

## **Masterarbeit**

Maschinenbau

# **Identifikation und Bewertung von Auswirkungsmustern zwischen Testergebnissen und Anforderungen im Model-Based Systems Engineering**

vorgelegt von  
B.Sc. Thomas Hesse  
Matr.-Nr. 7171479

Betreuende:

Prof. Dr.-Ing. Iris Gräßler  
Dr.-Ing. Jens Pottebaum  
M.Sc. Marcel Ebel  
M.Sc. Dominik Wiechel

Paderborn, 29.08.2024

## Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abkürzungsverzeichnis .....	IV
1    Einführung .....	1
1.1    Motivation .....	1
1.2    Zielsetzung .....	2
1.3    Wissenschaftliches Vorgehen .....	2
2    Theoretische Grundlagen .....	7
2.1    Systems Engineering.....	7
2.2    V-Modell .....	7
2.3    Model-Based Systems Engineering.....	9
2.4    RFLPV <sup>2</sup> .....	10
2.5    Entwicklungsartefakte.....	12
2.5.1    Anwendungsfälle.....	12
2.5.2    Anforderungen .....	13
2.5.3    Testfälle .....	15
2.5.4    Testszenarien .....	16
2.6    Modellierungssprache SysML .....	18
2.6.1    Diagrammtypen.....	18
2.6.2    Modellierungselemente .....	19
2.6.3    Relationen.....	20
2.7    Engineering Change-Management.....	20
2.7.1    Strategien für Umgang mit technischen Änderungen.....	22
2.7.2    Auswirkungsmuster und Propagationseffekte .....	23
3    Stand der Technik.....	25
3.1    Auswirkungsanalyse mit Hilfe der Designstruktur Matrix .....	25
3.1.1    Auswirkungsanalyse mit SysML für die Identifizierung des Aufwands .....	25
3.1.2    Auswirkungsanalyse anhand von Mustern.....	26
3.2    Management von Propagationseffekten für technische Änderungen .....	27
3.3    Auswirkungsanalyse auf Basis der Systemartefakte und Relationen .....	28
3.3.1    Auswirkungsanalyse für Testfälle.....	28

3.3.2	Auswirkungsanalyse anhand der Entwicklungsartefakte und Relationen.....	29
3.3.3	Methode für die Identifizierung von Propagationspfaden .....	30
3.4	Auswirkungsanalyse mit dem CIRA-Verfahren.....	32
3.5	Abgrenzung der Ansätze .....	33
4	Analyse verwandter Ansätze .....	37
4.1	Literaturrecherche .....	37
4.2	Modellierungsmethoden für cyberphysische Systeme.....	38
4.2.1	OOSEM.....	39
4.2.2	SysMOD.....	40
4.2.3	Harmony SE.....	41
4.2.4	ISO 15288.....	42
4.2.5	Consens.....	44
4.2.6	Arcadia.....	45
4.2.7	JPL State Analysis .....	46
4.2.8	Ruph SE.....	47
4.2.9	Pattern Based Systems Engineering .....	48
4.2.10	Munich agile MBSE Concept (MAGIC) .....	49
4.3	Fallstudien für die Modellierung cyberphysischer Systeme .....	51
4.3.1	Luftwindenergieanlagen .....	51
4.3.2	Flugzeugfahrwerk .....	52
4.3.3	Demonstrator für die Luftfahrt .....	53
4.3.4	Turbine einer Windkraftanlage .....	55
4.3.5	Bahnstreckensignal.....	56
4.3.6	Weltraumrobotersystem .....	57
4.3.7	Fahrerassistenzsystem .....	58
4.3.8	Produktionsmaschinen.....	59
5	Analyse testinduzierter Änderungen .....	61
5.1	Identifikation von Auswirkungsmustern .....	61
5.2	Darstellung der Auswirkungsmuster .....	62
5.3	Bewertung der Kritikalität der Auswirkungen .....	66
5.4	Vorgehen für die Identifikation betroffener Systemartefakte.....	68
5.5	Prototypische Umsetzung anhand von Testfällen .....	71
6	Validierung .....	73
6.1	Planung der Validierungsgespräche.....	73

6.2	Durchführung der Validierungsgespräche .....	75
6.3	Ergebnisse der Validierung .....	76
6.3.1	Auswertung des Usability-Fragebogens.....	76
6.3.2	Auswertung der Leitfragen .....	77
6.3.3	Umsetzung von Verbesserungsvorschlägen.....	81
6.3.4	Beantwortung der Forschungsfragen .....	82
7	Zusammenfassung und Ausblick .....	85
8	Literaturverzeichnis.....	89
Anhang	.....	1

## **Anhang**

A-1	Ebenen-spezifische Auswirkungsmuster .....	A-3
A-2	Ebenen-übergreifende Auswirkungsmuster .....	A-12
A-3	Prüfalgorithmen für die Identifikation betroffener Systemartefakte.....	A-25
A-4	Unterlagen für die Validierung .....	A-31
A-5	Ergebnisse der Probanden .....	A-37

## Abkürzungsverzeichnis

### Spezifische Abkürzungen

ARCADIA	Architecture Analysis & Design Integrated Approach
PE:	Produktentstehung
CAx	Computer Aided Technologies
CAD	Computer Aided Design
PBSE	Pattern Based Systems Engineering
DDM	Design Dependency Matrix
DSM	Designstruktur Matrix
CONSENS	CONceptual design Specification technique for the ENgineering of complex Systems
CPM	characteristics and properties modelling
CRM	Change Request Management
FMEA	Fehlermöglichkeits- und Einfluss-Analyse
JPL	Jet Propulsion Laboratory
IPO	Input, Process, Output
MAGIC	Munich agile MBSE Concept
MoE	Measurement of Effectiveness
MDE	model driven Engineering
Ruph SE	Rational Unified Process
RFLPV <sup>2</sup>	Requirement, Functions, logical Elements, physical Elements, Verification, Validation
RFLP	Requirement, Functions, logical Elements, physical Elements
ISO	International Organization for Standardization

ID	Identifikationsnummer
IEEE	Institute of Electrical and Electronics Engineering
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
OOSEM	Object Orientated System Engineering Method
SE	Systems Engineering
SysMOD	System Modification
SSCI	Systems and Software Consortium
SUS	System Usability Score
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
PDD	property driven development
PERT	program evaluation and review technique
SysML	System Modelling Language
UML	Unified Modelling Language
VDI	Verein deutscher Ingenieure
Req	Requirement Diagram
Bdd	Block definition diagram
lbd	internal block diagram

**Lateinische Abkürzungen**

etc.:	et cetera
et al.	et alii/ et aliae/ et alia
f.:	folio
ff.:	folio folio

## Abbildungsverzeichnis

Abbildung 1.1: Wissenschaftliches Vorgehen für die Forschungsfrage in Anlehnung an das Vorgehen nach Ulrich [Gei81] .....	4
Abbildung 1.2: Erkenntnisgewinnung für die Identifizierung von Auswirkungsmustern basierend auf den Entwicklungsartefakten und deren Relationen.....	5
Abbildung 2.1: V-Modell nach der VDI-Richtlinie 2206:2021 [VDI21] .....	9
Abbildung 2.2: In Anlehnung an die Integration des RFLPV <sup>2</sup> -Ansatzes in das V- Modell nach Gräßler et al. [GWO22] .....	11
Abbildung 2.3: Beispiel für einen erstellten Anwendungsfall mit Hilfe von SysML .....	13
Abbildung 2.4: Anforderungen dargestellt in SysML.....	14
Abbildung 2.5: Exemplarische Darstellung eines Testfalls in SysML .....	15
Abbildung 2.6: Grafische Darstellung für die Beschreibung eines Testfalls in Anlehnung an Conrad [Con04].....	16
Abbildung 2.7: Darstellung der Beschreibung eines Testszenarios.....	17
Abbildung 2.8: Übersicht von Diagrammtypen in SysML.....	18
Abbildung 2.9: Beispiele für Relationen in SysML Diagrammen.....	20
Abbildung 2.10: In Anlehnung an den Generischen Produktentwicklungsprozess nach Ulrich und Eppinger [UE11] .....	21
Abbildung 2.11: In Anlehnung an den generischen Engineering Change Prozess nach Jarrett [Jar05] .....	22
Abbildung 3.1: IPO-Diagramm für das Vorgehen in Anlehnung an Ellsel et al. [ESS22].....	25
Abbildung 3.2: Änderungsarten gemäß des Ansatzes nach Ellsel et al. [ESS22] .....	26
Abbildung 3.3: IPO-Diagramm für das Vorgehen in Anlehnung an Li und Chen [LC10] .....	26
Abbildung 3.4: IPO-Diagramm für das Vorgehen in Anlehnung an Reddi und Moon [RM09] .....	27
Abbildung 3.5: IPO-Diagramm für das Vorgehen in Anlehnung an Intana et al. [ILS23; IS19] .....	28
Abbildung 3.6: IPO-Diagramm für das Vorgehen in Anlehnung an Lin et al. [LPS+15] .....	29
Abbildung 3.7: Vorgehen für die Identifikation von Propagationspfaden als Aktivitätsdiagramm in Anlehnung an Yang und Duan [YD12] .	31
Abbildung 3.8: IPO-Diagramm für das Vorgehen in Anlehnung an Deubel et al. [DCK+07] .....	33
Abbildung 3.9: Abgrenzungsmatrix der Handlungsmaßnahme .....	34
Abbildung 4.1: Ergebnisse der Literaturrecherche gemäß dem gewählten Vorgehen .....	38
Abbildung 4.2: Darstellung der elementaren Entwicklungsartefakte [Fri14] der OOSEM-Methode auf dem V-Modell verortet .....	39
Abbildung 4.3: Darstellung der elementaren Entwicklungsartefakte der SysMOD-Methode auf dem V-Modell verortet .....	40

Abbildung 4.4: Darstellung der elementaren Entwicklungsartefakte der Harmony SE-Methode auf dem V-Modell verortet .....	42
Abbildung 4.5: Darstellung der elementaren Entwicklungsartefakte gemäß der ISO 15288 auf dem V-Modell verortet.....	43
Abbildung 4.6: Darstellung der elementaren Entwicklungsartefakte der Consens-Methode auf dem V-Modell verortet.....	44
Abbildung 4.7: Darstellung der elementaren Entwicklungsartefakte der Arcadia-Methode auf dem V-Modell verortet.....	45
Abbildung 4.8: Darstellung der elementaren Entwicklungsartefakte der JPL State Analysis-Methode auf dem V-Modell verortet .....	47
Abbildung 4.9: Darstellung der elementaren Entwicklungsartefakte der Ruph SE-Methode auf dem V-Modell verortet .....	48
Abbildung 4.10: Darstellung der elementaren Entwicklungsartefakte des PBSE-Metamodells auf dem V-Modell verortet.....	49
Abbildung 4.11: Darstellung der elementaren Entwicklungsartefakte der MUNICH-Methode auf dem V-Modell verortet .....	50
Abbildung 4.12: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für die Luftwindenergieanlage auf dem V-Modell verortet.....	51
Abbildung 4.13: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für ein Flugzeugfahrwerk auf dem V-Modell verortet .....	53
Abbildung 4.14: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für den Demonstrator eines Flugzeugs auf dem V-Modell verortet .....	54
Abbildung 4.15: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für die Turbine einer Windkraftanlage auf dem V-Modell verortet .....	55
Abbildung 4.16: Darstellung der elementaren Entwicklungsartefakte der Fallstudie eines Bahnstreckensignals auf dem V-Modell verortet .....	56
Abbildung 4.17: Darstellung der elementaren Entwicklungsartefakte der Fallstudie Weltraumroboters auf dem V-Modell gemäß der Systemebenen verortet.....	58
Abbildung 4.18: Darstellung der elementaren Entwicklungsartefakte der Fallstudie eines Fahrerassistenzsystems auf dem V-Modell gemäß der Systemebenen verortet.....	59
Abbildung 4.19: Darstellung der elementaren Entwicklungsartefakte der Fallstudie von Produktionsmaschinen auf dem V-Modell gemäß der Systemebenen verortet.....	60
Abbildung 5.1: Klassifizierungssystem für die Bezeichnung der Auswirkungsmuster.....	62
Abbildung 5.2: Auswirkungsmuster für das Erstellen einer neuen Stakeholderanforderung.....	64
Abbildung 5.3: Ebenen-übergreifendes Auswirkungsmuster für die Erstellung einer neuen Anforderung .....	65



Abbildung 5.4: Template für die Berechnung der Änderungsklassifikationsnummer .....	67
Abbildung 5.5: Bewertungsschema für die Einordnung der Art der Änderung..	68
Abbildung 5.6: Algorithmus A4 dargestellt als SysML Aktivitätsdiagramm .....	70
Abbildung 6.1: Auflistung der teilnehmenden Probanden.....	73
Abbildung 6.2: Ablauf der Experteninterviews .....	74
Abbildung 6.3: Ergebnisse der Befragung der Probanden mit dem SUS- Fragebogen.....	77

# 1 Einführung

Durch Änderungen in Eigenschaftsabsicherung kann der Projekterfolg bei einer falschen Verfahrensweise gefährdet werden. Für die Beschreibung möglicher Änderungsauswirkungen werden aus diesem Grund Ansätze benötigt. Diese müssen den Ingenieur befähigen Änderung hinsichtlich der Kritikalität zu beschreiben und die Identifikation betroffener Entwicklungsartefakte ermöglichen. In diesem Fall wird der Validierungs- & Verifikations-Ingenieur (V&V-Ingenieur) die Maßnahmen anwenden, da dieser für die Planung und Durchführung der Eigenschaftsabsicherung im Unternehmen zuständig ist. Das Einführungskapitel erläutert die Motivation für das Forschungsvorhaben detailliert dargestellt. Aufbauend darauf, werden die Zielsetzung des Forschungsvorhabens und die zugrunde liegenden Forschungsfragen detailliert dargelegt. Außerdem wird das gewählte wissenschaftliche Vorgehen für die Beantwortung der Forschungsfragen beschrieben.

## 1.1 Motivation

In der Entwicklung innovativer Produkte werden die Ansprüche an die Qualität stetig höher. Zusätzlich entstehen durch den Konkurrenzdruck und die steigende Komplexität der Produkte höhere Ansprüche an die Prozesse der Produktentwicklung [GO22]. Für die Überprüfung der Produktqualität und der daraus resultierenden Erfüllung der Anforderungen sind die Eigenschaften der Produkte frühzeitig abzusichern. Die Eigenschaftsabsicherung erfolgt im Produktentwicklungsprozess durch die Verifikation und Validierung. [Wal23] Die in diesem Abschnitt auftretenden Entwicklungsartefakte, die Testfälle und Testszenarien weisen Abhängigkeiten zu den Entwicklungsartefakten der Anforderungserhebung auf. [GWO22] Diese umfassen dabei die Anforderungen und die für das Produkt vorgesehenen Anwendungsfälle. Aufgrund der direkten Verbindung der Eigenschaftsabsicherung und der Anforderungserhebung können durch Änderungen im Verifikations- und Validierungsprozess Auswirkungen für die Anforderungen entstehen. Diese Änderungen wirken sich auf den weiteren Entwicklungsprozess aus. Die geänderten Anforderungen können die im Entwicklungsprozess folgenden Funktionen, logischen und physischen Elemente verändern. Diese Änderungen können im späten Verlauf der Entwicklung zu hohen Kosten führen und den Projekterfolg verringern. Für die Ermittlung und Darstellung der Auswirkung aus Änderungen der Anforderungen bestehen bereits Vorgehen für die Auswirkungsanalyse. Für die Analyse der Auswirkung werden das System modelliert und die Abhängigkeiten der Entwicklungsartefakte untersucht. Für diese Untersuchung existieren verschiedene Ansätze wie beispielsweise die Analyse von Designstruktur Matrizen oder die Graph-Analyse. Für die Ermittlung der Auswirkung von Änderungen ausgehend aus der Eigenschaftsabsicherung sind bisher keine Ansätze für die Identifikation und Bewertung vorhanden. Die Änderungen treten dabei in der Planung sowie in der Durchführung der Verifikation und Validierung auf und erzeugen hierdurch Veränderungen. Diese Veränderungen weiten sich auch auf die Anforderungen der Anwendungsfälle aus. Für die frühzeitige Identifizierung der möglichen Auswirkungen sind weiterführende Maßnahmen wie beispielsweise die Bewertung der Auswirkungen und

Handlungsmaßnahmen für die Änderungsdurchführung notwendig. Dies erleichtert die Durchführung der Änderung und fördert die Kommunikation innerhalb des Entwicklungsprojektes. Zusätzlich führt die Bewertung der Kritikalität der möglichen Auswirkungen auch zu einer Planungssicherheit. Dies ermöglicht, frühzeitig kritische Testszenarien genauer zu planen und durchzuführen. Die für die Bewertung nötigen Datengrundlagen sind durch die gängigen Vorgehensweisen des Model-Based Systems Engineering einzubeziehen.

## 1.2 Zielsetzung

Aus der identifizierten Forschungslücke ergibt sich das Ziel, eine Handlungsempfehlung für die Modellierung und Bewertung von Auswirkungsmustern basierend auf Änderungen in der Eigenschaftsabsicherung zu erarbeiten. Für die Analyse der Ausgangslage sind die Entwicklungsartefakte der Eigenschaftsabsicherung und die der Anforderungserhebung zu definieren. Dies umfasst dabei die formale Definition der einzelnen Artefakte als auch die Analyse der Abhängigkeiten in Form von Relationen zwischen den Artefakten. Durch die Analyse der Abhängigkeiten sind mögliche Auswirkungsmuster zu identifizieren und zu beschreiben. Zusätzlich ist eine Bewertung dieser Auswirkungsmuster durchzuführen, wobei die Bewertung sich auf die Kritikalität der Auswirkung bezieht. Für die Bewertung werden dabei direkte als auch indirekte Auswirkungsmuster betrachtet. Für die Bewertung der Kritikalität der Auswirkung sind die Auswirkungsmuster zu analysieren und weitere Bewertungsmerkmale mit einzubinden. Die Bewertungsmerkmale umfassen beispielsweise Randbedingungen, welche durch die einzelnen Artefakte beeinflusst werden können. Auf Basis der Analyseergebnisse der Auswirkungsmuster ist die Handlungsempfehlung für die jeweiligen Muster abzuleiten. Das erarbeitete Vorgehen ist dabei auf exemplarische Testfälle und Testszenarien anzuwenden. Dieses wird im Rahmen von Experteninterviews auf Basis der Testfälle und Testszenarien validiert. Somit ergeben sich für den Forschungsansatz folgende Forschungsfragen:

**Forschungsfrage 1:** Sind Auswirkungsmuster basierend auf Änderungen in der Eigenschaftsabsicherung zu identifizieren?

**Forschungsfrage 2:** Wie wirken sich die Abhängigkeiten der Entwicklungsartefakte auf die Auswirkungsmuster aus?

**Forschungsfrage 3:** Wie lassen sich die Auswirkungsmuster bewerten?

**Forschungsfrage 4:** Welche Handlungsmaßnahmen ergeben sich aus diesen Auswirkungsmustern?

## 1.3 Wissenschaftliches Vorgehen

Für die Beantwortung der Forschungsfragen ist das Vorgehen an das Vorgehen nach ULRICH [Gei81] orientiert. In dem ersten Schritt, wie in Abbildung 1.1 zu sehen, werden die

praxisrelevanten Probleme erfasst und typisiert. In der betrachteten Forschungslücke bedeutet dies, dass aus Planung und Durchführung der Eigenschaftsabsicherung Auswirkungen auf das Gesamtsystem entstehen können. Diese Auswirkungen basieren auf dem Einfluss der Anforderungen an das System. Für Lösung der Problematik ist eine Handlungsempfehlung zur Modellierung und Analyse der Zusammenhänge zu erarbeiten. Dies umfasst die Analyse der Abhängigkeiten der Entwicklungsartefakten in etablierten Vorgehensweisen.

In dem zweiten Schritt stehen die Erfassung und Interpretation problemrelevanter Theorien an. Hierzu wird eine systematische Literaturrecherche durchgeführt. Das Vorgehen wird in Anlehnung an das PRIMSA-Verfahren [PMB+21] durchgeführt. Die Suche ist so aufgebaut, dass auf Basis eines erarbeiteten Suchstrings in den Datenbanken Web of Science, Scopus und IEEE Xplore die verwandte Literatur ermittelt wird. Für die Vervollständigung wird mit demselben Suchstring in der Suchmaschine Google Scholar nach noch nicht erfasster Literatur gesucht. Die aus diesem Verfahren ermittelte Literatur ist in einem mehrstufigen Verfahren zu selektieren, um die relevante Literatur herauszufiltern. Das Sortieren der Literatur ist an dem Vorgehen nach KORNMEIER [Kor18] orientiert. Ausgehend von der ermittelten Literatur ist mit Hilfe gegebener Primärquellen die Analyse der grundlegenden Theorien zu vervollständigen.

Im dritten Schritt wird der Stand der Technik ermittelt, wobei die in der systematischen Literaturrecherche ermittelten Literatur verwendet wird. Im Fokus stehen hier artverwandte Vorgehen aus dem Bereich der Einflussanalyse. Die ermittelten Verfahren werden anhand der Charakteristika voneinander unterschieden und auf die Projizierbarkeit hinsichtlich des Forschungsvorhabens geprüft. Darüber hinaus werden Aspekte der Ausbreitung von Änderungen, sogenannte Propagationseffekte, innerhalb technischer Systeme genauer untersucht. Im Fokus steht dabei die Bestimmung der Art des Verhaltens, welches die Entwicklungsartefakte hinsichtlich der Propagation aufweisen. Zusätzlich sind Ansätze für die Bewertung der Kritikalität von Auswirkungen zu betrachten. Die daraus gewonnenen Erkenntnisse werden für eine Abgrenzung der Theorien gegenüber des Forschungsansatzes genutzt.

Im vierten Schritt wird der praxisrelevante Anwendungszusammenhang ermittelt. Für die Ermittlung werden exemplarische Testfälle und Testszenarien verwendet wie in Abbildung 1.1 dargestellt. Diese Testfälle basieren auf Ergebnissen von praxisrelevanten Projekten. Zum einen werden Testfälle herangezogen, die im Zusammenhang mit dem Forschungsprojekt CrexData durchgeführt werden. Auf der anderen Seite stehen Testfälle, die im Rahmen der Entwicklung des Rennautos des UPBracing-Teams verwendet werden. Die exemplarischen Testfälle werden analysiert. Als Referenz stehen die aus der

Theorie und den Fallstudien ermittelten Erkenntnisse. Aus den resultierenden Ergebnissen werden Randbedingungen, die für die Umsetzung und die Modellierung gelten, herausgestellt.

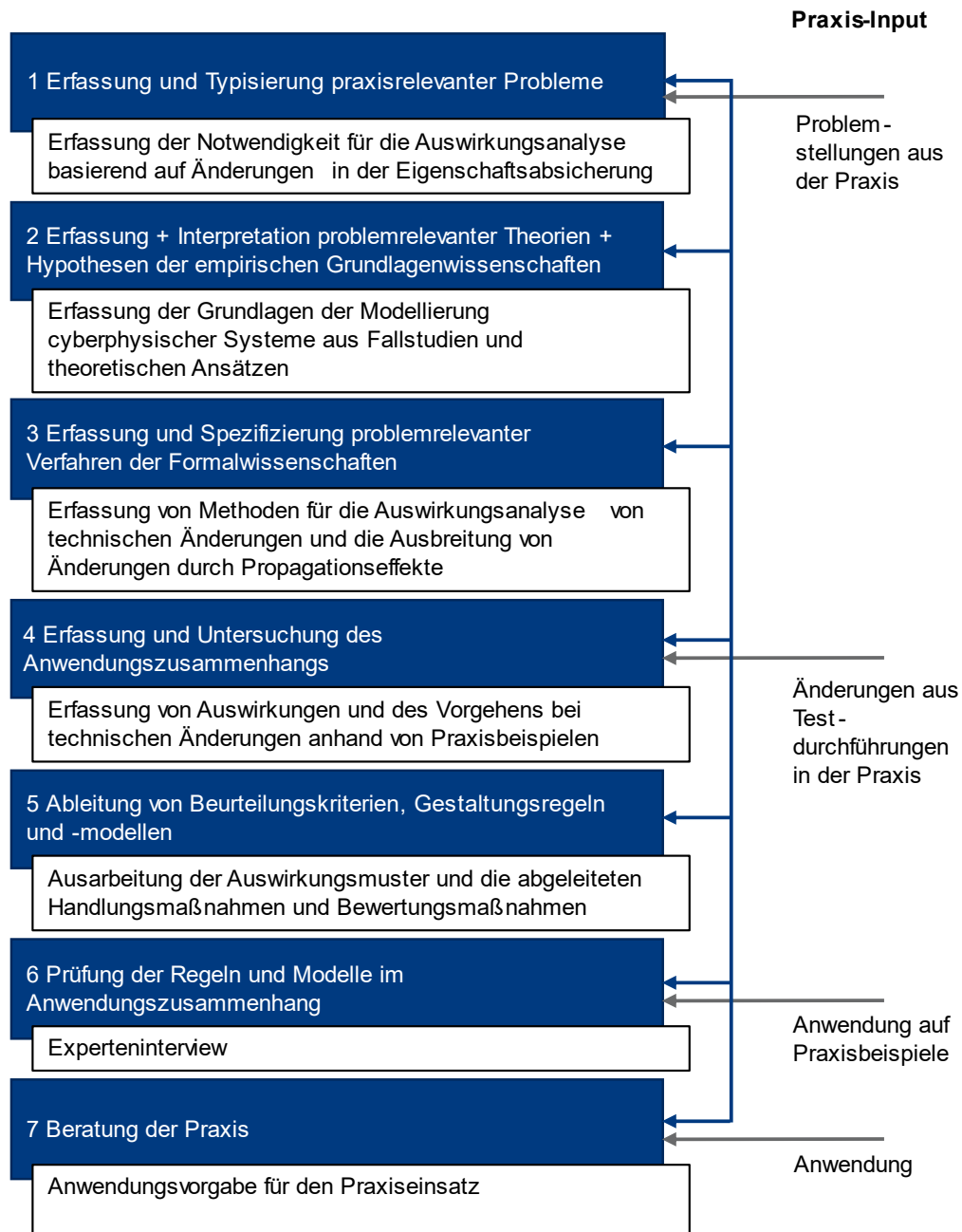
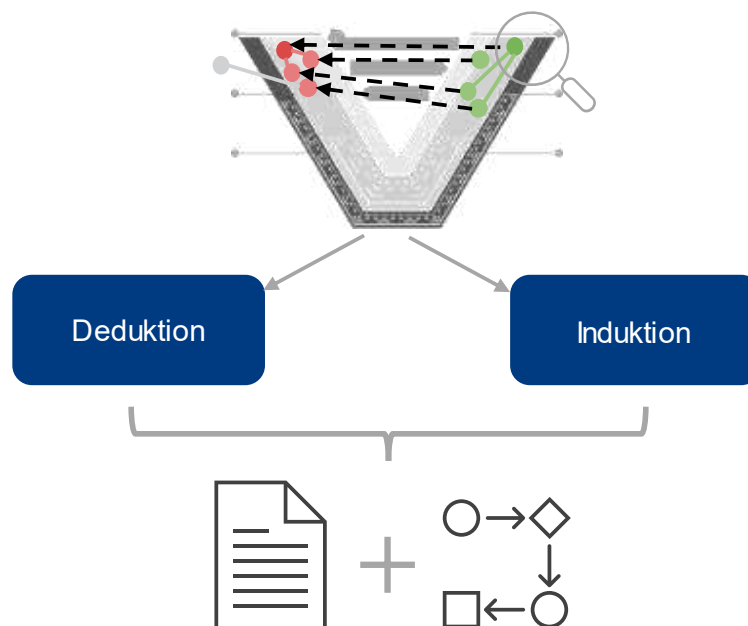


Abbildung 1.1: Wissenschaftliches Vorgehen für die Forschungsfrage in Anlehnung an das Vorgehen nach Ulrich [Gei81]

Im fünften Schritt wird auf Basis der Ergebnisse der vorangegangenen Schritte die Handlungsmaßnahme abgeleitet. Das Vorgehen für die Wissenserkenntnis erfolgt durch die Gegenüberstellung der deduktiven und der induktiven Betrachtung der Modellierung von

cyberphysischen Systemen. In Abbildung 1.2 ist dargestellt, wie die Erkenntnisgewinnung auf Basis der Modellierung der Entwicklungsartefakte durchgeführt wird. Hierzu wird die Modellierung der Entwicklungsartefakte und die Relationen zwischen den Artefakten aus theoretischer Sicht genauer untersucht. Die Untersuchung umfasst dabei das Verhalten der unterschiedlichen modellierten Systeme auf bestimmte Arten von Änderungen. Diese werden hinsichtlich verschiedener Perspektiven genauer untersucht. Zum einem wird das Verhalten bei geänderten Anforderungen und zum anderen die Auswirkung bei neu erstellten Anforderungen untersucht. Darüber hinaus wird analysiert, welche Auswirkungen auftreten, wenn die Anwendungsfälle sich ändern oder gemäß der Hauptmerkmalgruppen von Anforderungen Änderungen in einer bestimmten Eigenschaftsgruppe auftreten. Dieselben Betrachtungen werden für die Modelle aus den Fallstudien durchgeführt. Die Erzeugung der Muster erfolgt durch das Clustering der Artefakte hinsichtlich ihres Verhaltes in dem betrachteten Fall. Die daraus entstehenden Auswirkungsmuster werden hinsichtlich der Kritikalität für das Projekt bewertet. Mit Hilfe der Bewertung und den Mustern können die Handlungsmaßnahmen für den Anwender abgeleitet werden. Zusätzlich werden bestimmte Gestaltungsrichtlinien und nötige Erweiterungen hinsichtlich der Modellierung definiert. Die Handlungsmaßnahmen umfassen präventive als auch reaktive Ansätze.



*Abbildung 1.2: Erkenntnisgewinnung für die Identifizierung von Auswirkungsmustern basierend auf den Entwicklungsartefakten und deren Relationen*

Im vorletzten Schritt wird die Tauglichkeit der Handlungsempfehlungen und der identifizierten Auswirkungsmuster geprüft. Die Überprüfung erfolgt in Experteninterviews mit Probanden aus den unterschiedlichen Projekten. In den Experteninterviews werden mit Hilfe von Praxisbeispielen auftretende Änderungen aus Testfällen besprochen. Auf Basis

dieser Erkenntnisse wird die Qualität der Muster und der daraus abzuleitenden Maßnahmen mit den Maßnahmen aus der Praxis verglichen. Aus diesem Vorgehen lassen sich Aussagen über die Übertragbarkeit der Muster und nötige Randbedingungen treffen.

Im letzten Schritt der Beratung der Praxis sind mit Hilfe der Ergebnisse aus den Experteninterviews bestehende Restriktionen und Entwicklungspotentiale herauszustellen. Die Restriktionen umfassen Sowohl Handlungseinschränkungen als auch bestehende Randbedingungen und empfohlene Modellierungsregeln. Die Entwicklungspotentiale erstrecken sich dabei auf mögliche Einbindungen in Tools oder bestehenden Methoden.

## 2 Theoretische Grundlagen

In diesem Kapitel werden die zugrunde liegenden theoretischen Grundlagen zur Ausarbeitung der Handlungsmaßnahmen für die Identifikation und Bewertung der Auswirkungsmuster dargelegt. Die Grundlagen umfassen die Einordnung in den Produktentstehungsprozess und damit einhergehende Modelle wie beispielsweise die VDI/VDE 2206:2021. Aufbauend darauf steht die Erläuterung der Eigenschaften und Abhängigkeiten der Entwicklungsartefakte wie beispielsweise Anforderungen, Anwendungsfälle und Testfälle. Darüber hinaus wird eine Einordnung in das Änderungsmanagement durchgeführt.

### 2.1 Systems Engineering

Die Entwicklung der Handlungsmaßnahme für die Identifikation von Auswirkungsmustern bedingt durch testinduzierte Änderungen lässt sich dem Bereich des Systems Engineering zuordnen. Das Systems Engineering beschreibt einen interdisziplinären Ansatz, der mit Hilfe bestimmter Methoden und Grundsätze für die Entwicklung cyberphysischer Systeme genutzt wird. Im Zusammenwirken des Systems Engineering Ansatz mit systematischem Denken soll ein ganzheitliches Optimum geschaffen werden. Zusätzlich soll durch die Ansätze des Systems Engineering die Problematik der steigenden Komplexität der Systeme gelöst werden. Die steigende Komplexität ist durch die Anforderungen des Marktes in Form von Wiederverwertbarkeit und oder einer neutralen Klimabilanz zu erklären. Zusätzlich wird diese Komplexität durch die steigende Vernetzung der Systeme, also durch den Übergang zu cyber-physischen Systemen, erhöht. Für die Entwicklung cyberphysischer Systeme ist die Richtlinie VDI/VDE 2206:2021 entwickelt worden. Diese beschreibt die Entwicklung von Systemen mit Hilfe des V-Modells. Dieses Modell wird im Folgenden genauer erläutert. [GO22; Wal23]

### 2.2 V-Modell

Das V-Modell wird für die Entwicklung cyber-physischer Systeme verwendet und ist durch die Richtlinie VDI/VDE 2206:2021 [VDI21] definiert. In dem Modell werden die Kernaufgaben von der Einordnung in das zu verfolgende Geschäftsmodell bis zu der Übergabe des Produktes an den Kunden betrachtet. Die einzelnen Kernaufgaben in dem V-Modell werden mit Hilfe von Kontrollpunkten überprüft. An den Kontrollpunkten kann die Durchführung der vorausgelaufenen Kernaufgaben anhand von Fragen überprüft werden. Generell handelt es sich bei der Anwendung des V-Modells nicht um einen linearen, sondern einem iterativen Prozess. Der Aufbau des V-Modells ist in der Abbildung 2.1 dargestellt. [GO22; VDI21]

In dem ersten Kontrollpunkt ist zu überprüfen, welche Art von Geschäftsmodell zu verfolgen ist. Für die Erfassung wird die Sicht des Unternehmens eingenommen, in dem das Entwicklungsvorhaben durchgeführt wird. Hierzu müssen die Stakeholder identifiziert



und bewertet werden. Zusätzlich ist die Geschäftsidee und die Stellung des Unternehmens auf dem Markt zu bewerten. Diese Bewertung wird mit Hilfe der Geschäfts- und Vorhabens-Analyse durchgeführt. [GO22; VDI21]

In dem zweiten Kontrollpunkt der Spezifikation werden die Bedürfnisse der Stakeholder identifiziert. Im Rahmen der Identifikation werden explizit erwähnte aber auch implizit verlangte Stakeholderanforderungen für das System bestimmt. Für die Identifikation von impliziten Anforderungen können beispielsweise Anwendungsfälle mit den Stakeholdern ausgearbeitet werden. Diese Anwendungsfälle beschreiben dabei den von den Stakeholdern erwarteten Umgang mit dem System. Hieraus ergeben sich Leistungen, die durch das System zu erbringen sind, ohne vorab bei der Erhebung der Stakeholderanforderungen genannt werden. Aus den Stakeholderanforderungen sind in dem nächsten Schritt Systemanforderungen abzuleiten. Die Systemanforderungen stellen Anforderungen dar, welche beispielsweise durch Parameter beschreibbar sind. Wenn der Schritt der Erzeugung von Systemanforderungen abgeschlossen ist oder vorerst nicht weitergeführt werden kann, findet der nächste Schritt in der Entwicklung statt. [GO22; VDI21]

In dem dritten Kontrollpunkt des in Abbildung 2.1 dargestellten Modells wird die Systemarchitektur festgelegt: die Art wie die einzelnen Elemente des Systems wirken und wie die Struktur des Systems aufgebaut werden soll. Hierzu sind die Schnittstellen in der Systemarchitektur zwischen den einzelnen Teilsystemen auszuarbeiten. Ebenfalls sind gemäß, dem in Kapitel 2.4 erläuterten, RFLPV<sup>2</sup>-Ansatz die Elemente der Funktionen und logischen Elemente zu definieren. Die in diesem Schritt festgelegte Architektur bestimmt maßgeblich die Eigenschaften wie die Testbarkeit oder Einfachheit des Systems. Nach Abschluss der Definition der Systemarchitektur kann im Folgenden mit der Implementierung fortgefahren werden. [GO22; VDI21]

Im Kontrollpunkt der Implementierung werden gemäß der zuvor festgelegten Systemarchitektur die physischen Elemente umgesetzt. Dies erfolgt unter den Vorgaben der vorausgegangenen Prozesse der Systemarchitekturdefinition und der Systemanforderungsdefinition. Die Umsetzung erfolgt zunächst mit Hilfsmitteln wie CAx oder ähnlichen Werkzeugen. Es werden dabei beispielsweise für mechanische Teile die Konstruktion durchgeführt oder diese Teile gefertigt. Im Fall der Elektronik sind Leiterplatten zu entwerfen oder der Aufbau der elektronischen Systeme zu planen und umzusetzen. Darüber hinaus werden ebenfalls die Softwarelösungen implementiert. [GO22; VDI21]

Im Kontrollpunkt der Integration werden die einzeln entwickelten Systemelemente auf die nächsthöhere Systemebene zusammengeführt. Während des Prozesses müssen die einzelnen Systemelemente kontinuierlich verifiziert und validiert werden, sofern dies möglich ist. Es ist zu beachten, dass während der Eigenschaftsabsicherung die Systemebene beibehalten werden muss. Die Verifikation und Validierung kann ebenfalls durch virtuelle Werkzeuge unterstützt werden und erweitert somit den Lösungsraum um die virtuelle Umgebung. [GO22; VDI21]

Im dem letzten Kontrollpunkt der Übergabe wird das System an den Stakeholder übergeben. Im Vorfeld ist das System durch die Eigenschaftsabsicherung mit den System- und den Stakeholderanforderungen abzugleichen. Der Prozess der Eigenschaftsabsicherung verlagert sich mit einer Annäherung an den Kontrollpunkt mehr auf die Validierung. Im Rahmen der anschließenden Übergabe wird das Produkt gemäß des geforderten Entwicklungsstandes übergeben. Dies bedeutet, dass die Übergabe in Form eines Prototyps aber auch eines fertigen Produktes erfolgen kann. Der Umfang der Übergabe hängt damit zusammen, welchen Komplexitätsgrad das erstellte Modell aufweist. Daraus lässt sich ableiten, ob es direkt übergeben werden kann. Die Übergabe damit zusammenhängender Produktionsschritte und Anweisungen ist zusätzlich eine mögliche Ausgabe. [GO22; VDI21]

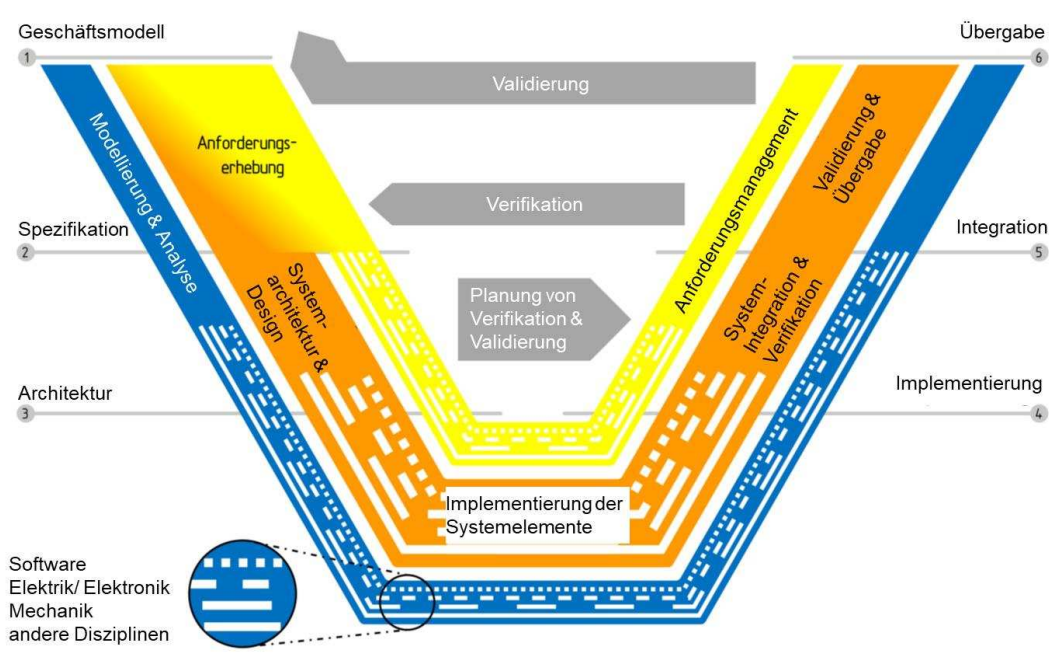


Abbildung 2.1: V-Modell nach der VDI-Richtlinie 2206:2021 [VDI21]

Die Entwicklungsartefakte, welche in den einzelnen Kontrollpunkten des V-Modells erzeugt werden, sind in dem RFLPV<sup>2</sup>-Ansatz [GWO22] genauer erläutert. Es werden das grundlegende Vorgehen und die Elemente definiert. Diese werden im folgenden Unterkapitel genauer erläutert.

