), S. 796–805
- [SBB12] Seidl, R.; Baumgartner, M.; Bucsics, T.: Basiswissen Testautomatisierung: Konzepte, Methoden und Techniken. 1. Aufl. Heidelberg: dpunkt-Verl., 2012
- [SBW⁺10] Sawyer, P.; Bencomo, N.; Whittle, J.; Letier, E.; Finkelstein, A.: Requirements-Aware Systems: A research agenda for RE for self-adaptive systems. 2010
- [Sch96] Schwab, A.: *Elektromagnetische Verträglichkeit*. Vierte, neubearbeitete Auflage. Berlin, Heidelberg and s.l.: Springer Berlin Heidelberg, 1996
- [Sch16] Schweers, C.: Adaptive Sigma-Punkte-Filter-Auslegung zur Zustands- und Parameterschätzung an Black-Box-Modellen: Dissertation. Universität Paderborn. 2016

[Sch17] SCHMITZ, W.: Tablet statt Tafel. In: VDI Nachrichten 2017 (15.09.2017), Nr. 37

- [Sha75] Shannon, R.: Systems simulation: The art and science. Englewood Cliffs, NJ: Prentice-Hall, 1975
- [Sim73] Simon, H. A.: The structure of ill structured problems. In: *Artificial Intelligence* 4 (1973), Nr. 3-4, S. 181–201
- [Spa17] Sparx Systems: *Enterprise Architect*. https://www.sparxsystems.de/start/startseite/. Version: 2017
- [Spi18] Spiegel Online: Weniger Geldautomaten in Deutschland, http://www.spiegel.de/wirtschaft/service/warum-geldautomaten-in-deutschland-an-bedeutung-verlieren-a-1203016.html. 2018
- [SRE14] SCHMIDT, E.; ROSENBERG, J.; EAGLE, A.: How Google works. 1st ed. New York, NY: Grand Central Publ, 2014
- [SRWL08] SPILLNER, A.; ROSSNER, T.; WINTER, M.; LINZ, T.: *Praxiswissen Softwaretest Testmanagement: Aus- und Weiterbildung zum Certified Tester Advanced Level nach ISTQB-Standard*. 2., überarb. und aktualisierte Aufl. Heidelberg: dpunkt-Verl., 2008
- [SSB10] Sneed, H.; Seidl, R.; Baumgartner, M.: Software in Zahlen: Die Vermessung von Applikationen. München: Hanser, 2010
- [SSS15] Steininger, U.; Schöner, H.; Schiementz, M.: Requirements on tools for assessment an validation of assisted and automated driving systems. München, 25.-26.11.2015
- [Sta83] Stachowiak, H.: Modelle: Konstruktion der Wirklichkeit. München, 1983
- [Ste11] Steffenhagen, B.: *Kleine Formelsammlung Regelungstechnik*. München: Fachbuchverl. Leipzig im Carl-Hanser-Verl., 2011
- [Sti17] STIMULUS: Stimulus. https://argosim.com/. Version: 2017
- [TG18] Trächtler, A.; Gausemeier, J.: Steigerung der Intelligenz mechatronischer Systeme. Springer Vieweg, 2018
- [Tru50] TRUXAL, J. G.: Servomechanism Synthesis Through Pole-Zero Configurations. In: *Technical Report Research Laboratory of Electronics, Masachusetts Institute of Technology* (1950), Nr. 162
- [Tru58] Truxal, J. G.: Control Engineers' Handbook: ServoMechanisms, Regulators, And Automatic Feedback Control Systems. McGraw-Hill Book Company, 1958
- [Tru60] Truxal, J. G.: Entwurf automatischer Regelsysteme: Signalflußdiagramme und Theorie der Rückkopplung, Netzwerktheorie, Analyse und Synthese von linearen, nichtlinearen und diskontinuierlichen Regelkreisen; Anwendung statistischer Methoden. Oldenbourg, 1960
- [Ulr92] Ulrich, K.: The Role of Product Architecture in the Manufacturing Firm. Cambridge, USA, 1992

[Unb08] Unbehauen, H.: Regelungstechnik I: Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme, Fuzzy-Regelsysteme. 15., überarbeitete und erweiterte Auflage. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, 2008

