

427

Verlagsschriftenreihe des Heinz Nixdorf Instituts
Prof. Dr.-Ing. Iris Gräßler (Hrsg.)
Produktentstehung

Daniel Preuß

Methodik zur automatisierten Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Daniel Preuß

***Methodik zur automatisierten
Extrahierung, Formalisierung und
Handhabung von Änderungen
technischer Anforderungen***

***Methodology for automated
extraction, formalisation and handling
of changes to technical requirements***

Bibliografische Information Der Deutschen Bibliothek

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

Band 427 der Verlagsschriftenreihe des Heinz Nixdorf Instituts

© Heinz Nixdorf Institut, Universität Paderborn – Paderborn – 2024

ISSN (Online): 2365-4422

ISBN: 978-3-947647-46-0

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

Als elektronische Version frei verfügbar über die Digitalen Sammlungen der Universitätsbibliothek Paderborn.

Satz und Gestaltung: Daniel Preuß

**Methodik zur automatisierten Extrahierung,
Formalisierung und Handhabung von Änderungen
technischer Anforderungen**

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

genehmigte
DISSERTATION

von
M.Sc. Daniel Preuß
aus Bielefeld

Tag des Kolloquiums: 28.11.2024
Referent: Prof. Dr.-Ing. Iris Gräßler
Korreferent: Prof. Dr.-Ing. Klaus-Dieter Thoben

Vorveröffentlichungen

- [GOP23] Gräßler, I.; Oleff, C.; Preuß, D.; Koch, A.-S.: Resilient Requirements Engineering. Anforderungsentwicklung im Engineering 4.0 neu gedacht. In Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF), 3 (2023).
- [GP23] Gräßler, I.; Preuß, D.: Automatisierte Abhängigkeitsanalyse von Anforderungen zur Wirkkettenmodellierung: Stuttgarter Symposium für Produktentwicklung 2023. Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart 2023.
- [GPB23] Gräßler, I.; Preuß, D.; Brandt, L.; Mohr, M.: Efficient Formalisation of Technical Requirements for Generative Engineering: Proceedings of 24th International Conference on Engineering Design (ICED 2023), Bordeaux, France 2023.
- [GPB22] Gräßler, I.; Preuß, D.; Brandt, L.; Mohr, M.: Efficient Extraction of Technical Requirements Applying Data Augmentation: Proceedings of 8th IEEE International Symposium on Systems Engineering 2022, Wien 2022.
- [GTT22] Graessler, I.; Tusek, A. M.; Thiele, H.; Preuß, D.; Grewe, B.; Hieb, M.: Literature study on the potential of Artificial Intelligence in Scenario-Technique: XXXIII ISPIM Innovation Conference Proceedings, Copenhagen, Denmark 2022.
- [GOP22] Gräßler, I.; Oleff, C.; Preuß, D.: Proactive Management of Requirement Changes in the Development of Complex Technical Systems. In Applied Sciences, 12 (2022) 4; S. 1874.
- [GOH22] Gräßler, I.; Oleff, C.; Hieb, M.; Preuß, D.: Automated Requirement Dependency Analysis for Complex Technical Systems. In Proceedings of the Design Society, Volume 2: DESIGN2022, 2 (2022); S. 1865–1874.
- [GWK22] Gräßler, I.; Wiechel, D.; Koch, A.-S.; Preuß, D.; Oleff, C.: Model-Based Effect-Chain Analysis for Complex Systems. In Proceedings of the Design Society, Volume 2: DESIGN2022, 2 (2022); S. 1885–1894.
- [GPP22] Gräßler, I.; Preuß, D.; Pottebaum, J.: Extrahierung von Anforderungen aus natürlichsprachlichen Lastenheften: Was erschwert eine KI-basierte Extrahierung? In Softwaretechnik-Trends, 42 (2022) 1.
- [GRW21] Gräßler, I.; Roesmann, D.; Wiechel, D.; Preuß, D.: Determine similarity of assembly operations using semantic technology. In 24th CIRP Design Conference, 104 (2021); S. 1245–1250.
- [GOP21] Gräßler, I.; Oleff, C.; Preuß, D.: Holistic change propagation and impact analysis in requirements management: Proceedings of R&D Management Conference 2021, Strathclyde/Glasgow 2021.
- [GPO21] Gräßler, I.; Pottebaum, J.; Oleff, C.; Preuß, D.: Handling of explicit uncertainty in requirements change management. In DS 109: Proceedings of the Design Society: 23rd International Conference on Engineering Design (ICED21), 1 (2021); S. 1687–1696.
- [GP21] Gräßler, I.; Preuß, D.: Anwendbarkeit von Requirement Mining in Benutzerrezensionen für die Entwicklung mechatronischer Produkte im B2C-Markt. In: Bertram, T. et al.: Digital-Fachtagung VDI MECHATRONIK 2021, Darmstadt 2021; S. 68–73.
- [GPO20] Gräßler, I.; Preuß, D.; Oleff, C.: Automatisierte Identifikation und Charakterisierung von Anforderungsabhängigkeiten – Literaturstudie zum Vergleich von Lösungsansätzen: Proceedings of the 31st Symposium Design for X (DFX2020) 2020; S. 199–208.

Zusammenfassung

Unvollständige Anforderungslisten, mehrdeutige Anforderungen und Anforderungsänderungen sind Hauptgründe für das Scheitern von Entwicklungsprojekten. Die manuelle Analyse von natürlich-sprachlichen Anforderungsdokumenten – zur Erhöhung der Vollständigkeit der Anforderungsliste – ist aufwändig, fehleranfällig und führt zu hohen Kosten. Es besteht ein Bedarf nach einer automatisierten Extrahierung von Anforderungen, um Aufwände und Kosten in der Analyse natürlich-sprachlicher Anforderungsdokumente zu reduzieren. Natürliche Sprache ist inhärent mehrdeutig. Es ist für Software-Werkzeuge nicht möglich, diese direkt zu interpretieren. Außerdem müssen Anforderungen formalisiert werden, um diese für verschiedene Software-Werkzeuge in der Produktentwicklung interpretierbar zu machen. Zur Handhabung von Anforderungsänderungen wird ein Risikomanagement benötigt, welches frühzeitig im Projekt anwendbar ist. Durch die Methodik zur Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen werden Lösungen zur Überwindung dieser Schwierigkeiten zur Verfügung gestellt. In der Methodik werden KI-basierte Methoden als „Enabler“ für vollständige und eindeutige Anforderungslisten genutzt. Außerdem wird ein Mehrwert für das Risikomanagement von Anforderungsänderungen generiert, indem Anforderungsabhängigkeiten aufwandsarm erkannt werden und die Berechnung von Änderungsrisiken von Anforderungen in der industriellen Praxis ermöglicht werden.

Abstract

Incomplete requirement specifications, ambiguous requirements and requirement changes are the main reasons of failure of engineering projects. Manual analysis of natural language specification documents – to increase completeness of requirement specifications – is time-consuming, error-prone and leads to high costs. There is a need for automated extraction of requirements to reduce the effort and cost of analysing natural language specification documents. Natural language is inherently ambiguous. It is not possible for software tools to interpret it directly. Furthermore, requirements need to be formalised to make them interpretable for different engineering tools. To handle changes in requirements, risk management is needed that can be applied early in the project. The methodology for extracting, formalising and handling changes to technical requirements provides solutions for overcoming these difficulties. In the methodology, AI-based approaches are used as enablers for complete and unambiguous requirement specifications. In addition, value is generated for the risk management of requirements changes, in that requirement dependencies can be identified with little effort and the calculation of change risks is enabled in practice.

Methodik zur automatisierten Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Zielsetzung	2
1.2	Wissenschaftliches Vorgehen.....	3
2	Grundlagen	7
2.1	Anforderungsentwicklung	8
2.2	Resilient Requirements Engineering	11
3	Stand der Forschung.....	13
3.1	Extrahierung von Anforderungen.....	14
3.2	Formalisierung von Anforderungen	17
3.3	Abhängigkeitsanalyse von Anforderungen	20
4	Feldstudie zur Untersuchung des Anwendungszusammenhangs..	25
4.1	Erkenntnisse zur Abhängigkeitsanalyse von Anforderungen.....	25
4.2	Erkenntnisse zur Risikoberechnung von Anforderungsänderungen	29
5	Forschungsbedarf und Anforderungen an die Methodik.....	31
5.1	Forschungsbedarf und Forschungsfragen	31
5.2	Erfolgsfaktoren der Methodik	37
6	Entwicklung der Methodik	41
6.1	Modelle	42
6.1.1	Software-Architektur.....	42

6.1.2	Referenzprozess der Methodik-Anwendung	48
6.2	Entwicklung der Methoden	50
6.2.1	Extrahierung von Anforderungen	51
6.2.2	Formalisierung von Anforderungen	56
6.2.3	Abhängigkeitsanalyse von Anforderungen.....	60
6.2.4	Berechnung der Änderungsrisiken von Anforderungen.....	65
6.3	Software-Werkzeug	71
6.4	Abgrenzung zu Vorarbeiten des Lehrstuhls für Produktentstehung.....	76
7	Validierung der Methodik.....	79
7.1	Extrahierung und Klassifizierung von Anforderungen	80
7.1.1	Vorstudie: Augmentation von Anforderungsdaten mittels GPT-J	80
7.1.2	Extrahierung von Anforderungen mittels Daten-Augmentierung	82
7.2	Formalisierung von Anforderungen	88
7.3	Abhängigkeitsanalyse von Anforderungen	91
7.4	Validierung des Methodik-Erfolgs	96
7.4.1	Auswertung der Bewertungsbögen	97
7.4.2	Erfüllung der Erfolgsfaktoren.....	104
7.5	Diskussion	114
8	Zusammenfassung und Ausblick.....	119
9	Literaturverzeichnis.....	121

Anhang

A1	Ergänzende Unterlagen zu den Grundlagen	141
A2	Unterlagen zu den Vorarbeiten.....	146
A2.1	Wirkkettenmodellierung.....	147
A2.2	Nutzwertanalyse.....	151
A2.3	ARCA Projekt.....	153
A2.4	ProMaRC-Methodik.....	156
A2.5	Bewertung von Änderungsauswirkungen von Anforderungen	165
A2.6	Projektcharakteristika zur Auswahl von Vergleichsprojekten	166
A2.7	Systemmodellierung.....	167
A3	Aktivitäten zur Einbindung von Industrieanwendern	169
A3.1	ARCA: Rahmendaten von Aktivitäten und Unterlagen	169
A3.2	BIKINI: Rahmendaten von Aktivitäten und Unterlagen.....	184
A4	Unterlagen zur Methodikentwicklung	200
A4.1	Anforderungsdiagramme.....	200
A4.2	Use-Case Diagramme.....	204
A4.3	Aktivitätsdiagramme.....	207
A4.4	Paketdiagramme	210
A4.5	Algorithmen zum Extrahieren von Textsegmenten	212
A4.6	Algorithmen zur Klassifizierung von Anforderungen.....	213
A4.7	Algorithmen zur Formalisierung von Anforderungen	214
A4.8	Domänenlexika	215
A4.9	Algorithmen zur Abhängigkeitsanalyse von Anforderungen.....	235
A5	Unterlagen zur Validierung der Methodik	236

Abkürzungsverzeichnis

AM	Anforderungsmanagement
ARCA	Automated Requirement Change Analysis for complex technical systems
ATOS	Atos Information Technology GmbH
BERT	Bidirectional Encoder Representations from Transformers
BIKINI	Bionik und KI zur nachhaltigen Integration der Entwicklung von Leichtbau-Produkten
BMBF	Bundesministerium für Bildung und Forschung
BMWK	Bundesministerium für Wirtschaft und Klimaschutz
CAD	Computer-aided Design
CoNLL	Conference on Natural Language Learning
CSV	Comma-separated values
DFG	Deutsche Forschungsgemeinschaft
DMM	Domain Mapping Matrix
DSM	Design Structure Matrix
ECM	Engineering Change Management
EDAG	Edag Engineering GmbH
EF	Erfolgsfaktor
FF	Forschungsfrage
GaIP	Gestaltevolution durch algorithmisierte Informationsrückführung aus dem Produktlebenszyklus
gPLC	generic product lifecycle
GPT	Generative Pre-trained Transformer
GUI	Graphical User Interface
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of things
IPO	Input-Processing-Output

IT	Informationstechnologie
KI	Künstliche Intelligenz
LTL	Linearer Temporaler Logik
MECA	Model-based Effect Chain Analysis
MLR	Multinomiale Logistische Regression
NER	Named Entity Recognition
NLP	Natural Language Processing
NN	Neuronales Netz
OptiAMix	Mehrzieloptimierte und durchgängig automatisierte Bauteilentwicklung für additive Fertigungsverfahren im Produktentstehungsprozess
OOSEM	Object-Oriented Systems Engineering Method
POS	Part-of-speech
PDF	Portable Document Format
ProMaRC	Proactive Management of Requirement Changes
RCM	Requirement Change Management
RF	Random Forest
RNN	Recurrent Neural Network
RRE	Resilient Requirements Engineering
RTM	Requirements Traceability Matrix
SVM	Support Vector Machine
SYSMOD	Systems Modeling Toolbox
SysML	Systems Modeling Language
UML	Unified Modeling Language
UNECE	Wirtschaftskommission für Europa
VUCA	Volatilität, Unsicherheit, Komplexität, Ambiguität
WiRPro	Erweiterung des Product Lifecycle Managements durch wissensbasierte Rückführung von Produktnutzungsinformationen in die Produktentwicklung
ZWM	Zertifizierungsgerechte Wirkkettenmodellierung

Symbolverzeichnis

E_{exogen}	Erwartete Anzahl an Änderungen durch exogene Einflüsse
$E_{endogen}$	Erwartete Anzahl an Änderungen durch endogene Einflüsse
E_{gesamt}	Gesamte erwartete Anzahl an Änderungen
f_n	False negative
f_p	False positive
$\mu_{\text{Änderungen}}$	Durchschnittliche Anzahl an Änderungen pro Anforderung
μ_{PR}	Mittelwert des PageRank
$P_{endogen}$	Endogene Änderungswahrscheinlichkeit
P_{gesamt}	Gesamte Änderungswahrscheinlichkeit
$PR(r_i)$	PageRank von Anforderung i
t_n	True negative
t_p	True positive

1 Einleitung

Eine zentrale Aufgabe der Anforderungsentwicklung ist es, ein hoch-qualitatives Anforderungsset zu ermitteln. Qualitätsdefizite sowie Ineffizienzen in der Handhabung von Anforderungsänderungen führen dazu, dass Projektziele hinsichtlich Qualität, Zeit und Kosten nicht erreicht werden [HVM18, The17, Wic17]. Dieses Problem besteht seit längerer Zeit und wurde in einer Studie der Standish Group vor fast 30 Jahren umfassend untersucht [The95]. Qualitätsdefizite von Anforderungen – *Eindeutigkeit* und *Vollständigkeit* – sowie *Anforderungsänderungen* sind Hauptursachen fehlgeschlagener Entwicklungs-Projekte: 13 % der Projekte scheitern aufgrund unvollständiger Anforderungslisten und 9 % der Projekte scheitern aufgrund von Anforderungsänderungen. Insgesamt sind ca. 44 % der Ursachen, weshalb ein Projekt scheitert, auf eine ineffektive Anforderungsentwicklung zurückzuführen: unvollständige Anforderungslisten, fehlender Einbezug von Nutzern, Anforderungsänderungen und unrealistische Erwartungen (siehe Bild 1-1, links). Eine Studie, in welcher 226 Unternehmen aus 10 Ländern befragt wurden, zeigt die Aktualität dieser Problematik [MWK16]. Hauptursachen fehlgeschlagener Projekte (siehe Bild 1-1, rechts) sind weiterhin unvollständige Anforderungslisten (48 %), nicht-eindeutige Anforderungslisten (33 %) und Anforderungsänderungen (33 %)¹. Diese Qualitätsdefizite sind nach einer Erhebung der Robert Bosch GmbH insbesondere für die Entwicklung komplexer technischer Systeme relevant [GHS12]. Ein Großteil der Feldbeanstandungen sind laut dieser Erhebung auf eine unzureichende Anforderungsentwicklung zurückzuführen.

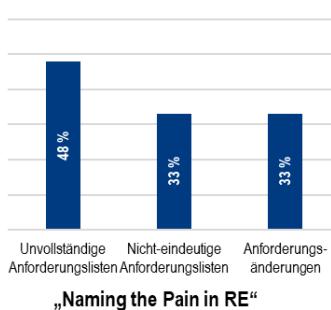
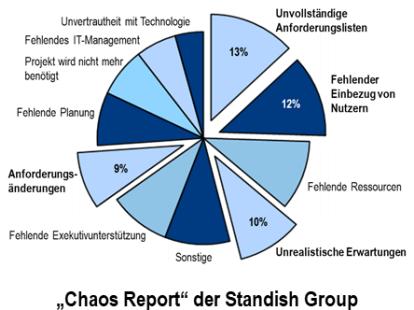


Bild 1-1: Ursachen für das Scheitern von Projekten [MWK16, The95]

¹ Im „Chaos Report“ wurde eine Ursache fehlgeschlagener Projekte abgefragt (eine Antwortmöglichkeit) und in „Naming the Pain in RE“ konnten die Befragten mehrere Ursachen angeben (mehrere Antwortmöglichkeiten). Aus diesem Grund kommt es zu deutlichen Unterschieden in den angegebenen Prozentzahlen der Ursachen für fehlgeschlagene Projekte. Die Prozentzahlen sind nicht direkt vergleichbar. Es wird aber deutlich, dass ein Bedarf zur Adressierung dieser Ursachen besteht.

1.1 Motivation und Zielsetzung

Um ein *vollständiges* Anforderungsset zu ermitteln, müssen in der Entwicklung komplexer technischer Systeme verschiedene informelle, natürlich-sprachliche Anforderungsdokumente wie Interview-Transkripte oder narrative Szenarien analysiert werden. 80 % der Anforderungen an das zu entwickelnde System werden in solchen Dokumenten festgehalten [MMM15, NL03]. Natürliche Sprache ist ausdrucksstark und wird für die Kommunikation zwischen einem breiten Spektrum von Stakeholdern genutzt [CGC10]. Jedoch ist natürliche Sprache inhärent mehrdeutig [CKZ04]. Durch die Verwendung von natürlicher Sprache in Spezifikationsdokumenten [VDI21] – z. B. Lastenhefte – sind diese per se *mehrdeutig* [WRH97]. Außerdem ist es für Software-Werkzeuge nicht möglich, natürlich-sprachliche Spezifikationsdokumente direkt zu interpretieren [MMM15]. Beispielsweise müssen für die Erstellung von Geometrien in der generativen Entwicklung sämtliche Parameter wie Produkteigenschaften, verfügbarer Bauraum, zu verwendende Materialien oder anzuwendendes Fertigungsverfahren manuell vom Anforderungsingenieur¹ aus dem Lastenheft ermittelt und in das entsprechende Software-Werkzeug – z. B. Synera [Syn23] – überführt werden. In dieser Arbeit bezieht sich die Mehrdeutigkeit von Anforderungen auf die Fähigkeit von Software-Werkzeugen, Anforderungen interpretieren zu können. Damit Anforderungen durch Software-Werkzeuge interpretiert werden können, müssen diese in eine eindeutige Darstellung überführt werden. Daher besteht ein Bedarf nach einer Formalisierung.

Anforderungsänderungen sind unvermeidbar, treten in allen Phasen eines Entwicklungsprojekts auf und über die Hälfte aller Anforderungen an das System unterliegen einer hohen Änderungswahrscheinlichkeit [HVM18]. Beispielsweise wurden in der Entwicklung einer Flugzeugturbine von Rolls-Royce ca. 700 Anforderungen ermittelt [FHS15]. Es wurden im Verlauf des Projekts über 1000 Änderungen vorgenommen. Bisher werden Anforderungsänderungen in der industriellen Praxis nicht auf der Grundlage objektiver Daten, sondern reaktiv, unsystematisch und auf Basis von subjektiven Einschätzungen gehandhabt [GO19, JL17]. Es existieren nur vereinzelt Vorgehensmodelle oder Software-Werkzeuge für eine systematische Bewertung des Änderungsrisikos von Anforderungen [HVM18, Neu17]. Um die Anzahl und die negativen Auswirkungen von Anforderungsänderungen zu reduzieren, bietet die Abhängigkeitsanalyse ein großes Potenzial [GPO20, KCC12]. Anforderungsänderungen werden zunächst durch externe Auslöser (z. B. veränderte Kundenpräferenzen oder ein verbessertes Systemverständnis) verursacht, pflanzen sich aber auch innerhalb des Anforderungssets fort. Solche konsekutiven Änderungen umfassen etwa 50 % aller Anforderungsänderungen [GWB09]. Die Abhängigkeitsanalyse erhöht nicht nur das Systemverständnis, sondern ermöglicht auch eine ganzheitliche Analyse der Auswirkungen von Änderungen und ein effizientes Änderungsmanagement [GOP21, MSS12]. Die Abhängigkeitsanalyse muss kontinuierlich durchgeführt werden, um das

¹ Das in dieser Arbeit gewählte generische Maskulinum bezieht sich zugleich auf beliebige Geschlechteridentitäten.

Änderungsmanagement in allen Entwicklungsphasen zu unterstützen. Eine besondere Herausforderung stellt sie in frühen Entwicklungsphasen dar, bevor ausreichend Daten über die Lösungselemente und deren Wirkzusammenhänge vorliegen [GP21, GP22]. Eine frühzeitige Abhängigkeitsanalyse erschließt Effizienzpotenziale im Änderungsmanagement und in der Bewertung von Änderungsauswirkungen [GOS20].

Die manuelle Extrahierung und Formalisierung von Anforderungen aus natürlich-sprachlichen Spezifikationsdokumenten ist zeitaufwändig, fehleranfällig, monoton und führt zu hohen Kosten für Unternehmen [AG06]. Bei der Entwicklung komplexer technischer Systeme können Anforderungssets über 700 Anforderungen umfassen [FHS15]. Aufgrund der hohen Anzahl an Anforderungen ist der Zeitaufwand für die Extrahierung und Formalisierung von Anforderungen hoch und Abhängigkeitsanalysen werden aufgrund des hohen Zeitbedarfs in der industriellen Praxis häufig nicht durchgeführt [GOP22]. Darum ist in diesem Anwendungskontext eine automatisierte Extrahierung und Formalisierung sowie Abhängigkeitsanalyse besonders notwendig. In der Software-Entwicklung werden Algorithmen, die auf künstlicher Intelligenz (KI) basieren, zur Verarbeitung natürlicher Sprache für die Anforderungsentwicklung angewendet. Dies gilt z. B. für die Klassifizierung von Anforderungen [MMM15]. Es wird untersucht, wie Anforderungen aus einer natürlich-sprachlichen Software-Spezifikation extrahiert werden können [HK20] oder Anforderungsbeschreibungen formalisiert werden können [GPM21a]. KI-Algorithmen besitzen das Potenzial, natürlich-sprachliche Anforderungen aus Spezifikationsdokumenten automatisiert zu extrahieren und in eine formale, eindeutige Darstellung zu transformieren. Diese Ansätze müssen für die Entwicklung komplexer technischer Systeme weiterentwickelt und in einer Methodik anwendbar gemacht werden. Ziel dieser Arbeit ist daher eine

Methodik zur automatisierten Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen.

1.2 Wissenschaftliches Vorgehen

Diese Arbeit ist ein Teil der ingenieurwissenschaftlichen Forschung, welche der angewandten Wissenschaften nach ULRICH und HILL [UH76] zuzuordnen ist. Die angewandten Handlungswissenschaften folgen dem Ziel, Probleme des praktisch handelnden Menschen, für die keine Lösungen bzw. ausreichendes Wissen existiert, zu bewältigen. Aufgrund des praxisnahen, offenen und integrativen Charakters und des Bezugs sowohl zu den Ingenieur- als auch den Betriebswissenschaften wird innerhalb der Arbeit der Forschungsprozess nach ULRICH verwendet [Ulr81]. Entsprechend ist der Aufbau nach der Strategie der angewandten Forschung von ULRICH strukturiert.

Zu Beginn der Arbeit (*Kapitel 1*) erfolgen die Erfassung und Typisierung praxisrelevanter Probleme. Die Motivation und Zielsetzung der Arbeit wird durch Literaturanalysen sowie vorbereitete und moderierte Workshops mit Industriepartnern erarbeitet. Eine Übersicht über die durchgeführten Workshops ist in Anhang A3 aufgeführt. In *Kapitel 2* werden

Grundlagen erläutert und zentrale Begriffe definiert: Generischer Produktlebenszyklus, Anforderungsentwicklung, und Resilient Requirements Engineering. Es werden Herausforderungen in diesen Bereichen dargestellt, welche in folgenden Kapiteln genutzt werden, um Anforderungen an die Methodik zu ermitteln. Zur Analyse dieser Grundlagen wird eine Literaturanalyse durchgeführt.

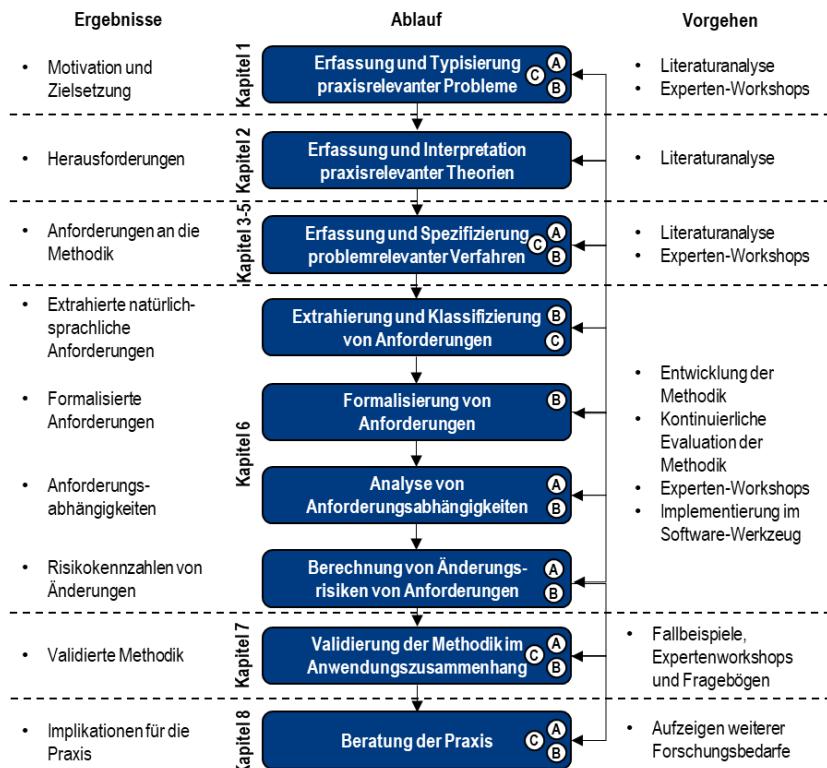
Anschließend werden spezifische problemrelevante Vorarbeiten aus der Verarbeitung natürlicher Sprache hinsichtlich der Zielsetzung dieser Arbeit untersucht (*Kapitel 3*). Betrachtet werden Methoden zur Extrahierung und Formalisierung von Anforderungen sowie zur Abhängigkeitsanalyse. Es wird eine Literaturrecherche nach MACHI und MCEVOY durchgeführt [MM16]. In *Kapitel 4* werden Validierungsergebnisse in der Anwendung von Methoden zur Verarbeitung natürlicher Sprache aus dem BMBF-geförderten Forschungsprojekts „Automated Requirement Change Analysis of Complex Technical Systems“ (ARCA) dargestellt und Erkenntnisse abgeleitet (siehe Anhang A2). Die Erkenntnisse beziehen sich auf die Abhängigkeitsanalyse von Anforderungen sowie zur Risikoberechnung von Anforderungsänderungen. Die gewonnenen Erkenntnisse fließen als Anforderungen an die zu entwickelnde Methodik ein. Aus den Reviews bestehender Ansätze sowie den Erkenntnissen, die aus der Anwendung und Analyse der Ergebnisse von Methoden zur Verarbeitung natürlicher Sprache gewonnen worden sind, resultiert ein wissenschaftlicher Handlungsbedarf, den diese Arbeit adressiert. Der Forschungsbedarf und die Anforderungen an die Methodik werden in *Kapitel 5* ermittelt.

Aufbauend auf dem Forschungsbedarf und den Anforderungen wird in *Kapitel 6* die Methodik-Entwicklung dargestellt. Eine Methodik ist nach GRÄBLER eine Sammlung von Modellen, Methoden und Hilfsmitteln zur Lösung einer Aufgabenstellung [Grä00]. Hilfsmittel sind beispielsweise Software-Werkzeuge. Es wird ein Modell der Software-Architektur des Werkzeugs zur Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen entwickelt und ein Referenzprozess zur Methodik-Anwendung abgeleitet. Die Methodik besteht aus vier Methoden: Extrahierung von Anforderungen (1), Formalisierung von Anforderungen (2), Abhängigkeitsanalyse von Anforderungen (3) sowie Berechnung der Änderungsrisiken von Anforderungen (4). Die Methoden werden in einem Software-Werkzeug implementiert.

In *Kapitel 7* erfolgt die Validierung der Methodik im Anwendungszusammenhang. Die implementierten Methoden werden anhand von Fallbeispielen im Rahmen eines Forschungsprojekts validiert. Dazu werden Lastenhefte von vergangenen Entwicklungsprojekten der Industriepartner analysiert und aufbereitet, um Anforderungsdaten zu generieren. Die Methoden werden angewendet, um Anforderungsdaten zu verarbeiten. Es werden Kennzahlen berechnet, um die Leistungsfähigkeit der Methoden zu bewerten. Außerdem werden Experten-Workshops durchgeführt, um die Methoden im Anwendungskontext zu bewerten. Es werden Bewertungsbögen erstellt, von Industrievertretern ausgefüllt und ausgewertet. Abschließend wird die Erfüllung der Anforderungen an die Methodik bewertet.

In *Kapitel 8* werden die Ergebnisse der Arbeit zusammengefasst und ein Fazit hinsichtlich der Zielerreichung gezogen. Es wird eine Beratung für die Praxis gegeben und es werden auf den Ergebnissen aufbauende Forschungsbedarfe aufgezeigt.

Das Vorgehen der Arbeit wird in Bild 1-2 in Anlehnung an die Darstellung und das Vorgehen nach ULRICH [Ulr81] visualisiert. Links in Bild 1-2 werden die Ergebnisse, die aus den Schritten des Vorgehens resultieren, dargestellt. Rechts werden die wissenschaftlichen Methoden, die innerhalb der einzelnen Schritte des Vorgehens angewendet werden, aufgeführt. Die Nutzung von Erkenntnissen aus der Praxis, die im Kontext von Forschungsprojekten gewonnen worden sind, werden innerhalb des Ablaufs, bzw. der Schritte des Vorgehens dargestellt. Innerhalb des Ablaufs werden rechts von den Schritten des Vorgehens Pfeile dargestellt, um Iterationen aufzuzeigen, die in dem Vorgehen nach ULRICH [Ulr81] möglich sind.



Legende: Nutzen von Ergebnissen und Erkenntnissen aus ARCA (A), BIKINI (B) und ZWM (C)

Bild 1-2: Vorgehen dieser Arbeit in Anlehnung an ULRICH [Ulr81]

2 Grundlagen

Die Entwicklung komplexer technischer Systeme folgt dem Generischen Produktlebenszyklus („generic product lifecycle“ – gPLC, siehe Bild 2-1). Entscheidungen in der Entwicklung müssen unter Berücksichtigung von Ambiguität, Unsicherheit, Komplexität und Volatilität (VUCA-Dimensionen) getroffen werden [BL14]. Die Produktentstehung erfordert geeignete Modelle, Methoden und Werkzeuge von der Strategie bis zum Ende des Produktlebens [GP22, Grä15].

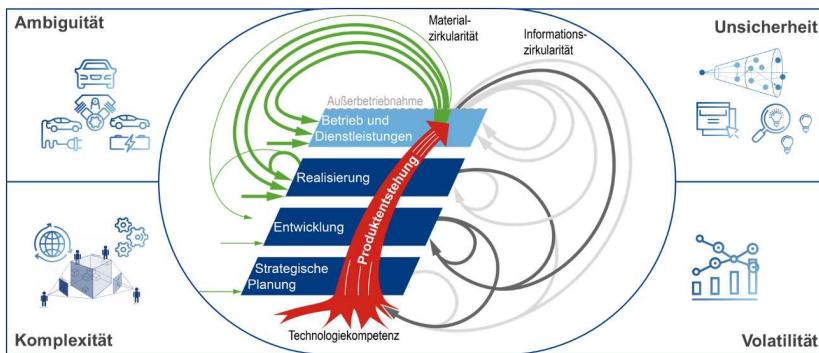


Bild 2-1: Generischer Produktlebenszyklus (gPLC) nach [GP22]

Die Produktentstehung umfasst die Ausarbeitung von Produktideen sowie deren Realisierung. Ausgehend von der strategischen Ausrichtung eines Unternehmens werden in der *Strategischen Planung* die erfolgversprechendsten Produktideen identifiziert. Es werden zukünftige Kundenbedürfnisse bestehender Märkte sowie potenzielle neue Märkte antizipiert. Bei komplexen technischen Systemen wird eine Produktidee interdisziplinär entwickelt und nach einem Gesamtoptimum über die verschiedenen Disziplinen iteriert (*Entwicklung*). Parallel zur Entwicklung werden in der *Realisierung* geeignete Umsetzungs- und Produktionskonzepte geplant. Daraus resultierende Restriktionen werden an die Entwicklung übermittelt. Die Produktentstehung umfasst explizit die ersten drei Phasen des gPLC. Die spätere Phase *Betrieb und Dienstleistungen* wird implizit einbezogen, da Einschränkungen und Möglichkeiten von Anwendungsfällen und Szenarien der Außerbetriebnahme als Restriktionen berücksichtigt werden. [GP22]

Die Produktentwicklung wird durch Entwicklungsmethodiken systematisiert. Dazu zählen u. a. das Systems Engineering [WRF15] die Richtlinien VDI/VDE 2206 [VDI21], VDI 2221 [VDI19] oder die Pahl/Beitz Konstruktionslehre [BG21]. Die Anwendung von Entwicklungsmethodik zielt auf die erfolgreiche Produktentwicklung trotz wachsender *System- und Lebenszykluskomplexität* ab. Das Wachstum der *Systemkomplexität* resultiert

aus grundlegenden Trends wie z. B. Individualisierung, Globalisierung oder neuer Konsummuster [KG18]. Unternehmen werden durch diese Trends gezwungen, die Bedürfnisbefriedigung der Stakeholder zu fokussieren, wodurch komplexe Anforderungslisten entstehen [Fra09]. Für die Erfüllung dieser Anforderungen werden wiederum komplexe technische Systeme benötigt. Die Komplexität der Systeme ist struktureller und dynamischer Natur [Ole22, Sen92]. Strukturelle Komplexität ist durch die Anzahl, Varietät und Konnektivität der Systemelemente begründet [Pat82]. Die dynamische Komplexität resultiert aus der Vielzahl unterschiedlicher Systemzustände [Sen92]. Die *Lebenszykluskomplexität* [GO22] erfordert aus dem Streben nach Zirkularität eine vorausschauende Berücksichtigung nachgelagerter Lebenszyklusphasen des zu entwickelnden Systems, woraus Anforderungsänderungen im Laufe der Entwicklung resultieren [GP21].

Aufgrund der Komplexität von technischen Systemen ist es eine Herausforderung, ein vollständiges und eindeutiges Anforderungsset zu ermitteln. Es muss eine hohe Anzahl an Anforderungen aus dem gesamten Lebenszyklus des Systems berücksichtigt werden. Aufgrund der Dynamik in der Entwicklung resultieren Anforderungsänderungen.¹

2.1 Anforderungsentwicklung

Unzureichendes Wissen über Anforderungen in der Entwicklung komplexer technischer Systeme kann schwerwiegende Folgen im Lebenszyklus haben [WHT14, WPH15]. Wenn fehlerhafte Informationen nicht erkannt werden, Anforderungen falsch interpretiert werden oder sich im Laufe der Entwicklung ändern, müssen Projekte möglicherweise abgebrochen oder mit erheblichen zusätzlichen Entwicklungskosten gerettet werden [DS19, GHS12]. Besonders die Entwicklung von Cyber-Physischen- und Produkt-Service-Systemen stellt die Anforderungsentwicklung vor Herausforderungen [WHT15, WMT17]. „Gute“ Anforderungssets erfüllen die *Qualitätskriterien* Vollständigkeit und Eindeutigkeit [GHS12, WRF15]. Die Informationsgrundlage muss alle relevanten Informationen zur Spezifikation des Produkts enthalten (Vollständigkeit). Anforderungen müssen eindeutig interpretiert werden können (Eindeutigkeit). Hauptaktivitäten der Anforderungsentwicklung sind die systematische Anforderungserhebung und das kontinuierliche Management von Anforderungsänderungen durch die Analyse, Dokumentation, Prüfung und Verwaltung von Anforderungen [GO22]. Die Aufgaben zur Ermittlung einer validierten Spezifikation sind in Bild 2-2 visualisiert. Die Zahlen, die in den folgenden Sätzen genannt werden, referenzieren auf diese Abbildung. Zunächst muss durch die Anwendung geeigneter Methoden die Sozialisierung von Wissen angeregt werden (1). Gleichzeitig ist es notwendig, Problem bzw. Aufgabe im Kontext eines Problemraums durch Beobachtungen zu identifizieren (2). Auf diesen beiden Wegen externalisiertes implizites Wissen der Bedarfsträger wird analysiert (3). Auf Grundlage

¹ In dieser Arbeit werden eingerahmte Absätze genutzt, um Kernaussagen aus dem vorherigen Abschnitt zusammenzufassen.

expliziter Bedürfnisse und des Problemraums werden die Benutzeranforderungen spezifiziert (4). Anhand der Benutzeranforderungen werden Bedingungen und Fähigkeiten festgelegt, die von einem System erfüllt werden müssen: die Spezifikation (5). Insbesondere müssen Randbedingungen wie Fertigungsrestriktionen bestimmt werden, die den Lösungsraum einschränken (6). Änderungen der Anforderungen müssen verwaltet werden (7). Die Verifikation stellt sicher, dass eine implementierte Version des Systems mit der Spezifikation konform ist (8). Durch die Validierung wird sichergestellt, dass das System die Anforderungen erfüllt (9). [Ehr07, GPO21, VDI21, WRF15]

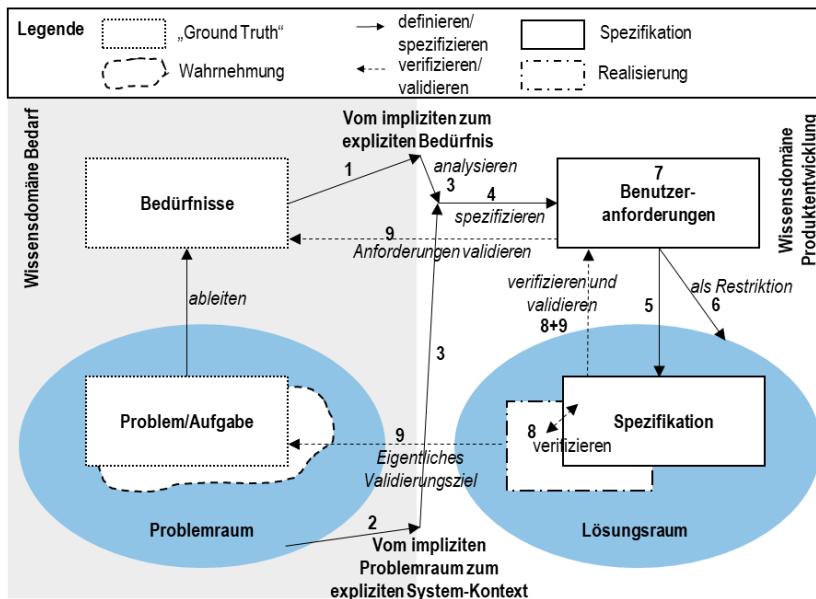


Bild 2-2: Aufgaben zur Definition einer validierten Spezifikation nach [GPO21]

Nach IEEE 610.12-1990 sind Anforderungen lösungsorientiert [IEE90]. Im Gegensatz dazu wird der Begriff „Bedürfnisse“ für problem- bzw. aufgabenorientierte Aussagen verwendet [WRF15]. Benutzeranforderungen (lösungsorientiert, aber weiter lösungsneutral) können von Systemanforderungen (lösungsspezifisch) unterschieden werden [JHH13]. Eine *Anforderung* wird definiert als [PR21]:

- 1) Ein notwendiges Bedürfnis eines Stakeholders.
- 2) Eine Fähigkeit oder Eigenschaft, die ein System erfüllen muss.
- 3) Eine dokumentierte Repräsentation eines Bedürfnisses, einer Fähigkeit oder Eigenschaft.

In dieser Arbeit werden Anforderungen in *funktionale* und *nicht-funktionale* Anforderungen (*Anforderungskategorien*) unterteilt. Die nicht-funktionalen Anforderungen werden unterteilt in Qualitäts- und technologische Anforderungen, Anforderungen an die Benutzungsoberfläche und rechtlich-vertragliche Anforderungen [Rup14].

Anforderungsänderungen und -abhängigkeiten

Aufgrund der Komplexität in der Produktentwicklung unterliegen Anforderungen kontinuierlich Änderungen. Eine *Anforderungsänderung* ist definiert als jede Änderung, die an einer Anforderung vorgenommen wird [GOS20]. Anforderungsänderungen resultieren u. a. aus Anpassungen [WB85, Wic17]. Aufgrund von Abhängigkeiten zwischen Anforderungen umfasst dies auch Anpassungen an weiteren Anforderungen, welche aufgrund von Anforderungsänderungen durchgeführt werden müssen. Anforderungsänderungen werden durch externe Auslöser (z. B. veränderte Kundenpräferenzen) verursacht. Aufgrund von Anforderungsabhängigkeiten kann dieser Änderungsbedarf aber auf weitere Anforderungen propagieren (*Änderungspropagation*).

Ein Beispiel für Änderungspropagation wird anhand der Entwicklung eines Traktors erläutert (siehe Bild 2-3). Aufbauend auf der Architektur eines Traktors soll eine neue Variante entwickelt werden, die „Internet of Things“ (IoT-)Systeme verwendet. Hierbei wurde während einer Feldstudie festgestellt, dass die Beleuchtung der Traktorfront für den Einsatz der IoT-Systeme (optische Sensorik) nicht ausreichend ist. Deshalb muss die Traktorfront besser ausgeleuchtet werden (*initialer Änderungsimpuls*). Die Anzahl an LEDs muss erhöht werden (*Änderungspropagation*) und dementsprechend die Kapazität der Batterie angepasst werden (*Weitere Änderungspropagation*), damit die Laufzeit des Systems nicht verändert wird.

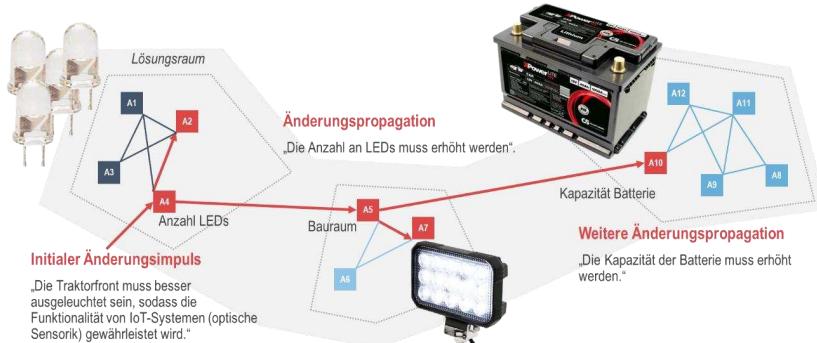


Bild 2-3: Propagationseffekte von Anforderungsänderungen am Beispiel der Entwicklung eines Traktors

Die Abhängigkeitsanalyse erhöht nicht nur das Systemverständnis, sondern ermöglicht auch eine ganzheitliche Analyse der Auswirkungen von Änderungen [GOP21, MSS12]. Um Abhängigkeiten anhand ihrer Eigenschaften zu unterscheiden, können bestehende Referenzmodelle für Abhängigkeitsarten verwendet werden [DP03, Poh96].

Aufgabe der Anforderungsentwicklung ist es, Anforderungen inklusive deren Abhängigkeiten vollständig und eindeutig zu ermitteln. Zur Ermittlung eines vollständigen Anforderungssets muss eine hohe Anzahl an Informationen aus verschiedenen Dokumenten (insbesondere Lastenhefte) berücksichtigt werden. Um Auswirkungen von Änderungspropagationen bewerten zu können, müssen Anforderungsabhängigkeiten bekannt sein. Aufgrund der Komplexität des Anforderungssets existiert eine Vielzahl an Abhängigkeiten zwischen Anforderungen.

2.2 Resilient Requirements Engineering

Die Auswirkungen von VUCA auf die Produktentwicklung zeigen sich beispielsweise in der Anforderungsentwicklung. Volatilität führt zu einer Vielzahl an Anforderungsänderungen. Unsicherheit und Komplexität zeigen sich in umfangreichen und doch unvollständigen Anforderungslisten. Ambiguität wird durch Missverständnisse hinsichtlich nicht eindeutiger Anforderungsbeschreibungen sichtbar. Um das Risiko von Projektfehlschlägen in der Entwicklung komplexer technischer Systeme trotz VUCA-Rahmenbedingungen zu reduzieren, muss die Resilienz der Produktentwicklung gestärkt werden. Nach Pariès et al. steht Resilienz für "die einem System innewohnende Fähigkeit, seine Funktionsweise vor, während oder nach Veränderungen und Störungen so anzupassen, dass es den erforderlichen Betrieb sowohl unter erwarteten als auch unerwarteten Bedingungen aufrechterhalten kann" [PWH17]. Durch die Anforderungsentwicklung wird eine vollständige und eindeutige sowie jederzeit aktuelle Erfassung der Stakeholderbedürfnisse gewährleistet und als Zielsystem für die weiteren Entwicklungsaktivitäten formalisiert. Das Resilient Requirements Engineering (RRE) beschreibt die drei wichtigsten Innovationsfelder der Anforderungsentwicklung, in denen methodische und technische Verbesserungen zu mehr Resilienz und einer höheren Erfolgswahrscheinlichkeit von Entwicklungsprojekten führen. [GOP23]

Den Anforderungen an ein System müssen eindeutige Projektziele und ein zukunfts-fähiges Geschäftsmodell zugrunde liegen. Außerdem müssen extreme Anforderungswerte effektiv gehandhabt werden [HTB23]. Dafür ist im Rahmen der Anforderungsentwicklung systematisch *Vorausschau* zu betreiben. Die so ermittelten Megatrends und daraus abgeleiteten Marktentwicklungen werden durch das Geschäftsmodell aufgegriffen und spiegeln sich in der Anforderungsliste wider.

Neben der Vorausschau ist die *Effizienz* der Tätigkeiten innerhalb der Anforderungsentwicklung ein wesentlicher Einflussfaktor auf die Resilienz. Etablierte Methoden der Anforderungsentwicklung und die Funktionalität der verfügbaren Werkzeuge zeigen, dass trotz der Digitalisierung einzelner Aktivitäten ein manuelles Vorgehen

dominiert [PR21]. Der Anwendungsaufwand führt aufgrund der Vielzahl an Anforderungen zu Qualitätseinbußen, da nach einer initialen Anforderungserhebung nur noch reaktiv gehandelt wird (z. B. Änderungsantrag). Erst durch eine Automatisierung zeitintensiver Tätigkeiten kann die Anforderungsentwicklung kontinuierlich durchgeführt und die Qualität des Anforderungssets auf hohem Niveau gehalten werden.

Die *Nachhaltigkeit* ist der dritte und letzte Bereich des RRE. Nachhaltigkeit beruht auf den drei Säulen Ökologie, Ökonomie und Soziales [Bru87]. Im Rahmen der Anforderungsentwicklung gilt es, Stakeholderbedürfnisse entlang dieser drei Säulen zu ermitteln, in Anforderungen abzubilden und zu priorisieren. Diese drei Innovationsfelder sind in einem Trendradar visualisiert (siehe Bild 2-4). Es werden verschiedene Technologien der Felder, sowie deren Reifegrade abgebildet. [GOP23]

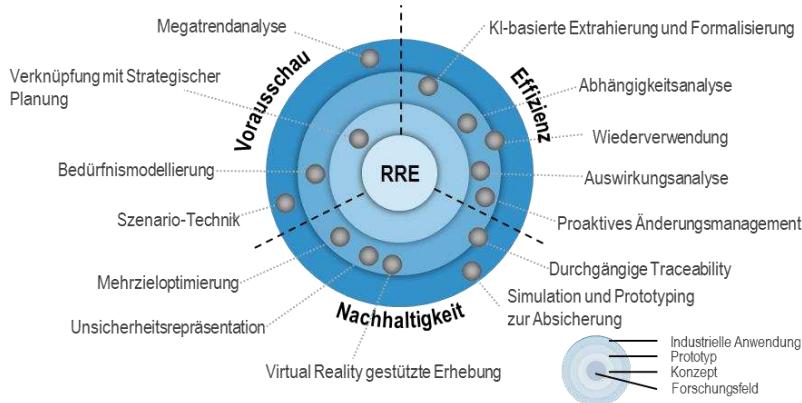


Bild 2-4: Trendradar des Resilient Requirements Engineering [GOP23]

Eine Automatisierung der Ermittlung und Formalisierung von Anforderungen stärkt auch die Resilienz der Anforderungsentwicklung. Anforderungen werden effizient aus Lastenheften extrahiert. Dadurch wird Zeit in der Entwicklung reduziert, um eine vollständige Spezifikation zu generieren. Diese Anforderungen werden formalisiert, um eine eindeutige Anforderungsbeschreibung zu realisieren und Unterstützung durch Software-Werkzeuge zu ermöglichen. Anforderungsabhängigkeiten werden identifiziert, um als Grundlage für das Risikomanagement zu fungieren und besser Einflüsse aufgrund VUCA handhaben zu können. Einerseits werden Effizienzpotenziale in der Ermittlung eines vollständigen und eindeutigen Anforderungssets erschlossen. Andererseits können dadurch informierte und nachvollziehbare Änderungsentscheidungen getroffen werden, sodass Änderungskosten reduziert werden. Weitere grundlegende Begriffe dieser Arbeit werden in Anhang A1 aufgeführt.

3 Stand der Forschung

Zur Erfassung und Spezifizierung der problemrelevanten Vorgehensweisen wird eine ergänzende Literaturrecherche nach MACHI und McEVOY durchgeführt [MM16]. Es werden relevante Veröffentlichungen zu Ansätzen zur *Extrahierung von Anforderungen*, *Formalisierung von Anforderungen* und *Abhängigkeitsanalyse von Anforderungen* gesucht. Suchbegriffe werden direkt aus diesen Bereichen abgeleitet und mit booleschen Operatoren verknüpft. Diese Suchbegriffe werden in den Online-Forschungsdatenbanken Google Scholar, Web of Science und Scopus verwendet. Die verwendeten Suchbegriffe sind in Tabelle 3-1 aufgeführt.