## 2.3 Model-Based Systems Engineering

Für die Modellierung eines Systems werden Ansätze des Model-Based Systems Engineering verwendet. Das Model-Based Systems Engineering wird verkürzt als MBSE bezeichnet. In dem Ansatz des Systems Engineerings wird die Informationsübermittlung und Sicherung noch dokumentenbasiert durchgeführt. In dem Model-Based Systems Engineering wird hingegen mit digitalen Partialmodellen gearbeitet. Mit Hilfe dieser Partialmodelle werden alle Phasen des Produktlebenszyklus miteinander verknüpft. Mit den Systemmodellen, welche aus den Partialmodellen bestehen, werden verschiedene Sichten

auf das System ermöglicht. Durch diese Sichten wird das System in der Komplexität reduziert. Somit kann der Anwender befähigt werden für das Anwendungsgebiet der spezifischen Information zu verarbeiten. Zudem wird es im Rahmen des MBSE ermöglicht ein durchgängiges Modell aufzubauen, welches den Vorteil bietet, für Nachfolgermodelle diese Systemmodelle wiederverwerten zu können. Die Rückverfolgbarkeit zwischen einzelnen Systemelementen und den Entwicklungsartefakten muss gegeben sein. Dies kann im Rahmen von Änderungen hilfreich sein, schneller die nötigen Informationen zu erhalten. Für die Modellierung von Systemmodellen wird eine Modellierungssprache verwendet, wobei eine grafische Modellierungssprache für technische Systeme entwickelt wurde. Diese wird im folgenden Kapitel genauer erläutert. [Eig21; GO22]

## 2.4 RFLPV<sup>2</sup>

Neben dem V-Modell, welches ein Modell für den sachlogisch zusammenhängenden Entwicklungsaufgaben darstellt, beschreibt der RFLP-Ansatz die nötigen Entwicklungsartefakte. Im Rahmen dieser Arbeit wird sich auf den erweiterten RFLPV<sup>2</sup>-Ansatz nach GRÄBLER ET AL. [GWO22] bezogen. Dieser umfasst zusätzlich noch die Elemente der Verifikation und Validierung. Neben dieser Erweiterung werden die Elemente der Anforderung, Funktionen sowie logischer und physischer Elemente definiert und wie in Abbildung 2.2 dargestellt auf dem V-Modell verortet. Die einzelnen Elemente des Ansatzes werden im Folgenden genauer erläutert. [GWO22]

Den ersten Elementtyp stellen wie in Abbildung 2.2 zu sehen die Anforderungen dar. Unter den Anforderungen sind neben den Systemanforderungen auch die Stakeholderanforderungen eingeordnet. Betrachtet als Entwicklungsartefakte lassen sich diese in das V-Modell in den Kontrollpunkt Spezifikation einordnen. Die zugeordneten Merkmale variieren dabei je nach Ansatz. [GWO22]

Der zweite Abschnitt umfasst alle Funktionen, welche durch das System zu erfüllen sind. Hierzu kann beispielsweise eine Funktionshierarchie aufgebaut werden, in der beginnend mit der Hauptfunktion des Systems alle weiteren Funktionen abgeleitet werden können. Diese Funktionen sind nötig, um im weiteren Schritt die logischen Elemente zu definieren. Die Funktionen sind dabei als Entwicklungsartefakt mit Betrachtung auf das V-Modell im Bereich der Architektur zu verorten.

Die logischen Elemente bilden den dritten Elementtyp des Ansatzes. Unter logischen Elementen sind lösungsneutrale Wirkprinzipien wie beispielsweise Wirkstrukturen zu fassen. Mit Hilfe dieser Elemente lässt sich das Wirkprinzip des Systems bestimmen und beschreiben. Zusätzlich soll durch die Lösungsneutralität die Lösungsfindung nicht beeinträchtigt werden, um den Lösungsraum nicht zu früh einzuschränken. Ähnlich wie die Funktionen lassen sich die Entwicklungsartefakte der logischen Elemente ebenfalls in dem Bereich der Architektur des V-Modells verorten. [GWO22]

Das vierte Element des RFLPV<sup>2</sup>-Ansatzes bilden die physischen Elemente. Aus physischen Elementen sind in diesem Zusammenhang alle Entwicklungsartefakte zu fassen, welche eine physische Umsetzung oder Implementierung für die Systemstruktur bilden. Das heißt, dass sich die physischen Elemente der Mechanik mit Hilfe von „Computer Aided Design“ (CAD) von der Konstruktion bis zu der tatsächlichen physischen Ausarbeitung eines Elementes des Systems erstrecken. Neben der Disziplin der Mechanik sind die anderen Disziplinen des Systems ebenfalls mit eingebunden. Für den Bereich der Informationstechnik bedeutet dies, dass neben den Elementen für die Informationsverarbeitung sich ebenfalls der erzeugte Programmcode den physischen Elementen zuordnet. Die Verortung dieser Elemente in das V-Modell lässt sich im Bereich der Implementierung durchführen. [GWO22]

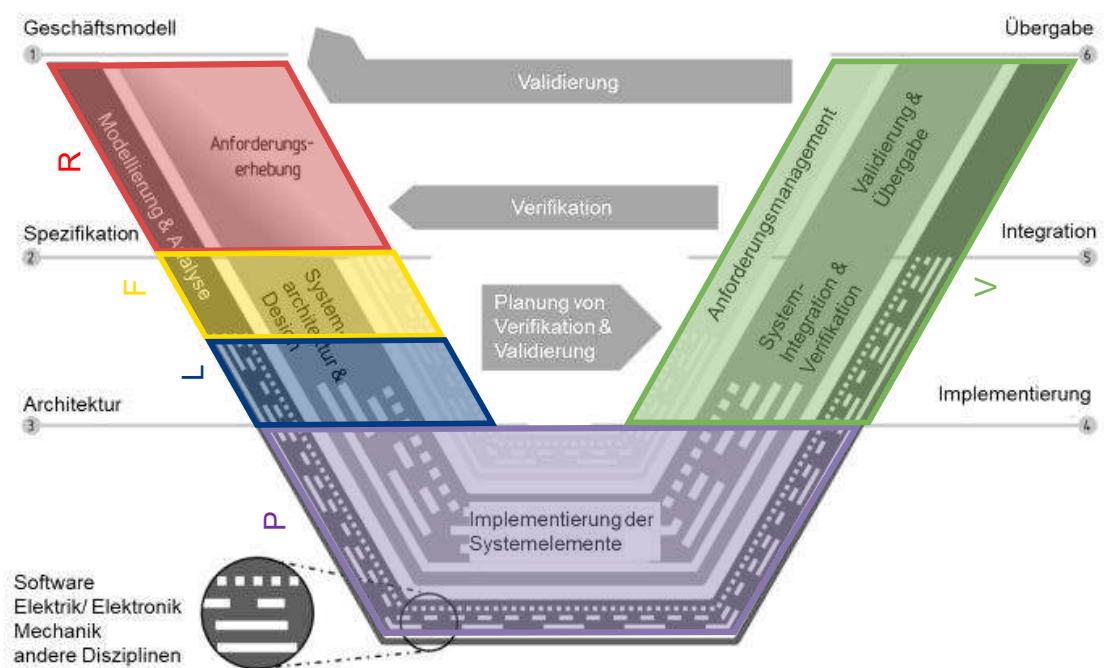


Abbildung 2.2: In Anlehnung an die Integration des RFLPV<sup>2</sup>-Ansatzes in das V-Modell nach Gräßler et al. [GWO22]

Das letzte Element des Ansatzes beschreibt die Verifikation und Validierung. Unter Verifikation und Validierung lassen sich Entwicklungsartefakte wie Testfälle und Testszenarien einordnen. Verifikation und Validierung dienen der Eigenschaftsabsicherung während der Integration. Diese ist nötig, um sicherzustellen, dass das Produkt zum einem richtig entwickelt wird und zum anderen den Bedürfnissen des Stakeholders entspricht. Der Verifikations- und Validierungsprozess lässt sich mit Blick auf das V-Modell auf dem rechten Schenkel, also in dem Kontrollpunkt der Integration bis zu dem Kontrollpunkt der Übergabe verorten. Diese in dem Verlauf der Entwicklung auftretenden Artefakte sind zum einen zu dokumentieren und zum anderen ist bedingt durch den Komplexitätsanstieg ein Bedarf der Modellierung dieser Artefakte nötig. [GWO22]

## 2.5 Entwicklungsartefakte

Unter Entwicklungsartefakte werden alle Elemente gefasst, welche im Verlauf des Produktlebenszyklus durch den Menschen erstellt werden. Dies umfasst Dokumente, Modelle und Zeichnungen. Beispiele für die Entwicklungsartefakte können erhobene Anforderungen, Konstruktionen, Teilelisten oder Betriebsanweisungen sein. Im Folgenden werden die für diese Forschungsfrage wichtigen Entwicklungsartefakte genauer erläutert.

### 2.5.1 Anwendungsfälle

Die Anwendungsfälle werden für die Beschreibung der Systeminteraktion mit einem bestimmten Stakeholder genutzt. Aus der Aufnahme der Anwendungsfälle wird es ermöglicht, ebenfalls implizite Anforderungen der Stakeholder zu erfassen oder bereits erfasste Anforderungen auf ihre Vollständigkeit und Qualität zu überprüfen. In dem Harmony SE Ansatz nach Hoffmann [Hof13-ol], welcher in Kapitel 4.2.3 detailliert erklärt wird, werden aus den Anwendungsfällen die Anforderungen hergeleitet. Für die Definition der Anwendungsfälle werden verschiedene Sichtweisen eingenommen. Im Folgenden werden zwei Definitionen beleuchtet.

In der Definition nach DOUGLASS [Dou21] beschreibt ein Anwendungsfall eine Sammlung an Szenarien oder *User Stories* um eine Nutzung des Systems. Es wird dabei aufgefasst, dass ein Anwendungsfall eine Abfolge an Systemfunktion geleitet durch eine System-Anwender Interaktion darstellt. Somit hat ein Anwendungsfall eine Verbindung zu mehreren Funktionen und mehreren Anforderungen. [Dou21]

In dem Ansatz nach DELLIGATTI [Del14] werden Anwendungsfälle als eine Interaktion zwischen einem Stakeholder und dem beschriebenen System aufgefasst. Dabei initiiert der Stakeholder eine Handlung des Systems. Das System reagiert je nach geltenden Bedingungen mit verschiedenen Verhaltenssequenzen. Diese verschiedenen Verhaltenssequenzen werden durch den Anwendungsfall zusammengefasst. Dabei umfassen die Anwendungsfälle nicht alle Funktionen des Systems, sondern nur die Funktionen, welche durch Interaktionen mit den Akteuren gestartet werden und ablaufen. Die Darstellung mit Hilfe von SysML erfolgt dabei in einem sogenannten Anwendungsfall-Diagramm, welches in Abbildung 2.3 dargestellt ist und in Kapitel 2.6.1 genauer erläutert wird. [Del14]

Für die Beschreibung von Anwendungsfällen können die Anwendungsfall-Spezifikationen genutzt werden. Zum einen muss der Anwendungsfall benannt werden und zum anderen müssen Informationen zu den Vor- und Nachbedingungen und verknüpften Informationen gegeben werden. Darüber hinaus muss übergeben werden, wie der Anwendungsfall gestartet wird und was der optimale Weg der Durchführung ist, also welche Funktionen ausgeführt werden sollen, falls keine Störung vorliegt. Ebenfalls müssen Informationen zu dem primären Akteur gemacht werden und den eingebundenen Stakeholdern. [Del14]

Die beiden Definitionen weisen dabei eine Schnittmenge auf, wobei ein Anwendungsfall als Beschreibung der Kernfunktion des Systems dient. Zusätzlich kann er auch die Interaktion mit Stakeholdern beschreiben, welche das System nicht in der Hauptfunktion verwenden. Anwendungsfälle können dabei in textueller als auch in modellierter Form beschrieben werden, wie bereits DELLIGATTI [Del14] ausführt. Anwendungsfälle können auch dazu genutzt werden, ein genaueres Bild über die Systemfunktion aus Sicht des Stakeholders zu gewinnen. Somit können sie als Hilfsmittel genutzt werden die impliziten Bedürfnisse der Stakeholder zu identifizieren und diese in Anforderungen zu überführen. In diesem Fall besteht eine direkte Abhängigkeit zwischen den einzelnen Anforderungen, bestimmten Systemfunktionen und an den Anwendungsfällen. Darüber hinaus können die Verhaltenssequenzen der Anwendungsfälle für die Gestaltung von Testfällen genutzt werden [Wei07; Was15].

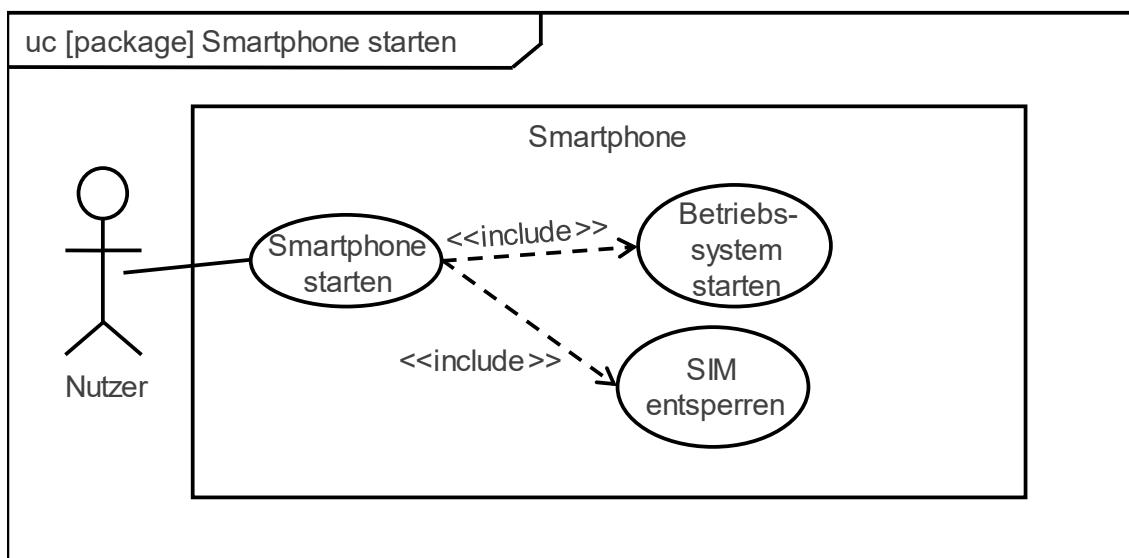


Abbildung 2.3: Beispiel für einen erstellten Anwendungsfall mit Hilfe von SysML

## 2.5.2 Anforderungen

Die Anforderungen werden für die Beschreibung eines zu entwickelnden Systems genutzt. Es gibt unterschiedliche Arten von Anforderungen. Zum einen kann eine Unterscheidung darin getroffen werden, ob die Anforderungen das System direkt beschreiben oder aus den Bedürfnissen des Stakeholders hergeleitet sind. Die Anforderungen, die aus den Bedürfnissen der Stakeholder abgeleitet sind, werden Stakeholderanforderungen genannt. Diese weisen nicht direkt eine technische Beschreibung auf und sind somit nicht immer direkt quantifizierbar. Die Stakeholderanforderungen werden dazu genutzt durch die Validierung nachzuweisen, dass das richtige System entwickelt wurde. Im Gegensatz dazu stehen Systemanforderungen. Diese ermöglichen eine technische Beschreibung des Systems. Mit Hilfe der Systemanforderungen wird die Verifikation durchgeführt. In der Verifikation wird nachgewiesen, dass das System richtig entwickelt wurde. [GO22]

Diese Anforderungen lassen sich weiter in funktionale und nicht funktionale Anforderungen unterteilen. Funktionale Anforderungen sind Anforderungen, die während der Nutzung des Systems durch den Nutzer direkt wahrgenommen werden können. In der Modellierung sind diese mit den Funktionen verknüpft. Diese Anforderungen können also durch die Funktionen erfüllt werden. Nicht funktionale Anforderungen hingegen werden nicht direkt durch den Nutzer wahrgenommen. Diese definieren Randbedingungen oder Systemeigenschaften, die in der Entwicklung eingehalten werden müssen. [GO22]

Zur Bezeichnung der Anforderungen muss diesen eine eindeutige ID und ein Name zugeteilt werden. Darüber hinaus werden sie durch eine kurze textuelle Beschreibung ergänzt. Die Anforderungen können zusätzlich noch in Kategorien eingeteilt werden, welche Hauptgruppen genannt werden [GDB18]. Diese Hauptgruppen beschreiben einzelne Disziplinen oder Beschreibungsmerkmale wie beispielsweise Energie und Geometrie [GDB18]. [GO22]

Die Darstellung von Anforderungen kann mit Hilfe von SysML erfolgen, wie in der Abbildung 2.4 ersichtlich ist. Durch das Anforderungsdiagramm wird die Möglichkeit geschaffen, die Zusammenhänge zwischen einzelnen Anforderungen darzustellen. Dazu gehört zum einen, durch eine Verknüpfung mit der „Satisfy“-Relation darzustellen, welche Elemente die Anforderung erfüllen oder zum anderen, abzubilden, wie aus einzelnen Anforderungen mehrere weitere Anforderungen entstehen können. In diesem Fall wird die „derive“-Relation genutzt. [GO22; Fri14]

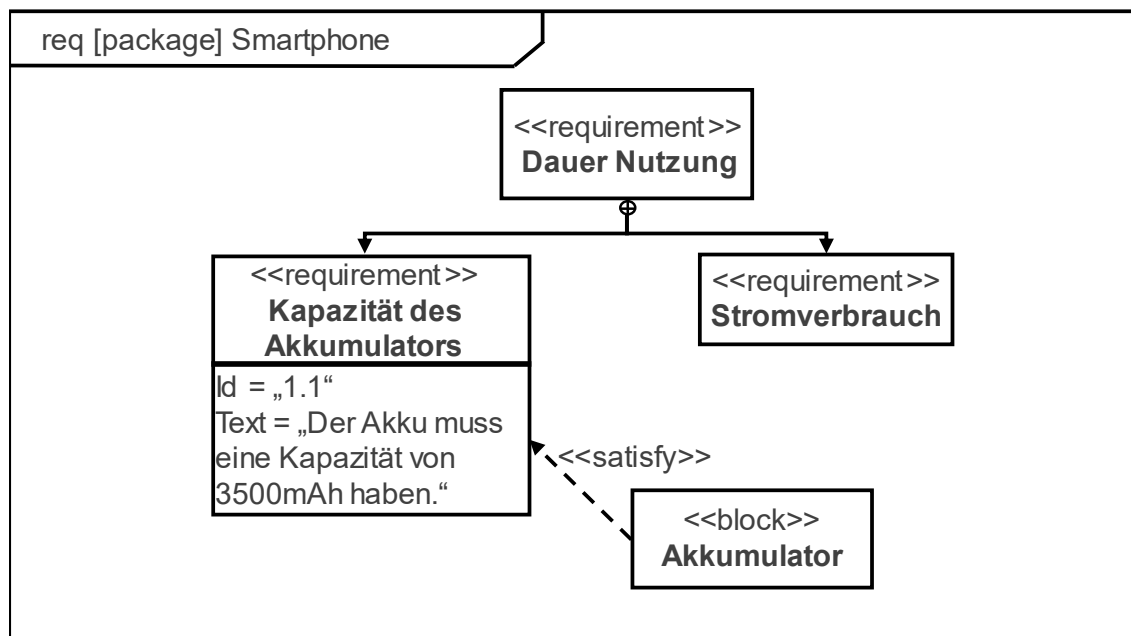


Abbildung 2.4: Anforderungen dargestellt in SysML

### 2.5.3 Testfälle

Testfälle als Entwicklungsartefakte sind der Verifikation und Validierung zugeordnet. Für die Definition von Testfällen existieren verschiedene Ansätze. Definitionsübergreifend werden mit einem Testfall Eigenschaften in Form einer Anforderung abgesichert. Es existieren verschiedene Arten von Tests. Tests können dazu in vier Kategorien unterteilt werden. Diese umfassen Demonstrationen, Tests, Analysen oder Inspektionen. [Wal23]

In der Definition nach HIRSHORN ET AL. [HVB17] stellt ein Test die Nutzung des Endproduktes für die Datensammlung dar. Die daraus resultierenden Daten können für die Verifizierung oder Validierung genutzt werden. Dies wird durch einen Abgleich durchgeführt, ob die Daten die benötigten Leistungen erfüllen. Für den Fall, dass die Daten nicht den direkten Nachweis liefern können, können mit Hilfe der Daten Analysen durchgeführt werden, um diese Leistung zu verifizieren oder validieren. [HVB17]

In der Definition nach DOUGLASS [Dou21] im Rahmen des Model-Based Testing wird ein Testfall so definiert, dass eine Testarchitektur übergeben wird. Zusätzlich müssen Daten mit genauen Werten übergeben werden. Diese umfassen die Sequenz, das Timing und das erwartete Ergebnis. Die Testfälle müssen mit einer Anforderung verbunden werden, wobei mehrere Testfälle einer Anforderung zugeordnet werden können. [Dou21]

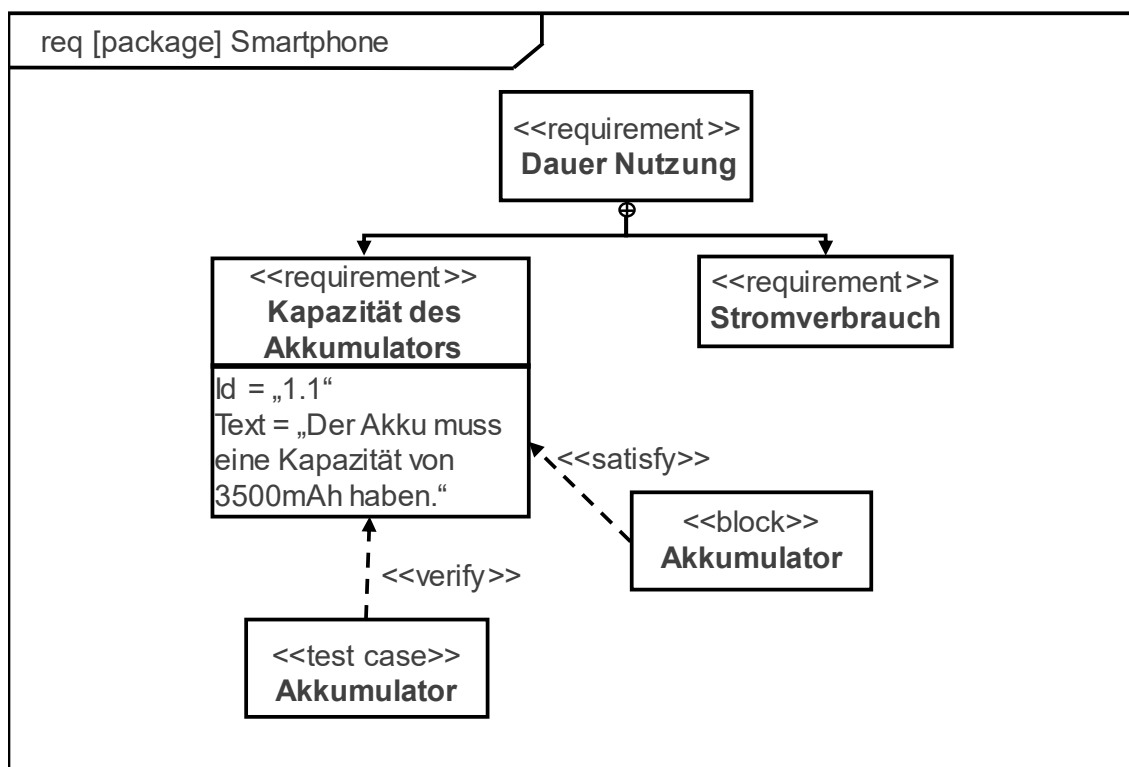


Abbildung 2.5: Exemplarische Darstellung eines Testfalls in SysML

In der Definition nach CONRAD [Con04] bezieht sich ein Testfall auf einen einzelnen Zeitpunkt. Es wird das System in einem bestimmten Ausschnitt betrachtet. Das System



wird mit einem bestimmten Input beaufschlagt, was zu einer Systemreaktion führt. Betrachtet wird dabei ein Testobjekt, welches eine definierte Anzahl an Eingängen bekommt und eine definierte Anzahl an Ausgaben aufweist. [Con04]

Für die Darstellung im Bereich des Model-Based Systems Engineering werden die Testfälle mit Hilfe eines Stereotyps von den zwei UML Metaklassen *Operation* und *Activity* in dem Anforderungsdiagramm definiert. Dies ist exemplarisch in der Abbildung 2.5 dargestellt. Für die genaue Beschreibung besteht die Möglichkeit mit Hilfe von Sequenzdiagrammen oder eines Aktivitätsdiagramms den Ablauf des Testfalls abzubilden. Diese Vorgehensweise entspricht der Auffassung nach WEILKIENS [Wei07] und WASSON [Was15]. Es besteht die Möglichkeit, dass jede *Anwendungsfall*-Aktivität auch ein potentieller Testfall ist [Wei07; Was15]. Gemäß WASSON können aus einem Anwendungsfall bis zu  $n$  von Testfällen erzeugt werden [Was15]. [Fri14; Del14]

Im Rahmen dieser Forschungsfrage wird sich auf Erkenntnisse der genannten Definitionen bezogen. Dabei stellt ein Testfall ein statisches Ereignis dar, wobei ein Testobjekt mit einem definierten Eingang und Ausgang beschrieben wird. Dies ist exemplarisch in der Abbildung 2.6 dargestellt. Dabei weisen Anwendungsfälle - besonders in Bezug auf funktionale Tests – eine Verbindung zur Erstellung der Testfälle auf. Zusätzlich werden die Testfälle dazu genutzt eine bestimmte Anforderung zu verifizieren oder zu validieren.

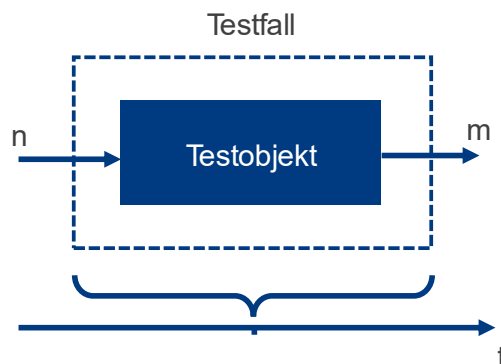


Abbildung 2.6: Grafische Darstellung für die Beschreibung eines Testfalls in Anlehnung an Conrad [Con04]

#### 2.5.4 Testszenarien

Das Entwicklungsartefakt eines Testszenarios ist gemäß dem Testfall in der Verifikation und Validierung zu verorten. In der Definition nach CONRAD [Con04] weisen Testszenarien dabei eine zeitliche Sequenz auf, in der mehrere Testfälle nacheinander durchgeführt werden. Dabei werden für die einzelnen Testfälle bestimmte Zeitpunkte definiert, zu denen diese durchgeführt werden. Ebenfalls muss ein Testfall als Initiator übergeben werden. [Con04]

Die Darstellung von Testszenarien als eine dynamische Abhandlung an Ereignissen wird ebenfalls in der Definition nach ULBRICH ET AL. [UMR+15] aufgeführt. Im Unterschied zu CONRAD [Con04] besteht bei ULBRICH ET AL. [UMR+15] ein Testfall aus mehreren möglichen Testszenarien, wobei sich das Testszenario aus den Anwendungsfällen ableitet und den funktionellen Gebrauch des Systems nachweist. Ebenfalls müssen Informationen zu den Randbedingungen und den eingebundenen Akteuren gegeben werden. Zusätzlich ist anzugeben, welche Ziele die einzelnen Abfolgen an Ereignissen haben. [UMR+15]

Für die textuelle Beschreibung von Testszenarien ist in dem Ansatz nach BAI ET AL. [BTP+02] ein Template zu übergeben. In diesem Ansatz werden die zu übergebenden Informationen genauer spezifiziert. Die Informationen umfassen dabei grundsätzliche Informationen wie eine ID, einen Namen, eine Beschreibung und die Ein- bzw. Ausgabe. Für das Testszenario müssen ebenfalls alle Referenzen übergeben werden. Dies umfasst die Entwicklungsartefakte, welche in mit dem Testszenario in Verbindung stehen. Zusätzlich muss die Information übergeben werden, welche Sub-Szenarios enthalten sind. [BTP+02]

Im Rahmen dieses Forschungsansatzes wird das Testszenario als ein dynamisches Ereignis aufgefasst. Es liegen somit einzelne Testfälle in sequenzieller Abfolge gemäß CONRAD [Con04] vor. Dies ist in der Abbildung 2.7 dargestellt. Zusätzlich ist eine genaue Beschreibung gemäß dem Ansatz nach BAI ET AL. [BTP+02] vorzusehen. Hierdurch kann sichergestellt werden, dass die Informationen zu den zugeordneten Anforderungen und den Anwendungsfällen übergeben werden. Des Weiteren wird gemäß dem Ansatz nach ULBRICH ET AL. [UMR+15] die Verifikation und Validierung dynamischer funktioneller Anforderungen mit Hilfe von Testszenarien durchgeführt. Somit kann übersetzt in den SysML-Kontext angenommen werden, dass das Testszenario dekomponiert werden kann und auf einer tieferen Systemebene die Testfälle stehen, welche das Testszenario beschreiben.

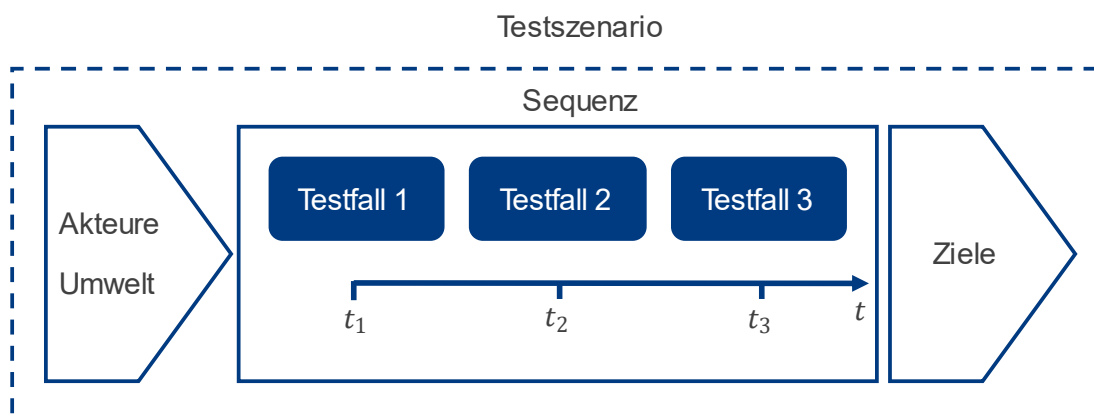


Abbildung 2.7: Darstellung der Beschreibung eines Testszenarios

## 2.6 Modellierungssprache SysML

Die Modellierungssprache System Modelling Language, kurz SysML, ist eine grafische Modellierungssprache für technische Systeme. Diese Modellierungssprache basiert auf der Unified Modelling Language, kurz UML. Die Modellierungssprache UML wird für die Entwicklung von IT-Systemen und Software verwendet. Somit können Basisfunktionen übernommen werden. Zusätzlich werden die Erweiterungen für die ingenieurstechnische Anwendung hinzugefügt. Die erarbeiteten Modelle ermöglichen die Visualisierung von Zusammenhängen der einzelnen Entwicklungsartefakte mit unterschiedlichen Sichten auf das System. Zusätzlich wird durch die Modellierungssprache die interne Struktur der Systeme besser identifizierbar, wodurch Wirkungszusammenhänge leichter abgebildet und analysiert werden können. Für die Auswertung dieser Zusammenhänge können beispielsweise hinsichtlich bestimmter Relationen die Abhängigkeiten zweier Systemartefakte untersucht werden. Damit die einzelnen Sichtweisen auf das System ermöglicht werden, bedarf es unterschiedlicher Diagramme, Modellierungselemente und Relationen zwischen den Elementen. Diese Diagramme, Modellierungselemente und Relationen werden in den folgenden Unterkapiteln genauer erläutert. [Fri14; Del14; GO22]

### 2.6.1 Diagrammtypen

Insgesamt gibt es neun unterschiedliche Diagrammtypen in der Modellierungssprache SysML. Diese sind in der Abbildung 2.8 dargestellt. Dabei weisen alle Diagrammtypen Gemeinsamkeiten auf. Alle Diagramme haben einen Rahmen und ein Kürzel in dem Diagrammkopf. In den folgenden Abschnitten werden die Diagramme, welche in den Modellen verwendet werden, erläutert. [GO22; Del14]

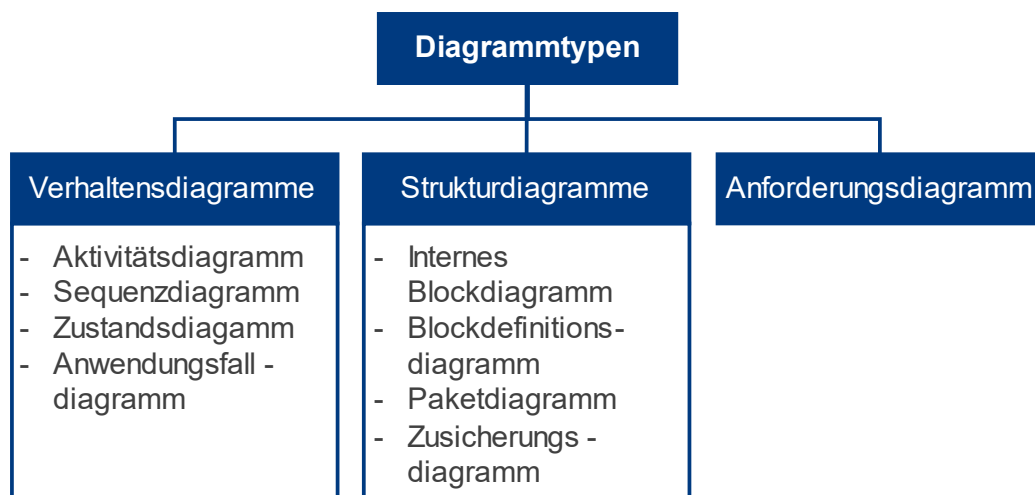


Abbildung 2.8: Übersicht von Diagrammtypen in SysML

Aktivitätsdiagramme stellen Diagramme dar, in denen die Funktionen oder Anwendungsfälle als Aktivität dargestellt werden können. Das Kürzel aus dem Diagrammkopf ist „act“. Dabei wird der Ablauf des Verhaltens mit einzelnen Aktivitäten abgebildet. Das

Diagramm weist dabei jeweils einen Start- und Endknoten auf. In dem Aktivitätsdiagramm ist nur die Verwendung von Aktivitäten erlaubt. [Del14]

Das Anforderungsdiagramm hat das Kürzel „req“ und ermöglicht die Verwendung von Elementen des Typs „package“, „model“, „modelLibrary“, „view“ und „Requirement“. Diese Diagramme werden genutzt, um die Anforderungsstruktur darzustellen. Hieraus kann die Verknüpfung aus den Stakeholderanforderungen zu den Systemanforderungen abgeleitet werden. Die Anforderungen in diesem Diagramm werden mit den Entwicklungsartefakten verknüpft, durch die sie erfüllt werden. Beispiele für diese Entwicklungsartefakte sind für funktionale Anforderungen die zugehörige Funktion oder für nicht funktionale Anforderungen entweder ein zugeordneter Parameter oder ein Systemelement. [Del14; GO22]

Das Blockdefinitionsdiagramm ermöglicht die Verwendung der Elementtypen „package“, „model“, „modelLibrary“, „view“, „block“ and „constraintBlock“. Die Kopfzeile beinhaltet das Kürzel „bdd“. Mit Hilfe des Blockdefinitionsdiagramms kann die Verbindung einzelner Elemente untereinander oder bezüglich gewisser Parameter dargestellt werden. Zum Beispiel kann die Hierarchie eines Systems mit einem Blockdefinitionsdiagramm dargestellt werden.

Das interne Block Diagramm hingegen ermöglicht lediglich die Nutzung des Elementtyps „block“ und weist das Kürzel „ibd“ auf. Das interne Blockdiagramm wird verwendet, um beispielsweise die interne Struktur eines Systems und die Energieflüsse darzustellen, welche zwischen den Systemelementen vorliegen. Durch diese innere Struktur wird die Verknüpfung der gegebenen Eingaben und der erstellten Ausgaben hergestellt. [GO22; Del14]

Das Anwendungsfall Diagramm wird verwendet, um die Anwendungsfälle eines Systems darzustellen. Die zu verwendenden Systemelementtypen umfassen „package“, „model“, „modelLibrary“, „view“; das Diagramm hat den Diagrammkopf „uc“. Für die Darstellung des Anwendungsfalls wird der Stakeholder, welcher in dem Anwendungsfall involviert ist, außerhalb einer festgelegten Systemgrenze platziert. Vergleichbar wie in der Gestaltung der Funktionshierarchie werden die durchzuführenden Systemaktionen modelliert. Die Gestaltung dieser Systemaktionen erfolgt dabei ebenfalls in funktioneller Schreibweise. [Del14; GO22]

## 2.6.2 Modellierungselemente

Für die Modellierung der einzelnen Sichten des Systems werden verschiedene Arten von Modellierungselementen verwendet. Jedes Element hat andere Eigenschaften und kann nur innerhalb vorgesehener Diagramme verwendet werden. Die Modellierungssprache sieht jedoch auch die Möglichkeit vor, mit Hilfe von Erweiterungen über das Anlegen von Stereotypen die Auswahl an Modellierungselementen zu erweitern. Im Folgenden wird eine kurze Übersicht der zentralen Modellierungselemente gegeben. [Del14; GO22]

Ein zentrales Modellierungselement stellt der Block dar. Einem Block können verschiedene Arten an Eigenschaften oder Operationen übergeben werden. Im Rahmen der Modellierung technischer Systeme wird mit Hilfe von Blöcken die Systemstruktur in dem Blockdefinitionsdiagramm erstellt. Die Blöcke zählen dabei zu den Definitionstypen und stellen eine Einheit dar und keine Instanz. [GO22; Del14]

Ein weiteres Modellierungselement bildet die Anforderung. Mit Hilfe dieses Elements kann zum einen die Anforderungsstruktur erstellt werden. Zum anderen werden hier die nötigen Informationen übergeben wie eine ID und eine textuelle Beschreibung. Die Anforderungen können mit Hilfe von Stereotypen so erweitert werden, dass weitere Unterteilungen neben funktionalen und nicht funktionalen Anforderungen möglich sind. [Del14; GO22]

Ein weiteres zentrales Modellierungselement ist die Aktivität. Dieses Element kann dazu genutzt werden ein Verhalten des Systems darzustellen. Die Aktivitäten können innerhalb des Aktivitätsdiagramms verwendet werden. Mit Hilfe weiterer Elemente wie Entscheidungsknoten oder eine Nachrichtenübermittlung können komplexe Systemverhalten dargestellt werden. [GO22; Del14]

### 2.6.3 Relationen

Für die Verbindung der Systemelemente werden verschiedene Arten von Relationen verwendet. Hierdurch können Relationen zwischen den einzelnen Systemelementen vergeben werden. Die verwendbaren Relationen unterscheiden sich je nach Diagrammtyp. Zusätzlich besteht die Möglichkeit die Liste an Relationen zu erweitern. Ein Auszug der üblichen Relationen ist in der Abbildung 2.9 dargestellt. [GO22]

Bezeichnung	Bedeutung
<<satisfy>>	Verknüpfung zwischen Anforderung und Entwicklungsartefakt
<<verify>>	Verknüpfung zwischen Anforderung und zugehörigen Testfall
<<trace>>	Verknüpfung zwischen zwei Elementen für die Rückverfolgbarkeit
<<derive>>	Verknüpfung für eine abgeleitete Anforderung
<<refine>>	Verknüpfung zwischen zwei Anforderungen oder zu einem Use Case

Abbildung 2.9: Beispiele für Relationen in SysML Diagrammen

## 2.7 Engineering Change-Management

Unter Engineering Change-Management gefasste Ansätze befassen sich damit, wie eine Änderung eines bestehenden Designs durchzuführen ist. Für die genaue Definition des Engineering Change-Management gibt es verschiedene Quellen. Im Rahmen dieser Arbeit wird sich auf die Definition nach JARRATT [Jar05] bezogen. In dieser Definition

wird ein Engineering Change als eine Änderung von Teilen, Zeichnungen und Software bezeichnet, wenn diese nach der Entwurfsphase bereits freigegeben sind. Der Umfang hat dabei keine Bedeutung und umfasst somit kleine Änderungen mit wenig Personalbedarf sowie Änderungen mit einem sehr hohen zeitlichen und personellen Aufwand. [Jar05] Diese Definition wird nach LASHIN [BG21] erweitert, durch das einbinden von produktbeschreibenden Dokumenten und der beispielhaften Beschreibung der Änderung hinsichtlich der Funktion und Gestalt. [BG21]

Für die genaue Betrachtung in welchen Stadien der Entwicklung in welchen Stadien der Entwicklung ein solcher technischer Änderungsprozess denkbar ist, kann die Abbildung 2.10 eingesehen werden. Dargestellt ist der Produktlebenszyklus von der Planung hin bis zu der Dekommissionierung. Eine Änderung ab dem Prozessschritt des System Designs bis zu der Fertigung des Produkts wird als technische Änderung betrachtet. Somit ist in diesem Sinne eine Änderung während des Testprozesses ebenfalls eine technische Änderung.