- [Uys16] Uysal, M. P.: In search of Software Engineering Foundations: A Theoretical and Trans-disciplinary Perspective. 2016 (International Journal of Computer Theory and Engineering)
- [van01] van Lamsweerde, A.: Goal-oriented requirements engineering: A guided tour. 2001 (Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on Requirements Engineering)
- [VC98] Vischer, P.; Clavel, R.: Kinematic calibration of the parallel Delta robot: Robotica. 1998
- [VDA17] VDA VERBAND DER AUTOMOBILINDUSTRIE: Automotive SPICE Process Assessment / Reference Model. 2017
- [VDI93] VDI VEREIN DEUTSCHER INGENIEURE: VDI 2221, Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. 1993
- [VDI97] VDI Verein Deutscher Ingenieure: VDI 2222, Konstruktionsmethodik Methodisches Entwickeln von Lösungsprinzipien. 1997
- [VDI04a] VDI Verein Deutscher Ingenieure: VDI 2206, Entwicklungsmethodik für mechatronische Systeme. 2004
- [VDI04b] VDI Verein Deutscher Ingenieure: VDI 2223, Methodisches Entwerfen technischer Produkte. 2004
- [WAC12] WHITTAKER, J.; ARBON, J.; CAROLLO, J.: How Google tests software: [help me test like Google; life of a TE; life of an SET; interviews with Googlers and more]. Upper Saddle River, NJ: Addison-wesley, 2012
- [WC09] Walia, G. S.; Carver, J. C.: A systematic literature review to identify and classify software requirement errors. In: *Information and Software Technology* 51 (2009), Nr. 7, S. 1087–1109
- [Web71a] Weber, W.: Methoden der Regelungstechnik: Allgemeine Struktur und Erkennungsmethoden: Adaptive Regelungssysteme I. 1971
- [Web71b] Weber, W.: Methoden der Regelungstechnik: Entscheidungsprozesse und Anwendungsbeispiele: Adaptive Regelungssysteme II. 1971
- [Wei14] Weilkiens, T.: Systems engineering mit SysMLUML: Anforderungen, Analyse, Architektur. 3., überarb. und aktualisierte Aufl. Heidelberg: dpunkt.Verl., 2014
- [Wie14] Wielenberg, A.: Entwurf mechatronischer Fahrzeugfederungen am Beispiel eines geländegängigen Nutzfahrzeugs, Universität Paderborn, Dissertation, 2014
- [Wor15] World Economic Forum: Industrial Internet of Things: Unleashing the Potential of Connected Products and Services. 2015

[WRF⁺15] Walden, D.; Roedler, G.; Forsberg, K.; Hamelin, R.; Shortell, T.: *INCO-SE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Fourth Edition. 2015

- [YZ80] Yeh, R. T.; Zave, P.: Specifying software requirements. In: *Proceedings of the IEEE* 68 (1980), Nr. 9, S. 1077–1085
- [ZJ97] ZAVE, P.; JACKSON, M.: Four dark corners of requirements engineering. In: *ACM Transactions on Software Engineering and Methodology* 6 (1997), Nr. 1, S. 1–30
- [ZS10] ZESEVIC, A.; SILJAK, D.: Control of complex systems: Structural constraints and uncertainty. New York: Springer, 2010 (Communications and control engineering)

Anhang

Inhaltsverzeichnis

Α1	Begriffe und Zusammenhänge	9
	A1.1 Industrie 4.0 Terminologie	9
	A1.2 Zusammenhänge einiger Normen im RE	1
A2	Übersicht ausgewählter Methoden	3
	A2.1 Methoden zur Ermittlung von Zielen	3
	A2.2 Methode zur Darstellung des Lastenheftes	4
А3	Mathematische Grundlagen des Kompensationsreglers	5
Α4	Elemente der Bondgraphen	7
	Elemente der Bondgraphen 18 Ergänzende Bilder der Anwendung 19	
		1