Tabelle 3-1: Suchbegriffe für die Literaturrecherche zum Stand der Forschung in Anlehnung an MACHI und McEVOY [MM16]

Kategorie	Bereich	Suchbegriff
A	Extrahierung von Anforderungen	A1: „Extract requirement“
		A2: „Classify requirement“
B	Formalisierung von Anforderungen	B1: „Formal“ B2: „Pattern“
C	Abhängigkeitsanalyse von Anforderungen	C1: „Detect“
		C2: „Identify“
		C3: „Classify“
D	Quelle	D1: „Specification“
E	Objekt	E1: „Requirement“
		E2: „Dependency“

Die Suchbegriffe werden zu Suchstrings kombiniert, um pro Bereich einen Suchstring zu erstellen (siehe Tabelle 3-2). Innerhalb dieser Artikel werden relevante Methoden entdeckt. Die Relevanz der Ergebnisse wird schrittweise anhand der Titel, Kurzfassungen und des Textes bewertet. Die Artikel werden ausgeschlossen, wenn sich diese wiederholen oder keine Methode für die jeweiligen Bereiche darstellen.

Tabelle 3-2: Suchstrings für die Bereiche

Bereich	Suchstring	Relevante Beiträge
Extrahierung von Anforderungen	„(A1 ODER A2) UND D1 UND E1“	16
Formalisierung von Anforderungen	„(B1 ODER B2) UND D1 UND E1“	12
Abhängigkeitsanalyse von Anforderungen	„(C1 ODER C2 ODER C3) UND D1 UND (E1 ODER E2)“	17

3.1 Extrahierung von Anforderungen

Aufgrund der hohen Anzahl von Anforderungen an komplexe technische Systeme und des damit verbundenen Aufwands muss die Extrahierung von Anforderungen mit einem hohen Automatisierungsgrad durchgeführt werden. Hierzu werden Techniken zur Verarbeitung natürlicher Sprache (NLP) eingesetzt. KURTANOVIC und MAALEJ wenden überwachtes Machine-Learning an, um Sätze einer Software-Spezifikation zu klassifizieren [KM17]. Es werden die Klassen „funktionale Anforderung“ und „nicht-funktionale Anforderung“ unterschieden. Die Merkmale, die für die Klassifizierung genutzt werden, sind beispielsweise N-Gramme¹ der Wörter und der „Part-of-speech-Tags“ (POS-Tags)². HARIS und KURNIAWAN definieren semantische Regeln, die auf POS-Tags basieren, um natürlich-sprachliche Anforderungen in einer Software-Spezifikation zu identifizieren, die einem vordefinierten Schema (Anforderungsschablonen – z. B. nach [Rup16]) folgen [HK20, HKR20]. Solche Anforderungsschablonen umfassen in Abhängigkeit der Anforderungsart die jeweiligen Bestandteile einer Anforderung. BOUTKOVA und HOUDEK extrahieren Merkmale von natürlich-sprachlichen Anforderungen in Spezifikationsdokumenten von Referenzprojekten, um relevante Anforderungen für die Wiederverwendung zu identifizieren [BH11]. METH, MUELLER und MAEDCHE verarbeiten natürlich-sprachliche Rezensionen von Nutzern auf Online-Marktplätzen, um Anforderungen für eine neue Produktgeneration einer Software-Applikation zu ermitteln [MMM15]. GRÄBLER, DATTNER und BOTHEN führen eine Strukturierung der Anforderungen mit Hilfe einer erweiterten Hauptmerkmalliste für mechatronische und Cyber-physische Systeme durch [GDB18]. [GPB22]

Die automatisierten Ansätze haben gemein, dass zunächst eine Text-Segmentierung durchgeführt werden muss (siehe Bild 3-1). Hierbei werden z. B. Co-Referenzen aufgelöst. Dies sind Anforderungen mit Bezug zu Objekten, die in einem vorherigen Segment beschrieben wurden. Im nächsten Schritt werden die Text-Segmente klassifiziert: *funktionale Anforderung*, *nicht-funktionale Anforderung* oder *keine Anforderung*. Um den Aufwand zu reduzieren, werden Qualitätsdefizite mit einem hohen Automatisierungsgrad erkannt. Hierzu existieren verschiedene Ansätze. OSMAN und ZAHARIN prüfen, ob natürlich-sprachliche Anforderungen eindeutig formuliert sind [OZ18]. GNESI, LAMI und TRENTANNI definieren Indikatoren, um die Vollständigkeit von Anforderungen automatisiert zu bewerten [GLT05]. ARORA ET AL. prüfen, ob natürlich-sprachliche Anforderungen sämtliche Bestandteile einer Anforderungsschablone enthalten (Vollständigkeit) [AMB15]. HARIS und KURNIAWAN definieren semantische Regeln, um die Einzigartigkeit von natürlich-sprachlichen Anforderungen automatisiert zu prüfen [HK20, HKR20]. In einem DFG-geförderten Projekt wird untersucht, wie bestehende Produktmodelle in der Entwicklung wiederverwendet und gleichzeitig Inkonsistenzen sowie Irrelevantes entfernt werden können (COFFEE [SW16]). [GPB22]

¹ Ein N-Gramm ist eine Sequenz von N aufeinanderfolgenden Fragmenten/Wörtern in einem Text.

² Part-of-speech-Tagging (POS-Tagging) ist die Zuordnung von Wörtern und Satzzeichen eines Textes zu Wortarten, beispielsweise Nomen oder Verben.

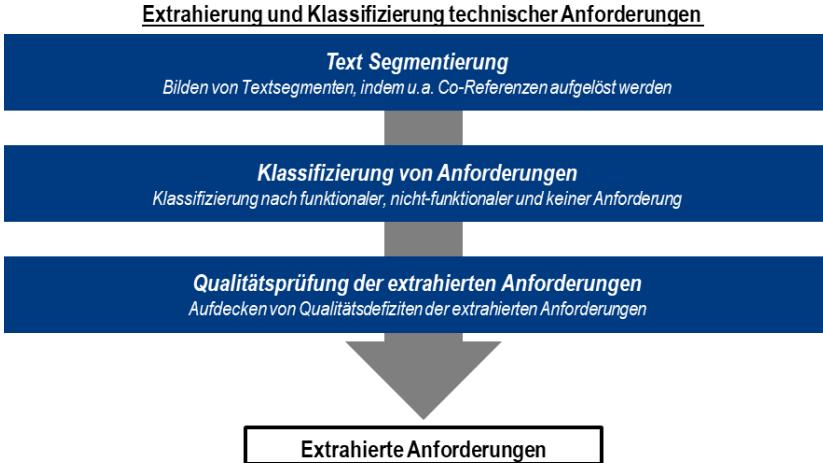


Bild 3-1: Extrahierung von Anforderungen in Anlehnung an [GPP22]

Aspekte, die eine KI-basierte Extrahierung erschweren

In den oben beschriebenen automatisierten Ansätzen wird Künstliche Intelligenz zur Verarbeitung natürlicher Sprache eingesetzt. Es existieren verschiedene Aspekte, die eine KI-basierte Extrahierung erschweren und berücksichtigt werden müssen. Das *Dateiformat* des Lastenhefts muss für die Extrahierung berücksichtigt werden. Beispielsweise werden in .docx-Dokumenten bestimmte Word-Vorlagen genutzt, um Textelemente zu kennzeichnen. In der Text-Segmentierung müssen die entsprechenden Informationen verarbeitet werden. Die Bildung von Textsegmenten wird erschwert aufgrund verschiedener *Formatierungen im Text*: beispielsweise Aufzählungen, die auf einen einleitenden Satz aufbauen. Im Lastenheft werden *Verweise* auf Richtlinien und Normen gesetzt (Querverweise). Co-Referenzen müssen für die Bildung von Textsegmenten aufgelöst werden. Anforderungen können *Qualitätsdefizite* enthalten. Unvollständige oder mehrdeutige Anforderungen führen zu Klassifizierungsfehlern. *Anforderungsbeschreibungen* sind teilweise unstrukturiert. Ein einheitliches Formulierungsschema kann im Gegensatz zu strukturierten Anforderungsdokumenten nicht vorausgesetzt werden. Außerdem existieren Unterschiede in der Formulierung verschiedener *Anforderungsarten*. Die Datengrundlage muss ausreichend groß sein, damit das Modell verschiedene Formulierungen korrekt klassifizieren kann. Zusätzlich müssen ausreichend viele fachspezifische Daten in den jeweiligen *Sprachen* verfügbar sein. Die Aspekte, beispielhafte Ausprägungen und Herausforderungen zur Extrahierung und Klassifizierung von Anforderungen werden in der folgenden Tabelle 3-3 zusammengefasst. [GPP22]

Tabelle 3-3: Aspekte, die eine KI-basierte Extrahierung erschweren [GPP22]

Aspekt	Ausprägung	Herausforderung
Dateiformat	.docx, .pdf, .xlsx	Textelemente erkennen
Formatierung im Text	Aufzählung, Tabelle	Bildung von Textsegmenten
Verweise	Querverweise, Co-Referenzen	Bildung von Textsegmenten
Qualitätsdefizite	Unvollständig, mehrdeutig	Unvollständige Merkmale
Anforderungsbeschreibung	Unstrukturiert, strukturiert	Keine konsistenten Regeln
Anforderungsarten	Funktional, nicht-funktional	Multiklassen-Klassifizierung
Sprache	Deutsch, Englisch	Datengrundlage für mehrere Sprachen benötigt

Die Extrahierung von Anforderungen aus Lastenheften besteht aus der Bildung von Textsegmenten, der Klassifizierung von Anforderungen und einer abschließenden Qualitätsprüfung. Verschiedene Aspekte erschweren die automatisierte Extrahierung von Anforderungen. Lastenhefte und Anforderungsbeschreibungen sind nicht standardisiert, weshalb eine intelligente Verarbeitung der Lastenhefte durch KI-Ansätze unter Berücksichtigung der aufgeführten Aspekte benötigt wird.

Verfügbarkeit von Trainingsdaten in der Praxis

Ein weiterer Aspekt erschwert die Anwendung auf klassische Spezifikationen komplexer technischer Systeme insbesondere in der Praxis: die *Verfügbarkeit von Trainingsdaten* für KI-Modelle zur Klassifizierung, bzw. Klassifizierungsmodelle. Zum Trainieren von Klassifizierungsmodellen wird eine ausreichend große Anzahl an gekennzeichneten domänen spezifischen Daten benötigt. Aufgrund des verwendeten Vokabulars ist es notwendig, domänen spezifische Daten zu verwenden. Andernfalls führt das zu einer niedrigen Leistungsfähigkeit der Modelle. Diese Anzahl an Daten ist in der Praxis oft nicht verfügbar und muss generiert werden [KM17, KO14]. Dies geschieht z. B. durch die manuelle Kennzeichnung von Texten in Spezifikationen – es wird ermittelt, ob der jeweilige Textabschnitt eine Anforderung ist. Dieser Prozess ist sehr zeitaufwändig. Methoden zur Daten-Augmentation haben ein großes Potenzial, den Aufwand für die manuelle Kennzeichnung von Anforderungen zu reduzieren und um domänen spezifische Daten zu generieren [Nau21]. Dabei wird eine niedrige Anzahl an gekennzeichneten Daten benötigt, um künstliche Daten zu erzeugen. [GPB23]

Daten-Augmentation

Zur Identifikation von Methoden zur Daten-Augmentation wurde eine Literaturrecherche durchgeführt. Die Anwendung verschiedener Methoden zur Daten-Augmentation auf Klassifizierungsdatensätze in einer breit angelegten Untersuchung zeigte für die meisten Datensätze und Klassifizierungsmodelle eine Verbesserung der Leistungsfähigkeit im Vergleich zum Training der Modelle ausschließlich mit dem Originaldatensatz [BKR21]. Insbesondere die Methode der Daten-Augmentation, bei der komplett neue Texte generiert werden, anstatt Teile des bestehenden Textes zu verändern (z. B. durch Ersetzen von Synonymen), zeigt erhebliche Verbesserungen der Klassifizierungskennzahlen für reale Datensätze [BKB21]. Dieser Vorteil kommt insbesondere dann zum Tragen, wenn die Variabilität möglicher Satzstrukturen und Formulierungsoptionen im Datensatz nicht repräsentiert ist, da dies nicht durch Modifikation der vorhandenen Texte kompensiert werden kann. Allerdings hat die Generierung neuer Texte mit einem Klassifizierungsmodell in der Regel zwei Nachteile: Abhängig von den Quelldaten des Klassifizierungsmodells kann der generierte Text Verzerrungen und Vorurteile wiedergeben [SBC19]. Außerdem kann die Generierung langer Texte aufgrund der Komplexität von Klassifizierungsmodellen hohe Rechenzeiten verursachen [HDL22]. Beide Nachteile können jedoch im spezifischen Bereich komplexer technischer Systeme vernachlässigt werden, da die Domäne weniger anfällig für soziologische Probleme durch Verzerrungen ist und die Anforderungstexte in der Regel kurz sind. Die Steigerung der Leistungsfähigkeit für domänenspezifische Textklassifizierungsaufgaben durch Daten-Augmentation wurde beispielsweise in einem Experiment in der medizinischen Domäne unter Beweis gestellt [AIV20]. Hierbei wurde unter anderem das Textgenerierungsmodell GPT-2 [RWC19] verwendet. Ein Nachfolger für GPT-2 wurde bereits veröffentlicht, ist aber nicht öffentlich zugänglich. Eine verfügbare Open-Source-Alternative zu GPT-2 ist GPT-J [GPT22]. [GPB22]

3.2 Formalisierung von Anforderungen

Die extrahierten Anforderungen müssen formalisiert werden, um automatisiert von Software-Werkzeugen interpretiert und weiterverarbeitet werden zu können. Relevante Ansätze zur Formalisierung von Anforderungen werden im Folgenden erläutert. Es werden Ansätze zur Formalisierung natürlich-sprachlicher Texte recherchiert. Zusätzlich werden Ansätze recherchiert, welche Ansätze zur Formalisierung natürlich-sprachlicher Texte anwenden, um Anforderungen zu formalisieren.

Formalisierung natürlich-sprachlicher Texte

Für die Formalisierung natürlich-sprachlicher Texte, d. h. die Extraktion unstrukturierter, in Texte eingebetteter Informationen in ein strukturiertes und maschinenlesbares Datenformat, werden die Ansätze in drei Aufgabentypen unterteilt [JJM21]. [GPB23]

- *Relationsextraktion* (engl. „Relation Extraction“): Erkennen von Entitäten und Beziehungen zwischen Entitäten, sodass eine Graphenstruktur, z. B. ein Wissensgraph (engl. „Knowledge Graph“), erstellt werden kann. Die Erstellung des Graphen kann entweder von Grund auf neu erfolgen, oder die gefundenen Entitäten können mit einem bestehenden Graphen verknüpft werden, um diesen zu erweitern.
- *Ereignis-Extraktion* (engl. „Event Extraction“): Identifizierung von Erwähnungen von Ereignissen in Texten, einschließlich der Erkennung der beteiligten Entitäten. Zusätzlich muss der zeitliche Ausdruck des bestimmten Punktes oder Intervalls in der Zeit des Ereignisses extrahiert sowie in ein strukturiertes Format normalisiert werden.
- *Ausfüllen von Vorlagen* (engl. „Template filling“): Eine Vorlage ist definiert als eine Menge von semantischen Schlüsseln und entsprechenden Datentypen. Jeder Schlüssel der Vorlage muss mit Konzepten wie Zeiten, Beträgen oder Ontologie-Entitäten aus dem Ausgangstext gefüllt werden, die den richtigen Datentyp haben und als Wert des Schlüssels mit der erwarteten Semantik des Schlüssels übereinstimmen. Ein Schlüssel mit einem zugewiesenen Wert wird als Schlüssel-Wert-Paar bezeichnet.

Im Kontext der Formalisierung von Anforderungen sind Ereignisse vernachlässigbar. Die Erstellung eines Wissensgraphen durch Relationsextraktion und das Ausfüllen von Vorlagen sind anwendbare Ansätze. Wissensgraphen sind eine umfassende Formalisierung des Textes, in der viele Informationen enthalten sind, während Schlüssel-Wert-Paare eine sehr enge, aber konkrete Formalisierung darstellen. [GPB23, JJM21]

Wenn ein Programm die formalisierten Daten verwendet, um eine Frage zu beantworten, hat ein vollständiger Wissensgraph eine höhere Chance, die Informationen zu enthalten, aber das Programm muss den Graphen durchsuchen und die gefundenen Graphenknoten und -kanten interpretieren, um eine Antwort auf die Frage zu erhalten. Andererseits enthält eine gefüllte Vorlage nur dann die Antwort, wenn zuvor ein Schlüssel für die spezifische Frage definiert wurde, aber in diesem Fall ist keine weitere Verarbeitung erforderlich und der Wert des Schlüssels kann direkt als Antwort ohne weitere Interpretation verwendet werden. Die Auswahl einer Textformalisierungsmethode ist also abhängig von der Frage, wie bekannt die zukünftigen Fragen sind, die über die extrahierten und formalisierten Informationen beantwortet werden sollen. [GPB23, JJM21]

Beiden Aufgaben ist gemein, dass ein Ansatz zur Erkennung von Entitäten notwendig ist, sodass die Entitäten entweder in Bezug auf Beziehungen in Graphen oder als mögliche Werte für Schlüssel analysiert werden können. Entitäten sind Wörter aus dem Ausgangstext, die einer bestimmten Kategorie angehören, z. B. Person, Ort oder Organisation. Daher werden diese Entitäten auch mit dem Begriff „Named Entities“ bezeichnet und die Erkennung wird als „Named Entity Recognition“ (NER) bezeichnet [JJM21]. [GPB23]

Gängige Techniken/Algorithmen zum Trainieren eines NER-Modells auf der Grundlage geeigneter Trainingsdaten sind zum Beispiel die vier Folgenden. [JJM21]

- *Versteckte Markov-Modelle* (engl. „Hidden Markov Models“)
- *Bedingte Zufallsfelder* (engl. „Conditional Random Fields“)
- *Rekurrente Neuronale Netze* (engl. „Recurrent Neural Networks“)
- *Transformer-Modelle* (z. B. BERT)

Ein weit verbreiteter Benchmark für NER-Modelle ist der „CoNLL 2003 NER Task“, bei dem Personen, Orte, Organisationen und verschiedene Entitäten aus Nachrichtenberichten extrahiert werden müssen [STM03]. Aktuell wurde das beste Benchmark-Ergebnis von einem Ansatz erzielt, der transformerbasierte Einbettungen verwendet [WJB21]. [GPB23]

Ansätze zur Formalisierung von Anforderungen

SHWETA, SANYAL und GHOSHAL erstellen ein Klassen-Diagramm von Anforderungen aus einer semi-formalen Use-Case Beschreibung [SG18]. BTOUSH und HAMMAD erstellen ein Entity-Relationship-Modell eines Software-Systems aus einer natürlichsprachlichen Software-Spezifikation durch die Anwendung von NLP-Techniken [SM15]. MORE und PHALNIKA entwickeln eine Methode zur Erstellung von UML-Modellen aus natürlichsprachlichen Anforderungen unter Verwendung von NLP-Techniken wie POS-Tagging und Domänen Ontologien [MP12]. Ein anderer Ansatz wird im ARSENAL-Tool gewählt, das einen Graphen mit Beziehungen und Abhängigkeiten zwischen Entitäten verwendet, um eine Darstellung der Anforderungen als Formeln mit Linearer Temporaler Logik (LTL) als Formalismus zu erstellen [GEL14]. Andere Ansätze zielen ebenfalls darauf ab, natürlichsprachliche Anforderungen in einen logischen Formalismus mit einer eigenen Formalismus-Syntax anstelle von LTL und mit verschiedenen Transformationsmethoden anstelle von Graph-Transformationen zu übertragen, z. B. ein hierarchisches Clustering zur Strukturierung erkannter Entitäten und Beziehungen [GPM21b] oder ein regelbasierter Ansatz auf der Grundlage der gefundenen Entitäten und grammatischen Abhängigkeiten [KGG21]. In Abhängigkeit der Anforderungsart definiert RUPP Anforderungsschablonen [Rup16]. Beispielsweise besteht die Schablone für funktionale Anforderungen aus den Bestandteilen: Das [System] muss/sollte/wird dem [Akteur] die Möglichkeit bieten/fähig sein [Objekt] [Prozesswort]. Beispiel: „Der Staubsaugerroboter muss fähig sein, Tierhaare mit einer Länge von 3 mm aufzunehmen.“. Solch ein Bestandteil der Anforderungsschablone – z. B. Prozesswort – entspricht einem Attribut im Metadaten-Modell der Anforderung. Im DFG-geförderten Projekt WiRPro werden Produktnutzungs-Informationen, die durch „Condition Monitoring“ gesammelt werden, für die Entwicklung einer neuen Produktgeneration formalisiert [AF09]. In einem weiteren DFG-geförderten Projekt werden durch statistische Analysen der Nutzungsdaten entwicklungsrelevante Informationen generiert und mit Modellen des Systems verknüpft, um Produktvarianten zu generieren (Gestaltevolution durch algorithmisierte Informationsrückführung aus dem Produktlebenszyklus – DFG GaIP [LM13]). [GPB23]

Für die Formalisierung von Anforderungen ist besonders die Relationsextraktion sowie das Ausfüllen von Vorlagen relevant. Aufgrund des hohen Grads der Formalisierung eignen sich insbesondere Ansätze zum Ausfüllen von Vorlagen. Es müssen Schlüssel-Wert-Paare in Anforderungen erkannt werden. Hierzu muss u. a. eine Named Entity Recognition (NER) durchgeführt werden.

3.3 Abhängigkeitsanalyse von Anforderungen

Zur Identifikation und Auswahl von Ansätzen zur Abhängigkeitsanalyse von Anforderungen werden auf Basis eines Industrieworkshops im Kontext des ARCA Projekts mit IAV Stakeholder-Anforderungen (im Sinne von [WRF15]) aufgenommen [GPO20]. Die Workshops wurden vorbereitet, indem relevante Ansätze zur Abhängigkeitsanalyse von Anforderungen recherchiert worden sind. Durch den Input der Industrievertreter soll die Anwendbarkeit der Ansätze in der Praxis gewährleistet werden. Zur Auswahl relevanter Ansätze, die in der Literaturanalyse identifiziert worden sind, wird eine Nutzwertanalyse durchgeführt. Die Kriterien der Nutzwertanalyse werden auf Basis der Anforderungen erarbeitet und in einem zweiten Workshop mit sechs Industrievertretern diskutiert und gewichtet. Die sechs Industrievertreter sind bei IAV tätig. Sie haben die folgenden Positionen: Abteilungsleiter, Projektleiter, Teamleiter, Prozess-Manager, Anforderungsingenieur und Entwickler. Es wird ein paarweiser Vergleich der Kriterien durchgeführt, um die Priorität zu ermitteln. Dies erfolgt ausschließlich durch die Industrievertreter, um die Bedeutung der Kriterien aus Anwender-Perspektive zu berücksichtigen. Anschließend wird unter Berücksichtigung der Anwender-Priorisierung sowie der Anzahl und Priorität der zugrunde liegenden Anforderungen eine qualitative Gewichtung für jedes Kriterium festgelegt. Um unterschiedliche Rollen in der Entwicklung zu berücksichtigen, werden für beide Workshops dieselben Industrievertreter aus allen für die Anwendung relevanten Aufgabenfeldern beteiligt [GPO20].

Bewertungskriterien

Die Kriterien werden inhaltlich anhand der *vier Kategorien* Input/Datengrundlage, Output/Ergebnisse, Anwendung und Aufwand gruppiert. Die Bewertungskriterien sind überschneidungsfrei und unabhängig von Aspekten der Nutzerfreundlichkeit. [GPO20]

Der Lösungsansatz soll *Expertenwissen (K1.1)* bei der Anwendung berücksichtigen. Dies erlaubt eine Anwendbarkeit in frühen Entwicklungsphasen ohne umfassende Datenbasis und stellt eine Möglichkeit dar, das Erfahrungswissen der Anwender einzubeziehen. Dieses implizite Wissen der Experten ist häufig die wichtigste Entscheidungsgrundlage in Unternehmen. Da es neben Erfahrungswerten auch auf Fachwissen beruht, ermöglicht es zudem die Identifikation von Abhängigkeiten, welche einer rein retrospektiven Bewertungsgrundlage unzugänglich sind. Um bereits *früh in der Entwicklung (K1.2)* das Änderungsrisiko von Anforderungen beurteilen zu können, muss die eingeschränkte

Datengrundlage, die zu diesem Zeitpunkt vorliegt (insbesondere natürlich-sprachliche Anforderungstexte), ausgewertet werden können. [GPO20]

Für den Output bzw. die Ergebnisse werden drei Kriterien unterschieden. Erstes Kriterium ist die *Differenzierung der Abhängigkeiten* (K2.1) nach Intensität und Art. Hierarchische oder kausale Abhängigkeiten (z. B. naturwissenschaftliche Gesetzmäßigkeiten) führen beispielsweise zwangsläufig zu einer Propagation der Änderung, wenn auch mit unterschiedlichen Auswirkungen. Weniger ausgeprägte Abhängigkeitsarten (z. B. „konfliktär“) hingegen lassen nur bedingt Rückschlüsse auf das Propagationsverhalten zu und erfordern eine kontextspezifische Analyse. Auch wenn eine Abhängigkeit vorliegt, muss diese nicht zwangsläufig zu einer Änderungspropagation führen. Ein weiteres Kriterium ist die *Lernfähigkeit* (K2.2) des Ansatzes. Dieses Kriterium beschreibt, ob die Ergebnisgüte des Ansatzes mit wiederholter Anwendung zunimmt. Ein Ansatz gilt als *erprobt* (K2.3), wenn dieser für die Anwendung in interdisziplinären Entwicklungsprojekten regelmäßig Anwendung findet. [GPO20]

Die *Transferfähigkeit* (K3.1) beschreibt, ob der Ansatz auch auf andere Anwendungsfälle übertragbar ist. Die Ansätze werden hauptsächlich in der Software-Entwicklung angewandt. Der Aufwand zur Übertragbarkeit wird nicht betrachtet, sondern ausschließlich die Machbarkeit. Der Lösungsansatz ist *nachvollziehbar* (K3.2), wenn es für den Anwender transparent ist, wie die Ergebnisse gebildet wurden und diese auf kausalen Zusammenhängen beruhen. Dies ist wichtig für die Akzeptanz und Weiterverwendung der Ergebnisse. Die *Robustheit* (K3.3) beschreibt, inwiefern die Ergebnisse sich ändern, wenn bestimmte Randbedingungen nicht eingehalten wurden. Dies bezieht sich vor allem auf das Format des Inputs. Viele der Ansätze wenden Techniken zur natürlichen Sprachverarbeitung an. Ein Ansatz ist z. B. robust, wenn er keine kontrollierte Sprache oder die Einhaltung einer Vorlage verlangt, um eine hohe Ergebnisgüte zu erzeugen. [GPO20]

Ein *hoher Automatisierungsgrad* (K4.1) ist notwendig, um umfangreiche Anforderungssets mit dem Lösungsansatz bearbeiten zu können, ohne dass ein hoher Anwendungsaufwand für den Nutzer resultiert. Dies ist besonders für komplexe interdisziplinäre Entwicklungsprojekte relevant. Das Kriterium *Datenvorbereitung* (K4.2) beschreibt, wie hoch der Aufwand ist, um den Ansatz anwenden zu können. [GPO20]

Die verschiedenen Lösungsansätze zur Erkennung von Anforderungsabhängigkeiten werden hinsichtlich der Differenzierung der Anforderungen geordnet. Es wird zwischen Ansätzen unterschieden, die eine differenzierte, semi-differenzierte und singuläre Betrachtung vornehmen. Ansätze zur differenzierten Betrachtung bestimmen die Art der Abhängigkeit zwischen Anforderungen. Ansätze zur semi-differenzierten Betrachtung bestimmen den Grad der Abhängigkeit zwischen Anforderungen. Ansätze zur singulären Betrachtung bestimmen, ob eine bestimmte Abhängigkeitsart vorliegt. [GPO20]

Ansätze zur differenzierten Betrachtung von Abhängigkeitsarten

Bei den Ansätzen zur differenzierten Berücksichtigung von Abhängigkeitstypen wird zwischen *ontologiebasierten Ansätzen* und maschinellen Lernansätzen unterschieden. Bei sogenannten ontologiebasierten Ansätzen wird das Expertenwissen in einer Wissensbasis formalisiert. Dabei werden Entitäten mit bestimmten Abhängigkeitstypen vernetzt [MBP19, ZH17]. Da ontologiebasierte Ansätze einen hohen Erstellungsaufwand haben und Wissen voraussetzen, das in frühen Entwicklungsstadien nicht ausreichend vorhanden ist, werden sie nicht weiter betrachtet. [GPO20]

In Ansätzen, die Techniken des *Maschinellen Lernens* verwenden [ASF18, DAR19, SSA19], werden Klassifizierungsmodelle mit dem Lerntyp (schwaches) überwachtes Lernen trainiert. Input ist die textuelle Beschreibung von Anforderungen. Beim überwachten Lernen wird der Output vom Experten für das Training bewertet (K1.1). Nachteilig ist, dass eine große Menge an Daten notwendig ist, um die Klassifizierungsmodelle zu trainieren. Die frühzeitige Bereitstellung solcher Daten kann zudem nur gewährleistet werden, wenn das Entwicklungsprojekt ähnlich zu vergangenen Projekten ist und das Klassifizierungsmodell mit historischen Daten trainiert werden kann (K1.2). Eine größere Datengrundlage führt zu einer höheren Ergebnisgüte (K2.2). Entwicklungsprojekte besitzen einen hohen Neuheitsgrad, wodurch die Übertragbarkeit eines Klassifizierungsmodells, das für ein anderes Entwicklungsprojekt trainiert wurde, auf ein weiteres Projekt jedoch nicht möglich ist (K3.1) oder zu einer schlechten Ergebnisgüte führen kann. Ansätze des Maschinellen Lernens basieren auf probabilistischen Methoden und sind daher in der Regel nicht nachvollziehbar (K3.2). Anforderungen können als textuelle Beschreibungen ohne spezielle Restriktionen verarbeitet werden (K3.3). Der Output der Ansätze ist die Klassifizierung der Art der Abhängigkeit zwischen den Anforderungen und wird automatisiert berechnet (K4.1). Der Aufwand zur Anwendung eines Klassifizierungsmodells, welches bereits trainiert worden ist, ist gering (K4.2). [GPO20]

In der Literaturrecherche wurde der Fokus auf Ansätze mit hohem Automatisierungsgrad gelegt, die für eine hohe Anzahl an Anforderungen anwendbar sind (siehe Abschnitt 2). Aus diesem Grund wird nur ein einzelner, exemplarischer manueller Ansatz betrachtet. Ein Beispiel präsentieren KÖHLER ET AL. Sie prüfen, welche Eigenschaften des Produkts von einem Änderungsantrag abhängig sind [KCW08]. Anschließend wird anhand einer manuell erstellten Abhängigkeitsmatrix geprüft, welche Merkmale des Produkts die Eigenschaften beeinflussen. Daraus wird ersichtlich, welche Merkmale durch den Änderungsbedarf betroffen sind. [GPO20]

Ansätze zur semi-differenzierten Betrachtung von Abhängigkeitsarten

HEIN ET AL. berechnen semantische Ähnlichkeiten zwischen Anforderungen [HVM18]. Semantisch ähnliche Anforderungen werden als abhängig klassifiziert und in einer Matrix repräsentiert. Sie gewichten den Grad der Abhängigkeit zwischen den Anforderungen anhand der Pfadlänge. Es ist keine präzise Aussage zur Änderungspropagation möglich.

Ursache-Wirkungs-Beziehungen von Anforderungsänderungen werden nicht erkannt. Aus diesem Grund wurde nur ein exemplarischer Ansatz ausgewählt, der typische Charakteristika semi-differenzierter Ansätze vereint und diese Art von Ansätzen nicht weiter betrachtet. [GPO20]

Ansätze zur singulären Betrachtung von Abhängigkeitsarten

Weiterhin existieren Ansätze, die einzelne Abhängigkeitsarten detektieren und keinen übergreifenden Ansatz anstreben. HAMDAQA hat einen Ansatz zur Erkennung externer Querverweise entwickelt [HH11]. Es werden automatisiert Textstellen hervorgehoben. OCH DAG ET AL. verwenden lexikalische Analysen in Kombination mit „Stemming“ und der Entfernung von Stopp-Wörtern, um die Ähnlichkeiten von Anforderungen untereinander zu bestimmen [NRC02]. Sie setzen dabei Ähnlichkeit mit Abhängigkeit gleich. MARTINEZ ET AL. verwenden ebenfalls semantische Ähnlichkeiten, um Abhängigkeiten zu erkennen. Sie vergleichen Anforderungsbeschreibungen mit „Use-Cases“ und Szenarien [MCG19]. PARK ET AL. berechnen die Ähnlichkeit zwischen Anforderungssets, um Redundanzen und Inkonsistenzen zu identifizieren, und extrahieren die möglichen mehrdeutigen Anforderungen [PKK00]. Die Ähnlichkeitsmessmethode kombiniert ein Schiebefenster- und ein „Parsermodell“. Es existieren weitere Ansätze, um Inkonsistenzen in Anforderungssets zu detektieren [Mis16, ZJ05]. DI THOMAZZO ET AL entwickeln einen Ansatz zur automatischen Erstellung einer „Requirements Traceability Matrix“ (RTM) auf der Grundlage von „Fuzzy“-Logik [DRO13]. Um die Abhängigkeit zwischen funktionalen Anforderungen zu bestimmen, werden die Frequenzvektor- und die Cosinus-Ähnlichkeitsmethode verwendet. Abadi et al. vergleichen die Effektivität verschiedener „Information Retrieval“-Techniken zum Auffinden von „Traceability-Links“ vom Code zur Dokumentation (Anforderungen, Benutzerhandbücher, ...) [ANS08]. [GPO20]

Auswertung der Ergebnisse

In der Matrix in Anhang A2.2 wird die Bewertung der Ansätze nach den erarbeiteten Kriterien visualisiert. Anhand der Gewichtung der Kriterien und der Bewertung wird der Nutzwert berechnet. Umso höher der Nutzwert ist, umso besser erfüllt der Ansatz die Kriterien. Die Nutzwerte der Ansätze werden zusammenfassend in Tabelle 3-4 dargestellt. Die Nutzwertanalyse zeigt, dass Ansätze, die eine differenzierte Betrachtung vornehmen, einen besonders hohen Nutzen haben. Zu diesen Ansätzen gehören Ansätze des maschinellen Lernens. [GPO20]

Tabelle 3-4: Nutzwerte der Ansätze der verschiedenen Gruppen [GPO20]

Gruppe	Ansätze	Nutzwert
Differenzierte Ansätze: Maschinelles Lernen	[DAR19], [ASF18], [SSA19]	61 %, 55 %, 55 %
Differenzierte Ansätze: Manuell	[KCW08]	55 %
Semi-differenzierte Ansätze	[HVM18]	60 %
Singuläre Ansätze	[Mis16], [DRO13], [PKK00], [NFS14], [LGN07], [MCG19]	55 %, 55 %, 53 %, 51 %, 49 %, 46 %

Besonders relevant für die automatisierte Abhängigkeitsanalyse von Anforderungen sind Ansätze des maschinellen Lernens. Die Vorteile dieser Ansätze sind der hohe Automatisierungsgrad und die Robustheit. Propagationseffekte höherer Ordnung und umfangreiche Anforderungslisten können effizient analysiert werden. Ein Nachteil ist, dass eine hohe Anzahl an Trainingsdaten benötigt wird, um eine hohe Leistungsfähigkeit in der Abhängigkeitsanalyse von Anforderungen zu erreichen.

Zusammenfassung: Stand der Forschung

Zur *Extrahierung von Anforderungen* aus Lastenheften existieren KI-basierte Ansätze zur Extrahierung von Textsegmenten und zur Klassifizierung nach Anforderungsart. Diese Ansätze besitzen einen hohen Automatisierungsgrad und sind deshalb geeignet, um eine Vielzahl an Anforderungen zu extrahieren. Aufgrund der Verfügbarkeit von Trainingsdaten in der Praxis ist die Anwendbarkeit dieser Ansätze eingeschränkt. Zur effizienten Generierung von Trainingsdaten für KI-Modelle existieren Ansätze zur Daten-Augmentation, z. B. GPT-J. Die extrahierten Anforderungen müssen formalisiert werden, um automatisiert von Software-Werkzeugen interpretiert und weiterverarbeitet werden zu können. Ansätze zum Ausfüllen von Vorlagen besitzen für die *Formalisierung von Anforderungen* aufgrund der engen Formalisierung einen hohen Mehrwert, um relevante Parameter aus Anforderungen zu extrahieren. Solch eine enge Formalisierung wird durch Schlüssel-Wert-Paare realisiert: z. B. „Masse des Bauteils“ und der dazugehörige Wert „max. 50 kg“. Zur Erkennung von Schlüssel-Wert-Paaren sind NER-Modelle geeignet. NER-Modelle ermöglichen beispielsweise die Erkennung der Phrase „Robert Bosch GmbH“ innerhalb eines Texts und die automatisierte Klassifizierung als „Unternehmen“. Ansätze des maschinellen Lernens sind aufgrund des hohen Automatisierungsgrads für die *Abhängigkeitsanalyse* einer Vielzahl von Anforderungen geeignet. Allerdings wird eine hohe Anzahl an Trainingsdaten benötigt, um eine hohe Leistungsfähigkeit für die Klassifizierung von Abhängigkeiten zu erreichen. Es wird ein Ansatz benötigt, um effizient Daten für das Training von Modellen des maschinellen Lernens zu generieren.

4 Feldstudie zur Untersuchung des Anwendungszusammenhangs

Zur Untersuchung des Anwendungszusammenhangs wurde eine Feldstudie zum proaktiven Management von Anforderungsänderungen in der Entwicklung komplexer technischer Systeme durchgeführt. Ziel der Feldstudie war es, eine Methodik für das Risikomanagement von Anforderungsänderungen zu entwickeln, die frühzeitig in der Entwicklung komplexer technischer Systeme angewendet werden kann. Hierzu wurden verschiedene Ansätze zur Verarbeitung natürlicher-Sprache im Kontext der Abhängigkeitsanalyse von Anforderungen untersucht. Es wurden *Erkenntnisse zur Abhängigkeitsanalyse von Anforderungen sowie zur Risikoberechnung von Anforderungsänderungen* gewonnen, welche in die Methodik-Entwicklung einfließen sollen. Es wird die Ursache von Projektfehlschlägen durch Anforderungsänderungen adressiert. Die Feldstudie wurde im Rahmen des öffentlich geförderten BMBF-Projekts ARCA („Automated Requirement Change Analysis of Complex Technical Systems“) als Initiative des Software Campus durchgeführt und dauerte zwei Jahre. Das Projekt bestand aus vier Schritten [GOP22]:

- 1) Problemanalyse und Ermittlung von Anforderungen an die Methode
- 2) Entwicklung einer Methode für das proaktive Management von Anforderungsänderungen
- 3) Implementierung der Methoden in einem Software-Prototypen
- 4) Validierung der Methode in Zusammenarbeit mit Industrievertretern

Die vier Schritte des ARCA Projekts werden in Anhang A2.3 detailliert dargestellt. Die Unterlagen zur Zusammenarbeit mit den Industrievertretern sind in Anhang A3.1 dargestellt. Kernergebnis des Projektes ist die ProMaRC-Methodik (siehe Anhang A2.4).

4.1 Erkenntnisse zur Abhängigkeitsanalyse von Anforderungen

Besonders die Abhängigkeitsanalyse von Anforderungen wurde während der Feldstudie untersucht. Um den besten Ansatz für die Abhängigkeitsanalyse zu finden, wurde ein systematischer Vergleich von Ansätzen des maschinellen Lernens durchgeführt [GOH22a]. Speziell für BERT zeigen frühere Forschungen ein hohes Potenzial hinsichtlich der Leistungsfähigkeit und der Möglichkeit, Abhängigkeitstypen zu unterscheiden [AAV21, PMM21]. Das von VASWANI ET AL. [VSP17] entwickelte BERT erzielt in verschiedenen Anwendungskontexten gute Ergebnisse [GG20]. Aus diesem Grund wurde eine Vergleichsstudie durchgeführt, um die Leistungsfähigkeit von BERT für die Abhängigkeitsanalyse von Anforderungen zu ermitteln.

Als Benchmark für den Vergleich mit alternativen Ansätzen werden die etablierten statistischen Ansätze Multinomiale Logistische Regression (MLR) und „Support Vector Machine“ (SVM) sowie „Recurrent Neural Network“ (RNN) verwendet. Die Fallbeispieldaten wurden auf Basis des Open-Source-Roboterprojekts "BCN3D Moveo" erstellt, um einen geeigneten Datensatz für den Vergleich zu haben¹. Der Datensatz umfasst 145 Anforderungen, die manuell durch eine Abhängigkeitsmatrix mit 21.025 Einträgen angereichert wurden. Diese werden für den Vergleich in einen Trainingssatz (16.820 Einträge) und einen Validierungssatz (4205 Einträge) aufgeteilt (80/20).

Kriterien für den Vergleich sind die Präzision (engl. „Precision“), die Sensitivität (engl. „Recall“) und die daraus resultierende Kennzahl F1. Die Präzision gibt die Anzahl der vorhergesagten richtig positiven Ergebnisse im Vergleich zu den richtig und falsch positiven Klassen an. Eine Präzision nahe bei 1 weist auf zuverlässige Modelle hin. Die Sensitivität gibt den Anteil der richtig-positiv klassifizierten Ergebnisse an den gesamten positiven Ergebnissen an. F1 ist der harmonisierte Durchschnitt zwischen Präzision und Sensitivität. F1 misst das Ausmaß, in dem die Klassen ausgewogen verteilt sind, und bewertet somit die Gesamtqualität des Modells [Alp22]. Da der Datensatz eine Hauptklasse ("Keine") mit vielen Einträgen und mehrere Nebenklassen mit wenigen Einträgen (z. B. "Benötigt") enthält, mussten die Kriterien als Makro-Durchschnitt betrachtet werden (gleiches Gewicht für alle Klassen), um aussagekräftig zu sein (siehe Tabelle 4-1).

Tabelle 4-1: Vergleich der Ergebnisse für die Erkennung von Anforderungs-abhängigkeiten mit verschiedenen Modellen [GOH22a]

Kriterien	MLR	SVM	RNN	BERT
Präzision (M.D.)	20,47 %	24,00 %	22,07 %	54,10 %
Sensitivität (M.D.)	21,30 %	23,43 %	60,92 %	56,98 %
F1 (M.D.)	6,29 %	23,92 %	12,92 %	55,12 %

M.D.: Makro-Durchschnitt

Der Vergleich zeigt, dass die Präzision, die Sensitivität und F1 von BERT für ein Mehrklassen-Klassifikationsproblem vielversprechend sind (vergleiche [PMM21]). Besonders der hohe F1 von BERT (55,12 %) zeigt, dass die Abhängigkeitsanalyse zuverlässig ist. Daher wurde BERT ausgewählt, um im Detail untersucht zu werden.

Vorstudie zur Evaluation der Abhängigkeitsanalyse mit BERT

Die Methode zur automatisierten Anforderungsabhängigkeitsanalyse wurde in einem vierstufigen Ansatz evaluiert. Zunächst werden Anforderungen aus einem Fallbeispiel zur Entwicklung eines intelligenten Handhabungsroboters analysiert, um die Leistungsfähigkeit der Abhängigkeitsanalyse zu bestimmen. Weiterhin werden Anforderungen aus

¹ <https://github.com/BCN3D/BCN3D-Moveo>

einem interdisziplinären Entwicklungsprojekt eines Entwicklungsdienstleisters untersucht. Zweitens wird die Transferfähigkeit evaluiert, indem das ursprüngliche Modell (trainiert mit Daten des intelligenten Knickarmroboters) verwendet wird, um die Anforderungen von vier studentischen Entwicklungsprojekten von Elektrowerkzeugen zu klassifizieren und die Leistungsfähigkeit zu bestimmen. Drittens wird das Modell sowohl mit den Daten des intelligenten Knickarmroboters als auch mit den Daten der studentischen Entwicklungsprojekte trainiert und auch mit den Daten dieser Projekte getestet, um die Leistungsfähigkeit bei der Verwendung heterogener Daten für das Training zu untersuchen. Viertens wird die Anwendbarkeit und Nützlichkeit der Methode und des Software-Werkzeugs durch Pilotanwender aus der Industrie bewertet.

Zur *Bewertung der Leistungsfähigkeit der Anforderungsabhängigkeitsanalyse* werden Anforderungen aus dem Fallbeispiel des intelligenten Knickarmroboters "BCN3D Moveo" verwendet. Zur Bewertung werden die Kennzahlen Präzision, Sensitivität und F1 für die jeweiligen Abhängigkeitstypen ermittelt (siehe Tabelle 4-2).

Tabelle 4-2: Ergebnisse für den intelligenten Knickarmroboter [GOH22a]

Abhängigkeiten	Anzahl	Präzision	Sensitivität	F1
„keine“	3081	98,70 %	98,44 %	98,57 %
„spezifiziert“	15	38,89 %	46,67 %	42,42 %
„wird spezifiziert“	15	38,46 %	33,33 %	35,71 %
„benötigt“	31	44,44 %	38,71 %	41,38 %
„wird benötigt“	31	50,00 %	67,74 %	57,53 %
M.D..	-	54,10 %	56,98 %	55,12 %
G.D.	-	97,12 %	97,01 %	97,05 %

M.D.: *Makro-Durchschnitt*, G.D.: *Gewichteter Durchschnitt*

Die Ergebnisse zeigen, dass "keine" Abhängigkeiten mit einer hohen Leistungsfähigkeit korrekt klassifiziert werden (F1: 98,57 %). Zur Unterscheidung verschiedener Abhängigkeitstypen ist das Modell nur bedingt geeignet (Makro-Durchschnitt F1: 55,12 %). Daher wird die Leistungsfähigkeit *ohne Unterscheidung der Abhängigkeitstypen* untersucht. Alle Abhängigkeitstypen werden als "abhängig" eingestuft (siehe Tabelle 4-3), was legitim ist, da sie alle auf eine zwangsläufige Propagation hinweisen [GOP21].

Tabelle 4-3: Ergebnisse: Abhängigkeitsarten werden nicht unterschieden [GOH22a]

Abhängigkeiten	Anzahl	Präzision	Sensitivität	F1
keine	3081	99,12 %	99,06 %	99,09 %
abhängig	92	69,15 %	70,65 %	69,89 %
M.D.	-	84,14 %	84,86 %	84,49 %
G.D.	-	98,25 %	98,24 %	98,24 %

M.D.: *Makro-Durchschnitt*, G.D.: *Gewichteter Durchschnitt*

Der Makro-Durchschnitt von F1 liegt bei 84,49 %, was zeigt, dass die Leistungsfähigkeit bei der Erkennung von Abhängigkeiten höher ist, wenn nicht zwischen verschiedenen Arten von Abhängigkeiten unterschieden wird. Der Grund für diesen Effekt ist, dass die Daten für die Feinabstimmung des BERT-Modells heterogener sind, was die Probleme der Überanpassung (engl. „overfitting“) verringert. Um die Leistungsfähigkeit zu steigern und gleichzeitig genügend Informationen über das Ausbreitungsverhalten zu erhalten, wird bei den folgenden Untersuchungen nicht mehr zwischen den Abhängigkeitstypen unterschieden.

Um die *Transferfähigkeit* von fein abgestimmten Modellen zu bewerten, wird ein Fallbeispiel von vier Studentenprojekten zur Entwicklung von Elektrowerkzeugen (z. B. Akkuschrauber) untersucht. Aufgrund der verschiedenen Produkte sind die Anforderungsdaten heterogen. Die Anforderungen dieser Studentenprojekte werden mit Hilfe des BERT-Modells analysiert, das mit Daten des intelligenten Handhabungsroboters feinabgestimmt wurde. Für die Berechnung der Kennzahlen werden die Ergebnisse der vier Projekte kumuliert (siehe Tabelle 4-4).

Tabelle 4-4: Ergebnisse: Transferfähigkeit des BERT-Modells [GOH22a]

Abhängigkeiten	Anzahl	Präzision	Sensitivität	F1
keine	4291	86,79%	99,39%	92,67%
abhängig	655	18,75%	0,92%	1,75%
M.D.	-	52,77%	50,16%	47,21%
G.D.	-	77,78%	86,35%	80,63%

M.D.: Makro-Durchschnitt, G.D.: Gewichteter Durchschnitt

"Keine" Abhängigkeiten werden mit einer hohen Leistungsfähigkeit klassifiziert (F1-Wert: 92,67 %), während "abhängige" Anforderungen mit einer niedrigen Leistungsfähigkeit klassifiziert werden (F1-Wert: 1,7 %). Die Ergebnisse zeigen, dass die Leistungsfähigkeit des Modells abnimmt, wenn das Modell nicht anhand von Daten aus dem untersuchten Projekt feinabgestimmt wird, bzw. die Anforderungsdaten heterogen sind (Makro-Durchschnitt F1 von 47,21 % vs. 84,49 %, siehe Tabelle 4-3).

Die Vorstudie zeigt, dass BERT im Vergleich zu anderen Modellen die höchste Leistungsfähigkeit für die Abhängigkeitsanalyse besitzt. Durch die Reduzierung der Abhängigkeitsarten von drei auf zwei Klassen wird die Ergebnisgüte höher. Für die Propagationsanalyse ist die Betrachtung von zwei Klassen ausreichend. Die Ergebnisgüte für die Transferfähigkeit des trainierten BERT-Modells ist niedrig. Vortrainierte Modelle können nur begrenzt für die Klassifizierung heterogener Daten angewendet werden. Für die industrielle Anwendbarkeit ist es notwendig, dass heterogene Daten aufwandsarm klassifiziert werden können. Besonders bei der Klassifizierung von Daten aus neuen Projekten durch BERT-Modelle besteht im Kontext der Abhängigkeitsanalyse Forschungsbedarf.

Aktives Lernen für die Analyse von Anforderungsabhängigkeiten

Das Training von Modellen durch aktives Lernen besitzt für die Klassifizierung von Daten aus neuen Projekten Potenzial. Aktives Lernen ist eine Form des maschinellen Lernens, bei der ein Lernalgorithmus interaktiv ein Orakel (typischerweise einen menschlichen Experten) befragt, um die gewünschte Kennzeichnung für neue Daten zu erhalten [Set09]. Es hat sich als effektiv erwiesen, um den menschlichen Aufwand bei der Datenanalyse zu reduzieren [ASN19, DPA18]. Es existieren mehrere Abfragestrategien, um auszuwählen, welche Daten gekennzeichnet werden sollen. Bei der Verwendung von „Least Confidence Uncertainty Sampling“ werden beispielsweise die Daten mit der höchsten Unsicherheit bei der Klassifizierung durch das Orakel für die Kennzeichnung ausgewählt [Set09]. Dies bedeutet, dass nur eine Teilmenge der Daten vom Experten manuell klassifiziert werden muss. Die restlichen Daten werden automatisiert durch das Modell klassifiziert.