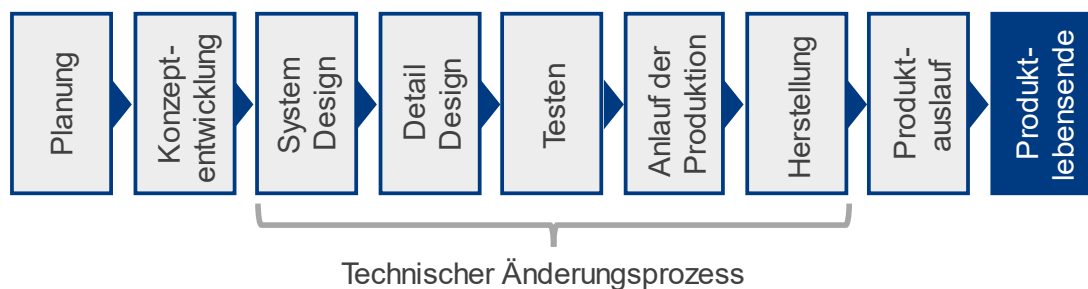


Abbildung 2.10: In Anlehnung an den Generischen Produktentwicklungsprozess nach Ulrich und Eppinger [UE11]

Der Prozess des Engineering Change wird nach JARRETT [Jar05] in einem generischen Engineering Change Prozess beschrieben. Der Prozess besteht dabei wie in Abbildung 2.11 dargestellt aus sechs Schritten. Zwischen den sechs Schritten existieren vier Haltepunkte, an denen Absprachen getroffen werden oder auf eine Zustimmung für das Fortfahren des Prozesses gewartet wird. Zusätzlich sind zwei mögliche Iterationsschleifen vorgesehen. Es wird darauf abgezielt, dass die Prozesse der Lösungsfindung und der Risikoanalyse bei Bedarf wiederholt durchzuführen sind. Eine zusätzliche Einteilung der Schritte erfolgt darüber, ob sich der Engineering Change noch vor der Beantragung befindet, noch abgewickelt wird oder bereits beantragt wurde. Bevor ein Engineering Change beantragt werden kann, müssen zu Beginn eine Anfrage aufkommen und für diese Anfrage mögliche Lösungen sowie die zugehörigen Risikobeurteilungen erarbeitet werden. Im Rahmen dieser Forschungsfrage befindet sich das Thema in dem dritten Schritt, der Risikobeurteilung in Form der Auswirkungsanalyse für eine Änderung. [Jar05]

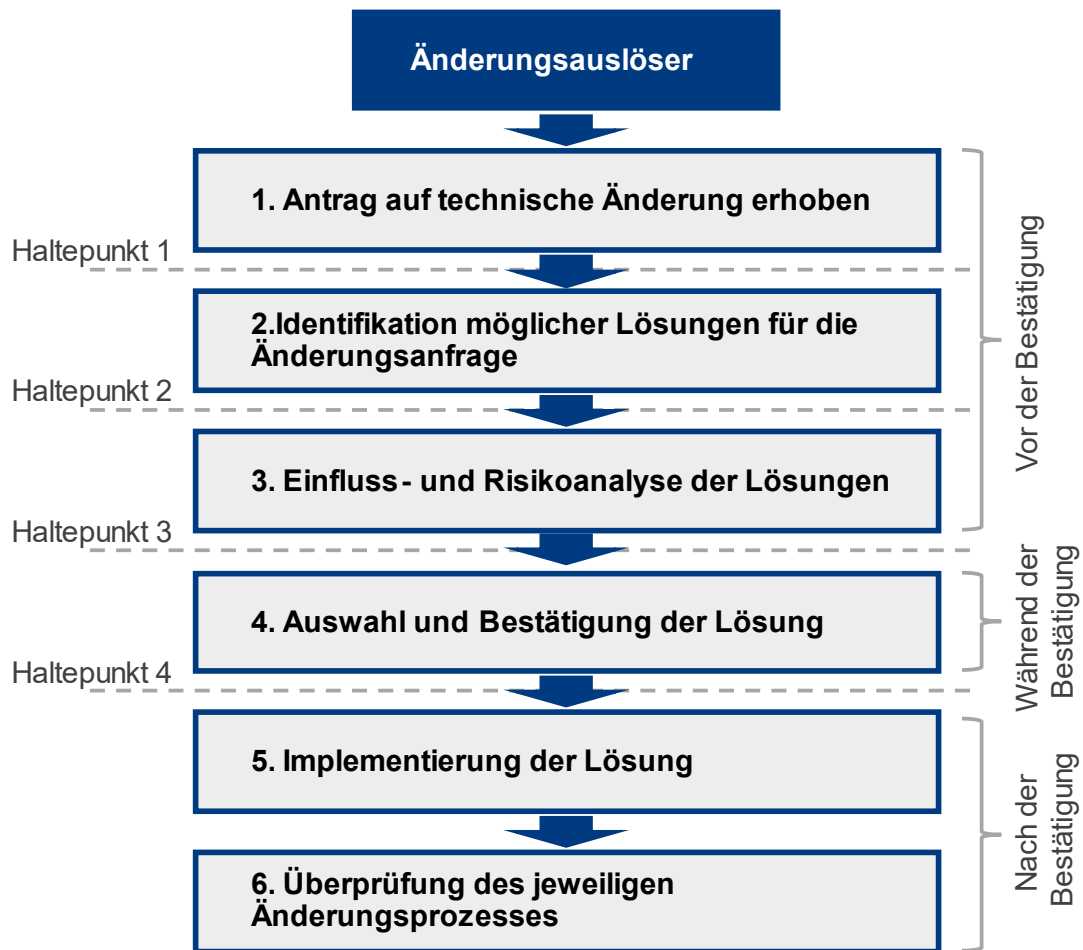


Abbildung 2.11: In Anlehnung an den generischen Engineering Change Prozess nach Jarrett [Jar05]

### 2.7.1 Strategien für Umgang mit technischen Änderungen

Für den Umgang mit technischen Änderungen gibt es verschiedene Strategien. In der Veröffentlichung nach FRICKE ET AL. [FGN+00] werden fünf Strategien dargelegt, welche aus Studien hervorgehen. Diese Strategien umfassen präventive als auch reaktive Maßnahmen.

Die erste Strategie ist die Prävention. Präventive Maßnahmen umfassen beispielsweise eine genauere Analyse der Anforderungen oder des Systems. Durch diese kann das Risiko von Änderungen minimiert werden. Neben der reinen Analyse wird herausgestellt, dass ebenfalls das Prüfen auf unnötige Anforderungen das Risiko minimiert Änderungen durchführen zu müssen. Zusätzlich sollten nicht zu früh das Design oder Anforderungen festgelegt werden, um die Flexibilität im Entwurfsprozess nicht einzuschränken. [FGN+00; Lan16]

Die zweite Strategie ist das „Frontloading“. Bei dieser Strategie wird frühzeitig im Entwicklungsprozess die Verifikation und Validierung durchgeführt. Durch das Verlagern der Eigenschaftsabsicherung in die früheren Phasen des Entwicklungsprozesses werden

die Änderungskosten durch die Fehlererkennung minimiert. Für frühzeitige Eigenschaftsabsicherungen müssen Simulationen mit eingebunden werden, ebenso bereits jetzt die späteren Nutzer des Systems. [FGN+00; GO22]

Die dritte Strategie umfasst die Steigerung der Effektivität. In dieser Strategie wird darauf abgezielt unnötige Änderungen zu vermeiden. Dies umfasst Änderungen, welche nicht die Nutzeranforderungen erfüllen. Dabei beruhen unnötige technische Änderungen häufig auf persönlichem Empfinden gegenüber der präferierten Lösung. Zusätzlich besteht die Gefahr, dass Änderungsschleifen durchlaufen werden. Das bedeutet, dass nach einigen Änderungen wieder die Ausgangslösung vorliegt. Dies ist darin begründet, dass die Änderungen am Ende gegenüber der Stakeholderanforderungen zu validieren sind. [FGN+00]

Die vierte Strategie beschreibt die Steigerung der Effizienz. Es steht im Fokus, dass Änderungen auf Entscheidungen basieren. Diese Entscheidungen müssen getroffen und auch umgesetzt werden. Bei einer Verzögerung der Umsetzung kann dies dazu führen, dass sich die Basis für die Entscheidung bereits geändert hat. Somit kann wiederum die Änderung hinfällig werden oder eine Änderung der Basis wäre nicht notwendig gewesen. [FGN+00]

Die letzte Strategie ist das Lernen aus den Änderungen. Diese Strategie zielt darauf ab in den Änderungen eine Möglichkeit zu sehen, das Produkt und den zugrundeliegenden Prozess zu verbessern. Somit sollten durchgeführte Änderungsprozesse analysiert werden, um hier Schlüsse für zukünftige Änderungen ziehen zu können. [FGN+00]

### **2.7.2 Auswirkungsmuster und Propagationseffekte**

Änderungen können dazu führen, dass diese sich über das geänderte Element hinaus auf weitere Systemelemente ausbreiten. Dieses Phänomen wird Propagationseffekt genannt. Für die Analyse dieser Auswirkungen müssen die Relationen einzelner Systemelemente zueinander untersucht werden. Es können direkt verknüpfte Parameter zu den Änderungen führen, als auch indirekt verknüpfte Parameter. Um die Ausbreitung von einem Element zu einem anderen zu bestimmen, sind in einer Studie nach ECKERT ET AL. [ECZ04] Verhalten von einzelnen Elementen definiert. Aus der Studie gehen vier Arten von Elementen hervor:

1. Konstanten (engl.:constants)
2. Absorber (engl.:absorbers)
3. Träger (engl.:carriers)
4. Vervielfältiger (engl.:multiplier)

Unter den Konstanten sind Systemelemente zu verstehen, welche durch die Änderung nicht beeinflusst werden. Diese verursachen selbst keine Änderung, nehmen aber auch



keine Änderung auf. Die Absorber geben eine geringere Zahl an Änderungen weiter als diese aufnehmen. Sonderfälle dieser Gruppe absorbieren eine Änderung komplett. Die Träger nehmen dieselbe Anzahl an Änderungen auf, wie sie selbst wieder verursachen und erhöhen somit nicht die Änderungskomplexität. Vervielfältiger hingegen verursachen mehr Änderungen als sie aufnehmen können. Hierdurch wird die Änderungskomplexität erhöht. [ECZ04]

Neben den Verhaltensweisen der einzelnen Elemente werden in der Veröffentlichung ebenfalls daraus resultierende Muster erläutert. Es treten drei Arten von Mustern auf: die Wellen, die Blüten und die Lawinen. Die Muster unterscheiden sich in dem zahlenmäßigen Anstieg der auftretenden Änderungen, als auch durch die Vorhersehbarkeit des Auftretens der Muster. Bei den Wellen und den Blüten handelt es sich um Muster, die vorauszusagen sind und aufgrund der Verhaltensweise verarbeitbar sind. Die Wellen beschreiben häufig auftretende Änderungen, die jedoch nur geringe Änderungen umfassen. Diese Änderungen werden durch Absorber aufgenommen. Die Blüten hingegen weisen einen hohen Anstieg der Anzahl an Änderungen auf. Es entsteht zu Beginn ein hohes Arbeitsvolumen, jedoch werden diese Änderungen ebenfalls durch Absorber aufgenommen. Im Gegensatz dazu stehen die sogenannten Lawinen, diese sind unvorhersehbar und haben ein hohes Potential, das Volumen der Änderung zu steigern. Dadurch kann die Änderungszahl durch Vervielfältiger ansteigen und die Änderungskomplexität erhöhen.

### 3 Stand der Technik

Dieses Kapitel stellt unterschiedliche Ansätze dar, welche die Auswirkungen auf Anforderungen, Anwendungsfälle und Testfälle bedingt durch Änderungen beschreiben. Dabei werden die unterschiedlichen Vorgehen sowie die Konzepte hinter den Ansätzen erläutert. Die Ansätze sind, durch die in Kapitel 4 beschriebene Literaturrecherche identifiziert worden. Die identifizierten Ansätze werden mit Hilfe von Kriterien für die zu entwickelnde Handlungsmaßnahme bewertet und abgegrenzt.

#### 3.1 Auswirkungsanalyse mit Hilfe der Designstruktur Matrix

Für die Auswertung von Auswirkungen werden entweder Designstruktur-Matrizen oder Designabhängigkeits-Matrizen genutzt. Ziel ist die Nachvollziehbarkeit der Änderungsauswirkung. Es werden die Verbindungen zwischen Komponenten, Parametern oder Funktionen hergestellt, welche die Grundlage für die Auswertung bilden. Zur Auswertung können sowohl rechnerische Verfahren genutzt als auch Verfahren herangezogen werden, die mit der Auswertung von Mustern arbeiten.

##### 3.1.1 Auswirkungsanalyse mit SysML für die Identifizierung des Aufwands

In dem Ansatz nach ELLSEL et al. [ESS22] wird eine Auswirkungsanalyse mit Hilfe eines SysML-Modells durchgeführt. Für die Auswirkungsanalyse werden die Abhängigkeiten zwischen den einzelnen Entwicklungsartefakten betrachtet. Mit Hilfe einer Änderungs-Propagations-Analyse wird eine Kategorisierung der Auswirkungen durchgeführt. Das Vorgehen ist als IPO-Diagramm in der Abbildung 3.1 dargestellt. [ESS22]

Input	Prozess	Output
Systemmodell	<ol style="list-style-type: none"> <li>1. Abhängigkeitstypen erstellen</li> <li>2. Systemelemente parametrisieren</li> <li>3. Wirkungsanalyse durchführen</li> <li>4. Erweiterung durch Programmierung</li> </ol>	Liste mit Auswirkungen

Abbildung 3.1: IPO-Diagramm für das Vorgehen in Anlehnung an Ellsel et al. [ESS22]

Das Vorgehen ist in dem Ansatz so aufgebaut, dass als Ausgangslage die Entwicklungsartefakte modelliert vorliegen. Dabei muss der Aufwand für die Umsetzung einer Änderung des Artefakts in Stunden quantifizierbar sein. Es wird zwischen vier Änderungstypen unterschieden, die in der Abbildung 3.2 dargestellt sind. Die Änderung umfassen dabei einfache Änderungen hin bis zu einer kompletten Änderung des Systemartefakts. Zusätzlich werden die Verbindungen der einzelnen Systemartefakte quantifiziert. Somit wird je nach Einfluss ein anderer Zahlenwert aus dem Array der Relation zugeordnet. Für die Berechnung des möglichen Aufwandes wird der PERT-Ansatz gewählt. Bei diesem

Ansatz wird ein Mittel aus einem optimistischen, einem pessimistischen und dem wahrscheinlichsten Wert verrechnet. Für die genaue Modellierung der einzelnen Systemartefakte wird in diesem Ansatz keine Angabe gemacht. Es ist lediglich das Hinzufügen der Kategorisierungen gemäß der Abbildung 3.2 vorzusehen. [ESS22]

Änderungsart	Beschreibung
Geringfügige Änderung	Nur geringfügige Teile des Systemelements müssen geändert werden, wie Parameteränderungen.
Einfache Änderung	Einfache Änderungen an dem Systemelement wie das Hinzufügen von Parametern
Umfassende Änderung	Substanzielle Änderung des Systemelements wie dessen logisches Verhalten.
Komplette Änderung	Die Änderung erfordert denselben Aufwand, wie eine komplette Neuentwicklung.

Abbildung 3.2: Änderungsarten gemäß des Ansatzes nach Ellsel et al. [ESS22]

### 3.1.2 Auswirkungsanalyse anhand von Mustern

Der Ansatz für die Auswirkungsanalyse nach LI & CHEN [LC07; LC10] ist für Änderungen im Design des Systems vorgesehen. Dabei handelt es sich um einen Ansatz, der mit Hilfe der Weiterverarbeitung der Designabhängigkeits-Matrix (DDM) durchgeführt wird. Im Rahmen der Weiterverarbeitung werden entstehende Muster bewertet und anhand derer die Auswirkung bestimmt. Die Muster werden dann einem vorab entwickelten Muster aus der Bibliothek zugeordnet. Somit kann dann das weitere Vorgehen bestimmt werden. Diese können ohne weitere Zerlegung des bestehenden Systems implementiert werden. Die Methode ist in der Abbildung 3.3 als IPO-Diagramm dargestellt. [LC07; LC10]

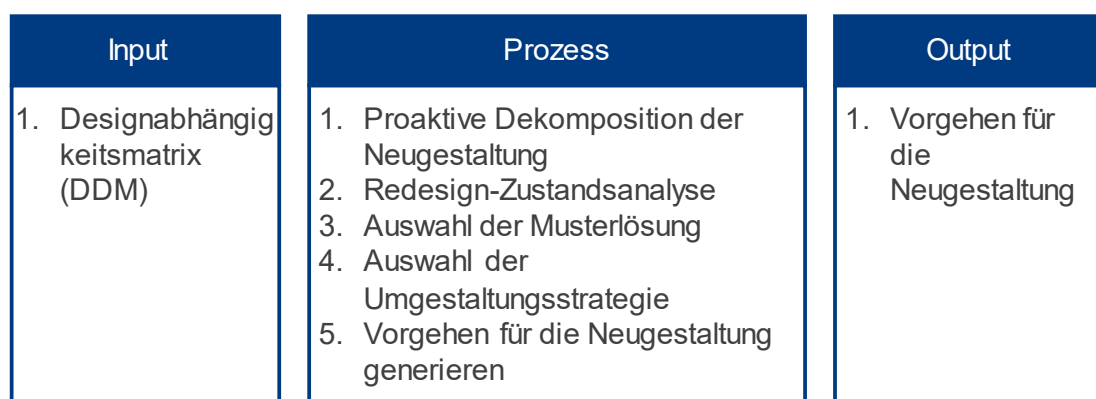


Abbildung 3.3: IPO-Diagramm für das Vorgehen in Anlehnung an Li und Chen [LC10]

In dem Vorgehen wird als Ausgangslage die DDM gewählt, in der die Zeile die Funktionen und die Spalte die Parameter beinhaltet. Im nächsten Schritt wird mit Hilfe der Designabhängigkeits-Analyse die DDM umstrukturiert. Der Vorgang kann zu zwei Ergeb-

nissen führen. Zum einen kann eine entkoppelte Matrix entstehen. Das resultierende Muster stellen dar, dass die einzelnen Cluster keine Interaktionen bei Änderungen haben. Somit bleibt die Änderung innerhalb eines Blocks. Zum anderen ist es möglich, dass eine Banddiagonalmatrix entsteht. In diesem Fall bestehen Abhängigkeiten zwischen den einzelnen Clustern. In diesem häufigeren Fall muss die Matrix für die Identifizierung der Abhängigkeiten aufgeteilt werden. Dazu wird die Neugestaltungs-Partitionierungs-Analyse durchgeführt, indem die Matrix so partitioniert wird, dass für alle anwendbaren Musterlösungen eine Neugestaltungs-Lösung generiert werden kann. Im letzten Schritt kann mit der Neugestaltungs-Planungs-Analyse das Vorgehen für die Neugestaltung erarbeitet werden. Dabei wird zu Beginn ein Neugestaltungs-Problem mit dem geringsten Aufwand ausgewählt. Auf dieser Basis wird die Strategie für die Umgestaltung erarbeitet. [LC10; LC07]

### 3.2 Management von Propagationseffekten für technische Änderungen

In dem Ansatz nach REDDI UND MOON [RM09] ist das Management von Propagationseffekten mit Hilfe eines entwickelten Tools erläutert. Das generelle Vorgehen ist in der Abbildung 3.4 dargestellt. In dem Ansatz wird die Analyse der Änderungen auf Systeme beschränkt, bei denen der Entwurfsprozess abgeschlossen ist. Dies ist darin begründet, dass in diesem Teil des Entwicklungsprozesses weniger Änderungen auftreten. Das Vorgehen basiert darauf, dass durch den Anwender während des Entwurfsprozesses Daten bezüglich der Komponenten und der Relationen übergeben werden. [RM09]

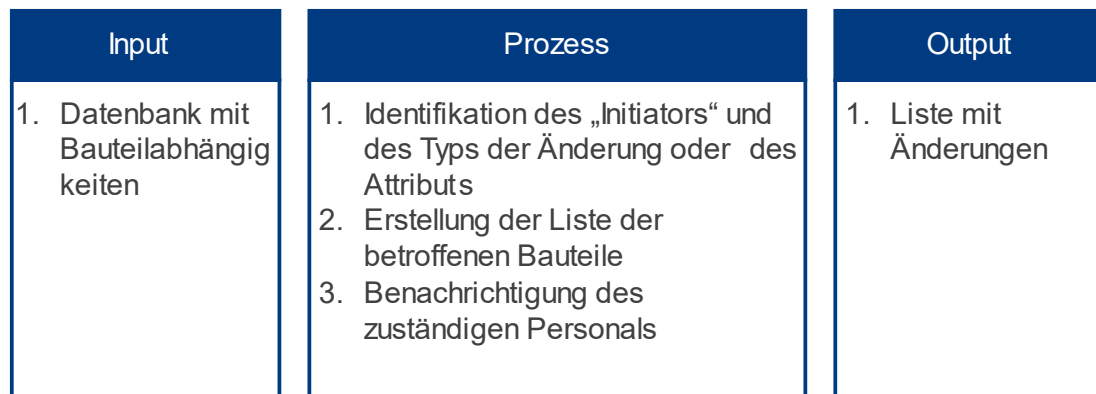


Abbildung 3.4: IPO-Diagramm für das Vorgehen in Anlehnung an Reddi und Moon [RM09]

Nach der Fertigstellung des Entwicklungsprozesses wird eine Datenbank als Input vorausgesetzt, die wie folgt erstellt wird: Das erarbeitete Tool wird verwendet, um die Daten zu erheben. Die Daten werden für den Aufbau der Datenbank verwendet. Die erhobenen Daten umfassen Informationen, welche in diesem Ansatz benötigt werden. Dies sind zum einen die sogenannten technischen Änderungs-Klassen, zum anderen die Auslöser – „Initiatoren“ – der Änderungen. Der Auslöser ist das Element, welches den Anlass der Änderung gibt. Zusätzlich wird die vorgesehene Änderung so definiert, dass diese mit der

Art der Änderung und dem Auslöser eindeutig identifizierbar ist. Das dritte Element ist die Änderungsart, welche definiert, ob es sich um beispielsweise eine Geometrieänderung handelt. Diese Änderungsart ist branchenspezifisch. Als letztes muss die Wahrscheinlichkeit übergeben werden. Diese bestimmt, ob ein Element anfällig für eine bestimmte Art der Änderung ist. Im nächsten Schritt ist nach dem Auftreten einer Änderung das Element zu bestimmen, welches die Änderung auslöst. Zusätzlich muss die Art der Änderung übergeben werden oder die Informationen, welches Attribut geändert werden soll. Durch das anschließende Ausführen des entwickelten Tools wird eine Liste betroffener Elemente ausgegeben. Anschließend werden die zuständigen Personen aus den Fachabteilungen benachrichtigt, um die Änderungen zu prüfen. [RM09]

### 3.3 Auswirkungsanalyse auf Basis der Systemartefakte und Relationen

Eine weitere Art von Auswirkungsanalysen verwendet die Bewertung von Relationen zwischen den Entwicklungsartefakten, wobei hier zwei Darstellungsformen gebräuchlich sind. Zum einen wird das Modell als Graphen dargestellt und zum anderen werden SysML-Modelle genutzt. Die verwendeten Systemartefakte für die Analyse variieren je nach Ansatz.

#### 3.3.1 Auswirkungsanalyse für Testfälle

In den Ansätzen nach INTANA ET AL. [ILS23; IS19] wird die Auswirkung auf Testfälle durch Änderungen in den Anforderungsspezifikationen betrachtet. Der Ansatz ist in der Abbildung 3.5 dargestellt. Hierzu werden die Anwendungsfälle vor und nach der Änderung miteinander verglichen und aus den Mustern die Auswirkungen abgeleitet. [IS19; ILS23]

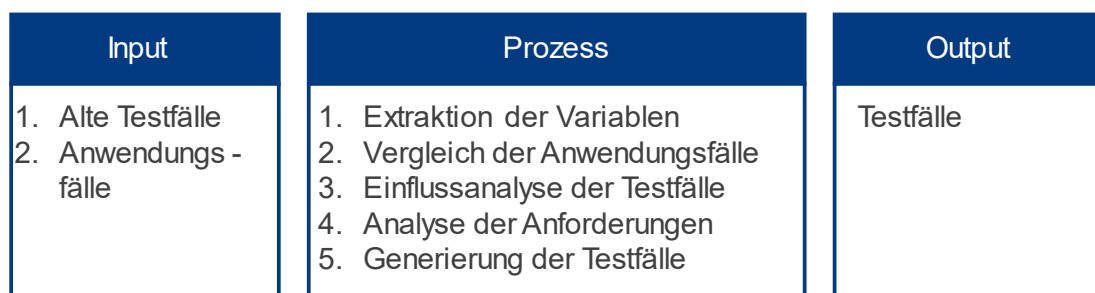


Abbildung 3.5: IPO-Diagramm für das Vorgehen in Anlehnung an Intana et al. [ILS23; IS19]

Zu Beginn werden aus den Anwendungsfällen die Variablen extrahiert, die auf den Anforderungen basieren. Hierzu werden die Anwendungsfälle vor und nach der Änderung betrachtet. Anhand der Ergebnisse wird in dem nächsten Schritt die Änderung durch den Vergleich der Variablen identifiziert. Die identifizierten Variablen werden genutzt, um den Einfluss auf die Testfälle zu detektieren. Die in der Datenbank hinterlegten Vorgänger-Testfälle werden mit Hilfe der Variablen auf Änderungen geprüft. Für den Fall, dass

keine Änderung vorhanden ist, wird der Testfall unverändert übernommen. Wenn eine Änderung festgestellt wird, muss der Testfall entweder geändert oder ein neuer Testfall bezüglich der Anforderungen hergeleitet werden. [ILS23; IS19]

Die Relationen zwischen den einzelnen Entwicklungsartefakten lassen sich in der Form übertragen, dass die Anforderungen die wichtigen Systemparameter und die Anwendungsfälle definieren. Zusätzlich werden die Systemfunktionen mit Hilfe der Anforderungen und Anwendungsfälle beschrieben. Die für die Verifizierung der Anforderungen nötigen Testfälle werden aus den Anwendungsfällen und den zugeordneten Parametern abgeleitet. Das heißt, dass gemäß dem Ansatz eine Verbindung zwischen den einzelnen Anforderungen, den Testfällen, den Parameter und den Anwendungsfällen zu ermitteln ist. [ILS23; IS19]

### 3.3.2 Auswirkungsanalyse anhand der Entwicklungsartefakte und Relationen

In dem Ansatz nach LIN ET AL. [LPS+15] werden die Konzepte des Modellgetriebenen Engineerings (MDE) und dem Management der Änderungsanträge (CRM) vereint. Das Vorgehen ist als IPO-Diagramm in der Abbildung 3.6 dargestellt. Das Ziel ist, durch die Methode die Ingenieure zu befähigen, die Auswirkungen von Änderungen zu ermitteln und die Durchführbarkeit zu prüfen. Die Ergebnisse können anschließend zur Abstimmung über die Änderungen herangezogen werden. Das Vorgehen sieht aufgrund der möglichen Automatisierung von Teilprozessen die Nutzung von SysML vor. [LPS+15]

Input	Prozess	Output
1. Systemmodell in SysML	1. Identifikation funktionaler und nicht funktionaler Anforderungen 2. Identifikation betroffener Systemelemente 3. Identifikation möglich betroffener Systemelemente 4. Lösungsmodell erstellen 5. Ersetzen betroffener Elemente 6. Integrieren neuer Elemente 7. Betroffene Beziehungen und Modellelemente auflisten	1. Modell mit möglichen Änderungen 2. Liste der Änderungen

Abbildung 3.6: IPO-Diagramm für das Vorgehen in Anlehnung an Lin et al. [LPS+15]

Das Vorgehen besteht aus sieben Schritten in zwei Teilprozessen, wobei in dem ersten Teilprozess vier Schritte durchzuführen sind. Der erste Teilprozess umfasst die Identifikation funktionaler und nicht funktionaler Anforderungen, die Identifikation betroffener und möglich betroffener Systemelemente und die Erstellung des Lösungsmodells. Der

zweite Teilprozess umfasst das Ersetzen betroffener Elemente, das Integrieren neuer Elemente und die Auflistung der betroffenen Modellelemente und deren Beziehung. Beginnend mit der technischen Prüfung werden anhand der Problembeschreibung die betroffenen funktionalen und nicht funktionalen Anforderungen identifiziert, welche geändert oder erstellt werden müssen. Für die geänderten oder neu implementierten Anforderungen muss der Anwender die betroffenen Elemente bestimmen, welche eine *trace*- oder eine *satisfy*-Relation zu den Anforderungen aufweisen. Somit werden der Typ und die Instanzen aufgezeigt, bei denen mögliche Änderungen zu finden sind. Alle anderen Elemente, die eine *allocate*-, *aggregation*-, *satisfy*- und *trace*-Relation aufweisen, werden in dem nächsten Schritt ebenfalls als Element mit einer möglichen Änderung markiert. In dem vierten Schritt des ersten Abschnitts wird ein Modell mit den neuen oder geänderten Anforderungen und allen verbundenen Elementen aufgestellt.

In dem zweiten Teilprozess werden die existierenden und zu ersetzenden Elemente identifiziert. Diese werden mit den Elementen des Lösungssystems verbunden. Die neuen Elemente werden integriert und neue nötige Relationen werden geschaffen. Als finaler Schritt wird eine Liste betroffener Elemente und Relationen erstellt. Die Liste beinhaltet die funktionalen und nicht funktionalen Anforderungen. Nach einer Prüfung der Durchführbarkeit kann entschieden werden, ob die Änderung durchgeführt wird. [LPS+15]

### 3.3.3 Methode für die Identifizierung von Propagationspfaden

In der Ausarbeitung nach YANG & DUAN [YD12] ist eine Methode zur Identifizierung von Propagationspfaden anhand von Parameter-Verbindungen dargelegt. In dem Ansatz werden zwei Mechanismen herausgestellt, durch die sich die Änderungen in dem Modell ausbreiten. Als Grundlage für den Ansatz ist das Systemmodell als Netzwerkmodell ausgeführt. Für die Bestimmung der Propagationspfade wird ein Algorithmus verwendet, der in der Abbildung 3.7 als Aktivitätsdiagramm dargestellt ist.

Das Vorgehen ist so aufgebaut, dass zu Beginn der initiale Parameter identifiziert wird, der geändert werden soll. Dieser muss darauf untersucht werden, ob es sich um einen Zielparameter oder einen direkten Parameter handelt. Der Unterschied besteht darin, dass sich ein geänderter Zielparameter auf mehrere Kind-Parameter auswirkt. Somit wird eine sogenannte Änderungsdiffusion angeregt. Für den Fall, dass es sich um einen direkten Parameter handelt, wirkt sich die Änderung auf die Eltern-Parameter aus. Wird in dem ersten Schritt ein direkter Parameter bestimmt, kann der zweite Schritt durchgeführt werden. In dem zweiten Schritt wird untersucht, ob mögliche Folgen auftreten. Wenn dieser Fall eintritt oder im ersten Schritt ein Zielparameter identifiziert wird, wird der dritte Schritt durchgeführt. Dabei wird die Machbarkeit durch eine Untersuchung des Propagationspfades geprüft. Wenn die Änderung nicht durchführbar ist, wird diese verworfen. Andernfalls wird der vierte Schritt durchgeführt. Im vierten Schritt wird geprüft, ob es verbundene Parameter gibt. Wenn dem so ist muss das Vorgehen ab Schritt zwei erneut starten. Ansonsten kann der neue Pfad gewählt werden.

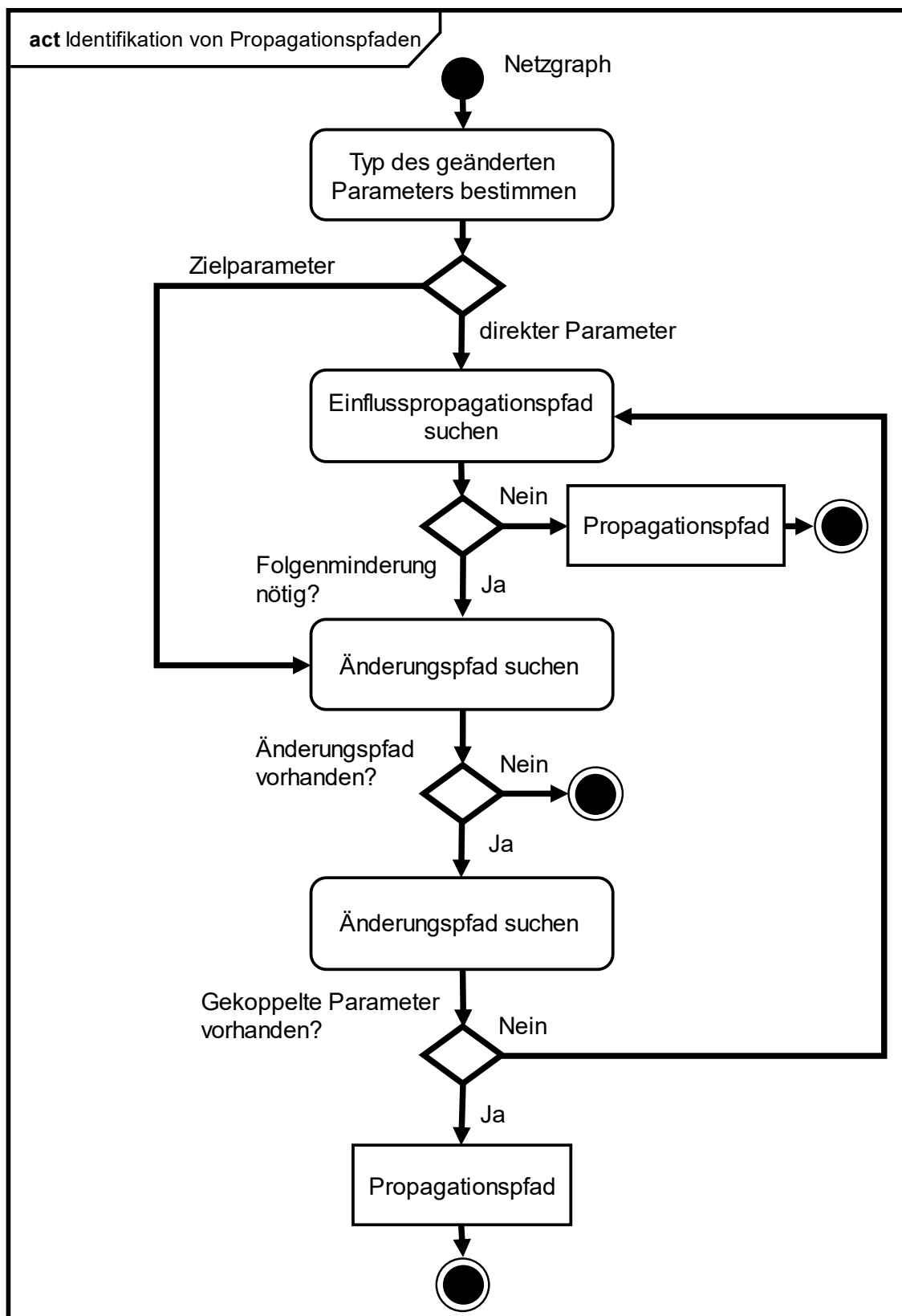


Abbildung 3.7: Vorgehen für die Identifikation von Propagationspfaden als Aktivitätsdiagramm in Anlehnung an Yang und Duan [YD12]



In der Modellierung müssen mathematische Randbedingungen wie beispielsweise Naturgesetze eingehalten werden. Die daraus resultierenden Verbindungen werden fundamentale Verbindung genannt. Ebenfalls existieren Verbindungen durch Nebenbedingungen. Unter Verbindungen durch Nebenbedingungen werden die modellierten Abhängigkeiten gefasst. Der wesentliche Unterschied liegt darin, dass auf fundamentale Verbindungen kein Einfluss genommen werden kann, wobei im Vergleich die Verbindungen durch Nebenbedingungen veränderlich sind. Somit verhält sich die daraus entstehende Ausbreitung der Veränderungen anders. Im Fall der fundamentalen Verbindung wird bei der Änderung eines Kind-Parameters kein weiterer Kind-Parameter verändert, sondern die Änderung breitet sich auf den Eltern-Parameter aus. Zusätzlich können auch innerhalb des Systems weitere mathematische Abhängigkeiten auftreten. Diese sind beispielsweise ein vorgegebenes Maß, welches von mehreren Komponenten gemeinsam nicht überschritten werden darf.