A1 Begriffe und Zusammenhänge

A1.1 Industrie 4.0 Terminologie

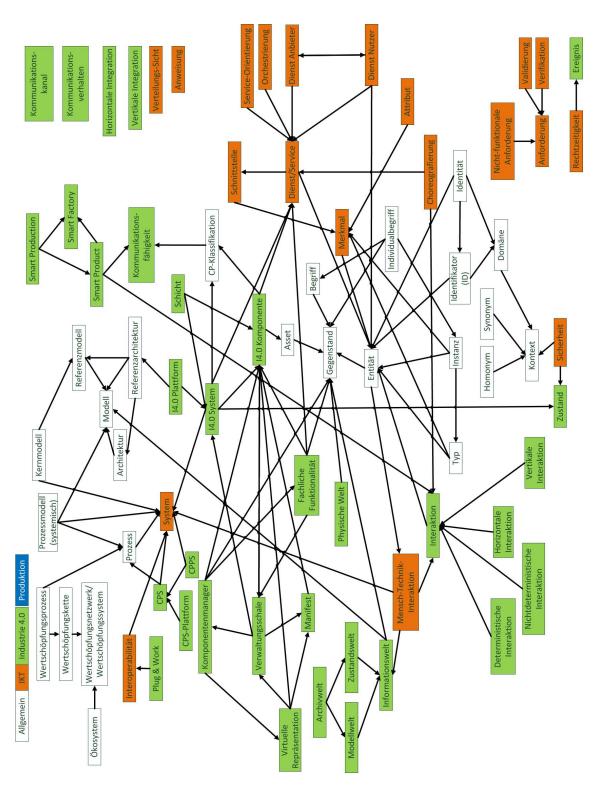


Bild A1-1: Begriffe im Themenfeld Industrie 4.0. Anforderungen stehen auffallend isoliert. Quelle: https://www.iosb.fraunhofer.de/servlet/is/48960/

A1.2 Zusammenhänge einiger Normen im RE

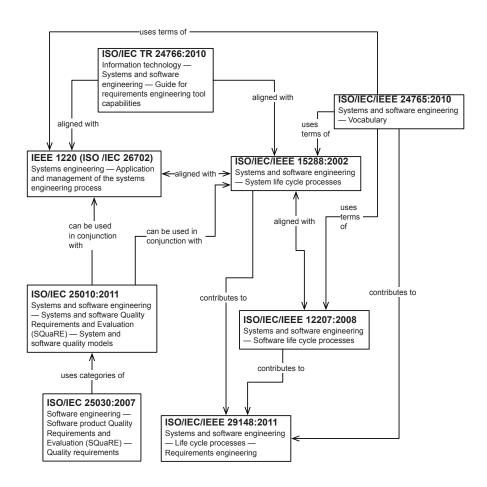


Bild A1-2: Einige Normen im Kontext des Requirement Engineering und ihre Zusammenhänge [SB13]