4.2 Erkenntnisse zur Risikoberechnung von Anforderungsänderungen

Als übergreifende Methode für die Evaluierung der Risikoberechnung von Anforderungsänderungen wurde die Erprobung in einem industriellen Kontext mit anschließenden Bewertungsbögen und Interviews gewählt. Nach ULRICH [Ulr81] ist es notwendig, die Evaluation im vorgesehenen Anwendungskontext durchzuführen. Daher wurde sie ohne zusätzliche Unterstützung und auf der Grundlage realer Entwicklungsdaten durchgeführt. Um eine Pilotnutzung zu ermöglichen, wurde den Pilotanwendern das Software-Werkzeug sowie ein Benutzerhandbuch und ein Demonstrationsvideo zur Verfügung gestellt. Die Pilotanwender haben die ProMaRC-Methodik für ein Projekt angewendet. Die Pilotanwender wurden nach ihrem Hintergrund und ihrer Erfahrung in der Anforderungsentwicklung ausgewählt. Um die Einschätzung der Pilotanwender zu objektivieren und emotionale Voreingenommenheit zu reduzieren, wurde ein Bewertungsbogen für die anfängliche Bewertung verwendet. Um über den Bewertungsbogen hinausgehendes detailliertes Feedback und Bewertungssaspekte zu ermöglichen, wurden anschließend mit jedem Pilotanwender halbstrukturierte Interviews (30 bis 45 Minuten) geführt.

Ein Kriterium, welches von den Pilotanwendern bewertet werden sollte, ist die *Verfügbarkeit von Informationen*, die für die Anwendung der ProMaRC-Methodik notwendig sind (siehe Tabelle 4-5). Das Feedback der Pilotnutzer zeigt überwiegend die Verfügbarkeit aller erforderlichen Informationen: Der Medianwert für alle Fragen ist "verfügbar". Ausgehend von den rollenspezifischen Verantwortlichkeiten wurde teilweise angegeben, dass die Informationen nicht verfügbar sind. Anforderungsingenieure antworten, dass Informationen zu Änderungsinitiatoren von Anforderungen teilweise nicht verfügbar sind. Insgesamt wurden drei Pilotanwender befragt. Zwei der Anwender haben angegeben, dass Informationen zu den Änderungsinitiatoren nicht verfügbar sind.

Außerdem wurde angemerkt, dass in der Änderungshistorie in der Praxis häufig nicht die Ursache der Änderung inklusive des zugehörigen Änderungsinitiators dokumentiert wird.

Da Informationen zu Änderungsinitiatoren häufig nicht in der Praxis verfügbar sind, besteht ein Bedarf nach einer Berechnung der Änderungswahrscheinlichkeit von Anforderungsänderungen, für deren Anwendung diese Informationen nicht benötigt werden.

Tabelle 4-5: Auswertung der Bewertungsbögen im zur Kategorie "Verfügbarkeit von Informationen" [GOP22]

Verfügbarkeit von Informationen	verfügbar/nicht verfügbar
Natürlich-sprachliche Anforderungen	3/0
Priorität der Anforderungen	3/0
Änderungsinitiatoren	3/0
Änderungen durch Änderungsinitiator im laufenden Projekt	1/2
Änderungen durch Änderungsinitiator im vergangenen Projekt	1/2
Korrektheit der Anforderungsabhängigkeiten	2/1
Änderungsauswirkungen	2/1
Anwendbarkeit der ausgewählten Risikosteuerungsmaßnahmen	2/1

Zusammenfassung der Vorstudie

Es wurde die *Transferfähigkeit von BERT-Modellen* untersucht. Die Leistungsfähigkeit für die Klassifizierung heterogener Anforderungsdaten ist niedrig. Für die industrielle Anwendbarkeit muss die Leistungsfähigkeit hoch sein und Modelle müssen effizient generiert werden können. Besonders bei der Klassifizierung von Daten aus neuen Projekten durch BERT-Modelle besteht im Kontext der Abhängigkeitsanalyse Forschungsbedarf.

Außerdem wurden Industrievertreter zum Risikomanagement von Anforderungsänderungen befragt. Die Befragung hat ergeben, dass Informationen zu Änderungsinitiatoren in der Praxis teilweise nicht verfügbar sind. Diese Informationen werden laut Industrievertretern in der Entwicklung von Automobilen häufig nicht dokumentiert. Um die Anwendbarkeit der Methodik zu verbessern, muss die *Berechnung der Änderungswahrscheinlichkeit* angepasst werden, sodass Informationen zu Änderungsinitiatoren nicht mehr benötigt werden.

5 Forschungsbedarf und Anforderungen an die Methodik

Eine hohe Anzahl an Entwicklungsprojekten scheitert an einer unzureichenden Anforderungsentwicklung. Ursachen dafür sind Qualitätsdefizite von Anforderungen sowie Anforderungsänderungen (siehe Abschnitt 1.1). Aufgrund der Komplexität von technischen Systemen ist es eine Herausforderung, ein vollständiges und eindeutiges Anforderungsset zu ermitteln. Zur Ermittlung eines vollständigen Anforderungssets muss eine hohe Anzahl an Informationen aus verschiedenen Dokumenten (insbesondere Lastenheften) berücksichtigt werden. Um Auswirkungen von Änderungspropagationen bewerten zu können, müssen Anforderungsabhängigkeiten bekannt sein (siehe Abschnitt 2.1). Es wird eine Methodik benötigt, um Anforderungen effizient aus Lastenheften zu extrahieren. Dadurch wird Bearbeitungszeit in der Entwicklung reduziert, um eine vollständige Spezifikation zu generieren. Diese Anforderungen müssen formalisiert werden, um eine eindeutige Anforderungsbeschreibung und die Interpretation durch Entwicklungswerzeuge zu ermöglichen. Anforderungsabhängigkeiten müssen identifiziert werden, um als Grundlage für das Risikomanagement zu fungieren und besser Einflüsse aufgrund der VUCA-Dimensionen zu handhaben. Außerdem muss die Berechnung des Änderungsrisikos von Anforderungen mittels Informationen, die in der Praxis verfügbar sind, durchgeführt werden.

5.1 Forschungsbedarf und Forschungsfragen

In der Literatur existieren verschiedene Ansätze zur automatisierten *Extrahierung von Anforderungen* aus Lastenheften. Aufgrund verschiedener Aspekte (Dateiformat, Formatisierung im Text, Verweise, Qualitätsdefizite, Anforderungsbeschreibung, Anforderungsarten und Sprache – siehe Tabelle 3-3, S. 16) wird eine automatisierte Extrahierung erschwert und die industrielle Anwendbarkeit ist nur bedingt gegeben. Lastenhefte und Anforderungsbeschreibungen sind nicht standardisiert, weshalb eine intelligente Verarbeitung der Lastenhefte durch KI-Ansätze unter Berücksichtigung der aufgeführten Aspekte benötigt wird (siehe Abschnitt 3.1). Außerdem ist für die industrielle Anwendbarkeit der KI-basierten Extrahierung von Anforderungen die Verfügbarkeit von Trainingsdaten für Klassifikationsmodelle problematisch. Zum Trainieren von Klassifikationsmodellen wird eine ausreichend große Anzahl an domänen spezifischen Daten benötigt. Die Verwendung domänen spezifischer Daten ist aufgrund des verwendeten Vokabulars notwendig. Andernfalls können die Daten von den Klassifikationsmodellen nicht verarbeitet werden, was zu einer geringen Leistungsfähigkeit der Modelle führt. Diese Anzahl an domänen spezifischen Daten ist in der Praxis oft nicht verfügbar und Daten müssen generiert werden (siehe Abschnitt 3.1). Dies geschieht z. B. durch die manuelle Kennzeichnung von Texten in Spezifikationen – es wird ermittelt, ob das jeweilige Textsegment eine Anforderung ist. Methoden zur Daten-Augmentation haben großes Potenzial, den Aufwand für die manuelle Kennzeichnung von Anforderungen zu

reduzieren und domänenspezifische Daten zu generieren. Mit Hilfe der Daten-Augmentation wird nur eine geringe Anzahl von realen Daten benötigt, um einen signifikanten Datensatz für das Training von Modellen zu generieren (siehe Abschnitt 3.1).

Forschungsbedarf: Es besteht der Bedarf nach einem automatisierten Ansatz zur Extrahierung von Anforderungen, der industriell anwendbar ist. Die industrielle Anwendbarkeit bezieht sich auf die Verfügbarkeit der benötigten Daten, um eine hohe Leistungsfähigkeit der KI-Modelle in der praktischen Anwendung zu erreichen. Auch bei einer geringen Anzahl an real-verfügbaren und gekennzeichneten Daten, muss die Leistungsfähigkeit ausreichend hoch sein.

Aktuell existiert noch kein Ansatz für die *Formalisierung von Anforderungen* für komplexe technische Systeme, um Anforderungen in eine eindeutige, maschinenlesbare Form zu bringen. Sie sollen direkt von Entwicklungswerkzeugen, wie für die generative Entwicklung von Bauteilen (u. a. Synera [Syn23]) verarbeitet werden können. Für die Formalisierung von Anforderungen ist besonders die Relationsextraktion sowie das Ausfüllen von Vorlagen relevant. Aufgrund des hohen Grads der Formalisierung, der zur automatisierten Verarbeitung der Anforderungen erforderlich ist, eignen sich Ansätze zum Ausfüllen von Vorlagen. Es müssen Schlüssel-Wert-Paare in Anforderungen erkannt werden. Diese Schlüssel-Wert-Paare müssen Parameter definieren, welche für die generative Entwicklung von Bauteilen benötigt werden. Hierzu muss u. a. eine NER durchgeführt werden (siehe Abschnitt 3.2). Es wurde bisher nicht evaluiert, welche Ansätze sich für diesen Anwendungskontext eignen und wie hoch die Leistungsfähigkeit dieser Ansätze ist.

Forschungsbedarf: Es besteht der Bedarf nach einem automatisierten Ansatz zur Formalisierung von Anforderungen in der Entwicklung komplexer technischer Systeme. Anforderungen müssen maschinenlesbar sein, sodass diese durch Entwicklungswerkzeuge zur generativen Entwicklung interpretiert werden können.

Die Entwicklung eines Ansatzes zur *Abhängigkeitsanalyse von Anforderungen* ist ein multikriterielles Optimierungsproblem. Kriterien sind der Anwendungsaufwand, die Vollständigkeit der Ergebnisse und die Differenzierung der Abhängigkeiten nach Ausbreitungsverhalten. Da weder singuläre noch ontologiebasierte Ansätze in diesem Kontext geeignet sind, werden nur maschinelle Lernansätze betrachtet. Die Vorteile dieser Ansätze sind der hohe Automatisierungsgrad und die Robustheit. Änderungen höherer Ordnung und große Anforderungsmengen können effizient analysiert werden. Nachteilig ist die hohe Menge an Trainingsdaten, die benötigt wird, um eine hohe Leistungsfähigkeit in der Abhängigkeitsanalyse zu erreichen (siehe Abschnitt 3.3). In einer Feldstudie wurden drei Anforderungen an eine Abhängigkeitsanalyse von Anforderungen komplexer technischer Systeme ermittelt: „Verarbeitbarkeit einer hohen Anzahl an Anforderungen sowie von Anforderungen aus unterschiedlichen Disziplinen“, „Berücksichtigung von Propagationseffekten zwischen Anforderungen“ und „Verfügbarkeit der erforderlichen Daten und Informationen in frühen Entwicklungsphasen“

(siehe Abschnitt 4.1). Die Feldstudie zeigt, dass BERT im Vergleich zu anderen Modellen die höchste Ergebnisgüte besitzt. Für die Propagationsanalyse ist die Betrachtung von zwei Klassen ausreichend. Es werden die Anforderungen als abhängig definiert, die eine zwangsläufige Propagation aufweisen. Die Leistungsfähigkeit für die Klassifizierung heterogener Daten (bzw. bei abweichenden Trainings- und Testsets) ist niedrig. Vortrainierte Modelle können nur begrenzt für die Klassifizierung unbekannter Daten (z. B. aus unterschiedlichen Projekten) angewendet werden. Für die industrielle Anwendbarkeit ist es notwendig, dass Daten aufwandsarm klassifiziert werden können (siehe Abschnitt 4.1). In der Literatur wird „aktives Lernen“ als Lernverfahren eingesetzt, um den Aufwand zum Kennzeichnen unbekannter Daten zu reduzieren und aufwandsarm die Leistungsfähigkeit von Modellen zu verbessern. Die Anwendbarkeit von „aktivem Lernen“ in Kombination mit BERT muss im Kontext der Abhängigkeitsanalyse von Anforderungen untersucht werden.

Forschungsbedarf: Es besteht der Bedarf nach einer automatisierten Abhängigkeitsanalyse, welche industriell anwendbar ist. Für eine industrielle Anwendbarkeit ist es notwendig, dass unbekannte Daten (heterogenes Trainings- und Testset; z. B. Daten aus neuen Projekten) mit einer hohen Leistungsfähigkeit korrekt klassifiziert werden und der Aufwand für das Training neuer Modelle gering ist.

Für das *Risikomanagement von Anforderungsänderungen* wird eine Methodik benötigt, um systematisch Änderungsentscheidungen zu treffen und geeignete Steuerungsmaßnahmen auszuwählen. Die ProMaRC Methodik wurde dazu in Vorarbeiten des Lehrstuhls für Produktentstehung entwickelt (siehe Anhang A2.4). In einer Vorstudie wurde die Methodik evaluiert. Hierbei wurde festgestellt, dass für die Berechnung des Änderungsrisikos nicht alle Informationen verfügbar sind, damit die Methodik in der Praxis anwendbar ist (siehe Abschnitt 4.2). Dies betrifft vor allem die Anzahl an Änderungen durch Änderungsinitiatoren im laufenden Projekt.

Forschungsbedarf: Es besteht der Bedarf nach einer Risikoberechnung von Anforderungsänderungen, welche industriell anwendbar ist. Die industrielle Anwendbarkeit bezieht sich auf die Verfügbarkeit der benötigten Daten – speziell auf die Verfügbarkeit von Informationen zu Änderungsinitiatoren von Anforderungen.

Neuheitsgrad der Arbeit

Die Ansätze, die im Stand der Forschung beschrieben werden, werden hinsichtlich verschiedener Kriterien (K1-5) bewertet und von dieser Arbeit abgegrenzt, um den Neuheitsgrad darzustellen (siehe Bild 5-1). Zur automatisierten Extrahierung von natürlich-sprachlichen Anforderungen aus dem Lastenheft müssen diese klassifiziert werden (K1: *Automatisierte Klassifizierung von Anforderungen*). Für eine industrielle Anwendbarkeit ist es notwendig, dass ausreichend Daten zum Trainieren von Modellen für die Extrahierung vorliegen, bzw. effizient generiert werden. Hierzu können beispielsweise Ansätze zur Daten-Augmentation genutzt werden (K2: *Effiziente Generierung*

einer Datengrundlage zur Extrahierung von Anforderungen). Zur Formalisierung der Anforderungen müssen Informationen, die für die generative Entwicklung relevant sind, aus einem Text extrahiert und strukturiert werden, sodass sie von Entwicklungswerkzeugen direkt interpretiert werden können (K3: Formalisierung entwicklungsrelevanter Informationen). Anforderungsabhängigkeiten müssen automatisiert erkannt werden. Für eine industrielle Anwendbarkeit ist es ebenfalls notwendig, dass unbekannte Daten (z. B. aus neuen Projekten) mit einer hohen Leistungsfähigkeit erkannt werden (K4: Hohe Leistungsfähigkeit von Modellen der Abhängigkeitsanalyse für neue Projekte). Für das Risikomanagement von Anforderungsänderungen muss das Änderungsrisiko von Anforderungen berechnet werden. Für eine industrielle Anwendbarkeit müssen die benötigten Daten verfügbar sein (K5: Risikoberechnung von Anforderungsänderungen).

Abgrenzungskriterien	Forschungsansätze	Extrahierung und Klassifizierung von Anforderungen	Formalisierung von Anforderungen	Abhangigkeitsanalyse von Anforderungen	Risikomanagement von Anforderungsänderungen	DFG: COFFEE Prof. Watzack	DFG: WIRPro Prof. Abramovici	DFG: GaIP Prof. Lachmayer	Vorliegendes Dissertationsmanuskript
K1: Automatisierte Klassifizierung von Anforderungen	●	-	-	-	●	-	-	●	●
K2: Effiziente Generierung einer Datengrundlage zur Extrahierung von Anforderungen	●	-	-	-	-	-	-	●	●
K3: Formalisierung entwicklungsrelevanter Informationen	-	●	-	-	-	●	●	●	●
K4: Hohe Leistungsfähigkeit von Modellen der Abhängigkeitsanalyse für neue Projekte	-	-	●	-	-	-	-	●	●
K5: Risikoberechnung von Anforderungsänderungen	-	-	-	●	-	-	-	●	●

Legende: Lösung ● Teillösung ○ Keine Lösung -

Bild 5-1: Abgrenzungsmatrix zum Stand der Forschung

Die folgenden Ansätze sind relevant für die einzelnen Gruppen:

- *Extrahierung und Klassifizierung von Anforderungen:* [AIV20, AMB15, BH11, BKB21, BKR21, GDB18, GLT05, GPT22, HDL22, HKR20, HKR20, KM17, KO14, MMM15, Nau21, OZ18, Rup16, RWC19, SBC19]
- *Formalisierung von Anforderungen:* [GEL14, GPM21b, JJM21, KGG21, MP12, Rup16, SG18, SM15, STM03, WJB21]

- *Abhängigkeitsanalyse von Anforderungen:* [ANS08, ASF18, DAR19, DRO13, HH11, HVM18, KCW08, LGN07, MCG19, Mis16, NFS14, NRC02, Ole22, PKK00, SSA19, ZJ05]
- *Risikomanagement von Anforderungsänderungen:* [CSE04, Die13, DIN18, Ger11, GOP22, GOS20, MBP19, MSS12, Neu17, Ole22, SPL17, Van12]

In der Literatur existieren Ansätze, die eine automatisierte Klassifizierung von Anforderungen ermöglichen (*K1*). Es werden Modelle des maschinellen Lernens zum Klassifizieren von Anforderungen angewendet. Auf den Ansätzen des Stands der Forschung wird in diesem Bereich deshalb aufgebaut. **Bewertung:** *Teillösung. Es werden bestehende Ansätze des maschinellen Lernens angepasst und eingebunden.*

Aktuell existieren keine Ansätze, in denen Daten-Augmentation genutzt wird, um künstliche Anforderungen zu generieren (*K2*). Es muss untersucht werden, wie hoch die Leistungsfähigkeit von Modellen ist, welche mit künstlichen Anforderungen trainiert worden sind. **Bewertung:** *Lösung. Bestehende Ansätze greifen in diesem Anwendungskontext nicht auf Daten-Augmentation zurück.*

Es existieren bereits Ansätze zur Formalisierung von natürlicher Sprache (*K3*). Im Rahmen dieser Arbeit werden die Ansätze zur Formalisierung hinsichtlich der zu extrahierenden Informationen angepasst und weiterentwickelt. Die Methode wird für die generative Entwicklung von Bauteilen ausgelegt. In diesem Anwendungskontext existieren keine Ansätze zur automatisierten Formalisierung der benötigten Informationen. **Bewertung:** *Lösung. Im Anwendungskontext von Anforderungen komplexer technischer Systeme und speziell der generativen Entwicklung von Bauteilen existieren keine Ansätze zur automatisierten Formalisierung von Anforderungen.*

In der Literatur und in Vorarbeiten bestehen bereits Ansätze zur automatisierten Abhängigkeitsanalyse von Anforderungen (*K4*). Ein Problem dieser Ansätze ist aktuell, dass die Leistungsfähigkeit von Modellen des maschinellen Lernens bei der Klassifizierung von unbekannten Daten aufgrund der Heterogenität der Daten zum Testset deutlich abnimmt. BERT-Modelle werden in diesem Anwendungskontext nicht durch aktives Lernen trainiert. **Bewertung:** *Lösung. Bestehende Ansätze, bzw. Modelle haben eine niedrige Leistungsfähigkeit in der Klassifizierung von unbekannten Anforderungen aus neuen Projekten.*

Für das Risikomanagement von Anforderungsänderungen bestehen in der Literatur Ansätze, um das Änderungsrisiko von Anforderungen zu berechnen (*K5*). Eine Methode zur Berechnung des Änderungsrisikos wurde in Vorarbeiten (als Teil der ProMaRC-Methodik [Ole22]) entwickelt [GOP22]. Es wurde festgestellt, dass teilweise Informationen zu Änderungsinitiatoren nicht verfügbar sind, um die Methode anzuwenden. **Bewertung:** *Teillösung. Es wird auf der Methode von ProMaRC zur Risikoberechnung von Anforderungsänderungen aufgebaut. Diese Methode muss angepasst werden, damit industriell-verfügbare Daten zur Anwendung ausreichen.*

Forschungsfragen

Aus dem Forschungsbedarf ergeben sich die für die vier Bereiche Extrahierung von Anforderungen, Formalisierung von Anforderungen, Abhängigkeitsanalyse von Anforderungen sowie Risikomanagement von Anforderungsänderungen jeweils Forschungsfragen (FF), die im Rahmen dieser Arbeit beantwortet werden.

Für die *Extrahierung von Anforderungen* muss untersucht werden, inwiefern Daten-Augmentation für die Generierung von künstlichen Anforderungen angewendet werden kann. Hierzu muss die Leistungsfähigkeit der Daten-Augmentation bewertet werden. Außerdem muss bewertet werden, wie hoch die Leistungsfähigkeit von Modellen ist, die mit künstlichen Anforderungsdaten trainiert worden sind:

- „Wie hoch ist die Leistungsfähigkeit der Daten-Augmentation von Anforderungen durch künstliche Intelligenz?“ (FF 1)
- „Wie hoch ist die Leistungsfähigkeit von Klassifizierungsmodellen, die augmentierte Daten verwenden, im Vergleich zu Modellen, die auf realen Daten basieren?“ (FF 2)

Zur *Formalisierung von Anforderungen* muss untersucht werden, welche Ansätze für eine automatisierte Formalisierung der benötigten Informationen aus Anforderungen geeignet sind. Speziell wird die generative Entwicklung als Anwendungskontext untersucht, weil die Anwendung einen hohen Mehrwert zu erwarten hat:

- „Welche Ansätze sind für die Formalisierung von Anforderungen im Anwendungskontext der generativen Entwicklung anwendbar?“ (FF 3)
- „Wie hoch ist die Leistungsfähigkeit von Ansätzen zur Formalisierung von Anforderungen im Kontext der generativen Entwicklung?“ (FF 4)

Für die *Abhängigkeitsanalyse von Anforderungen* muss untersucht werden, wie hoch die Leistungsfähigkeit von BERT-Modellen ist, die mit aktivem Lernen trainiert worden sind:

- „Wie hoch ist die Leistungsfähigkeit von BERT-Modellen in der Abhängigkeitsanalyse von Anforderungen, welche mit aktivem Lernen trainiert worden sind?“ (FF 5)

Für das *Risikomanagement von Anforderungsänderungen* muss eine Berechnung des Änderungsrisikos von Anforderungen definiert werden, für die alle notwendigen Informationen zur Anwendung der Methode verfügbar sind:

- „Wie muss eine Berechnung des Änderungsrisikos von Anforderungen definiert sein, damit die benötigten Informationen zur Anwendung der Berechnung in der industriellen Praxis verfügbar sind?“ (FF 6)

5.2 Erfolgsfaktoren der Methodik

Zur Operationalisierung des Forschungsbedarfs wurden Anforderungen an die zu entwickelnde Methodik definiert. Es wurden Use Cases für die Rollen des Anforderungsingenieurs und des Entwicklungingenieurs erstellt und in einem Workshop mit Industrievertretern diskutiert, um die Anwendbarkeit in der Praxis zu gewährleisten (siehe Anhang A3.2: B-W3). Insgesamt wurden 66 Anforderungen ermittelt, die in Anlehnung an HAMRAZ in die Kategorien *Input*, *Funktionalität*, *Anwendbarkeit* und *Output* [HCW13] eingeteilt wurden. Diese Anforderungen wurden dann nach BALZERT in Bezug auf ihre Notwendigkeit priorisiert [Bal10]. Dabei wurde zwischen *wesentlichen*, *bedingt notwendigen* und *optionalen* Anforderungen unterschieden. Im Folgenden werden die 13 Anforderungen mit der Priorität *wesentlich* beschrieben. Im Kontext dieser Arbeit werden Anforderungen an die Methodik *Erfolgsfaktoren* genannt. Dieser Begriff wird gewählt, um die Lesbarkeit zu verbessern, da die Forschungsgegenstände dieser Arbeit ebenfalls Anforderungen sind.

Input

Erfolgsfaktor 1: Das System muss dem Anwender die Möglichkeit bieten, ein branchenübliches Lastenheft einlesen zu können.

Branchenübliche Lastenhefte müssen von der Methodik verarbeitet werden können. Anforderungen werden in Lastenheften standardmäßig als natürlich-sprachlicher Text dokumentiert. Diese können sowohl als Fließtext oder als Tabellen formatiert werden. Außerdem müssen verschiedene Aspekte, die eine KI-basierte Extrahierung erschweren, berücksichtigt werden (siehe Tabelle 3-3). Der Erfolgsfaktor ist erfüllt, wenn:

- PDF-Dokumente verarbeitet werden können (Dateiformat),
- Fließtext, Listen und Tabellen verarbeitet werden können (Formatierung im Text),
- Co-Referenzen aufgelöst werden können (Verweise),
- Anforderungen mit niedriger Qualität verarbeitet werden können (Qualitätsdefizite),
- unstrukturierte Anforderungsbeschreibungen verarbeitet werden können (Anforderungsbeschreibung) und
- funktionale und nicht-funktionale Anforderungen mit einer hohen Leistungsfähigkeit erkannt werden können (Anforderungsarten).

Erfolgsfaktor 2: Das System muss dem Anwender die Möglichkeit bieten, ein englischsprachiges Lastenheft einlesen zu können.

Englischsprachige Anforderungen müssen verarbeitet werden können. Dies führt dazu, dass die Anwendbarkeit der Methodik auch international ermöglicht wird. Außerdem werden Lastenhefte deutscher Unternehmen häufig in englischer Sprache geschrieben.

Der Erfolgsfaktor ist erfüllt, wenn die Leistungsfähigkeit der Methoden hoch ist, wenn englischsprachige Anforderungen eingelesen werden.

Funktionalität

Erfolgsfaktor 3: Das System muss Anforderungsdaten aufwandsarm generieren können.

Für das Training von Modellen zur automatisierten Klassifizierung von Anforderungen muss eine ausreichend große Anzahl an Anforderungsdaten verfügbar sein. Es soll eine aufwandsarme Generierung von Anforderungsdaten möglich sein, damit die Klassifizierung, bzw. Extrahierung von Anforderungen durch KI-Modelle aufwandsarm möglich ist und die Anwendbarkeit in der industriellen Praxis erhöht wird. Der Erfolgsfaktor ist erfüllt, wenn Anforderungsdaten mit einer hohen Leistungsfähigkeit generiert werden und der Aufwand zur Generierung der Anforderungsdaten niedrig ist.

Erfolgsfaktor 4: Das System muss Anforderungen nach ihrer Art klassifizieren können.

Aufbauend auf den extrahierten Textsegmenten muss bestimmt werden, ob es sich bei dem Textsegment um eine Anforderung handelt. Der Erfolgsfaktor ist erfüllt, wenn Textsegmente mit einer hohen Ergebnisgüte hinsichtlich funktionaler, nicht-funktionaler und keiner Anforderung klassifiziert werden können. Außerdem muss die Ergebnisgüte hoch sein, wenn künstliche Anforderungsdaten für das Training des KI-Modells genutzt werden sind. Dadurch wird die Anwendbarkeit der Methode aufgrund des niedrigen Aufwands zur Generierung von Trainingsdaten erhöht.

Erfolgsfaktor 5: Das System muss Anforderungen formalisieren können.

Die extrahierten Anforderungen müssen in eine formale Darstellung überführt werden, um maschinenlesbar zu werden. Der Erfolgsfaktor ist erfüllt, wenn Anforderungen mit einer hohen Ergebnisgüte formalisiert werden können. Speziell wird der Anwendungsfall für die generative Entwicklung von Bauteilen untersucht. Es muss geprüft werden, ob Parameter, die für die generative Entwicklung relevant sind und in Lastenheften dokumentiert werden, formalisiert werden können.

Erfolgsfaktor 6: Das System muss Abhängigkeiten von Anforderungen erkennen können.

Wissen über Anforderungsabhängigkeiten ist beispielsweise für das Risikomanagement von Anforderungsänderungen entscheidend. Der Erfolgsfaktor ist erfüllt, wenn die Abhängigkeitsarten, die eine Propagation von Anforderungsänderungen verursachen (siehe Abschnitt 4.1), mit einer hohen Leistungsfähigkeit erkannt werden.

Anwendbarkeit

Erfolgsfaktor 7: Das System muss so gestaltet sein, dass nur branchenüblich verfügbare Informationen zur Anwendung benötigt werden.

Um eine hohe Anwendbarkeit in der Praxis zu gewährleisten, sollen für die Anwendung der Methodik nur Informationen benötigt werden, die in der Produktentwicklung für komplexe technische Systeme üblicherweise vorliegen. Speziell muss für die Risikoberechnung von Anforderungen berücksichtigt werden, dass Informationen zu Änderungsinitiatoren von Anforderungen in der Praxis häufig nicht verfügbar sind. Der Erfolgsfaktor ist erfüllt, wenn die Anwendungspartner die benötigten Informationen als „branchenüblich verfügbar“ bewerten. Außerdem muss die Leistungsfähigkeit der Methoden hoch sein, wenn branchenübliche Lastenhefte zur Anwendung der Methoden verwendet werden.

Erfolgsfaktor 8: Das System muss so gestaltet sein, dass die benötigten Informationen in der Projektinitiierungsphase verfügbar sind.

Die benötigten Informationen zur Anwendung der Methodik müssen früh im Projekt verfügbar sein. Die Methoden unterstützen primär in frühen Phasen, weshalb eine frühe Anwendbarkeit notwendig ist. Beispielsweise sollen frühzeitig Anforderungsabhängigkeiten bekannt sein, um diese für das Risikomanagement von Anforderungsänderungen berücksichtigen zu können. Der Erfolgsfaktor ist erfüllt, wenn die Anwendungspartner die benötigten Informationen als „früh verfügbar“ bewerten.

Erfolgsfaktor 9: Das System muss so gestaltet sein, dass eine generische Anwendbarkeit in der Entwicklung komplexer technischer Systeme gegeben ist.

Für eine hohe Anwendbarkeit der Methodik soll diese für unterschiedliche, komplexe technische Systeme eingesetzt werden können. Der Erfolgsfaktor ist erfüllt, wenn Anwendungspartner die Methodik als „generisch anwendbar“ bewerten. Außerdem müssen in der Validierung der Methoden verschiedene Produkte untersucht werden und die Leistungsfähigkeit hoch sein.

Erfolgsfaktor 10: Das System muss so gestaltet sein, dass der Vorbereitungsaufwand akzeptabel ist.

Die Methodik soll effizient für eine Vielzahl an Anforderungen anwendbar sein. Damit die Methodik angewendet werden kann, müssen verschiedene Tätigkeiten durchgeführt werden. Beispielsweise müssen KI-Modelle trainiert werden. Der Erfolgsfaktor ist erfüllt, wenn die Industriepartner den Vorbereitungsaufwand als „akzeptabel“ bewerten.

Erfolgsfaktor 11: Das System muss so gestaltet sein, dass der Anwendungsaufwand akzeptabel ist.

Der Anwendungsaufwand der Methodik ist abhängig vom Umfang des Projekts und des Anforderungssets, weshalb kein absoluter Wert für den maximalen Anwendungsaufwand definiert wird. Der Erfolgsfaktor ist erfüllt, wenn die Industriepartner den Anwendungsaufwand als „akzeptabel“ bewerten. Dabei muss die weitere Zeitersparnis im Projekt durch die Anwendung der Methodik berücksichtigt werden.

Erfolgsfaktor 12: Die Methodik muss so gestaltet sein, dass sie in das Vorgehen einer Methodik zur Entwicklung komplexer technischer Systeme in der industriellen Praxis integriert werden kann.

Damit die Methodik im Unternehmen Anwendung finden kann, muss diese in den Entwicklungsprozess integriert werden. Der Erfolgsfaktor ist erfüllt, wenn die Integration der Methoden in eine Entwicklungsmethodik für komplexe technische Systeme vom Anwendungspartner als „geeignet“ bewertet wird. Außerdem muss die Methodik als „einfach anwendbar bewertet“ werden. Durch die einfache Anwendung wird die Akzeptanz der Methodik in der industriellen Praxis erhöht.

Output

Erfolgsfaktor 13: Die Ergebnisse des Systems müssen in einem gängigen Austauschformat ausgegeben werden, sodass diese in branchenübliche Anforderungsmanagement-Software eingelesen werden können.

Die Anwendbarkeit der Methodik wird erhöht, wenn die Ergebnisse in einem gängigen Austauschformat ausgegeben werden. Dadurch können die extrahierten und formalisierten Anforderungen sowie deren Abhängigkeiten in ein branchenübliches Anforderungsmanagement-Werkzeug eingelesen werden. Der Erfolgsfaktor ist erfüllt, wenn die Ergebnisse (extrahierte und formalisierte Anforderungen sowie deren Abhängigkeiten) in ein branchenübliches Anforderungsmanagement-Werkzeug eingelesen werden können. Exemplarisch wird in dieser Arbeit IBM DOORS als branchenübliches Anforderungsmanagement-Werkzeug betrachtet. Die 13 Erfolgsfaktoren der Methodik werden in Tabelle 5-1 zusammenfassend dargestellt.

Tabelle 5-1: Erfolgsfaktoren der Methodik

ID	Kategorie	Erfolgsfaktor
1	Input	Verarbeitung branchenüblicher Lastenhefte
2	Input	Verarbeitung englischsprachiger Anforderungen
3	Funktionalität	Generierung von Anforderungsdaten
4	Funktionalität	Klassifizierung von Anforderungen
5	Funktionalität	Formalisierung von Anforderungen
6	Funktionalität	Abhängigkeitsanalyse von Anforderungen
7	Anwendbarkeit	Verfügbarkeit benötigter Informationen
8	Anwendbarkeit	Frühzeitige Anwendbarkeit
9	Anwendbarkeit	Generische Anwendbarkeit
10	Anwendbarkeit	Akzeptabler Vorbereitungsaufwand
11	Anwendbarkeit	Akzeptabler Anwendungsaufwand
12	Anwendbarkeit	Integrierbarkeit in den Entwicklungsprozess
13	Output	Gängiges Austauschformat

6 Entwicklung der Methodik

Aufbauend auf dem Stand der Forschung, der Vorstudie und den Erfolgsfaktoren wird eine Methodik zur automatisierten Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen entwickelt. Die Methodik besteht aus einem Software-Werkzeug, einem Modell der Software-Architektur, einem Referenzprozess der Methodik-Anwendung und vier Methoden:

- 1) Extrahierung und Klassifizierung natürlich-sprachlicher Anforderungen
- 2) Formalisierung von Anforderungen
- 3) Erkennung von Anforderungsabhängigkeiten
- 4) Berechnung der Änderungsrisiken

Es sollen Textsegmente aus Lastenheften extrahiert und hinsichtlich Anforderungsart klassifiziert werden. Die Anforderungen sollen formalisiert werden. Es sollen Anforderungsabhängigkeiten erkannt werden. Auf Basis der Anforderungsabhängigkeiten und Experteneinschätzungen sollen Änderungsrisiken der Anforderungen berechnet werden. Eine Übersicht der Methodik wird in Bild 6-1 dargestellt.



Bild 6-1: Methodik zur Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen

6.1 Modelle

Teil der Methodik ist ein Modell der Software-Architektur sowie ein Modell des Referenzprozesses zur Methodik-Anwendung. Die Software-Architektur ist eine funktionale Sicht auf die zu entwickelnde Software. Ziel der Modellierung ist es, das Software-Werkzeug systematisch zu entwickeln, damit die Erfolgsfaktoren der Methodik erfüllt werden. Grundlagen zur Systemmodellierung werden in Anhang A2.7 beschrieben. Das Systemmodell wurde im Rahmen der Betreuung einer Bachelorarbeit entwickelt [Zha22]. Der Referenzprozess dient als Orientierung für die unternehmensspezifische Ausgestaltung der Anwendung der entwickelten Methodik.

6.1.1 Software-Architektur

Für die Modellierung der Software-Architektur muss eine geeignete Modellierungssprache, ein Modellierungswerkzeug sowie eine Modellierungssprache ausgewählt werden. Die Modellierungssprache muss Diagramme unterstützen, um eine Software in Struktur und Verhalten abzubilden. Für die Struktur des Software-Werkzeugs (z. B. in Form von Softwaremodulen) sind Klassendiagramme und Komponentendiagramme relevant. Das (gewünschte) Verhalten der Software wird durch Anforderungs-, Use-Case- und Aktivitätsdiagramme abgebildet. Die Modellierungssprache UML ermöglicht die Erstellung dieser Diagramme, weshalb sie ausgewählt wird. Zur Erstellung des Anforderungsdiagramms wird SysML als Erweiterung der UML verwendet. Das Modellierungswerkzeug muss die Sprache UML/SysML unterstützen und die Erstellung der benötigten Diagramme (s.o.) ermöglichen. Das Modellierungswerkzeug MagicDraw von NoMagic erfüllt diese Anforderungen, weshalb es zur Modellierung ausgewählt worden ist.

Als Modellierungsmethode wird eine Vorgehensweise in Anlehnung an SYSMOD von Weilkiens ausgewählt [Wei14]. Diese Methode gibt Schritte vor, wie unter Berücksichtigung der Sprache UML/SysML die Software-Architektur modelliert werden kann. Es werden konkret die folgenden Schritte durchgeführt: Anforderungen ermitteln, Systemkontext modellieren, Anwendungsfälle modellieren und Anwendungsfälle realisieren. Um die Software-Architektur modellieren zu können, ist ein umfangreiches Verständnis der Problemstellung, die durch die Nutzung des Software-Werkzeugs in der Praxis adressiert werden soll, notwendig. Aus diesem Grund wurde ein Workshop zur Absicherung der Anwendbarkeit in der Praxis durchgeführt (siehe Anhang A3.2.: B-W9).

Anforderungen ermitteln

Die Erfolgsfaktoren werden zugrunde gelegt, um Anforderungen an die Software-Architektur zu ermitteln. Der Fokus wurde auf die Ermittlung funktionaler Anforderungen gelegt. Es wurden für die vier Methoden „Extrahierung und Klassifizierung natürlichsprachlicher Anforderungen“, „Formalisierung von Anforderungen“, „Erkennung von Anforderungsabhängigkeiten“ und „Berechnung der

Änderungsrisiken“ funktionale Anforderungen abgeleitet. Beispielsweise wurde für die „Extrahierung und Klassifizierung natürlichsprachlicher Anforderungen“ die funktionale Anforderung „Das Werkzeug muss Wörter in Sätzen segmentieren können.“ definiert. Außerdem wurden übergeordnete Funktionalitäten des Software-Werkzeugs betrachtet und dazu Anforderungen abgeleitet. Hierzu zählen beispielsweise die Verwaltung von Projekten, die im Software-Werkzeug angelegt werden können oder der Import von Lastenheften. Als Anforderung an den Import von Lastenheften wurde beispielsweise ermittelt: „Das Dateiformat des Lastenhefts muss PDF sein.“. Die Anforderungsdiagramme sind in Anhang A4.1 dargestellt. Exemplarisch wird das Anforderungsdiagramm der Methode „Extrahierung und Klassifizierung natürlichsprachlicher Anforderungen“ in Bild 6-2 dargestellt. Es wurden die folgenden Attribute definiert: „ID“, „Art der Anforderung“, „Extrahierte Schlüssel-Wert-Paare“, „Tags“. Insgesamt wurden 66 Anforderungen ermittelt.

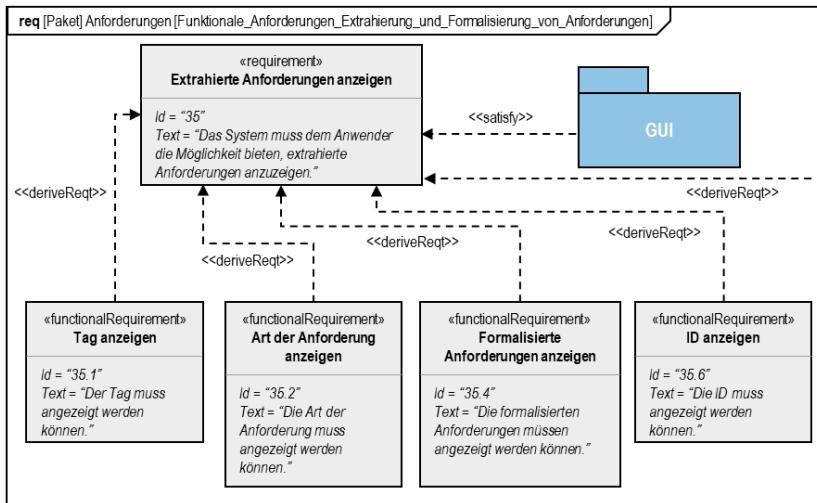


Bild 6-2: Ausschnitt aus dem Anforderungsdiagramm der "Extrahierung und Klassifizierung natürlichsprachlicher Anforderungen"

Systemkontext modellieren

In der Modellierung des Systemkontexts werden Akteure (Menschen oder Software-Systeme) identifiziert, die mit dem Software-Werkzeug interagieren. Es wird zwischen Benutzer, Fremdsystem, Benutzersystem, Schnittstellensystem, Umwelteinfluss, Aktuator und Sensor unterschieden [Wei14]. Es wurden die folgenden menschlichen Akteure identifiziert: Anforderungsingenieur, Domänenexperte und Projekt-Manager. Der Anforderungsingenieur bedient das Software-Werkzeug zur Extrahierung,

Formalisierung und Handhabung von Änderungen technischer Anforderungen. Der Domänenexperte bewertet Anforderungsabhängigkeiten. Der Projekt-Manager interagiert mit der Anforderungsmanagement-Software, in welcher die Ergebnisse der Anwendung des Software-Werkzeugs zur Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen (Extrahierte Anforderungen, Formalisierte Anforderungen, Anforderungsabhängigkeiten, Änderungsrisiko, ...) importiert worden sind. Beispielsweise legt er auf Basis der Änderungsrisiken der Anforderungen Risikosteuerungsmaßnahmen in der Anforderungsmanagement-Software fest. Identifizierte Software-Systeme sind: Software-Werkzeug zur Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen (Benutzer-System), eine Wissensbasis (Schnittstellensystem), Anforderungsmanagement-Software (Fremd-System) und Synera (Fremd-System). In der Wissensbasis werden Ergebnisse gespeichert. Beispielsweise werden Anforderungen, die durch das zu entwickelnde Software-Werkzeug formalisiert werden, in der Wissensbasis gespeichert. Die formalisierten Anforderungen werden aus der Wissensbasis in Synera importiert und sind Randbedingungen für die generative Entwicklung von Bauteilen. In das Software-Werkzeug der Methodik soll ein Export von „Comma-separated values“ (CSV-)Dateien implementiert werden. In die CSV-Dateien werden die Ergebnisse der Anwendung der Methodik (extrahierte Anforderungen etc.) geschrieben und sie können in branchenübliche Anforderungsmanagement-Software (z. B. IBM DOORS) importiert werden.

Anwendungsfälle modellieren

Auf Basis der funktionalen Anforderungen und des Systemkontexts (bzw. der beteiligten Akteure) werden Use-Cases definiert. Es wird ein Paket-Diagramm erstellt, um das Zusammenspiel der Use-Cases, Akteure und Software-Komponenten abzubilden (siehe Bild 6-3). Der Anforderungsingenieur führt Use-Cases des Software-Werkzeugs zur Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen aus. Dies sind die folgenden: „Projekt erstellen“, „Anforderungen in Wissensbasis exportieren“, „Extrahierte Anforderungen anzeigen“, „Anforderungsabhängigkeiten erkennen“, „Schlüssel-Wert-Paare detailliert anzeigen“, „Risiko von Anforderungsänderungen analysieren“ und „CSV-Export“. Für diese Use-Cases wurden Use-Case-Diagramme erstellt (siehe Anhang A4.2).

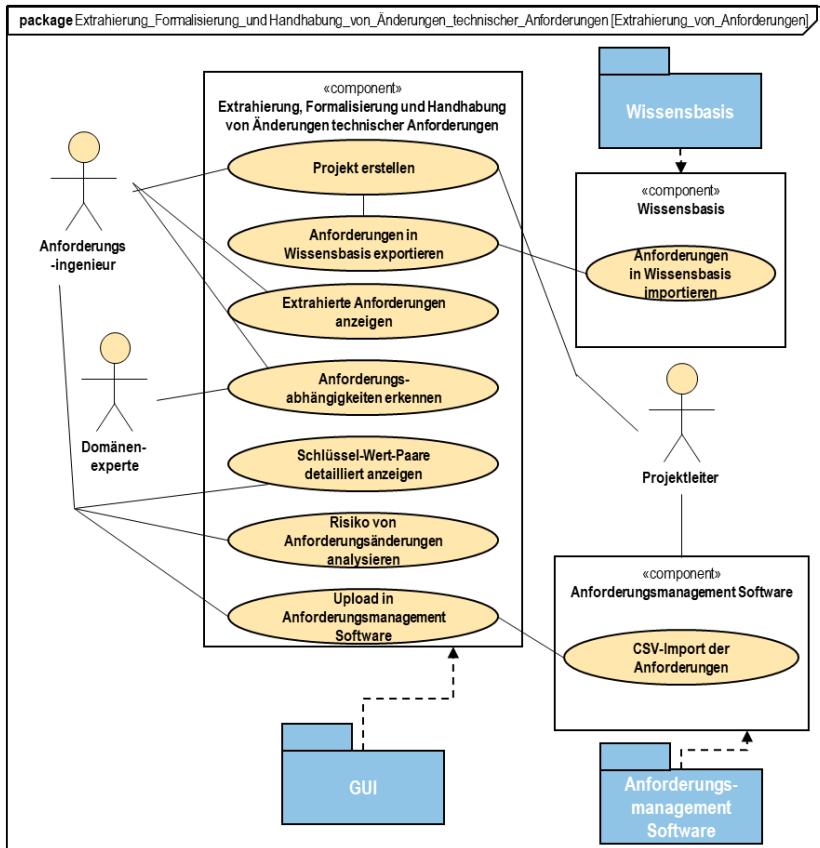


Bild 6-3: Paket-Diagramm der Use-Cases, Akteure und Software-Komponenten

Das Use-Case-Diagramm zu „Extrahierte Anforderungen anzeigen“ ist beispielhaft in Bild 6-4 dargestellt. In diesem Use-Case-Diagramm wird detailliert, welche Informationen dem Anforderungsingenieur angezeigt werden, wenn ihm extrahierte Anforderungen angezeigt werden. Beispiele hierfür sind verschiedene Attribute der Anforderungen, wie „ID“, „Anforderungsart“, „Anforderungsbeschreibung“ oder „Extrahierte Schlüssel-Wert-Paare“. Außerdem soll ihm eine Detailansicht der extrahierten Anforderungen angezeigt werden. In dieser Detailansicht werden weitere Informationen der Anforderungen dargestellt, wie z. B. Anforderungsabhängigkeiten. Außerdem hat der Anforderungsingenieur die Möglichkeit, Schlüssel-Wert-Paare (die in der betrachteten Anforderung enthalten sind), hinzuzufügen oder zu löschen (falls beispielsweise ein Schlüssel-Wert-Paar falsch erkannt worden ist). Insgesamt wurden 6 Use-Case-Diagramme mit insgesamt 73 Use-Cases erstellt (siehe Anhang A4.2).

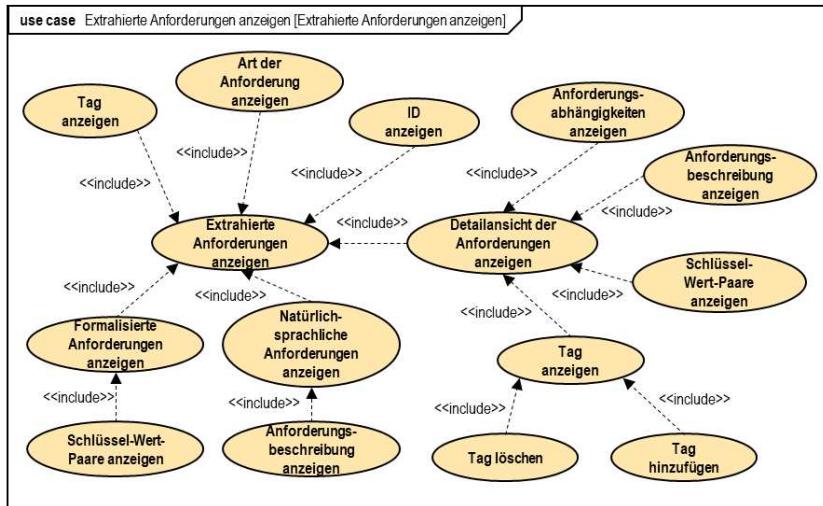


Bild 6-4: Ausschnitt aus dem Use-Case-Diagramm "Extrahierte Anforderungen anzeigen"

Aktivitäten modellieren

Für die vier Use-Cases, welche die vier Methoden zur Extrahierung und Formalisierung technischer Anforderungen repräsentieren, werden Aktivitätsdiagramme erstellt (siehe Anhang A4.3). Aktivitätsdiagramme bilden den Ablauf der Use-Cases ab. Außerdem werden Objektflüsse modelliert. Diese bilden ab, welche Inputs und Outputs an Daten innerhalb der Use-Cases erhalten und verarbeitet werden [Wei14]. Insgesamt wurden vier Aktivitätsdiagramme erstellt. Beispielsweise ist das Aktivitätsdiagramm des Use-Cases „Extrahierung und Klassifizierung von Anforderungen“ in Bild 6-5 dargestellt. Zu Beginn muss ein PDF-Lastenheft hochgeladen werden. Es werden Textelemente extrahiert. Anschließend werden die Textelemente hinsichtlich Anforderungsart klassifiziert. Die Klassifizierung erfolgt durch ein KI-Modell. Outputs sind die extrahierten und klassifizierten Anforderungen.