### 3.4 Auswirkungsanalyse mit dem CIRA-Verfahren

In dem Ansatz nach DEUBEL ET AL. [DCK+07] ist ein Verfahren dargestellt, welches sich aus den Vorgehensweisen der Modellierung von Merkmalen und Eigenschaften (CPM) und der eigenschaftsgesteuerten Entwicklung (PDD) bedient. Diese Verfahrensweisen werden mit einem Ansatz ähnlich der Fehlereinfluss- und Auswirkungsanalyse (FMEA) kombiniert. [DCK+07]

Das Vorgehen besteht aus sechs Schritten, welche in der Abbildung 3.8 dargestellt sind. In dem ersten Schritt wird die Struktur der Analyse festgelegt, indem die Kopfzeile ausgefüllt wird. Darauf aufbauend sind alle kritischen Eigenschaften zu definieren. Die kritischen Eigenschaften werden mit Hilfe des CPM/PDD- Modells identifiziert. In dem nächsten Schritt, der Relevanzklassifikation, werden die Eigenschaften bezüglich der Relevanz für das System beschrieben und die Auswirkung durch eine nicht korrekte Realisierung herausgestellt. Es muss beachtet werden, ob es sich um Kundenanforderungen handelt, einzuhaltende Gesetze oder ähnliches. Diese Einordnung beeinflusst, ob es sich um zwingende Anforderungen handelt oder nicht. Der dritte Schritt ist die Lösungssynthese. Für die Synthese wird das CPM-Produktmodell verwendet. Das Ziel ist es potenzielle Lösungen für die technische Änderung zu finden, welche diese positiv beeinflussen. In dem nächsten Schritt der Einflussanalyse wird geprüft, welchen Einfluss die Änderung auf die anderen Eigenschaften des Systems hat. Also werden direkte sowie indirekte Änderungen identifiziert. Der fünfte Schritt ist die Risikobewertung. Die Risikobewertung wird mit drei Kennzahlen durchgeführt, denen ein Wert zwischen eins und zehn zugeordnet wird. Dabei entspricht die eins dem besten erreichbaren Wert und die zehn dem schlechtesten Wert. Die erste Kennzahl beschreibt die Signifikanz der Änderung für das System. Die zweite Kennzahl beschreibt die Wahrscheinlichkeit des Erfolgs der Änderung. Die dritte Kennzahl beschreibt die Wahrscheinlichkeit von Folgeänderungen, welche durch das CPM-Modell bestimmt werden. Die Werte werden multipliziert, wodurch die Reihenfolge der Änderungen festgelegt wird. Der letzte Schritt des Verfahrens ist die

Absicherung, wobei hier die Maßnahmen für die Messung der Lösung festgelegt werden. [DCK+07]

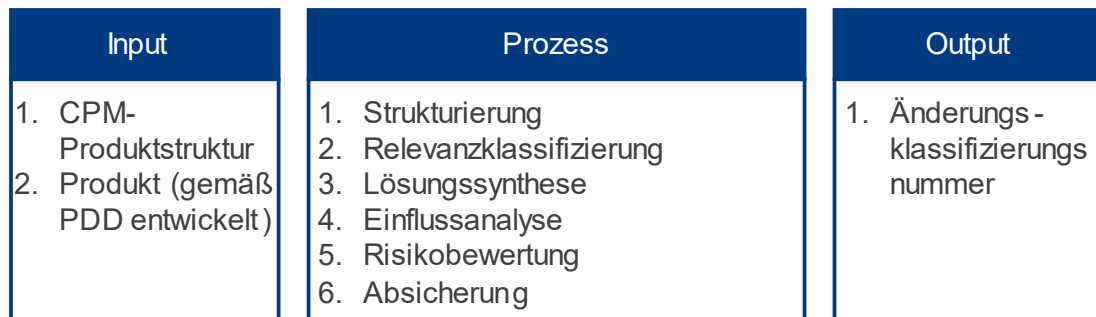


Abbildung 3.8: IPO-Diagramm für das Vorgehen in Anlehnung an Deubel et al. [DCK+07]

### 3.5 Abgrenzung der Ansätze

In dem folgenden Abschnitt werden die aus der Literaturrecherche ermittelten Ansätze für die Auswirkungsanalyse voneinander abgegrenzt. Zusätzlich werden die Methoden im Hinblick auf die Forschungsfragen mit Hilfe der Abgrenzungskriterien bewertet. Aus den Forschungsfragen haben sich elf Bewertungskriterien ergeben, die erfüllt werden müssen. Diese Bewertungskriterien sind mit der Zuordnung des Erfüllungsgrades der einzelnen Ansätze in der Abbildung 3.9 dargestellt. Abschließend wird die ermittelte Forschungslücke herausgestellt sowie die für die Schließung dieser Forschungslücke erarbeitete Handlungsmaßnahme.

Für die **Identifizierung** der auftretenden Änderung müssen zunächst **Muster** identifiziert werden. Mit Hilfe dieser Muster soll der Anwender befähigt werden, geeignete Maßnahmen zur Umsetzung der Änderung zu ergreifen. In dem Ansatz nach LI UND CHEN [LC10] findet die Erstellung eines geeigneten Vorgehens dadurch statt, dass die DDM umstrukturiert wird. Durch die Umstrukturierung wird es möglich Muster zwischen den Relationen der geänderten Elemente zu erkennen. Somit kann im Vorfeld ein bestimmtes Vorgehen gewählt werden, um die Änderung durchzuführen. Dahingegen besteht der Ansatz der untersuchten Forschungslücke darin, dass es ebenfalls möglich sein sollte, diese Muster in dem Systemmodell zu identifizieren. Das Systemmodell soll mit Hilfe von SysML implementiert werden.

Darauf aufbauend soll die **Betrachtung** des Systemmodells hinsichtlich der Entwicklungsartefakte des **RFLPV<sup>2</sup>-Ansatzes** durchgeführt werden. Die Relevanz besteht darin, dass in dem Ansatz alle wesentlichen Entwicklungsartefakte abgebildet werden, welche notwendig für die Entwicklung cyber-physischer Systeme sind. Für die Betrachtung der Auswirkung hinsichtlich der jeweiligen Artefakte ist kein Ansatz zu identifizieren. Die Forschungsfragen fokussieren explizit, dass Änderungen ausgehend von Testszenarien und Testfällen auftreten. Diese können sowohl in der Planung als auch in der Durchführung dieser Artefakte auftreten. Für die Betrachtung von Änderungen ausgehend von der

Planung oder Durchführung der Testfälle und Testszenarien sind ebenfalls keine Ansätze vorhanden.

Kernthema ● Teilweise behandelt ◐ Randthema ○ Nicht behandelt –		ELLSEL ET AL. [ELL22]	LI UND CHEN [LC10]	REDDI UND MOON [RM09]	INTANA ET AL. [ILS23]	LIN ET AL. [LPS+15]	YANG UND DUAN [YD12]	DEUBEL ET AL. [DCK+07]	EIGENE HANDLUNGSMAßNAHME
Abgrenzungskriterien	Identifikation von Mustern	–	●	–	–	–	–	–	●
	Betrachtung des RFLPV <sup>2</sup> -Ansatzes	–	–	–	–	–	–	–	●
	Analyse testinduzierter Änderungen	–	–	–	–	–	–	–	●
	Schnittstelle mit SysML	◐	–	–	–	●	–	–	◐
	Analyse der Relationen der Artefakte	●	◐	●	○	●	●	●	◐
	Analyse direkter und indirekter Auswirkungen	–	–	●	–	●	●	●	●
	Bewertung der Auswirkungen	●	●	–	●	–	●	●	●
	Kategorisierung des Änderungsumfangs	●	–	–	–	–	–	●	●
	Reaktive Handlungsmaßnahme	●	●	●	●	●	●	●	●
	Verwendung des Systemmodells	●	–	–	–	●	–	–	●
	Präventive Maßnahmen für die Vermeidung von Änderungen	–	–	–	–	–	–	–	●

Abbildung 3.9: Abgrenzungsmatrix der Handlungsmaßnahme

Ein weiteres Kriterium ist die **Einbindung von SysML**, welche aufgrund der Betrachtung aus dem Kontext des MBSE notwendig ist. Für die direkte Anbindung oder Verwendung von SysML sind zwei relevante Ansätze zu identifizieren. In dem Ansatz nach ELLSEL ET AL. [ESS22] werden mit Hilfe der Modellierung jedem Artefakt mögliche Auswirkungen bei einer Änderung zugewiesen. Diese dient der Bewertung des Aufwands der Änderung. In dem Ansatz nach LIN ET AL. [LPS+15] werden die einzelnen Artefakte, welche in

SysML modelliert werden, hinsichtlich der Relationen zueinander untersucht. Somit bildet das Systemmodell die Grundlage des Ansatzes.

Neben der Umsetzung mit Hilfe von SysML ist es notwendig die **Relationen** der einzelnen Entwicklungsartefakte untereinander zu **analysieren**. Für die Auswirkungsanalyse hinsichtlich der Relationen sind dabei unterschiedliche Ansätze anwendbar. Der Ansatz nach REDDI UND MOON [RM09] setzt voraus, dass die Abhängigkeiten der einzelnen Artefakte durch den Anwender in der Design Phase eingelesen werden. Der Ansatz nach YANG UND DUAN [YD12] verarbeitet die Relationen über einen Netzgraphen. Für die Forschungsfrage ist jedoch relevant die Relationen zu untersuchen, welche aus Gründen der Rückverfolgbarkeit als auch für die Generierung des jeweiligen Artefakts notwendig sind. Daher bietet es sich an, durch die Analyse mittels mehrerer Modellierungsmethoden alle Relationen zu erfassen.

Neben der Betrachtung der Relationen ist es ebenfalls von Bedeutung, welche **Auswirkungen direkt** entstehen und welche **indirekt**. Dies ist notwendig, um Aussagen über das Propagationsverhalten treffen zu können. In den Ansätzen nach DEUBEL ET AL. [DCK+07] und REDDI UND MOON [RM09] werden die indirekten Auswirkungen dadurch bestimmt, dass jedem Artefakt Wahrscheinlichkeiten als auch Relationen zu anderen Artefakten zugeordnet werden. Somit kann über die Wahrscheinlichkeit jeder Änderung bestimmt werden, welche Artefakte noch betroffen sein können. In dem Ansatz nach YANG UND DUAN [YD12] wird die Art der Relation durch die Funktion bestimmt, welche den Parameter beinhaltet. In der entwickelten Handlungsmaßnahme wird jedoch zwischen direkter Auswirkung und indirekter Auswirkung unterschieden. Unterschieden wird dabei in der Form, ob sich ein Artefakt auf dem direkten Änderungspfad befindet oder nicht.

Zusätzlich sind die **Auswirkungen** nicht nur zu identifizieren, sondern auch zu **bewerten**. Eine Bewertung hinsichtlich der Kritikalität der Auswirkungen wird in dem Ansatz nach ELLSEL ET AL. [ESS22] durchgeführt. Für die Bewertung wird ein zugeordneter Wert in Arbeitsstunden dem Objekt hinterlegt. Durch die Relation zu dem änderungsauslösenden Artefakt wird der jeweilige Arbeitsumfang bestimmt. Im Gegensatz zu den Ansatz nach ELLSEL ET AL. wird nach LI UND CHEN [LC10] eine Bewertung je nach ausgewähltem Muster übergeben. Der Ansatz nach YANG UND DUAN [YD12] verwendet hingegen eine Berechnung über die eingebundenen und gewichteten Knoten, welche von der Änderung betroffen sind. Im Fall der entwickelten Handlungsmaßnahme wird ein Verfahren angewendet, welches dem Ansatz nach DEUBEL ET AL. [DCK+07] ähnelt. Für die Bewertung der Kritikalität werden dabei Kennzahlen je nach Ausprägung des Musters oder der Art des Artefaktes übergeben. Aus diesen Kennzahlen lässt sich eine Bewertung der Kritikalität der Änderung ableiten. Diese Einordnung der Zahl erfolgt über ein weiteres Kriterium. Durch das Kategorisierungs-Kriterium lässt sich abschließend das Handlungspotential ableiten.



## 4 Analyse verwandter Ansätze

In dem vierten Kapitel werden die aus der Literaturrecherche ermittelten Ansätze für die Modellierung cyberphysischer Systeme vorgestellt. Zu Beginn wird das Vorgehen der Literaturrecherche sowie die Ergebnisse der einzelnen Suchstrings dargelegt. Darauf aufbauend werden die theoretischen Ansätze vorgestellt und die dafür notwendigen Entwicklungsartefakte in dem V-Modell verortet. Zusätzlich werden Modellierungsansätze aus Fallstudien der Industrie genauer analysiert.

### 4.1 Literaturrecherche

Die Recherche verwandter Arbeiten im Zusammenhang mit der Identifikation und Bewertung von Auswirkungsmustern ist im Rahmen dieser Forschungsfrage mit Hilfe mehrerer Suchstrings durchzuführen. Für die Suchstrings wird die Recherche mit vier verschiedenen Suchmaschinen durchgeführt. Diese umfassen dabei Web of Science, Scopus, IEEEExplore und Google Scholar. Die verwendeten Suchstrings und die Anzahl der Suchergebnisse sind in der Abbildung 4.1 dargestellt.

Für das Ausarbeiten der relevanten Literatur ist das Vorgehen an den PRISMA-Ansatz [PMB+21] angelehnt und besteht aus einem dreistufigen Verfahren. Zu Beginn sind die Suchergebnisse zusammenzuführen und Duplikate zu entfernen. Anschließend werden die Suchergebnisse anhand ihrer Titel sortiert und in die sogenannte C-Liste eingeteilt. Die Differenzierung in der Liste erfolgt zwischen „aussortierte Quellen“ und „weiter zu untersuchende Quellen“. Im darauffolgenden Schritt wird die B-Liste erstellt. Hierzu werden die näher zu betrachtenden Titel anhand des Abstracts beurteilt. Quellen, die ein falsches Themengebiet betrachten, werden aussortiert und die übrig gebliebenen weiter untersucht. Im letzten Schritt der Ausarbeitung der A-Liste werden die Quellen genauer untersucht. Es werden die verwendeten Abbildungen betrachtet sowie die Zusammenfassung der Ergebnisse. Die Quellen, welche nicht aussortiert werden, sind im Anschluss vollständig zu untersuchen und werden gegebenenfalls mit aufgenommen. Anhand dieser aufgenommenen Titel wird die weiterführende Literatur in Form von Primärquellen und Verweise identifiziert.

Die Literaturrecherche mit Hilfe des ersten Suchstrings hat ergeben, dass keine artverwandten Ansätze vorhanden sind. Somit hat sich gezeigt, dass Handlungsbedarf bezüglich des Schließens der Forschungslücke besteht. Aus dieser Erkenntnis ergibt sich das Verfahren, dass über weitere Suchstrings allgemeine theoretische Ansätze für die Modellierung von cyberphysischen Systemen zu identifizieren sind. Zusätzlich sind die theoretischen Ansätze durch die Grundlagenliteratur aus dem Bereich MBSE und SysML zu ergänzen. Für die Betrachtung der Praxis sind Ansätze zu identifizieren, in denen durch Fallstudien cyberphysische Systeme im Industriekontext modelliert worden sind. Die Erweiterung der relevanten Ansätze durch Betrachtung der Primärliteratur ist mit Hilfe der Literaturreviews nach BRAHMA UND WYNN [BW23] und nach GRÄBLER ET AL. [GWT22] durchzuführen. Die in der Theorie und in der Praxis als relevant identifizierten Ansätze

werden im Folgenden genauer beleuchtet. Die Analyse bezieht sich im Kern auf die verwendeten Entwicklungsartefakte und deren Relationen zueinander.

	IEEE Xplore	Web of Knowledge	Scopus	Google Scholar	Ohne Duplikate	C-Liste	B-Liste	A-Liste	Verwendet
( "test*" OR "Use Case" OR "requirement*" OR "Verification" OR "Validation" ) AND ( "model*" OR "systems engineering" OR "Product development" ) AND ( "impact analysis" )	207	205	798	855	883	55	22	13	3
"pattern-based systems engineering" and "test"	-	-	-	71	71			7	3
"model-based systems engineering" AND "case study" AND ("Use Case*" OR "Requirement*") AND ("Verification" OR "Validation" OR "Test*")	44	22	80	831	907	39	25	11	8

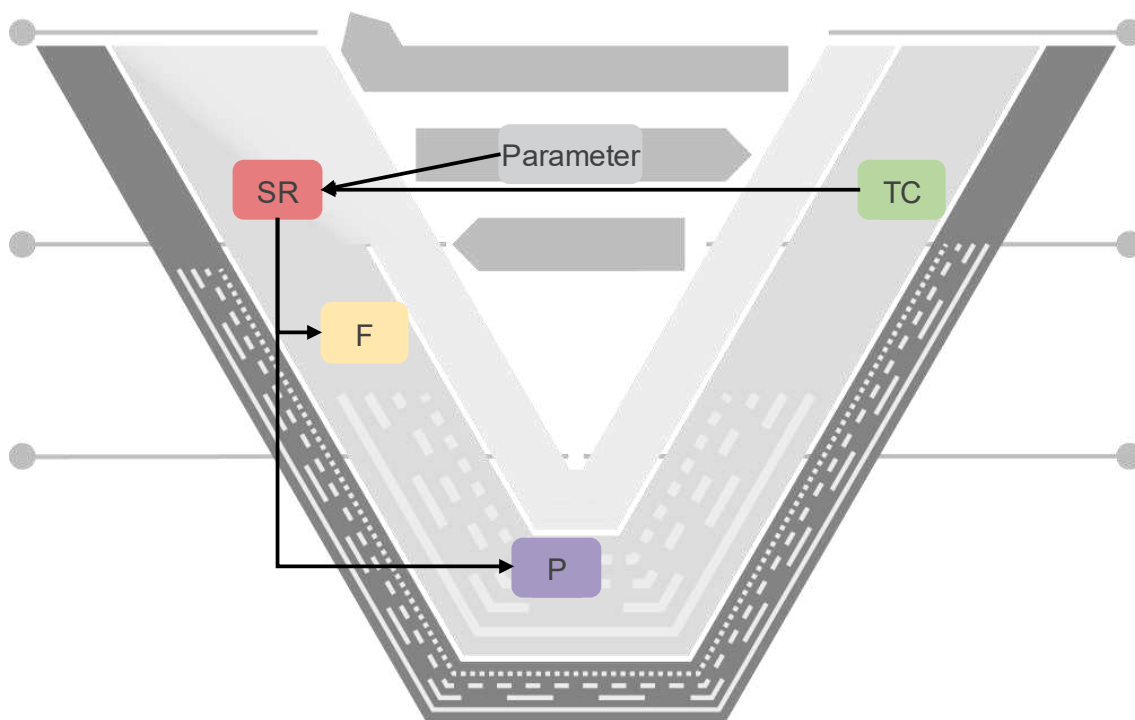
Abbildung 4.1: Ergebnisse der Literaturrecherche gemäß dem gewählten Vorgehen

## 4.2 Modellierungsmethoden für cyberphysische Systeme

Für die Modellierung von cyberphysischen Systemen sind verschiedene Methoden vorhanden. Die Abhängigkeiten der einzelnen Entwicklungsartefakte sind in unterschiedlicher Form ausgeprägt. Die unterschiedlichen Darlegungsformen beruhen zum einen auf unterschiedlichen Modellierungssprachen, welche verwendet werden. Zum anderen werden Quellen betrachtet, welche keine Modellierungsmethode zu Grunde legen und sich somit nur auf die Entwicklungsartefakte und deren Relationen beschränken. Für die Vergleichbarkeit der unterschiedlichen Ansätze sind die Artefakte auf dem V-Model verortet.

### 4.2.1 OOSEM

Die Object Orientated System Engineering Method (OOSEM) ist eine objektorientierte Methode zur Modellierung technischer Systeme. Die Methode wurde im Jahr 1998 durch einen Zusammenschluss von Lockheed Martin und dem Systems and Software Consortium (SSCI) entwickelt. Im Rahmen der Forschungsfrage wird sich auf die Beschreibung nach FRIEDENTHAL [Fri14] bezogen. Die verwendeten Entwicklungsartefakte und die Relationen sind in Anlehnung an die Modellierungsmethode in der Abbildung 4.2 auf dem V-Modell verortet [Fri14]



SR = Systemanforderung; F = Funktion; P = Physisches Element; TC = Testfall

Abbildung 4.2: Darstellung der elementaren Entwicklungsartefakte [Fri14] der OOSEM-Methode auf dem V-Modell verortet

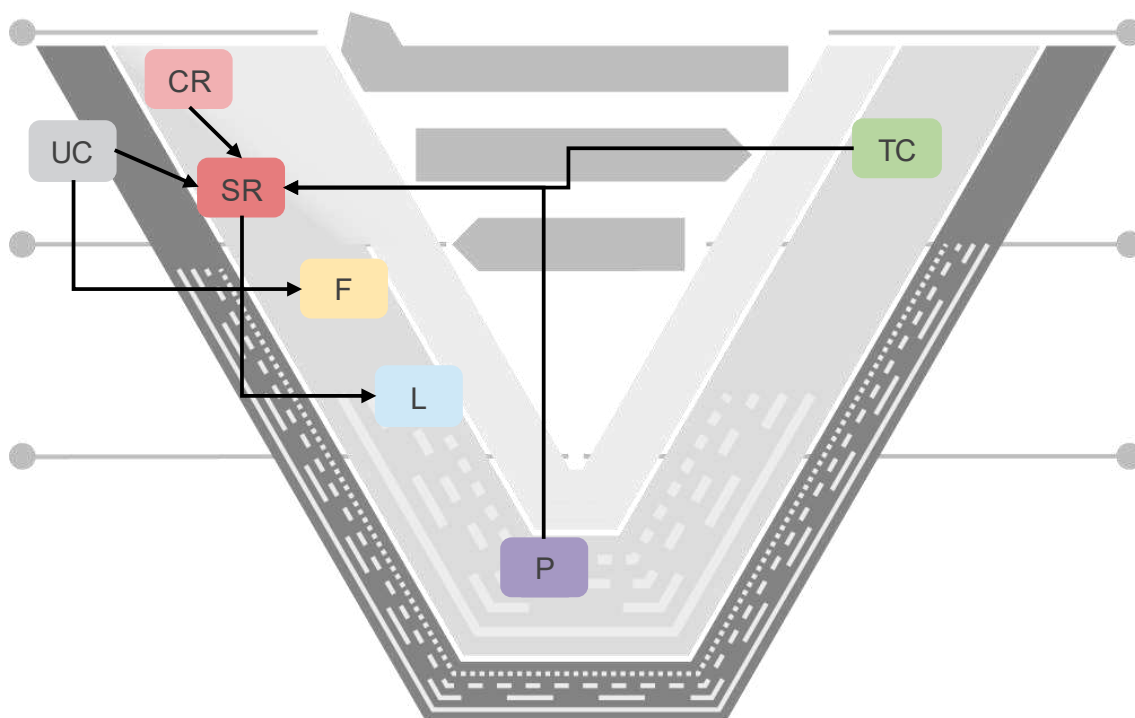
Das Vorgehen ist so aufgebaut, dass zu Beginn eine Struktur für die Diagramme angelegt werden muss. Anschließend sind die Stakeholderbedürfnisse zu analysieren. Zusätzlich müssen die Anwendungsfälle und die Effektivitätsmaßstäbe (MoE) für die Validierung der Stakeholderanforderungen erstellt werden. Mit Hilfe der Anwendungsfälle können die Systemanforderungen, die das Systemverhalten beschreiben, abgeleitet werden. Zusätzlich sind die Schnittstellen zu der Umgebung zu definieren. Aufbauend darauf kann das System zerlegt und die Verbindungen der einzelnen Systemelemente definiert werden. Genauer gesagt die Stoff-, Energie- und Signalflüsse. Die Analyse des Systems kann mit dem Parameterdiagramm und angelegten Nebenbedingungen durchgeführt werden. Anschließend muss die Rückverfolgbarkeit zwischen den Entwicklungsartefakten hergestellt werden. Dabei muss das Systemdesign die Anforderungen erfüllen. Abschließend sind die Testfälle zu definieren, welche die Anforderungen verifizieren. [Fri14]



Die Entwicklungsartefakte der OOSEM sind so aufgebaut, dass die Anforderungen in einer hierarchischen Struktur modelliert werden. Die Anforderungen auf einer niedrigeren Systemebene werden mit der „derived“-Relation verbunden. Ebenfalls werden die Anforderungen mit den Funktionen über die „refine“-Relation und das System in Form von Blöcken über die „satisfy“-Relation verbunden. Zusätzlich werden die Anforderungen mit Aktivitäten erfüllt, welche mit den involvierten Systemelementen über die „allocate“-Relation verbunden sind. [Fri14]

#### 4.2.2 SysMOD

Das Vorgehen gemäß SysMOD ist dafür vorgesehen, die Modellierung von SysML-Modellen zu vereinfachen. Es wird kein genaues Vorgehen vorgegeben, jedoch eine Vorgehensweise, die aus „Best Practises“ gewonnen wird. Die Vorgehensweise ist praxisorientiert und an die Bedürfnisse des Anwenders anpassbar. Die identifizierten Entwicklungsartefakte sind in der Abbildung 4.3 auf dem V-Modell verortet. [Wei07]



CR = Stakeholderanforderung; SR = Systemanforderung; L = logisches Element;  
F = Funktion; P = Physisches Element; TC = Testfall; UC = Anwendungsfall

Abbildung 4.3: Darstellung der elementaren Entwicklungsartefakte der SysMOD-Methode auf dem V-Modell verortet

Zu Beginn der Modellierung wird das System gegenüber der Umwelt abgegrenzt. Das bedeutet es werden die Systemgrenze, die Ziele und alle Randbedingungen definiert. Der nächste Schritt ist die Definition von Anforderungen, hierzu werden die Stakeholder identifiziert und abgebildet. Als nächstes können die Stakeholderanforderungen analysiert

und in Systemanforderungen überführt werden. Darauf aufbauend werden die Schnittstellen des Systems zu äußeren Objekten definiert. Danach werden die Anwendungsfälle aufgestellt und modelliert, um die erhobenen Anforderungen zu verfeinern. Mit den Anwendungsfällen kann das Systemverhalten in Aktivitätsdiagrammen modelliert werden. Hieraus lassen sich Funktionen herleiten. Mit den gewonnenen Informationen kann die Systemstruktur modelliert werden. Die daraus abzuleitenden Systemelemente können die Systemanforderungen erfüllen. Die Anforderungen werden mit Hilfe von Testfällen verifiziert. [Wei07]

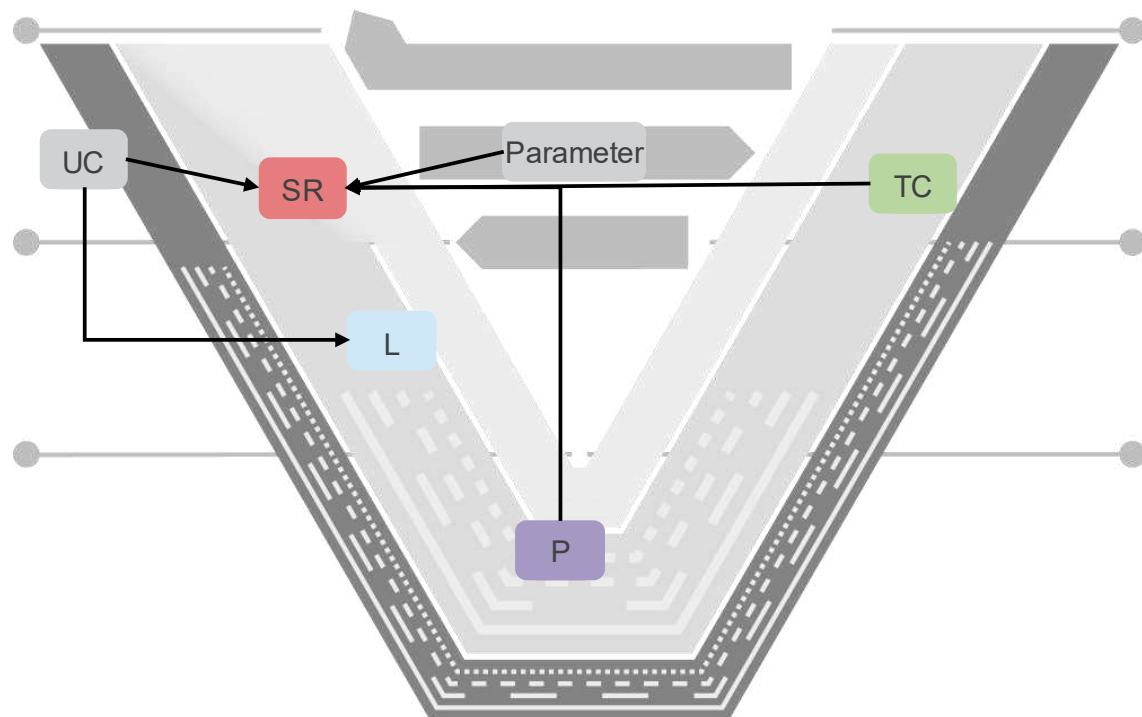
Die Entwicklungsartefakte für die SysMOD-Methode sind so aufgebaut, dass grundlegend die Anwendungsfälle verwendet werden, um die Systemanforderungen zu verfeinern. Für die Anforderungen sind optionale Kategorisierungen vorgesehen, welche eine Unterscheidung der Anforderungen nach den beschriebenen Merkmalen ermöglichen. Bei den Anwendungsfällen werden zusätzlich Relationen untereinander modelliert. Dies umfasst das Ausschließen gewisser Funktionen oder das Einbinden je nach Ausprägung des Anwendungsfalls. Die Anforderungen detaillieren die Systemstruktur, welche mit Hilfe des internen Blockdiagramms modelliert wird. Es ist eine „trace“-Verbindung vorzusehen. Die aus der Systemstruktur abgeleiteten Elemente wiederum erfüllen die zugehörigen Anforderungen. Zusätzlich wird jeder Anforderung ein Testfall zugeordnet mit Hilfe der „verify“-Relation. [Wei07]

#### **4.2.3 Harmony SE**

Die Harmony SE Methode nach HOFFMANN [Hof13-ol] beschreibt einen Modellierungsansatz für das Systems Engineering. Der Ansatz ist dabei in zwei Teilprozesse aufgeteilt. Der erste Teil ist dabei auf dem linken Schenkel des V-Modells zu verorten und beschreibt den Harmony for Systems Engineering Prozess. Der zweite Teilprozess beschreibt den Harmony for Embedded Real Time Development und ist auf dem rechten Schenkel des V-Modells zu verorten. Die Entwicklungsartefakte beider Ansätze sind in der Abbildung 4.4 dargestellt. [Hof13-ol]

Der Aufbau sieht dabei so aus, dass der Harmony for Systems Engineering Teilprozess den Design Flow beschreibt. Es handelt sich um ein Top-Down Verfahren. Für das Vorgehen gemäß Harmony SE muss bei einer Änderungsanfrage der Design Prozess mit der Anforderungsanalyse gestartet werden. Der Teilprozess umfasst darauffolgend die Systemfunktionsanalyse und die Design Synthese. Das Vorgehen in dem ersten Teilprozess ist als iterativer Prozess vorgesehen und muss mit Hilfe von Anwendungsfällen umgesetzt werden. Der zweite Teilprozess für das Embedded Real Time Development beinhaltet die Implementierung sowie Tests. Es wird ein Bottom-Up-Verfahren gewählt. Die Teilsysteme werden sukzessiv getestet und integriert. Die einzelnen Module und Subsysteme werden nacheinander verifiziert. [Hof13-ol]

Der Aufbau der Entwicklungsartefakte sieht dabei so aus, dass aus den Anwendungsfällen die Anforderungen definiert und über die „trace“-Relation zugeordnet werden. Ebenfalls wird die logische Struktur aus den Anwendungsfällen abgeleitet. Für die Erfüllung der Anforderungen werden die Effektivitätsmaßstäbe (MoE's) verwendet, wobei hier relevante Systemparameter hinterlegt sind und mit einer „Satisfy“-Relation verbunden werden. Zusätzlich erfüllt das System of Interest in Form von Blöcken die Anforderungen und wird ebenfalls über die „Satisfy“-Relation verbunden. Für die Verifikation der Anforderungen werden Testfälle durchgeführt. [Hof13-ol]



SR = Systemanforderung; L = logisches Element; P = Physisches Element;  
TC = Testfall; UC = Anwendungsfall

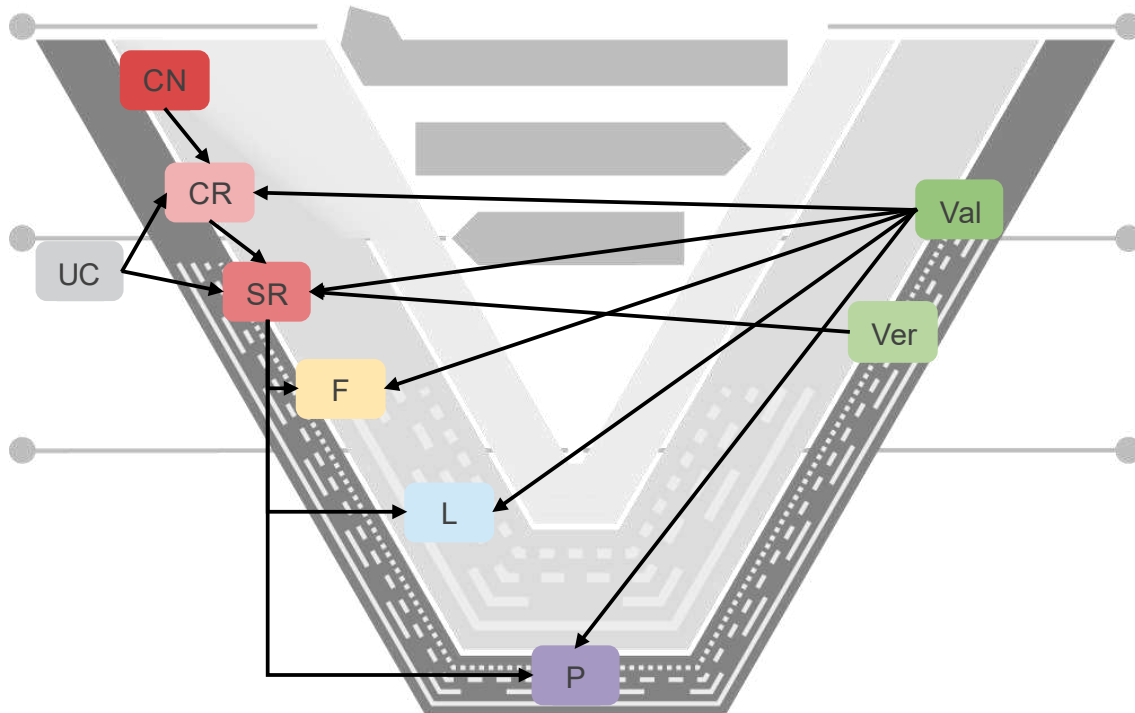
Abbildung 4.4: Darstellung der elementaren Entwicklungsartefakte der Harmony SE-Methode auf dem V-Modell verortet

#### 4.2.4 ISO 15288

Die ISO 15288 [15288:2023(E)] ist die Norm für die System-Lebenszyklus-Prozesse. Somit beschreibt die Norm keine Methode, sondern definiert stattdessen, wie die einzelnen technischen Prozesse in der Entwicklung der technischen Produkte durchgeführt werden müssen. Zusätzlich werden Informationen über die zu entwickelnden Systemartefakte gegeben. Aus der Beschreibung werden die grundlegenden Entwicklungsartefakte und deren Relation hergeleitet und in der Abbildung 4.5 dargestellt. [15288:2023(E)]

Für die Entwicklung eines technischen Systems müssen neun technische Prozesse durchgeführt werden. Diese sind direkt in dem V-Modell zu verorten. Darüber hinaus werden alle Prozesse dargestellt, welche die restlichen Lebenszyklusabschnitte beschreiben. Für

die Entwicklung technischer Systeme muss mit der Stakeholderbedürfnis- und -anforderungsanalyse begonnen werden. Aus den Stakeholderbedürfnissen müssen erst die Stakeholderanforderungen und dann die Systemanforderungen abgeleitet werden, um daraus die Systemarchitektur zu bestimmen. Darauf aufbauend folgt der Designdefinitionsprozess und der Systemanalyseprozess. Die entwickelten Systemelemente werden in dem Implementierungsprozess umgesetzt und im Integrationsprozess zusammengeführt. Abschließend können in dem Verifikations- und Validierungsprozess die entwickelten Artefakte gegenüber den Anforderungen geprüft werden. [15288:2023(E)]



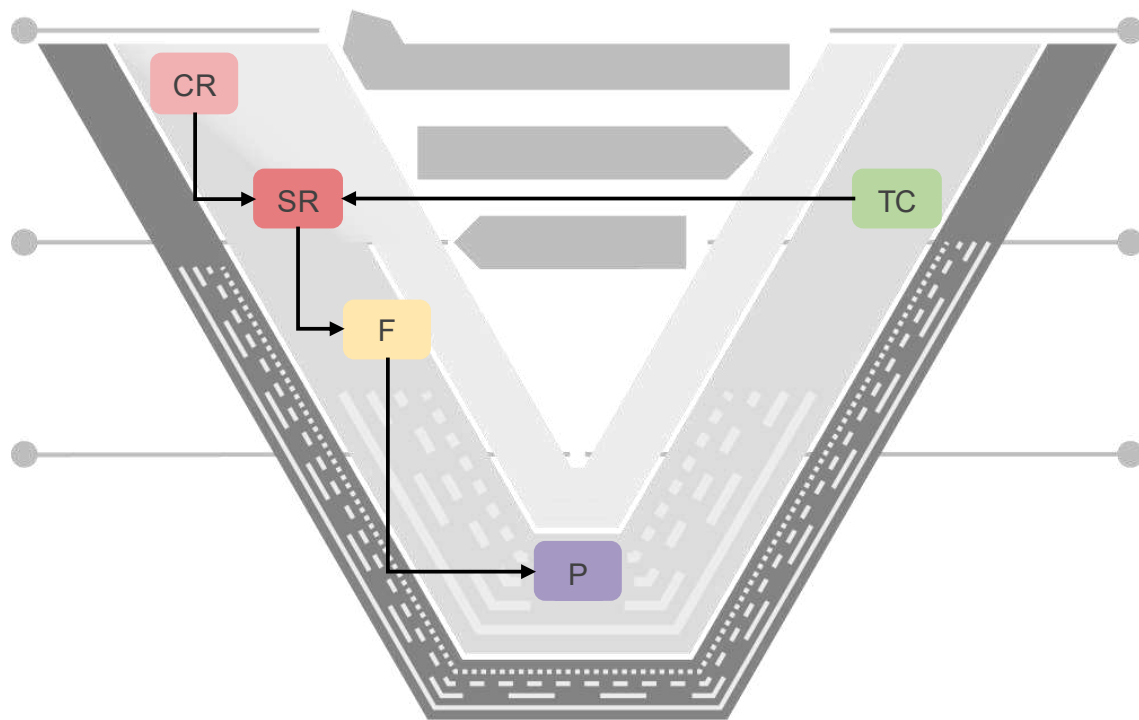
CR = Stakeholderanforderung; SR = Systemanforderung; F = Funktion;  
 L = logisches Element; P = Physisches Element; Ver = Verifikation; Val = Validierung;  
 UC = Anwendungsfall; CN = Stakeholderbedürfnis

Abbildung 4.5: Darstellung der elementaren Entwicklungsartefakte gemäß der ISO 15288 auf dem V-Modell verortet

Die aus der Norm ableitbaren Systemartefakte sind dabei gemäß der Verortung auf dem V-Modell so anzuordnen, dass zu Beginn die Anwendungsfälle und Stakeholderbedürfnisse zu modellieren sind. Diese werden verknüpft mit den Stakeholderanforderungen. Die Stakeholderanforderungen werden über die „derived“-Relation aus den Systemanforderungen abgeleitet. Den Anforderungen werden die Funktionen, die logischen Elemente und die physischen Elemente über die „allocate“-Relation zugeordnet. Die Systemartefakte aus der Verifikation und Validierung sind mit der „trace“-Relation den Funktionen, logischen und physischen Elementen zugeordnet. Zusätzlich wird die „verify“-Relation zu den zugehörigen Stakeholder- und Systemanforderungen gezogen. [15288:2023(E)]

### 4.2.5 Consens

Die CONSENS-Methode ist für die Reduzierung der Komplexität der Entwicklung technischer Systeme ausgelegt. Das System wird über mehrere Partialmodelle erstellt. Diese Partialmodelle sind untereinander vernetzt, so dass ein kohärentes Gesamtsystem entsteht. Die aus den Partialmodellen ableitbaren Entwicklungsartefakte sind in der Abbildung 4.6 auf dem V-Modell verortet.[GGT13]



CR = Stakeholderanforderung; SR = Systemanforderung; F = Funktion;  
P = Physisches Element; TC = Testfall

Abbildung 4.6: Darstellung der elementaren Entwicklungsartefakte der Consens-Methode auf dem V-Modell verortet