A2 Übersicht ausgewählter Methoden

A2.1 Methoden zur Ermittlung von Zielen

	Brainstorming	Brainstorming paradox	Methode 6-3-5	Wechsel d. Perspektive	Walt Disney Methode	Bionik / Bisoziation	Osbom Checkliste	Feldbeobachtung	Apprenticing	SOPHIST-Regelwerk	Fragebogen	Interview	Selbstaufschreibung	On-Site-Customer	Systemarchäologie	Reuse	Workshop	Mind Mapping	CRC-Karten	Snowcard	Audio-Aufzeichnung	Video-Aufzeichnung	Anwendungsfälle	Essenzbildung	Anforderungen erahnen
Mensch																									
geringe Motivation	-						-	+	-	0	+	+			0	0	+	-	0	+	0	+	0	0	++
schlechte kommunikative Fähigkeiten	-	-	-	-	0	0	-	-	++	++	0	-		-	0	0	-	0	-	-	0	0	+	0	+
implizites Wissen	+	++	++	+	+	+	+	++	++	++		0		+	0	0	0	+	0	0	0	+	0	0	++
geringes Abstraktionsvermögen	-		-	-	-	++	-	++	++	+	+	+	-	-	0	0	0	-	-	-	0	0	-	++	+
divergierende Stakeholdermeinungen	-	+	-	-	-	0	+	-	-	+	++	0	-		0	0	+	-	+	+	0	0	+	++	0
problematische Gruppendynamik			+	0		0	+		++	-	0	0	+	+	0	0		-	-	0			0	0	0
Organisatorische Rahmenbedingunger	n																								
Neuentwicklung	++	+	++	++	++	+	+	0	0	0	+	+	+	+		0	+	++	++	++	0	0	0	+	+
Altsystemerweiterung	0	+	0	0	0	0	++	+	+	0	+	+	+	+	++	+	+	0	+	+	0	0	0	0	+
Individualentwicklung	0	0	0	0	0	0	+	+	+	0	+	+	++	++	+	0	+	0	+	0	0	0	0	0	+
Produktentwicklung	++	+	++	++	++	+	+	-		0	+	-	-		+	0	0	++	0	-	-	-	0	0	+
fixiertes, knappes Projektbudget	0	0	-	0	0		-	-		+	++	+	+	-		++	0	0	0	0	++	-	0	0	++
hohe Verteilung d. Stakeholder				-	-	-	0	0	0	0	+	-	++	-	0	0		-			0	0	0	0	0
schlechte Verfügbarkeit d. Stakeholder	-	-	+	-	-	-	-	++		0	+	-			0	++		-			++	++	0	0	++
hohe Zahl von Stakeholdern	0	-	-	0	0	-	-	-		0	++				0	0		0			-	-	+	+	0
Fachlicher Inhalt der Anforderungen																									
hohe Kritikkalität des Systems	0	0	+	+	+	-	+	+		++	+	+	+	++	++	-	+	0	+	+	++	+	0	+	0
großer Systemumfang	0	0	0	0	0	0		++	+	0		+	+	+	+	++	+	0	-	+	0	0	++	++	+
hohe Komplexität der Systemabläufe	+	-	0	+	+	+	+			0		+	+	+	+	+	+	++	0	0	0	0	+	0	0
geringe Beobachtung	+	0	0	+	0	++	+		+	0	+	+	+	+	+	+	+	+	0	0	0		0	+	0
nicht funktionale Anforderungen	-	-	+	+	0	+	0	-	+	0		-	-	-	+	+	-	0	+	0	0	+	0	0	0
unbekanntes Fachgebiet	0	0	+	0	-	-	0	+	++	0		+	++	+	++		+	0	0	0	0	0	+	0	
abstrakte Anforderung	+	+	+	+	+	+	++		-	+	++	++	+	+	-	0	+	+	++	++	0	0	++	++	-
detaillierte Anforderung	-	-	-	-	-	-	0	+	++	++	0	++	+	++	++	++	0	-	-	-	+	+	-	0	++

Bild A2-1: In diesem Abschnitt werden eine Reihe an Methoden aufgelistet, um Ziele zu ermitteln. Sicherlich können so auch spezifischere Anforderungen ermittelt werden, jedoch stößt die Ermittlung mit diesen Methoden auf gewisse Grenzen. Eine Auswahl von Methoden und ihre Eignung zur Ermittlung von Zielen oder Anforderungen im weitesten Sinne wird gezeigt, abgewandelt nach [Rup07]. Ein Plus steht für eine gute Eignung, ein Minus entsprechend für eine schlechte.

A2.2 Methode zur Darstellung des Lastenheftes

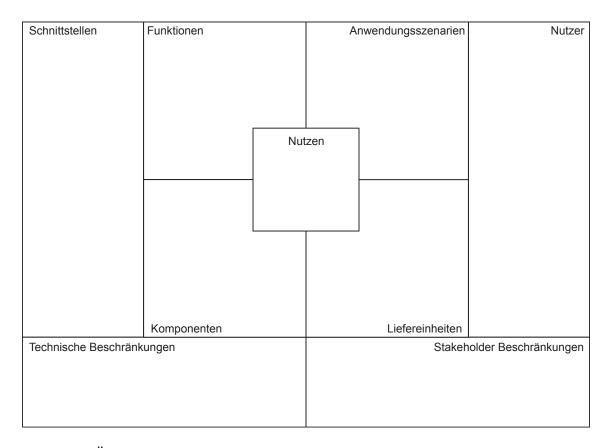


Bild A2-2: Übersichtliche grafische Anordnung des Inhalts eines Lastenheftes, abgewandelt nach [Pfi17]