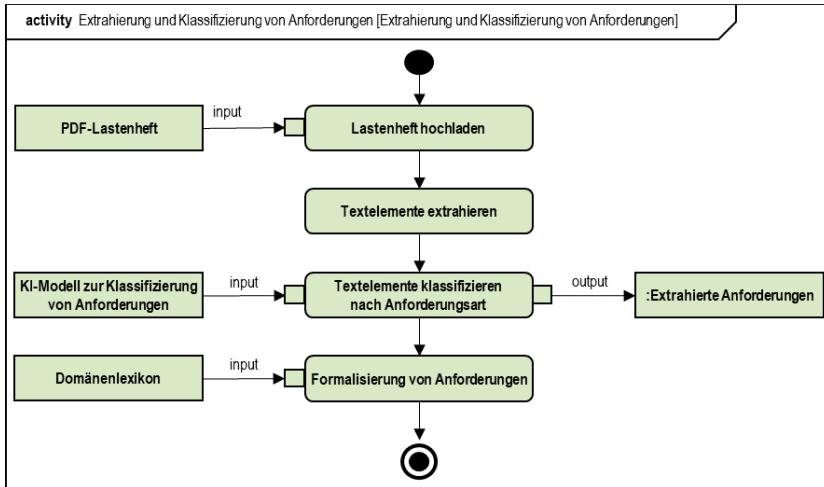


Bild 6-5: Aktivitätsdiagramm "Extrahierung und Klassifizierung von Anforderungen"

Anwendungsfälle realisieren

Parallel zur Realisierung des Software-Werkzeugs werden Diagramme erstellt, um die Erfüllung der Erfolgsfaktoren und Use-Cases durch die Komponenten des Software-Werkzeugs nachzuvollziehen. Es wird ein Paketdiagramm erstellt, in dem die hierarchische Struktur der Komponenten des Software-Werkzeugs abgebildet sind (siehe Anhang A4.4, Bild A-71). Außerdem wird ein Komponentendiagramm erstellt, um den Austausch von Daten zwischen den Komponenten des Software-Werkzeugs zu visualisieren (siehe Anhang A4.4, Bild A-70). Mittels Klassendiagramm (siehe Anhang A4.4, Bild A-68) werden Attribute und Vererbungen von verschiedenen Klassen dargestellt. Beispielsweise besitzt die Klasse „Anforderung“ die folgenden Attribute: „ID“, „Tags“, „Beschreibung“, „Art der Anforderung“, „Schlüssel“, „Wert“, „Art der Abhängigkeit“. Die Klasse „Anforderung“ erbt die Attribute „Schlüssel“ und „Wert“ von der Klasse „Schlüssel-Wert-Paar“ und das Attribut „Art der Abhängigkeit“ von der Klasse „Anforderungsabhängigkeit“.

Das Modell der Software-Architektur bildet das Verhalten und die Struktur des Software-Werkzeugs ab. Es wurde der Systemkontext modelliert, um Interaktionen mit verschiedenen Akteuren abzubilden. Die Interaktionen wurden durch Use-Cases beschrieben und in Aktivitätsdiagrammen detailliert. Das Modell wird genutzt, um die Methoden und das Software-Werkzeug gemäß den Erfolgsfaktoren und unter Berücksichtigung des Systemkontexts zu entwickeln.

6.1.2 Referenzprozess der Methodik-Anwendung

In der Software-Architektur werden funktionale und strukturelle Eigenschaften des Software-Werkzeugs beschrieben. Darauf aufbauend wird ein Referenzprozess entwickelt, der als Ausgangspunkt für das unternehmensspezifische Tailoring [WRF15] dient. Der Referenzprozess wird in Anlehnung an OLEFF [Ole22] in die drei Schritte: *Potenzialanalyse* der Methodik-Anwendung einer Organisationseinheit (1), *Inhaltliche Einordnung* der Methodik in den Entwicklungsprozess einer Organisationseinheit (2) und *Projektspezifische Ausgestaltung* der Methodik-Anwendung (3) unterteilt. Der Referenzprozess der Methodik-Anwendung ist in Bild 6-6 dargestellt.

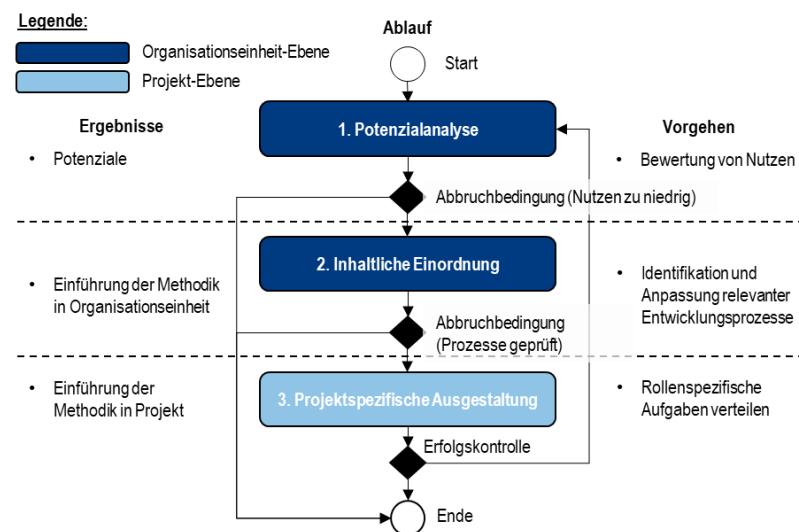


Bild 6-6: Referenzprozess der Methodik-Anwendung

In der *Potenzialanalyse* wird der Mehrwert der Methodik-Anwendung für die Organisationseinheit ermittelt. Die Bewertung wird von der Unternehmensführung durchgeführt. Es werden die vier Kern-Anwendungsfälle (bzw. Methoden), die in der Software-Architektur modelliert worden sind, betrachtet: Extrahierung von Anforderungen, Formalisierung von Anforderungen, Abhängigkeitsanalyse von Anforderungen sowie Risikomanagement von Anforderungsänderungen. Für die Anwendungsfälle muss der Nutzen bewertet werden.

Zur Potenzialanalyse müssen verschiedene Kennzahlen herangezogen werden. Sechs Beispiele für Kennzahlen sind im Folgenden aufgelistet.

- 1) Durchschnittliche Anzahl an Anforderungen
- 2) Durchschnittliche Anzahl an Seiten natürlichsprachlicher Anforderungsdokumente
- 3) Durchschnittlich benötigte Zeit für die Extrahierung von Anforderungen
- 4) Durchschnittliche Anzahl benötigter Parameter
- 5) Durchschnittliche Anzahl an Änderungsanträgen
- 6) Durchschnittliche Änderungskosten

Kennzahlen 1-4 sind für die Bewertung des Nutzens der Extrahierung und Formalisierung von Anforderungen anwendbar. Beispielsweise nimmt der Aufwand für die manuelle Extrahierung von Anforderungen mit der Anzahl an Anforderungen oder der Anzahl an Seiten natürlichsprachlicher Anforderungsdokumente zu. Für die Abhängigkeitsanalyse von Anforderungen und das Risikomanagement von Anforderungsänderungen sind beispielsweise die Anzahl an Änderungsanträgen oder die Änderungskosten relevant. Falls nach der Bewertung der Nutzen von mindestens einem Anwendungsfall als hoch bewertet wird, dann werden die weiteren Schritte durchlaufen. Falls der Nutzen jedes Anwendungsfalls als niedrig bewertet wird, dann müssen die weiteren Schritte nicht durchgeführt werden und die Einführung der Methodik wird abgebrochen (Abbruchbedingung). Ergebnis von Schritt 1 ist das Potenzial der Methodik-Anwendung.

In der *inhaltlichen Einordnung* wird der Entwicklungsprozess der Organisationseinheit analysiert. Die Analyse erfolgt durch prozessverantwortliche Fach- und Führungskräfte. Es müssen relevante Entwicklungsprozesse identifiziert werden und für die Anwendung der Methodik angepasst werden. Grundlage für den unternehmensspezifischen Entwicklungsprozess komplexer technischer Systeme ist häufig die Methodik der Richtlinie VDI/VDE 2206:2021 [VDI21]. Die Anwendungspotenziale der Methodik werden in das V-Modell der VDI/VDE 2206:2021 eingeordnet (siehe Bild 6-7).

Die Extrahierung von Anforderungen ist in der Aufgabe Anforderungserhebung relevant. Natürlichsprachliche Anforderungsdokumente (speziell Lastenhefte) werden effizient analysiert. Die Formalisierung von Anforderungen ist für die Modellierung & Analyse relevant. Ein Anwendungsfall ist beispielsweise die generative Entwicklung von Bauteilen. Parameter, die für die generative Entwicklung relevant sind, werden effizient aus Anforderungen extrahiert und sind maschinenlesbar. Für die Abhängigkeitsanalyse von Anforderungen ist ebenfalls die Modellierung & Analyse relevant. Die Wirkkettenmodellierung [GWK22] ist beispielsweise ein Anwendungsfall. Durch die Abhängigkeitsanalyse von Anforderungen werden effizient Abhängigkeiten zwischen Elementen des Artefakts „Anforderung“ erkannt. Die Berechnung des Änderungsrisikos von Anforderungen ist im Anforderungsmanagement relevant. Es wird das Änderungsmanagement unterstützt. Ergebnis von Schritt 2 ist die Einführung der Methodik in der Organisationseinheit, bzw. in den Entwicklungsprozess.



Bild 6-7: Einordnung der Anwendungspotenziale der Methodik in das V-Modell der VDI/VDE-Richtlinie 2206:2021

In der *projektspezifischen Ausgestaltung* müssen Entwicklungsteams zur Anwendung der Methodik befähigt werden. Die Durchführung des Schritts erfolgt durch Projektleiter. Einerseits müssen die benötigten Informationen zur Anwendung der Methodik bereitgestellt werden. Andererseits müssen Projektbeteiligte das notwendige Wissen zur Anwendung der Methodik haben. In der Software-Architektur wurden die notwendigen Informationsbedarfe zur Anwendung der Methodik modelliert. Dazu zählen natürlichsprachliche Lastenhefte, ein Domänenlexikon und historische Änderungsdaten. Es muss projektspezifisch geprüft werden, ob die notwendigen Informationen vorliegen oder generiert werden müssen. Beispielsweise müssen Strukturregeln definiert werden, um Parameter durch die Formalisierung von Anforderungen zu erkennen.

In Abhängigkeit der Rollen müssen Projektbeteiligte bestimmte Aktivitäten zur Anwendung der Methodik ausführen. Diese Aktivitäten sind in der Software-Architektur in Form von Aktivitätsdiagrammen modelliert. Projektbeteiligte müssen im Umgang mit dem Software-Werkzeug geschult werden. Anforderungsingenieure müssen beispielsweise Lastenhefte einlesen und Schlüssel-Wert-Paare festlegen können. Domänenexperten müssen Anforderungsabhängigkeiten bewerten können. Das Ergebnis von Schritt 3 ist die Befähigung der Projektbeteiligten zur Anwendung der Methodik.

Die Methodik-Anwendung muss einer Erfolgskontrolle unterzogen werden. Es muss nach Projekt-Abschluss geprüft werden, ob bestimmte Kennzahlen (die u. a. in Schritt 1 festgelegt worden sind) verbessert worden sind. Dies können beispielsweise die Zeit zur Extrahierung von Anforderungen oder die Änderungskosten sein. Falls sich die Kennzahlen nicht verbessern, dann besteht ggf. Anpassungsbedarf der Ziele in Schritt 1.

6.2 Entwicklung der Methoden

Die Methodik zur automatisierten Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen besteht aus vier Methoden. Diese werden im Folgenden beschrieben. Der Nutzen der Methoden wird jeweils anhand eines Anwendungsfalls erläutert.

6.2.1 Extrahierung von Anforderungen

Ziel der *Extrahierung von Textsegmenten aus Lastenheften* ist es, den natürlich-sprachlichen Text aus dem Lastenheft vorzubereiten, sodass nachfolgende Algorithmen diesen verarbeiten können, um z. B. Anforderungen zu klassifizieren oder zu formalisieren. Das branchenübliche Dateiformat von Lastenheften ist PDF. Es wird ein PDF-Parser verwendet, um verschiedene Elemente zu extrahieren. Elemente sind „Fließtext“, „Listen“ und „Tabellen“. Innerhalb der Elemente wird ein Satzteiler-Algorithmus angewendet, um abgeschlossene Sätze zu erkennen. Die Regeln zur Satztrennung werden an 2 Sätzen (Satz 1 und Satz 2) erläutert. Satz 1 ist ein beliebiger Satz im Lastenheft. Satz 2 ist der auf Satz 1 folgende Satz. Der Software-Code zur Trennung von Sätzen ist auszugsweise in Anhang A4.5, Bild A-72 dargestellt. Die Sätze 1 und 2 werden nicht getrennt, wenn:

- Satz 1 mit einem Satzzeichen endet, welches indiziert, dass Satz 1 nicht abgeschlossen ist: ":", "," , "-" oder "/";
- Satz 2 mit einem Satzzeichen startet, welches indiziert, dass Satz 1 nicht abgeschlossen ist: ":", "," , "-" oder "/";
- oder Satz 2 mit einem Kleinbuchstaben beginnt.

Anschließend werden Co-Referenzen mittels Coreferee¹ aufgelöst. Hierzu wird satzübergreifend POS-Tagging durchgeführt. Wenn innerhalb eines Satzes ein POS-Tag identifiziert wird, welches auf eine Co-Referenz hindeutet („Determinatoren“ und „Pronomen“ [Bet22]), dann wird im vorherigen Satz das referenzierte Nomen gesucht und die Co-Referenz aufgelöst. Das Vorgehen wird anhand eines Beispiels erläutert (siehe Bild 6-8). In dem Beispiel wird ein Fließtext aus einem Lastenheft zur Entwicklung eines Türmoduls betrachtet. Zunächst wird der PDF-Parser durchgeführt, um den Fließtext zu erkennen. Die Sätze im Fließtext werden getrennt. In dem Beispiel muss bei der Trennung der Sätze beachtet werden, dass der „.“ im ersten Satz in der Zahl „300.000“ nicht das Ende eines Satzes markiert. Anschließend werden Co-Referenzen aufgelöst. Im zweiten Satz wird das Wort „Es“ durch „Das Türmodul“ ersetzt. Dieses Vorgehen wird für weitere Elemente im Lastenheft wiederholt. Der Software-Code zum Auflösen von Co-Referenzen ist auszugsweise in Anhang A4.5, Bild A-73 dargestellt.

¹ <https://spacy.io/universe/project/coreferee>, zuletzt geprüft am 02.01.2023

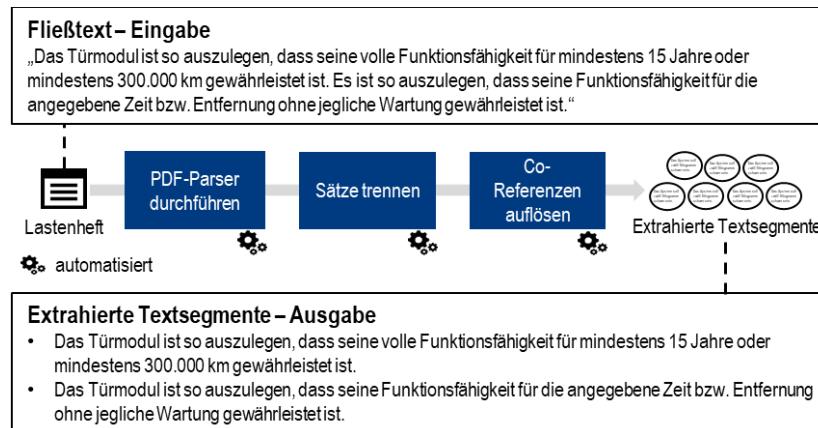


Bild 6-8: Extrahierung von Textsegmenten aus Lastenheften im PDF-Format

Ergebnis der Extrahierung von Textsegmenten sind getrennte Sätze, die aus PDF-Dokumenten extrahiert worden sind und keine Co-Referenzen enthalten. Zur Bildung der Sätze werden Elemente aus dem Fließtext, Listen und Tabellen berücksichtigt. Die Extrahierung der Textsegmente erfolgt automatisiert.

Klassifizierung nach Anforderungsart

Ziel der *Klassifizierung nach Anforderungsart* ist es, die extrahierten Textsegmente nach „funktionale Anforderung“, „nicht-funktionale Anforderung“ oder „keine Anforderung“ zu klassifizieren. Das Vorgehen wird beispielhaft in Bild 6-9 dargestellt.

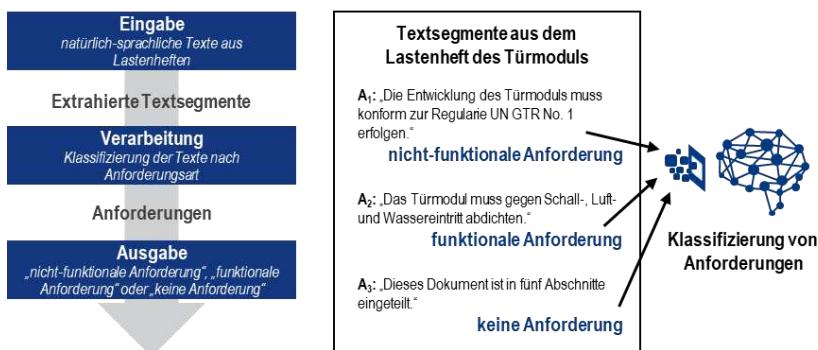


Bild 6-9: Beispiel für die Klassifizierung nach Anforderungsart. Der Output ist in dunkelblau unterhalb der Anforderungen dargestellt.

Für die Klassifizierung nach Anforderungsart wird aufbauend auf dem Ansatz von KURTANOVIC und MAALEJ Künstliche Intelligenz, speziell: ein überwachtes maschinelles Lernverfahren, angewendet [KM17]. Hierzu werden die extrahierten Textsegmente klassifiziert. Die Klassifizierung nach Anforderungsart ist in die fünf Schritte Generierung von Daten, Vorverarbeitung der Daten, Training des Klassifizierungsmodells, Klassifizierung der Daten und Ausgabe der Daten für eine Anforderungsmanagement-Software aufgeteilt.

Generierung von Daten

Um Anforderungen mittels maschinellem Lernen zu klassifizieren, müssen Daten verfügbar sein. Diese Daten werden aus Spezifikationsdokumenten extrahiert. Wenn überwachtes maschinelles Lernen verwendet wird, dann müssen diese Daten gekennzeichnet werden („funktional“, „nicht-funktional“ oder „keine Anforderung“). In dieser Methode wird GPT-J (siehe Abschnitt 3.1) für die Augmentation von englischen Anforderungsdaten genutzt. Die generierten Daten (reale und augmentierte Daten) müssen vorverarbeitet werden (nächster Schritt), sodass ein Klassifizierungsmodell trainiert werden kann.

Vorverarbeitung der Daten

Zur Vorverarbeitung der Anforderungsdaten werden verschiedene Operationen durchgeführt. Für jeden Satz aus den extrahierten Textsegmenten werden die Wörter als Tokens formatiert (Tokenisierung). Hierzu wird der Tokenizer von spaCy¹ verwendet. Die einzelnen Tokens werden als Vektoren mit Hilfe des vortrainierten Universal Sentence Encoders² von Tensorflow dargestellt.

Training des Klassifikationsmodells

Zur Klassifizierung von Anforderungen hinsichtlich „funktional“, „nicht-funktional“ oder „keine Anforderung“ müssen Modelle trainiert werden. Dazu werden die (im vorherigen Schritt) generierten Daten benötigt. Als Teil dieser Arbeit werden verschiedene Modelle mittels Scikit-learn³ und Tensorflow⁴ trainiert und deren Leistungsfähigkeit verglichen. Diese Modelle werden untersucht, weil frühere Untersuchungen auf eine hohe Leistungsfähigkeit in der Klassifizierung von Anforderungsarten hindeuten [KM17]:

- Support Vector Machine (SVM) mit radialem Basisfunktions-Kernel,
- Random Forest (RF) und
- Neuronales Netz (NN) mit zwei versteckten Schichten und einer Ausgabe-Schicht.

¹ <https://spacy.io/api/tokenizer>

² https://www.tensorflow.org/hub/tutorials/semantic_similarity_with_tf_hub_universal_encoder

³ <https://scikit-learn.org>

⁴ <https://www.tensorflow.org>

Klassifizierung der Daten

Die vorverarbeiteten Anforderungsdaten werden durch das Modell klassifiziert. Es werden verschiedene Modelle zur Klassifizierung der Anforderungsdaten und zum Vergleich der Leistungsfähigkeit angewendet (siehe oben).

Ausgabe der Daten für die Anforderungsmanagement-Software

Die klassifizierten Anforderungen werden im CSV-Format exportiert. Pro Anforderung werden die Attribute ID, Anforderungstext und Anforderungstyp („funktional“ oder „nicht-funktional“) ausgegeben. Daten, welche als „keine Anforderung“ klassifiziert worden sind, werden für den Import nicht berücksichtigt. Für den Import in Anforderungsmanagement-Software sind werkzeugspezifische Schnittstellen zu beachten. So müssen z. B. für einen Import in IBM DOORS die Namen der Spaltenüberschriften in der CSV-Datei mit den Namen der Attribute der Anforderungen im Projekt übereinstimmen. Der Ablauf der Klassifizierung nach Anforderungsart ist in Bild 6-10 visualisiert. In der Generierung der Daten müssen teilweise Daten manuell gekennzeichnet werden. Die restlichen Schritte sind automatisiert. Der Algorithmus zur Klassifizierung von Anforderungen ist auszugsweise in Anhang A4.6 dargestellt.

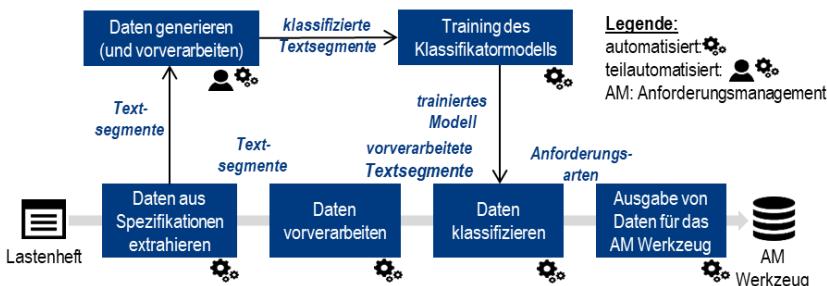


Bild 6-10: Ablauf der Klassifizierung nach Anforderungsart [GPB22]

Ergebnis der Klassifizierung von Anforderungen ist eine Liste von Textsegmenten, die als Attribute „Anforderungsart“ und „textuelle Beschreibung“ besitzen. Die textuelle Beschreibung enthält den extrahierten natürlich-sprachlichen Text aus dem Lastenheft. Die Anforderungsart gibt das Ergebnis der Klassifizierung wieder. Textsegmente, die als Anforderungsart „keine Anforderung“ besitzen, sind keine Anforderungen und werden nicht weiter betrachtet. Weitere Ausprägungen dieses Attributs sind „funktionale Anforderung“ und „nicht-funktionale Anforderung“. Textsegmente, welche diese Anforderungsart als Attribut besitzen, sind Anforderungen und werden für die weiteren Methoden der entwickelten Methodik berücksichtigt.

Die Klassifizierung durch maschinelles Lernen erfolgt automatisiert. Durch das Klassifizierungsmodell können unstrukturierte Anforderungen sowie Anforderungen mit niedriger Qualität verarbeitet werden. Durch den CSV-Export können Anforderungen in branchenübliche Anforderungsmanagement-Software eingelesen werden.

Anwendungsfall der Methode

Eine aktuelle Herausforderung in der Entwicklung von Automobilen besteht darin, dass Automobilhersteller aufgrund der UN-ECE R156 Regularie verpflichtet sind, die Auswirkungen von technischen Änderungen auf Softwarekomponenten jederzeit nachvollziehen zu können [GWK22]. Aufgrund des komplexen Zusammenspiels von Systemelementen (wie Softwarekomponenten) haben Änderungen nicht nur unmittelbare Auswirkungen auf ein bestimmtes Systemelement, sondern pflanzen sich innerhalb des Gesamtsystems fort und führen zu steigenden Kosten oder sogar zum Scheitern des Projekts [EEC07]. Die daraus resultierenden Auswirkungen sind ohne systematischen Ansatz und Softwareunterstützung kaum zu bewerten. Die Auswirkungsanalyse kann genutzt werden, um vorhandene Informationen zu analysieren und den Entwicklern zu helfen, die Auswirkungen von auftretenden Änderungswünschen abzuschätzen [GW21]. Die Modellierung von Wirkketten ist eine Möglichkeit, ein solches Informationsmodell zu erstellen. Hierbei werden Artefakte miteinander verknüpft, um eine Nachverfolgbarkeit herzustellen (siehe Bild 6-11).

Die „Model-based Effect Chain Analysis“ (MECA) Methode wurde von GRÄBLER, WIECHEL, KOCH, PREUß und OLEFF veröffentlicht [GWK22]. Die MECA-Methode wird im Anhang A2.1 ausführlich beschrieben. Ein Problem in der Wirkkettenmodellierung besteht darin, dass die detaillierte Modellierung bestimmter Artefakte aufwändig ist. Dies betrifft unter anderem das Umgebungs-Artefakt „Regularien“. In Regularien werden bestimmte Anforderungen an Automobile gestellt. Innerhalb des Projekts zur Wirkkettenmodellierung [GWK22] wurde von Praktikern berichtet, dass Regularien aktuell lediglich als Klasse „Regularie“ mit den Instanzen auf Regularien-Ebene – z. B. „UN/ECE R26“ – angelegt werden. Dieser niedrige Detailgrad wurde aufgrund des Aufwands zur Extrahierung und Klassifizierung der Anforderungen aus den Regularien gewählt. Wenn eine technische Änderung auftritt, dann ist nicht direkt ersichtlich, ob die Anforderungen der verknüpften Regularien weiterhin erfüllt sind. Ein Mehrwert besteht darin, die Anforderungen, die in den Regularien dokumentiert sind, direkt mit den Entwicklungsaufgaben zu verknüpfen. Außerdem ist eine Verknüpfung der Anforderungen aus den Regularien mit Entwicklungsaufgaben notwendig, um effizient zu prüfen, ob sämtliche Anforderungen in der Entwicklung des Automobils berücksichtigt worden sind.

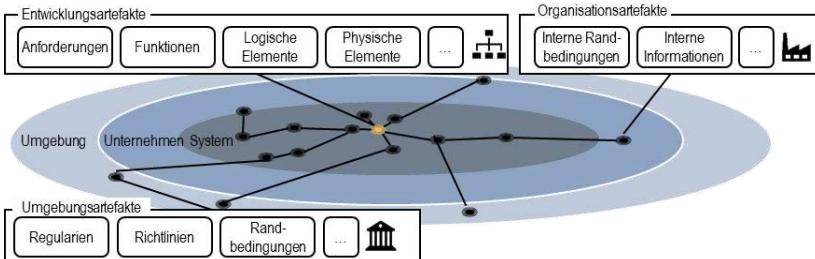


Bild 6-11: Zertifizierungsgerechte Wirkkettenmodellierung [GWK22]

Die entwickelte Methode zur Extrahierung und Klassifizierung von Anforderungen unterstützt die Wirkkettenmodellierung, indem Anforderungen effizient aus Regularien extrahiert werden. Diese Anforderungen werden wiederum im Systemmodell abgebildet und mit Elementen der Wirkkette verknüpft. Dadurch kann im Rahmen einer Analyse festgestellt werden, ob die verknüpften Elemente die Anforderungen aus den Regularien erfüllen und das System zertifizierungsgerecht entwickelt worden ist. Außerdem kann bei Änderungen an Elementen nachvollzogen werden, ob die Änderungen Einfluss auf die Zertifizierung des Systems haben.

6.2.2 Formalisierung von Anforderungen

Ziel der Formalisierung von Anforderungen ist es, natürlich-sprachliche Anforderungen in eine eindeutige Darstellungsform zu überführen. Dadurch werden Anforderungen eindeutig interpretierbar.

Damit die extrahierten Informationen maschinenlesbar sind und durch Werkzeuge zur generativen Entwicklung interpretiert werden können, ist ein hoher Formalisierungsgrad erforderlich. Aus diesem Grund wird ein Ansatz zum Ausfüllen von Vorlagen in Kombination mit NER gewählt (siehe Abschnitt 3.2). Für diese Aufgabe werden Transformer-Modelle verwendet, da diese die besten Ergebnisse für NER zeigen [WJB21]. Um Anforderungen zu formalisieren, werden Schlüssel-Wert-Paare in natürlich-sprachlichen Anforderungen erkannt. Ein Beispiel für ein Schlüssel-Wert-Paar ist in Bild 6-12 dargestellt.

Das Schlüssel-Wert-Paar der Anforderung "Die Lebensdauer des Systems muss min. 300.000 km betragen." muss extrahiert werden. In diesem Beispiel ist der Schlüssel "lifespan_range". Die Lebensdauer wird in einer Entfernungseinheit angegeben. Der Wert des Schlüssels ist " $\geq 300.000 \text{ km}$ ". Das Schlüssel-Wert-Paar lautet also "lifespan_range: $\geq 300.000 \text{ km}$ ". Zur Erkennung des Schlüssel-Wert-Paares werden verschiedene Strukturregeln und Textindikatoren verwendet, die im Domänenlexikon definiert sind. Für jedes Schlüssel-Wert-Paar werden Wertklassen definiert. In diesem Beispiel ist der Wert eine Zahl (in Kombination mit dem relationalen Operator "min." bzw. " \geq " und der

Einheit Kilometer bzw. "km"). Werte müssen im Zusammenhang mit der Formalisierung von Anforderungen nicht unbedingt reelle Zahlen sein. In der Beispielanforderung "Das Türmodul soll mittels additiven Fertigungsverfahren hergestellt werden" ist der Schlüssel "Fertigungsverfahren" und der Wert "additive Fertigung".

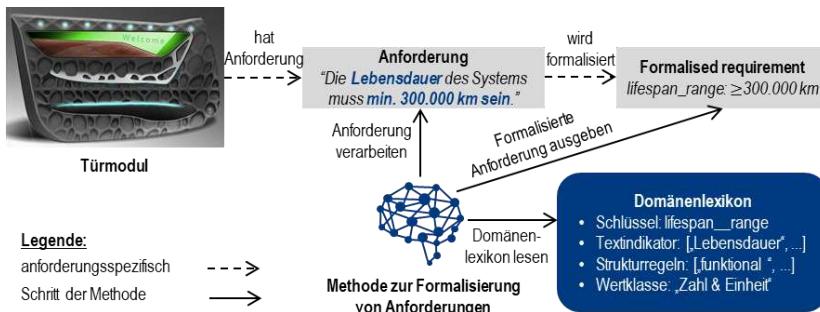


Bild 6-12: Beispiel-Anforderung (Lebensdauer als Entfernungseinheit) eines Türmoduls zur Bildung von Schlüssel-Wert-Paaren [GPB23]

Die Methode zur Formalisierung von Anforderungen besteht aus drei Schritten. *Schritt 1:* Die Anforderungskandidaten müssen identifiziert werden. Um die Laufzeit und die Generierung falsch positiver Ergebnisse des Algorithmus zur Formalisierung von Anforderungen zu reduzieren, sollen nur bestimmte Anforderungen auf das Vorhandensein bestimmter Schlüssel-Wert-Paare geprüft werden. Eine Anforderung wird nur dann nach einem Schlüssel-Werte-Paar untersucht, wenn alle Strukturregeln erfüllt sind. Anforderungen, die diese Bedingungen erfüllen, werden Anforderungskandidaten genannt. Strukturregeln werden pro Schlüssel-Wert-Paar definiert. Diese Strukturregeln werden als Positiv- oder Negativliste verwendet. Zum Beispiel lauten die Strukturregeln für die Anforderung an die Lebensdauer des Türmoduls:

- *Strukturregel 1:* "Der Schlüssel "lifespan_range" kann nur dann in einer Anforderung vorkommen, wenn eine Entfernungseinheit vorhanden ist". (positiv)
- *Strukturregel 2:* "Der Schlüssel "lifespan_range" kann in einer funktionalen Anforderung nicht vorkommen." (negativ)

Ist eine Entfernungseinheit (z.B. "km") in einer Anforderung nicht vorhanden, so wird die Anforderung nicht daraufhin überprüft, ob ein Schlüssel-Wert-Paar für den Schlüssel "lifespan_range" existiert (Strukturregel 1). Darüber hinaus können Metadaten der Anforderungen zur Bildung von Strukturregeln verwendet werden. Ein Schlüssel-Wert-Paar zum Schlüssel "lifespan_range" kann nicht in einer funktionalen Anforderung vorkommen, weshalb funktionale Anforderungen nicht auf dieses Schlüssel-Wert-Paar geprüft werden (Strukturregel 2).

Schritt 2: Die Anforderungskandidaten werden gefiltert, indem sie mit Textindikatoren von Schlüsseln verglichen werden. Textindikatoren für "lifespan_range" sind z. B. "Lebensdauer" oder "Haltbarkeit". Diese Textindikatoren werden manuell im Domänen-Lexikon definiert. Die Definition kann teilweise automatisiert werden, indem Synonyme aus lexikalischen Datenbanken wie WordNet [Fel05] identifiziert werden. Alle Wörter der Anforderungskandidaten (nach Entfernung der Stopwörter) werden als Kandidaten für Schlüssel-Wert-Paare verwendet. Die Schlüssel-Wert-Paar-Kandidaten werden lemmatisiert und direkt mit den Lemmata der Textindikatoren verglichen und nach Übereinstimmungen durchsucht.

Schritt 3: Wird mindestens eine Übereinstimmung gefunden, werden die Schlüssel-Wert-Paar-Kandidaten daraufhin überprüft, ob sie einen Wert für den passenden Schlüssel enthalten. Je nach Schlüssel werden unterschiedliche Ansätze zur Erkennung von Werten verwendet. Zum Beispiel kann der Schlüssel "lifespan_range" die Werte 0 bis n und eine Entfernungseinheit (wie "km" oder "kilometer") annehmen. Diese Wertklasse wird als "Zahl & Einheit" bezeichnet. Reguläre Ausdrücke (RegEx) werden zur Erkennung verwendet. Wenn es mehrere mögliche Werte für einen Schlüssel gibt, die nicht einer festen Syntax folgen, wie die Werteklasse "Zahl & Einheit", wird NER verwendet. Zum Beispiel erfordert der Schlüssel "manufacturing_site" einen Wert für den Standort der Produktionsstätte (z. B. Land "Deutschland" oder Stadt "Paderborn"). Für einige NER-Aufgaben existieren bereits trainierte Modelle. In diesem Ansatz werden die NER-Modelle von spaCy für die Werteklasse "Standort" verwendet. Für andere Werteklassen wie "Material" oder "Fertigungsprozess" werden NER-Modelle speziell trainiert. Wenn es nur wenige mögliche Werte für einen Schlüssel gibt, wird ein String-Matching-Ansatz verwendet. Eine Liste von Zeichenketten für diesen Wert wird manuell definiert. Die lemmatisierten Zeichenfolgen werden mit den lemmatisierten Wertkandidaten abgeglichen. Zum Beispiel erfordert der Schlüssel "energy_type" einen Wert für die Art der Energie für den Antrieb eines Fahrzeugs (z. B. "Diesel" oder "elektrisch"). Wenn der Schlüssel-Wert-Paar-Kandidat keinen Wert enthält, der mit dem Schlüssel übereinstimmt, wird er als Kandidat für diesen Schlüssel verworfen. Enthält die Anforderung einen einzigen passenden Wert, wird dieser als Wert für das Schlüssel-Wert-Paar verwendet. Wenn es nicht möglich ist, einen eindeutigen Wert zu ermitteln, müssen alle Wertkandidaten dem Benutzer zur Überprüfung vorgelegt werden. Die Methode ist in Bild 6-13 dargestellt. Der Software-Code zur Formalisierung von Anforderungen ist auszugsweise in Anhang A4.7 dargestellt. Außerdem werden die implementierten Domänenlexika der Schlüssel-Wert-Paare in Anhang A4.8 dargestellt (siehe Bild A-77 bis Bild A-96).

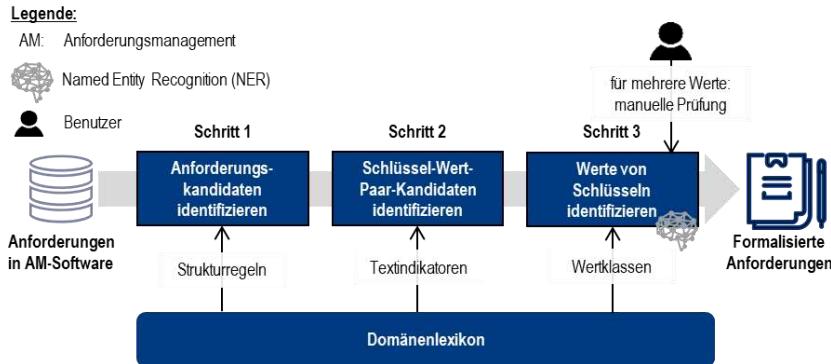


Bild 6-13: Ablauf der Formalisierung von Anforderungen [GPB23]

Ergebnis der Formalisierung von Anforderungen sind Schlüssel-Wert-Paare, die in den klassifizierten Anforderungen erkannt worden sind. Schlüssel sind Parameter, wie z. B. „Lebensdauer“. Werte sind die Ausprägungen der Parameter, beispielsweise „min. 300.000 km“. Die zu erkennenden Schlüssel-Wert-Paare werden im Domänenlexikon definiert. In diesem Dokument werden Regeln festgelegt, um Schlüssel-Wert-Paare zu erkennen. Die formalisierten Anforderungen, bzw. die Schlüssel-Wert-Paare können in verschiedene Entwicklungswerzeuge eingelesen werden. Ein Anwendungsfall ist beispielsweise die generative Entwicklung von Bauteilen in Synera. Die Schlüssel-Wert-Paare definieren Restriktionen für die Bauteilentwürfe.

Anwendungsfall der Methode

Bei der generativen Entwicklung werden Geometrien automatisch auf der Grundlage von Restriktionen erzeugt. Die Restriktionen werden aus Anforderungen abgeleitet, die in Lastenheften enthalten sind. Die generierten Geometrien werden im Hinblick auf bestimmte Kriterien optimiert. So kann z. B. ein Türmodul hinsichtlich des Kriteriums "Masse" optimiert werden und gleichzeitig Restriktionen hinsichtlich verschiedener Lastfälle einhalten. In der generativen Entwicklung ist es das Ziel, einen Workflow zu definieren, der die Systemkomponente anhand bestimmter vorgegebener Parameter automatisch generiert. Da der Prozess der manuellen Modellierung einer Systemkomponente sehr zeitaufwändig ist, spart der generative Ansatz Zeit durch einen hohen Automatisierungsgrad. Software-Werkzeuge wie Synera ermöglichen einen solchen Ansatz. Eine beispielhafte Ansicht eines Synera-Workflows zur Erzeugung von Geometrien (in diesem Fall für ein Türmodul) ist in Bild 6-14 dargestellt.

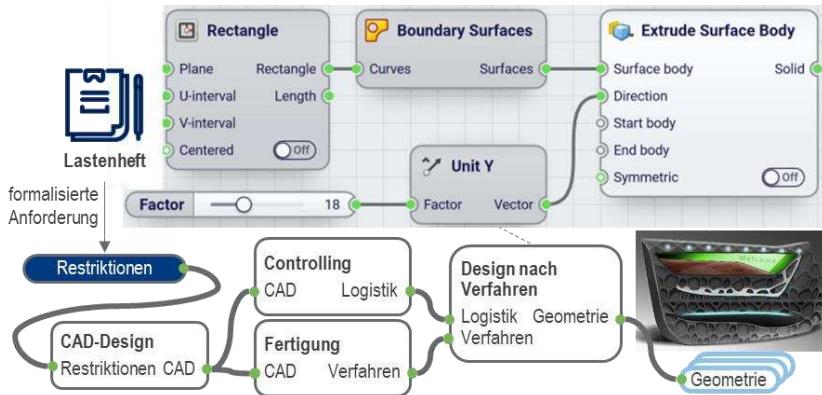


Bild 6-14: Generative Entwicklung eines Türmoduls in Synera [GPB23]

Aktuell müssen sämtliche Parameter, die für die Restriktionen relevant sind, vom Ingenieur manuell aus den Anforderungen, die im Lastenheft definiert sind, ermittelt und in das Werkzeug zur generativen Entwicklung eingegeben werden. Dieser Prozess ist zeitaufwändig und fehleranfällig. Durch die Methoden zur Formalisierung von Anforderungen werden diese Parameter automatisiert ermittelt. Die Anforderungen können beispielsweise in Synera über eine Schnittstelle (CSV-Import) direkt als Input eingelesen werden. Durch die Formalisierung der Anforderungen und das automatische Einlesen der Parameter wird die Zeit an Routinetätigkeiten des Ingenieurs verringert.

6.2.3 Abhängigkeitsanalyse von Anforderungen

Durch die Abhängigkeitsanalyse von Anforderungen sollen Abhängigkeiten zwischen den extrahierten Anforderungen identifiziert werden. In der Vorstudie (siehe Abschnitt 4.1.) wurde erkannt, dass BERT-Modelle für die Erkennung von Anforderungsabhängigkeiten besonders geeignet sind. In Bild 6-15 wird eine Übersicht über die einzelnen Schritte der Abhängigkeitsanalyse von Anforderungen mittels BERT-Modell gegeben.

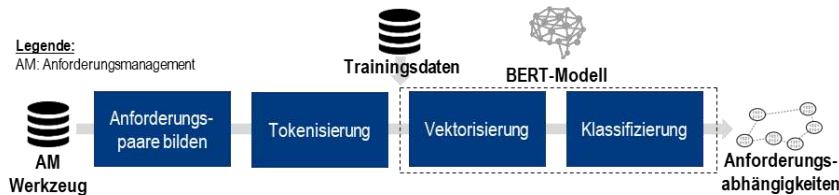


Bild 6-15: Übersicht: Abhängigkeitsanalyse von Anforderungen

Im ersten Schritt werden *Anforderungspaare gebildet*. Durch die Abhängigkeitsanalyse von Anforderungen sollen Abhängigkeiten zwischen sämtlichen Anforderungen des Projekts geprüft werden. Zur Datenvorbereitung werden daher einzelne Anforderungen zu Anforderungspaaren kombiniert, um in der nachfolgenden *Klassifizierung* die Abhängigkeiten zwischen den Anforderungen festzustellen. Es werden exemplarisch zwei Anforderungen betrachtet, um das Vorgehen zu erläutern:

- A1: „Das Türmodul ist so auszulegen, dass seine volle Funktionsfähigkeit für mindestens 15 Jahre oder mindestens 300.000 km gewährleistet ist.“
- A2: „Das Türmodul muss gegen Schall-, Luft- und Wassereintritt abdichten.“

Zur Bildung der Anforderungspaare werden die natürlich-sprachlichen Anforderungsbeschreibungen miteinander verkettet. Beispiel für das Anforderungspaar *A1+A2*: „*Das Türmodul ist so auszulegen, dass seine volle Funktionsfähigkeit für mindestens 15 Jahre oder mindestens 300.000 km gewährleistet ist. Das Türmodul muss gegen Schall-, Luft- und Wassereintritt abdichten.*“

Im nächsten Schritt werden die Wörter in den Anforderungspaaren *tokenisiert*. Der gesamte String mit den verketteten Anforderungsbeschreibungen der Anforderungspaare wird heruntergebrochen in einzelne Strings pro Wort in den Anforderungsbeschreibungen. Dies ist notwendig, um in nachfolgenden Schritten einzelne Wörter in den Anforderungspaaren verarbeiten zu können. Beispiel: *|Türmodul|*, *|Funktionsfähigkeit|* und *|gewährleistet|* sind Tokens aus dem Anforderungspaar.

Anschließend werden die tokenisierten Anforderungspaare *vektoriert*. Die natürlich-sprachlichen Tokens werden durch das BERT-Modell in eine mathematische Darstellung überführt, um im nächsten Schritt für die Klassifizierung der Abhängigkeiten verarbeitbar zu sein. Durch BERT werden den einzelnen Tokens in Abhängigkeit des Kontexts, in dem die Wörter der Tokens auftreten, einem Vektor zugewiesen.

Im letzten Schritt werden die Anforderungspaare *klassifiziert*. Durch das BERT-Modell wird festgestellt, ob zwischen den Anforderungen, die durch das Anforderungspaar repräsentiert werden, eine Abhängigkeit vorliegt. Für die Propagationsanalyse von Anforderungsänderungen ist eine binäre Klassifizierung von Abhängigkeiten („abhängig“ oder „nicht-abhängig“) ausreichend, weshalb in dieser Arbeit eine binäre Klassifizierung durchgeführt wird.

Abhängigkeitsanalyse mit aktivem Lernen

Die Klassifizierung von Anforderungsabhängigkeiten durch BERT Modelle wurde bereits in ARCA erforscht (siehe Abschnitt 4.1). Hierbei wurde festgestellt, dass Forschungsbedarf in der Verbesserung der Transferfähigkeit der Modelle besteht. Für die industrielle Anwendbarkeit ist es notwendig, dass Daten aus neuen Projekten mit einer hohen Leistungsfähigkeit klassifiziert werden können. Potenzial bietet hierbei das Trai-

nung von Modellen mittels aktivem Lernen. Aktives Lernen ist eine Form von Maschinellem Lernen, bei der ein Lernalgorithmus interaktiv ein Orakel (typischerweise einen menschlichen Experten) befragt, um das gewünschte Label für neue Datenpunkte zu erhalten [Set09]. Es hat sich als effektiv erwiesen, um den menschlichen Aufwand bei der Datenanalyse zu reduzieren [ASN19, DPA18]. Dadurch muss nur eine Teilmenge der Daten aus neuen Projekten vom Experten manuell klassifiziert werden. Die restlichen Daten werden automatisiert durch das Modell klassifiziert.

Die Abhängigkeitsanalyse mittels BERT wird durch aktives Lernen erweitert, um die Leistungsfähigkeit der Modelle in der Klassifizierung von Daten aus neuen Projekten zu verbessern und gleichzeitig den Aufwand für die Kennzeichnung neuer Daten zu reduzieren. Das Vorgehen besteht aus den Schritten: Datenvorbereitung, initiales Training, Aktives Lernen und Klassifizierung (siehe Bild 6-16).

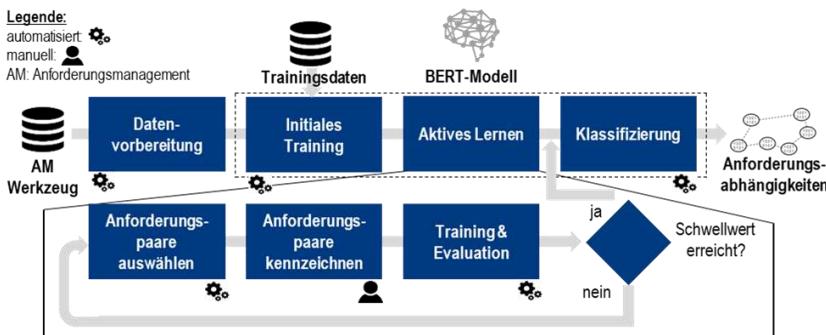


Bild 6-16: Abhängigkeitsanalyse von Anforderungen mit aktivem Lernen [GP23]

In der *Datenvorbereitung* werden, analog zum vorherigen Vorgehen, die Schritte „Anforderungspaares bilden“ und „Tokenisierung“ durchlaufen. Es werden sämtliche Anforderungsdaten, die klassifiziert werden sollen, aus dem Anforderungsmanagement-Werkzeug exportiert und vorbereitet.

Während des *initialen Trainings* wird das BERT-Modell mit Anforderungsdaten fein abgestimmt („Fine-Tuning“). Diese Trainingsdaten sind, analog zum vorherigen Vorgehen, Daten aus vergangenen Projekten, in denen Anforderungsabhängigkeiten gekennzeichnet worden sind.

Im Schritt *Aktives Lernen* werden iterativ Anforderungspaares ausgewählt, die manuell vom Orakel bzw. Experten gekennzeichnet werden müssen. Die *Auswahl der Anforderungspaares* erfolgt unter Anwendung der poolbasierten Stichprobentechnik [Set09]. Aus der Menge der zu klassifizierenden Anforderungspaares wird eine Teilmenge gebildet. Aus dieser Teilmenge werden die unsichersten Daten mittels „Least Confidence Uncertainty Sampling“ bestimmt [Set09]. Das sind die Daten mit der höchsten Un-

sicherheit der Klassifizierung. In jeder Iteration, werden dem Orakel die unsichersten Daten angezeigt, und das Orakel *kennzeichnet* diese Daten manuell [DMP20]. Diese Daten werden der Menge der Trainingsdaten für das aktive Lernen hinzugefügt. Dies ist eine separate Menge an Trainingsdaten, die nur während des aktiven Lernens genutzt wird. Während des Schritts *Training & Evaluation* wird das BERT-Modell fein abgestimmt mit den Trainingsdaten des aktiven Lernens. Anschließend wird das BERT-Modell getestet und es werden Kennzahlen bestimmt, um die Verbesserung des Modells zu messen. Hierzu wird der F1 berechnet. Wenn die Änderung des F1 für eine bestimmte Anzahl an Iterationen unter einem definierten Wert liegt (Abbruchbedingung), dann wird das aktive Lernen beendet. Andernfalls wird die nächste Iteration gestartet. Weitere mögliche Abbruchbedingungen sind beispielsweise eine bestimmte Anzahl an Iterationen oder Erreichung einer definierten Anzahl an gekennzeichneten Daten.

In der *Klassifizierung* werden sämtliche verbleibende Anforderungspaare durch das BERT-Modell hinsichtlich Abhängigkeitsart klassifiziert. Für die weitere Verwendung der Ergebnisse muss der Output des BERT-Modells in ein geeignetes Datenformat transformiert werden – z. B. xls oder ReqIF. Ziel ist es, eine interpretierbare Darstellung der Anforderungen und ihrer Abhängigkeiten zu erstellen. Der Software-Code zur Abhängigkeitsanalyse von Anforderungen ist auszugsweise in Anhang A4.9 dargestellt.

Ergebnis der Abhängigkeitsanalyse von Anforderungen sind die klassifizierten Abhängigkeiten zwischen den Anforderungen. Anforderungsabhängigkeiten werden binär als „abhängig“ und „nicht abhängig“ definiert. Anforderungen sind abhängig, wenn die Änderung einer Anforderung zu einer Änderung der abhängigen Anforderung führt. Es kommt demnach zu einer Änderungspropagation. Die Informationen bezüglich der Abhängigkeiten von Anforderungen sind für verschiedene Anwendungsfälle relevant. Ein Anwendungsfall ist das Risikomanagement von Anforderungsänderungen. Im Vergleich zu bisherigen Forschungsergebnissen besteht der Neuheitsgrad der Abhängigkeitsanalyse in der Verwendung von aktivem Lernen in Kombination mit einem BERT Modell. Durch das aktive Lernen soll die Leistungsfähigkeit des BERT Modells in der Klassifizierung von Anforderungsdaten aus neuen Projekten gesteigert werden.