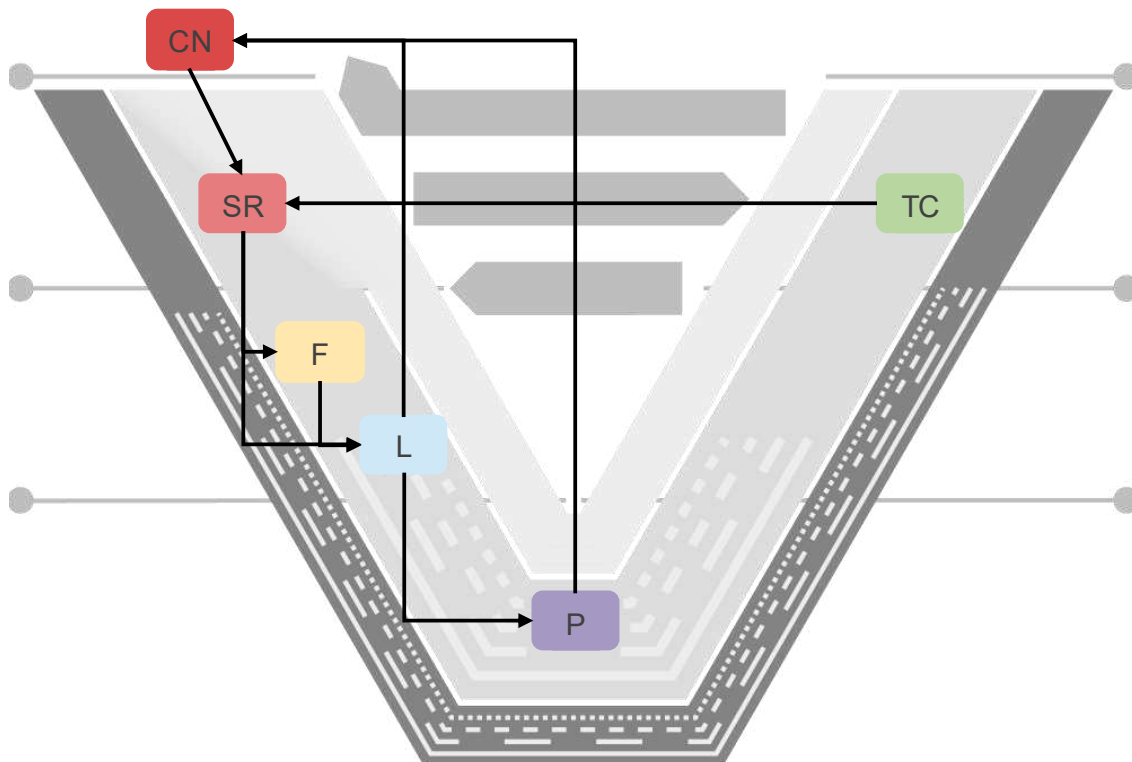
Das Vorgehen besteht darin, die einzelnen Partialmodelle des Systems zu erzeugen. Partialmodelle bilden alle Einflüsse ab, die auf das System einwirken. So sind z. B. Wetteneinflüsse im Partialmodell der Umwelt berücksichtigt. Ein weiteres Modell bildet das Anwendungsszenario ab, welches vergleichbar mit den Anwendungsfällen die Verwendung des Systems darstellt. Zusätzlich werden die Anforderungen, Funktionen, die Systemstruktur und die Gestalt in separaten Modellen berücksichtigt. Abschließend ist ein Modell für das Systemverhalten zu erstellen. Das Systemverhalten wird unter Berücksichtigung der äußeren Einflüsse abgebildet. [GGT13]

Der Aufbau der Entwicklungsartefakte für die CONSENS-Methode muss in die SysML Logik überführt werden, da für die Methode eine andere Modellierungssprache vorgesehen ist. Für die Schaffung der Vergleichbarkeit sind alle Ansätze in eine einheitliche Modellierungssprache zu überführen, wobei in Anlehnung an SysML die Artefakte darge-

stellt werden. Für die CONSENS-Methode werden zu Beginn die Stakeholderanforderungen in Form einer User Story aufgenommen bzw. aus dieser hergeleitet. Aus den Stakeholderanforderungen können die ihnen zuzuordnenden Systemanforderungen abgeleitet werden. Aus den Systemanforderungen werden die Funktionen abgeleitet, aus denen wiederum die Systemelemente abgeleitet und definiert werden. Den Anforderungen wird ein Testfall zugeordnet, welcher mit der „verify“-Relation zu modellieren ist. [GGT13]

#### 4.2.6 Arcadia

Die ARCADIA-Methode steht kurz für Architecture Analysis and Design Integrated Approach. Die Methode wurde von der Thales-Gruppe im Jahr 2000 entwickelt, als diese sich von einem Zulieferer von Equipment zu einem Zulieferer komplexer technischer Produkte weiterentwickelt hat. Im Wesentlichen beruht das Verfahren auf einer Funktionsanalyse. Die Abbildung 4.7 zeigt die Entwicklungsartefakte verortet auf dem V-Modell. [Roq16; BGO+23]



CR = Stakeholderanforderung SR = Systemanforderung; L = logisches Element;  
P = Physisches Element; TC = Testfall; F = Funktion

Abbildung 4.7: Darstellung der elementaren Entwicklungsartefakte der Arcadia-Methode auf dem V-Modell verortet

Der Aufbau der Methode teilt sich dabei in vier Ebenen auf, welche auf unterschiedliche Art und Weise erarbeitet werden können. Es kann sowohl top-down, bottom-up oder auch iterativ vorgegangen werden. Die erste Ebene ist dabei die operationale Ebene, in der eine operative Analyse durchzuführen ist. Die nächste Ebene bildet die Systembedürfnis-

Ebene, in der eine Systemanalyse durchzuführen ist. Aus den Ergebnissen der Systemanalyse und der logischen Architektur kann das System modelliert werden. Zusätzlich wird die physische Architektur umgesetzt. [Roq16; BGO+23]

Die Arcadia-Methode basiert auf einem anderen Werkzeug als SysML, aus diesen Gründen ist die Modellierung in die SysML-Logik übersetzt. Die Systemartefakte weisen dabei die Abhängigkeiten auf, dass die Stakeholderbedürfnisse die Anforderungen an das System verfeinern. Die Anforderungen führen zu Spezifizierungen der logischen und physischen Architektur. Zusätzlich weisen die Funktionen eine „allocate“-Relation zu den einzelnen Systemelementen auf. Die entstehende Systemarchitektur besitzt eine „satisfy“-Relation zu den modellierten Stakeholderbedürfnissen. Die Verifikation und Validierung wird anhand von Testfällen durchgeführt. [Roq16; BGO+23]

#### 4.2.7 JPL State Analysis

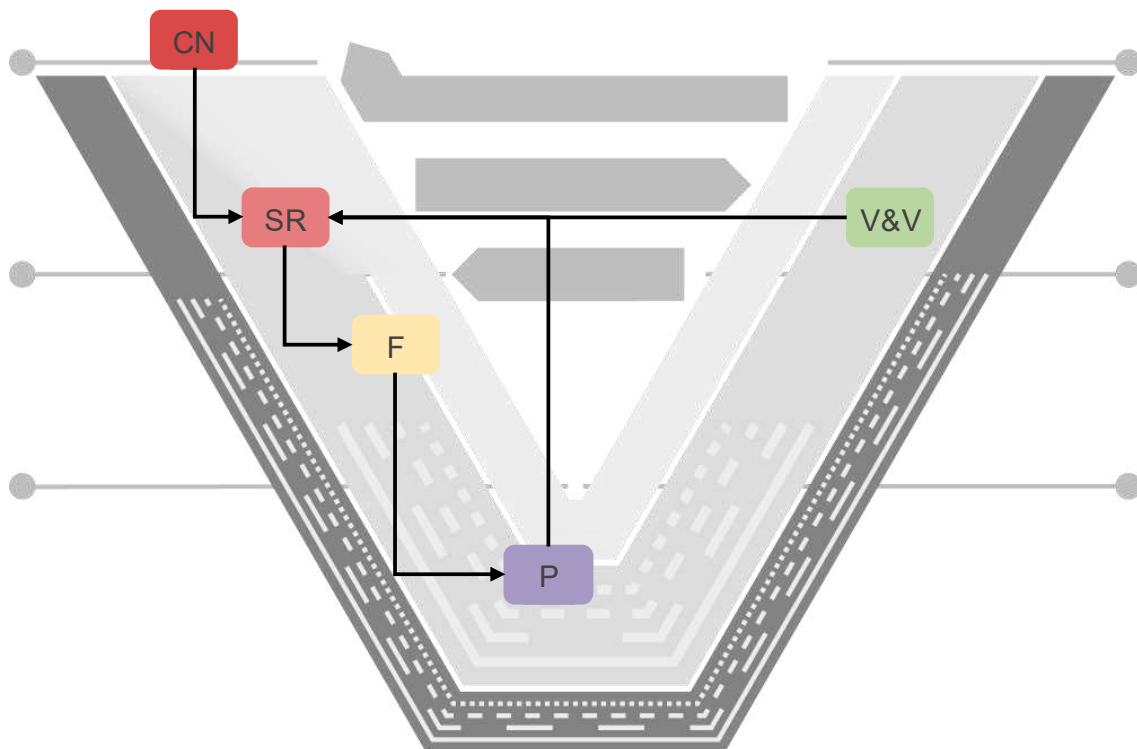
Die JPL State Analysis ist ein von dem Jet Propulsion Laboratory entwickeltes technisches System zur Darstellung von Weltraummissionen. Damit soll die Dynamik der Systeme erfasst werden, die mit herkömmlichen Methoden schwer beschreibbar ist. Aufgrund der hohen Kosten und potenziellen Personalfährdungen während der Einsätze stehen die Aufgaben der Validierung und Verifikation hier besonders im Fokus. Die verwendeten Entwicklungsartefakte sind in Abbildung 4.8 dargestellt. [KWS+07]

Das Vorgehen ist dabei so aufgebaut, dass zu Beginn die Bedürfnisse der Stakeholder analysiert werden. Ebenfalls werden die Objekte der obersten Systemebene definiert. Im nächsten Schritt steht die Identifizierung der Zustandsgrößen. Diese müssen reguliert werden, um die Anforderungen zu erfüllen. Für die Zustandsgrößen müssen Zustandsmodelle aufgestellt werden, wobei in dieser Phase weitere Zustandsgrößen identifiziert werden können. Der Einfluss dieser Zustandsgrößen auf das System ist von großer Bedeutung. Für die Messung der Zustandsgrößen müssen die nötigen Messmethoden hergeleitet werden, wofür ebenfalls Modelle aufzustellen sind. Zuletzt müssen die Befehle identifiziert und modelliert werden, welche für die Regelung nötig sind. Bei allen Schritten ist darauf zu achten, alle Zustandsgrößen zu identifizieren, welche das System beeinflussen können. [KWS+07]

Die Entwicklungsartefakte der State Analysis sind bei der Projektion auf das SysML-Modell so aufgebaut, dass die Stakeholderbedürfnisse zu den Systemanforderungen transformiert werden. Die Systemanforderungen spezifizieren die funktionale und physische Systemarchitektur und weisen dabei die „specify“-Relation auf. Zusätzlich werden die physischen Systemelemente mit den Funktionen verbunden. Die physischen Elemente



werden zusätzlich über die „allocate“-Relation den Anforderungen zugeordnet. Die Testfälle aus der Verifikation und Validierung werden mit einer „verify“-Relation verbunden. [KWS+07]



CR = Stakeholderanforderung; SR = Systemanforderung; F = Funktion;  
P = Physisches Element; V&V = Verifikation und Validierung

Abbildung 4.8: Darstellung der elementaren Entwicklungsartefakte der JPL State Analysis-Methode auf dem V-Modell verortet

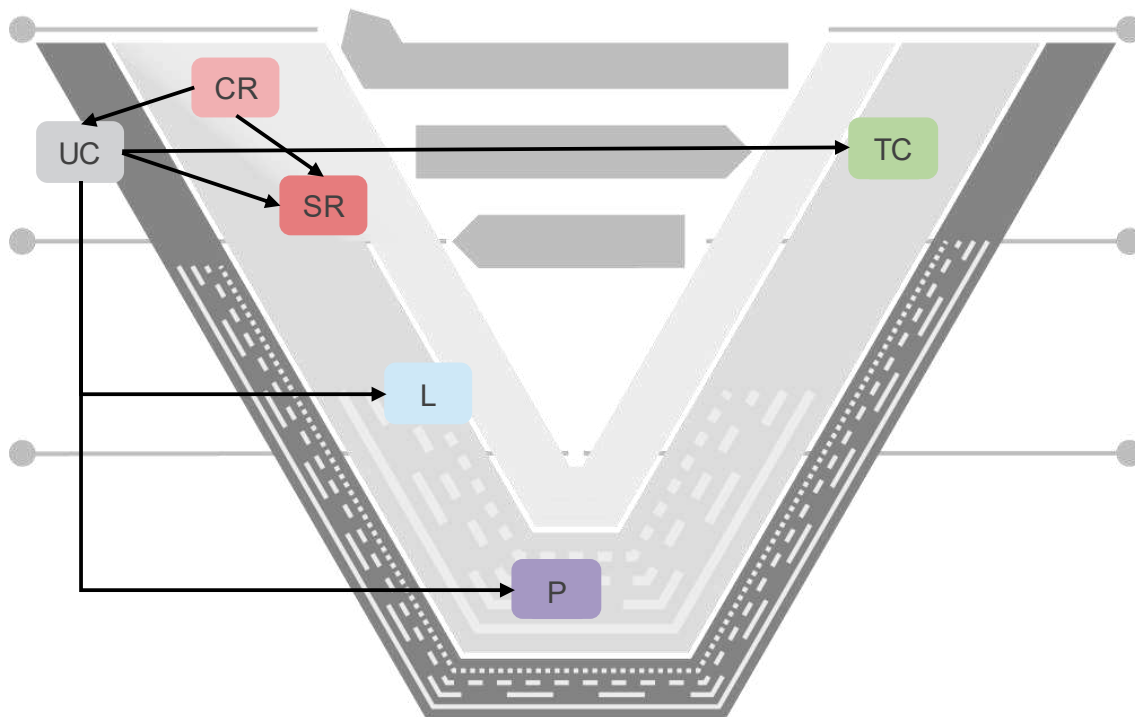
#### 4.2.8 Ruph SE

Der Rational Unified Process (Ruph SE) ist ein Rahmenwerk für die Entwicklung von Software, der genutzt wird, um den Entwicklungsprozess formalisiert zu beschreiben. Der Entwicklungsprozess ist dabei iterativ aufgebaut und umfasst zudem die Verifikation und mögliche Änderungen. Es besteht dadurch die Möglichkeit, den Prozess an die Anwenderbedürfnisse anzupassen. Aus der Methode ergeben sich die Entwicklungsartefakte, die in der Abbildung 4.9 auf dem V-Modell verortet sind. [Kru03]

Das Vorgehen beruht auf den Anwendungsfällen, welche für das Produkt vorgesehen sind. Die Anwendungsfälle werden dabei zusammen mit den Stakeholderanforderungen aufgenommen. Auf Basis der Anwendungsfälle und den Stakeholderanforderungen können die Anforderungen an das System und das System selbst definiert werden. Darauf aufbauend müssen die Spezifikationen für das System ermittelt werden. Zudem bilden sie die Basis zur Realisierung des Design Modells und Implementierung des physischen Modells. Die Verifikation erfolgt durch Prüfung der einzelnen Anwendungsfälle. [Kru03]



Gemäß dem Ansatz der Ruph SE-Methode muss zu Beginn eine Stakeholderanfrage modelliert werden. Diese wird genutzt, um Anwendungsfälle zu definieren. Die Anwendungsfälle werden genutzt, um die Anforderungen detailliert zu beschreiben und auf Vollständigkeit zu prüfen. Zudem bilden sie die Basis zur Realisierung des Design Modells und Implementierung des physischen Modells. Die Verifikation erfolgt durch Prüfung der einzelnen Anwendungsfälle. Aus den Ergebnissen können die Anforderungen erfüllt werden. [Kru03]



CR = Stakeholderanforderung; SR = Systemanforderung; L = logisches Element;  
P = Physisches Element; TC = Testfall; UC = Anwendungsfall

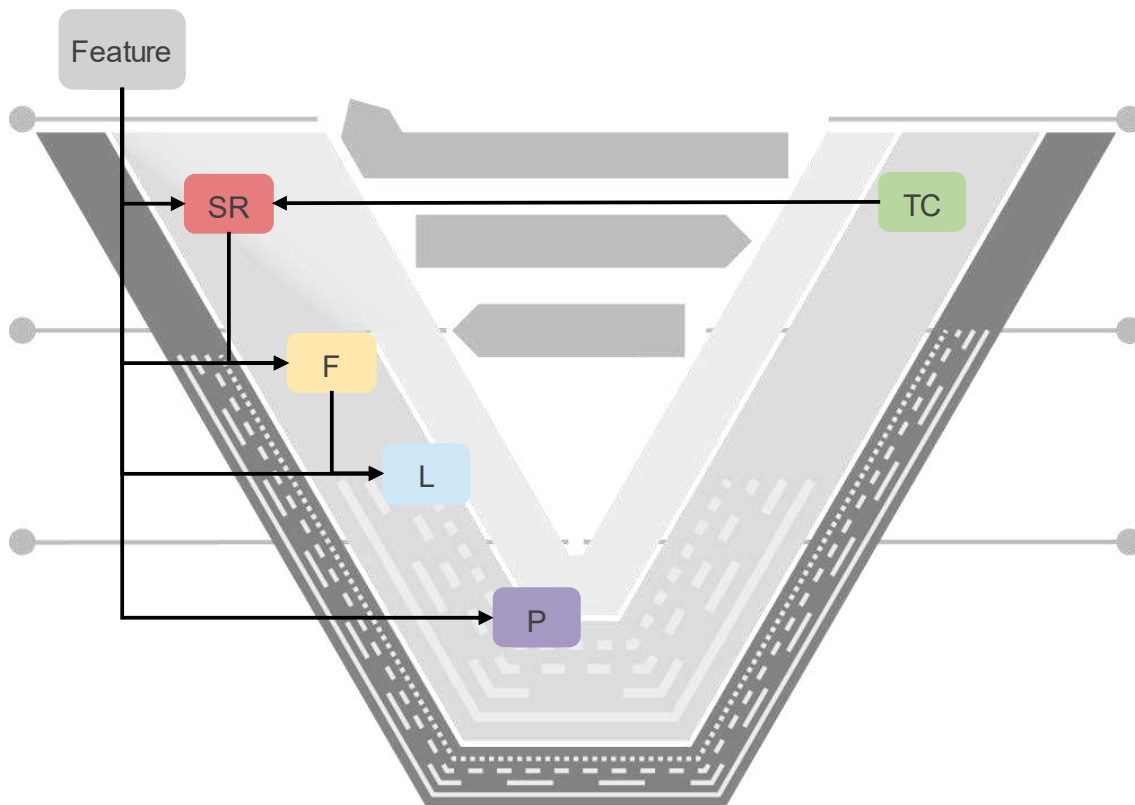
Abbildung 4.9: Darstellung der elementaren Entwicklungsartefakte der Ruph SE-Methode auf dem V-Modell verortet

#### 4.2.9 Pattern Based Systems Engineering

Ein weiterer Ansatz für die grundlegende Modellierung ist in dem Pattern Based Systems Engineering (PBSE) zu verorten. Der grundlegende Ansatz des PBSE beruht darauf, dass sich Wiederholungen in Form von Mustern in beispielsweise der Anforderungserhebung oder Systemarchitektur finden lassen. Das im Rahmen des PBSE nach Schindel [Sch20] entwickelte S\*Metamodell bildet dabei die Abläufe der Systementwicklung ab. Für die Entwicklung des Modells sind unterschiedliche Ansätze untersucht worden, um ein übergeordnetes Modell zu erzeugen. Der Ansatz beinhaltet grundlegende Entwicklungsartefakte, die in der Abbildung 4.10 dargestellt sind. [Wal23; Sch23]

Das Metamodell ist keiner genauen Modellierungssprache zugeordnet und Bedarf deshalb einer Übertragung in SysML. Außerdem existiert kein genaues Vorgehen für die

Methode, da dies je nach Ausprägung variieren kann. Das Verfahren beschreibt ein Feature, welches durch alle Funktionen und Komponenten erfüllt wird. Aus dem Feature, welches erfüllt werden muss, ergeben sich die Stakeholderanforderungen in Form eines „Stakeholderstatements“. Aus den Stakeholderanforderungen und dem Feature folgen die Funktionen, die durch das System erfüllt werden sollen. Aus den Funktionen leitet sich das Design ab, also die Architektur des Systems. Diese Architektur wird in das eigentliche System überführt. Für die Verifizierung des Systems muss ein Testfall mit der „verify“-Relation der Anforderung zugeordnet werden. [Sch23]



SR = Systemanforderung; F = Funktion; L = logisches Element;  
P = Physisches Element; TC = Testfall

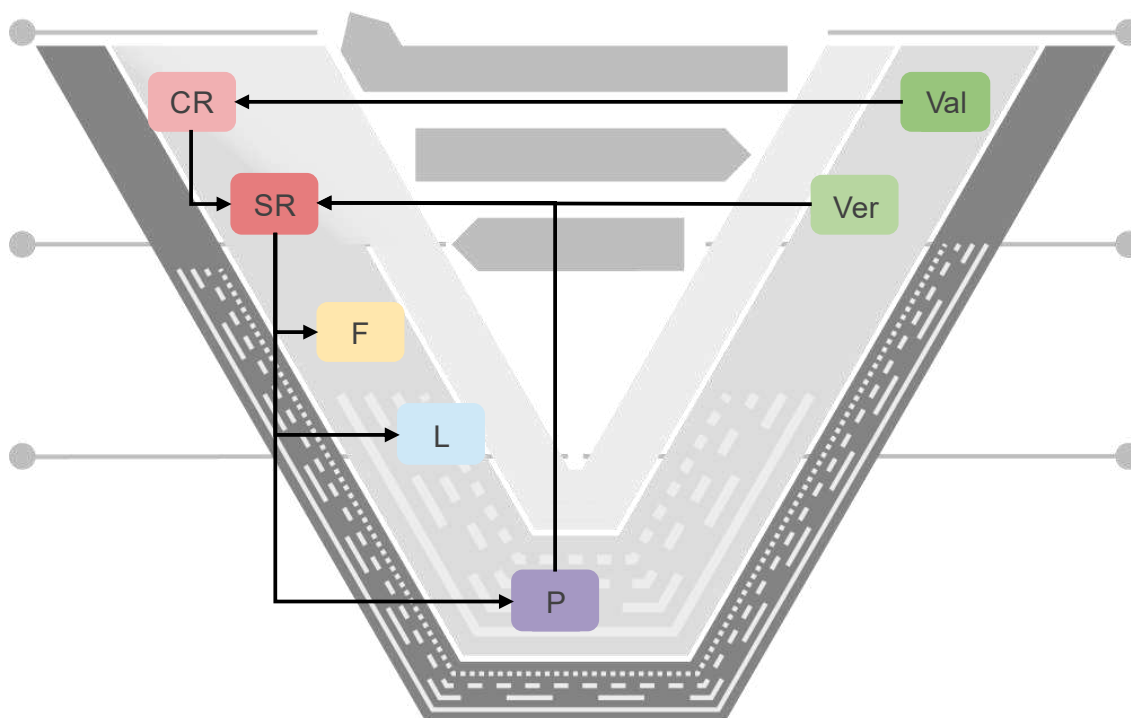
Abbildung 4.10: Darstellung der elementaren Entwicklungsartefakte des PBSE-Metamodells auf dem V-Modell verortet

#### 4.2.10 Munich agile MBSE Concept (MAGIC)

Der Ansatz der MAGIC-Methode liegt darin, einen agilen Ansatz aus der Softwareentwicklung mit Methoden des MBSE für cyberphysische Systeme zu kombinieren. Im Fokus steht, dass die Anforderungen jeder Lebenszyklusphase betrachtet werden. Durch den kontinuierlichen Ansatz soll vermittelt werden, dass die Prozesskette nicht mit der Auslieferung des Systems endet. Die in der Entwicklung grundlegenden Artefakte sind in der Abbildung 4.11 dargestellt [SW19]

Das MAGIC Modell beschreibt ein kontinuierliches Modell, welches aus sechs Ebenen besteht. Die erste Ebene bildet die Systemziel- und Anforderungsebene. Aufbauend darauf folgen die Systemfunktionsebene, die Systemarchitektur- und logische Ebene. Es schließen sich das Systemverhalten mit der Verifikation und Validierung an. Den Abschluss bilden die die Systemproduktions- und Testebene sowie die Systemnutzung und Service Ebene. [SW19]

Die MAGIC Methode beinhaltet als Entwicklungsartefakte die Stakeholderbedürfnisse, welche modelliert werden müssen. Diese Stakeholderbedürfnisse müssen in technische Anforderungen übersetzt werden, wobei die „define“-Relation verwendet wird. Zusätzlich ist eine Validierung der Stakeholderbedürfnisse durchzuführen, weshalb hier die Validierung mit der „verify“-Relation verbunden wird. Aus den technischen Anforderungen kann die funktionelle Systemarchitektur hergeleitet werden. Aus dieser werden die logische und die technische Systemarchitektur erarbeitet. Diese drei unterschiedlichen Entwicklungsartefakte werden alle mit der technischen Anforderung über die „trace“-Relation verbunden. Die technische Systemarchitektur ist darüber hinaus mit den technischen Anforderungen über die „satisfy“-Relation verknüpft. Die Verifikation wird mit Hilfe eines Artefakts mit der zugehörigen technischen Anforderung verbunden, wobei hier die „verify“-Relation genutzt wird. [SW19]



CR = Stakeholderanforderung; SR = Systemanforderung; F = Funktion;  
L = logisches Element; P = Physisches Element; Ver = Verifikation; Val = Validierung

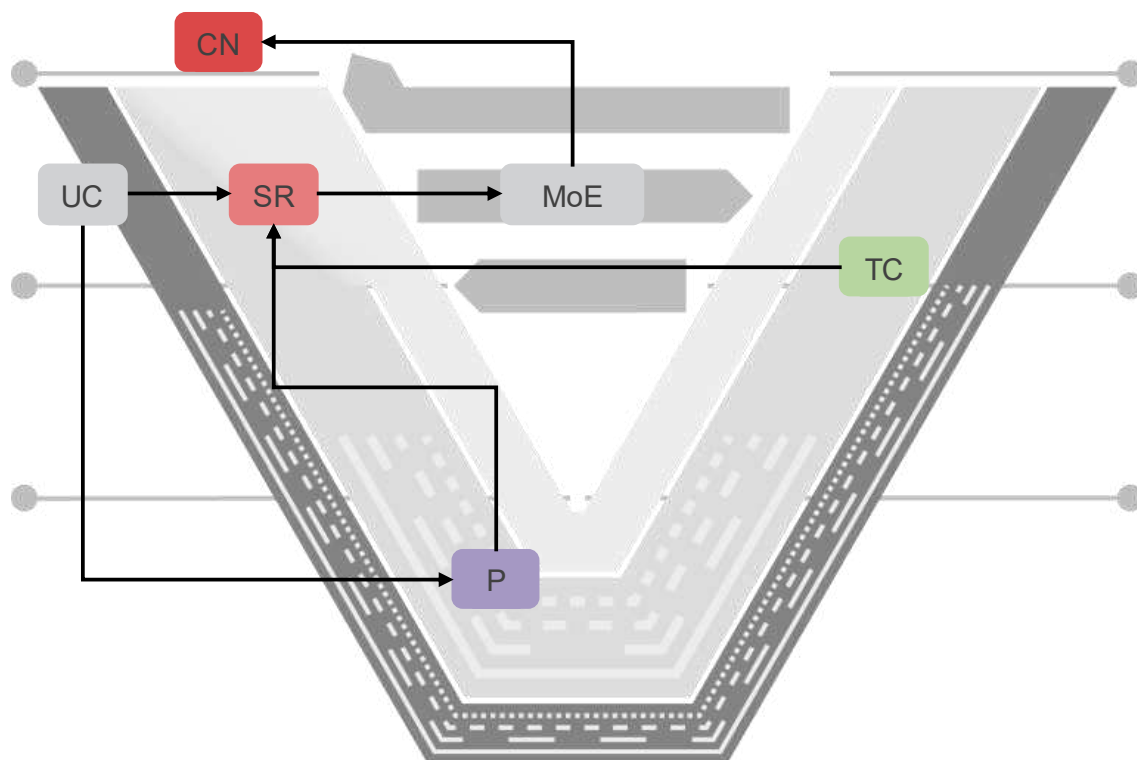
Abbildung 4.11: Darstellung der elementaren Entwicklungsartefakte der MUNICH-Methode auf dem V-Modell verortet

### 4.3 Fallstudien für die Modellierung cyberphysischer Systeme

Für die Ansätze der Modellierung cyberphysischer Systeme sind im Rahmen der Literaturrecherche acht relevante Ansätze in Form von Fallstudien aus der Industrie zu identifizieren. Die Identifikation der Ansätze ist mit Hilfe des dritten Suchstrings durchgeführt. Die betrachteten Ansätze verwenden alle die Modellierungssprache SysML. Im Folgenden werden die grundlegenden Annahmen der einzelnen Ansätze sowie die verwendeten Entwicklungsartefakte und deren Relationen genauer betrachtet.

#### 4.3.1 Luftwindenergieanlagen

In dem Journalbeitrag nach AZRAT UND HASSIM [AH18] wird die konzeptuelle Modellierung eines luftgestützten Windenergiesystem dargelegt. Die Modellierung orientiert sich an der TOPSIS-Methode. Das System soll die besonderen geografischen Faktoren des Einsatzgebietes erfüllen. Die identifizierten Entwicklungsartefakte und deren Relationen sind in der Abbildung 4.12 dargestellt. [AH18]



CN = Stakeholderbedürfnis ; SR = Systemanforderung ; MoE = Effektivitätsmaßstab ;  
P = Physisches Element ; TC = Testfall ; UC = Anwendungsfall

Abbildung 4.12: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für die Luftwindenergieanlage auf dem V-Modell verortet

Das Vorgehen ist so aufgebaut, dass zu Beginn die Stakeholderanforderungen analysiert werden. Dabei werden der Ist-Zustand aufgenommen und mögliche Entwicklungspotentiale herausgestellt. Im nächsten Schritt wird eine Funktionsanalyse durchgeführt, bei der

das System mit Entitäten außerhalb des Systems interagieren muss. Mit Hilfe der Ergebnisse kann die logische Struktur des Systems erstellt werden. Mit der logischen Struktur kann die physische Architektur des Systems festgelegt werden. Die erzeugten Systemartefakte müssen zu den Anforderungen rückverfolgbar sein, aus denen diese erzeugt wurden. Im letzten Schritt kann das System optimiert werden. [AH18]

Der strukturelle Aufbau der Systemartefakte sieht dabei so aus, dass mit Hilfe des Blockdefinitionsdiagramms die Systemstruktur mit den Signal- und Energieflüssen modelliert wird. Ebenfalls wird der Systemkontext definiert, in dem alle Umwelteinflüsse durch Akteuren oder anderen Systemen dargestellt werden. Zusätzlich müssen die Stakeholderbedürfnisse modelliert werden. Mit Hilfe der Stakeholderbedürfnisse können im Anforderungsdiagramm die Anforderungen abgeleitet und zerlegt werden. Die Anforderungen definieren die Effektivitätsmaßstäbe (MoE). Die Systemfunktionen werden mit dem Anwendungsfall-Diagramm dargestellt. Der Ablauf der einzelnen Funktionen wird im Aktivitätsdiagramm modelliert. Die Anforderungen werden durch die Systemelemente erfüllt und zusätzlich zu den Testfällen mit der „Trace“-Relation verbunden. [AH18]

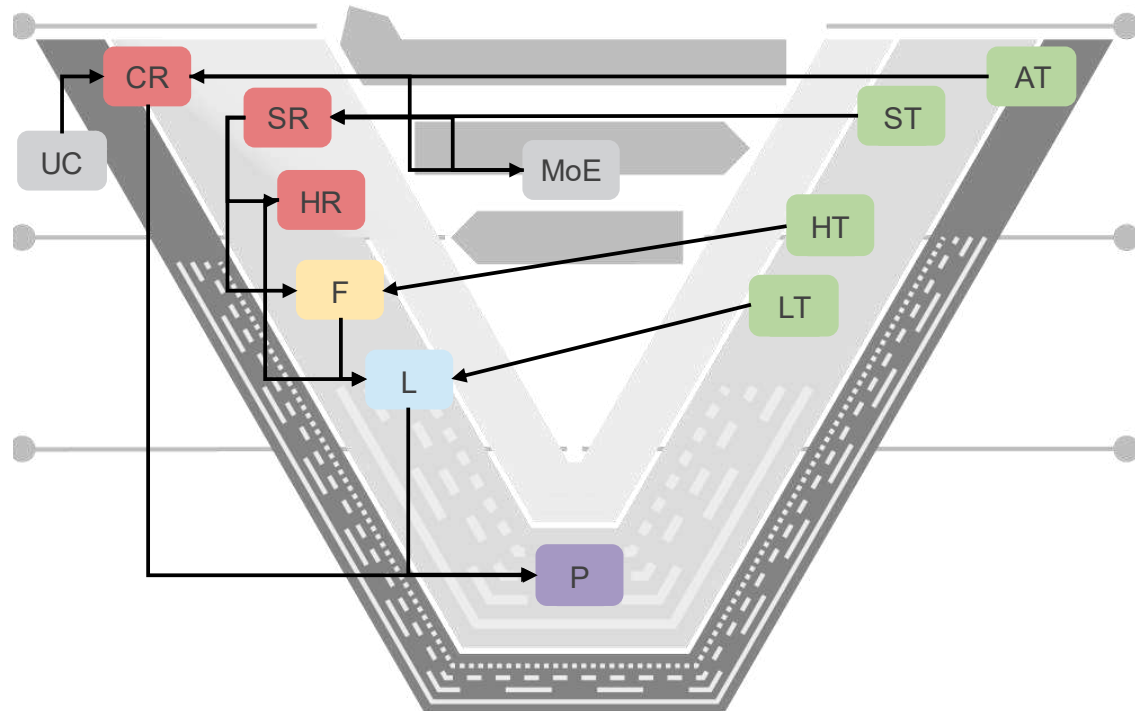
#### **4.3.2 Flugzeugfahrwerk**

In einer Fallstudie nach WILICH ET AL. [FCA+21] wird die Entwicklung eines Flugzeugfahrwerks detailliert dargelegt. Die Entwicklung wird über die Anwendungsfälle hergeleitet und zeigt auf, wie die unterschiedlichen Disziplinen miteinander interagieren. Zusätzlich wird die Relevanz des MBSE für die Produktsicherheit aufgezeigt. Das Vorgehen ist an der CONSENS-Methode orientiert. Die im Rahmen der Fallstudie verwendeten Systemartefakte werden in der Abbildung 4.13 dargestellt. [FCA+21]

Zu Beginn der Modellierung werden mit Hilfe der Anwendungsfälle und Umwelteinflüsse ein einheitliches Bild des Systems geschaffen. Die Anwendungsfälle beschreiben das Gesamtsystem, also bezogen auf dieses Beispiel das gesamte Flugzeug. Aus den Anwendungsfällen werden die Stakeholderanforderungen abgeleitet. Es ist in dem Ansatz zu beachten, die Stakeholderanforderungen nicht allein aus den Bedürfnissen abzuleiten, da diese nicht vollständig sein müssen. In dem nächsten Schritt werden aus den Stakeholderanforderungen die Systemanforderungen transformiert. Die Systemanforderungen können in weitere Anforderungen zerlegt werden. Hieraus ergibt sich die logische Struktur des Systems. Aufbauend darauf können aus der Systemarchitektur die High-Level-Anforderungen abgeleitet werden. Es müssen den Anforderungen bereits Parameter übergeben werden. Im letzten Schritt kann die Architektur der physischen Elemente aus den High-Level-Anforderungen abgeleitet werden, wobei diese aus unterschiedlichen Design-Elementen besteht. [FCA+21]

Aus dem Vorgehen ergibt sich für die Struktur, in der die Artefakte zueinanderstehen, dass in diesem Ansatz vier Ebenen betrachtet werden. In der ersten Ebene werden mit Hilfe der Anwendungsfälle und den Umgebungseinflüssen die Fehlerbedingungen, die

Stakeholderanforderungen und die Akzeptanztests für das zu entwickelnde System hergeleitet. Zur Erfüllung der Stakeholderanforderungen werden in der Systemebene die Systemanforderungen definiert, denen Systemtests zugeordnet sind. Mit den Systemanforderungen und den Funktionen ist die Systemstruktur verknüpft. Aus der Systemstruktur sind die High-Level-Anforderungen abgeleitet, welche durch High-Level-Tests verifiziert werden. Diese sind wiederum mit dem Design verknüpft, welches durch Low-Level-Tests verifiziert wird. [FCA+21]



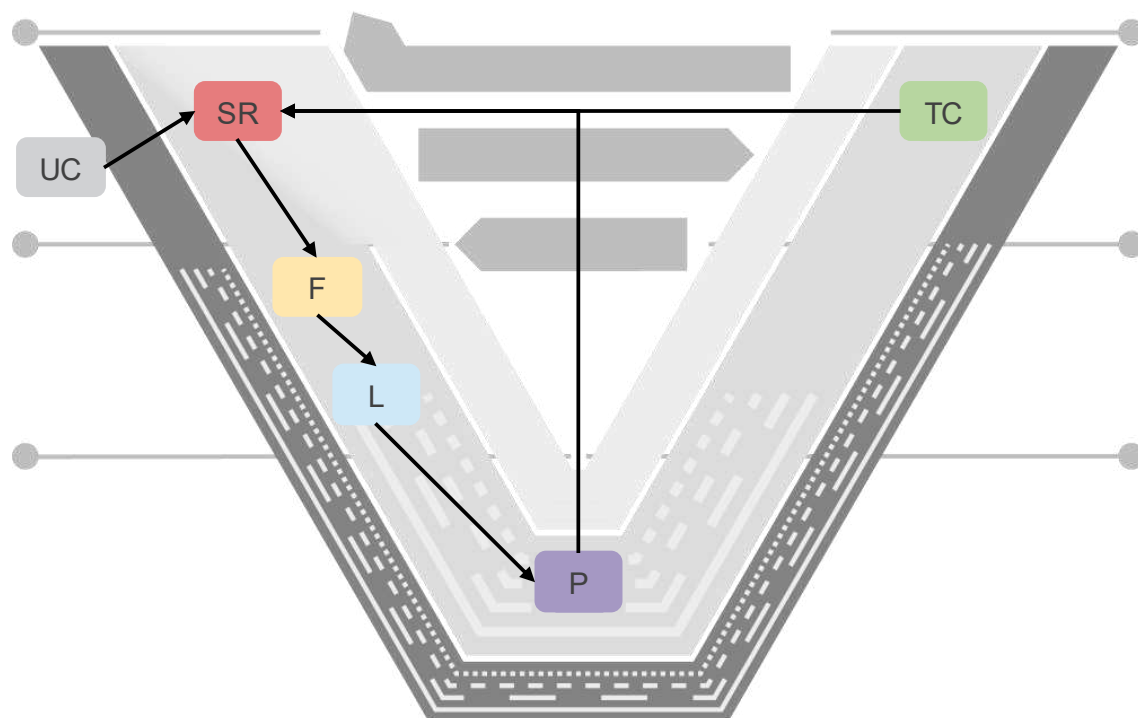
CR = Stakeholderanforderung; SR = Systemanforderung; L = logisches Element;  
 F = Funktion; P = Physisches Element; AT = Akzeptanztest; UC = Anwendungsfall;  
 MoE = Effektivitätsmaßstab; ST = Systemtest; HT = High-Level Test;  
 LT = Low-Level Test

Abbildung 4.13: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für ein Flugzeugfahrwerk auf dem V-Modell verortet

### 4.3.3 Demonstrator für die Luftfahrt

Im Rahmen einer Ausarbeitung nach CUI ET AL. [ZMC+22] wird ein herunterskalierter Demonstrator für die Luftfahrt entwickelt. Mit Hilfe des Demonstrators sollen kostengünstig Tests von verschiedenen Konfigurationen erstellt werden. Das Vorgehen ist an der Magic Grid-Methode orientiert; die verwendeten Artefakte sind in der Abbildung 4.14 dargestellt. [ZMC+22]

Das Vorgehen für die Entwicklung sieht dabei zu Beginn die Erfassung der Stakeholderbedürfnisse vor. Die Bedürfnisse der Stakeholder werden in Anforderungen überführt, wobei hier eine Aufteilung gemäß den Systemebenen vorgesehen wird. Die Anforderungen werden in funktionale und nicht funktionale Anforderungen unterteilt, da diese eine unterschiedliche Art der Verifikation benötigen. Die nicht funktionalen Anforderungen werden durch Parameter, welche sich aus dem System ergeben, verifiziert. Die funktionalen Anforderungen hingegen bedürfen einer Verifikation durch das Systemverhalten. Auf Basis der Stakeholderanforderungen und der Erfahrung aus Vorgänger-Versionen wird die Funktionsanalyse durchgeführt. Mit Hilfe der Funktionsanalyse werden die Anwendungsfälle entwickelt. Nachdem die Funktionsanalyse durchgeführt worden ist, kann die Funktionshierarchie aufgebaut werden, welche die Basis für den Aufbau der logischen Struktur bildet. Zusätzlich müssen die Effektivitätsmaßstäbe modelliert werden, welche als Maß für den Erfüllungsgrad der Stakeholderanforderungen dienen. [ZMC+22]



SR = Systemanforderung; L = logisches Element; F = Funktion;  
P = Physisches Element; TC = Testfall; UC = Anwendungsfall