A3 Mathematische Grundlagen des Kompensationsreglers

Um einen universell einsetzbaren Regler entwerfen zu können, wird ein Regler der folgenden Form genutzt

$$G_k(s) = \frac{F_o(s)^{-1} Z_w(s)}{N_w(s) - Z_w(s)} = \frac{N_o(s) Z_w(s)}{Z_o(s) [N_w(s) - Z_w(s)]} = \frac{Z_k(s)}{N_k(s)}.$$

An dieser Stelle sei darauf hingewiesen, dass beide Typen des Kompensationsreglers als

"...eine explizite Darstellung (Parametrisierung) der Klasse aller stabilisierender Regler, die sogenannte Youla-Parametrisierung" [FKL+13, S. 215]

interpretiert werden können. Regler können also auch komplexe Systeme in einen stabilen Zustand überführen. Das Ziel ist der Entwurf eines universell einsetzbaren Reglers. Daher dürfen Pole und Nullstellen rechts oder auf der j-Achse nicht (weg-)kompensiert werden. Die Grundlagen entstammen der Literatur [FKL+13]. Es soll gelten:

Grad
$$Zk \leq \text{Grad } N_k$$
.

Zunächst wird dann ein Zählerpolynom zu z.B.

$$F_w(s) = \frac{1}{N_w(s)} = \frac{1}{[1 + \frac{s}{D}]^r}$$

oder

$$F_w = \frac{1}{N_w(s)} = \frac{1}{\frac{1}{(1+\alpha^2)D_2} \left[1 + \frac{s}{5D}\right]^k \left[s^2 + 2Ds + (1+\alpha^2)D^2\right]}$$

hinzugefügt, also:

$$F_w = \frac{Z_w(s)}{N_w(s)}.$$

Damit ist der Kompensationsregler wie folgt modifiziert:

$$G_k(s) = \frac{F_o(s)^{-1} Z_w(s)}{N_w(s) - Z_w(s)} = \frac{N_o(s) Z_w(s)}{Z_o(s) [N_w(s) - Z_w(s)]} = \frac{Z_k(s)}{N_k(s)}$$

mit

Grad
$$N_w$$
 – Grad $Z_w \ge$ Grad N_o – Grad Z_0 .

Zähler und Nenner der Strecke werden nun in jeweils zwei Faktoren zerlegt:

$$Z_o(s) = Z_{ou}(s)Z_{ok}(s) = \{\prod_{u=1}^{r_1} (s - n_{u,\mu})\}Z_{ok}(s),$$

$$N_o(s) = N_{ou}(s)N_{ok}(s) = \{ \prod_{v=1}^{r_2} (s - p_{u,v}) \} N_{ok}(s).$$

Dabei enthalten $Z_{ou}(s)$ und $N_{ou}(s)$ die r1 Nullstellen $n_{u,\mu}$ bzw. die r2 Pole $p_{u,\nu}$, welche nicht kompensiert werden sollen. Damit diese nicht kompensiert werden, muss

$$Z_w(s) = Z_{ou}(s)\widetilde{Z}(s) = Z_{ou}(s)\widetilde{b_{\tilde{m}}}s^{\tilde{m}} + \dots + \widetilde{b_1}s + \widetilde{b_0}$$

und

$$N_w(s) - Z_w(s) = N_{ou}(s)\widetilde{N}(s) = N_{ou}(s)\widetilde{a_n}s^{\tilde{n}} + ... + \widetilde{a_1}s + \widetilde{a_0}$$

sein. Es ergibt sich:

$$G_k(s) = \frac{N_{ok}(s)\tilde{Z}(s)}{Z_{ok}(s)\tilde{N}(s)} = \frac{Z_k(s)}{N_k(s)}$$

und

$$N_w(s) = Z_{ou}(s)\tilde{Z} + N_{ou}(s)\tilde{N}.$$

Diese Identität kann durch das gewünschte Nennerpolynom

$$N_w(s) = c_{n_w} s^{n_w} + \dots + c_1 s + c_0$$

und geeignete Wahl von

$$\widetilde{Z}(s) = \widetilde{b_m} s^{\widetilde{m}} + ... + \widetilde{b_1} s + \widetilde{b_0} \text{ und } \widetilde{N}(s) = \widetilde{a_n} s^{\widetilde{n}} + ... + \widetilde{a_1} s + \widetilde{a_0}$$