Anwendungsfall der Methode

Ein Anwendungsfall ist die Bewertung des Änderungsrisikos im Risikomanagement von Anforderungsänderungen [Ole22]. Das Risiko von Anforderungsänderungen besteht aus den Dimensionen „Auswirkungen“ und „Eintrittswahrscheinlichkeit“ einer Änderung. Anforderungsänderungen werden initial durch einen exogenen Änderungsinitiator (z. B. Änderung von Kundenwünschen) verursacht. Die Implementierung dieser Änderung führt zu einer Auswirkung, die direkt mit der Änderung der betroffenen Anforderung verbunden ist (lokale Änderungsauswirkung). Die Implementierung der Änderung kann aber auch Auswirkungen auf nachfolgende Anforderungen haben (konsekutive Änderungsauswirkungen). Diese entstehen aufgrund von Abhängigkeiten zwischen den

Anforderungen und müssen umgesetzt werden, um die Konsistenz und Gültigkeit des Anforderungssatzes zu gewährleisten (Propagationseffekte). Durch Addition der lokalen Änderungsauswirkungen aller Anforderungen auf einem Propagationspfad kann die kollektive Änderungsauswirkung der ursprünglichen Änderungsanforderung bewertet werden (siehe Bild 6-17, [GOP21]).

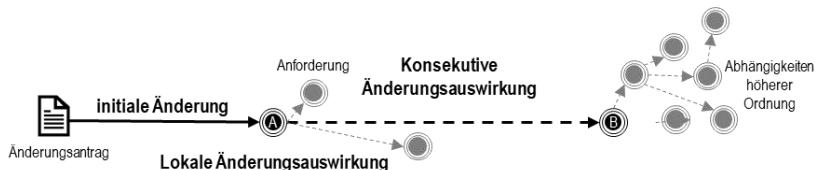


Bild 6-17: Auswirkungen von Anforderungsänderungen in Anlehnung an [GOP21]

Die Anforderungsänderungswahrscheinlichkeit setzt sich aus der exogenen und endogenen Änderungswahrscheinlichkeit zusammen. Exogene Änderungsauslöser lösen eine erste Anforderungsänderung aus. Diese haben nichts mit dem Zusammenspiel der Anforderungen zu tun und kommen von außerhalb des Anforderungssatzes. Endogene Änderungen können nur auftreten, nachdem eine initiale Änderung ausgelöst wurde und entstehen durch Propagationseffekte [GOS20]. Zur Berechnung der exogenen Änderungswahrscheinlichkeit können historische Änderungsdaten analysiert werden [GOP22]. Die endogene Änderungswahrscheinlichkeit kann auf Basis der Anforderungsabhängigkeiten und deren Ausbreitungsverhalten berechnet werden. Das Anforderungsnetz kann mittels PageRank-Algorithmus analysiert werden, um die Konnektivität der Anforderungen zu berechnen [GOP22]. Die Unterscheidung zwischen exogenen und endogenen Änderungsursachen ist in Bild 6-18 dargestellt.

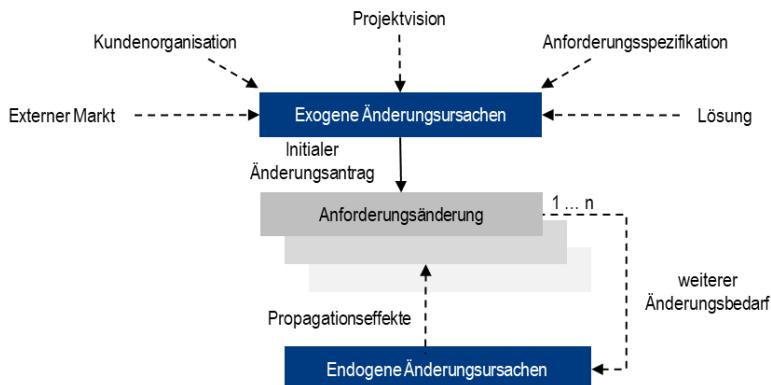


Bild 6-18: Wahrscheinlichkeit von Anforderungsänderungen [GOP22]

Sowohl für die Bewertung der Auswirkungen als auch für die Berechnung der Wahrscheinlichkeit von Anforderungsänderungen ist es notwendig, dass Anforderungsabhängigkeiten bekannt sind. Sie werden benötigt, um die Auswirkungen von konsekutiven Änderungen (bzw. Propagationseffekte) bewerten zu können. Sie werden ebenfalls benötigt, um die Konnektivität des Anforderungsnetzes (z. B. mittels PageRank) berechnen zu können und so Aussagen zur Änderungswahrscheinlichkeit von Anforderungen ableiten zu können. Nach einer empirischen Studie sind ca. 50 % aller Anforderungsänderungen auf solche konsekutiven Änderungen zurückzuführen [GWB09]. Aus diesem Grund ist es notwendig, dass Anforderungsabhängigkeiten bekannt sind, um Änderungskosten zu senken. Ein weiterer Anwendungsfall für die Abhängigkeitsanalyse ist die Wirkkettenmodellierung. Abhängigkeiten von Anforderungen werden automatisiert erkannt, um den Aufwand zur Wirkkettenmodellierung zu reduzieren (siehe Anhang A2.1, Überschrift „Abhängigkeitsanalyse von Anforderungen zur Wirkkettenmodellierung“).

6.2.4 Berechnung der Änderungsrisiken von Anforderungen

In Vorarbeiten des Lehrstuhls für Produktentstehung wurde eine Methodik für das proaktive Risikomanagement von Anforderungsänderungen entwickelt („ProMaRC“, siehe Anhang A2.3). Diese Methodik erlaubt eine frühzeitige Bewertung von Änderungsrisiken von Anforderungen. Forschungspotenziale wurden für die Risikoberechnung von Anforderungsänderungen in der Verbesserung der industriellen Anwendbarkeit identifiziert (siehe Abschnitt 4.2). In der Validierung der Methodik wurde durch Industrie-Anwender erkannt, dass bestimmte Informationen zur Anwendung der ProMaRC-Methodik üblicherweise nicht verfügbar sind. Dazu zählt der Initiator einer Änderung, bzw. die Ursache einer Anforderungsänderung. Daher wird aufbauend auf der ProMaRC-Methodik im Folgenden die Methodik mit dem Ziel einer Erhöhung der praktischen Anwendbarkeit weiterentwickelt.

Risikoberechnung

Zur Ermittlung der Anforderungsänderungswahrscheinlichkeit werden exogene und endogene Änderungswahrscheinlichkeiten berechnet und zu einer Gesamtänderungswahrscheinlichkeit zusammengeführt. Zur Berechnung der exogenen Änderungswahrscheinlichkeit werden die historischen Änderungsdaten analysiert. Die historischen Änderungsdaten bestehen aus der Anzahl an Anforderungsänderungen pro Projekt sowie der Anzahl an Anforderungen pro Projekt. Um die Gültigkeit der historischen Änderungsdaten für das betrachtete Projekt sicherzustellen, werden die historischen Änderungsdaten hinsichtlich Relevanz eingegrenzt. Dies erfolgt durch die manuelle Auswahl von Referenzprojekten. Referenzprojekte sind durch ähnliche Merkmale wie das betrachtete Projekt gekennzeichnet. Um Referenzprojekte zu identifizieren, werden zwölf Projektmerkmale sowie eine kurze Projektbeschreibung erfasst und verglichen. Eine vollständige Liste der Projektmerkmale ist in Anhang A2.6 dargestellt. Exemplarische Merkmale sind Neuheitsgrad, Art des Kunden (Neu- oder Bestandskunde) und

Komplexität des Systems. Bei der Auswahl der Projektmerkmale wurden die Verfügbarkeit der benötigten Informationen zu Projektbeginn berücksichtigt. Die Erfassung der Merkmale und die Auswahl der Referenzprojekte werden manuell von Experten durchgeführt. Auf Basis der Anforderungsabhängigkeiten kann die endogene Änderungswahrscheinlichkeit berechnet werden. Dieser Schritt ist vollständig automatisiert und basiert auf dem PageRank-Algorithmus. Anschließend wird die gesamte Änderungswahrscheinlichkeit automatisiert berechnet.

Berechnung der exogenen Änderungswahrscheinlichkeit

Exogene Änderungen werden von Ursachen außerhalb des Anforderungssatzes ausgelöst. Nach einer empirischen Studie werden ca. 50 % der gesamten Anforderungsänderungen in Projekten durch exogene Änderungen und die restlichen Änderungen werden durch endogene Änderungen verursacht [GWB09]. Für die endogene Änderungswahrscheinlichkeit $P_{endogen}$ wird daher die Annahme getroffen, dass sie 50 % der gesamten Änderungswahrscheinlichkeit P_{gesamt} beträgt (siehe Gleichung 1).

$$P_{endogen} = 0,5 \cdot P_{gesamt}$$

Gleichung 1: Annahme zur endogenen Änderungswahrscheinlichkeit

Die exogene Änderungswahrscheinlichkeit der Anforderungen wird auf Basis von historischen Änderungsdaten von Referenzprojekten berechnet. Die durchschnittliche Anzahl an Änderungen pro Anforderung $\mu_{Änderungen}$ des Projekts wird nach Gleichung 2 berechnet.

$$\mu_{Änderungen} = \frac{\text{Anzahl Änderungen}}{\text{Anzahl Anforderungen}}$$

Gleichung 2: Durchschnittliche Anzahl an Änderungen pro Anforderung

Es wird die Annahme getroffen, dass aufgrund der Ähnlichkeit der Projekte bzw. der Projektmerkmale die durchschnittliche Anzahl an Änderungen pro Anforderung $\mu_{Änderungen,Referenz}$ der Referenzprojekte der durchschnittlichen Anzahl an Änderungen pro Anforderungen $\mu_{Änderungen,Projekt}$ des betrachteten Projekts entspricht. Diese Annahme ist in Gleichung 3 dargestellt.

$$\mu_{Änderungen,Projekt} = \mu_{Änderungen,Referenz} = \mu_{Änderungen}$$

Gleichung 3: Annahme zur Anzahl an durchschnittlichen Änderungen pro Anforderung

Mit der Annahme aus Gleichung 1 ergibt sich für die erwartete Anzahl an Änderungen durch exogene Einflüsse E_{exogen} des Projekts die folgende Gleichung 4:

$$E_{exogen} = 0,5 \cdot \mu_{\text{Änderungen}}$$

Gleichung 4: Erwartete Anzahl an Änderungen durch exogene Einflüsse

Das Vorgehen ist in Bild 6-19 dargestellt.

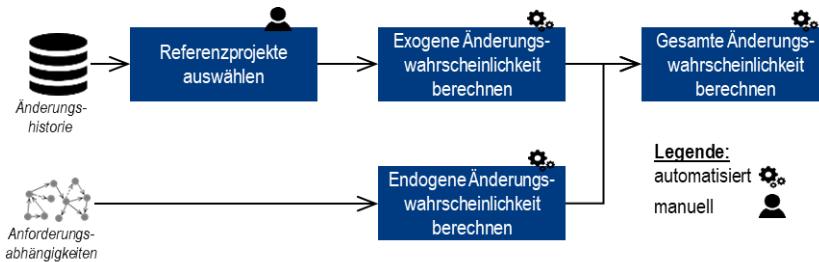


Bild 6-19: Berechnung der Änderungswahrscheinlichkeit einer Anforderung

Berechnung der endogenen Änderungswahrscheinlichkeit

Endogene Änderungen werden durch Propagationseffekte von Änderungen innerhalb des Anforderungsnetzes ausgelöst. Daher wird ein Maß benötigt, das auf der Konnektivität einer Anforderung in Bezug auf die Propagation von Änderungen basiert. Der PageRank ist solch ein Konnektivitätsmaß. Der PageRank-Algorithmus wurde bereits im Kontext von Anforderungsabhängigkeiten untersucht [GTO19]. Dieser Algorithmus ist geeignet, um effizient umfangreiche Anforderungsnetze zu analysieren. Der PageRank setzt sich aus verschiedenen Elementen zusammen (siehe Gleichung 5). d ist ein Dämpfungs faktor, um unendliche Schleifen in Anforderungsnetzen zu lösen. n ist die Anzahl an Anforderungen im Anforderungsnetz. r sind die betrachteten Anforderungen. v sind die Kanten gewichte bzw. die gewichteten Anforderungsabhängigkeiten. In der Methodik werden Anforderungsabhängigkeiten binär betrachtet, sodass diese Variable entweder 0 oder 1 beträgt. E^{in} bzw. E^{out} sind die Mengen der eingehenden bzw. ausgehenden Abhängigkeiten, die für die Berechnung des PageRanks betrachtet werden müssen.

$$PR(r_i) = \frac{1-d}{n} + d \sum_{r_j \in E_i^{in}} \left(\sum_{r_k \in E_j^{out}} \frac{PR(r_j)}{v_{kj}} \right)$$

d: Dämpfungsfaktor, *n*: Anzahl an Anforderungen, *r*: betrachtete Anforderung, *v*: Kantengewichte, E^{in} bzw. E^{out} : Mengen der eingehenden bzw. ausgehenden Abhängigkeiten

Gleichung 5: PageRank

Die gesamte Anzahl an endogenen Anforderungsänderungen entspricht aufgrund der Annahme in Gleichung 1 der Anzahl an exogenen Anforderungsänderungen. Das Ziel ist es,

durch einen Wert für die Konnektivität (PageRank) festzustellen, welche Anforderung mit einer hohen Anzahl an Anforderungen vernetzt ist. Umso höher der PageRank ist, umso höher ist die erwartete Anzahl an endogenen Änderungen. Wenn die Anforderung eine durchschnittliche Konnektivität im Anforderungsnetz besitzt (wenn $PR(r_i)$ gleich den Mittelwert des PageRanks μ_{PR} annimmt), dann entspricht der Erwartungswert der endogenen Änderungen dem Erwartungswert der exogenen Änderungen (siehe Gleichung 1). Für den Erwartungswert für eine Änderung durch endogene Ursachen $E_{endogen}$ für eine Anforderung r_i gilt daher Gleichung 6:

$$E_{endogen}(r_i) = 0,5 \cdot \mu_{\text{Änderungen}} \cdot \frac{PR(r_i)}{\mu_{PR}}$$

Gleichung 6: Erwartete Anzahl an Änderungen durch endogene Ursachen

Berechnung der gesamten Änderungswahrscheinlichkeit

Der gesamte Erwartungswert E_{gesamt} für eine Änderung durch exogene und endogene Ursachen für eine Anforderung ist die Summe der Erwartungswerte für Änderungen durch exogene oder endogene Ursachen (siehe Gleichung 7).

$$E_{gesamt}(r_i) = E_{exogen}(r_i) + E_{endogen}(r_i) = 0,5 \cdot \mu_{\text{Änderungen}} \cdot \left(1 + \frac{PR(r_i)}{\mu_{PR}}\right)$$

Gleichung 7: Gesamte Anzahl erwarteter Anforderungsänderungen

E_{gesamt} wird als Kennzahl für die Änderungswahrscheinlichkeit einer Anforderung verwendet. Die Interpretation der Kennzahl wird anhand eines Beispiels erläutert: wenn $E_{gesamt}(r_i) = 1,2$ beträgt, wird erwartet, dass sich die Anforderung r_i im Laufe des Projekts 1,2 Mal ändert. Um die Interpretation in der industriellen Praxis zu vereinfachen, werden die folgenden Kategorien definiert (siehe Tabelle 6-1).

Tabelle 6-1: Interpretation der Änderungswahrscheinlichkeit einer Anforderung in Anlehnung an [GOP22]

Erwartete Anzahl an Anforderungsänderungen	Änderungswahrscheinlichkeit
$E_{gesamt}(r_i) \leq 0,25$	Niedrig: es werden für diese Anforderung weniger als 0,25 Änderungen erwartet
$0,25 < E_{gesamt}(r_i) \leq 0,75$	Mittel: es werden für diese Anforderung zwischen 0,25 und 0,75 Änderungen erwartet
$0,75 < E_{gesamt}(r_i) \leq 1$	Hoch: es werden für diese Anforderung zwischen 0,75 und 1 Änderungen erwartet
$E_{gesamt}(r_i) > 1$	Sehr hoch: es werden für diese Anforderung mehr als eine Änderung erwartet

Für die Berechnung des Änderungsrisikos von Anforderungen werden neben der Änderungswahrscheinlichkeit auch die Auswirkungen von Anforderungsänderungen benötigt. Außerdem müssen auf Basis der Änderungswahrscheinlichkeit und der Auswirkungen das Gesamtrisiko berechnet werden und Handlungsempfehlungen (Auswahl von Risikosteuerungsmaßnahmen) für den Anwender gegeben werden. Die ProMaRC Methodik ermöglicht bereits eine Bewertung der Auswirkungen von Anforderungsänderungen, die Berechnung des Gesamtrisikos und die Unterstützung des Nutzers durch Handlungsempfehlungen und wurde von Industrievertretern im Rahmen des ARCA Projekts als anwendbar für die Entwicklung komplexer technischer Systeme bewertet. Aus diesem Grund wird für diese Schritte die ProMaRC Methodik angewendet.

Das Änderungsrisiko einer Anforderungsänderung besteht aus den Dimensionen Änderungswahrscheinlichkeit und Auswirkungen. Defizite in der ProMaRC-Methodik bestehen darin, dass Informationen hinsichtlich Änderungsinitiatoren in der Praxis häufig nicht verfügbar sind, weshalb die industrielle Anwendbarkeit eingeschränkt ist. Diese Informationen sind in der ProMaRC Methodik für die Berechnung der Änderungswahrscheinlichkeit zwingend notwendig. Aus diesem Grund wurde in der Methodik, die in dieser Dissertation entwickelt worden ist, die Berechnung der Änderungswahrscheinlichkeit weiterentwickelt.

Anwendungsfall der Methode

Analog zur Abhängigkeitsanalyse von Anforderungen ist ein Anwendungsfall der Berechnung der Änderungsrisiken das Risikomanagement von Anforderungsänderungen. Ziel des Risikomanagements ist es, die Handhabung von Anforderungsänderungen zu unterstützen, um Änderungskosten zu senken. Es soll frühzeitig in der Entwicklung das Risiko von Anforderungsänderungen beherrscht werden. Hierzu ist es notwendig, dass für jede Anforderung das Änderungsrisiko bekannt ist, um risikospezifisch Steuerungsmaßnahmen festzulegen. Die proaktive Auswahl von Risikosteuerungsmaßnahmen erfolgt in drei Schritten [GOP22]. Zunächst werden die Anforderungen auf der Basis eines Risikoportfolios priorisiert und ein Anforderungsprofil mit Risikoparametern erstellt. Anschließend werden jeder Anforderung eine Handlungsstrategie und zugehörige Steuerungsmaßnahmen zugeordnet. Daraus wählen die Anwender die bevorzugte Steuerungsmaßnahme aus. Die drei Schritte sind demnach die Folgenden.

- 1) Priorisierung der Anforderungen in einem Portfolio
- 2) Zuordnung von Handlungsstrategien und Steuerungsmaßnahmen
- 3) Auswahl von Risikosteuerungsmaßnahmen

Für die Priorisierung der Anforderungen (*Schritt 1*) wird ein Risikoportfolio mit den Dimensionen Änderungswahrscheinlichkeit und Änderungsauswirkung durch Propagation verwendet (siehe Bild 6-20).

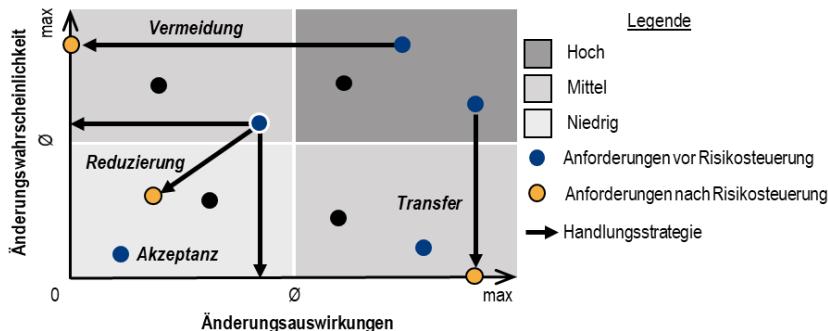


Bild 6-20: Risikoportfolio mit exemplarischen Anforderungen [GPO21]

Um die Anforderungen zu priorisieren, wird für jede Portfoliodimension ein Schwellwert definiert. Auf der Grundlage der Schwellwerte wird das Portfolio in vier Quadranten eingeteilt, die einer Risikokategorie zugewiesen werden (hoch, mittel und niedrig). Die Anforderungen werden kategorisiert und entsprechend dem zugehörigen Quadranten priorisiert. Die Schwellwerte können abhängig von der Risikoaffinität angepasst werden. Als Standard werden Durchschnittswerte der gewichteten Änderungsrisiken der Anforderungen im Projekt verwendet. Auf der Grundlage des Risikoportfolios wird für jede Anforderung eine Handlungsstrategie festgelegt (*Schritt 2*). Es werden vier Handlungsstrategien unterschieden: Vermeidung, Übertragung, Reduzierung und Akzeptanz [SM02].

- **Vermeidung:** Eliminierung der Eintrittswahrscheinlichkeit (z. B. Ablehnung eines Auftrags oder von Änderungswünschen [SHB19]).
- **Verringerung:** Minderung der Wahrscheinlichkeit des Auftretens oder der Auswirkungen (z. B. die Vereinbarung von Aufwandsbeschränkungen [Die13])
- **Übertragung:** Teilen oder Übertragen der Änderungsauswirkungen (z. B. Versicherung oder vertragliche Weitergabe der Änderungskosten an den Kunden [Ger11]).
- **Akzeptanz:** Keine Aktion und keine Auswirkung.

Die Auswahl geeigneter Maßnahmen wird durch einen Empfehlungsmechanismus unterstützt (*Schritt 3*). Es werden alle Maßnahmen hervorgehoben, die der Risikokategorie der Anforderungen zugeordnet sind. Auf Basis der vorgeschlagenen Maßnahmen erfolgt eine Auswahl durch den Anwender.

6.3 Software-Werkzeug

Es wird ein Software-Werkzeug entwickelt, um die Anwendung der Methoden zur Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen zu unterstützen. Die Methoden werden in Python implementiert. Die graphische Benutzeroberfläche (GUI) wird genutzt, um die Anwendung der Methoden zu demonstrieren. Die 6 Schritte zur Anwendung der Methoden im Software-Werkzeug sind in Bild 6-21 dargestellt.

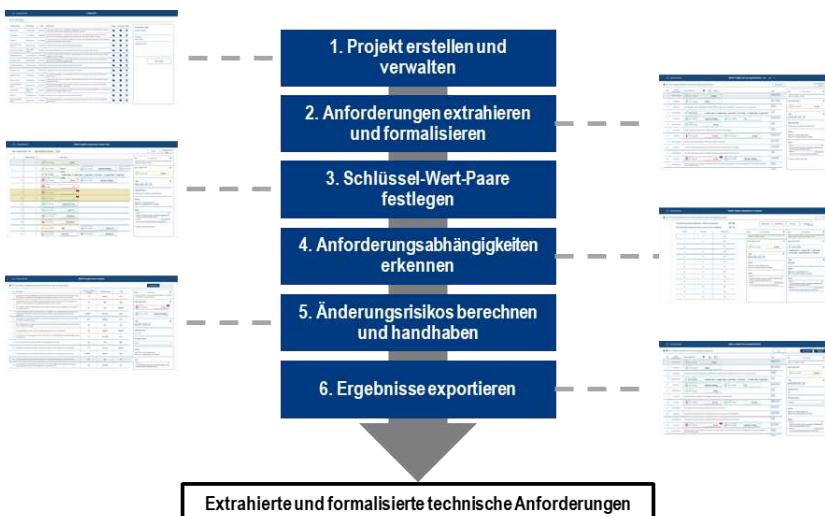


Bild 6-21: Übersicht der sechs Schritte im Software-Werkzeug der Methodik

In Schritt 1 muss der Anwender im Software-Werkzeug ein Projekt erstellen (siehe Bild 6-22). Nach dem Öffnen des Software-Werkzeugs wird dem Anwender eine tabellarische Ansicht der bisher erstellten Projekte („Projektverwaltung“) angezeigt. Rechts in der Projektverwaltung wird ein rechteckiges Element zur Erstellung von Projekten angezeigt. Der Anwender hat die Möglichkeit, Attribute von Projekten („Name des Projekts“, „Erstellt von“ und „Beschreibung“) festzulegen. Durch das Betätigen des Buttons „Neues Projekt“ wird ein neues Projekt mit den vorher definierten Attributen angelegt. In der tabellarischen Ansicht (links) können bestehende Projekte geöffnet, archiviert oder gelöscht werden. In dieser Tabelle werden für die jeweiligen Projekte die o. g. Attribute von Projekten angezeigt. Durch das Archivieren wird das Projekt in der tabellarischen Liste von Projekten für den Anwender ausgeblendet.

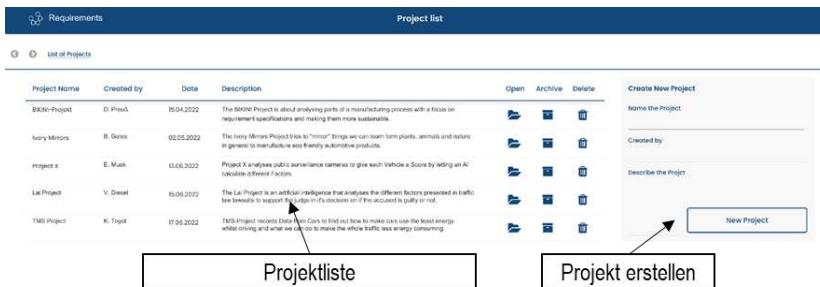


Bild 6-22: Projekt erstellen und verwalten

Nach dem Erstellen eines Projektes muss ein Lastenheft importiert werden (Schritt 2). Nach dem Import des Lastenhefts startet die automatisierte Extrahierung, Klassifizierung und Formalisierung von Anforderungen. Anschließend öffnet sich eine Ansicht mit den extrahierten und formalisierten Anforderungen (siehe Bild 6-23). Diese Ansicht ist die Hauptansicht des Software-Werkzeug. Es werden Buttons angezeigt, um die verschiedenen Funktionen des Werkzeugs auszuführen und die zugehörigen Ansichten anzuzeigen. Dazu zählen: „Schlüssel-Wert-Paare (KVP) festlegen“, „Anforderungsabhängigkeiten erkennen“, „Risikomanagement von Anforderungsänderungen“ sowie „CSV-Export“. Die Anforderungen werden in dieser Ansicht links in einer Tabelle dargestellt. Dargestellte Attribute der Anforderungen in der Tabelle sind: „ID“, „Anforderungssart“ (funktional oder nicht-funktional), „Anforderungsbeschreibung“ oder „Schlüssel-Wert-Paare“ und „Tags“. Der Anwender hat die Möglichkeit, entweder die extrahierten Schlüssel-Wert-Paare der Anforderungen oder die textuelle Beschreibung der Anforderungen anzeigen zu lassen. Oberhalb der Tabelle befindet sich ein Schalter, um zwischen der Anzeige der textuellen Beschreibung oder den extrahierten Schlüssel-Wert-Paaren der Anforderungen umzuschalten. Die Tags sind Stichworte zu den Anforderungen, welcher der Anwender manuell anlegen kann, um Anforderungen beispielsweise zu gruppieren. Wenn eine Anforderung vom Anwender markiert worden ist, dann öffnet sich auf der rechten Seite in der Ansicht eine Detailansicht der markierten Anforderung. In dieser Detailansicht werden zusätzlich zu den oben beschriebenen Informationen noch erkannte Anforderungsabhängigkeiten angezeigt. Anforderungsabhängigkeiten werden durch die ID der abhängigen Anforderung dargestellt. Die Anforderungsabhängigkeiten werden erst dargestellt, wenn die Abhängigkeitsanalyse von Anforderungen ausgeführt worden ist. Außerdem hat der Anwender in der Detailansicht die Möglichkeit, den Wert von erkannten Schlüssel-Wert-Paaren zu editieren.

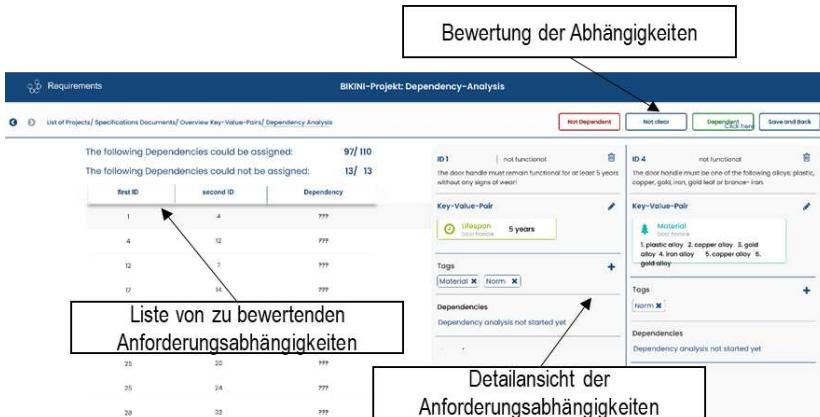


Bild 6-23: Ansicht der extrahierten und formalisierten Anforderungen

In Schritt 3 müssen die Schlüssel-Wert-Paare geprüft und festgelegt werden (siehe Bild 6-24). Hierzu muss der Anwender den Button „Schlüssel-Wert-Paare festlegen“ betätigen. Als Anwendungsfall der Extrahierung und Formalisierung wird die generative Entwicklung eines Bauteils betrachtet. Der Anwender hat in dieser Ansicht die Möglichkeit, sämtliche Schlüssel-Wert-Paare für die generative Entwicklung des Bauteils festzulegen. Die Liste der Schlüssel-Wert-Paare wird initial automatisiert durch die erkannten Schlüssel-Wert-Paare gefüllt. Die Methode zur Formalisierung von Anforderungen wurde bereits in Schritt 2 durchgeführt. Der Anwender muss diese Liste prüfen und falsch erkannte Schlüssel-Wert-Paare editieren. In der Spalte „Schlüssel-Wert-Paare“ werden sämtliche Schlüssel-Wert-Paare angezeigt, die für die generative Entwicklung relevant sind. Für die einzelnen Schlüssel kann der Anwender über ein Drop-Down-Menü verschiedene Werte auswählen. Auf der rechten Seite in dieser Ansicht wird, sofern das Schlüssel-Wert-Paar einer Anforderung zugewiesen werden konnte, die Detailansicht der zugehörigen Anforderung des markierten Schlüssel-Wert-Paares dargestellt. Falls durch den Algorithmus keine Anforderung mit dem markierten Schlüssel-Wert-Paar erkannt werden konnte, dann ist die Detailansicht leer. Falls für ein Schlüssel-Wert-Paar, welches für die generative Entwicklung relevant ist, aktuell kein Wert festgelegt worden ist, dann wird die Zeile der betroffenen Schlüssel-Wert-Paare farbig hinterlegt. Außerdem wird dem Anwender oberhalb der Tabelle die Anzahl der Schlüssel ohne Wert angezeigt.

Bild 6-24: *Schlüssel-Wert-Paare festlegen*

Im nächsten Schritt müssen Anforderungsabhängigkeiten erkannt werden (*Schritt 4*). Durch das Klicken des Buttons „Abhängigkeitsanalyse starten“ in der Hauptansicht des Software-Werkzeugs beginnt die Abhängigkeitsanalyse der Anforderungen. Der Anwender muss zunächst einen Datensatz mit Trainingsdaten für das BERT-Modell im Explorer auswählen. Das BERT-Modell wird initial mit den Trainingsdaten trainiert. Anschließend beginnt das Aktive Lernen und dem Anwender werden verschiedene Anforderungsabhängigkeiten präsentiert, welche er manuell als „abhängig“ oder „nicht-abhängig“ kennzeichnen muss (siehe Bild 6-25).

Bild 6-25: *Abhängigkeitsanalyse von Anforderungen*

Links werden in dieser Ansicht die Attribute „ID“ und „Abhängigkeitsart“ des zu kennzeichnenden Anforderungspaares tabellarisch dargestellt. Wenn der Anwender eine Zeile in der Tabelle markiert, dann werden rechts die Detailansichten des Anforderungspaares dargestellt. Oberhalb der Tabelle werden dem Anwender angezeigt, wie viele Anforderungsabhängigkeiten gekennzeichnet werden müssen. Oberhalb der Detailansicht des Anforderungspaares werden dem Nutzer Buttons („abhängig“ und „nicht-abhängig“) angezeigt, um die Abhängigkeit des Anforderungspaares zu bewerten. Außerdem

hat der Anwender die Möglichkeit, das aktive Lernen manuell zu beenden (Button: „Speichern und zurück“). Nachdem der Anwender die Anforderungsabhängigkeiten bewertet hat, wird das BERT-Modell mit den neu gekennzeichneten Daten aktualisiert. Das Aktive Lernen stoppt, wenn ein Abbruchkriterium erfüllt ist, oder wenn der Anwender das aktive Lernen manuell beendet. Anschließend werden die verbleibenden (bzw. nicht durch den Anwender gekennzeichneten) Anforderungsabhängigkeiten durch das BERT-Modell automatisiert klassifiziert.

Nachdem Anforderungsabhängigkeiten erkannt worden sind, kann das Risikomanagement von Anforderungsänderungen gestartet werden (*Schritt 5*). Hierzu muss der Button „Risikomanagement von Anforderungsänderungen“ in der Hauptansicht betätigt werden. Zu Beginn muss die Änderungshistorie von Anforderungen eingelesen werden. Anschließend wird die Änderungswahrscheinlichkeit bzw. der Erwartungswert der Anzahl an Änderungen, automatisiert berechnet. Dem Anwender wird die Ansicht des Risikomanagements von Anforderungsänderungen angezeigt (siehe Bild 6-26).

ID	Description	Expected number of changes	Change Impact	Risk
1	Any deviation from the specification or standard requirements not described in this document must be indicated immediately by the supplier and requires the written consent of the customer.	0.1	Medium	Low
2	Any content or point not described or poorly described must be documented by the supplier and requires a written agreement and subsequent approval of the customer.	0.5	High	High
3	If the supplier is aware of alternatives to better fulfill the quality, cost and reliability, he must point it out to the customer.	0.3	Low	Medium
4	In order to achieve optimum customer benefit, the supplier must critically deal with proposals and specifications taking part of the customer and, if necessary, elaborate improved solutions in cooperation.	0.00956	Very Low	High
7	Fastening elements, wrench size and tightening torques are to be standardized.	0.6	Medium	Medium
8	For all screws of a mounting group, screws with uniform screw head must be used.			
9	For the same thread diameter, a uniform screw length must be used.			
10	If possible, the fastening elements must be pre-mounted on the components to be fastened.	0.3	Very Low	Medium

Detailansicht zur Bewertung der Änderungsauswirkung

Change Impact: Low

Bild 6-26: Risikomanagement von Anforderungsänderungen

Links sind die Anforderungen des Projekts tabellarisch dargestellt. Attribute sind: „ID“, „textuelle Beschreibung“, „Erwartungswert der Anzahl an Änderungen“, „Änderungsauswirkungen“ und „Änderungsrisiko“. Die Änderungsauswirkungen und das Änderungsrisiko sind initial nicht bewertet. Der Anwender hat die Möglichkeit, einzelne Anforderungen in der Tabelle links in der Ansicht des Risikomanagements von Anforderungsänderungen zu markieren. Die Detailansicht der markierten Anforderung wird rechts in der Ansicht geöffnet. Dadurch werden dem Anwender Anforderungsabhängigkeiten angezeigt. Durch die Kenntnis über die Abhängigkeiten der Anforderungen werden vom Anwender die Änderungsauswirkungen bestimmt. Der Anwender kann bei der Bewertung der Auswirkungen beispielsweise durch Leitfragen (siehe Anhang A2.5) unterstützt werden. Das gesamte Änderungsrisiko der Anforderung berechnet sich

automatisiert aus dem Erwartungswert der Anzahl an Änderungen und den Änderungsauswirkungen. Im letzten Schritt werden die extrahierten und formalisierten Anforderungen sowie deren Abhängigkeiten im CSV-Format exportiert (*Schritt 6*). Hierzu wird in der Hauptansicht der Button „CSV-Export“ vom Anwender betätigt. Die exportierten Daten können in ein Anforderungsmanagement-Werkzeug eingelesen werden. Die CSV-Datei wurde so formatiert, dass ein Import in das Anforderungsmanagement-Werkzeug IBM DOORS möglich ist.

6.4 Abgrenzung zu Vorarbeiten des Lehrstuhls für Produktentstehung

In diesem Abschnitt wird die Methodik dieses Dissertationsmanuskript von Vorarbeiten des Lehrstuhls für Produktentstehung abgegrenzt, um den Neuheitsgrad hervorzuheben. Diese Abgrenzung ist ergänzend zu Abschnitt 5.1 und wird nachgestellt, weil detaillierte Kenntnisse zur entwickelten Methodik notwendig sind, die in den Abschnitten 6.1 bis 6.3 beschrieben werden.

Abgrenzung zur Dissertation „Proaktives Management von Anforderungsänderungen in der Entwicklung komplexer technischer Systeme“ (kurz: Diss. Oleff)

Die in Diss. Oleff entwickelte ProMaRC-Methodik [Ole22] basiert auf gemeinsamen Vorarbeiten [GOH22b, GOP21, GOP22, GPO20, GPO21]. In diesen Vorarbeiten wurden Validierungen im Kontext des ARCA Projekts durchgeführt und dargestellt. In der vorliegenden Arbeit werden die Validierungsergebnisse analysiert, um Erkenntnisse zu den Forschungsbereichen Abhängigkeitsanalyse von Anforderungen und Risikomanagement von Anforderungsänderungen zu gewinnen. Die für diese Arbeit relevanten Erkenntnisse, die auf Validierungsergebnissen aus ARCA aufbauen, werden in Kapitel 4 beschrieben.

Kern-Erkenntnis für die *Abhängigkeitsanalyse von Anforderungen* mittels BERT-Modell ist, dass vortrainierte Modelle nur begrenzt für die Klassifizierung heterogener Daten angewendet werden können. Dies hat den Bedarf, der von der vorliegenden Dissertation aufgenommen worden ist, nach einer Überarbeitung der Abhängigkeitsanalyse mit BERT offen gelegt. Die Entwicklung der Methode und die Validierungsergebnisse der Abhängigkeitsanalyse mittels BERT werden in Diss. Oleff beschrieben. In der vorliegenden Arbeit werden die Validierungsergebnisse zu diesem Thema auszugsweise vorgestellt (siehe Tabelle 4-4) und analysiert, um den Forschungsbedarf aufzudecken und Handlungsmaßnahmen abzuleiten. Der Forschungsbedarf wird durch die in dieser Arbeit vollständig entwickelten Methode zur Abhängigkeitsanalyse mittels BERT und aktivem Lernen adressiert (siehe Abschnitt 6.2.3). Diese Methode ist im Software-Werkzeug implementiert (siehe Abschnitt 6.3).

Die Kern-Erkenntnis für das *Risikomanagement von Anforderungsänderungen* ist, dass die *Risikoberechnung* überarbeitet werden muss, um die Anwendbarkeit in der industriellen Praxis zu erhöhen. Der Grund dafür ist, dass Informationen zu Änderungsinitiatoren üblicherweise nicht in der Praxis verfügbar sind. In dieser Arbeit werden die Validierungsergebnisse zu diesem Forschungsbereich auszugsweise vorgestellt (siehe Tabelle 4-5) und analysiert, um den Forschungsbedarf aufzudecken. Die Risikoberechnung von Anforderungsänderungen, die gemeinsam erarbeitet und in ARCA validiert worden ist, wird in der vorliegenden Arbeit erweitert, sodass Informationen zu Änderungsinitiatoren nicht länger benötigt werden (siehe Abschnitt 6.2.4). Das Risikomanagement von Anforderungsänderungen, das auf dieser Berechnung aufbaut, wird in dieser Arbeit nur als Anwendungsperspektive für diese Risikoberechnung betrachtet (siehe Abschnitt 6.2.4). Im Software-Werkzeug dieser Arbeit ist die überarbeitete Risikoberechnung implementiert. Das im Software-Werkzeug implementierte Risikomanagement stimmt konzeptuell mit dem gemeinsam entwickelten Risikomanagement überein. Die Abgrenzung der Arbeiten wird in Bild 6-27 visualisiert.

Legende:

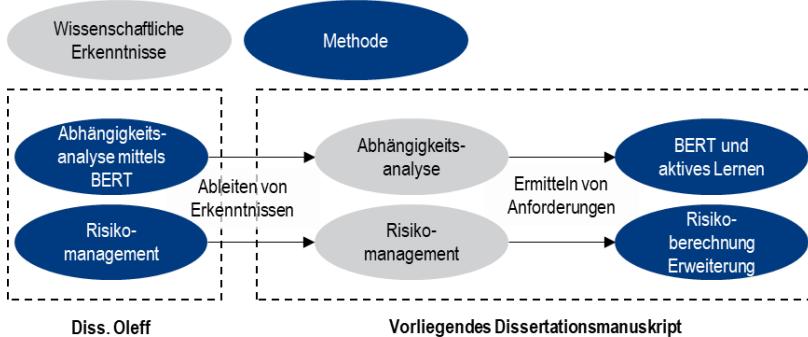


Bild 6-27: Abgrenzung der Arbeiten

Abgrenzung zur Methode „Model-based Effect Chain Analysis“ (MECA)

Die MECA-Methode wurde von GRÄBLER, WIECHEL, KOCH, PREUB und OLEFF veröffentlicht [GWK22] und im Rahmen des ZWF-Projekts validiert. In dem vorliegenden Dissertationsmanuskript wird die Entwicklung der Methode nicht beschrieben. Es wird lediglich beschrieben, dass ein maßgeblicher Teil der Informationsgrundlage, die zur Anwendung der MECA-Methode benötigt wird, effizient durch die Methodik dieser Arbeit erstellt werden kann. Anforderungen können mit der Methodik automatisiert aus Lastenheften extrahiert und in modellbasierte Anforderungselemente transformiert werden. Somit wird in dieser Arbeit der Anknüpfungspunkt zwischen MECA-Methode und dieser Methodik dargestellt. Startpunkt zur Wirkkettenmodellierung sind extrahierte Anforderungen.

7 Validierung der Methodik

Die entwickelte Methodik wird im Rahmen der Projekte ARCA, BIKINI und ZWM validiert (siehe Projektbeschreibungen in Anhang A2). Das Validierungskonzept innerhalb der Projekte wurde in diesem Promotionsvorhaben erarbeitet. Die Validierung ist unterteilt in die Validierung der einzelnen Methoden sowie die Erfolgsvalidierung der entwickelten Methodik. Zur Validierung der einzelnen Methoden werden verschiedene Fallbeispiele betrachtet. Es werden Kennzahlen berechnet, um die Leistungsfähigkeit der Methoden zu messen. Zum Beispiel werden Klassifizierungs-Kennzahlen, wie Präzision oder Sensitivität berechnet, um die Leistungsfähigkeit der trainierten KI-Modelle zu bewerten. Zur Validierung der gesamten Methodik wird die Methodik von Unternehmensvertretern bewertet. Die Unternehmensvertreter füllen einen Bewertungsbogen aus, um den Erfolg der Anwendung der Methodik zu messen. Außerdem wird ein Interview mit den Unternehmensvertretern durchgeführt, um die Ergebnisse der Auswertung der Bewertungsbögen zu diskutieren. Abschließend wird anhand der Validierungsergebnisse die Erfüllung der Erfolgsfaktoren (siehe Abschnitt 5.2) bewertet. Die Produkte, die zur Validierung als Fallbeispiele genutzt werden, sind in Bild 7-1 visualisiert. Aufgrund von Geheimhaltungsvereinbarungen werden keine Originalbilder, sondern Bilder ähnlicher Produkte verschiedener Hersteller gezeigt.



*Bild 7-1: Produkte, die als Fallbeispiele zur Validierung der Methodik genutzt werden
[For23, MAC23, Maz23]*

7.1 Extrahierung und Klassifizierung von Anforderungen

Die *Extrahierung und Klassifizierung von Anforderungen* wird im Rahmen des Projekts BIKINI validiert. Durch die entwickelte Methode werden Textsegmente aus Lastenheften des Entwicklungsdienstleisters EDAG extrahiert und hinsichtlich Anforderungsart klassifiziert. Die Leistungsfähigkeit der Methode wird mittels Klassifizierungs-Kennzahlen bewertet und die Ergebnisse werden diskutiert.

Für die Methode zur *Extrahierung und Klassifizierung von Anforderungen* werden Klassifizierungsmodelle mit Anforderungsdaten trainiert. Um die Anwendbarkeit der Methode in der Praxis zu erhöhen, sollen augmentierte Anforderungsdaten für das Training der Modelle genutzt werden. Es wird eine Vorstudie durchgeführt, um zu prüfen, wie hoch die Leistungsfähigkeit der Augmentation von Anforderungsdaten ist.

7.1.1 Vorstudie: Augmentation von Anforderungsdaten mittels GPT-J

In dieser Vorstudie wird untersucht, wie hoch die Leistungsfähigkeit der Augmentation von Anforderungsdaten durch künstliche Intelligenz ist (FF 1, siehe Abschnitt 5.1). Es soll überprüft werden, ob künstliche Anforderungsdaten zuverlässig generiert werden können, sodass Aufwand in der Generierung von Anforderungsdaten reduziert werden kann. Für die Studie wird GPT-J zur Daten-Augmentation verwendet (siehe Abschnitt 3.1). Zu diesem Zweck wird ein Datensatz aus "funktionalen Anforderungen", "nicht-funktionalen Anforderungen" und "keine Anforderungen" benötigt. Zur Generierung des Datensatzes werden Textsegmente in Lastenheften von Maschinenbaustudierenden im Rahmen der Master-Vorlesung „Systems Engineering“ des Lehrstuhls für Produktentstehung an der Universität Paderborn gekennzeichnet. Insgesamt kennzeichnen 35 Studierende 5609 Textsegmente (955 funktionale, 1019 nicht-funktionale und 3380 keine Anforderungen) aus sechs Open-Source-Anforderungsspezifikationen von Softwareprojekten (z. B. "UK Fishing Vessel's Electronic Logbook" [AK09, Are08, Ath09, Pat10, Sum09, Vah20]) hinsichtlich „funktionale“, „nicht-funktionale“ und „keine Anforderung“. Die Anforderungsdaten der gekennzeichneten Open-Source Lastenhefte sind in Anhang A5, Bild A-98 auszugsweise dargestellt.

Für die Generierung der künstlichen Anforderungsdaten wird das GPT-J-Modell verwendet. Es werden insgesamt 193 funktionale, 226 nicht-funktionale und 112 "keine Anforderungen" generiert. Die künstlich generierten Anforderungsdaten werden in Anhang A5, Bild A-99 auszugsweise dargestellt. Für jede Anforderung wird manuell bewertet, zu welcher Klasse die künstlich erzeugte Anforderung gehört. Zur Messung der Leistungsfähigkeit der Augmentierung von Anforderungsdaten werden die Klassifikationskennzahlen *Präzision*, *Sensitivität* und *F1* verwendet. Um diese Klassifikationskennzahlen zu ermitteln, müssen *wahre positive* (t_p), *wahre negative* (t_n), *falsch positive* (f_p) und *falsch negative* (f_n) Ergebnisse berechnet werden. Im Kontext der Augmentierung

von Anforderungsdaten (unter Betrachtung der binären Klassifizierung von "Anforderung" und "keine Anforderung") sind diese Ergebnisse folgendermaßen zu interpretieren.

- tp : aus einem Textsegment der Klasse "Anforderung" wird ein Textsegment der Klasse "Anforderung" erzeugt.
- tn : aus einem Textsegment der Klasse "keine Anforderung" wird ein Textsegment der Klasse "keine Anforderung" erzeugt.
- fp : aus einem Textsegment der Klasse "keine Anforderung" wird fälschlicherweise ein Textsegment der Klasse "Anforderung" generiert.
- fn : aus einem Textsegment der Klasse "Anforderung" wird fälschlicherweise ein Textsegment der Klasse "keine Anforderung" generiert.

Die Klassifizierungskennzahlen zur Messung der Leistungsfähigkeit der Augmentierung von Anforderungsdaten durch das GPT-J-Modell werden in Tabelle 7-1 dargestellt.

Tabelle 7-1: Ergebnisse der Augmentierung von Anforderungen durch GPT-J [GPB22]

Klasse	Anzahl	Präzision	Sensitivität	F1
funktionale Anforderung	193	93,90 %	39,90 %	56,00 %
nicht-funktionale Anforderung	226	84,68 %	41,59 %	55,78 %
Keine Anforderung	112	32,54 %	98,21 %	45,90 %
<i>Makro-Durchschnitt</i>	-	70,37 %	59,90 %	64,71 %
<i>Gewichteter Durchschnitt</i>	-	77,03 %	52,92 %	62,74 %

Die F1 Kennzahlen der einzelnen Klassen liegen zwischen 45,90 % und 56,00 %. Der Makro-Durchschnitt von F1 sowie der gewichtete Durchschnitt von F1 liegen bei 64,71 % bzw. 62,74 %. Demnach entspricht ein großer Teil der künstlich generierten Anforderungen nicht der Klasse der Anforderungen, die zur Generierung der künstlichen Anforderungen genutzt worden sind. Die Leistungsfähigkeit wird weiter untersucht, indem eine binäre Klassifizierung (Klassen "Anforderung" oder "keine Anforderung") betrachtet wird. Es wird untersucht, ob die Generierung verschiedener Anforderungsarten oder die Generierung von Anforderungen („Anforderung“ oder „keine Anforderung“) fehlerbehaftet ist. Die Ergebnisse sind in Tabelle 7-2 dargestellt.

Tabelle 7-2: Ergebnisse der Augmentierung von Anforderungen (binär) [GPB22]

Kennzahlen	Werte
$tp/tn/f_p/f_n$	191/110/2/228
Präzision	98,96 %
Sensitivität	45,59 %
F1	62,42 %

Die Zahl der falsch negativen Ergebnisse ist hoch (228). Das bedeutet, dass häufig aus einem Text der Klasse "Anforderung" ein künstlicher Text der Klasse "keine Anforderung" erzeugt wurde. Dies zeigt auch die niedrige Sensitivität (45,59 %). Ein Text der Klasse "keine Anforderung" wird zu einem hohen Anteil korrekt aus einem Text der Klasse "keine Anforderung" künstlich erzeugt (Präzision: 98,96 %).

Die Ergebnisse zeigen, dass GPT-J zur Erzeugung künstlicher Anforderungen verwendet werden kann. Allerdings muss aufgrund der Leistungsfähigkeit (F1: 62,42 %) eine Qualitätskontrolle der künstlichen Anforderungen durchgeführt werden, um bei der Be- trachtung von Daten der Klasse "Anforderung" korrekte Kennzeichnungen zu vergeben.

7.1.2 Extrahierung von Anforderungen mittels Daten-Augmentierung

Die künstlichen Anforderungen sollen für das Training von Modellen zur Klassifizierung von Anforderungen genutzt werden. Dadurch wird dem Problem der unzureichenden Verfügbarkeit von Daten in der industriellen Praxis entgegengewirkt. Es wird ermöglicht, effizient eine Datengrundlage zu generieren, wodurch die industrielle Anwendbarkeit erhöht wird. Es muss untersucht werden, wie hoch die Leistungsfähigkeit bei der Klassifizierung von Anforderungen ist, wenn Modelle mit augmentierten Anforderungsdaten trainiert worden sind, im Vergleich zu Modellen, die mit realen Anforderungsdaten trainiert worden sind (FF 2). Dies erlaubt eine Aussage darüber, ob augmentierte Daten für das Training von Klassifizierungsmodellen genutzt werden können, sodass mit diesen Modellen reale Anforderungsdaten klassifiziert werden können. Es werden mehrere Modelle trainiert und getestet. Anhand der Ergebnisse wird die Leistungsfähigkeit der Methode bewertet.