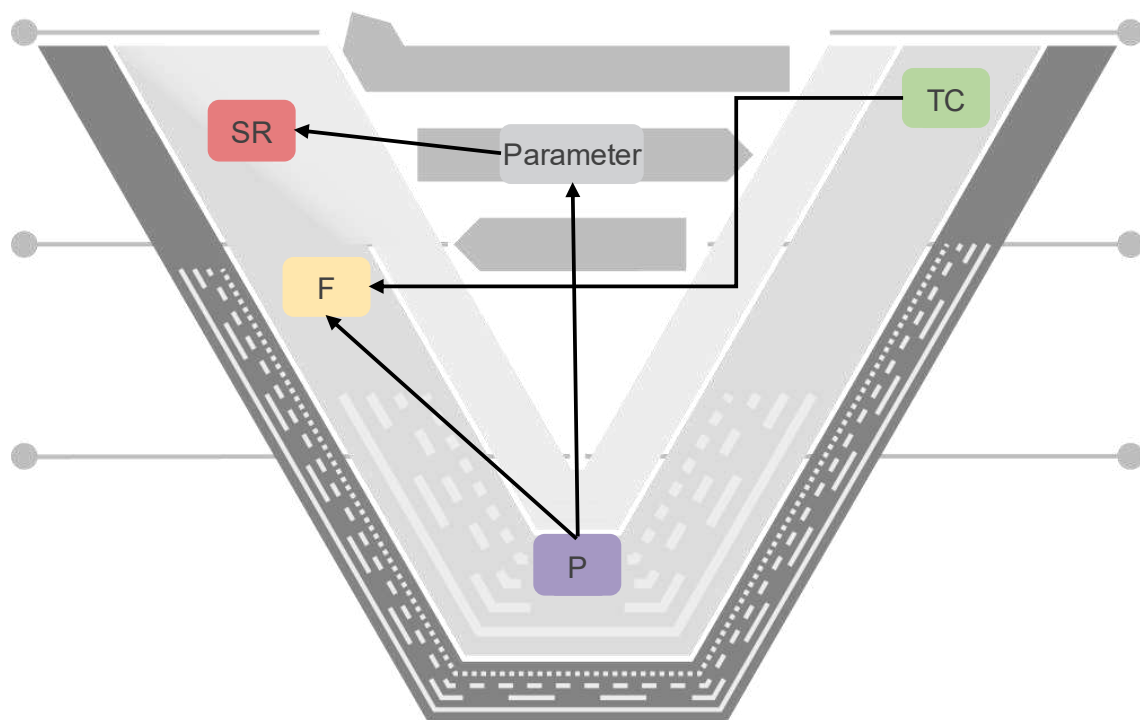
Abbildung 4.14: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für den Demonstrator eines Flugzeugs auf dem V-Modell verortet

Der Aufbau der Systemartefakte sieht dabei vor, dass die Anforderungen in einem Anforderungsdiagramm gemäß der Hierarchie modelliert werden. Diese werden durch die Systemparameter oder durch das Systemverhalten erfüllt. Die Parameter werden als Randbedingung im Parameterdiagramm abgebildet. Das Systemverhalten hingegen wird mit Hilfe eines Zustandsdiagramms dargestellt. Die für die Modellierung des Verhaltens und der Parameter nötige Systemstruktur wird in einem internen Blockdiagramm abgebildet. Die Modellierung des Systems und der Subsysteme wird in Form von Blöcken in einem Blockdefinitionsdiagramm modelliert. [ZMC+22]



#### 4.3.4 Turbine einer Windkraftanlage

In einem Artikel nach ZHANG ET AL. [ZHB+21] soll gezeigt werden, dass auf Basis eines Systemmodells mit der Verbindung zu anderen Software-Tools die Möglichkeit besteht, Simulationen durchzuführen. Hierzu wird das Systemmodell einer Turbine aus einer Windkraftanlage verwendet. Zur Modellierung wurde sich in diesem Rahmen nicht an einer Methode für die Modellierung von Systemmodellen orientiert. Die modellierten Entwicklungsartefakte und die Relationen sind in der Abbildung 4.15 dargestellt. [ZHB+21]



SR = Systemanforderung; F = Funktion; P = Physisches Element;  
TC = Testfall

Abbildung 4.15: Darstellung der elementaren Entwicklungsartefakte der Fallstudie für die Turbine einer Windkraftanlage auf dem V-Modell verortet

Nach diesem Ansatz wird mit der Definition von Anforderungen gestartet. Dabei ist grundlegend zwischen funktionalen und nicht funktionalen Anforderungen zu unterscheiden. In diesem Ansatz werden die nicht funktionalen Anforderungen durch Eigenschaftswerte, die funktionalen Anforderungen durch funktionales Testen erfüllt. Aufbauend darauf werden die Funktionen modelliert. [ZHB+21]

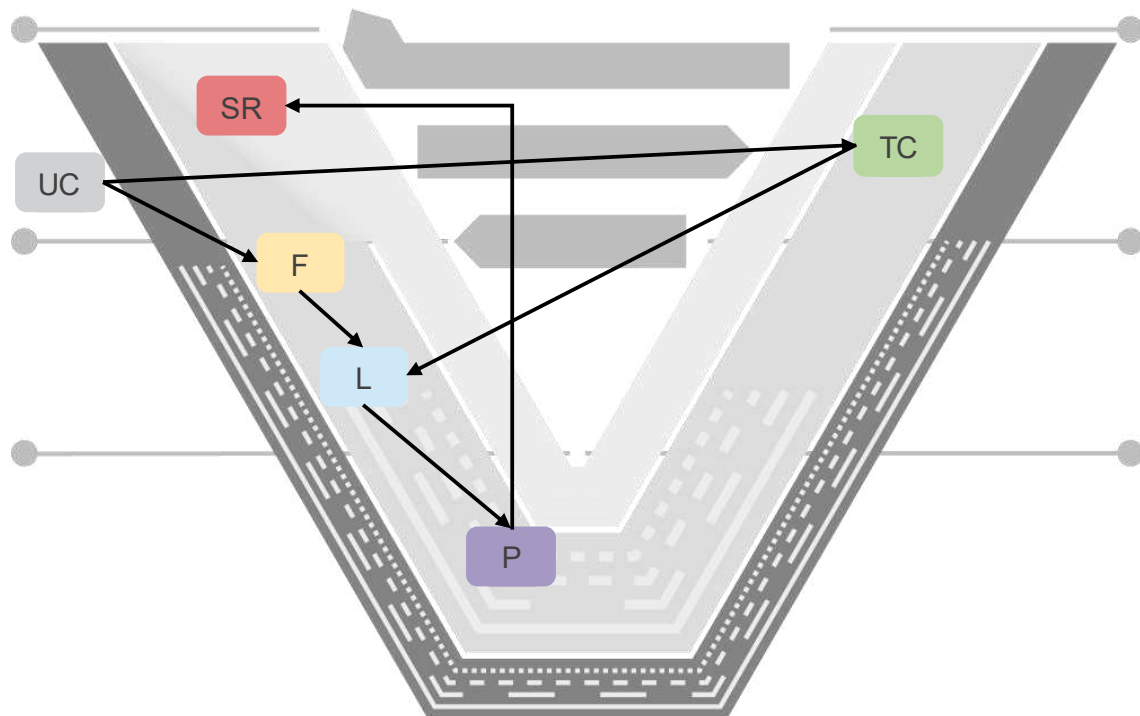
Die verwendeten Entwicklungsartefakte umfassen dabei Anforderungen, welche hierarchisch aufgebaut werden. Im Fall der funktionalen Anforderungen werden diese durch die Funktionen erfüllt. Diese Funktionen sind als Block modelliert. Die nicht funktionalen Anforderungen werden durch die Parameter erfüllt, welche als „valueType“ modelliert sind. Zur Modellierung der funktionalen Tests werden Aktivitätsdiagramme verwendet. Zusätzlich werden über die „trace“-Relation Artefakte mit den Anforderungen verknüpft,



falls dieser Einfluss auf Parameter der Anforderung haben. Die interne Systemstruktur wird in diesem Fall mit Hilfe des internen Blockdiagramms modelliert und umfasst alle Energie und Stoffflüsse. Zusätzlich werden hierdurch Berechnungen durchgeführt. [ZHB+21]

#### 4.3.5 Bahnstreckensignal

Im Rahmen eines Konferenzbeitrags nach STEVESON ET AL. [SVT18] ist die Modellierung und Simulation für ein Bahnstreckensignal dargelegt. Die Umsetzung ist im Rahmen der Entwicklung mit Hilfe von der OMG Systems Modelling Language durchgeführt. Die Entwicklung hat sich darauf beschränkt, ein Modell aufzubauen, welches drei Sichten unterstützt. Diese sind die Kommissionierung, Betrieb und Instandhaltung des Bahnstreckensignals. Die für die Modellierung verwendeten Artefakte und Relationen sind in der Abbildung 4.16 dargestellt. [SVT18]



SR = Systemanforderung; L = logisches Element; F = Funktion;  
P = Physisches Element; TC = Testfall; UC = Anwendungsfall

Abbildung 4.16: Darstellung der elementaren Entwicklungsartefakte der Fallstudie eines Bahnstreckensignals auf dem V-Modell verortet

Für die Modellierung ist zu Beginn der Systemkontext festzulegen. Auf Basis des Systemkontextes werden die nötigen Funktionalitäten mit Hilfe von Anwendungsfällen hergeleitet. Für die Verknüpfung der funktionalen Anforderungen wird die logische Struktur des Systems ermittelt. Um das Verhalten der einzelnen Systemelemente genau zu definieren, sind mit Hilfe von Verhaltensdiagrammen die einzelnen Systemelemente zu modellieren. Aus den zuvor aufgestellten Anwendungsfällen werden die Testfälle abgeleitet,

welche für die Verifikation des Modells verwendet werden. Die Verifikation umfasst dabei Blackbox- und Whitebox-Tests. Bei den Blackbox-Tests wird nur die Ausgabe für die Benutzeroberfläche betrachtet, bei den Whitebox werden die Werte für die einzelnen Systemelemente ins Auge gefasst. Abschließend wird das System mit Hilfe von Szenarios validiert. [SVT18]

Für den Aufbau der Systemstruktur ist das System zunächst zu zerlegen, um einen hierarchischen Aufbau zu erzeugen. Es wird mit der logischen Struktur begonnen, welche der physischen Struktur zugeordnet ist. Die Systemfunktionen werden nicht direkt zugeordnet, sondern nur über die Anwendungsfälle modelliert, aus denen das Systemverhalten abgeleitet wird. Die Anforderungen weisen jedoch wie die Systemelemente eine hierarchische Struktur auf und werden durch die physischen Elemente erfüllt. [SVT18]

#### **4.3.6 Weltraumrobotersystem**

In dem Paper nach YOUNSE ET AL. [YCB22] ist die Modellierung für ein Weltraumrobotersystem dargelegt. Die Entwicklung des Systems basiert auf einer Abwandlung des V-Modells. Die Darstellung der einzelnen Entwicklungsartefakte wird mit Hilfe von SysML dargestellt. Das System betrachtet mehrere Systemebenen, welche in der Abbildung 4.17 mit den zugehörigen Entwicklungsartefakten abgebildet sind. [YCB22]

Der generelle Aufbau sieht eine Erweiterung des V-Modells vor. Es wird unterschieden zwischen einem physischen und virtuellen System sowie zwischen einem Konzept und einem computergestützten Modell. Das physische Modell folgt der Entwicklung gemäß dem V-Modells. Das konzeptionelle Modell beinhaltet wichtige Systemelemente, deren Funktionen und Annahmen. Das Simulationsmodell basiert auf dem konzeptionellen Modell. Das virtuelle System stellt die Instanziierung durch das Computermodell auf Basis der Simulation dar. [YCB22]

Die verwendeten Artefakte für die Systemstruktur umfassen die grundlegende Struktur und die aus der Zerlegung entstehenden Subsysteme. Hieraus entstehen Artefakte, welche auf einem anderen hierarchischen Niveau sind. Vereinfacht wird hier das System mit dem Subsystem und den einzelnen Systemelementen dargestellt. Zusätzlich werden mathematische Abhängigkeiten gebildet und den Elementen Parameter zugeordnet. Das Systemverhalten wird gemäß der Systemstruktur aufgebaut. Es werden Funktionen zerlegt und in die Systemebenen eingeordnet. Den Funktionen werden die Systemelemente zugeordnet, welche diese erfüllen sollen. Für die Anforderungen an das System wird ebenfalls eine hierarchische Struktur aufgebaut. Zusätzlich werden Funktionen und Systemelemente zugeordnet, welche diese Anforderung erfüllen. Auf Basis dieser Informationen

können die Simulationen durchgeführt werden, wobei die Simulation mit Hilfe eines zusätzlichen Tools durchgeführt wird. [YCB22]

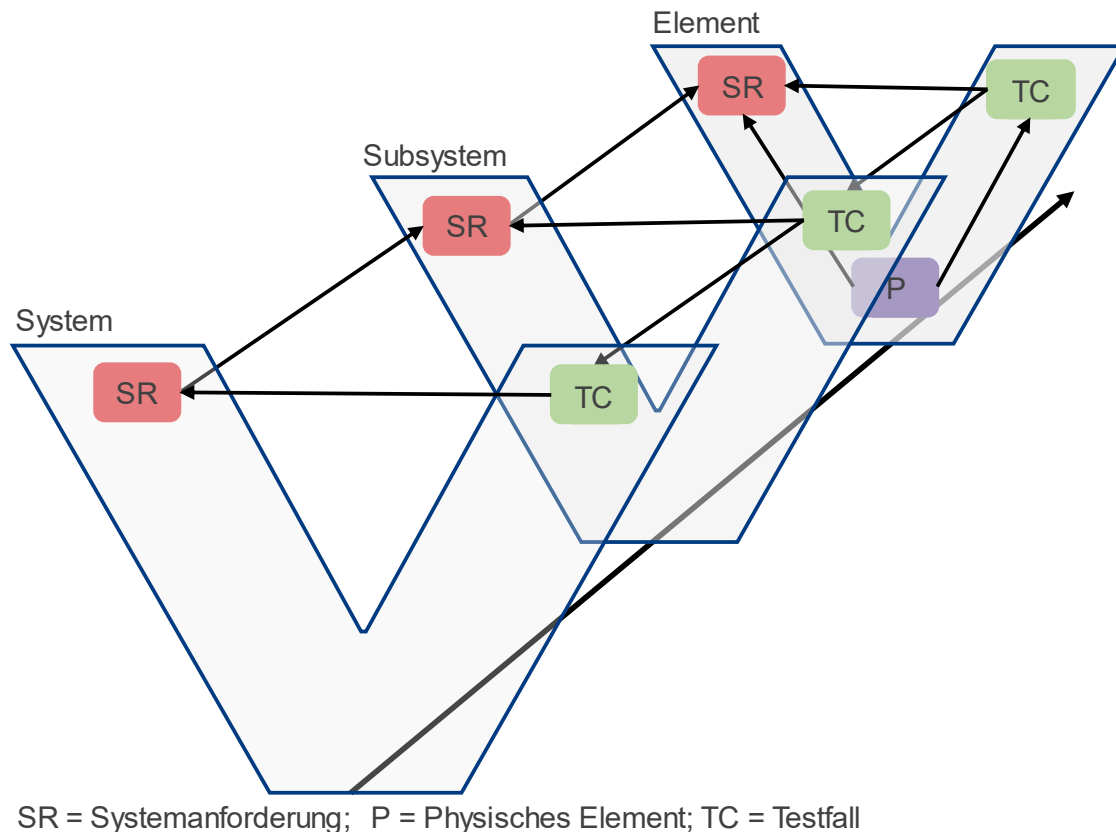


Abbildung 4.17: Darstellung der elementaren Entwicklungsartefakte der Fallstudie Weltraumroboters auf dem V-Modell gemäß der Systemebenen verortet

#### 4.3.7 Fahrerassistenzsystem

Im Rahmen einer Ausarbeitung nach QAMAR ET AL. [QMW17] ist für die Unterstützung der Fehlervermeidung eine Modellierung eines Fahrerassistenzsystems dargelegt. Das Ziel besteht darin, durch die Modellierung und mit Hilfe verschiedener Sichten auf das System früher Fehlereinflüsse zu identifizieren. Der Ansatz soll die Rückverfolgbarkeit des Modells erhöhen. Die in dem Systemmodell verwendeten Entwicklungsartefakte sowie die Relationen sind in der Abbildung 4.18 dargestellt. [QMW17]

Die Umsetzung der Modellierung basiert auf drei Systemebenen: der Kundenebene, der Systemebene und der Implementierungsebene. Alle Attribute, welche den Kunden betreffen, werden durch das Fahrzeugteam getestet. Für die Umsetzung wird zusätzlich eine domänenspezifische Sprache (DSL) implementiert. Diese wird genutzt, um die firmeninternen Definitionen verwenden zu können. [QMW17]

Die generelle Systemstruktur ist so aufgebaut, dass die Anforderungen der jeweiligen Ebene durch die Systemstruktur erfüllt werden. Die Systemstruktur ist mit dem betrachteten System verbunden. Der Test auf derselben Systemebene verifiziert die jeweilige Anforderung. Die Systemstruktur wird in weitere Subsysteme zerlegt, wobei die Anforderungen für das Subsystem die Systemstruktur verfeinern. Die jeweiligen Anforderungen des Subsystems werden aus den Anforderungen der höheren Systemebene abgeleitet. [QMW17]

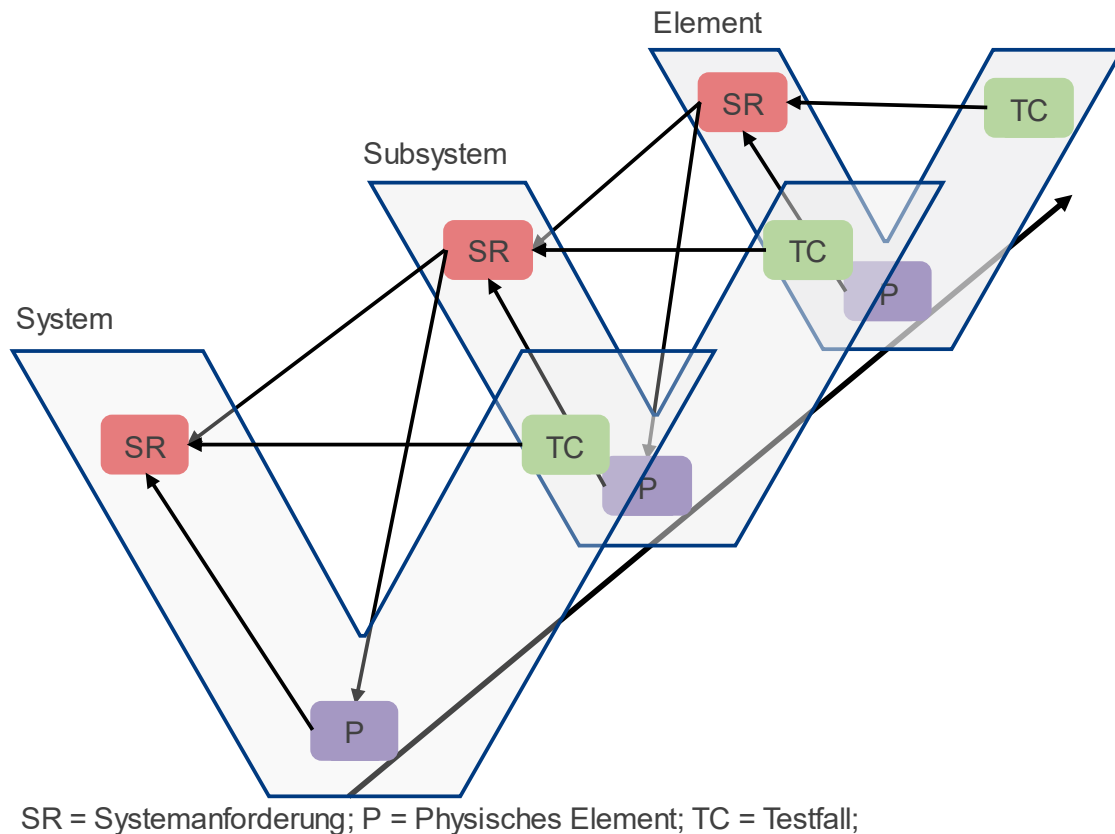


Abbildung 4.18: Darstellung der elementaren Entwicklungsartefakte der Fallstudie eines Fahrerassistenzsystems auf dem V-Modell gemäß der Systemebenen verortet

#### 4.3.8 Produktionsmaschinen

In der Veröffentlichung nach BASSI ET AL. [BSB+10] wird eine Methode für die Modellierung komplexer technischer Systeme anhand von Produktionsmaschinen erläutert. Mit Hilfe der Modellierung soll Optimierung des Systems begünstigt werden. Die Umsetzung ist mit Hilfe von SysML durchzuführen, wobei die verwendeten Systemartefakte und deren Relation in der Abbildung 4.19 dargestellt sind. [BSB+10]

Das Vorgehen zur Modellierung komplexer Produktionsmaschinen sieht drei Systemebenen vor. Begonnen wird mit der sogenannten „Refinement“-Ebene. In dieser Ebene werden die Kernkomponenten des Systems sowie das Gesamtsystem definiert. Aufbauend wird progressiv fortgefahren, bis ein ausführbares Model entsteht. Bei Entscheidungen sind hier Lösungsalternativen auszuarbeiten. [BSB+10]

Die Systemstruktur ist in der Fallstudie so aufgebaut, dass die Anforderungen abgeleitet werden. Hieraus ergeben sich die Systemebenen. Die Anforderungen werden im Anforderungsdiagramm abgebildet und können sich gegenseitig beeinflussen oder stören. Für die Systemstruktur wird ein Blockdefinitionsdiagramm verwendet, in dem das betrachtete System in Subsysteme und Elemente zerlegt wird. Mit Hilfe des internen Blockdefinitionsdiagramms kann die innere Struktur des Systems dargestellt werden. Die innere Struktur beinhaltet Signal-, Energie- und Stoffflüsse. Aus den Systemstrukturen und den Parametern können die Testfälle erfüllt werden. Diese Testfälle werden genutzt, um die Anforderungen des Elements zu verifizieren. [BSB+10]

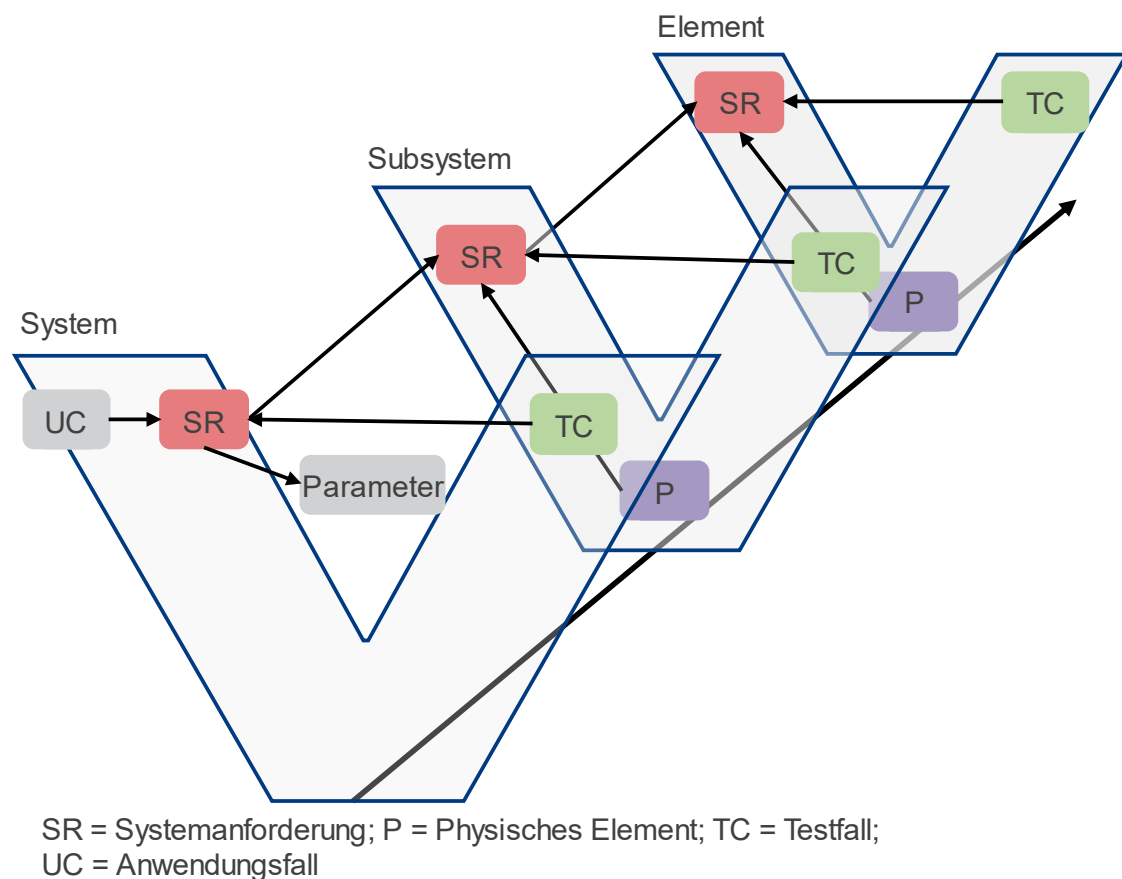


Abbildung 4.19: Darstellung der elementaren Entwicklungsartefakte der Fallstudie von Produktionsmaschinen auf dem V-Modell gemäß der Systemebenen verortet

## 5 Analyse testinduzierter Änderungen

In diesem Kapitel wird aus den Erkenntnissen des Stands der Technik und den theoretischen Grundlagen das Vorgehen für die Bewertung und Verarbeitung testinduzierter Änderungen mit Hilfe von Auswirkungsmustern hergeleitet. Für die einzelnen Auswirkungsmuster werden Steckbriefe entwickelt, welche die Informationen für die Bestimmung und Handhabung beinhalten. Die Beschreibung der Auswirkungsmuster wird durch Steckbriefe durchgeführt, damit der Anwender bei der Anwendung eine ganzheitliche Beschreibung des Problems hat. Diese kann direkt überblickt werden, bei der Suche oder der Anwendung. Auf Basis der Auswirkungsmuster und vorgegebenen Randbedingungen werden die Handlungsmaßnahmen zur Identifizierung betroffener Entwicklungsartefakte erarbeitet. Anschließend soll mit den Auswirkungsmustern eine Bewertung der Kritikalität der auftretenden Änderungen erzeugt werden. Das Vorgehen für die Bewertung und Verarbeitung wird im letzten Unterkapitel exemplarisch mit Hilfe eines Praxisbeispiels erörtert.

### 5.1 Identifikation von Auswirkungsmustern

Für das Bewerten von Änderungen und dem Ableiten der Handlungsmaßnahmen sind als Grundlage Auswirkungsmuster zu erarbeiten. Da bislang keine Ansätze vorhanden sind, die vergleichbare Auswirkungsmuster herleiten, werden diese aus theoretischen und praktischen Ansätzen erarbeitet. Für die Erarbeitung werden verschiedene Ansätze hinsichtlich der Gestaltung der Systemmodelle untersucht. Die Auswertung der verwendeten Ansätze ist in dem Kapitel 4 dargestellt. Für die Sicherstellung der Praxisrelevanz und Anwendbarkeit erfolgt ein Abgleich zwischen den theoretischen und praktisch umgesetzten Modellierungsmethoden. Die praktisch umgesetzten Modellierungsmethoden basieren auf Fallstudien, die für den industriellen Sektor aufgestellt werden. Das Vorgehen ist exemplarisch in der Abbildung 1.2 dargestellt. Für die Erkenntnisgewinnung liegen zehn Ansätze aus der Theorie und acht Ansätze aus der Praxis vor.

Die erarbeiteten Auswirkungsmuster basieren auf den Entwicklungsartefakten und Relationen der einzelnen Entwicklungsartefakte. Auf Basis dieser Ansätze werden durch Testfälle bedingte Änderungen betrachtet. Es werden für jede betrachtete Änderungsart die Auswirkungen auf die einzelnen Ansätze analysiert. Die betroffenen Elemente werden entsprechend der jeweiligen Auswirkung markiert. Diese Markierungen variieren, abhängig davon, ob Entwicklungsartefakte direkt oder nur indirekt betroffen sind. Anschließend werden die einzelnen Systemartefakte geclustert und die entstehenden Cluster in resultierende Systemartefakte überführt. Die möglichen Relationen werden zwischen den einzelnen Clustern anhand der Erkenntnisse aus der Ausarbeitung interpretiert. Es folgt die Untersuchung der unterschiedlichen Möglichkeiten hinsichtlich der Auswirkung auf die Propagationswirkung. Für die einzelnen Auswirkungsmuster werden hierzu unterschiedliche Arten an Änderungen angenommen beispielsweise die Art des geänderten Artefakts oder dass ein Testfall zu einer neuen Anforderung führt.

Um die verschiedenen Ansätze vergleichen zu können, werden die verwendeten Artefakte jedes Ansatzes auf dem V-Modell verortet. Somit ist es möglich, nach der Analyse der Auswirkung einer Änderung die auftretenden Muster miteinander zu vergleichen. Es zeigt sich dadurch, welches änderungsbedingte Verhalten die Systemmodelle aufweisen. Weiterführend lässt sich ableiten, welche Auswirkung am häufigsten auftritt und wo eventuelle Unterschiede zwischen den einzelnen Systemmodellen bestehen.

## 5.2 Darstellung der Auswirkungsmuster

Im folgenden Kapitel wird die entwickelte Darstellung der Auswirkungsmuster erläutert. Die Darstellung wird mit einem Steckbrief umgesetzt. Dessen Struktur orientiert sich dabei an dem Aufbau des Steckbriefs für Systemelemente aus dem Ansatz nach GRÄBLER ET AL. [GWO22]. Der Steckbrief beinhaltet dabei die Informationen zur Bestimmung des Auswirkungsmusters. Darüber hinaus ist das detektierte Auswirkungsmuster, die nötigen Handlungsmaßnahmen und die Bewertung des jeweiligen Musters dargestellt

Der Aufbau des Steckbriefs sieht in der obersten Zeile die Bezeichnung des Auswirkungsmusters vor. Diese Bezeichnung ist aus drei Komponenten aufgebaut wie in der Abbildung 5.1 dargestellt wird, wobei die Abkürzungen für die englischen Worte stehen. Dies ist so gewählt, damit eine bessere Übertragbarkeit für internationale Anwender gegeben ist und Umlaute vermieden werden.

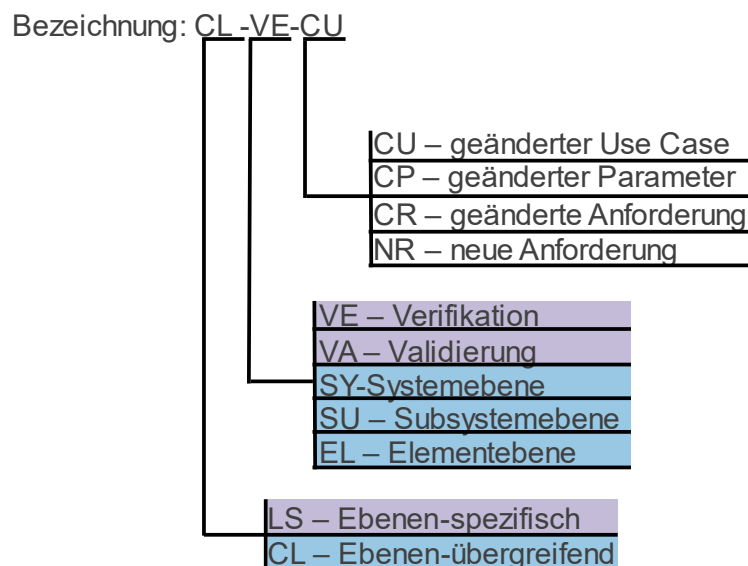


Abbildung 5.1: Klassifizierungssystem für die Bezeichnung der Auswirkungsmuster

Die ersten beiden Buchstaben bestimmen die Kategorie des Musters. Es wird unterschieden zwischen den ebenen-spezifischen Mustern und den ebenen-übergreifenden Mustern. Der zweite Block unterscheidet sich je nach Kategorie und steht bezüglich der ebenen-übergreifenden Muster für die initiale Ebene, auf der die Eigenschaftsabsicherung durchgeführt wird. Somit stehen die drei Systemebenen zu Verfügung. Im Fall der ebenen-spezifischen Auswirkungsmuster wird unterschieden zwischen den Abkürzungen für die

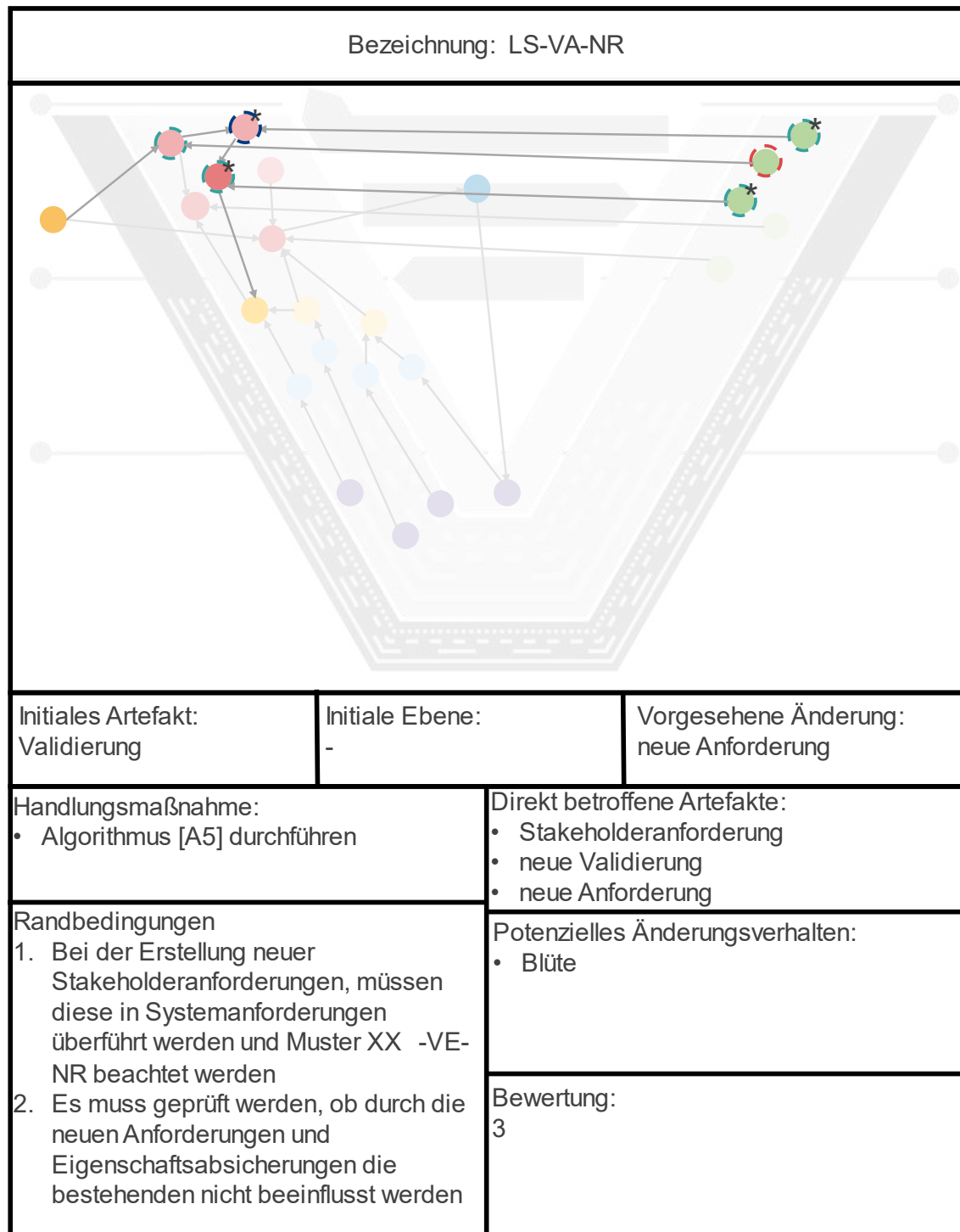
Art der Eigenschaftsabsicherung. Somit wird zwischen Validierung und Verifikation unterschieden. Der letzte Block der Bezeichnung ist einheitlich definiert für beide Kategorien von Auswirkungsmustern. Dieser Block gibt Aufschluss über die vorgesehene Änderung, welche aus dem Änderungsbedarf resultiert. Hierbei wird zwischen den folgenden vier Änderungsmaßnahmen unterschieden. Die Abbildung 5.2 zeigt das Beispiel eines Steckbriefs zur Erstellung einer neuen Stakeholderanforderung.

Im Zentrum des Steckbriefs steht das Auswirkungsmuster und stellt die Abhängigkeiten zwischen den Entwicklungsartefakten dar. In der Darstellung sind die Auswirkungsmuster auf dem V-Modell verortet, wobei zwei Betrachtungen vorliegen. Es werden die Auswirkungen auf einer Ebene analysiert und zusätzlich über die drei Systemebenen. Mit Hilfe der Darstellung soll aufgezeigt werden, wie sich die Änderung über die einzelnen Entwicklungsartefakte erstreckt.

In den ebenen-spezifischen Mustern des Beispiels in der Abbildung 5.2 werden neben den Systemanforderungen auch die Stakeholderanforderungen dargestellt, um die Auswirkung einer Änderung ausgehend von der Validierung darzustellen. Das initiale Artefakt ist mit einer roten Markierung versehen. Die vorgesehene Änderung hingegen ist blau markiert. Durch diese beiden Artefakte wird der Änderungspfad definiert. Zusätzlich sind für die Verdeutlichung des Änderungspfades alle auf dem Pfad liegenden Artefakte türkis markiert, wobei diese Artefakte ebenfalls die direkt betroffenen Artefakte darstellen. Die restlichen Artefakte, welche hervorgehoben sind, stellen die indirekt betroffenen Artefakte dar. Ausgehend von diesen Artefakten können indirekte Auswirkungsmuster entstehen, falls das Artefakt von der Änderung betroffen ist. Für die Beschreibung des indirekten Musters kann die Auswirkung mit den Steckbriefen bestimmt werden.

In der Abbildung 5.3 ist ein Beispiel für ein ebenen-übergreifendes Auswirkungsmuster dargestellt. Im Vergleich zu den ebenen-spezifischen Auswirkungsmustern werden in dieser Darstellung weniger Entwicklungsartefakte betrachtet. Dies ist vorgesehen, da in den auszuwertenden Modellierungsansätzen für die Betrachtung über mehrere Ebenen nicht alle Entwicklungsartefakte betrachtet werden. Zusätzlich wird die Komplexität so verringert, dass die Muster durch den Nutzer leicht verarbeitbar sind und die Aussagen bestehen bleiben. Im Fall der Validierung ist zu beachten, dass die Stakeholderanforderung dasselbe Verhalten über die verschiedenen Ebenen aufweist wie eine Systemanforderung. Daraus lässt sich schließen, dass das bestimmte Verhalten der Systemanforderung auf die Stakeholderanforderung übertragbar ist. Für eine Aussage über das gesamte Auswirkungsverhalten muss somit eine Betrachtung über die Systemebenen hinaus durchgeführt und diese durch die Auswirkungsmuster auf einer Ebene ergänzt werden.





(C) ≙ betroffene Artefakte; (C) ≙ Initiales Artefakt; (C) ≙ Ziel; (C) ≙ Funktion;  
 (C) ≙ Stakeholderanforderung ; (C) ≙ Anwendungsfall; (C) ≙ Systemanforderung;  
 (C) ≙ logisches Element; (C) ≙ physisches Element; (C) ≙ Validierung  
 (C) ≙ Verifikation; (C) ≙ Parameter/Formel; (C)\* ≙ neues Element

Abbildung 5.2: Auswirkungsmuster für das Erstellen einer neuen Stakeholderanforderung

Die Beschreibung des Auswirkungsmusters über die vorgesehene Änderung und das initiale Element ist in Anlehnung an den Ansatz nach REDDI & MOON [RM09] ausgearbeitet. Hierbei sind Ergänzungen durch die initiale Ebene zu tätigen und die Art der Änderung auf die zu bearbeitenden Entwicklungsartefakte zu projizieren. Diese Informationen sind auch zusätzlich in der Abbildung 5.2 beschrieben. Dadurch soll die Suche erleichtert und die textuell wichtigen Daten im Steckbrief aufgeführt werden. Die vorgesehene Änderung legt im Rahmen dieser Ausarbeitung ebenfalls die Art der Änderung fest. Zusätzlich ist zu beachten, dass begründet durch die Notwendigkeit der Kombination der ebenen-spezifischen und ebenen-übergreifenden Auswirkungsmuster nur die charakteristischen Informationen übergeben werden. Das bedeutet, dass die ebenen-spezifischen Muster keine Information über die Systemebene wegen der beinhalten. Die ebenen-übergreifenden Auswirkungsmuster beinhalten hingegen keine Information bezüglich des initialen Elements.