erfüllt werden. Wenn $Z_o(s)$ und $N_ou(s)$ teilerfremd sind, sind auch $Z_ou(s)$ und $N_o(s)$ teilerfremd. Damit besitzt

$$N_w(s) = Z_{ou}(s)\tilde{Z} + N_{ou}(s)\tilde{N}$$

immer eine Lösung, wenn

$$n_w \leq \tilde{n} + \tilde{m} + 1$$

ist. Dann lässt sich ein Gleichungssystem aufstellen, in welchem $\widetilde{a_i}$ und $\widetilde{b_i}$ ermittelt werden können. Der eindeutig bestimmte Regler, welcher der Führungsübertragungsfunktion das gewünschte Nennerpolynom $N_w(s)$ verleiht lautet also:

$$G_k(s) = \frac{N_{ok}(s)\tilde{Z}(s)}{Z_{ok}(s)\tilde{N}(s)} = \frac{Z_k(s)}{N_k(s)}.$$

A4 Elemente der Bondgraphen

Zum Verständnis der Elemente der Bondgraphen (Die hier beschriebenen Grundlagen über Bondgraphen wurden mit Unterstützung von Eickhoff aufgearbeitet [EL15]) sind einige Gedanke vorab hilfreich. Bei der Integration der Leistung P muss je nach Speichertyp zwischen dem Integrieren der e- oder f-Variable unterschieden werden. Daher ergeben sich zwei weitere generalisierte Variablen, einerseits der generalisierte Impuls p(t) und andererseits die generalisierte Verschiebung q(t). An der Namensgebung ist die Anlehnung an die Mechanik erkennbar. Weitere Details finden sich in der angegebenen Literatur [Bor00] [Rod13].

Compliance-Elemente

Das Compliance-Element stellt die erste hier vorgestellte Art der Energiespeicher dar. Es wird durch ein *C* visualisiert (Bild A4-1). Beispielhaft sind diesem Element die mechanische Feder, die Speicherkammer und die Kapazität zugeordnet. Die Namensgebung des C-Elements orientiert sich an der elektrischen Kapazität (Capacity) [Bor00] [Rod13].

Der Beziehung der Variablen bei einem linearen Zusammenhang zeigt sich in den folgenden Gleichungen:

$$q = C \cdot e \Leftrightarrow \frac{1}{C} \cdot q = e.$$

Durch die zeitliche Ableitung ergibt sich:

$$f = C \cdot \frac{de}{dt} \Leftrightarrow \frac{de}{dt} = \frac{1}{C} \cdot f.$$

Der Parameter C repräsentiert die hydraulische und die elektrische Kapazität, sowie in der Mechanik die Federnachgiebigkeit.

Die Gleichungen der lassen sich somit hieraus ableiten. Dabei bildet der Kehrwert der Federnachgiebigkeit die Federsteifigkeit c ab [Bor00] [Rod13]:

$$\frac{de}{dt} = \frac{1}{C} \cdot f \Rightarrow \dot{u} = \frac{1}{C} \cdot i \text{ (elektrisches Beispiel)},$$

$$\frac{de}{dt} = \frac{1}{C} \cdot f \Rightarrow \dot{F} = c \cdot v \text{ (mechanisches Beispiel)},$$

$$\frac{de}{dt} = \frac{1}{C} \cdot f \Rightarrow \dot{p} = \frac{1}{C_{hyd.}} \cdot q \text{ (hydraulisches Beispiel)}.$$

$$\frac{e}{f = \frac{dq}{dt}} C$$

Bild A4-1: Darstellungsweise eines C-Elements [Rod13]

$$\frac{e = \frac{dp}{dt}}{f}I$$

Bild A4-2: Darstellungsweise eines I-Elements [Rod13]

$$\frac{e}{f}$$
 R

Bild A4-3: Darstellungsweise eines R-Elements [Rod13]

Inductance-/Intertia-Elemente

Die zweite hier vorgestellte Art der Energiespeicher wird durch das *I-Elemente* in Bild A4-2 repräsentiert. Sie stehen stellvertretend für die Masse, die hydraulische Leitung und die Induktivität.