Generierung von Anforderungsdaten

Um Anforderungsdaten für die Validierung zu generieren, werden Textsegmente aus drei technischen Spezifikationen von EDAG gekennzeichnet. Diese Produkte sind: *Montagerriegelhaube (A)*, *verstellbare Stopperhaube (B)* und *Kofferraumrollo (C)* (siehe Tabelle 7-3). Die Textsegmente werden mit der in Abschnitt 6.2.1 beschriebenen Methode aus den Spezifikationen extrahiert und anschließend klassifiziert. Insgesamt werden 793 Textsegmente extrahiert. Diese Textsegmente werden von einem Entwicklungsingenieur von EDAG gekennzeichnet. Zusätzlich werden die mit Hilfe von GPT-J (siehe Vorstudie

in Abschnitt 7.1.1) aus Anforderungen von Open-Source-Softwareentwicklungsprojekten generierten augmentierten Anforderungsdaten (D) verwendet. Es werden drei Untersuchungen durchgeführt, um die Leistungsfähigkeit der Methode zur Extrahierung und Klassifizierung von Anforderungen zu bewerten. Es wird der Einfluss der augmentierten Daten auf die Leistungsfähigkeit der Klassifizierung für drei verschiedene Modelle des Maschinellen Lernens – Stützvektormaschine (englisch: „Support Vector Machine“, SVM), Zufallswald (englisch: „Random Forest“, RF) und Neuronales Netz (NN) gemessen. Die Anforderungsdaten werden in Anhang A5 auszugsweise dargestellt. Diese Daten werden für verschiedene Untersuchungen genutzt.

Tabelle 7-3: Anforderungsdaten zur Validierung der Extrahierung und Klassifizierung von Anforderungen mit augmentierten Anforderungen [GPB22]

Spezifikation	Funktionale Anforderungen	Nicht-funktionale Anforderungen	Keine Anforderungen
Montage Riegel Haube (A)	163	95	87
Haubenanpassung Stöpsel (B)	116	36	85
Kofferraum rollo (C)	90	51	70
Augmentierte Daten (D)	193	226	112
<i>Gesamte Anforderungen</i>	<i>562</i>	<i>408</i>	<i>354</i>

Untersuchung 1: Reale Anforderungsdaten von EDAG

In einer ersten Untersuchung wird die Leistungsfähigkeit bei der Verwendung von realen Anforderungsdaten aus EDAG-Projekten (siehe Tabelle 7-3, Montageriegelhaube (A), verstellbare Stopperhaube (B) und Kofferraumrollo (C)) bewertet. Die Leistungsfähigkeit wird mittels zehnfacher Kreuzvalidierung (englisch: „ten-fold cross validation“) [Alp22] bewertet, um repräsentative Ergebnisse zu erhalten. Bei der zehnfachen Kreuzvalidierung wird der Datensatz zufällig in zehn Teilmengen aufgeteilt. Eine Teilmenge wird als Testdaten beibehalten, die anderen neun Teilmengen werden als Trainingsdaten verwendet. Der Vorgang wird zehnmal wiederholt, wobei jede der zehn Teilmengen einmal als Testdaten verwendet werden. Schließlich werden die Ergebnisse der Untersuchung der verschiedenen Teilmengen gemittelt. Bei der SVM und dem RF werden die Daten in 90 % Trainingsdaten und 10 % Testdaten aufgeteilt. Für das Neuronale Netz (NN) werden die Daten in 70 % Trainingsdaten, 20 % Validierungsdaten und 10 % Testdaten aufgeteilt. Es werden die Anforderungsdaten der einzelnen Projekte (A, B und C) und als Kombination der Projekte (A+B+C) verwendet. Es wird der gewichtete durchschnittliche F1 der Modelle berechnet. Die Ergebnisse sind in Bild 7-2 dargestellt.

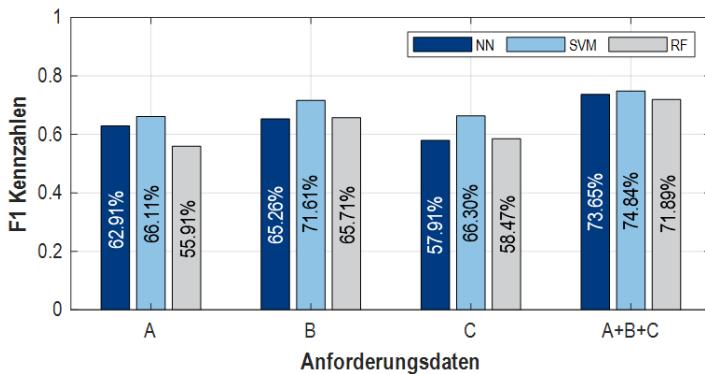


Bild 7-2: Ergebnisse für die Klassifizierung realer Anforderungsdaten (NN: Neuronales Netz, SVM: Stützvektormaschine und RF: Zufallswald) [GPB22]

Der F1 der Modelle für die kombinierten Anforderungsdaten (A+B+C) ist höher als für die einzelnen Anforderungsdaten (A, B und C). Das Training mit einer größeren domänen spezifischen Datenbank erhöht die Leistungsfähigkeit in dieser Untersuchung. Die Leistungsfähigkeit der SVM ist bei allen Tests höher als die der anderen Modelle. Im Durchschnitt ist der F1 der SVM um 6,97 % höher als der des Modells mit dem niedrigsten F1, wenn alle Tests betrachtet werden. Bei der Untersuchung mit den Anforderungsdaten A+B+C sind die Leistungsunterschiede zwischen dem besten und dem schlechtesten Modell am geringsten (F1 Differenz von 2,95 %). Das Modell mit der niedrigsten durchschnittlichen Leistungsfähigkeit ist RF. Der durchschnittliche F1 für alle Tests beträgt 62,74 %.

Untersuchung 2: Reale Anforderungsdaten von EDAG und augmentierte Anforderungsdaten

In dieser Untersuchung wird die Leistungsfähigkeit der Modelle bei Verwendung der realen Anforderungsdaten von EDAG und der augmentierten Daten (siehe Tabelle 7-3, Montageriegelhaube (A), verstellbare Stopperhaube (B), Kofferraumrollo (C) und augmentierte Daten (D)) untersucht. Die Aufteilung der Anforderungsdaten ist dieselbe wie in Untersuchung 1. Die augmentierten Daten werden für diese Untersuchung im Vergleich zu Untersuchung 1 zu den Anforderungsdaten vollständig hinzugefügt. Die Ergebnisse sind in Bild 7-3 dargestellt.

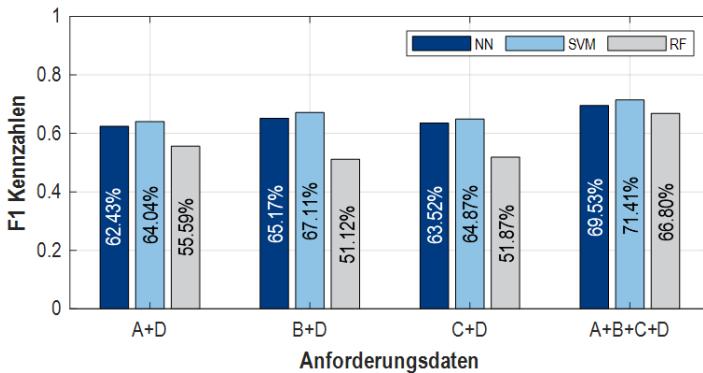


Bild 7-3: Ergebnisse für reale Anforderungsdaten und augmentierte Daten (NN: Neuronales Netz, SVM: Stützvektormaschine und RF: Zufallswald) [GPB22]

Die Leistungsfähigkeit der Modelle ist strikt niedriger als in Untersuchung 1 (außer in Gruppe C+D von Modell NN). Im Durchschnitt sinken die F1 der Modelle um 3,1 %. Obwohl die Anzahl der Daten jeweils zunimmt, nimmt die Leistungsfähigkeit ab. Die Anforderungsdaten stammen nicht mehr ausschließlich aus dem Automobilbereich (Daten von EDAG), sondern auch aus der Softwareentwicklung (augmentierte Anforderungen D). Aufgrund der Daten aus unterschiedlichen Domänen sinkt die Leistungsfähigkeit der Modelle. Auch in diesem Test hat die SVM in jeder Untersuchung die höchste Leistungsfähigkeit. Im Durchschnitt ist die SVM um 10,60 % besser als das schlechteste Modell, wenn alle Untersuchungen berücksichtigt werden. Auch hier ist der Leistungsunterschied der Modelle in der Untersuchung, in der die gesamten Daten kombiniert werden (A+B+C+D), am geringsten. Die Differenz der F1 des besten und des schlechtesten Modells beträgt 4,61 %. Insbesondere die Leistungsfähigkeit des RF hat sich im Vergleich zu Untersuchung 1 verschlechtert: Der durchschnittliche F1 sinkt von 62,74 % auf 56,35 %.

Untersuchung 3: Reale Anforderungsdaten von EDAG und augmentierte Anforderungsdaten zur Bewertung der Leistungsfähigkeit zur Klassifizierung heterogener Daten

In dieser Untersuchung wird die Leistungsfähigkeit der Modelle in der Klassifizierung heterogener Daten bewertet. Es wird untersucht, wie sich die Leistungsfähigkeit der Modelle verändert, wenn die Modelle mit unbekannten Daten (im Vergleich zum Trainingsset) aus neuen Projekten getestet werden. Der primäre Anwendungsfall von KI-Modellen in der Praxis ist die Klassifizierung von unbekannten Daten aus neuen Projekten. Da besonders die industrielle Anwendbarkeit der Klassifizierung von Anforderungen durch die Anwendung augmentierter Daten geprüft werden soll, ist diese Untersuchung besonders wichtig für die Bewertung der Leistungsfähigkeit der Modelle.

Um die Transferfähigkeit zu prüfen, werden zwei Tests durchgeführt. In Test 1 werden Anforderungsdaten aus den Spezifikationen Montageriegelhaube (A) und verstellbare Stopperhaube (B) als Trainingsdaten verwendet. Als Testdaten werden die Anforderungsdaten des Kofferraumrollos (C) verwendet. In Test 2 werden die Trainingsdaten von Untersuchung 1 mit den augmentierten Daten (D) angereichert. Die Testdaten von Untersuchung 1 (Anforderungsdaten von Kofferraumrollo (C)) werden ebenfalls zur Bewertung der Modelle in Untersuchung 2 verwendet. Es wird der gewichtete durchschnittliche F1 der Modelle berechnet. Die Ergebnisse sind in Bild 7-4 dargestellt.

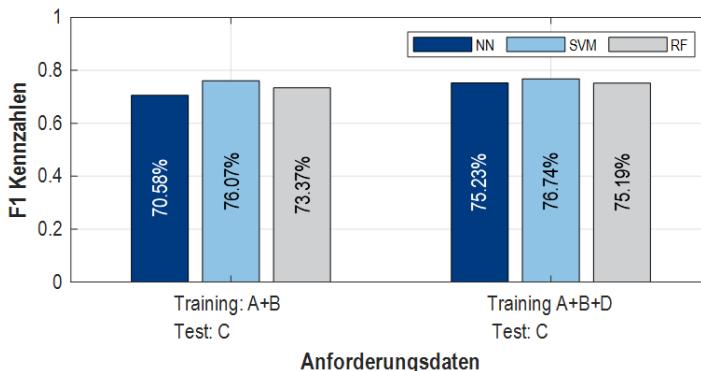


Bild 7-4: Ergebnisse zur Untersuchung der Leistungsfähigkeit zur Klassifizierung unbekannter, heterogener Daten aus neuen Projekten (NN: Neuronales Netz, SVM: Stützvektormaschine und RF: Zufallswald) [GPB22]

Wenn die augmentierten Daten verwendet werden (Test 2), steigt die Leistungsfähigkeit der Modelle (F1 der Modelle: NN: + 4,65 %, SVM: + 0,67 % & RF: +1,82 %). Im Durchschnitt steigt der F1 um 2,38 %. Die Leistungsfähigkeit der Modelle wird durch die Verwendung augmentierter Daten in der Klassifizierung unbekannter, heterogener Daten aus neuen Projekten verbessert. Dies ist für die industrielle Praxis besonders relevant, da es bedeutet, dass Daten aus neuen Projekten mit bestehenden, bereits gekennzeichneten Daten klassifiziert werden können (ohne neue Daten zu kennzeichnen und neue Modelle zu trainieren). Dies ist der beabsichtigte Anwendungsfall für die Extrahierung und Klassifizierung von Anforderungen in der industriellen Praxis.

Insbesondere die Leistungsfähigkeit des NN steigt in der Transferfähigkeit, wenn augmentierte Daten hinzugefügt werden (NN: +4,65 % F1). Auch in dieser Untersuchung besitzt die SVM die höchste Leistungsfähigkeit. In den gesamten Untersuchungen hat die SVM die besten Ergebnisse erzielt (was auch die Ergebnisse aus [KM17] zeigen).

Fazit

Es wurde untersucht, wie hoch die Leistungsfähigkeit in der Extrahierung und Klassifizierung von Anforderungen ist. Insbesondere wurde untersucht, wie die Extrahierung von Anforderungen mittels Daten-Augmentation für industrielle Anwendungen implementiert werden kann, um Anforderungsdaten mit geringem Aufwand zu erzeugen. Die erste Forschungsfrage dieser Arbeit (FF 1) bezieht sich auf die Höhe der Leistungsfähigkeit bei der Generierung von augmentierten Anforderungen. Die Ergebnisse zeigen, dass künstliche Anforderungen nur bedingt mit der Klasse der ursprünglichen Anforderung übereinstimmen (gewichteter durchschnittlicher F1 62,74 %). Dies bedeutet, dass vor der Anwendung der augmentierten Anforderungen zum Trainieren von KI-Modellen eine Qualitätsprüfung durchgeführt werden muss.

Die zweite Forschungsfrage (FF 2) bezieht sich auf die Leistungsfähigkeit von KI-Modellen, die mit augmentierten Daten trainiert worden sind, im Vergleich zu Modellen, die ausschließlich mit realen Daten trainiert worden sind. Die Ergebnisse zeigen, dass die augmentierten Daten aus der gleichen Domäne wie die Testdaten stammen müssen, um eine hohe Leistungsfähigkeit zu erzielen (F1 sinkt im Durchschnitt um 3,1 % von Untersuchung 1 zu Untersuchung 2). In diesen Untersuchungen wurden reale Anforderungsdaten der EDAG aus der Automobilindustrie und augmentierte Daten aus der Softwareentwicklung verwendet. Für die Anwendung in der Praxis muss darauf geachtet werden, dass die Modelle mit domänen spezifischen Daten trainiert werden. Das Training mit augmentierten Daten führt zu einer Leistungssteigerung bei der Untersuchung der Leistungsfähigkeit von Modellen für unbekannte, heterogene Anforderungsdaten (F1 steigt zwischen 0,67 % und 4,65 %). Dies bedeutet, dass die Leistungsfähigkeit für die Klassifizierung von Daten aus neuen Projekten steigt, wenn Modelle mit augmentierten Daten trainiert werden. Da dies der beabsichtigte Anwendungsfall für den Einsatz von KI-Modellen in der Praxis ist – Klassifizierung neuer Daten – ist dieses Ergebnis besonders relevant. Die Untersuchungen zeigen, dass die Verwendung augmentierter Daten die Leistungsfähigkeit von KI-Modellen zur Extrahierung von Anforderungen verbessert.

Insgesamt verbessert die entwickelte Methode die Anwendbarkeit der Anforderungsextraktion in der industriellen Praxis. Während der gesamten Untersuchungen lieferte das Modell der Stützvektormaschine (SVM) die besten Ergebnisse bei der Klassifizierung von Anforderungen. Weiteres Forschungspotenzial besteht in der Festlegung der Hyperparameter des Modells zur Augmentation von Daten speziell für Anforderungsdaten, um die Leistungsfähigkeit zu verbessern. Dies führt zu einer weiteren Verbesserung der Effizienz durch die Verringerung des Aufwands bei der Kennzeichnung der augmentierten Anforderungen. Darüber hinaus besteht weiteres Forschungspotenzial in der Verwendung von domänen spezifischen Daten zur Untersuchung der Leistungsfähigkeit in der Klassifizierung unbekannter, heterogener Daten aus neuen Projekten. In diesen Untersuchungen wurden augmentierte Daten aus der Softwareentwicklung verwendet. Die Ergebnisse zeigen, dass die Verwendung von augmentierten Daten zum Training aus einer anderen Domäne als die Testdaten zu einer geringeren Leistungsfähigkeit führt.

7.2 Formalisierung von Anforderungen

Um die Methode zur Formalisierung von Anforderungen zu validieren, werden acht Schlüssel-Wert-Paare identifiziert und Kennzahlen innerhalb eines Fallbeispiels berechnet, um die Leistungsfähigkeit der Methode zu bewerten. Die zu analysierenden Schlüssel-Wert-Paare wurden so gewählt, dass alle unterschiedlichen Formalisierungsansätze der Methode getestet werden können. Ein halbstrukturiertes Interview wird mit zwei Anforderungsingenieuren geführt. Ihnen werden die Methode und die Ergebnisse vorgestellt und sie bewerten den Nutzen für die generative Entwicklung in der Praxis.

Kennzahlen der Leistungsfähigkeit

Anforderungen aus den drei Anforderungsspezifikationen von EDAG ("Montage Riegelhaube", "verstellbare Stopperhaube" und "Kofferraumvorhang Rollo") werden formalisiert, um Schlüssel-Wert-Paare zu extrahieren. Die Leistungsfähigkeit der Formalisierung wird gemessen. Zu diesem Zweck wird ein Testset der zu formalisierenden Schlüssel-Wert-Paare erstellt. Um das Testset zu erstellen, werden Schlüssel-Wert-Paare aus dem Domänenlexikon in Anforderungsspezifikationen von EDAG gekennzeichnet. In den gekennzeichneten Anforderungen von EDAG existieren acht verschiedene Schlüssel-Wert-Paare, die aus den Anforderungen extrahiert werden müssen (siehe Tabelle 7-4).

Tabelle 7-4: *Schlüssel-Wert-Paare für die Validierung [GPB23]*

Schlüssel	Wertklasse	Erklärung
manufacturing_process	NER Modell (neu)	Zu verwendendes Fertigungsverfahren
manufacturing_site	NER Modell (spaCy)	Fertigungsstätte
material	NER Modell (neu)	Zu verwendetes Material
energy_type	String-Matching	Energieart, z. B. "elektrisch"
mass	Zahl & Einheit	Masse des Systems
lifespan_range	Zahl & Einheit	Lebensdauer des Systems in einer Einheit der Entfernung
lifespan_time	NER Modell (neu)	Lebensdauer des Systems in einer Einheit der Zeit
supplier_selected	NER Modell (spaCy)	Zulieferer des Systems

neu: trainiertes NER Modell mit Daten von EDAG

spaCy: vortrainiertes NER Modell von spaCy

Zur Messung der Leistungsfähigkeit bei der Formalisierung von Anforderungen, werden die Kennzahlen Präzision, Sensitivität und F1 verwendet. Um diese Kennzahlen zu bestimmen, müssen die wahren Positiven (t_p), die wahren Negativen (t_n), die falschen Positiven (f_p) und die falschen Negativen (f_n) berechnet werden:

- t_p : Die Anforderung enthält das Schlüssel-Wert-Paar und wird richtig klassifiziert.
- t_n : Die Anforderung enthält das Schlüssel-Wert-Paar nicht und wird richtig klassifiziert.
- f_n : Die Anforderung enthält das Schlüssel-Wert-Paar und wird falsch klassifiziert.
- f_p : Die Anforderung enthält das Schlüssel-Wert-Paar nicht und wird falsch klassifiziert.

Die Methode wird verwendet, um 76 Anforderungen aus den drei Spezifikationen von EDAG zu formalisieren, indem die in Tabelle 7-4 beschriebenen Schlüssel-Wert-Paare extrahiert werden. Für jeden Schlüssel wird eine Anzahl von Anforderungen ausgewählt, die einen Wert des jeweiligen Schlüssels enthalten. Alle anderen Anforderungen, die den jeweiligen Schlüssel nicht enthalten, werden ebenfalls getestet, um t_n und f_p zu bestimmen. Die Ergebnisse sind in Tabelle 7-5 dargestellt.

Tabelle 7-5: Ergebnisse der Formalisierung von Anforderungen [GPB23]

Schlüssel-Wert-Paar	Präzision	Sens.	F1	Anz. Anf. für Test/Training
manufacturing process	93,33 %	77,78 %	84,85 %	18/68
manufacturing site	100,00 %	100,00 %	100,00 %	3/-
material	100,00 %	60,00 %	75,00 %	10/59
energy type	100,00 %	100,00 %	100,00 %	4/-
mass	100,00 %	90,00 %	94,74 %	10/-
lifespan range	100,00 %	100,00 %	100,00 %	10/-
lifespan time	100,00 %	75,00 %	85,71 %	11/9
supplier selected	100,00 %	77,78 %	87,50 %	9/-
Makro-Durchschnitt	99,26 %	76,73 %	86,55 %	76/146

Anz. Anf.: „Anzahl an Anforderungen“; Sens.: „Sensitivität“

Insgesamt zeigen die Ergebnisse, dass die Methode die Schlüssel-Wert-Paare in den Anforderungen mit hoher Leistungsfähigkeit extrahiert (Makro-Durchschnitt F1 von 86,55 %). Insbesondere die Präzision ist hoch (99,26 %), was bedeutet, dass kaum falsch-positive Ergebnisse gefunden wurden. Dies deutet darauf hin, dass die Textindikatoren für die Erkennung von Schlüsseln korrekt definiert wurden. Die Ansätze "String-Matching" und "Zahl & Einheit" schneiden gut ab (durchschnittliches F1 100 % und 97,97 %). Auch die vortrainierten spaCy-Modelle zur Extraktion von "manufacturing_site" und "supplier_selected" schneiden in diesem Zusammenhang gut ab (F1

von 100,00 % und 87,50 %). Ein möglicher Grund für die geringe Leistungsfähigkeit bei der Extraktion von "Material" (Sensitivität von 60,00 %) ist, dass sich die Satzstruktur der Testdaten stark von den Trainingsdaten unterscheidet. Verbesserungspotenzial besteht darin, eine größere Menge an Trainingsdaten für "Material" zu generieren, sodass mehrere Satzstrukturen verarbeitet werden können.

Interview

Es wird ein semi-strukturiertes Interview mit zwei Anforderungsingenieuren von EDAG geführt. Dazu werden den Anforderungsingenieuren die Methode und die Ergebnisse vorgestellt und Fragen (F 1 bis F 3) gestellt. Die Ergebnisse werden aus den Antworten der Ingenieure konsolidiert (A 1 bis A 3).

F 1: "Sind die Informationen ausreichend, um die generative Entwicklung durchzuführen? Sind weitere Informationen erforderlich?"

A 1: Die Methode ist geeignet, um relevante Parameter für die generative Entwicklung einschließlich einer Nachhaltigkeitsbewertung des generierten Bauteils zu extrahieren. Die Formalisierungsansätze innerhalb der Methode (z. B. "Zahl & Einheit") sind auf andere Parameter übertragbar, z. B. "Volumen der Bauteile". In einigen Fällen müssen relevante Parameter (z. B. Material) aus anderen Parametern (z. B. Materialeigenschaften, wie wasserdicht) abgeleitet werden. Dies sind Parameter, die von Ingenieuren manuell abgeleitet werden müssen.

F 2: Bringt die Methode einen Mehrwert für die generative Entwicklung?

A 2: Aufgrund der Anzahl an Parametern, die für die Entwicklung eines Bauteils in komplexen technischen Systemen benötigt werden, ist die manuelle Extraktion der Parameter zeitaufwändig. Insbesondere durch die Integration der Formalisierung in den Arbeitsablauf der generativen Entwicklung wird Zeit gespart.

F 3: Ist die Methode anwendbar? Welche Informationen werden benötigt, um die Anwendbarkeit in der Praxis zu ermöglichen?

A 3: Eine natürlich-sprachliche Spezifikation ist die Grundlage für die Entwicklung und liegt vor. Intern müsste ein Prozess zur Formalisierung der Anforderungen definiert werden, da der Synera-Ingenieur keinen Zugriff auf die Anforderungsspezifikation hat. Da NER-Modelle nur eine geringe Anzahl an Trainingsdaten benötigen, ist der Aufwand für die Datengenerierung in der Praxis akzeptabel.

Fazit

Die Methode unterstützt die Formalisierung von natürlich-sprachlichen Anforderungen in englischer Sprache, die keiner festen Vorlage folgen. Aufgrund der vielfältigen Ansätze (z. B. NER-Modelle oder "Zahl & Einheit") ist die Methode auf verschiedene Arten

von Parametern übertragbar. Allerdings müssen einige Parameter, z. B. Material basierend auf Materialeigenschaften, manuell von Experten abgeleitet werden und können nicht direkt mit diesem Ansatz formalisiert werden. Der Aufwand für die Formalisierung von Anforderungen wird reduziert. Um den Aufwand weiter zu reduzieren, ist eine implementierte Schnittstelle für ein generatives Entwicklungs-Werkzeug erforderlich. Die benötigten Informationen sind in der Praxis verfügbar und der Aufwand für die Generierung von Trainingsdaten ist gering. Bei der Validierung wurden nur 146 Anforderungen für das Training benötigt, um die dargestellte Leistungsfähigkeit zu erreichen.

Die Metriken zeigen, dass Schlüssel-Wert-Paare automatisch mit hoher Leistungsfähigkeit extrahiert werden. Die extrahierten Parameter können im CSV-Format exportiert werden, so dass sie nach Implementierung einer Schnittstelle in generative Entwicklungs-Werkzeuge importiert werden können.

7.3 Abhängigkeitsanalyse von Anforderungen

Die Validierung der Abhängigkeitsanalyse von Anforderungen ist in drei Untersuchungen unterteilt. Es wird die Leistungsfähigkeit von active-BERT für die Klassifizierung von Anforderungsdaten aus neuen Projekten bewertet (1). Es wird der Einfluss einer heterogenen Datenbasis auf die Leistungsfähigkeit von active-BERT gemessen (2). Außerdem wird der Einfluss der Anzahl an Samples pro Iteration auf die Leistungsfähigkeit beim aktiven Lernen untersucht (3). Die Untersuchungen wurden im Rahmen der Betreuung einer Masterarbeit durchgeführt [Ngu23].

Die Anforderungsdaten für die drei Untersuchungen wurden im Rahmen der Feldstudie ARCA generiert. Hierzu werden Anforderungsdaten der technischen Produkte „intelligenter Knickarmroboter“ (BCN3D Moveo), „Toaster“ und „elektrische Gewürzmühle“ verwendet (siehe Anhang A5, Bild A-100 bis Bild A-102). Die technischen Produkte werden in Bild 7-5 dargestellt.



Knickarmroboter



Elektrische Gewürzmühle



Toaster

Bild 7-5: Fallbeispiele zur Validierung der Abhängigkeitsanalyse [Cla23, Mes23]

Die Anforderungsdaten des intelligenten Knickarmroboters wurden erstellt, indem Dokumente des Entwicklungsprojekts BCN3D Moveo analysiert sowie Komponenten des Produkts gefertigt und montiert worden sind [GOP22]. Insgesamt wurden 99 Anforderungen aufgenommen und deren Abhängigkeiten (4900) erkannt. Von 66 Studenten der Master-Maschinenbau Lehrveranstaltung „Produktentstehung“ wurde jeweils eine Anforderungsliste zu einer elektrischen Gewürzmühle (insgesamt 34 Anforderungslisten) oder zu einem Toaster (insgesamt 32 Anforderungslisten) mit durchschnittlich 26 Anforderungen generiert. Insgesamt wurden von den Studenten 1715 Anforderungen für die beiden Produkte aufgenommen: 837 Anforderungen für den Toaster und 878 Anforderungen für die Gewürzmühle. Zu jeder Anforderungsliste hat der jeweilige Student die Abhängigkeiten zwischen den Anforderungen bewertet (siehe Tabelle 7-6).

Tabelle 7-6: Anforderungsdaten der Vorstudie [GP23]

Produkt	Anzahl Anforderungen	Anzahl Anforderungsabhängigkeiten
Intelligenter Knickarmroboter	99	4900
Elektrische Gewürzmühle	878	11050
Toaster	837	10040

Untersuchung 1: active-BERT zur Klassifizierung von Daten aus neuen Projekten

In der ersten Untersuchung wird active-BERT initial mit den Anforderungsabhängigkeiten des intelligenten Knickarmroboters trainiert. In der Schleife des aktiven Lernens werden die Daten der elektrischen Gewürzmühle iterativ zum Trainieren und Testen des BERT-Modells verwendet. Es wird untersucht, wie die Leistungsfähigkeit von active-BERT durch das aktive Lernen verändert wird. In jeder Iteration werden 1000 Daten (bzw. Samples) der elektrischen Gewürzmühle zum Trainingsset hinzugefügt, ein neues BERT-Modell trainiert und mit den gesamten Anforderungsabhängigkeiten der elektrischen Gewürzmühle getestet. Die Anforderungsdaten, die zum Trainingsset hinzugefügt werden, werden mittels „Least-Confidence Uncertainty Sampling“ bestimmt. Es werden 1000 Samples in jeder Iteration für die Kennzeichnung durch das Orakel ausgewählt. Insgesamt werden 10 Iterationen durchgeführt und die Kennzahlen Präzision, Sensitivität und F1 berechnet (siehe Bild 7-6). In der Iteration 0 wurde die Schleife des aktiven Lernens nicht initiiert. Dieses Modell wurde vollständig mit den Anforderungsdaten des intelligenten Knickarmroboters trainiert.

Durch das aktive Lernen ist die Leistungsfähigkeit des BERT-Modells deutlich gestiegen (F1 Iteration 0: 26,32 % vs. F1 Iteration 10: 80,09 %). Bereits nach einer Iteration ist eine Steigerung des F1 von +20,08 % erfolgt. Nach Iteration 7 ändert sich F1 nur noch geringfügig und beträgt 78,54 %. Für Iteration 7 mussten vom Orakel insgesamt 7000 Anforderungsabhängigkeiten gekennzeichnet werden. Im Vergleich dazu mussten für Iteration 10 bereits 10000 Anforderungsabhängigkeiten gekennzeichnet werden und F1 steigt um weniger als 2 % an.

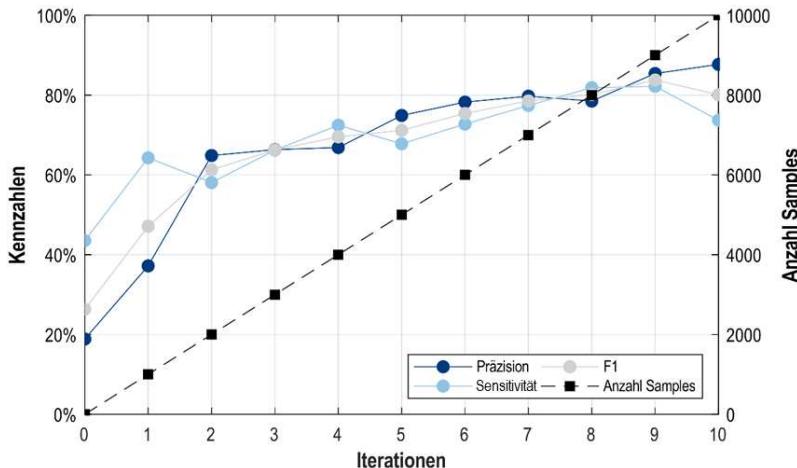


Bild 7-6: Ergebnisse von Untersuchung 1 [GP23]

Untersuchung 2: active-BERT mit einer heterogenen Trainingsdatenbasis zur Klassifizierung von Daten aus neuen Projekten

In Untersuchung 2 wird der Einfluss einer heterogenen Datenbasis im initialen Training von active-BERT untersucht. Das Modell wird initial mit jeweils 2450 Anforderungsabhängigkeiten des intelligenten Knickarmroboters und des Toasters trainiert. Die Summe der Anzahl an Daten der Produkte (4900) entspricht der Anzahl an Daten, die in Untersuchung 1 für das initiale Training verwendet worden sind. Im aktiven Lernen werden ebenfalls Daten der elektrischen Gewürzmühle zum Training und zum Testen ausgewählt. Es werden die Kennzahlen analog berechnet. Weitere Randbedingungen aus Untersuchung 1 wurden beibehalten: die Anzahl an Samples pro Iteration beträgt 1000, die Anzahl an Iterationen beträgt 10 und als Abfrage-Strategie wurde „Least Confidence Uncertainty Sampling“ ausgewählt. Die Ergebnisse sind in Bild 7-7 dargestellt.

Die Leistungsfähigkeit des BERT-Modells steigt durch das aktive Lernen an (F1 Iteration 0: 48,60 % vs. F1 Iteration 10: 81,63 %). Im Vergleich zu Untersuchung 1 wird deutlich, dass durch die heterogene Datenbasis der Einfluss des aktiven Lernens geringer ist. Die Leistungsfähigkeit des BERT-Modells ohne aktives Lernen ist deutlich höher als

in Untersuchung 1 (F1 Untersuchung 2 Iteration 0: 48,60 % vs. F1 Untersuchung 1 Iteration 0: 26,32 %). Insgesamt ist bei Untersuchung 2 eine Steigerung des F1 von +33,04 % und bei Untersuchung 1 eine Steigerung des F1 von +53,76 % des BERT-Modells durch das aktive Lernen aufgetreten. Nach 10 Iterationen erreicht active-BERT bei beiden Untersuchungen eine ähnlich hohe Leistungsfähigkeit (F1 Untersuchung 1 Iteration 10: 80,08 % vs. F1 Untersuchung 2 Iteration 10: 81,63 %).

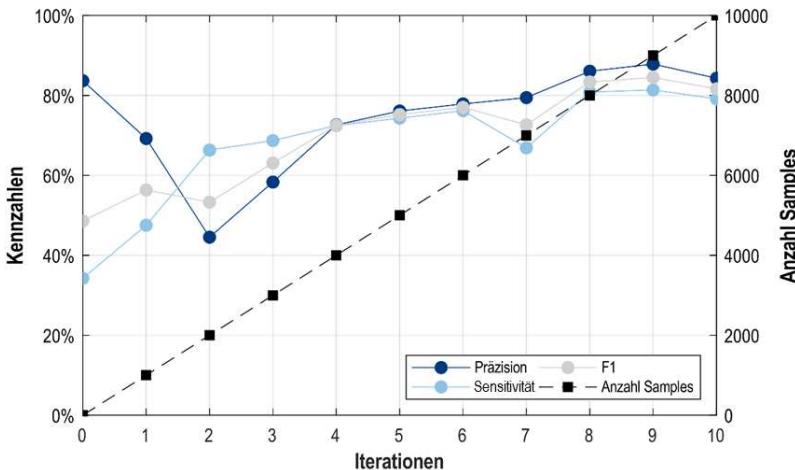


Bild 7-7: Ergebnisse von Untersuchung 2 [GP23]

Untersuchung 3: active-BERT mit einer niedrigen Anzahl an Samples pro Iteration zur Klassifizierung von Daten aus neuen Projekten

In Untersuchung 3 wird der Einfluss der Anzahl an Samples pro Iteration untersucht. Das Modell wird analog zu Untersuchung 1 mit Anforderungsabhängigkeiten des intelligenten Knickarmroboters trainiert. Statt mit einer Anzahl an Samples pro Iteration von 1000 werden in dieser Untersuchung pro Iteration 250 Samples zu den Trainingsdaten hinzugefügt. Das führt dazu, dass häufiger (im Verhältnis zu der Anzahl an Samples) das aktive Lernen durchgeführt wird. Beispielsweise werden in Untersuchung 1 bereits nach 2 Iterationen 2000 Samples zu den Trainingsdaten hinzugefügt. In Untersuchung 3 wird diese Anzahl an Samples erst nach 8 Iterationen erreicht. Das bedeutet, dass häufiger die Daten mit der höchsten Unsicherheit in der Klassifizierung bestimmt werden und der Experte häufiger Anforderungsdaten (in einer niedrigeren Anzahl) kennzeichnen muss. Weitere Randbedingungen aus Untersuchung 1 werden beibehalten: die Anzahl an Iterationen beträgt 10 und als Abfrage-Strategie wurde „Least Confidence Uncertainty Sampling“ ausgewählt. Die Ergebnisse sind in Bild 7-8 dargestellt. Für den besseren Vergleich mit Untersuchung 1 werden F1 von Untersuchung 1 und 3 gegenübergestellt. Dies ist durch die Punkte bei Iterationen 0, 4 und 8 dargestellt.

Im Vergleich zu Untersuchung 1 wird deutlich, dass sich die Leistungsfähigkeit des Modells nur geringfügig ändert, wenn die Anzahl an Samples pro Iteration verändert wird. Nach Iteration 4 mussten durch den Experten bei Untersuchung 3 1000 Anforderungsabhängigkeiten gekennzeichnet werden. F1 in Untersuchung 3 beträgt nach 4 Iterationen 49,86 %. Diese Anzahl an Samples musste in Untersuchung 1 nach Iteration 1 gekennzeichnet werden. F1 in Untersuchung 1 beträgt nach einer Iteration 47,12 %. Der F1 in Untersuchung 3 ist nach 1000 Samples, welcher der Trainingsdatenbasis hinzugefügt worden sind, um 1,74 % höher als in Untersuchung 1. Nach 2000 Samples, welcher der Trainingsdatenbasis hinzugefügt worden sind (siehe Iteration 8), ist der F1 in Untersuchung 3 um 0,97 % niedriger.

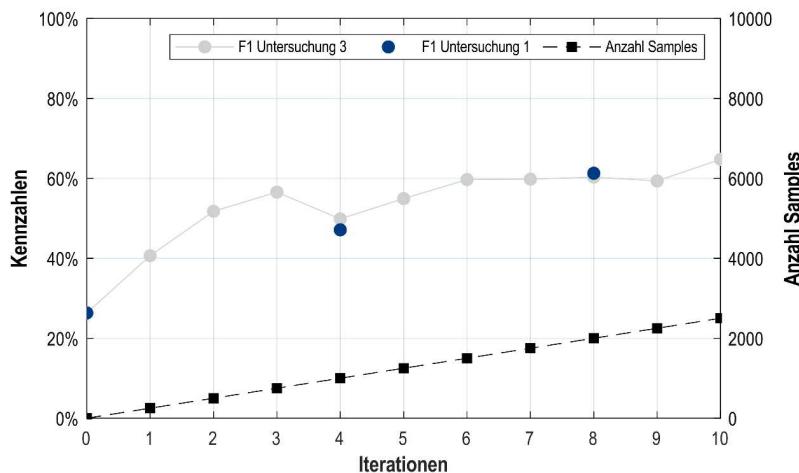


Bild 7-8: Ergebnisse von Untersuchung 3 [GP23]

Fazit

Durch das aktive Lernen nimmt die Leistungsfähigkeit von BERT-Modellen in der Klassifizierung von Daten aus neuen Projekten deutlich zu. Besonders für den Fall einer homogenen Trainingsdatenbasis (Anforderungsdaten von nur einem Produkt – Untersuchung 1) wurde durch das aktive Lernen nach 10 Iterationen eine Steigerung des F1 von +53,76 % erzielt. Für eine heterogene Datenbasis mit Daten von verschiedenen Produkten (Untersuchung 2) nimmt die Leistungsfähigkeit nach 10 Iterationen nur um +33,04 % zu. Durch die heterogene Datenbasis und die Ähnlichkeit der Produktdomänen (elektrische Gewürzmühle und Toaster sind aus der Domäne „Haushaltsgeräte“) sind mehr Trainingsdaten den Testdaten ähnlicher, woraus eine höhere Leistungsfähigkeit des BERT Modells ohne aktives Lernen (Iteration 0) resultiert. Nach einer bestimmten Anzahl an Iterationen (hier 4) nähern sich die F1 Werte für den Fall einer heterogenen und

homogenen Datenbasis aneinander an. Dem Trainingsset werden im aktiven Lernen eine hohe Anzahl an Daten des neuen Produkts hinzugefügt, sodass das BERT Modell diese mit einer höheren Leistungsfähigkeit klassifizieren kann und der Einfluss durch die unterschiedlichen initialen Trainingsdaten abnimmt.

Die Ergebnisse zeigen, dass BERT-Modelle in der Klassifizierung von Daten aus neuen Projekten eine hohe Leistungsfähigkeit erreichen, wenn das Modell entweder mit aktivem Lernen trainiert worden ist, wobei eine bestimmte Anzahl der unbekannten Daten von einem Experten gekennzeichnet werden muss, oder die Datenbasis für das Training des Modells möglichst heterogen und ähnlich (zum Beispiel gleiche Domäne) der neuen Daten ausgewählt worden ist. Die Ergebnisse aus Untersuchung 3 zeigen, dass die Anzahl an Samples pro Iteration nur einen geringen Einfluss auf die Leistungsfähigkeit des Modells hat.

7.4 Validierung des Methodik-Erfolgs

Der Methodik-Erfolg wird anhand des Erfüllungsgrads der Erfolgsfaktoren der Methodik bewertet. Die Validierung des Methodik-Erfolgs wird durchgeführt, indem die Methodik von fünf Unternehmensvertretern von EDAG und KRAUSE mittels Bewertungsbögen bewertet wird. Anschließend wird die Erfüllung der Erfolgsfaktoren der Methodik bewertet. Die Unternehmensvertreter führen die Rollen „Requirements Engineer“, „Projektleiter“ sowie „Data Scientist“ aus. Das Software-Werkzeug der Methodik wird den Unternehmensvertretern präsentiert. Die Präsentation ist anhand des Workflows des Software-Werkzeugs (siehe Bild 6-21) strukturiert. Es werden nacheinander die Schritte „Projekt erstellen und verwalten“ bis „Ergebnisse exportieren“ vorgestellt. Hierzu werden Erklärungen der Funktionsweisen der Schritte, bzw. der Methoden zur Extrahierung und Formalisierung technischer Anforderungen den Unternehmensvertretern gegeben. Die Erklärungen sind nach der „Input-Processing-Output“ (IPO) Logik aufgebaut. Es werden benötigte Inputs, die Funktionsweise der Verarbeitung (bzw. „Processing“) und die Outputs der Methoden erläutert. Die Erklärungen werden mit Beispielen von EDAG unterstützt. Die Validierungen der einzelnen Methoden (siehe Abschnitte 7.1 bis 7.3) wurden mit EDAG Produkten, bzw. deren Anforderungsdaten durchgeführt. Die Anforderungsdaten dieser Produkte werden als Beispiele für die Erklärungen der Funktionsweisen genutzt. Nach der Präsentation werden Implikationen der Methodik für die industrielle Praxis diskutiert. Beispielsweise wurde bei der Validierung der Abhängigkeitsanalyse festgestellt, dass für eine hohe Leistungsfähigkeit der BERT-Modelle mehrere hundert Anforderungsdaten von Experten gekennzeichnet werden müssen.

7.4.1 Auswertung der Bewertungsbögen

Nach der Präsentation des Software-Werkzeugs sowie der Methoden und der Diskussion, werden den fünf Unternehmensvertretern Bewertungsbögen mit 23 Thesen zu der Methodik (siehe Anhang A3.2) ausgehändigt. Der Bewertungsbogen ist in die folgenden vier Bereiche unterteilt:

- 1) Thesen zur Ist-Situation
- 2) Thesen zur Anwendbarkeit des Software-Werkzeugs
- 3) Thesen zum Anwendungsaufwand
- 4) Thesen zum Mehrwert der Methodik-Anwendung

Die Unternehmensvertreter bewerten die Thesen nach der folgenden Skala:

- 2: stimme zu
- 1: stimme eher zu
- 0: teilweise Zustimmung / teilweise Ablehnung
- 1: stimme eher nicht zu
- 2: stimme nicht zu

Nach dem Ausfüllen der Bewertungsbögen wird mit den Industrievertretern jeweils ein Interview geführt, um die Gründe für die Bewertung der Thesen zu erfragen. Falls den Thesen nicht vollständig zugestimmt wird (bzw. die vollständige Zustimmung erfolgt für die Thesen, die negativ formuliert sind), dann wird die Bewertung durch die Industrievertreter erläutert.

Ergebnisse der Bewertungsbögen

Die **Thesen zur Ist-Situation** adressieren Herausforderungen in der Produktentwicklung, welcher dieser Arbeit zugrunde legen. Die Ergebnisse sind in Bild 7-9 dargestellt.

- These 1: „Zur Extrahierung von Anforderungen aus Lastenheften ist ein hoher manueller Aufwand notwendig.“
- These 2: „Natürlich-sprachliche Anforderungen sind durch Software-Werkzeuge in der Produktentwicklung üblicherweise nicht direkt interpretierbar. Es besteht ein Bedarf nach formalisierten Anforderungen.“
- These 3: „Anforderungsänderungen werden in Unternehmen noch nicht systematisch gehandhabt.“

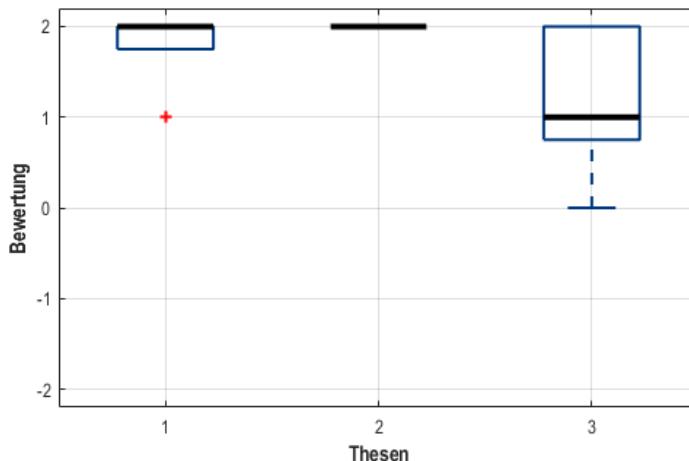


Bild 7-9: Ergebnisse der Thesen zur Ist-Situation (schwarz: Median)

Die Industrievertreter bewerten These 1 durchschnittlich mit 2 (Median). Sie stimmen zu, dass der Aufwand zur manuellen Extrahierung hoch ist. Im Unternehmen besteht ein Bedarf nach einer automatisierten Extrahierung von Anforderungen. Ebenfalls stimmen die Industrievertreter mit These 2 überein (Median = 2). Anforderungen müssen formalisiert werden, um durch Entwicklungswerkzeuge direkt interpretierbar zu werden. In den Unternehmen der Industrievertreter (EDAG und KRAUSE) ist vor allem der Anwendungsfall von formalisierten Anforderungen für die generative Entwicklung von Bauteilen in Synera relevant. These 3 wird nur teilweise zugestimmt (Median = 1). Industrievertreter von EDAG geben an, dass Anforderungsänderungsmanagement-Prozesse im Unternehmen durchgeführt werden. Es werden jedoch Auswirkungen von Anforderungsänderungen bestimmt ohne Anforderungsabhängigkeiten automatisiert zu erkennen. Sie geben an, dass der Aufwand für die manuelle Erkennung von Anforderungsänderungen hoch ist, weshalb diese häufig nicht erkannt werden. Sie sehen einen großen Mehrwert darin Anforderungsabhängigkeiten zu erkennen, um Auswirkungen von Anforderungsänderungen systematisch zu bewerten.

Die Ergebnisse der **Thesen zur Anwendbarkeit des Software-Werkzeugs** sind in Bild 7-10 dargestellt. Thesen 4-9 sind einem standardisierten „System Usability Scale“ (SUS) Bewertungsbogen entnommen, um die Nutzerfreundlichkeit des Systems zu messen [Bro95]. Thesen 5, 6, 7, 9 sind negativ formuliert. Umso niedriger die Bewertung dieser Thesen ist, umso positiver ist die Anwendbarkeit des Software-Werkzeugs zu werten:

- These 4: „Ich kann mir sehr gut vorstellen, das Software-Werkzeug regelmäßig zu nutzen.“
- These 5: „Ich empfinde das Software-Werkzeug als unnötig komplex.“

- These 6: „Ich denke, dass ich technischen Support brauchen würde, um das Software-Werkzeug zu nutzen.“
- These 7: „Ich finde, dass es im Software-Werkzeug zu viele Inkonsistenzen gibt.“
- These 8: „Ich kann mir vorstellen, dass die meisten Leute das Software-Werkzeug schnell zu beherrschen lernen.“
- These 9: „Ich muss voraussichtlich eine Menge Dinge lernen, bevor ich mit dem Software-Werkzeug arbeiten kann.“
- These 10: „Das Software-Werkzeug ist für branchenübliche Lastenhefte anwendbar.“
- These 11: „Das Software-Werkzeug ist generisch für die Entwicklung komplexer technischer Systeme anwendbar.“
- These 12: „Ich kann mir vorstellen, dass die Anwendung des Software-Werkzeugs in unternehmensinterne Entwicklungsprozesse integriert werden kann.“

Die Industrievertreter geben an, dass sie das Software-Werkzeug teilweise regelmäßig nutzen würden (These 4: Median = 1). Der Data-Scientist merkt an, dass er das Software-Werkzeug nicht direkt anwendet, sondern die Daten für die Anwendung der Software im Unternehmen vorbereitet. Die Bedienung ist für die Industrievertreter intuitiv. Der Funktionsumfang ist auf die Kernfunktionalitäten der Methodik beschränkt (These 5: Median = -1). Es wird voraussichtlich kein technischer Support benötigt, um die Software zu nutzen (These 6: Median = -1). Die Industrievertreter geben an, dass sie zur Bereitstellung der Daten zur Anwendung des Werkzeugs (vor allem für das Training von KI-Modellen) sich mit Data-Scientists austauschen müssen. Die Anwendung des Werkzeugs (sofern die Daten verfügbar sind) ist ohne technischen Support möglich. Im Software-Werkzeug sind keine Inkonsistenzen aufgefallen (These 7: Median = -2). Die Anwendung des Software-Werkzeugs ist einfach zu erlernen (These 8: Median = 2; These 9: Median = -2). Die Anwender geben an, dass das Software-Werkzeug überwiegend für branchenübliche Lastenhefte geeignet ist (These 10: Median = 1). Die Tabelle zur Übersicht der extrahierten Anforderungen visualisiert die Anforderungen geeignet. In Abhängigkeit des Anwendungsfalls müssen weitere Schlüssel-Wert-Paare erkannt werden. Im Prototyp wurde lediglich die Formalisierung von Schlüssel-Wert-Paaren im Kontext der generativen Entwicklung implementiert. Weitere Funktionalitäten sind für branchenübliche Lastenhefte geeignet. Das Software-Werkzeug ist teilweise generisch anwendbar, jedoch müssen Domänenlexika in Abhängigkeit des Anwendungsfalls angepasst werden (These 11: Median = 0). Das Werkzeug ist anwendbar in unternehmensinternen Entwicklungsprozessen (These 12: Median = 1). Der Referenzprozess zur Anwendung der Methodik zeigt Potenziale für die Integration des Werkzeugs, bzw. der Methodik in den Entwicklungsprozess.