Für die Einschätzung der möglichen Auswirkungen, die aus dem betrachteten Muster entstehen können, sind weitere Informationen nötig. In dem Steckbrief werden unter der vorgesehenen Änderung ebenfalls alle direkt betroffenen Elemente aufgelistet. Mit Hilfe

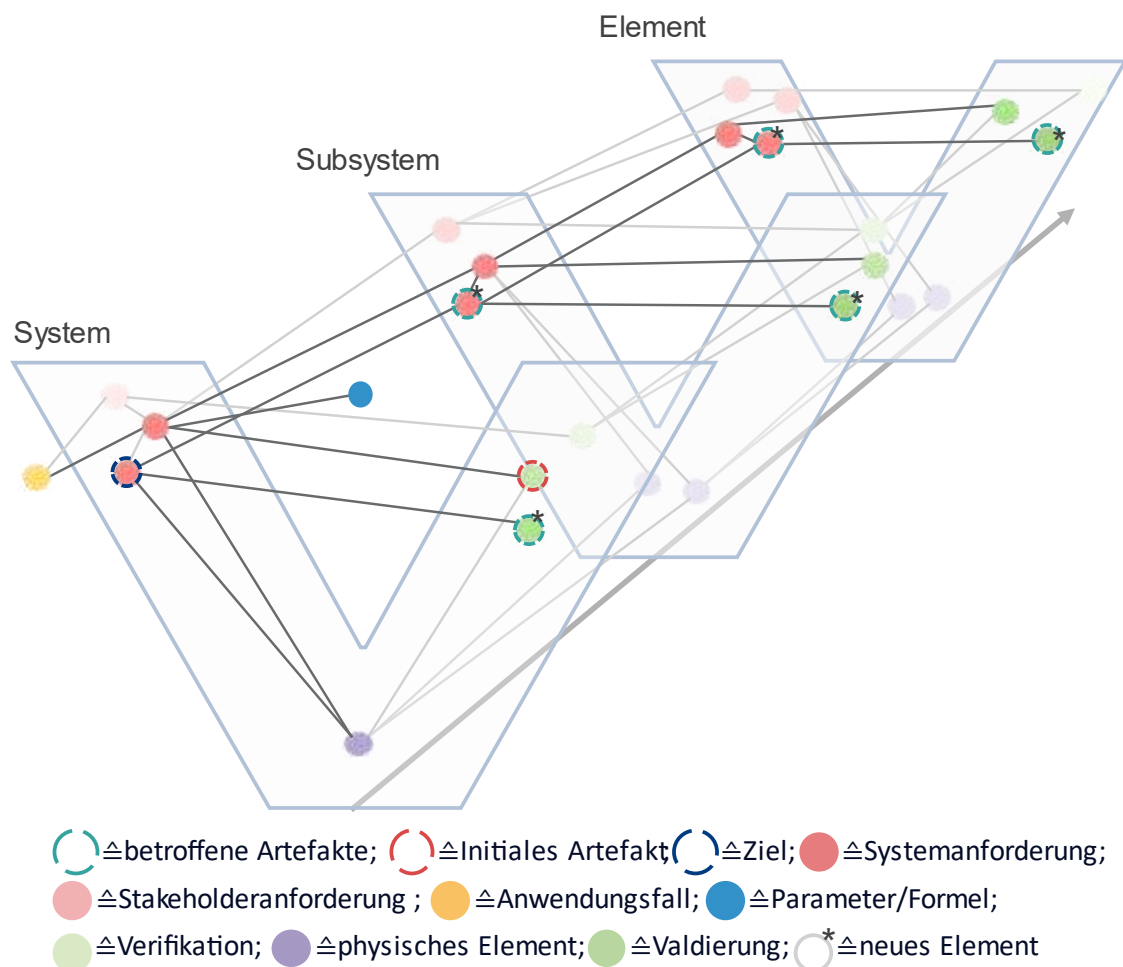


Abbildung 5.3: Ebenen-übergreifendes Auswirkungsmuster für die Erstellung einer neuen Anforderung

dieser Information kann der Anwender weitere Auswirkungsmuster bestimmen, die auftreten können. Diese Artefakte sind aus dem Grund direkt betroffen, da aufgrund der Verbindung über den Änderungspfad Informationen wie beispielsweise Attribute übergeben werden. Diese Attribute sind ebenfalls relevant für die verbundene Eigenschaftsabsicherung. Mit der Information der Anzahl direkter und indirekter Artefakte und die Information der Art dieser Artefakte lässt sich in Anlehnung an die Veröffentlichung nach ECKERT ET AL. [ECZ04] ein mögliches Änderungsverhalten zuordnen. Diese Information gibt dem Anwender Aufschluss darüber, ob es möglich ist, dass das Änderungsvolumen im Verlauf des Prozesses ansteigen kann.

Der Steckbrief enthält neben den bereits genannten Inhalten, ebenfalls Informationen über die Handlungsmaßnahmen und die zu beachtenden Randbedingungen. Die Verwendung dieser Handlungsmaßnahmen ist in dem Kapitel 5.4 detaillierter erklärt. Darüber hinaus wird jedem Auswirkungsmuster wie in der Abbildung 5.2 dargestellt eine Bewertung zugeteilt. Die Ermittlung der Bewertung des jeweiligen Musters sowie die Verwendung dieser Information ist in dem Kapitel 5.3 tiefgehend erläutert.

### **5.3 Bewertung der Kritikalität der Auswirkungen**

Die ausgearbeiteten Auswirkungsmuster sind hinsichtlich der Kritikalität der jeweiligen Änderung zu bewerten. Die Quantifizierung von Änderungsauswirkungen wird in der Regel anhand der Anzahl an Produktänderungen, der Durchlaufzeit der Produktänderung oder den anfallenden Kosten durchgeführt [BG21]. Dieses Vorgehen bedarf jedoch eines genauen Wissens über die jeweilige Unternehmung und die Unternehmensstruktur. Diese beeinflusst dabei maßgeblich, wieviel Ressourcen zu Verfügung steht, wenn Änderungen vorgesehen sind. Die eingebundenen Ressourcen führen dann im Umkehrschluss zu höheren Kosten. Im Rahmen dieser Ausarbeitung wird deshalb eine Bewertung auf generischer Ebene entwickelt. Diese orientiert sich an Informationen und Eigenschaften der Änderung, welche unabhängig von der durchgeführten Eigenschaftsabsicherung oder dem Unternehmen sind.

Das Ziel der Kritikalitätsbewertung besteht darin, eine Bewertungsmethode zu gestalten, die eine allgemeingültige Einordnung der Änderung zulässt. Diese soll den Anwender befähigen, mit wenigen Daten bereits eine Aussage über den möglichen Einfluss der Änderung zu treffen. Mit dieser Erkenntnis über den Einfluss der Änderung und dem zugehörigen Auswirkungsmuster wird dem Anwender eine Entscheidungsgrundlage für die Abwägung möglicher Lösungen gegeben.

Die Bewertung der Kritikalität orientiert sich dabei an dem Ansatz nach DEUBEL ET AL. [DCK+07]. Für die Bewertung wird somit vorgesehen, durch Multiplikation von Faktoren einen Kennwert zu ermitteln, von dem ausgehend die Einordnung der Änderung durchgeführt werden kann. Die Faktoren werden durch den Anwender anhand der Auswirkungsmuster und der Kenntnis über den Entwicklungsstand des Systems ermittelt.

Die Bewertung wird mit Hilfe der in Abbildung 5.4 dargestellten Tabelle durchgeführt. Zu Beginn sind die grundlegenden Daten durch den Anwender auszufüllen. Dies umfasst die ID des Testfalls, welcher zu der Änderung führt. Diese Eigenschaftsabsicherung wird in dem Bewertungsbogen als initiales Element bezeichnet. Dem initialen Element muss zusätzlich die Information übergeben werden, ob es sich bei der Eigenschaftsabsicherung um eine Validierung oder Verifizierung handelt. Neben der Information des initialen Elements wird die Information zu der initialen Ebene des Systems benötigt. Die initiale Ebene beschreibt die Systemebene, in der die Eigenschaftsabsicherung durchgeführt wird. Zusätzlich zu den genannten Informationen muss die vorgesehene Änderung angegeben werden, d. h., welches Element zu ändern ist. Zusätzlich ist die Information über die Art der Änderung zu übergeben. Die Art der Änderung umfasst dabei eine Neudefinition oder eine Änderung bestehender Artefakte. Eine Änderung bestehender Artefakte besteht aus dem Ändern von Informationen oder dem Erweitern der Informationen. Abschließend wird die zu ändernde Charakteristik beschrieben, beispielsweise der Transparenzwert einer Darstellung.

Initiales Artefakt	Initiale Ebene	Vorgesehene Änderung	Charakteristik	A	B	C	KN
TF1 (VA)	Subsystemebene	Use Case ändern	Ein-/Ausschalten der Darstellung	2	2	4	16
TF1 (VA)	Subsystemebene	Anforderung ändern	Transparenz der Darstellung	1	2	4	12

Abbildung 5.4: Template für die Berechnung der Änderungsklassifikationsnummer

Für die Bewertung müssen drei Faktoren bestimmt werden. Die Faktoren werden mit Hilfe der Auswirkungsmuster und dem Entwicklungsstand des Systems bestimmt. Für die Ermittlung der Bewertung eines Auswirkungsmusters muss dieses anhand der zu übergebenden Informationen identifiziert werden. Auf dem Steckbrief des jeweiligen Auswirkungsmusters ist der Wert für die Kritikalitätsbewertung hinterlegt. Der Anwender muss aus jeder Kategorie von Auswirkungsmustern eines bestimmen. Der Faktor A des Templates aus der Abbildung 5.4 wird durch das ebenen-spezifische Auswirkungsmuster festgelegt. Der Faktor B wird durch das ausgewählte ebenen-übergreifende Auswirkungsmuster bestimmt. Abschließend ist der Faktor C zu bestimmen, der durch den Entwicklungsstand festgelegt wird. Die Unterteilung richtet sich dabei nach der Unterteilung nach HEHENBERGER [Heh11]. Die Zuordnung wird wie in der Formel (5.1) durchgeführt.

$$C_i = \left\{ \begin{array}{l} 1, C_1 \triangleq \text{Konzept Prototyp} \\ 2, C_2 \triangleq \text{Gestalt Prototyp} \\ 3, C_3 \triangleq \text{Funktions Prototyp} \\ 4, C_3 \triangleq \text{Technischer Prototyp} \end{array} \right\} \quad (5.1)$$

Aus den erläuterten drei Faktoren lässt sich die Änderungsklassifikationsnummer (KN) berechnen. Anhand der Änderungsklassifikationsnummer wird die Art der Änderung bestimmt. In der Abbildung 5.5 ist das Schema für die Zuordnung der Änderungsart beschrieben. Dieses Schema ist dabei in Anlehnung an die Ausarbeitung nach ELLSEL ET AL. [ESS22] entwickelt. Die Einordnung bezweckt dabei nicht, einen konkreten Wert zu bestimmen, sondern dem Nutzer einen Eindruck über die mögliche Auswirkung der Änderung zu geben.

Art der Änderung	Einfache Änderung	Umfassende Änderung	Komplexe Änderung
Wert (KN)	1-4	5-16	17-64

Abbildung 5.5: Bewertungsschema für die Einordnung der Art der Änderung

## 5.4 Vorgehen für die Identifikation betroffener Systemartefakte

Um mit den identifizierten Auswirkungsmustern weiter zu verfahren, sind neben der Bewertung der Muster auch Handlungsmaßnahmen zu definieren, wie mit den Auswirkungen umgegangen werden soll. Im Rahmen der Ausarbeitung werden die Handlungsmaßnahmen in Form von Algorithmen entwickelt. Für die Durchführung der Algorithmen sind zusätzlich in den einzelnen Auswirkungsmustern Randbedingungen definiert. Diese bestimmen, was zusätzlich bei der Durchführung der Algorithmen zu beachten ist. Der Aufbau der Algorithmen ist mit Hilfe von Aktivitätsdiagrammen in SysML dargestellt. Insgesamt werden sechs Algorithmen für die Identifizierung von zu ändernden Entwicklungsartefakten erstellt.

In der Abbildung 5.6 ist exemplarisch der Algorithmus zur Identifikation betroffener Entwicklungsartefakte ausgehend von der Definition einer neuen Anforderung dargestellt. Der Aufbau der Algorithmen folgt dabei zu Beginn demselben Schema. Nach dem Start wird geprüft, ob das betrachtete Artefakt bereits mit derselben Änderung bearbeitet wurde. Wenn dies der Fall ist, wird der Vorgang abgebrochen, um Endlosschleifen zu vermeiden. Wenn das Artefakt nicht geändert wurde, wird begonnen und die Änderung eingesetzt. Im Anschluss muss die Änderung daraufhin geprüft werden, ob die verknüpften Stakeholderanforderungen durch die Anforderung noch abgebildet werden. Das Verfahren wird nur weitergeführt, wenn die Stakeholderanforderungen weiterhin abgebildet werden. Im nächsten Schritt wird geprüft, ob weitere Systemanforderungen verknüpft und von der Änderung betroffen sind. Für den Fall, dass die Anforderungen vorhanden sind, wird für diese Anforderung der Algorithmus A4 erneut durchgeführt. Während der Durchführung wird der zuvor ausgeführte Algorithmus pausiert und nach dem Abschluss der folgenden Algorithmen weiter ausgeführt. Sobald alle verknüpften Anforderungen identifiziert sind, wird überprüft, ob Anwendungsfälle verknüpft sind. Ist dies nicht der Fall, wird der Algorithmus an dieser Stelle beendet. Für den Fall, dass verknüpfte An-

wendungsfälle vorhanden sind, wird der Algorithmus A2 durchgeführt. Nachdem der Algorithmus abgeschlossen ist, wird ebenfalls der initiale Algorithmus für die Änderung der Anforderung beendet. Bei der Änderung der Anforderung ist darauf zu achten, dass bei jeder Änderung ebenfalls die zugehörige Eigenschaftsabsicherung angepasst wird. Bei der Überprüfung muss darauf geachtet werden, dass die angepasste Änderung noch mit der geplanten Methode der Eigenschaftsabsicherung und dem zugeordneten Prüfstand durchführbar ist.

Neben den Handlungsmaßnahmen sind, wie bereits genannt, ebenfalls Randbedingungen zu beachten. Diese Randbedingungen beinhalten mehrere Aspekte. Zum einen zeigen die Randbedingungen die notwendigen Verknüpfungen zu anderen Auswirkungsmustern auf, wodurch bedingt auch andere Algorithmen betrachtet werden müssen. Dies ist der Fall bei den Auswirkungsmustern der Validierung, weil bei den ebenen-übergreifenden Auswirkungsmustern die Stakeholderanforderungen nicht direkt betrachtet werden. Zum anderen werden durch die Randbedingungen Entwicklungsartefakte benannt, welche in der Betrachtung besonders wichtig sind. Dies umfasst im Fall der ebenen-spezifischen Auswirkungsmuster diejenigen Artefakte, welche das höchste Potential haben, bei der Änderung das Verhalten eines Vervielfältigers aufzuweisen. Bezüglich der ebenen-übergreifenden Auswirkungsmuster sind die abgeleiteten Artefakte von besonderer Bedeutung, da es sich in diesem Fall um eine höhere Anzahl an Artefakten handeln kann. Im Fall der Eltern-Artefakte ist zunächst nur ein Artefakt betroffen. Zu beachten ist in diesem Fall jedoch, dass bei einer Änderung des Eltern-Artefakts gegebenenfalls auch eines der von dem Punkt ausgehenden Kind-Artefakte betroffen sein kann.

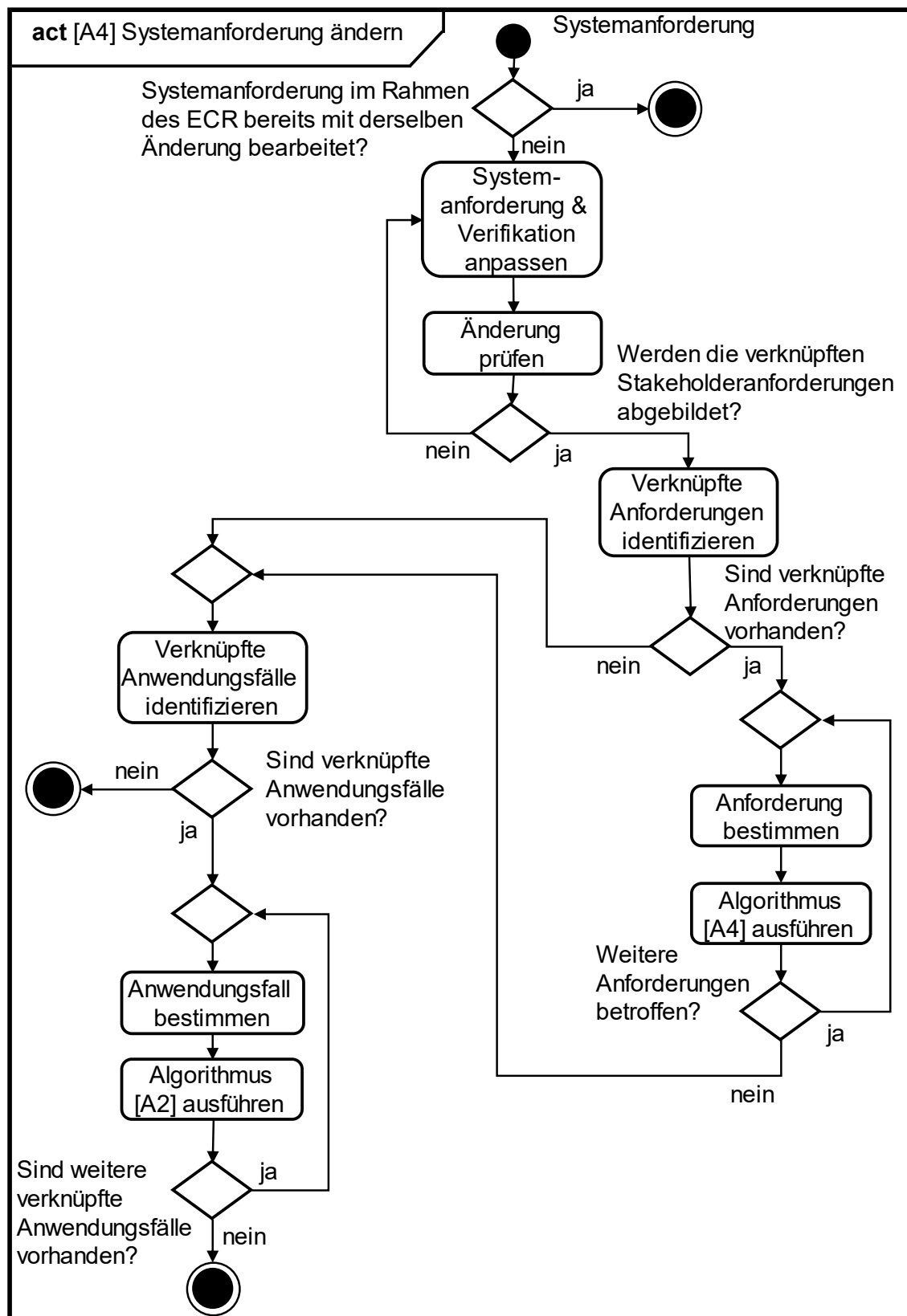


Abbildung 5.6: Algorithmus A4 dargestellt als SysML Aktivitätsdiagramm

## 5.5 Prototypische Umsetzung anhand von Testfällen

Für die Umsetzung anhand exemplarischer Testfälle wird als Praxisbeispiel der Rennwagen des UPBracing Teams verwendet. In dem UPBracing Team wird jährlich unter wechselnden Richtlinien ein neuer Rennwagen entwickelt. Somit ist die Änderung des Rennwagens aus dem Vorjahr ein essenzieller Teil der Entwicklungsaufgabe, sowie die Bestimmung möglicher Auswirkungen der Änderung. Für das Praxisbeispiel ist ein Szenario ausgearbeitet, welches ebenfalls durch ein exemplarisches Systemmodell beschrieben ist. Das Systemmodell ist in der Abbildung A - 30 dargestellt. Das Szenario beschreibt einen Feldversuch, in dem festgestellt wird, dass die Motortemperatur des Rennwagens zu hoch ist. Eine zu hohe Motortemperatur führt zu einem Verlust der Motorleistung oder einem Ausfall während des Rennens.

In dem Vorgehen wird zu Beginn das in der Abbildung 5.4 dargestellte Template ausgefüllt. Die Reihenfolge ist so festgelegt, dass zu Beginn das initiale Artefakt bestimmt wird. In dem Beispiel handelt es sich um einen Testfall während der Verifikation, dem eine Anforderung an die Motortemperatur zugeordnet ist. Es ergibt sich als initiales Artefakt der Testfall der Motortemperatur. Als nächstes wird die Systemebene bestimmt, auf der die Anforderung definiert ist. In diesem Beispiel handelt es sich um die Subsystemebene. Damit die Motortemperatur beeinflusst werden kann, stellt die Änderung der Kühlleistung des Kühlers einen möglichen Lösungsansatz dar. Somit ist die vorgesehene Änderung eine geänderte Anforderung. Die zugehörige Charakteristik ist die Kühlleistung. Damit der Testfall durchgeführt werden kann, benötigt es einen Funktionsprototyp. Somit ergibt sich für den Faktor C der Wert vier. Als nächstes müssen die zugehörigen Auswirkungsmuster bestimmt werden. Für die Bestimmung ist die passende Bezeichnung zu ermitteln. Die Informationen ergeben, dass die beiden Auswirkungsmuster die Bezeichnungen „CL-SU-CR“ und „LS-VE-CR“ aufweisen. Mit Hilfe der Bezeichnung werden die Auswirkungsmuster ausgewählt und die zugehörige Bewertung in die Tabelle eingetragen. Die drei Faktoren ergeben eine Änderungsklassifikationsnummer mit dem Wert 24. Gemäß der Skala handelt es sich um eine komplexe Änderung

In dem exemplarischen Vorgehen wird die dargestellte Änderung durchgeführt. Für den Fall, dass mehrere mögliche Lösungen vorhanden sind, kann bereits eine Entscheidung getroffen werden, welche Änderung durchgeführt wird. Andererseits ist es möglich, für eine Auswahl von Änderungen weitere betroffene Artefakte zu bestimmen, um eine genauere Aussage über das Änderungsvolumen zu erhalten. Für die Identifikation weiterer möglicherweise betroffener Artefakte wird in diesem Beispiel der Algorithmus [A4] durchgeführt. Es ergibt sich bei dem Durchführen des Algorithmus, dass die Kühlleistung durch den Durchmesser des Kühlungsflüters beeinflusst wird. Folglich sind die Anforderungen, die zugeordneten Eigenschaftsabsicherungen und die physischen Elemente zu ändern.





## 6 Validierung

Das sechste Kapitel beschreibt die Validierung der Auswirkungsmuster mit der zugehörigen Bewertung sowie die Handlungsmaßnahmen. Betrachtet werden die Planungsphase, die Durchführung und die ermittelten Ergebnisse. Das Unterkapitel der Planung umfasst dabei die Anzahl der Probanden und Ausführungen, die für die Vorbereitungen durchzuführen sind. In der Durchführung wird der Ablauf der Validierung erklärt und beschrieben welche Randbedingungen hierbei eingehalten werden. In dem Unterkapitel der Ergebnisse werden die Erkenntnisse aus der Validierung diskutiert.

### 6.1 Planung der Validierungsgespräche

Im Folgenden wird die Planung der Validierung detailliert erläutert. Für die Planung der Validierungsgespräche müssen die nötigen Probanden festgelegt werden. Zusätzlich ist ein Ablaufplan für die Validierung auszuarbeiten. Aufbauend auf dem Ablaufplan werden die benötigten Materialien zusammengestellt. Die Planung umfasst eine thematische Einführung, die Anwendungsbeispiele und die Unterlagen für die Auswertung.

Zur Validierung der Handlungsmaßnahme sind Experteninterviews vorgesehen. Die Experteninterviews werden mit insgesamt sechs Probanden durchgeführt. In der Abbildung 6.1 ist eine Übersicht der Probanden mit dem zugehörigen Fachbereich dargestellt. Unter den Probanden befinden sich vier Personen, welche als wissenschaftliche Mitarbeiter in zwei Fachbereichen tätig sind. Zusätzlich sind zwei Studenten in die Experteninterviews eingebunden. Somit ergibt sich, dass ein Teil der Probanden einen höheren Erfahrungswert hat und die weniger erfahrenen Probanden eine genauere theoretische Einführung in den Themenbereich benötigen.

ID	Rolle	Fachbereich
Proband #1	Wissenschaftlicher Mitarbeiter	Produktentstehung
Proband #2	Wissenschaftlicher Mitarbeiter	Produktentstehung
Proband #3	Student	Produktentstehung
Proband #4	Wissenschaftlicher Mitarbeiter	Produktentstehung
Proband #5	Student	-
Proband #6	Wissenschaftlicher Mitarbeiter	Leichtbau im Automobilbau

Abbildung 6.1: Auflistung der teilnehmenden Probanden

Der Ablauf des Experteninterviews ist in der Abbildung 6.2 dargestellt. Zu Beginn des Validierungsgesprächs erhalten die Probanden eine Einführung in die grundlegenden Theorien der Auswirkungsmuster und deren Bewertung. Aufbauend darauf werden die Auswirkungsmuster, die Bewertung und der Handlungsmaßnahmen erläutert. Zusätzlich

wird die Verwendung dieser Hilfsmittel beispielhaft vorgestellt. Das Vorgehen zielt darauf ab ein einheitliches Wissen bei den Probanden zu erzeugen, um die Vergleichbarkeit der Aussagen zu gewährleisten. Im Anschluss an die Erklärung der Theorien und Hilfsmittel wenden die Probanden die Hilfsmittel auf entwickelte Szenarien und ein Systemmodell an. Abschließend wird die Auswertung mit Hilfe des SUS-Fragebogens nach BROOKE [Bo96] und eines an Leitfragen orientierten Gesprächs durchgeführt.

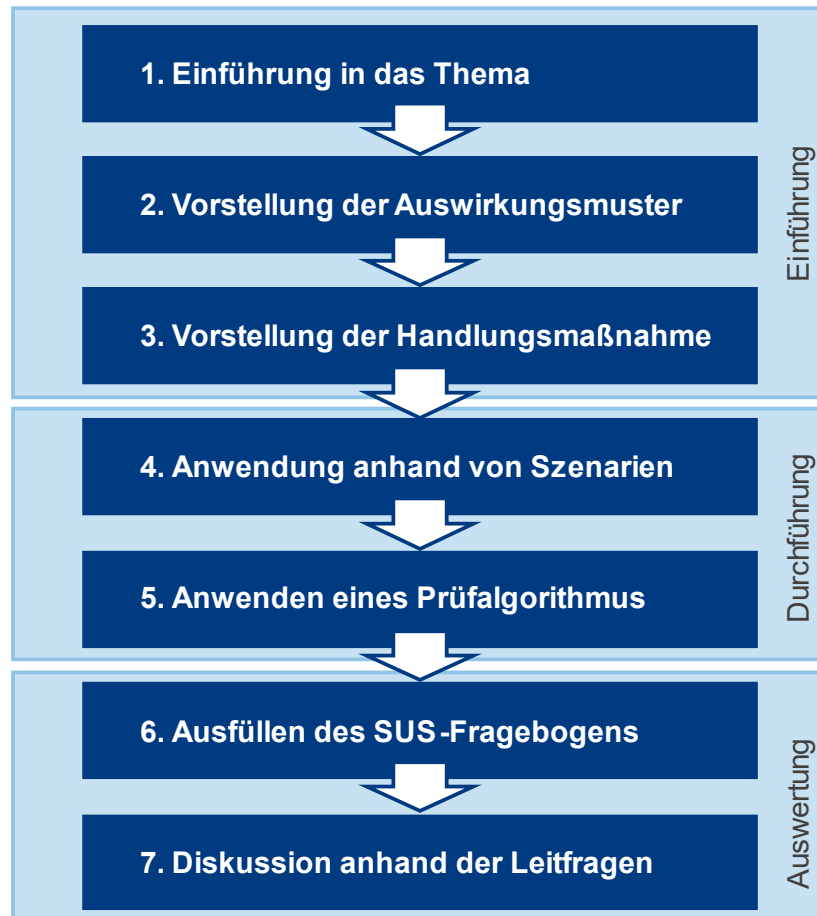


Abbildung 6.2: Ablauf der Experteninterviews

Für die Durchführung der Validierung sind drei Szenarien zu erarbeiten. Diese Szenarien bilden verschiedene Typen von Änderungen ab. Zwei Szenarien stellen Situationen dar, welche im Projektverlauf eines Forschungsprojektes und des UPBracing Teams auftreten. Das dritte Szenario bildet eine fiktive Situation anhand eines Demonstrators des Lehrstuhls ab. Anhand der erstellten Szenarien wenden die Probanden die Auswirkungsmuster mit der zugehörigen Bewertung an. Neben den Szenarien ist ebenfalls ein Systemmodell zu erarbeiten. Dieses Modell umfasst die Anforderungen an das Szenario des UPBracing Teams. Die Anforderungen beschreiben das Kühlungssystem des Rennwagens. Ihnen sind die zugehörigen Testfälle und die Systemelemente zugeordnet. Für dieses Systemmodell führen die Probanden die erarbeiteten Prüfalgorithmen durch.

## 6.2 Durchführung der Validierungsgespräche

Der Ablauf der Experteninterviews sieht eine Umsetzung von Verbesserungsvorschlägen nach den ersten drei Gesprächen vor. Für die Gespräche ist jeweils ein Zeitfenster von 1,5 Stunden vorgesehen. In den ersten zwanzig Minuten werden die Theorie und die Hilfsmittel erläutert, denen sich eine Besprechung aufkommender Fragen anschließt. Für die Durchführung sind weitere vierzig Minuten angesetzt. Die letzten dreißig Minuten sind für die Auswertung eingeplant. Es zeigt sich, dass die Zeit bei den erfahrenen Probanden nicht ausgeschöpft wird.

Bei der Durchführung der ersten Experteninterviews fällt auf, dass bei dem Ablauf Unklarheiten bezüglich des Vorgehens bestehen. Diese lassen sich darauf zurückführen, dass durch den Wechsel der Dateien die Übersicht verloren geht. Zusätzlich sind Unsicherheiten bei der Bewertung aufgetreten. Daneben zeigt sich durch die Szenarien, dass die gegebenen Auswirkungsmuster nicht vollständig sind. Die Notwendigkeit dieser Auswirkungsmuster hat sich durch den individuellen Lösungsansatz der Probanden bestätigt. Somit ergibt sich aus den ersten drei Experteninterviews, dass ein Handlungsbedarf hinsichtlich der Optimierung der Auswirkungsmuster und deren Bewertung besteht. Die Durchführung der Optimierungen ist in dem Kapitel 6.3.3 dargestellt.

Der generelle Ablauf der Experteninterviews sieht vor, dass den Probanden neben den Auswirkungsmustern weitere Unterlagen als Hilfestellung mitgegeben werden. Diese umfassen die Einführungspräsentation und ein Template, in dem die grundlegenden Informationen für das Ausfüllen der Tabelle übergeben sind. Somit soll dem Probanden die Möglichkeit eröffnet werden auf alle nötigen Informationen zuzugreifen.

Das Vorgehen zur Anwendung der Auswirkungsmuster ist so aufgebaut, dass die Probanden zu Beginn die Szenarien analysieren. Die Szenarien werden den Probanden samt der Aufgabenstellung in Form eines Handouts übergeben. Nach der Analyse wird das Template ausgefüllt, welches in Form der Excel-Datei zu Verfügung steht. Bei dem Eintragen der abzuleitenden Informationen ist es bei dem zweiten, vierten und fünften Probanden zu Schwierigkeiten gekommen. Beim Ausfüllen des Templates wird das Element, welches durch den Testfall geprüft wird, als initiales Element ausgewählt. Durch das Ausfüllen des Templates soll den Probanden alle relevanten Informationen wie beispielsweise das initiale Element und die initiale Ebene verdeutlicht werden.

Aus diesen Informationen kann dann die Auswahl des geeigneten Auswirkungsmusters stattfinden. Bei der Auswahl der Muster kommt es bei den Probanden Nummer drei und vier zu Schwierigkeiten aufgrund der Klassifikation. Hierzu wird durch die Probanden erläutert, dass eine Darstellung der Klassifizierung der Muster hilfreich sei. Die Auswirkungsmuster liegen in Form einer PDF-Datei vor. Aus den Auswirkungsmustern kann dann die Bewertung in das Template übertragen werden. Abschließend sind der Entwicklungsstand zu bestimmen und die Klassifikationsnummer zu errechnen“. In den Experteninterviews der ersten drei Probanden ist zusätzlich die Ebene, auf der die Änderung

stattfindet, in die Berechnung der Klassifikationsnummer einzubeziehen. Diese Anforderung wird in den letzten drei Experteninterviews nicht aufrechterhalten, um eine doppelte Bewertung der Ebenen zu vermeiden. Diese Entscheidungen sind in dem Kapitel 6.3.3 genauer erläutert. Auf Basis der resultierenden Bewertung der Auswirkungsmuster kann dann das Muster mit der geringsten Auswirkung ausgewählt werden. Die nötigen Handlungsmaßnahmen und Randbedingungen, die aus der Wahl resultieren, werden aus dem Muster abgelesen. Im Fall des letzten Szenarios ist die Handlungsmaßnahme zusätzlich auf ein Systemmodell anzuwenden, wobei Schwierigkeiten auftreten. Diese sind durch den offenen Lösungsraum zu erklären, da die Probanden Nummer sechs und vier andere Lösungen umsetzen. Diese sind in dem Modell nicht vorgesehen, sind jedoch in dem Verlauf zu ergänzen.

## **6.3 Ergebnisse der Validierung**

In diesem Kapitel werden die Ergebnisse der Experteninterviews vorgestellt und diskutiert. Zur Auswertung der Validierung durch die Experteninterviews dienen zwei Fragebögen. Die Fragebögen werden in den folgenden zwei Kapiteln erläutert. Die aus den Experteninterviews abgeleiteten Verbesserungsmaßnahmen sind im dritten Unterkapitel dargestellt. Abschließend werden die in Kapitel 1.2 aufgestellten Forschungsfragen diskutiert.

### **6.3.1 Auswertung des Usability-Fragebogens**

Der Auswertung der Handhabbarkeit der Auswirkungsmuster, der Bewertung und der abzuleitenden Handlungsmaßnahmen ist der SUS-Usability Score zu Grunde gelegt. Der SUS-Usability-Fragebogen nach BROOKE [Bo96] wird hinsichtlich der Handlungsmaßnahmen angepasst. Dies ist darin begründet, dass das Verfahren zur Identifikation und Bewertung der Auswirkungsmuster manuell durchgeführt wird und kein Userinterface aufweist. Die Probanden tragen in dem Fragebogen alternierend den Wert eins und fünf ein im Fall der vollständigen Zustimmung ein. Bei einer Ablehnung werden die Werte entgegengesetzt eingetragen. Aus den ermittelten Summen wird dann der „Usability-Score“ in Prozent gebildet.

Die Ergebnisse der Auswertung hinsichtlich der Handhabbarkeit sind in der Abbildung 6.3 dargestellt. Es ergibt sich ein durchschnittlicher „Usability-Score“ von 76,25 Prozent. Dies stellt ein gutes Ergebnis der Beurteilung der Handhabbarkeit dar. Es zeigt sich trotzdem, dass noch Potential für eine Verbesserung besteht. Auffällig ist, dass im Fall des fünften Probanden ein Wert von 35 Prozent vergeben wird. Hieraus lässt sich ableiten, dass das Vorgehen Schwierigkeiten für Personen mit weniger Vorwissen aufweist. Zur Verbesserung der Handhabbarkeit für diese Zielgruppe ist es möglich, eine Hilfestellung in Form von Beispielen zu geben oder bei einer späteren Implementierung eines Tools die Nutzeroberfläche mit Hilfestellungen zu versehen.

Von den Probanden wird herausgestellt, dass Verlinkungen zu den PDF-Dateien die Anwendung der Excel-Tabelle erleichtern würden. Einschränkungen bei der Eingabe mit Hilfe eines Drop-Down-Menüs werden ebenfalls für sinnvoll gehalten, um Anwendern die Entscheidung bei der Eingabe zu erleichtern. Zusätzlich ist es möglich, Informationen aus den Auswirkungsmustern in die Excel-Tabelle zu übergeben. Diese bestehen in der Bezeichnung der Klassifikation und der Angabe der anzuwendenden Algorithmen. Zusätzlich wird von den Probanden genannt, dass die Handhabung durch die unterschiedlichen Dateien erschwert werden.

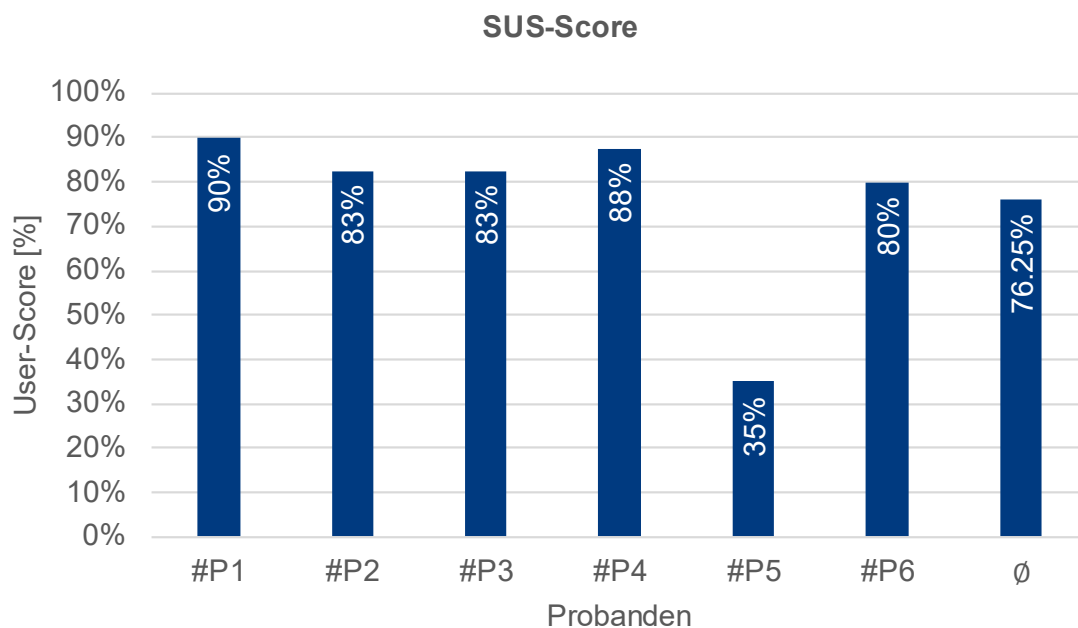


Abbildung 6.3: Ergebnisse der Befragung der Probanden mit dem SUS-Fragebogen

### 6.3.2 Auswertung der Leitfragen

Zur Beurteilung der Handhabbarkeit der Auswirkungsmuster und deren Bewertung wird eine Befragung anhand von Leitfragen durchgeführt. Insgesamt werden neun Leitfragen gestellt, die als Gesprächsgrundlage dienen. Die ersten acht Leitfragen beziehen sich auf fünf Erfolgsfaktoren. Die neunte Leitfrage betrachtet die mögliche Potenziale, die sich aus den Auswirkungsmustern ergeben. Aus der Beantwortung der Leitfragen wird dann eine Aussage über die Erfüllung der Erfolgsfaktoren hergeleitet. Die Erkenntnisse aus der Befragung sind im Folgenden dargestellt.

**Leitfrage 1:** Haben Sie die Verwendung der Auswirkungsmuster und die dazugehörige Bewertung der Kritikalität als hilfreich empfunden?

Die erste Frage bezieht sich auf den Nutzen, den die Auswirkungsmuster für die Probanden während der Durchführungsmuster aufweisen. Die Nutzung der Auswirkungsmuster soll dabei dem Probanden helfen, eine Ausgangslage für die Problemstellung zu finden.