Bei den folgenden linearen Gleichungen zeigt sich, dass die Rollen der Effort- und Flowgröße gegenüber den C-Elementen getauscht sind [Bor00] [Rod13]:

$$p = I \cdot f$$
,

beziehungsweise:

$$e = I \cdot \frac{df}{dt}$$
.

Hierbei entspricht der Parameter je nach Domäne dem Massenträgheit, beziehungsweise dem Massenträgheitsmoment (bezüglich des Schwerpunktes) oder der Induktivität der Hydraulik und Elekrotechnik.

Energieverbraucher

Die Dissipation von Energie wird in Bondgraphen durch *R-Elemente* gekennzeichnet. Das R-Element (Bild A4-3), gibt Energie hauptsächlich in Form von Wärmeenergie ab. Es wird in den Domänen der Mechanik, Hydraulik und Elektrotechnik durch den Dämpfer, die Drossel und den elektrischen Widerstand beschrieben [Bor00] [Rod13].

Transformer

In Transformern wird die an einem Port hineinfließende Leistung umgewandelt und anschließend über den anderen Port abgegeben (Bild A4-4). Als Symbol dienen die Buchstaben TF. Es gilt folgende mathematische Beschreibung:

$$e_1(t) \cdot f_1(t) = e_2(t) \cdot f_2(t)$$
.

Der Transformerfaktor m, gibt das Verhältnis der Ein- und Ausgangsgrößen an. Dieser kann sowohl konstant, als auch eine Funktion sein. Das Übersetzungsverhältnis gilt dabei

$$\begin{array}{c|c}
 & e_1 & m \\
\hline
 & f_1 & TF \\
\hline
 & TF 1 & f_2
\end{array}$$

Bild A4-4: Bondgraph Symbol für den Transformer [Rod13].

$$\frac{e_1}{f_1}$$
 \mathbf{GY} $\frac{e_2}{f_2}$

Bild A4-5: Gyrator-Bauelement [Rod13]

für die Effort- und für die Flowgrößen. Damit ergeben sich folgende Zusammenhänge [Bor00] [Rod13]:

$$e_1 = m \cdot e_2$$
,

$$f_2 = m \cdot f_1$$
.

Beispiele für Transformer sind der elektrische Transformator und das mechanische Getriebe.

Gyratoren

Gyratoren (GY), sind vergleichbar zu Transformern, mit dem Unterschied, dass die Gyratoren eine Beziehung zwischen Eingangseffort/Ausgangseffort und dem Ausgangsflow/Eingangsflow herstellen. Unter Verwendung des Faktors r (Bild: A4-5) lassen sich die Zusammenhänge wie folgt ausdrücken [Bor00] [Rod13]:

$$e_1 = r \cdot f_2$$

$$e_2 = r \cdot f_1$$
.

Als Beispiel dient hier der Gleichstrommotor, der elektrische in mechanische Energie umwandelt.

0-Junction

Komplexe Systeme sind oft verbunden mit mehreren parallelen Leistungszweigen. Um eine Leistungsverteilung zu gewährleisten, ist es notwendig weitere Elemente einzuführen. Hierzu dienen sogenannte Junctions (Verknüpfungen).

Die 0-Junction (Bild A4-6) ist durch die folgenden Gleichungen charakterisiert. Hierbei sind f_1^{in} ... f_m^{in} eingehende und f_1^{out} ... f_n^{out} ausgehende Flows:

$$e_1 = e_2 = \dots = e_n$$

$$\sum_{i=1}^{m} f_i^{in} - \sum_{j=1}^{n} f_j^{out} = 0.$$

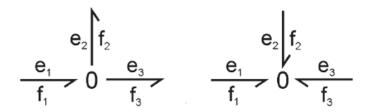


Bild A4-6: Flow-Junction mit unterschiedlicher Orientierung der Leistungsbonds, [Rod13, S. 52]

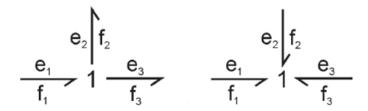


Bild A4-7: Effort-Junction mit unterschiedlicher Orientierung der Leistungsbonds [Rod13]

Efforts in einer 0-Junction sind somit identisch. Die Summe der zu- und abfließenden Flows ist null. Hier lässt sich u.a. das Kirchhoff´sche Knotengesetz wiedererkennen [Bor00] [GB07] [Rod13].