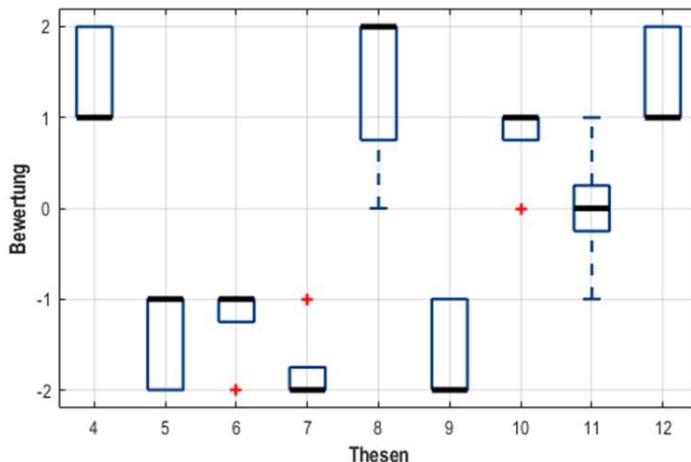


Bild 7-10: Ergebnisse der Thesen zur Anwendbarkeit des Software-Werkzeugs (schwarz: Median)

Zusätzlich wird in der Anwendbarkeit der Methodik bewertet, ob „zur Anwendung des Werkzeugs sind die Informationen in der Praxis frühzeitig (in der Regel ab der Projektinitierungsphase) in Form von Daten oder Expertenwissen verfügbar“ (These 13). Hierzu wird den Unternehmensvertretern eine Liste der benötigten Informationen zur Anwendung des Werkzeugs in der Praxis gegeben. In Tabelle 7-7 sind die Antworten der Unternehmensvertreter zu den benötigten Informationen dargestellt. In der rechten Spalte ist das arithmetische Mittel der Antworten der Unternehmensvertreter dargestellt. „Vorhanden“ wurde mit 1 bewertet, „nicht vorhanden“ wurde mit 0 bewertet.

In der Projektinitierungsphase liegen natürlich-sprachliche Lastenhefte vor (Bewertung: 1). Das Wissen zur Definition von Strukturregeln ist teilweise vorhanden (Bewertung: 0,4). Es muss geprüft werden, welches Schlüssel-Wert-Paar in welcher Art von Anforderung vorkommt. Die Unternehmensvertreter besitzen das notwendige Wissen, um diese Prüfung durchzuführen und die Strukturregeln entsprechend zu definieren. Analog zu den Strukturregeln müssen Textindikatoren für die Domänenlexika zusammengestellt werden. Die Industrievertreter besitzen teilweise das notwendige Wissen zur Definition von Textindikatoren (Bewertung: 0,2). Die gekennzeichneten Daten zum Training von NER-Modellen sind nicht vorhanden (Bewertung: 0). Jedoch ist das Wissen zur Generierung von Trainingsdaten vorhanden. Daten von Anforderungsabhängigkeiten für das initiale Training eines KI-Modells für die Abhängigkeitsanalyse von Anforderungen sind häufig nicht im Unternehmen vorhanden (Bewertung: 0,4). Das Wissen zur Erkennung von Anforderungsabhängigkeiten ist teilweise vorhanden (Bewertung: 1). Die Anzahl der auf-

getretenen Anforderungsänderungen in vergangenen Projekten wird üblicherweise dokumentiert (Bewertung: 0,6). Das Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen liegt üblicherweise in Unternehmen vor (Bewertung: 0,6).

Tabelle 7-7: Ergebnisse zu These 13: Frühzeitige Verfügbarkeit benötigter Informationen zur Anwendung der Methodik

ID	Benötigte Informationen zur Anwendung der Methodik	Bewertung
1	Natürlich-sprachliche Anforderungsspezifikation, bzw. Lastenheft	1
2	Wissen zur Definition von Strukturregeln von Schlüssel-Wert-Paaren	0,4
3	Wissen zur Definition von Textindikatoren von Schlüssel-Wert-Paaren	0,2
4	Gekennzeichnete Daten zum Training von Named-Entity-Recognition Modellen für Schlüssel-Wert-Paare	0
5	Anforderungsabhängigkeiten (natürlich-sprachliche Anforderungspaare sowie die Abhängigkeit zwischen den Anforderungen) für das initiale Training eines KI-Modells	0,4
6	Wissen zur Bewertung von Anforderungsabhängigkeiten	0,4
7	Anzahl der aufgetretenen Anforderungsänderungen in vergangenen Projekten	0,6
8	Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen	0,6

Die Ergebnisse der **Thesen zum Anwendungsaufwand des Software-Werkzeugs** sind in Bild 7-11 dargestellt. Die folgenden Thesen gehören zu dieser Kategorie:

- These 14: „Den Vorbereitungsaufwand, bevor die Anwendung möglich ist, empfinde ich als angemessen, Aktivitäten: Erstellen eines Domänenlexikons; Auswahl und Vorbereitung von Anforderungsabhängigkeitsdaten; Zusammenstellung einer Änderungshistorie vergangener Projekte“
- These 15: „Den Anwendungsaufwand schätze ich als angemessen ein, Aktivitäten: Import von Lastenheften, Prüfen und Auswahl von Schlüssel-Wert-Paaren, Abhängigkeitsanalyse, Bewertung von Änderungsauswirkungen; Ausgabe der Ergebnisse im CSV-Format“

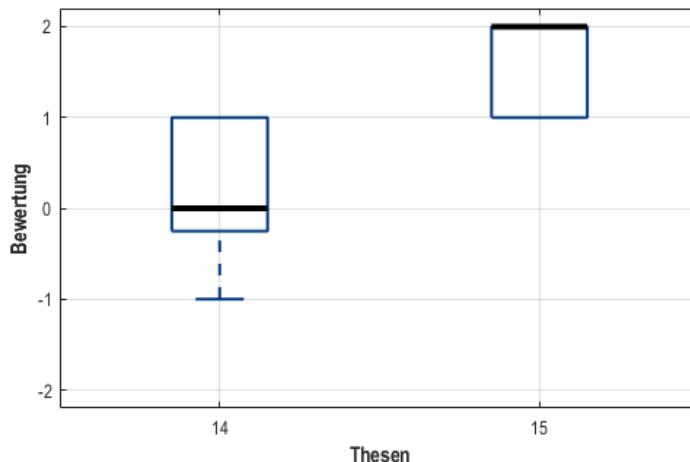


Bild 7-11: Ergebnisse der Thesen zum Anwendungsaufwand des Software-Werkzeugs (schwarz: Median)

Die Industrievertreter bewerten den Vorbereitungsaufwand überwiegend als angemessen (These 14: Median = 0). Bemängelt wird, dass der Aufwand zur Generierung der Domänenlexika hoch ist, weil dafür Anforderungsdokumente manuell analysiert werden müssen, um Strukturregeln und Textindikatoren zu definieren. Der Anwendungsaufwand wird als angemessen bewertet (These 15: Median = 2). Ein Defizit ist, dass während der Abhängigkeitsanalyse von Anforderungen der Anwender mehrere Minuten mit dem Kennzeichnen von Anforderungsabhängigkeiten warten muss. Während des aktiven Lernens werden erst nach dem Training der Modelle und der Berechnung der Unsicherheit in der Klassifizierung von Anforderungsabhängigkeiten Daten zum Kennzeichnen durch den Anwender bestimmt.

Die Ergebnisse der **Thesen zum Mehrwert der Methodik-Anwendung** sind in Bild 7-12 dargestellt. Die folgenden Thesen gehören zu dieser Kategorie:

- These 16: „Durch die automatisierte Extrahierung von Anforderungen aus Lastenheften wird Aufwand reduziert und es wird erleichtert, eine vollständige Anforderungsliste zu generieren.“
- These 17: „Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.“
- These 18: „Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.“

- These 19: „Auf Grundlage der Anforderungsabhängigkeiten können Auswirkungen von Anforderungsänderungen besser bewertet werden.“
- These 20: „Auf Grundlage der Informationen zum Änderungsrisiko von Anforderungen können Risiken realistischer abgeschätzt werden als zuvor.“
- These 21: „Anhand der teilautomatisierten Risikoanalyse kann ein höherer Anteil an Anforderungen eines Anforderungssets auf Änderungsrisiken untersucht werden.“
- These 22: „Die Anwendung des Software-Werkzeugs trägt zur Effizienzsteigerung in der Entwicklung komplexer technischer Systeme bei.“
- These 23: „Durch den CSV-Export können die Anforderungsdaten aufwandsarm in branchenübliche Anforderungsmanagement-Software importiert werden.“

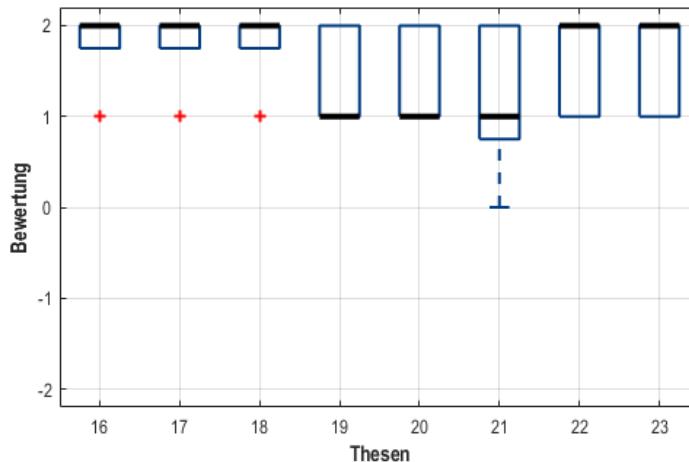


Bild 7-12: Ergebnisse der Thesen zum Mehrwert der Methodik-Anwendung (schwarz: Median)

Die Industrievertreter stimmen zu, dass der Aufwand zur Extrahierung und Formalisierung von Anforderungen reduziert wird (Thesen 16-18, Median = 2). Durch die Kenntnis von Anforderungsabhängigkeiten werden Auswirkungen von Anforderungsänderungen besser bewertbar (These 19, Median = 1). Risiken von Anforderungsänderungen können besser eingeschätzt werden. Jedoch wäre eine monetäre Bewertungseinheit des Risikos für die Praxis besser interpretierbar (These 20, Median = 1). Durch die Teilautomatisierung wird die Bewertung des Änderungsrisikos effizienter und die Bewertung von mehr Anforderungen ermöglicht. Außerdem werden durch die Priorisierung von Anforderungen anhand der Konnektivität im Anforderungsnetz (abgebildet durch den PageRank) wesentliche Anforderungen zur Bewertung aufgezeigt (These 21, Median

= 1). Durch das Software-Werkzeug werden Effizienzpotenziale in der Entwicklung komplexer technischer Systeme erschlossen (These 22, Median = 2). Der CSV-Export ermöglicht den aufwandsarmen Import der Anforderungsdaten in IBM-DOORS (These 23, Median = 2).

Fazit

Die Industrievertreter sehen Herausforderungen in der Produktentwicklung hinsichtlich des hohen Aufwands der Extrahierung und Formalisierung von Anforderungen sowie in der Handhabung von Anforderungsänderungen (Thesen zur Ist-Situation). Die Anwendbarkeit des Software-Werkzeugs ist hoch. Häufig sind gekennzeichnete Anforderungsabhängigkeiten für das initiale Training eines Modells nicht vorhanden. Das Wissen zur Kennzeichnung von Anforderungsabhängigkeiten liegt jedoch vor (Thesen zur Anwendbarkeit). Der Aufwand zur Nutzung des Software-Werkzeugs (Vorbereitungs- und Durchführungsaufwand) wird überwiegend als angemessen bewertet. Es wird bemängelt, dass Domänenlexika erstellt werden müssen (Thesen zum Anwendungsaufwand). Die Industrievertreter bewerten den Mehrwert der Methodik als hoch. Eine monetäre Bewertung des Änderungsrisikos wäre für die Praxis geeigneter. (Thesen zum Mehrwert).

7.4.2 Erfüllung der Erfolgsfaktoren

Auf Basis der Validierungsergebnisse der Methoden (siehe Abschnitte 7.1 bis 7.3) sowie der Ergebnisse der Auswertung der Bewertungsbögen (siehe Abschnitt 7.4.1) wird die Erfüllung der Erfolgsfaktoren bewertet.

Input

Erfolgsfaktor 1: „Das System muss dem Anwender die Möglichkeit bieten, ein branchenübliches Lastenheft einlesen zu können.“

Der Erfolgsfaktor ist erfüllt, wenn:

- PDF-Dokumente verarbeitet werden können (Dateiformat),
- Fließtext, Listen und Tabellen verarbeitet werden können (Formatierung im Text),
- Co-Referenzen aufgelöst werden können (Verweise),
- Anforderungen mit niedriger Qualität verarbeitet werden können (Qualitätsdefizite),
- unstrukturierte Anforderungsbeschreibungen verarbeitet werden können (Anforderungsbeschreibung) und
- funktionale und nicht-funktionale Anforderungen mit einer hohen Leistungsfähigkeit erkannt werden können (Anforderungsarten).

In der Extrahierung von Anforderungen werden Textsegmente aus PDF-Dokumenten extrahiert. Diese Textsegmente werden aus dem Fließtext, Listen sowie Tabellen entnommen. Co-Referenzen werden aufgelöst. Es können Anforderungen niedriger Qualität verarbeitet werden, da keine Randbedingungen der natürlich-sprachlichen Form (z. B. Anforderungsschablonen) vorgegeben werden. Es werden funktionale und nicht-funktionale Anforderungen mit einer hohen Leistungsfähigkeit erkannt. Der F1 der Extrahierung von Textsegmenten und der Klassifizierung der Anforderungsart beträgt 74,84 % (bei dem Test einer SVM mit Lastenheften von EDAG, siehe Bild 7-2). Die Unternehmensvertreter stimmen zu, dass branchenübliche Lastenhefte für die Untersuchungen in der Validierung verwendet worden sind (These 10: Median = 1, siehe Bild 7-10). Die Leistungsfähigkeit der Extrahierung von Anforderungen ist hoch bei der Verwendung branchenüblicher Lastenhefte. *Erfolgsfaktor 1 ist erfüllt.*

Erfolgsfaktor 2: „Das System muss dem Anwender die Möglichkeit bieten, ein englischsprachiges Lastenheft einlesen zu können.“

Der Erfolgsfaktor ist erfüllt, wenn die Leistungsfähigkeit der Methoden hoch ist, wenn englischsprachige Anforderungen eingelesen werden. Die Methoden zur Extrahierung und Formalisierung von Anforderungen sowie die Abhängigkeitsanalyse von Anforderungen wurden mit englischen Anforderungsdaten mittels Fallbeispielen validiert. Der F1 für die Extrahierung und Klassifizierung von Anforderungen beträgt 74,84 % (bei dem Test einer SVM mit Lastenheften von EDAG, siehe Bild 7-2). In der Formalisierung von Anforderungen wurden Schlüssel-Wert-Paare, die für die generative Entwicklung von Bauteilen relevant sind, mit einem F1 von 86,55 % (Makro Durchschnitt der Ergebnisse, siehe Tabelle 7-5) extrahiert. Abhängigkeiten von englischen Anforderungen wurden mit einer hohen Leistungsfähigkeit erkannt (F1 beträgt nach 10 Iterationen 80,09 %, siehe Bild 7-6). Englischsprachige Anforderungen können verarbeitet werden, da die Leistungsfähigkeit der Methoden hoch ist, wobei englische Anforderungen als Eingangsdaten verwendet worden sind. *Erfolgsfaktor 2 ist erfüllt.*

Tabelle 7-8: Bewertung der Erfolgsfaktoren (EF) der Kategorie "Input"

EF	Bewertungskriterien	Erfüllungsgrad
1	<ul style="list-style-type: none"> • F1 der Extrahierung von Textsegmenten und der Klassifizierung der Anforderungsart: 74,84 % • These 10, Median = 1 	erfüllt
2	<ul style="list-style-type: none"> • F1 der Extrahierung englischsprachiger Anforderungen: 74,84 % • F1 der Formalisierung englischsprachiger Anforderungen: 86,55 % • F1 der Abhängigkeitsanalyse englischsprachiger Anforderungen: 80,09 % 	erfüllt

Funktionalität

Erfolgsfaktor 3: „Das System muss Anforderungsdaten aufwandsarm generieren können.“

Der Erfolgsfaktor ist erfüllt, wenn Anforderungsdaten mit einer hohen Leistungsfähigkeit generiert werden und der Aufwand zur Generierung der Anforderungsdaten niedrig ist. In Abschnitt 7.1.1 wird die Untersuchung der Generierung von Anforderungsdaten beschrieben. Es wird GPT-J angewendet, um künstliche Anforderungsdaten zu generieren. GPT-J ist ein KI-Modell und ermöglicht die automatisierte Augmentierung von Textdaten. Der Anwendungsaufwand zur Nutzung von GPT-J ist aufgrund des hohen Automatisierungsgrads niedrig. Hierbei wurde festgestellt, dass die F1 Werte der einzelnen Klassen von künstlich generierten Anforderungen (funktional, nicht-funktional und keine Anforderung) zwischen 45,90 % und 56,00 % liegen. Der Makro-Durchschnitt von F1 sowie der gewichtete Durchschnitt von F1 liegen bei 64,71 % bzw. 62,74 % wenn generierte Daten nach „Anforderung“ und „keine Anforderung“ unterteilt werden. Demnach entspricht ein großer Teil der generierten Anforderungen nicht der Klasse der ursprünglichen Anforderung, welche zur künstlichen Generierung der Anforderungsdaten verwendet worden ist. Die Zahl der falsch negativen Ergebnisse ist hoch. Das bedeutet, dass häufig aus einem Text der Klasse "Anforderung" ein künstlicher Text der Klasse "keine Anforderung" erzeugt wurde. Dies zeigt sich auch in der niedrigen Sensitivität (45,59 %). Ein Text der Klasse "keine Anforderung" wird zu einem hohen Anteil korrekt aus einem Text der Klasse "keine Anforderung" künstlich erzeugt.

Die Ergebnisse zeigen, dass GPT-J zur Erzeugung künstlicher Anforderungen verwendet werden kann. Allerdings muss aufgrund der Leistungsfähigkeit eine Qualitätskontrolle der künstlichen Anforderungen durchgeführt werden, um bei der Betrachtung von Daten der Klasse "Anforderung" korrekte Kennzeichnungen zu vergeben. *Erfolgsfaktor 3 ist teilweise erfüllt.* Die Generierung von Anforderungsdaten ist aufwandsarm möglich. Allerdings muss eine Qualitätskontrolle der Anforderungsdaten der Klasse „Anforderung“ durchgeführt werden, um die korrekte Kennzeichnung zu ermitteln. Die korrekte Kennzeichnung ist erforderlich, um KI-Modelle mit den Anforderungsdaten zu trainieren und eine hohe Leistungsfähigkeit des Modells zu erreichen.

Erfolgsfaktor 4: „Das System muss Anforderungen nach ihrer Art klassifizieren können.“

Die extrahierten Textsegmente werden mit einer hohen Ergebnisgüte hinsichtlich Anforderungsart – funktional und nicht-funktional – klassifiziert (F1 bei der Verwendung einer SVM: 74,84 %, siehe Bild 7-2). Der Anforderungsingenieur von EDAG bewertet die Unterteilung in funktional und nicht-funktional als akzeptabel. Eine weitere Unterteilung der nicht-funktionalen Anforderungen in beispielsweise „rechtliche“ Anforderungen wäre hilfreich, um die Anforderungsliste weiter zu strukturieren und die Anforderungen für verschiedene Rollen besser selektieren zu können. In der Evaluation waren nicht ausreichend Anforderungsdaten für die jeweiligen Sub-Klassen nicht-funktionaler Anforderungen vorhanden, weshalb diese Untersuchung nicht durchgeführt worden ist.

Die Methode ist für weitere Sub-Klassen erweiterbar, sofern die Anzahl an Daten für das Training der KI-Modelle ausreichend ist. In den Untersuchungen dieser Arbeit wurden 562, 408 und 354 Anforderungsdaten (funktional, nicht-funktional, keine Anforderung) verwendet und eine hohe Leistungsfähigkeit erzielt. Zur Untersuchung von Sub-Klassen wird daher empfohlen, dass die Anzahl an Daten der Sub-Klassen mindestens der Anzahl an Daten in den Untersuchungen entsprechen (ca. 400-500 Daten pro Sub-Klasse).

Damit die Anwendbarkeit der Methodik hoch ist, müssen Modelle mittels künstlicher Anforderungsdaten trainiert werden können. Es wurden augmentierte Anforderungsdaten generiert und mittels eines heterogenen Testdatensatz die Leistungsfähigkeit der Klassifizierung von Anforderungen bestimmt. Der F1 beträgt 76,74 % (siehe Bild 7-4). Aufgrund der hohen Leistungsfähigkeit der Modelle zur Klassifizierung von Anforderungen bei gleichzeitiger Verwendung künstlicher Anforderungsdaten zum Training der Modelle, ist die Anwendbarkeit der Methode in der industriellen Praxis hoch. *Erfolgsfaktor 4 ist erfüllt.*

Erfolgsfaktor 5: „Das System muss Anforderungen formalisieren können.“

Der Erfolgsfaktor ist erfüllt, wenn Anforderungen mit einer hohen Leistungsfähigkeit formalisiert werden können. Speziell wird der Anwendungsfall für die generative Entwicklung von Bauteilen untersucht. Es muss geprüft werden, ob Parameter, die für die generative Entwicklung relevant sind und in Lastenheften dokumentiert werden, formalisiert werden können. Es wurden acht Schlüssel-Wert-Paare, die aus Anforderungen extrahiert werden sollen und für die generative Entwicklung relevant sind, definiert. Im Rahmen eines Fallbeispiels wurden für diese Schlüssel-Wert-Paare Domänenlexika definiert und die Methode zur Formalisierung von Anforderungen angewendet (siehe Abschnitt 7.2). Die Ergebnisse zeigen, dass die Methode die Schlüssel-Wert-Paare in den Anforderungen mit hoher Leistungsfähigkeit extrahiert (Makro-Durchschnitt F1 von 86,55 %). Insbesondere die Präzision ist hoch (99,26 %), was bedeutet, dass kaum falsch-positive Ergebnisse gefunden wurden. Dies deutet darauf hin, dass die Textindikatoren für die Erkennung von Schlüsseln korrekt definiert wurden. Die Ansätze "String-Matching" und "Zahl & Einheit" schneiden gut ab (durchschnittlicher F1 100 % und 97,97 %). Auch die vortrainierten spaCy-Modelle zur Extraktion von "manufacturing_site" und "supplier_selected" schneiden in diesem Zusammenhang gut ab (F1 von 100,00 % und 87,50 %). Aufgrund der hohen Leistungsfähigkeit sind die Ansätze „NER, Zahl & Einheit sowie String-Matching“ für die jeweils gewählten Schlüssel-Wert-Paare zur Formalisierung von Anforderungen geeignet. *Erfolgsfaktor 5 ist erfüllt.*

Erfolgsfaktor 6: „Das System muss Abhängigkeiten von Anforderungen erkennen können.“

Der Erfolgsfaktor ist erfüllt, wenn Abhängigkeitsarten mit einer hohen Leistungsfähigkeit erkannt werden. Es wurde im Rahmen eines Fallbeispiels mit Anforderungsdaten eines Knickarmroboters, einer elektrischen Gewürzmühle und einem Toaster die Leistungsfähigkeit der Abhängigkeitsanalyse mittels BERT-Modell, welches mit aktivem Lernen

trainiert worden ist, untersucht (siehe Abschnitt 7.3). Durch das aktive Lernen erreichen BERT-Modelle eine hohe Leistungsfähigkeit, bzw. F1 von 81,63 % (Iteration 10, Untersuchung 2). Es wurde untersucht, wie hoch die Leistungsfähigkeit zur Klassifizierung von Anforderungsabhängigkeiten mittels heterogener Anforderungsdaten ist. Dies kann z. B. der Fall sein, wenn das BERT-Modell mit Daten aus einem anderen Projekt trainiert worden ist als mit den Daten, die aus dem zu klassifizierenden Projekt stammen. Die Ergebnisse zeigen, dass BERT-Modelle in der Klassifizierung von Daten aus neuen Projekten eine hohe Leistungsfähigkeit erreichen, wenn dass das Modell entweder mit aktivem Lernen trainiert worden ist, wobei eine bestimmte Anzahl der unbekannten Daten von einem Experten gekennzeichnet werden muss, oder die Datenbasis für das Training des Modells möglichst heterogen und ähnlich (zum Beispiel gleiche Domäne) der neuen Daten ausgewählt worden ist. Aufgrund der hohen Leistungsfähigkeit der Abhängigkeitsanalyse wird *Erfolgsfaktor 6 als erfüllt bewertet*.

Tabelle 7-9: Bewertung der Erfolgsfaktoren (EF) der Kategorie "Funktionalität"

EF	Bewertungskriterien	Erfüllungsgrad
3	• Makro-Durchschnitt F1 sowie der gewichtete Durchschnitt F1 der Augmentierung von Anforderungen: 64,71 % bzw. 62,74 %	teilweise erfüllt
4	• F1 der Extrahierung von Anforderungen beträgt 76,74 %	erfüllt
5	• Makro-Durchschnitt F1 der Formalisierung von Anforderungen: 86,55 %	erfüllt
6	• F1 der Abhängigkeitsanalyse von Anforderungen mit aktivem Lernen: 81,63 %	erfüllt

Anwendbarkeit

Erfolgsfaktor 7: „Das System muss so gestaltet sein, dass nur branchenüblich-verfügbare Informationen zur Anwendung benötigt werden.“

Der Erfolgsfaktor ist erfüllt, wenn die Anwendungspartner die benötigten Informationen als „branchenüblich verfügbar“ bewerten. Folgende Informationen werden für die Anwendung der Methoden benötigt:

- *Extrahierung von Anforderungen:* (branchenübliche) Anforderungsdaten, die nach Anforderungsart gekennzeichnet worden sind, für das Training der KI-Modelle
- *Formalisierung von Anforderungen:* Domänenlexika mit Schlüsseln, Werten und Regeln zur Formalisierung
- *Abhängigkeitsanalyse von Anforderungen:* gekennzeichnete Abhängigkeiten (branchenüblicher) Anforderungsbeschreibungen für das Training der KI-Modelle

- *Risikoberechnung von Anforderungsänderungen:* eine Änderungshistorie von Projekten (Anzahl Anforderungen und Anzahl an Anforderungsänderungen)
- *Anwendung der Methoden:* englische (branchenübliche) Lastenhefte im PDF-Format

Die Anwendungspartner bewerten die Informationen teilweise als branchenüblich verfügbar (siehe Tabelle 7-7). Gekennzeichnete Anforderungsdaten liegen üblicherweise aus vergangenen Projekten im Anforderungsmanagement Werkzeug vor. Natürlich-sprachliche Lastenhefte liegen zu Projektbeginn vor (ID 1, Bewertung: 1 in Tabelle 7-7). Die Sprache der Lastenhefte ist üblicherweise Englisch. Das Domänenlexikon liegt nicht vor und muss generiert werden. Die Industrievertreter geben an, dass aktuell das Wissen zur Erstellung eines Domänenlexikons im Unternehmen nur teilweise vorhanden ist (ID 2 & 3, Bewertung: 0,4, bzw. 0,2 in Tabelle 7-7). Nach einer Erklärung des Vorgehens zur Identifikation von Strukturregeln und Textindikatoren während der Interviews geben die Industrievertreter an, dass sie sich kurzfristig in der Lage sehen, das notwendige Wissen für die Definition von Strukturregeln und Textindikatoren aneignen können. Trainingsdaten für NER-Modelle sind nicht vorhanden und müssen vor Anwendung der Formalisierung von neu-definierten Schlüssel-Wert-Paaren erstellt werden.

Daten von Anforderungsabhängigkeiten für das initiale Training eines KI-Modells für die Abhängigkeitsanalyse von Anforderungen sind teilweise im Unternehmen vorhanden (ID 5, Bewertung: 0,4 in Tabelle 7-7). Die Industrievertreter geben an, dass Anforderungsabhängigkeiten häufig im Anforderungsmanagement-Werkzeug des Unternehmens definiert werden. Diese Daten müssen vorbereitet werden, damit sie für das Training eines KI-Modells verwendet werden können. Das betrifft die Formatierung der Daten im Anforderungsmanagement-Werkzeug in einer CSV. Zur Anwendung des Trainings-Algorithmus von Abhängigkeitsmodellen dieser Arbeit müssen die natürlich-sprachlichen Beschreibungen der Anforderungen hintereinander in einem String zusammengefügt werden und es muss die Kennzeichnung (funktional, nicht-funktional oder keine Anforderung, bzw. numerisch: 0-2) im zugehörigen Feld der CSV-Datei ergänzt werden. Es werden nicht für sämtliche Projekte die Anforderungsabhängigkeiten erfasst. Wenn Anforderungsabhängigkeiten von Projekten erkannt werden müssen, zu denen keine inhaltlich-ähnlichen Daten vorliegen, dann müssen Anforderungsabhängigkeiten manuell gekennzeichnet werden. Das Wissen zur Bewertung von Anforderungsabhängigkeiten ist nur teilweise vorhanden (ID 6, Bewertung: 0,4 in Tabelle 7-7). Die Industrievertreter geben an, dass projektspezifisch (bzw. in Abhängigkeit der Domäne) Experten zur Bewertung herangezogen werden müssen. Für die dargestellte Problematik ist die entwickelte Methode zur Abhängigkeitsanalyse mit aktivem Lernen besonders geeignet, weil die Daten mit der höchsten Unsicherheit in der Klassifizierung zum Kennzeichnen ausgewählt werden und damit die Anzahl an zu kennzeichnender Daten für eine hohe Leistungsfähigkeit verringert wird. Es wird der Aufwand der Experten zur Bewertung der Anforderungsabhängigkeiten verringert und Kosten reduziert.

Eine Änderungshistorie, welche die Anzahl an Anforderungen sowie die Anzahl an Anforderungsänderungen pro Projekt erfasst, liegt üblicherweise vor (ID 7, Bewertung: 0,6 in Tabelle 7-7). Die Industrievertreter geben an, dass diese Werte häufig nach Projektende nicht berechnet werden. Eine Berechnung der Kennzahlen nach Projektende ist mit der üblichen Projektdokumentation und der Änderungshistorie des Anforderungsmanagement-Werkzeugs möglich. Auswirkungen von Anforderungsänderungen können ebenfalls üblicherweise bewertet werden (ID 8, Bewertung: 0,6 in Tabelle 7-7). Die Industrievertreter geben an, dass das Aufzeigen der Anforderungsabhängigkeiten bei der Bewertung von Anforderungsabhängigkeiten hilfreich ist und zu einer präziseren Einschätzung des Risikos führt. Dies spiegelt auch die Bewertung des Mehrwerts der Abhängigkeitsanalyse von Anforderungen und der Risikoberechnung von Änderungen wider (Thesen 19 & 20, Median: 1). Eine wichtige Erkenntnis ist, dass die benötigten Informationen zur Berechnung der Änderungsrisiken vorhanden sind. Es wird keine ursachenspezifische Dokumentation von Anforderungsänderungen benötigt (siehe ProMaRC-Methodik in Anhang A2.3). Dieser Punkt wurde in der Entwicklung der ProMaRC-Methodik von Industrievertretern bemängelt (siehe Abschnitt 4.2). Durch die Weiterentwicklung der Berechnung des Änderungsrisikos von Anforderungen wurde die industrielle Anwendbarkeit der ursprünglichen Änderungsrisiko-Berechnung der ProMaRC-Methodik erhöht.

Erfolgsfaktor 7 ist teilweise erfüllt, da die Informationen, die zur Anwendung benötigt werden teilweise verfügbar sind. Es besteht aber die Möglichkeit diese Informationen zu generieren und die Methoden unterstützen den Anwender in der Generierung der benötigten Informationen (z. B. aktives Lernen zur Auswahl von zu kennzeichnenden Anforderungsabhängigkeiten).

Erfolgsfaktor 8: „Das System muss so gestaltet sein, dass die benötigten Informationen in der Projektinitierungsphase verfügbar sind.“

Der Erfolgsfaktor ist erfüllt, wenn die Anwendungspartner die benötigten Informationen als „früh verfügbar“ bewerten. Die benötigten Informationen der Methoden (siehe Erfolgsfaktor 7) werden teilweise als frühzeitig verfügbar von den Anwendungspartnern bewertet. Analog zu Erfolgsfaktor 7 müssen verschiedene Informationen generiert, bzw. aufbereitet werden. Es ist jedoch möglich, sämtliche benötigte Informationen in der Projektinitierungsphase zu generieren, sodass diese zum Projektstart vorhanden sind.
Erfolgsfaktor 8 ist teilweise erfüllt.

Erfolgsfaktor 9: „Das System muss so gestaltet sein, dass eine generische Anwendbarkeit in der Entwicklung komplexer technischer Systeme gegeben ist.“

Der Erfolgsfaktor ist erfüllt, wenn Anwendungspartner die Methodik als „generisch anwendbar“ bewerten. Außerdem müssen in der Validierung der Methoden verschiedene Produkte untersucht werden und die Leistungsfähigkeit hoch sein. Die Methodik wird teilweise als generisch anwendbar von den Anwendungspartnern bewertet (siehe Bild 7-10, These 11: Median = 0). Hinsichtlich der Anwendbarkeit für verschiedene Produkte

werden von den Anwendungspartnern keine Einschränkungen erkannt. Es müssen lediglich die Domänenlexika für verschiedene Produkte erstellt werden (falls Schlüssel-Wert-Paare erkannt werden sollen, für die keine Domänenlexika vorliegen). Voraussetzung ist, dass die notwendigen Informationen zur Anwendung der Methodik verfügbar sind.

In der Validierung der Methoden (siehe Abschnitte 7.1-7.3) wurden verschiedene Fallbeispiele gewählt: Montage Riegel Haube, Haubenanpassung Stöpsel, Kofferraumrollo, Knickarmroboter, elektrische Gewürzmühle und Toaster). Die Methoden konnten für die verschiedenen Fallbeispiele angewendet werden und die Leistungsfähigkeit ist jeweils hoch. *Erfolgsfaktor 9 ist erfüllt.*

Erfolgsfaktor 10: Das System muss so gestaltet sein, dass der Vorbereitungsaufwand akzeptabel ist.

Der Erfolgsfaktor ist erfüllt, wenn die Industriepartner den Vorbereitungsaufwand als „akzeptabel“ bewerten. Zur Vorbereitung der Methodik-Anwendung müssen die folgenden Aktivitäten durchgeführt werden:

- Training eines KI-Modells zur Extrahierung von Anforderungen
- Prüfen und ggf. Erstellen von Domänenlexika
- Zusammenstellen eines Trainingsdatensatzes für das initiale Training des BERT-Modells zur Abhängigkeitsanalyse von Anforderungen
- Zusammenstellen einer Änderungshistorie von Anforderungen vergleichbarer Projekte

Die Industrievertreter bewerten den Vorbereitungsaufwand teilweise als akzeptabel (siehe Bild 7-11, These 14: Median = 1). Bemängelt wird, dass der Aufwand zur Generierung der Domänenlexika hoch ist, weil dafür Anforderungsdokumente manuell analysiert werden müssen, um Strukturregeln und Textindikatoren zu definieren. Es wird positiv hervorgehoben, dass die künstliche Generierung von Anforderungsdaten für das Training eines KI-Modells Aufwand reduziert, der für die manuelle Kennzeichnung von Anforderungsdaten andernfalls anfallen würde. Abhängigkeitsdaten von Anforderungen sind häufig im Anforderungsmanagement-Werkzeug vorhanden, sodass nur die Vorbereitung der Daten erfolgen muss. Textuelle Anforderungsabhängigkeiten sowie deren Abhängigkeiten müssen in eine CSV-Datei eingefügt werden. Das Zusammenstellen der Änderungshistorie von Anforderungen vergleichbarer Projekte ist nicht aufwändig. Diese Daten liegen üblicherweise vor. *Erfolgsfaktor 10 ist teilweise erfüllt.*

Erfolgsfaktor 11: „Das System muss so gestaltet sein, dass der Anwendungsaufwand akzeptabel ist.“

Der Erfolgsfaktor ist erfüllt, wenn die Industriepartner den Anwendungsaufwand als „akzeptabel“ bewerten. Dabei muss die weitere Zeitersparnis im Projekt durch die Anwendung der Methodik berücksichtigt werden. Für die Validierung der Methoden wurden

in Fallbeispielen jeweils mehrere hundert Anforderungen verwendet (siehe Abschnitte 7.1-7.3). Die Extrahierung von Anforderungen erfolgt vollautomatisiert. Die Formalisierung von Anforderungen erfolgt ebenfalls vollautomatisiert. Jedoch müssen laut den Anwendungspartnern Konflikte in den erkannten Schlüssel-Wert-Paaren aufgelöst werden, wenn z. B. mehrere Werte für einen Schlüssel erkannt worden sind. In der Abhängigkeitsanalyse von Anforderungen muss ein Experte Feedback geben, um eine Teil-Menge der zu klassifizierenden Anforderungsabhängigkeiten zu beschriften. Die restlichen Anforderungsabhängigkeiten werden vollautomatisiert durch ein KI-Modell klassifiziert. Die Anwendungspartner bewerten den Anwendungsaufwand als „akzeptabel“ (siehe Bild 7-11, These 15: Median = 2). Sie geben während des Interviews nach dem Ausfüllen der Bewertungsbögen an, dass durch die Methodik-Anwendung Aufwand in der Extrahierung und Formalisierung von Anforderungen reduziert wird. Durch die Anwendung des Risikomanagements von Anforderungsänderungen wird in nachgelagerten Phasen des Entwicklungsprojekts Aufwand reduziert, indem Anforderungsänderungen besser gehandhabt werden können. *Erfolgsfaktor 11 ist erfüllt.*

Erfolgsfaktor 12: „Die Methodik muss so gestaltet sein, dass sie in das Vorgehen einer Methodik zur Entwicklung komplexer technischer Systeme integriert werden kann.“

Der Erfolgsfaktor ist erfüllt, wenn die Integration der Methoden in eine Entwicklungs-methodik für komplexe technische Systeme vom Anwendungspartner als „geeignet“ bewertet wird. Außerdem muss die Methodik als „einfach anwendbar bewertet“ werden. Durch die einfache Anwendung wird die Akzeptanz der Methodik in der industriellen Praxis erhöht. Der Referenzprozess zur Methodik Anwendung wurde den Anwendungspartnern vorgestellt (siehe Abschnitt 6.1.2). Die Potenziale der Methodik-Anwendung wurden im Kontext der VDI/VDE-Richtlinie 2206 erläutert. Die Potenziale der Methoden sind in den folgenden Kernaufgaben, bzw. Strängen des V-Modells relevant:

- Extrahierung von Anforderungen: Anforderungserhebung
- Formalisierung von Anforderungen: Modellierung & Analyse
- Abhängigkeitsanalyse von Anforderungen: Modellierung & Analyse
- Risikoberechnung von Anforderungsänderungen: Anforderungsmanagement

Die Anwendungspartner bewerten die Methodik als „gut“ integrierbar in unternehmens-interne Entwicklungsprozesse (siehe Bild 7-10, These 12). Die Industrievertreter geben an, dass sie das Software-Werkzeug regelmäßig nutzen würden (These 4: Median = 2). Die Bedienung ist für die Industrievertreter intuitiv. Der Funktionsumfang ist auf die Kernfunktionalitäten der Methodik beschränkt (These 5: Median = 2). Es wird voraussichtlich technischer Support benötigt, um die Software zu nutzen (These 6: Median: 0). Die Industrievertreter geben an, dass sie zur Bereitstellung der Daten zur Anwendung des Werkzeugs (vor allem für das Training von KI-Modellen) sich mit Data-Scientists austauschen müssen. Die Anwendung des Werkzeugs (sofern die Daten verfügbar sind) ist ohne technischen Support möglich. Im Software-Werkzeug sind keine Inkonsistenzen

aufgefallen (These 7: Median = 2). Thesen 4-9 sind einem standardisierten „System Usability Scale“ (SUS-)Bewertungsbogen entnommen, um die Nutzerfreundlichkeit des Systems zu messen [Bro95]. Der SUS-Score ist eine Kennzahl zur Messung der Nutzerfreundlichkeit. Der SUS-Score wird für die einzelnen Bewertungsbögen berechnet, indem die Antworten der Unternehmensvertreter gewichtet werden („stimme nicht zu“ = 1 bis „stimme zu“ = 4). Für positiv formulierte Thesen wird der Wert der Antwort mit 1 subtrahiert. Für negativ formulierte Thesen wird der Wert der Antwort von 5 subtrahiert. Die Werte der einzelnen Thesen werden summiert und mit 2,5 multipliziert, um den SUS-Score zu erhalten. Das arithmetische Mittel der SUS-Scores beträgt 85,83. Damit liegt der SUS-Score über einem Schwellwert von 85 und die Nutzerfreundlichkeit wird als „exzellent“ bezeichnet [BKM09]. Aufgrund der guten Integrierbarkeit der Methodik in den Entwicklungsprozess von Unternehmen und dem exzellenten SUS-Score ist *Erfolgsfaktor 12* erfüllt.

Tabelle 7-10: Bewertung der Erfolgsfaktoren (EF) der Kategorie "Anwendbarkeit"

EF	Bewertungskriterien	Erfüllungsgrad
7	• Ergebnisse aus Tabelle 7-7 und Interviews	teilweise erfüllt
8	• Ergebnisse aus Tabelle 7-7 und Interviews	teilweise erfüllt
9	• These 11: Median = 0 und hohe Leistungsfähigkeit der Methoden für verschiedene Fallbeispiele	erfüllt
10	• These 14: Median = 0	teilweise erfüllt
11	• These 15: Median = 2	erfüllt
12	• SUS-Score: exzellent	erfüllt

Output

Erfolgsfaktor 13: „Die Ergebnisse des Systems müssen in einem gängigen Austauschformat ausgegeben werden, sodass diese in branchenübliche Anforderungsmanagement Software eingelesen werden können.“

Der Erfolgsfaktor ist erfüllt, wenn die Ergebnisse (extrahierte und formalisierte Anforderungen sowie deren Abhängigkeiten) in ein branchenübliches Anforderungsmanagement Werkzeug eingelesen werden können. Konkret wird in diesem Werk IBM DOORS als branchenübliches Anforderungsmanagement Werkzeug betrachtet. Die Ergebnisse der Methoden sind die Folgenden:

- Extrihierte und klassifizierte Anforderungen
- Anforderungsabhängigkeiten
- Risiko von Anforderungsänderungen

Die extrahierten und klassifizierten Anforderungen besitzen die Attribute „ID“, „textuelle Beschreibung“ und „Anforderungsart“. Die Anforderungsabhängigkeiten werden binär

als abhängig, bzw. nicht-abhängig angegeben. Die Attribute sind: „ID“, „ID Anforderung 1“, „ID Anforderung 2“, „Abhängigkeit“. Das Risiko von Anforderungsänderungen werden den jeweiligen Anforderungen als Attribut übergeben und als Dezimal-Zahl zwischen 0 und 1 angegeben. Die Informationen werden im CSV-Format ausgegeben und ein Import in Anforderungsmanagement Software ist möglich. *Erfolgsfaktor 13 ist erfüllt.*

Tabelle 7-11: Bewertung der Erfolgsfaktoren (EF) der Kategorie "Output"

EF	Bewertungskriterien	Erfüllungsgrad
13	<ul style="list-style-type: none"> Import der Ergebnisse der Methodik als CSV-Datei in IBM DOORS ist möglich 	erfüllt

7.5 Diskussion

Die Validierungsergebnisse zeigen, dass die Methodik 9 Erfolgsfaktoren vollständig und 4 Erfolgsfaktoren teilweise erfüllt (siehe Tabelle 7-12). Input der Methodik sind branchenübliche englische Lastenhefte im PDF-Format (EF 1 & EF 2). Aus diesen Lastenheften werden Anforderungen mit einer hohen Leistungsfähigkeit automatisiert mittels KI-Modellen extrahiert (EF 4). Zur aufwandsarmen Generierung von Anforderungsdaten wird GPT-J eingesetzt, um künstliche Anforderungsdaten zu generieren. Die generierten Anforderungsdaten der Klasse „Anforderung“ müssen einer Qualitätsprüfung unterzogen werden (EF 3). Extrahierte Anforderungen werden mit einer hohen Leistungsfähigkeit formalisiert (EF 5). Anforderungsabhängigkeiten werden mit einer hohen Leistungsfähigkeit erkannt (EF 6). Die Informationen zur Anwendung der Methodik sind teilweise in Unternehmen vorhanden. Domänenlexika und Anforderungsabhängigkeitsdaten müssen erstellt werden. Das Wissen zur Erstellung ist in Unternehmen vorhanden (EF 7). Die benötigten Informationen sind in der Projektinitiierungsphase vorhanden, müssen aber teilweise erstellt werden: Domänenlexika und Anforderungsabhängigkeitsdaten (EF 8). Die Methodik ist generisch für komplexe technische Systeme anwendbar (EF 9). Der Vorbereitungsaufwand der Methodik ist teilweise akzeptabel. Der Aufwand zur Generierung von Domänenlexika und zur Aufbereitung von Anforderungsabhängigkeitsdaten ist hoch (EF 10). Der Anwendungsaufwand der Methodik ist akzeptabel unter der Berücksichtigung der Aufwandsreduzierung im Projektverlauf (EF 11). Die Methodik kann in unternehmensspezifische Entwicklungsprozesse integriert werden und bietet Potenzial in der Methodik-Anwendung (EF 12). Die Ergebnisse der Methodik können mittels CSV-Import in branchenübliche Anforderungsmanagement-Software (wie IBM-DOORS) eingelesen werden (EF 13).

Zur Erfüllung der Erfolgsfaktoren hat das auf die Anwendung ausgerichtete wissenschaftliche Vorgehen beigetragen. Die kontinuierliche Einbindung von Industrieanwendern in die Entwicklung der Methodik hat eine Fokussierung auf praxisrelevante Aspekte ermöglicht und damit wesentlich zur fortwährenden Prüfung und Optimierung der Methodik beigetragen. Die Feldstudie hat Bedarfe der industriellen Praxis für die

Abhängigkeitsanalyse sowie für die Risikoberechnung von Anforderungsänderungen offen gelegt. Es konnte ein wichtiger Beitrag zur Wissenschaft und Praxis geleistet werden, der im Folgenden anhand der Forschungsfragen (siehe Abschnitt 5.1) erläutert wird.

Tabelle 7-12: Erfüllungsgrad der Erfolgsfaktoren

ID	Kategorie	Erfolgsfaktor	Erfüllungsgrad
1	Input	Verarbeitung branchenüblicher Lastenhefte	vollständig
2	Input	Verarbeitung englischsprachiger Anforderungen	vollständig
3	Funktionalität	Generierung von Anforderungsdaten	teilweise
4	Funktionalität	Klassifizierung von Anforderungen	vollständig
5	Funktionalität	Formalisierung von Anforderungen	vollständig
6	Funktionalität	Abhängigkeitsanalyse von Anforderungen	vollständig
7	Anwendbarkeit	Verfügbarkeit benötigter Informationen	teilweise
8	Anwendbarkeit	Frühzeitige Anwendbarkeit	teilweise
9	Anwendbarkeit	Generische Anwendbarkeit	vollständig
10	Anwendbarkeit	Akzeptabler Vorbereitungsaufwand	teilweise
11	Anwendbarkeit	Akzeptabler Anwendungsaufwand	vollständig
12	Anwendbarkeit	Integrierbarkeit in den Entwicklungsprozess	vollständig
13	Output	Gängiges Austauschformat	vollständig

Beiträge der Methodik zur Wissenschaft und Praxis

FF 1: „Wie hoch ist die Leistungsfähigkeit der Daten-Augmentation von Anforderungen durch künstliche Intelligenz?“

Es wurde mittels GPT-J ein F1 von 62,42 % in der Daten-Augmentation von Anforderungen erreicht. Für die industrielle Praxis bedeutet das, dass eine Qualitätskontrolle von Daten der Klasse „Anforderungen“ durchgeführt werden muss, um korrekte Kennzeichnungen zu vergeben und somit Trainingsdaten für Modelle zur Klassifizierung von Anforderungen zu generieren. Durch die Daten-Augmentation von Anforderungen wird in diesem Kontext der Aufwand in der Generierung von Trainingsdaten in der industriellen Praxis reduziert und die Anwendbarkeit von KI-Modellen zur automatisierten Klassifizierung von Anforderungen erhöht.

FF 2: „Wie hoch ist die Leistungsfähigkeit von Klassifizierungsmodellen, die augmentedierte Daten verwenden, im Vergleich zu Modellen, die auf realen Daten basieren?“

Der primäre Anwendungsfall von KI-Modellen in der Praxis ist die Klassifizierung von unbekannten Daten aus neuen Projekten. Da besonders die industrielle Anwendbarkeit

der Klassifizierung von Anforderungen durch die Anwendung augmentierter Daten geprüft werden soll, ist diese Untersuchung besonders wichtig für die Bewertung der Leistungsfähigkeit der Modelle. Das Training mit augmentierten Daten führt zu einer Leistungssteigerung bei der Untersuchung der Leistungsfähigkeit von Modellen für unbekannte, heterogene Anforderungsdaten – F1 Werte steigen zwischen 0,67 % und 4,65 %. Dies bedeutet, dass die Leistungsfähigkeit für die Klassifizierung von Daten aus neuen Projekten steigt, wenn Modelle mit augmentierten Daten trainiert werden. Die Untersuchungen zeigen deshalb, dass die Verwendung augmentierter Daten die Leistungsfähigkeit von KI-Modellen zur Extrahierung von Anforderungen verbessert. Insgesamt verbessert die entwickelte Methode die Anwendbarkeit der Anforderungsextraktion in der industriellen Praxis. Während der gesamten Untersuchungen lieferte das Modell der Stützvektormaschine (SVM) die besten Ergebnisse bei der Klassifizierung von Anforderungen. In diesen Untersuchungen wurden augmentierte Daten aus der Softwareentwicklung verwendet. Die Ergebnisse zeigen, dass die Verwendung von augmentierten Daten zum Training aus einer anderen Domäne als die Testdaten zu einer geringeren Leistungsfähigkeit führt.

FF 3: „Welche Ansätze sind für die Formalisierung von Anforderungen im Anwendungskontext der generativen Entwicklung anwendbar?“

Es wurden die Ansätze „String Matching“, „Zahl & Einheit“ sowie NER-Modelle (vortrainierte spaCy Modelle sowie eigene trainierte Modelle) untersucht. Aufgrund der vielfältigen Ansätze ist die Methode auf verschiedene Arten von Parametern übertragbar. Der Aufwand für die Formalisierung von Anforderungen wird in der industriellen Praxis durch diesen automatisierten Ansatz reduziert.