Alle Probanden sehen die Auswirkungsmuster für die Lösung des Problems als hilfreich an. Die manuelle Nutzung der Auswirkungsmuster schätzen zwei Probanden als einen Mangel ein, wobei Proband fünf auf eine digitale Lösung für die zukünftige Nutzung hinweist. Proband sechs hingegen stellt heraus, dass die Verfahrensweise in einem komplexen System hilfreicher ist als in den Szenarien der Validierung. Die übrigen Probanden sehen die Hilfe darin, dass durch die erzeugte Ausgangsposition die weitere Verfahrensweise erleichtert wird. Dies wird durch die Bewertung unterstützt, indem ein Richtwert als grobe Abschätzung vorhanden ist. Zusätzlich stellt Proband vier heraus, dass somit das komplexe Problem übersichtlicher wird.

**Leitfrage 2:** Konnten Sie mit Hilfe der Unterlagen das Vorgehen verstehen?

Die zweite Leitfrage will ermitteln, ob weitere Hilfestellungen für die Anwendung der Szenarien benötigt werden oder ob die mitgegebenen Materialien für eine Bewertung ausreichen. Die Probanden stellen heraus, dass die Anwendung durch ein Beispielszenario erleichtert wird. Hierzu muss die Tabelle beispielhaft ausgefüllt und mit Kommentaren versehen werden. Zusätzlich wird durch den zweiten Probanden dargelegt, dass die Integration eines Drop-Down-Menüs in der Excel-Tabelle die Übersichtlichkeit fördern würde. Dem fünften Probanden wurde die Übersichtlichkeit durch das Wechseln zwischen den PDF-Dateien erschwert.

**Leitfrage 3:** Sehen Sie die Handlungsmaßnahmen als geeignetes Hilfsmittel für die Identifikation und Bewertung der möglichen Auswirkungen?

Die dritte Leitfrage rückt in den Fokus, ob die Handlungsmaßnahme generell als Hilfsmittel aufgefasst wird. Die Handlungsmaßnahme umfasst die Auswirkungsmuster, die Bewertung und die Vorgehensweise zur Identifikation betroffener Systemelemente. Proband vier stellt heraus, dass durch die Auswirkungsmuster der Entscheidung ein gebündeltes Expertenwissen zugrunde gelegt wird. Dies ermöglicht ein grobes Verständnis für die möglichen Auswirkungen, welche sich durch die Propagationseffekte ergeben. Die Fortpflanzung der Änderung lässt sich im weiteren Verlauf nicht verallgemeinern, weshalb nur ein Startpunkt übergeben wird. Dies stellt ebenfalls der Proband zwei heraus. Zusätzlich benennt Proband eins den Vorteil, dass durch die Anwendung nicht zu früh in Lösungen gedacht wird und vorerst das Ausmaß der Auswirkungen verdeutlicht wird. Somit werden Lösungsansätze nicht zu früh ausgeschlossen. Proband fünf zeigt den Nutzen auf, den ein automatisiertes Tool als Hilfestellung für ein Unternehmen haben kann. Proband sechs stellt heraus, dass für die abgeleiteten Vorgehensweisen weitere Kriterien beachtet werden müssen, welche je nach Kontext variieren.

**Leitfrage 4:** Wie würden Sie den Mehrwert für die Identifikation und Bewertung von Änderungen mit Hilfe der Auswirkungsmuster einschätzen?

Die vierte Leitfrage stellt heraus, welcher Mehrwert aus der Anwendung der Auswirkungsmuster und der Bewertung resultiert. Der Mehrwert wird dabei auf die Verfahrensweise in dem Änderungsprozess bezogen. Das Verfahren ist gemäß der Auffassung des

fünften Probanden zu kleinschrittig und damit umständlich. Proband zwei nennt hingegen, dass eine Lösung geschaffen wird, welche wiederkehrende Änderungen bewertbar macht und somit im Alltagsgeschäft viele Anfragen abgedeckt. Zudem wird laut dem ersten Probanden ein Überblick geschaffen, wie sich die Änderung über mehrere Systemebenen verhält. Diese Übersicht fördert laut dem vierten Probanden die Transparenz und das Verständnis für das System, welches entwickelt wird. Proband drei stellt heraus, dass durch den Ablauf und der damit einhergehenden Routine langfristig die Produktivität und Effizienz gesteigert wird. Der zweite Proband sieht in den Auswirkungsmustern das Potenzial, diese unternehmensspezifisch zu erweitern und an die jeweiligen Bedürfnisse anzupassen. Sie ermöglichen laut dem Probanden sechs eine Bewertung auf der Basis gebündelten Wissens über die zugrundeliegenden Theorien.

**Leitfrage 5:** Würden Sie die Auswirkungsmuster und die Kritikalitätsbewertung für Änderungen verwenden?

Die fünfte Frage stellt heraus, ob die Probanden selbst für die Bewertung von Änderungen die erarbeiteten Auswirkungsmuster nutzen würden. Der Proband fünf würde die Muster selbst nicht nutzen und zieht vor, die Lösungen direkt zu testen. Die übrigen Probanden würden jedoch alle die Auswirkungsmuster nutzen. Proband sechs erläutert, dass die Nutzung leicht umzusetzen sei und schnell gehe. Somit werde langfristig Arbeitsvolumen gespart.

**Leitfrage 6:** Ist die ermittelte Kritikalität nachvollziehbar?

Die sechste Frage behandelt die Nachvollziehbarkeit der Bewertung. Diese setzt sich im Kern aus zwei Auswirkungsmustern und der Information über den Entwicklungsstand zusammen. Proband fünf nimmt die Bewertung als logisch wahr, obgleich viel Interpretationsspielraum gegeben sei. Der dritte und der sechste Proband beschreiben, dass die Bewertung nicht immer direkt nachvollziehbar sei. Dies bezieht sich auf die Einzelbewertung der Auswirkungsmuster. Drei Probanden können die Bewertung nachvollziehen, stellen jedoch heraus, dass Änderungspotenzial darin bestehe unternehmensspezifische Änderungen vorzusehen und den Entwicklungsstand über die Prototypenphase hinaus zu betrachten.

**Leitfrage 7:** Stimmt die Skala mit Ihrem technischen Einschätzungsvermögen überein?

Die siebte Leitfrage gleicht die errechnete Bewertung mit dem individuellen technischen Verständnis der Probanden ab. Proband fünf hält den nötigen Prototyp für nicht eindeutig bestimmbar und sieht darin eine Abweichung von seinem technischen Einschätzungsvermögen. Der Proband drei und sechs schätzen die Einteilung im Detail anders ein. Für drei der sechs Probanden stimmt die Bewertung mit dem eigenen technischen Verständnis überein, wobei durch den vierten Probanden herausgestellt wird, dass eine Änderung der Skala je nach Bedarf vorgesehen werden sollte.

**Leitfrage 8:** Sehen Sie Optimierungspotential in einem Teil der Handlungsmaßnahme oder im Vorgehen an sich?



Die achte Leitfrage dient der Identifizierung möglicher Optimierungspotenziale im Vorgehen und bei den Handlungsmaßnahmen. Der sechste Proband sieht kein direktes Optimierungspotential. Zwei der sechs Probanden sehen eine Verknüpfung der Dokumente als mögliches Optimierungspotential. Eine Optimierung kann laut Probanden eins und vier in der Anzeige der notwendigen Algorithmen in der Excel-Tabelle bestehen. Proband zwei sieht eine Teilautomatisierung in Form eines Excel-Tools als mögliche Optimierung an, um das Verfahren ergiebiger zu gestalten.

**Leitfrage 9:** Wie würden Sie die die Entwicklungspotentiale beschreiben, welche sich aus Ihrer Sicht für die Verwendung der Auswirkungsmuster ergeben?

Die neunte Leitfrage stellt darauf ab, wie das Potenzial einer Weiterentwicklung der Auswirkungsmuster eingeschätzt wird. Der erste Proband erläutert, dass im Verlauf des Änderungsprozesses Daten zurückgeführt werden könnten. Mit Hilfe dieser Daten wäre dann ein Abgleich zwischen dem angenommenen und dem in der Durchführung benötigten Änderungsvolumen möglich. Als mögliche Weiterentwicklung sehen vier der sechs Probanden, dass ein Tool entwickelt werden könnte. Dieses Tool würde die Übersichtlichkeit steigern und die Möglichkeit bieten, Schnittstellen zu bestehenden Anwendungen im Bereich es MBSE herzustellen. Der zweite Proband sieht eine weitere Untergliederung der Auswirkungen als mögliches Entwicklungspotential. Diese ermögliche eine Betrachtung komplexerer Muster. Zusätzlich kann laut des vierten Probanden eine Betrachtung hinsichtlich Qualität, Kosten und Zeit durchgeführt werden. Somit ergebe sich eine neue Perspektive, auf deren Basis dann die Änderung ausgewählt werde.

Aus den Ergebnissen der Leitfragen kann die Erfüllung der Erfolgsfaktoren bestimmt werden. Der erste Erfolgsfaktor wird durch die ersten beiden Leitfragen bestimmt und beschreibt die Anwendbarkeit. Aus den Ergebnissen ergibt sich, dass die Anwendbarkeit gegeben ist und die Muster hilfreich sind. Doch zeigt sich für die Verfahrensweise mit den Mustern, dass hier Beispiele als Einleitung nötig sind, um das Vorgehen zu verstehen. Der zweite Erfolgsfaktor, der Praxisbezug, wird durch die dritte Leitfrage bestimmt. Es lässt sich zusammenfassen, dass durch die Auswirkungsmuster eine gute Ausgangslage geschaffen wird. Der geschaffene Mehrwert ist der dritte Erfolgsfaktor und ist ebenfalls erfüllt. Dieser Faktor wird durch die vierte Frage bestimmt. Der Mehrwert besteht in der Bündelung des Expertenwissens und der dadurch entstehenden Transparenz zur Abschätzung der Änderung über mehrere Ebenen. Der vierte Erfolgsfaktor ist die Kritikalität, also die zugrundeliegende Bewertung für die Auswirkungsmuster, und wird durch die Leitfragen sechs und sieben bewertet. Dieser Erfolgsfaktor ist teilweise erfüllt, da die Auswirkung zwar nachvollziehbar ist, jedoch durch Rückführung von Daten aus Praxisanwendungen optimiert werden kann. Die Leitfragen fünf und acht beschreiben den Erfolgsfaktor Praxisbezug. Dieser Faktor ist erfüllt, da die Mehrheit der Probanden die Auswirkungsmuster verwenden würden. Zusätzlich sehen die Probanden in der Verwendung eine Produktivitätssteigerung für das Alltagsgeschäft.

### 6.3.3 Umsetzung von Verbesserungsvorschlägen

In diesem Abschnitt wird die Umsetzung von Verbesserungsvorschlägen erläutert, die sich aus der Durchführung der Experteninterviews ergeben haben. Die Verbesserungen beziehen sich auf die Handhabung der Auswirkungsmuster, der Handlungsmaßnahmen und der Bewertung. Die jeweiligen Verbesserungsvorschläge sowie deren Umsetzung werden erläutert.

Aus den ersten drei Experteninterviews ergibt sich, dass die Reihenfolge der Handlung beim Umgang mit den Auswirkungsmustern und Bewertungsmaßnahmen angepasst werden sollte. Das bedeutet, die Suche der Auswirkungsmuster sollte erst dann ausgeführt werden, wenn alle Randbedingungen festgelegt sind. Somit sieht die Arbeitsanweisung vor, dass der Anwender zu Beginn festlegt, welches initiale Element und welche Ebene vorliegen. Ebenfalls muss definiert werden, welcher Entwicklungsstand verwendet und welche Änderung vorgesehen ist. Dann ist es möglich, bereits die Bewertung des Faktors C zu bestimmen. Aufbauend darauf muss durch den Anwender auf Basis der eingetragenen Informationen noch die Bezeichnung des Auswirkungsmusters ermittelt werden. Dies ermöglicht eine vereinfachte Identifikation des Auswirkungsmusters. Dieses Vorgehen ist dann in einer beispielhaften Umsetzung vorzusehen.

Ein weiteres Verbesserungspotential wird in der Auswahl der Auswirkungsmuster identifiziert. Es wird herausgestellt, dass nicht alle möglichen Fälle dargestellt sind. Dies umfasst Anwendungsfalländerungen und Attributänderungen ausgehend von der Validierung. Die dazugehörigen Auswirkungsmuster sind zu ergänzen. Zusätzlich sind die Betrachtungen der ebenen-übergreifenden Auswirkungsmuster zu ergänzen. Im Verlauf der restlichen Experteninterviews werden diese Auswirkungsmuster mit zu Verfügung gestellt.

Das dritte Verbesserungspotential ist in der Bewertung zu identifizieren. Die Bewertung innerhalb der ersten drei Experteninterviews sieht einen zusätzlichen Faktor D vor, durch den die Ebenen mit bewertet werden sollen. Hieraus ergibt sich allerdings eine Inkonsistenz, da dem Anwender verschiedene Kombinationen der Bewertungskriterien ermöglicht werden, die widersprüchlich sind. Dies umfasst beispielsweise die Zuordnung einer Verifikation zu der Stakeholder-Ebene. Zusätzlich wird nicht beachtet, dass durch die ebenen-übergreifenden Auswirkungsmuster eine Bewertung hinsichtlich der Auswirkung bezüglich der Ebene vorgenommen wird. Ebenfalls wird der Einfluss der Stakeholder durch die Auswahl der Validierung betrachtet. Das bedeutet, dass durch die doppelte Betrachtung der Faktor Ebene verstärkt in die Rechnung einfließt. Somit wird der Faktor D von der Bewertung ausgenommen und die Skala dementsprechend angepasst. Die Optionen zur Einordnung der Änderungsart werden auf drei beschränkt, da der Unterschied zwischen einer geringfügigen und einer einfachen Änderung zu klein ist.

### 6.3.4 Beantwortung der Forschungsfragen

Für die Zielsetzung des Forschungsansatzes sind zu Beginn vier Forschungsfragen aufgestellt worden. Im Folgenden wird die Erfüllung der Forschungsfragen geprüft. Hierzu werden die zugeordneten Umsetzungen genauer erläutert.

**Forschungsfrage 1:** Sind Auswirkungsmuster basierend auf Änderungen in der Eigenschaftsabsicherung zu identifizieren?

In der Ausarbeitung des Forschungsansatzes sind mit Hilfe der Betrachtung von theoretischen und praktischen Methoden zur Modellierung zwanzig Auswirkungsmuster zu identifizieren. Ebenfalls ist im Verlauf der Validierung festzustellen, dass potenziell weitere Auswirkungsmuster identifizierbar sind. Diese Auswirkungsmuster teilen sich auf in ebenen-übergreifende und ebenen-spezifische Auswirkungsmuster. Der Unterschied besteht darin, dass die ebenen-spezifischen Auswirkungsmuster auf jeder Ebene ein ähnliches Verhalten aufweisen. In diesen Auswirkungsmustern wird die Ausbreitung der Änderung über alle Entwicklungsartefakte des RFLPV<sup>2</sup>-Ansatzes dargestellt. Die ebenen-übergreifenden Auswirkungsmuster hingegen zeigen die Abhängigkeiten der Entwicklungsartefakte zwischen den einzelnen Systemebenen auf. Daraus lässt sich das Verhalten der Änderungsfortpflanzung für die jeweiligen Fälle auf die Entwicklungsartefakte projizieren. In der Darstellung der ebenen-übergreifenden Auswirkungsmuster wird sich auf die Entwicklungsartefakte der Anforderung, der physischen Elemente und der Verifikation beschränkt. Darüber hinaus werden Parameter und Anwendungsfälle betrachtet. Für eine Aussage über das gesamte Änderungsverhalten benötigt es somit zwei Auswirkungsmuster, wobei eins aus jeder Kategorie zu wählen ist.

**Forschungsfrage 2:** Wie wirken sich die Abhängigkeiten der Entwicklungsartefakte auf die Auswirkungsmuster aus?

Für die Auswirkung der Abhängigkeiten der Entwicklungsartefakte auf die Auswirkungsmuster ergibt sich, dass das Änderungsverhalten maßgeblich durch die Abhängigkeit beeinflusst wird. Es zeigt sich, dass durch das Dekomponieren oder die Ableitung von Entwicklungsartefakten zwischen den einzelnen Systemebenen die Attribute weitergegeben werden. Das bedeutet, dass eine Änderung eines Artefakts auf der Systemebene diese weitergegeben werden an das Artefakt auf der Elementebene. Dieses Verhalten kann sich ebenfalls von der Elementebene auf die Systemebene fortpflanzen. Es lässt sich daraus schließen, dass das Änderungsvolumen durch die Art der Abhängigkeit beeinflusst wird. Somit bedarf es einer Betrachtung der formalen Abhängigkeiten, um frühzeitig diese Propagationseffekte zu berücksichtigen.

**Forschungsfrage 3:** Wie lassen sich die Auswirkungsmuster bewerten?

Die Bewertung der Auswirkungsmuster kann durchgeführt werden, indem die betroffenen Entwicklungsartefakte identifiziert werden. Es wird zwischen direkten und indirekten Auswirkungen zu differenziert. Die direkten Auswirkungen betreffen Entwicklungsarte-

fakte, die auf dem direkten Änderungspfad zwischen dem initialen Element und der geplanten Änderung liegen. Das bedeutet in dem Kontext des Forschungsansatzes, dass eine Eigenschaftsabsicherung eine Änderung eines verbundenen Entwicklungsartefaktes auslöst. Die bestehende Verknüpfung der beiden Entwicklungsartefakte kann jedoch so ausgeprägt sein, dass weitere Entwicklungsartefakte auf dem Pfad liegen. Diese müssen Informationen weitergeben, welche relevant für die Eigenschaftsabsicherung sind und somit auch relevant für die Änderung. Die indirekten Auswirkungen müssen ebenfalls mit in die Bewertung einfließen. Diese Artefakte können von der Änderung betroffene Informationen durch die Verknüpfung zu Artefakten auf dem Änderungspfad erhalten. Daraus lässt sich schließen, dass für die Bewertung die Anzahl möglicher indirekter und direkter Auswirkungen betrachtet wird.

**Forschungsfrage 4:** Welche Handlungsmaßnahmen ergeben sich aus diesen Wirkungsmustern?

Aus dem Forschungsansatz ergibt sich die Handlungsmaßnahme, dass eine Änderung auf Basis der Bewertung durch den Anwender ausgewählt wird. Für das weitere Verfahren sind dann auf Basis der erarbeiteten Algorithmen weitere Änderungen zu identifizieren. Die Notwendigkeit ergibt sich durch die Individualität des jeweiligen Systems und der daraus folgenden unterschiedlichen Abhängigkeiten der Entwicklungsartefakte. Auf Basis der Algorithmen wird der Anwender befähigt, weitere Änderungsauswirkungen zu identifizieren, welche für die Umsetzung notwendig sind. Zusätzlich ergeben sich Randbedingungen, die für eine erfolgreiche Umsetzung der Identifikation weiterer Entwicklungsartefakte nötig sind.



## 7 Zusammenfassung und Ausblick

Das Ziel der Arbeit ist die Identifizierung und Bewertung von Auswirkungsmustern, welche durch Änderungen in der Eigenschaftsabsicherung resultieren. Durch die entwickelten Handlungsmaßnahmen soll der V&V-Ingenieur befähigt werden, auf Basis der detektierten Änderung eine Aussage über die anstehenden Änderungsauswirkungen zu tätigen. Damit eine Entscheidung getroffen werden kann, muss der Anwender in die Lage versetzt werden, durch die Bestimmung der Ausgangslage und den vorliegenden Daten die Änderungsauswirkung zu bestimmen. Die Bewertung wird auf Basis der identifizierten Muster und der Kenntnis des Entwicklungsstandes durchgeführt. Darüber hinaus ist das weitere Vorgehen anhand der Handlungsvorgaben und gegebenen Randbedingungen festgelegt.

Für die Ausarbeitung wurden zu Beginn die theoretischen Grundlagen anhand einer Literaturrecherche erfasst. Im Fokus steht dabei, welche Abhängigkeiten die Entwicklungsartefakte der Anforderungserhebung und der Eigenschaftsabsicherung aufweisen. Die Abhängigkeiten ergeben sich aus den Modellierungsansätzen und dem Vorgehen gemäß den Richtlinien. Neben den Relationen der Entwicklungsartefakte umfassen die Grundlagen den Aufbau der Modellierungssprache mit den zugrundeliegenden Stereotypen der Modellierung.

Für die weitere systematische Literaturrecherche wurden drei Suchstrings aufgebaut, welche dabei das Augenmerk auf drei verschiedene Bereiche legen. Der erste Suchstring zielt darauf ab, artverwandte Ansätze zu dem Forschungsansatz zu identifizieren. Dabei wird nach Ansätzen gesucht, die mit Hilfe von Mustern die Änderung bestimmen oder in denen die Änderung ausgehend von der Eigenschaftsabsicherung beginnt. Hierbei wurden vier Ansätze identifiziert, welche eine Schnittmenge mit dem Geltungsbereich des Ansatzes aufweisen. Der zweite Suchstring ist darauf ausgelegt, innerhalb des Pattern-Based Systems Engineering nach Ansätzen für testinduzierte Änderungen zu suchen. Diesem Suchstring lassen sich drei Ansätze zuordnen. Der dritte Suchstring dient der Identifikation von Fallstudien, welche die Modellierung von Systemen in dem industriellen Rahmen darstellen. Mit Hilfe des Suchstrings wurden acht Ansätze zu identifiziert.

Die Literaturrecherche bezüglich des ersten Suchstrings verdeutlicht die Forschungslücke für die Auswirkungsanalyse ausgehend von testinduzierten Änderungen. Ebenfalls werden die Änderungen nicht über die Auswertung von Mustern der betroffenen Entwicklungsartefakte bestimmt. Dies konnte durch die Analyse der Literatur des zweiten Suchstrings untermauert werden, welche den Themenbereich des Pattern-Based Systems Engineering betrachtet.

Durch die Untersuchung der Veröffentlichungen, welche eine Schnittmenge mit dem angestrebten Forschungsansatz aufweisen, wurden grundsätzliche Erkenntnisse abgeleitet. Diese bestehen zum einen darin, eine Änderung nach ihrer Art und ihres initiierten Moments zu betrachten. Dadurch wird es möglich, den Änderungspfad innerhalb des Systemmodells zu identifizieren. Zusätzlich wurde identifiziert, wie das Elementverhalten

die Ausbreitung einer Änderung innerhalb des Systemmodells beeinflusst und somit maßgeblich das Änderungsvolumen bestimmt.

Aus den ermittelten Ansätzen für die Modellierung cyberphysischer Systeme wurden die Auswirkungsmuster entwickelt. Für die Bestimmung der Muster wurden die theoretischen und praktischen Ansätze mit verschiedenen Änderungen beaufschlagt, um die Auswirkung zu analysieren. Die jeweiligen Muster wurden zusammengeführt und geclustert. Daraus ergeben sich die resultierenden Artefakte. Für diese Artefakte wurden die Abhängigkeiten der einzelnen Elemente untersucht. Aus den Analysen ergeben sich die Auswirkungsmuster. Dabei wurden die Auswirkungen sowohl über die ebenen-spezifischen als auch über die ebenen-übergreifenden Auswirkungsmuster betrachtet.

Aufbauend auf die Auswirkungsmuster sind dann Bewertungen durchgeführt worden. Die Bewertung beinhaltet die Anzahl indirekt und direkt betroffener Elemente der Auswirkungsmuster. Auf Basis der Bewertung und der Art der Elemente wurde dann eine Einordnung des möglichen Verhaltens gemäß des Ansatzes nach Eckert et al. [ECZ04] durchgeführt. Für die Bewertung wird ein ebenen-übergreifendes Muster und ein ebenen-spezifisches Muster herangezogen. Zusätzlich zu den Bewertungen der einzelnen Muster wird für die Gesamtbewertung die Änderungsauswirkung und der Entwicklungsstand des Systems mit einbezogen. Aus den drei Werten wird die Änderungsklassifikationsnummer berechnet. Mit Hilfe der Änderungsklassifikationsnummer werden die Änderungen in drei mögliche Kategorien eingeordnet. Diese vermitteln dem V&V-Ingenieur einen Eindruck über das Änderungsausmaß.

Für den weiteren Umgang mit der Änderung sind zusätzlich zu der Bewertung Handlungsmaßnahmen definiert worden. Diese beinhalten Prüfalgorithmen, welche für die Identifizierung weiterer betroffener Systemartefakte genutzt werden. Somit wird der Anwender befähigt, das genaue Ausmaß der Änderung zu identifizieren. Damit bei der Identifizierung keine Fehler unterlaufen, wurden für die jeweiligen Auswirkungsmuster zusätzlich Randbedingungen definiert. Diese Handlungsmaßnahmen sind als Aktivitätsdiagramm dargestellt.

Die entwickelten Handlungsmaßnahmen werden dann mit Hilfe von Experteninterviews validiert. Die Experteninterviews werden mit insgesamt sechs Probanden aus der Wissenschaft und des UPBracing Teams durchgeführt. Die Probanden erhalten zu Beginn eine Einführung in die Theorie und das Vorgehen, um eine vergleichbare Wissensbasis herzustellen. Darauf aufbauend ist das Vorgehen anhand von Beispielszenarien umzusetzen. Für die Auswertung wird der SUS-Usability-Fragebogen verwendet sowie ein semi-strukturiertes Gespräch anhand von Leitfragen durchgeführt. Die Leitfragen basieren auf den definierten Erfolgsfaktoren.

Die Ergebnisse der Validierung ergeben, dass durch den Ansatz ein Ausgangspunkt für das Verarbeiten der Änderung geschaffen wird. Der Anwender wird befähigt, das mögliche Änderungsverhalten abzuschätzen und unterschiedliche Änderungen gegeneinander abzuwägen. Mit den Handlungsmaßnahmen wird die Identifikation weiterer betroffener

Elemente ermöglicht. Somit ist eine Entscheidungsgrundlage geschaffen worden, welche als Startpunkt für den weiteren Prozess dient.

Für die Nutzung der Auswirkungsmuster, des Bewertungsschemas und der Handlungsmaßnahmen ergeben sich Entwicklungspotenziale. Diese umfassen eine Teilautomatisierung, welche den Prozess effizienter gestaltet. Eine Teilautomatisierung verbessert die Handhabung des Verfahrens. Ebenfalls wird die Verfahrensweise mit einer Benutzeroberfläche übersichtlicher, da mit Hilfe der Teilautomatisierung die Auswahl der Muster automatisiert erfolgen kann. Das Risiko, falsche Muster auszuwählen, wird deutlich reduziert, wodurch sich die Fehleranfälligkeit des Prozesses insgesamt verringert. Die Umsetzung ist in diesem Fall mit einer Software zu gestalten, wodurch sich auch die Möglichkeit ergeben würde, eine Verknüpfung zu bestehenden Modellierungswerkzeugen in SysML herzustellen. Somit wird die weitere Datengrundlage für die Ermittlung betroffener Artefakte direkt verknüpft.

Darüber hinaus besteht das Potenzial, ausgehend von den entwickelten Mustern weitere Untergliederungen durchzuführen. Dadurch wird es möglich, eine höhere Anzahl individueller Fälle abzubilden, was dann eine genauere Aussage über das Änderungsverhalten zulässt. Dies bedeutet, dass über den geschaffenen Startpunkt hinaus das Verhalten beschreibbar wird, was jedoch mit einem Komplexitätszuwachs beaufschlagt ist.

Abschließend zeigt sich, dass die Auswirkungsanalyse durch ein vorab gut vernetztes Modell des Systems genauer wird. Dadurch ergibt sich neben den reaktiven Maßnahmen ab der Designphase und der damit einhergehenden Eigenschaftsabsicherung, dass präventive Maßnahmen getroffen werden können. Diese umfassen beispielsweise die Erweiterung der Kontrollfragen im V-Modell. Hinsichtlich des Forschungsauftrags ergeben sich hierzu zwei Fragen für die Erweiterung des zweiten Kontrollpunktes. Die Fragen lauten:

**Frage 1:** Sind die Abhängigkeiten der Systemparameter durch eine Formel modelliert?

**Frage 2:** Sind die Abhängigkeiten von den Entwicklungsartefakten zu fixen Anforderungen modelliert und erkennbar?

Diese erste Frage zielt darauf ab, Abhängigkeiten, die eingehalten werden müssen, mit Hilfe einer Formel zu definieren. Dies ist denkbar für Abmaße, welche sich gegenseitig beeinflussen. Die zweite Frage zielt darauf ab, dass Vorgaben, die nicht änderbar sind wie beispielsweise gesetzliche Anforderungen ebenfalls als nicht änderbar in dem Systemmodell zu kennzeichnen sind.

Aus dem Forschungsansatz resultiert ein Vorgehen mit Auswirkungsmustern, welches den Anwender befähigt die Änderungen zu identifizieren und zu bewerten. Die Bewertung schafft eine Grundlage für die Entscheidung, welche Änderung in dem betrachteten



Fall den geringsten Änderungsumfang erzeugt. Somit ist es möglich das weitere Vorgehen für den Änderungsprozess ausgehend von der Eigenschaftsabsicherung zu bestimmen.

## 8 Literaturverzeichnis

- [AH18]: Azrad, S.; Hassim, M.: Conceptual exploration for airborne wind energy system for electricity generation in Pulau Perhentian Kecil, Malaysia—MBSE approach. In *International Journal of Engineering & ...* 2018
- [BG21]: Bender, B.; Gericke, K. (Hrsg.): *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung*, 2021, Springer Berlin Heidelberg
- [BGO+23]: Baron, C.; Grenier, L.; Ostapenko, V.; Xue, R.: Using the ARCADIA/Cappella Systems Engineering Method and Tool to Design Manufacturing Systems—Case Study and Industrial Feedback. In *Systems* 2023
- [Bo96]: Brooke, John; others: SUS-A quick and dirty usability scale. In *Usability evaluation in industry*, S. 4–7, 189. Jg. 1996
- [BSB+10]: Bassi, L.; Secchi, C.; Bonfe, M.; ...: A SysML-based methodology for manufacturing machinery modeling and design. In *IEEE/ASME transactions ...* 2010
- [BTP+02]: Bai, Xiaoying; Tsai, Wei-Tek; Paul, Ray; Feng, Ke; Yu, Lian: Scenario-based modeling and its applications *Proceedings of the Seventh IEEE International Workshop on Object-Oriented Real-Time Dependable Systems*.(WORDS 2002), S. 253–260
- [BW23]: Brahma, A.; Wynn, D.: Concepts of change propagation analysis in engineering design. In *RESEARCH IN ENGINEERING DESIGN*, S. 117–151, 34. Jg. 2023
- [Con04]: Conrad, Mirko: *Modell-basierter Test eingebetteter Software im Automobil. Auswahl und Beschreibung von Testszenarien*, 2004, Deutscher Universitätsverlag
- [DCK+07]: Deubel, Till; Conrad, Jan; Köhler, Christian; Wanke, Sören; Weber, Christian: Change impact and risk analysis (CIRA): Combining the CPM/PDD theory and FMEA-methodology for an improved engineering change management 2007
- [Del14]: Delligatti, Lenny: *SysML distilled. A brief guide to the systems modeling language*, 2014, Addison-Wesley
- [Dou21]: Douglass, Bruce: *Agile model-based systems engineering cookbook. Improve system development by applying proven recipes for effective agile systems engineering*, 2021, Packt Publishing
- [ECZ04]: Eckert, Claudia; Clarkson, P.; Zanker, Winfried: Change and customisation in complex engineering domains. In *RESEARCH IN ENGINEERING DESIGN*, S. 1–21, 15. Jg. 2004
- [Eig21]: Eigner, Martin: *System Lifecycle Management. Digitalisierung des Engineering*, 2021, Springer Vieweg

- [ESS22]: Ellsel, C.; Senkal, G.; Stark, R.: Using MBSE With SysML to Analyze Change Impacts, Including Efforts, in Technical Systems 2022 IEEE International Symposium on Systems Engineering (ISSE), S. 1–8
- [FCA+21]: F. Willich; C. Wolff; A. Sutorma; U. Jahn; M. Stampa: Model-based Systems Engineering of an Active, Oleo-Pneumatic Damper for a CS-23 General Aviation Aircraft Landing Gear 2021 IEEE European Technology and Engineering Management Summit (E-TEMS), S. 166–172
- [FGN+00]: Fricke, Ernst; Gebhard, Bernd; Negele, Herbert; Igenbergs, Eduard: Coping with changes: causes, findings, and strategies. In Systems Engineering, S. 169–179, 3. Jg. 2000
- [Fri14]: Friedenthal, Sanford: A Practical Guide to SysML. The Systems Modeling Language, 2014, Elsevier Science
- [GDB18]: Graessler, Iris; Dattner, Michael; Bothen, Martin: Main Feature List as core success criteria of organizing Requirements Elicitation R&D Management Conference, Milan, Italy 30. Juni-4. Juli 2018
- [Gei81]: Geist, M. (Hrsg.): Die Führung des Betriebes. Curt Sandig zu seinem 80. Geburtstag gewidmet, 1981, Pöschel
- [GGT13]: Gausemeier, Jürgen; Gaukstern, Tobias; Tschirner, Christian: Systems Engineering Management Based on a Discipline-Spanning System Model. In Procedia Computer Science, S. 303–312, 16. Jg. 2013
- [GO22]: Gräßler, Iris; Oleff, Christian: Systems engineering. Verstehen und industriell umsetzen, 2022, Springer Vieweg
- [GWO22]: Gräßler, Iris; Wiechel, Dominik; Oleff, Christian: Extended RFLP for complex technical systems 2022 IEEE International Symposium on Systems Engineering (ISSE), S. 1–8
- [GWT22]: Gräßler, I.; Wiechel, D.; Thiele, H.: Controlling of the Modeling of Mechatronic Products: Fortschrittskontrolle der Modellierung mechatronischer Produkte. In engrxiv.org 2022
- [Heh11]: Hehenberger, Peter: Computerunterstützte Fertigung: Eine kompakte Einführung, 2011, Springer-Verlag
- [Hof13-ol]: Hoffmann, Hans: Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering Deskbook Release 4.1. Model-Based Systems Engineering with Rational Rhapsody and Rational Harmony for Systems Engineering, 2013 unter: <https://www.ibm.com/docs/en/engineering-lifecycle-management-suite/design-rhapsody/9.0.1?topic=secsyscontroller-harmony-process>
- [HVB17]: Hirshorn, Steven; Voss, Linda; Bromley, Linda: Nasa systems engineering handbook, 2017
- [ILS23]: Intana, A.; Laosen, K.; Sriraksa, T.: An Automated Impact Analysis Approach for Test Cases based on Changes of Use Case based Requirement

- Specifications. In INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS, S. 967–980, 14. Jg. 2023
- [IS19]: Intana, A.; Sriraksa, T.: Impact Analysis Framework of Test Cases Based on Changes of Use Case Based Requirements 2019 23rd International Computer Science and Engineering Conference (ICSEC), S. 230–235
- [15288:2023(E)]: ISO copyright office: Systems and software engineering (15288:2023(E):05.2023)
- [Jar05]: Jarratt, Timothy and Clarkson, John and Eckert, Claudia: Engineering change. In: Clarkson, John and Eckert, Claudia (Hrsg.) Design process improvement: A review of current practice, S. 262–285, 2005, Springer London
- [Kor18]: Kornmeier, Martin: Wissenschaftlich schreiben leicht gemacht: für Bachelor, Master und Dissertation, 2018, utb GmbH
- [Kru03]: Kruchten, Philippe: The rational unified process. An introduction, 2003, Addison-Wesley
- [KWS+07]: Kordon, Mark; Wall, Steve; Stone, Henry; Blume, William; Skipper, Joseph; Ingham, Mitch; Neelon, Joe; Chase, James; Baalke, Ron; Hanks, David; Salcedo, Jose; Solish, Benjamin; Postma, Mona; Machuzak, Richard: Model-Based Engineering Design Pilots at JPL 2007 IEEE Aerospace Conference, S. 1–20, IEEE
- [Lan16]: Langer, Stefan: Änderungsmanagement. In: Lindemann, Udo (Hrsg.) Handbuch Produktentwicklung, S. 513–539, 2016, Carl Hanser Verlag GmbH & Co. KG
- [LC10]: Li, Simon; Chen, Li: Pattern-based reasoning for rapid redesign: a proactive approach. In RESEARCH IN ENGINEERING DESIGN, S. 25–42, 21. Jg. 2010
- [LC07]: Li, Simon; Chen, Li: Towards Rapid Redesign: Pattern-based Redesign Planning for Large-Scale and Complex Redesign Problems. In JOURNAL OF MECHANICAL DESIGN, S. 227–233, 129. Jg. 2007
- [LPS+15]: Lin, Heng-You; Papakonstantinou, Nikolaos; Shalyto, Anatoly; Vyatkin, Valeriy: 2015 IEEE 13th International Conference on Industrial Informatics (INDIN 2015). Cambridge, United Kingdom, 22 - 24 July 2015, 2015, IEEE
- [PMB+21]: Page, Matthew; Moher, David; Bossuyt, Patrick; Boutron, Isabelle; Hoffmann, Tammy; Mulrow, Cynthia; Shamseer, Larissa; Tetzlaff, Jennifer; Akl, Elie; Brennan, Sue; Chou, Roger; Glanville, Julie; Grimshaw, Jeremy; Hróbjartsson, Asbjörn; Lalu, Manoj; Li, Tianjing; Loder, Elizabeth; Mayo-Wilson, Evan; McDonald, Steve; McGuinness, Luke; Stewart, Lesley; Thomas, James; Tricco, Andrea; Welch, Vivian; Whiting, Penny; McKenzie, Joanne: PRISMA 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews. In BMJ (Clinical research ed.), n160, 372. Jg. 2021

- [QMW17]: Qamar, Ahsan; Meinhart, Matthew; Walley, George: 2017 IEEE Aerospace Conference. Yellowstone Conference Center, Big Sky, Montana, March 4–11, 2017, 2017, IEEE
- [RM09]: Reddi, Krishna; Moon, Young: A framework for managing engineering change propagation. In International Journal of Innovation and Learning, S. 461–476, 6. Jg. 2009
- [Roq16]: Roques, Pascal: MBSE with the ARCADIA Method and the Capella Tool 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)
- [Sch20]: Schindel, W.: System Patterns in Engineering and Science. In Handbook of Systems Sciences 2020
- [Sch23]: Schindel, W.: Pattern-Based Methods and MBSE. In Handbook of Model-Based Systems Engineering 2023
- [SVT18]: Stevenson, D.; Vine, K.; Towers, J.: Verification and Validation of a new type of Railway Signal using MBSE and Simulation. In INCOSE Annual Systems ... 2018
- [SW19]: Salehi, Vahid; Wang, Shirui: Munich Agile MBSE Concept (MAGIC). In Proceedings of the Design Society: International Conference on Engineering Design, S. 3701–3710, 1. Jg. 2019
- [UE11]: Ulrich, Karl; Eppinger, Steven: EBOOK: Product Design and Development, 2011, McGraw Hill
- [UMR+15]: Ulbrich, Simon; Menzel, Till; Reschka, Andreas; Schuldt, Fabian; Maurer, Markus: Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving 2015 IEEE 18th International Conference on Intelligent Transportation Systems, S. 982–988, IEEE
- [VDI21]: VDI 2206: 2021-11: 2206: 2021-11, Development of mechatronic and cyber-physical systems 2021
- [Wal23]: Walden, D.D. (Hrsg.): Systems engineering handbook. A guide for system life cycle processes and activities, 2023, WILEY
- [Was15]: Wasson, Charles: System Analysis, Design, and Development. Concepts, Principles, and Practices, 2015, John Wiley & Sons Incorporated
- [Wei07]: Weilkiens, Tim: Systems engineering with SysML/UML. Modeling, analysis, design, 2007, Elsevier Morgan Kaufmann OMG Press
- [YCB22]: Younse, Paulo; Cameron, Jessica; Bradley, Thomas: Comparative analysis of model-based and traditional systems engineering approaches for simulating a robotic space system architecture through automatic knowledge processing. In Systems Engineering, S. 360–386, 25. Jg. 2022
- [YD12]: Yang, Fan; Duan, Gui-jiang: Developing a parameter linkage-based method for searching change propagation paths. In RESEARCH IN ENGINEERING DESIGN, S. 353–372, 23. Jg. 2012

- [ZHB+21]: Zhang, Y.; Hoepfner, G.; Berroth, J.; Pasch, G.; Jacobs, G.: Towards holistic system models including domain-specific simulation models based on sysml. In Systems, 9. Jg. 2021
- [ZMC+22]: Z. Cui; M. Luo; C. Zhang; Y. Hua; R. Xie; R. Yu; C. Huang: MBSE for Civil Aircraft Scaled Demonstrator Requirement Analysis and Architecting. In IEEE ACCESS, S. 43112–43128, 10. Jg. 2022