Bei den aufgeführten Varianten in Bild A4-6 ergeben sich damit die nachfolgenden Flowbeziehungen, die Efforts sind gleich.

• Linke Variante aus Bild A4-6

$$f_1(t) - f_2(t) - f_3(t) = 0$$

• Rechte Variante aus Bild A4-6

$$f_1(t) + f_2(t) + f_3(t) = 0$$

1-Junction

Analog zur vorherigen Junction, sind hier nicht die Effort-, sondern die Flowgrößen gleich sind und die Summe der Effortgrößen null ist (Bild A4-7) [Bor00] [GB07] [Rod13]:

$$f_1 = f_2 = \dots = f_n$$

$$\sum_{i=1}^{m} e_i^{in} - \sum_{j=1}^{n} e_j^{out} = 0.$$

A5 Ergänzende Bilder der Anwendung

A5.1 Die wesentlichen Bausteine des Entfaltens

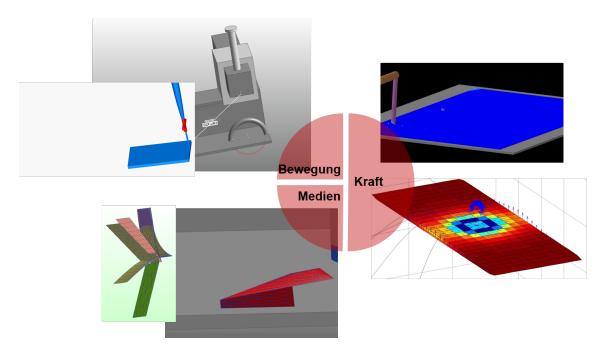


Bild A5-1: Die drei wesentlichen Bausteine für die Entfaltung einer Banknote. Einerseits wird die Banknote (Medien) benötigt. Auf diese muss eine Kraft wirken und diese entsprechend einer Trajektorie beibehalten werden (Bewegung). Diese Aspekte können systematisch untersucht und zu einer Gesamtlösung zusammengefügt werden.

A5.2 Kategorisierung von Noteneigenschaften

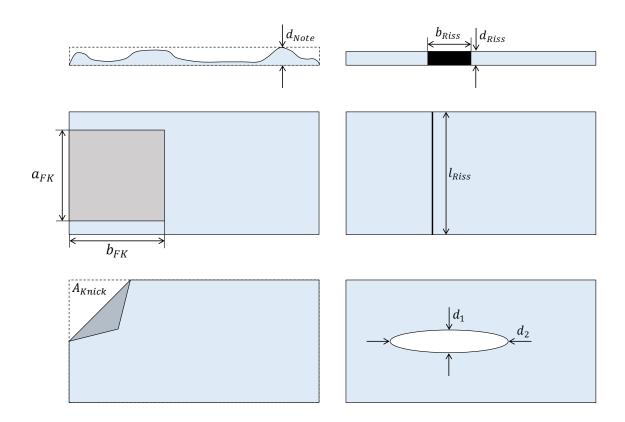


Bild A5-2: Sechs Kategorien zur systematischen Kategorisierung von Banknoten: Die Systmatisierung kann anhand von konkreten Werten erfolgen. Die Notendicke wird im Verlauf über den Parameter d_{Note} bestimmt, ein Riss anhand der Parameter b_{Riss} , d_{Riss} und l_{Riss} . Ein Knick kann eine Fläche $A_{Knick_{max}} = \frac{1}{2}A_{Note}$ haben. Nimmt der Knick zu, wird entsprechend der andere Notenabschnitt als Knick angenommen. Ein Loch kann die Dimensionen d_1 und d_2 aufweisen. Ein Fremdkörper wird über die maximalen Ausprägungen a_{FK} und b_{FK} definiert.