FF 4: „Wie hoch ist die Leistungsfähigkeit von Ansätzen zur Formalisierung von Anforderungen im Kontext der generativen Entwicklung?“

Die Validierungsergebnisse zeigen, dass die Ansätze zur Formalisierung von Anforderungen Schlüssel-Wert-Paare für die generative Entwicklung mit hoher Leistungsfähigkeit extrahieren (Makro-Durchschnitt F1 von 86,55 %). Insbesondere die Präzision ist hoch (99,26 %), was bedeutet, dass kaum falsch-positive Ergebnisse gefunden wurden. Für die industrielle Praxis bedeutet das, dass durch die Formalisierung Anforderungen maschinenlesbar werden. Es können verschiedene Schlüssel-Wert-Paare in Entwicklungswerkzeugen (z. B. Synera für die generative Entwicklung) aufwandsarm eingelesen werden.

FF 5: „Wie hoch ist die Leistungsfähigkeit von BERT-Modellen in der Abhängigkeitsanalyse von Anforderungen, welche mit aktivem Lernen trainiert worden sind?“

Durch das aktive Lernen nimmt die Leistungsfähigkeit von BERT-Modellen in der Klassifizierung von Daten aus neuen Projekten deutlich zu. Besonders für den Fall einer homogenen Trainingsdatenbasis (Anforderungsdaten von nur einem Produkt – Untersuchung 1) wurde durch das aktive Lernen nach 10 Iterationen eine Steigerung des F1

von +53,76 % erzielt. Die Ergebnisse zeigen, dass BERT-Modelle in der Klassifizierung von Daten aus neuen Projekten eine hohe Leistungsfähigkeit erreichen, wenn das Modell entweder mit aktivem Lernen trainiert worden ist, wobei eine bestimmte Anzahl der unbekannten Daten von einem Experten gekennzeichnet werden muss, oder die Datenbasis für das Training des Modells möglichst heterogen und ähnlich (zum Beispiel gleiche Domäne) der neuen Daten ausgewählt worden ist. Durch das aktive Lernen ist das effiziente Kennzeichnen von Anforderungsdaten möglich und die praktische Anwendbarkeit der Abhängigkeitsanalyse wird erhöht.

FF 6: „Wie muss eine Berechnung des Änderungsrisikos von Anforderungen definiert sein, damit die benötigten Informationen zur Anwendung der Berechnung in der industriellen Praxis verfügbar sind?“

Für die Risikoberechnung von Anforderungsänderungen wurde auf der Risikoberechnung der ProMaRC-Methodik aufgebaut. Es wurde die Berechnungslogik der Änderungswahrscheinlichkeit angepasst, sodass Informationen zu Ursachen von Anforderungsänderungen zur Anwendung der Methode nicht benötigt werden. Die Validierungsresultate haben gezeigt, dass die Risikoberechnung von Anforderungsänderungen industriell anwendbar ist. Es werden lediglich Daten für die Anwendung der Methode benötigt, die branchenüblich verfügbar sind oder frühzeitig generiert werden können.

Forschungsbedarfe der Methodik

Weiterer Forschungsbedarf besteht für die *Extrahierung von Anforderungen* in der Festlegung der Hyperparameter von KI-Modellen zur Augmentation von Daten speziell für Anforderungsdaten, um die Leistungsfähigkeit zu verbessern. Dies führt zu einer weiteren Verbesserung der Effizienz durch die Verringerung des Aufwands bei der Kennzeichnung der augmentierten Anforderungen. Darüber hinaus besteht weiteres Forschungspotenzial in der Verwendung von domänen-spezifischen Daten zur Untersuchung der Leistungsfähigkeit in der Klassifizierung unbekannter, heterogener Daten aus neuen Projekten.

Für die *Formalisierung von Anforderungen* für die generative Entwicklung besteht Forschungsbedarf in der Entwicklung einer Schnittstelle für ein generatives Entwicklungs-Werkzeug. Die durch die Methode extrahierten Parameter können im CSV-Format exportiert werden, so dass sie nach Implementierung einer Schnittstelle in generative Entwicklungs-Werkzeuge aufwandsarm importiert werden können und so weiterer manueller Aufwand in der generativen Entwicklung reduziert werden kann.

In der *Abhängigkeitsanalyse von Anforderungen* besteht weiterer Forschungsbedarf in der Erweiterung des Ansatzes für formalisierte Anforderungen, um die Leistungsfähigkeit der Methode zu erhöhen. Es können Anforderungen analog zu Schablonen wie beispielsweise der SOPHISTen definiert werden. Die Informationen der Anforderungsbestandteile können genutzt werden, um auf Abhängigkeiten zu schließen.

Für die *Berechnung der Änderungsrisiken von Anforderungen* besteht Forschungsbedarf in der Anwendung der Methode innerhalb eines vollständigen Projekts in der Industrie. Es muss beim Abschluss des Projekts geprüft werden, ob das Änderungsrisiko von Anforderungen zu Beginn des Projekts korrekt bewertet worden ist. Anhand der Ergebnisse muss untersucht werden, ob bestimmte festgelegte Parameter aus Annahmen – wie der Anteil an exogenen zu endogenen Anforderungsänderungen – angepasst werden müssen.

Limitationen der Validierungsergebnisse

Die Aussagekraft der Validierungsergebnisse ist hinsichtlich der *Anwendungsdomäne* limitiert. Innerhalb der Validierung wurden die Methoden mittels Daten aus Projekten, bzw. Fallbeispielen der Automobilindustrie angewendet und bewertet. Die Rahmenbedingungen innerhalb dieser Domäne sind repräsentativ für die Entwicklung komplexer technischer Systeme, weshalb sich die Gültigkeit der Ergebnisse auf ähnliche Branchen übertragen lässt – z. B. Luft- und Raumfahrt oder Maschinen- und Anlagenbau. Die generische Anwendbarkeit der Methodik wurde durch Experteneinschätzungen überprüft (siehe Bild 7-10, These 10).

Die Aussagekraft der Validierungsergebnisse zum *Erfolg der Methodik-Anwendung* unterliegt ebenfalls Limitationen. Es wurden Daten aus historischen Projekten, bzw. Fallbeispielen für die Validierung verwendet. Die Validierungsergebnisse wurden mit Experten aus der Anwendungsdomäne diskutiert und bewertet. Die Anwendung der Methoden muss in laufenden Projekten durch Industrieunternehmen erfolgen, um die Aussagekraft der Validierungsergebnisse weiter zu unterstützen.

8 Zusammenfassung und Ausblick

Hauptgründe für das Scheitern von Entwicklungsprojekten sind unvollständige, mehrdeutige Anforderungslisten und Anforderungsänderungen. Die manuelle Analyse von natürlich-sprachlichen Spezifikationsdokumenten (zur Erhöhung der Vollständigkeit) ist aufwändig, fehleranfällig und führt zu hohen Kosten. Natürliche Sprache ist inhärent mehrdeutig. Es ist für Software-Werkzeuge nicht möglich, diese direkt zu interpretieren. Es besteht ein Bedarf nach einer automatisierten Extrahierung von Anforderungen, um Aufwände und Kosten in der Analyse natürlich-sprachlicher Spezifikationsdokumente zu reduzieren. Außerdem müssen Anforderungen formalisiert werden, um diese für verschiedene Entwicklungswerkzeuge (z. B. Synera im Kontext der generativen Entwicklung) interpretierbar zu machen. KI-basierte Ansätze zur Verarbeitung natürlicher Sprache bieten in diesem Anwendungskontext ein hohes Effizienzpotenzial. Defizite bestehen aktuell in der Verfügbarkeit von Trainingsdaten, um leistungsfähige KI-Modelle zu erstellen. Zur Handhabung von Anforderungsänderungen wird ein Risikomanagement benötigt. Um das Änderungsrisiko von Anforderungen bewerten zu können, müssen Anforderungsabhängigkeiten bekannt sein. Aufgrund der hohen Anzahl an Anforderungen an komplexe technische Systeme wird ein automatisiertes Vorgehen benötigt. Bestehende KI-basierte Ansätze zeigen Defizite in der Leistungsfähigkeit für die Erkennung von Anforderungsabhängigkeiten unbekannter Anforderungsdaten.

Um diesen Defiziten entgegenzuwirken, wird eine Methodik zur automatisierten Extrahierung, Formalisierung und Handhabung von Änderungen technischer Anforderungen entwickelt. Das wissenschaftliche Vorgehen wird in Anlehnung an ULRICH gewählt und die Methodik wird praxisnah realisiert. Die Methodik besteht aus den Methoden: Extrahierung von Anforderungen, Formalisierung von Anforderungen, Abhängigkeitsanalyse von Anforderungen sowie Risikoberechnung von Anforderungsänderungen. Die entwickelte Methodik basiert auf eigenen Vorarbeiten in zwei öffentlich geförderten und einem bilateralen Forschungsprojekt. An der Validierung waren neben Studenten Entwicklungsdienstleister der Automobilindustrie (IAV GmbH & EDAG Engineering GmbH) sowie ein weltweit führender Automobilhersteller beteiligt.

Zur Konkretisierung des Forschungsbedarfs und zur Ableitung von Erfolgsfaktoren der Methodik wurde eine systematische Literaturrecherche nach MACHI und MCEVOY durchgeführt. Außerdem wurden Ergebnisse einer Feldstudie analysiert. Die Analysen zeigen für die Extrahierung von Anforderungen, dass die Leistungsfähigkeit in der industriellen Praxis erhöht werden muss. Potenzial bieten Ansätze zur Daten-Augmentation. Es wurde eine Methode entwickelt, um mittels GPT-J künstliche Anforderungsdaten zu generieren.

Zur Formalisierung müssen speziell für den Anwendungsfall geeignete Ansätze ausgewählt und implementiert werden. Es wird die generative Entwicklung von Bauteilen betrachtet. Es wurde eine Methode entwickelt, um aufbauend auf einem Domänenlexikon

Schlüssel-Wert-Paare aus Anforderungen zu extrahieren. Innerhalb des Domänenlexikons werden Strukturregeln und Textindikatoren pro Schlüssel-Wert-Paar festgelegt. Außerdem werden Ansätze zur Formalisierung festgelegt, z. B. Named-Entity-Recognition (NER) oder String-Matching.

Die Abhängigkeitsanalyse von Anforderungen hat Defizite in der Klassifizierung von Anforderungsabhängigkeiten heterogener Daten (z. B. unbekannte Daten aus neuen Projekten). Potenzial zur Erhöhung der Leistungsfähigkeit besteht durch aktives Lernen. Diese Lernform hat sich bewährt, um aufwandsarm unbekannte Daten zu kennzeichnen. Es wurde eine Methode entwickelt, um mittels BERT-Modellen, die durch aktives Lernen trainiert worden sind, Anforderungsabhängigkeiten zu erkennen.

Für die Anwendung von Ansätzen des Risikomanagements von Anforderungsänderungen in der industriellen Praxis müssen die notwendigen Informationen vorliegen. Hierzu muss eine Möglichkeit entwickelt werden, dass das Änderungsrisiko entsprechend berechnet werden kann. Es wurde eine Methode entwickelt, um das Änderungsrisiko im Vergleich zur ProMaRC-Methodik so zu berechnen, dass Anforderungsänderungen nicht nach Änderungsursache aufgeschlüsselt werden müssen.

Die Methoden werden in einem Software-Werkzeug in Python implementiert. Es wird die Leistungsfähigkeit der Methoden in mehreren Fallbeispielen aus dem Automobilbereich bewertet. Künstliche Anforderungsdaten erhöhen die Leistungsfähigkeit von KI-Modellen in der Extrahierung von Anforderungen aus neuen Projekten. Die Leistungsfähigkeit in der Formalisierung von Anforderungen ist hoch: F1 ist größer als 80 %. Durch das aktive Lernen nimmt die Leistungsfähigkeit von KI-Modellen zur Abhängigkeitsanalyse deutlich zu. Es konnte eine Steigerung des F1 von mehr als 30 % erzielt werden. Außerdem wird eine Gesamt-Validierung durchgeführt: die o. g. Entwicklungsdienstleister bewerten die entwickelte Methodik. Die Validierungsergebnisse bestätigen den hohen Mehrwert der Methodik-Anwendung.

Durch die entwickelte Methodik werden Ursachen für das Scheitern von Entwicklungsvorprojekten adressiert. KI-basierte Ansätze werden als „Enabler“ für vollständige und eindeutige Anforderungslisten genutzt. Außerdem wird ein Mehrwert für das Risikomanagement von Anforderungsänderungen generiert, indem Anforderungsabhängigkeiten aufwandsarm erkannt werden und die Berechnung von Änderungsrisiken in der Praxis ermöglicht wird.

Weiterer Forschungsbedarf besteht in der Analyse der Übertragbarkeit der Formalisierung von Anforderungen. Im Rahmen dieser Arbeit wurde als Anwendungsfall die generative Entwicklung von Bauteilen untersucht. Ein weiterer Anwendungsfall ist beispielsweise die Wirkkettenmodellierung. Zum Aufbau eines Systemmodells eines Automobils und zur effizienten Modellierung verschiedener Parameter ist die Formalisierung von Anforderungen relevant. Hierzu muss das Domänenlexikon für die verschiedenen Parameter erweitert werden. Es muss untersucht werden, welche Ansätze für die Formalisierung der verschiedenen Parameter geeignet sind.

9 Literaturverzeichnis

- [AAV21] ARSLAN, Y.; ALLIX, K.; VEIBER, L.; LOTHRITZ, C.; BISSYANDÉ, T. F.; KLEIN, J.; GOUJON, A.: A Comparison of Pre-Trained Language Models for Multi-Class Text Classification in the Financial Domain. In: Leskovec, J.: Companion Proceedings of the Web Conference 2021. Association for Computing Machinery, New York, NY, United States, 2021; S. 260–268.
- [AF09] Abramovici, M.; Fathi, M.: Erweiterung des Product Lifecycle Managements durch wissensbasierte Rückführung von Produktnutzungsinformationen in die Produktentwicklung (WiRPro). Förderung von 2009 bis 2013. DFG-Sachbeihilfe. <https://gepris.dfg.de/gepris/projekt/75855839>.
- [AG06] AMBRIOLA, V.; GERVASI, V.: On the Systematic Analysis of Natural Language Requirements with C IRCE. In Automated Software Engineering, 13 2006 1; S. 107–167.
- [AIV20] AMIN-NEJAD, A.; IVE, J.; VELUPILLAI, S.: Exploring Transformer Text Generation for Medical Dataset Augmentation. In Proceedings of the 12th Language Resources and Evaluation Conference 2020; S. 4699–4708.
- [AK09] Ayre, L.; Kress, L.: Software Requirements Specification for the Global Requirements of an Integrated Library System. https://galecia.com/sites/default/files/project_files/COMPLETEL_SET_-_KCLS_ILS_Requirements_0.pdf, 13.05.2022.
- [Alp22] ALPAYDIN, E.: Maschinelles Lernen. De Gruyter Oldenbourg, München, 2022.
- [Alt12] ALT, O.: Modellbasierte Systementwicklung mit SysML. Carl Hanser Fachbuchverlag, München, 2012.
- [AMB15] ARORA, C. S.; M.; BRIAND, L.; ZIMMER, F.: Automated Checking of Conformance to Requirements Templates Using Natural Language Processing. In IEEE Transactions on Software Engineering, 41 2015 10; S. 944–968.
- [ANS08] ABADI, A.; NISENSON, M.; SIMIONOVICI, Y.: A Traceability Technique for Specifications: 16th IEEE International Conference, 2008; S. 103–112.
- [Are08] Arenou, F.: DU438 Software Requirements Specification: Simulated Test Data for Non Single Stars processing. <https://wwwhip.obspm.fr/~arenou/PS-papers/gaia/GAIA-C4-SP-OPM-FA-051.pdf>, 13.05.2022.
- [ASF18] ATAS, M.; SAMER, R.; FELFERNIG, A.: Automated Identification of Type-Specific Dependencies between Requirements: IEEE/WIC/ACM

- International Conference on Web Intelligence (WI). IEEE, 2018; S. 688–695.
- [ASN19] ARORA, C.; SABETZADEH, M.; NEJATI, S.; BRIAND, L.: An Active Learning Approach for Improving the Accuracy of Automated Domain Model Extraction. In ACM Transactions on Software Engineering and Methodology, 28 2019 1; S. 1–34.
- [Ath09] Athanasios-Alexandros, L.: Software Requirements Specification for PeaZip. <https://peazip.github.io/documentation/SRS-PeaZip-2.7.1-EN.pdf>, 21.03.2023.
- [Bal10] BALZERT, H.: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Spektrum Akademischer Verlag, Heidelberg, 2010.
- [Bet22] Betti, M. J.: Determiners and Pronouns in English. https://www.researchgate.net/publication/357910279_Determiners_and_Pro_nouns_in_English.
- [BG21] BENDER, B.; GERICKE, K.: Pahl/Beitz Konstruktionslehre – Methoden und Anwendung erfolgreicher Produktentwicklung. Springer Vieweg, Berlin Heidelberg, 2021.
- [BH11] BOUTKOVA, E.; HOUDEK, F.: Semi-automatic identification of features in requirement specifications: 19th IEEE International Requirements Engineering Conference (RE), 2011. IEEE, Piscataway, NJ, 2011; S. 313–318.
- [KKB21] Bayer, M.; Kaufhold, M.-A.; Buchhold, B.; Keller, M.; Dallmeyer, J.; Reuter, C.: Data Augmentation in Natural Language Processing: A Novel Text Generation Approach for Long and Short Text Classifiers. <https://arxiv.org/pdf/2103.14453>, 21.03.2023.
- [BKM09] BANGOR, A.; KORTUM, P.; MILLER, J.: Determining what individual SUS scores mean: Adding an adjective rating scale. In Journal of Usability Studies 2009; S. 114–123.
- [BKR21] Bayer, M.; Kaufhold, M.-A.; Reuter, C.: A Survey on Data Augmentation for Text Classification. <https://arxiv.org/pdf/2107.03158>, 21.03.2023.
- [BL14] BENNETT, N.; LEMOINE, G. J.: What a difference a word makes: Understanding threats to performance in a VUCA world. In Business Horizons, 57 2014 3; S. 311–317.
- [Bro95] BROOKE, J.: SUS: A quick and dirty usability scale, 1995.
- [Bru87] BRUNDTLAND, G. H.: Report of the World Commission on Environment and Development. Our Common Future, Oslo, 1987.

- [CGC10] CASAMAYOR, A.; GODOY, D.; CAMPO, M.: Identification of non-functional requirements in textual specifications: A semi-supervised learning approach. In *Information and Software Technology*, 52 2010 4; S. 436–445.
- [CKZ04] CECCATO, M.; KIYAVITSAYA, N.; ZENI, N.; MICH, L.; BERRY, D. M.: Ambiguity Identification and Measurement in Natural Language Texts, 2004.
- [Cla23] Clatronic: Elektrische Pfeffermühle/Salzmühle/Gewürzmühle PSM 3004. https://m.media-amazon.com/images/G/03/apparel/rcxgs/tile._CB483369910_.gif, 12.02.2023.
- [CSE04] CLARKSON, P.; SIMONS, C.; ECKERT, C.: Predicting change propagation in complex design. In *Journal of Mechanical Design* 2004 126; S. 788–797.
- [CSL01] CARLSHAMRE, P.; SANDAHL, K.; LINDVALL, M.; REGNELL, B.; NATT OCH DAG, J.: An industrial survey of requirements interdependencies in software product release planning: Proceedings Fifth IEEE International Symposium on Requirements Engineering. IEEE, Piscataway, NJ, USA, 2001; S. 84–91.
- [DAR19] DESHPANDE, G.; ARORA, C.; RUHE, G.: Data-Driven Elicitation and Optimization of Dependencies between Requirements. In: Damian, D.; Perini, A.; Lee, S.-W.: 2019 IEEE 27th International Requirements Engineering Conference. IEEE, Piscataway, NJ, 2019; S. 416–421.
- [Del14] DELIGATTI, L.: SysML distilled – A brief guide to the systems modeling language. Addison-Wesley, Upper Saddle River, NJ, 2014.
- [Die13] DIEDERICHS, M.: Risikomanagement und Risikocontrolling. Vahlen, München, 2013.
- [DIN18] DIN ISO 31000:2018: Risikomanagement - Leitlinien. Beuth Verlag GmbH, Berlin.
- [DMP20] DESHPANDE, G.; MOTGER, Q.; PALOMARES, C.; KAMRA, I.; BIESIALSKA, K.; FRANCH, X.; RUHE, G.; HO, J.: Requirements Dependency Extraction by Integrating Active Learning with Ontology-Based Retrieval: IEEE 28th International Requirements Engineering Conference (RE), 2020; S. 78–89.
- [DP03] DAHLSTEDT, Å. G.; PERSSON, A.: Requirements Interdependencies - Moulding the State of Research into a Research Agenda. In: Salinesi, C.; Regnell, B.; Kamsties, E.: Ninth International Workshop on Requirements Engineering: Foundation for Software Quality. Universitätsbibliothek Essen, Essen, 2003; S. 71–80.
- [DPA18] DHINAKARAN, V. T.; PULLE, R.; AJMERI, N.; MURUKANNAIAH, P. K.: App Review Analysis Via Active Learning: Reducing Supervision Effort without

- Compromising Classification Accuracy. In Proceedings of the 26th IEEE International Requirements Engineering Conference (RE) 2018; S. 170–181.
- [DRO13] DI THOMMAZO, A.; RIBEIRO, T.; OLIVATTO, G.; WERNECK, V.; FABBRI, S.: An Automatic Approach to Detect Traceability Links Using Fuzzy Logic: 27th Brazilian Symposium on Software Engineering. IEEE, 2013; S. 21–30.
- [DS19] DAVEY, S.; SAUNDERS, J.: User Requirements Elicitation. In: Akhgar, B.: Serious games for enhancing law enforcement agencies. From virtual reality to augmented reality. Springer International Publishing, Cham, 2019; S. 117–130.
- [EEC07] EGER, T.; ECKERT, C. M.; CLARKSON, P. J.: Engineering Change Analysis during Ongoing Product Development. In DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design 2007; 629–630.
- [Ehr07] EHRENSPIEL, K.: Integrierte Produktentwicklung – Denkabläufe, Methodeneinsatz, Zusammenarbeit. Hanser, München, 2007.
- [Est08] ESTEFAN, J. A.: Survey of model-based systems engineering (MBSE) methodologies. Focus Group 25.8. INCOSE MBSE initiative, 2008.
- [Eur20] Europäisches Parlament: Was ist künstliche Intelligenz und wie wird sie genutzt?
<https://www.europarl.europa.eu/topics/de/article/20200827STO85804/was-ist-kunstliche-intelligenz-und-wie-wird-sie-genutzt#:~:text=K%C3%BCnstliche%20Intelligenz%20ist%20die%20F%C3%84igkeit%20einer%20Maschine%2C%20menschliche,zu%20l%C3%84sen%2C%20um%20ein%20bestimmtes%20Ziel%20zu%20erreichen.,> 31.05.2024.
- [FA02] FOEGEN, M.; ATAMANIUK, P.: Modellierung von Architekturen. In: Spitta, T.; Borchers, J.; Sneid, H. M.: Software Management 2002: Progress through Constancy. Gesellschaft für Informatik e.V, Bonn, 2002; S. 119–130.
- [Fel05] FELLBAUM, C.: WordNet and wordnets. In: Barber, A.: Encyclopedia of Language and Linguistics. Elsevier, 2005; S. 2–665.
- [FHS15] FERNANDES, J.; HENRIQUES, E.; SILVA, A.; MOSS, M. A.: Requirements change in complex technical systems – An empirical study of root causes. In Research in Engineering Design, 26 2015 1; S. 37–55.
- [For23] Ford: Montage Riegel Haube. <https://www.walmart.com/ip/Dorman-820-000-Hood-Latch-Assembly-for-Specific-Ford-Lincoln-Mercury-Models-Fits-2013-Ford-Edge/460892645>, 21.01.2023.

- [Fra09] FRANKE, H.-J.: Methodische Hilfen für den Umgang mit komplexen Produkten. In: Meerkamm, H.: Design for X. Beiträge zum 20. DfX Symposium. Design Society, Glasgow, UK, 2009; S. 27–30.
- [Fri14] FRIEDENTHAL, S.: A Practical Guide to SysML – The Systems Modeling Language. Elsevier Science, San Francisco, 2014.
- [GDB18] GRÄBLER, I.; DATTNER, M.; BOTHEN, M.: Main Feature List as core success criteria of organizing Requirements Elicitation. In: RADMA: R&D Management Conference 2018. R&Designing Innovation: Transformational Challenges for Organizations and Society, 2018; S. 1–16.
- [GEL14] Ghosh, S.; Elenius, D.; Li, W.; Lincoln, P.; Shankar, N.; Steiner, W.: ARSENAL: Automatic Requirements Specification Extraction from Natural Language. <https://arxiv.org/pdf/1403.3142>, 21.03.2023.
- [Ger11] GERICKE, K.: Enhancing Project Robustness. A Risk Management Perspective. Dissertation, Technische Universität Berlin, Berlin. Betreuer:innen: Schindler, V., Blessing, L., & Meyer, H. 2011.
- [GG20] González-Carvajal, S.; Garrido-Merchán, E. C.: Comparing BERT against traditional machine learning text classification. <https://arxiv.org/pdf/2005.13012>, 21.03.2023.
- [GGF] GULANOVÁ, J.; GULAN, L.; FORRAI, M.; HIRZ, M.: Generative engineering design methodology used for the development of surface-based components: Computer-Aided Design & Applications, 2017; S. 642–649.
- [GHS12] GRÄBLER, I.; HAAS, V.; SUCHOWERSKYJ, W.: Innovation Based on Applying Design Methodology: Proceedings of TMCE 2012 "Tools and Methods of Competitive Engineering", Karlsruhe, 2012.
- [GLT05] GNESI, S.; LAMI, G.; TRENTANNI, G.: An automatic tool for the analysis of natural language requirements: Comput. Syst. Sci. Eng., 2005.
- [GO19] GRÄBLER, I.; OLEFF, C.: Risikoorientierte Analyse und Handhabung von Anforderungsänderungen. In: Krause, D.; Kristin Paetzold; Wartzack, S.: Design for X. Beiträge zum 30. DfX-Symposium. TuTech Verlag, Hamburg, 2019; S. 49–60.
- [GO22] GRÄBLER, I.; OLEFF, C.: Systems Engineering – Verstehen und industriell umsetzen. Springer Berlin Heidelberg, Berlin, Heidelberg, 2022.
- [GOH22a] GRÄBLER, I.; OLEFF, C.; HIEB, M.; PREUB, D.: Automated Requirement Dependency Analysis for Complex Technical Systems: Proceedings of the Design Society: DESIGN Conference (2022). Cambridge University Press, 2022a.
- [GOH22b] GRÄBLER, I.; OLEFF, C.; HIEB, M.; PREUB, D.: Automated Requirement Dependency Analysis for Complex Technical Systems. In DS 109:

- Proceedings of the Design Society: 23rd International Conference on Engineering Design (ICED21), 2 2022b; S. 1865–1874.
- [GOP21] GRÄBLER, I.; OLEFF, C.; PREUB, D.: Holistic change propagation and impact analysis in requirements management: Proceedings of R&D Management Conference 2021, Strathclyde/Glasgow, 2021.
- [GOP22] GRÄBLER, I.; OLEFF, C.; PREUB, D.: Proactive Management of Requirement Changes in the Development of Complex Technical Systems. In *Applied Sciences*, 12 2022 4; S. 1874.
- [GOP23] GRÄBLER, I.; OLEFF, C.; PREUB, D.; KOCH, A.-S.: Resilient Requirements Engineering – Anforderungsentwicklung im Engineering 4.0 neu gedacht. In *Zeitschrift für wirtschaftlichen Fabrikbetrieb (ZWF)*, 3 2023.
- [GOS20] GRAESSLER, I.; OLEFF, C.; SCHOLLE, P.: Method for Systematic Assessment of Requirement Change Risk in Industrial Practice. In *Applied Sciences*, 10 2020 23; S. 8697.
- [GP21] GRÄBLER, I.; POTTEBAUM, J.: Generic Product Lifecycle Model: A Holistic and Adaptable Approach for Multi-Disciplinary Product–Service Systems. In *Applied Sciences*, 11 2021 10; S. 4516.
- [GP22] GRÄBLER, I.; POTTEBAUM, J.: From Agile Strategic Foresight to Sustainable Mechatronic and Cyber-Physical Systems in Circular Economies. In *Design Methodology for Future Products* 2022.
- [GP23] GRÄBLER, I.; PREUB, D.: Automatisierte Abhängigkeitsanalyse von Anforderungen zur Wirkkettenmodellierung: Stuttgarter Symposium für Produktentwicklung 2023. Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart, 2023.
- [GPB22] GRÄBLER, I.; PREUB, D.; BRANDT, L.; MOHR, M.: Efficient Extraction of Technical Requirements Applying Data Augmentation: Proceedings of 8th IEEE International Symposium on Systems Engineering 2022, Wien, 2022.
- [GPB23] GRÄBLER, I.; PREUB, D.; BRANDT, L.; MOHR, M.: Efficient Formalisation of Technical Requirements for Generative Engineering: Proceedings of 24th International Conference on Engineering Design (ICED 2023), Bordeaux, France, 2023.
- [GPM21a] GIANNAKOPOULOU, D.; PRESSBURGER, T.; MAVRIDOU, A.; SCHUMANN, J.: Automated formalization of structured natural language requirements. In *Information and Software Technology*, 137 2021a.
- [GPM21b] GIANTAMIDIS, G.; PAPANIKOLAOU, G.; MIRANDA, M.; SALINAS-HERNANDO, G.; VALVERDE-ALCALA, J.; VELURU, S.; BASAGIANNIS, S.: ReForm: A Tool for Rapid Requirements Formalization. In *Electronic Communications of the EASST*, 79 2021b.

- [GPO20] GRÄBLER, I.; PREUB, D.; OLEFF, C.: Automatisierte Identifikation und Charakterisierung von Anforderungsabhängigkeiten – Literaturstudie zum Vergleich von Lösungsansätzen: Proceedings of the 31st Symposium Design for X (DFX2020), 2020; S. 199–208.
- [GPO21] GRÄBLER, I.; POTTEBAUM, J.; OLEFF, C.; PREUB, D.: Handling of Explicit Uncertainty in Requirements Change Management: Proceedings of the 23rd International Conference on Engineering Design (ICED 21), Delft, Netherlands, 2021.
- [GPP22] GRÄBLER, I.; PREUB, D.; POTTEBAUM, J.: Extrahierung von Anforderungen aus natürlich-sprachlichen Lastenheften: Was erschwert eine KI-basierte Extrahierung? In Softwaretechnik-Trends, 42 2022 1.
- [GPT22] GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://academictorrents.com/details/feb5891fd364f357b03a9ebbf3b7d83a0aabe1ec>, 02.03.2022.
- [Grä00] GRÄBLER, I.: Informations- und zeitbasiertes Controlling einer integrierten Konstruktion und Arbeitsplanung. Dissertation, RWTH Aachen, Aachen 2000.
- [Grä15] GRÄBLER, I.: Implementation-oriented synthesis of mechatronic reference models. In: Bertram, T.; Corves, B.; Janschek, K.: Fachtagung Mechatronik 2015. Technische Universität Dresden, Dresden, 2015; S. 167–172.
- [GTO19] GRÄBLER, I.; THIELE, H.; OLEFF, C.; SCHOLLE, P.; SCHULZE, V.: Method for Analysing Requirement Change Propagation Based on a Modified Pagerank Algorithm. In: Design Society: Proceedings of the Design Society. 22nd International Conference on Engineering Design. Cambridge University Press, Cambridge, UK, 2019; S. 3681–3690.
- [GW21] GRÄBLER, I.; WIECHEL, D.: Systematische Bewertung von Auswirkungsanalysen des Engineering Change Managements: DS 111: Proceedings of the 32nd Symposium Design for X (DFX2021), 2021; S. 1–10.
- [GWB09] GIFFIN, M.; WECK, O. de; BOUNOVA, G.; KELLER, R.; ECKERT, C.; CLARKSON, P. J.: Change Propagation Analysis in Complex Technical Systems. In Journal of Mechanical Design, 131 2009 8; S. 81001.
- [GWK22] GRÄBLER, I.; WIECHEL, D.; KOCH, A.-S.; PREUB, D.; OLEFF, C.: Model-Based Effect-Chain Analysis for Complex Systems. In Proceedings of the Design Society, Volume 2: DESIGN2022, 2 2022; S. 1885–1894.
- [GWP21] GRÄBLER, I.; WIECHEL, D.; POTTEBAUM, J.: Role model of model-based systems engineering application. In IOP Conference Series: Materials Science and Engineering, 1097 2021 1; S. 12003.

- [HCW13] HAMRAZ, B.; CALDWELL, N. H.; WYNN, D. C.; CLARKSON, P. J.: Requirements-based development of an improved engineering change management method. In *Journal of Engineering Design*, 24 2013 11; S. 765–793.
- [HDL22] Hua, W.; Dai, Z.; Liu, H.; Le, Q. V.: Transformer Quality in Linear Time. [https://arxiv.org/pdf/2202.10447](https://arxiv.org/pdf/2202.10447.pdf), 21.03.2023.
- [HH11] HAMDAQA, M.; HAMOU-LHADJ, A.: An approach based on citation analysis to support effective handling of regulatory compliance. In *Future Generation Computer Systems*, 27 2011 4; S. 395–410.
- [HK20] HARIS, M. S.; KURNIAWAN, T. A.: Automated requirement sentences extraction from software requirement specification document. In: Budi, A. S.: *Proceedings of the 5th International Conference on Sustainable Information Engineering and Technology*. Association for Computing Machinery, New York, NY, United States, 2020; S. 142–147.
- [HKR20] HARIS, M. S.; KURNIAWAN, T. A.; RAMDANI, F.: Automated Features Extraction from Software Requirements Specification (SRS) Documents as The Basis of Software Product Line (SPL) Engineering. In *Journal of Information Technology and Computer Science*, 5 2020 3; S. 279.
- [HTB23] HASELSTEINER, A. F.; THOBEN, K.-D.; BLESSING, L.: DEALING WITH EXTREME REQUIREMENT VALUES: WHAT METHODS TO DESIGN SCHOOL CHAIRS AND OFFSHORE WIND TURBINES HAVE IN COMMON. In *DS 109: Proceedings of the Design Society: 23rd International Conference on Engineering Design (ICED21)*, 3 2023; S. 1147–1156.
- [HVM18] HEIN, P. H.; VORIS, N.; MORKOS, B.: Predicting requirement change propagation through investigation of physical and functional domains. In *Research in Engineering Design*, 29 2018 2; S. 1–20.
- [IEE90] IEEE: IEEE Standard Glossary of Software Engineering Terminology. In *IEEE Std 610.12-1990* 1990.
- [JHH13] JANZEN, A.; HOFFMANN, A.; HOFFMANN, H.: Anforderungsmuster im Requirements Engineering: Working Paper Series, Kassel, Deutschland, 2013.
- [JJM21] JURAFSKY, D.; JURAFSKY, D.; MARTIN, J. H.: *Speech and language processing – An introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall; Prentice-Hall International, Upper Saddle River, NJ, London, 2021.
- [JL17] JAYATILLEKE, S.; LAI, R.: A Systematic Review of Requirements Change Management. In *Information and Software Technology*, 93 2017.

- [JLR18] JAYATILLEKE, S.; LAI, R.; REED, K.: A method of requirements change analysis. In Requirements Engineering, 23 2018 4; S. 493–508.
- [KCC12] KOH, Y.; CALDWELL, M.; CLARKSON, J.: A method to assess the effects of engineering change propagation. In Research in Engineering Design, 23 2012 4; S. 329–351.
- [KCW08] KÖHLER, C.; CONRAD, J.; WANKE, S.; WEBER, C.: A Matrix Representation of the CPM/PDD Approach as a Means for Change Impact Analysis. In DS 48: Proceedings DESIGN 2008, the 10th International Design Conference, Dubrovnik, Croatia 2008; S. 167–174.
- [KG18] KRAUSE, D.; GEBHARDT, N.: Methodische Entwicklung modularer Produktfamilien – Hohe Produktvielfalt beherrschbar entwickeln. Springer Vieweg, Berlin, 2018.
- [KGG21] KOSCINSKI, V.; GAMBARDELLA, C.; GERSTNER, E.; ZAPPAVIGNA, M.; CASSETTI, J.; MIRAKHORLI, M.: A Natural Language Processing Technique for Formalization of Systems Requirement Specifications: 29th IEEE International Requirements Engineering Conference workshops. IEEE, Piscataway, NJ, 2021; S. 350–356.
- [KM17] KURTANOVIC, Z.; MAALEJ, W.: Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning: 2017 IEEE 25th International Requirements Engineering Conference (RE), 2017; S. 490–495.
- [KO14] KNAUSS, E.; OTT, D.: (Semi-) automatic Categorization of Natural Language Requirements. In: Salinesi, C.: Requirements Engineering: Foundation for Software Quality. REFSQ 2014. Springer, Cham, 2014; S. 39–54.
- [KOR97] KARLSSON, J.; OLSSON, S.; RYAN, K.: Improved practical support for large-scale requirements prioritising. In Requirements Engineering, 2 1997 1; S. 51–60.
- [KPK02] KNETHEN, A.; PAECH, B.; KIEDAISCH, F.; HOUDEK, F.: Systematic requirements recycling through abstraction and traceability. In: IEEE: Proceedings IEEE Joint International Conference on Requirements Engineering. IEEE, Piscataway, NJ, USA, 2002; S. 273–281.
- [LGN07] LIENHARD, A.; GREEVY, O.; NIERSTRASZ, O.: Tracking Objects to Detect Feature Dependencies. In: Wong, K.: 15th IEEE International Conference on Program Comprehension. IEEE, Piscataway, NJ, 2007; S. 59–68.
- [LM13] Lachmayer, R.; Mozgová, I.: Gestaltevolution durch algorithmisierte Informationsrückführung aus dem Produktlebenszyklus ((12) N04*). Förderung von 2013 bis 2017. Teilprojekt zu SFB 653. <https://gepris.dfg.de/gepris/projekt/240642620>.

- [MAC23] MACHSWON: Haubenanpassung Stöpsel.
<https://www.amazon.com/Universal-Adjustment-Cushion-Bumpers-Stoppers/dp/B094MPH4YJ>, 21.01.2023.
- [Maz23] Mazda: Kofferraumrollo. <https://www.amazon.de/-/en/Retractable-2011-2018-Protective-Interior-Accessories/dp/B08SVS4JBQ>, 21.01.2023.
- [MBP19] Motger de la Encarnación, Joaquim; Borrull Baraut, R.; Palomares Bonache, C.; Marco Gómez, J.: OpenReq-DD: A requirements dependency detection tool. https://ceur-ws.org/Vol-2376/NLP4RE19_paper01.pdf, 21.03.2023.
- [MCG19] MARTINEZ, G. G.; CARPIO, A. F. D.; GOMEZ, L. N.: A Model for Detecting Conflicts and Dependencies in Non-Functional Requirements Using Scenarios and Use Cases: 2019 XLV Latin American Computing Conference (CLEI). IEEE, 2019.
- [Mes23] Mesko: Toaster MS 3220. [https://i.otto.de/i/otto/84fc936a-58fe-5064-995e-f201669c9f4e/mesko-toaster-ms-3220-2-scheiben-toaster-900-watt-sandwichtoaster-toastautomat-elektrisch-broetchenaufsatz-braeunungsregler-kruemelschublade-broetchen-toast-schwarz.jpg?formatz\\$](https://i.otto.de/i/otto/84fc936a-58fe-5064-995e-f201669c9f4e/mesko-toaster-ms-3220-2-scheiben-toaster-900-watt-sandwichtoaster-toastautomat-elektrisch-broetchenaufsatz-braeunungsregler-kruemelschublade-broetchen-toast-schwarz.jpg?formatz$), 12.02.2023.
- [Mis16] MISRA, J.: Terminological inconsistency analysis of natural language requirements. In *Information and Software Technology*, 74 2016; S. 183–193.
- [MM16] MACHI, L. A.; MCEVOY, B. T.: *The literature review – Six steps to success*. Corwin, Thousand Oaks, California, 2016.
- [MMM15] METH, H.; MUELLER, B.; MAEDCHE, A.: Designing a Requirement Mining System. In *Journal of the Association for Information Systems*, 16 2015 9; S. 799–837.
- [MP12] MORE, P.; PHALNIKAR, R.: Generating UML Diagrams from Natural Language Specifications. In *International Journal of Applied Information Systems*, 1 2012 8; S. 19–23.
- [MSS12] MORKOS, B.; SHANKAR, P.; SUMMERS, J. D.: Predicting requirement change propagation, using higher order design structure matrices: an industry case study. In *Journal of Engineering Design*, 23 2012 12; S. 905–926.
- [MWK16] MÉNDEZ FERNÁNDEZ, D.; WAGNER, S.; KALINOWSKI, M.; FELDERER, M.; WIERINGA, R.: *Naming the Pain in Requirements Engineering: Contemporary Problems, Causes, and Effects in Practice*. In *Empirical Software Engineering* 2016.
- [Nau21] Naumcheva, M.: Deep Learning Models in Software Requirements Engineering. <https://arxiv.org/pdf/2105.07771.pdf>, 21.03.2023.

- [Neu17] NEUMANN, M.: Ein modellbasierter Ansatz zur risikoorientierten Entwicklung innovativer Produkte. Dissertation, Ruhr-Universität Bochum, Bochum. Betreuer:innen: Prof. Dr.-Ing. Beate Bender 2017.
- [NFS14] NINAUS, G.; FELFERNIG, A.; STETTINGER, M.; REITERER, S.: INTELLIREQ: Intelligent Techniques for Software Requirements Engineering: Frontiers in Artificial Intelligence and Applications, 2014.
- [Ngu23] NGUYEN, B. M. A.: Abhängigkeitsanalyse von Anforderungen durch aktiv-lernende Künstliche Intelligenz. Masterarbeit, Universität Paderborn, Heinz Nixdorf Institut, Paderborn. Betreuer:innen: Prof. Dr.-Ing. Iris Gräßler, Daniel Preuß 2023.
- [NL03] NEILL, C. J.; LAPLANTE, P. A.: Requirements engineering: The state of the practice. In IEEE Software, 20 2003 6; S. 40–45.
- [NRC02] NATT OCH DAG, J.; REGNELL, B.; CARLSHAMRE, P.; ANDERSSON, M.; KARLSSON, J.: A Feasibility Study of Automated Natural Language Requirements Analysis in Market-Driven Development. In Requirements Engineering, 7 2002 1; S. 20–33.
- [Obj05] Object Management Group: OMG: Unified Modeling Language Infrastructure Specification, Version 2.0. <https://www.omg.org/spec/UML/2.0/About-UML>, 21.03.2023.
- [Obj19] Object Management Group: OMG Systems Modeling Language (OMG SysML™). <http://www.omg.sysml.org/>, 21.03.2023.
- [Ole22] OLEFF, C.: Proaktives Management von Anforderungsänderungen in der Entwicklung komplexer technischer Systeme. Dissertation, Universität Paderborn, Paderborn. Betreuer:innen: Prof. Dr.-Ing. Iris Gräßler, Prof. Dr.-Ing. Rainer Stark 2022.
- [OZ18] OSMAN, M. H.; ZAHARIN, M. F.: Ambiguous software requirement specification detection: an automated approach: Proceedings of the 5th International Workshop on Requirements Engineering and Testing (RET '18). Association for Computing Machinery, New York, NY, USA, 2018; S. 33–40.
- [Pat10] Paterson, J.: UK Fishing Vessel's Electronic Logbook Functional Requirements Specification. <https://www.coursehero.com/file/41143795/12-uk-fishingvesselpdf/>, 13.05.2022.
- [Pat82] PATZAK, G.: Systemtechnik — Planung komplexer innovativer Systeme – Grundlagen, Methoden, Techniken. Shaker Verlag, Berlin, Heidelberg, 1982.
- [PKK00] PARK, S.; KIM, H.; KO, Y.; SEO, J.: Implementation of an efficient requirements-analysis supporting system using similarity measure

- techniques. In *Information and Software Technology*, 42 2000 6; S. 429–438.
- [PMM21] Prabhu, S.; Mohamed, M.; Misra, H.: Multi-class Text Classification using BERT-based Active Learning. <https://arxiv.org/pdf/2104.14289>, 21.03.2023.
- [Poh96] POHL, K.: *Process-centered requirements engineering*. Wiley, New York, N.Y., USA, 1996.
- [PR21] POHL, K.; RUPP, C.: *Basiswissen Requirements Engineering – Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. dpunkt-Verlag, Heidelberg, 2021.
- [PWH17] PARIÈS, J.; WREATHALL, J.; HOLLNAGEL, E.: *Resilience engineering in practice – A guidebook*. Ashgate, Farnham, Surrey, England, Burlington, VT, 2017.
- [RJ01] RAMESH, B.; JARKE, M.: Toward reference models for requirements traceability. In *IEEE Transactions on Software Engineering*, 27 2001 27; S. 58–93.
- [RPV03] ROBINSON, W. N.; PAWLOWSKI, S. D.; VOLKOV, V.: Requirements interaction management. In *ACM Computing Surveys*, 35 2003 2; S. 132–190.
- [Rup14] RUPP, C.: *Requirements-Engineering und -Management – Aus der Praxis von klassisch bis agil*. Hanser, München, 2014.
- [Rup16] RUPP, C.: *MASTeR - Schablonen für alle Fälle*. SOPHIST GmbH, Nürnberg, 2016.
- [RWC19] RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I.: *Language Models are Unsupervised Multitask Learners* 2019.
- [SBC19] Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J. W.; Kreps, S.; McCain, M.; Newhouse, A.; Blazakis, J.; McGuffie, K.; Wang, J.: *Release Strategies and the Social Impacts of Language Models*. <https://arxiv.org/pdf/1908.09203>, 21.03.2023.
- [Sch21] SCHÄWE, L.: Entwicklung der Systemarchitektur einer Softwareunterstützung zur ganzheitlichen Risikoabschätzung und -handhabung von Anforderungsänderungen. Studienarbeit, Universität Paderborn, Heinz Nixdorf Institut, Paderborn. Betreuer:innen: Prof. Dr.-Ing. Iris Gräßler, Christian Oleff, Daniel Preuß 2021.
- [Sen92] SENGE, P. M.: *The fifth discipline – The art and practice of the learning organization*. Doubleday, New York, N.Y., 1992.

- [Set09] Settles, B.: Active Learning Literature Survey.
<https://minds.wisconsin.edu/handle/1793/60660>, 21.03.2023.
- [SG18] SHWETA, R. S.; GHOSHAL, B.: Automatic Extraction of Structural Model from Semi Structured Software Requirement Specification: IEEE/ACIS 17th International Conference 2018, 2018; S. 543–558.
- [SHB19] SONG, Y.-W.; HERZOG, M.; BENDER, B.: Understanding the Initial Requirements Definition in Early Design Phases. In Proceedings of the Design Society: International Conference on Engineering Design, 1 2019 1; S. 3751–3760.
- [SM02] SMITH, P. G.; MERRITT, G. M.: Proactive risk management – Controlling uncertainty in product development. Productivity Press, New York, NY, 2002.
- [SM15] S. BTOUSH, E.; M. HAMMAD, M.: Generating ER Diagrams from Requirement Specifications Based On Natural Language Processing. In International Journal of Database Theory and Application, 8 2015 2; S. 61–70.
- [SPL17] SHAO, F.; PENG, R.; LAI, H.; WANG, B.: DRank: A semi-automated requirements prioritization method based on preferences and dependencies. In Journal of Systems and Software 2017 126; S. 141–156.
- [SSA19] SAMER, R.; STETTINGER, M.; ATAS, M.; FELFERNIG, A.; RUHE, G.; DESHPANDE, G.: New Approaches to the Identification of Dependencies between Requirements: 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, 2019; S. 1265–1270.
- [STM03] Sang, E. F.; Tjong, K.; Meulder, F. de: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition.
<https://arxiv.org/pdf/cs/0306050>, 21.03.2023.
- [Sum09] Summa Technologies: Construction Junction – Inventory Management Software Requirements Specification.
<https://1library.net/document/y83mjdqr-construction-junction-inventory-management-software-requirements-specification.html>, 13.05.2022.
- [SW16] Staab, S.; Wartzack, S.: COFFEE - Kollaboratives Vergessen für die Produktentwicklung. Förderung seit 2016. DFG-Teilprojekt zu SPP 1921.
<https://gepris.dfg.de/gepris/projekt/318363223?context=projekt&task=showDetail&id=318363223&>.
- [Syn23] Synera GmbH: Synera. <https://www.synera.io/>, 21.03.2023.
- [Tan23] TAN, X.: Neural Text-to-Speech Synthesis. Springer Nature Singapore; Imprint Springer, Singapore, 2023.
- [The17] The Standish Group: Chaos Manifesto 2018, West Yarmouth, USA, 2017.

- [The95] The Standish Group: The CHAOS report, 1995.
- [UH76] ULRICH, P.; HILL, W.: Wissenschaftstheoretische Grundlagen der Betriebswirtschaftslehre. In Wirtschaftswissenschaftliches Studium: Zeitschrift für Ausbildung und Hochschulkontakt, 5 1976 7+8.
- [Ulr81] ULRICH, H.: Die Betriebswirtschaftslehre als anwendungsorientierte Sozialwissenschaft. In: Geist, M.; Köhler, R.: Die Führung des Betriebes. Curt Sandig zu seinem 80. Geburtstag gewidmet. Poeschel, Stuttgart, 1981; S. 1–27.
- [Vah20] Vahabi, S.: Comprehensive Watershed Management Water Use Tracking Projekt. <https://ieee-dataport.org/documents/comprehensive-watershed-management-water-use-tracking-project>, 21.03.2023.
- [Van12] VANINI, U.: Risikomanagement – Grundlagen; Instrumente; Unternehmensspraxis. Schäffer-Poeschel Verlag, 2012.
- [VDI19] VDI-Fachbereich Produktentwicklung und Projektmanagement: Entwicklung technischer Produkte und Systeme - Modell der Produktentwicklung (VDI 2221, Blatt 1). Beuth Verlag GmbH, Düsseldorf.
- [VDI21] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik VDI: Entwicklung mechatronischer und cyber-physischer Systeme (VDI 2206). Beuth Verlag GmbH, Berlin.
- [Ver14] Verein Deutscher Ingenieure: Lastenheft/Pflichtenheft für den Einsatz von Automatisierungssystemen. Beuth Verlag, Berlin.
- [VSP17] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I.: Attention Is All You Need. <https://arxiv.org/pdf/1706.03762>, 21.03.2023.
- [WB85] WEISS, D. M.; BASILI, V. R.: Evaluating Software Development by Analysis of Changes: Some Data from the Software Engineering Laboratory. In IEEE Transactions on Software Engineering, SE-11 1985 2; S. 157–168.
- [Wei14] WEILKENS, T.: Systems Engineering mit SysML/UML – Anforderungen, Analyse, Architektur. dpunkt.verlag; Ciando, Heidelberg, München, 2014.
- [WHT14] WELLSANDT, S., HRIBERNIK, K. A.; THOBEN, K.-D.: Qualitative Comparison of Requirements Elicitation Techniques that are Used to Collect Feedback Information about Product Use. In 24th CIRP Design Conference, 21 2014.
- [WHT15] WIESNER, S.; HAUGE, J. B.; THOBEN, K.-D.: Challenges for Requirements Engineering of Cyber-Physical Systems in Distributed Environments. In: Umeda, S. et al.: Advances in Production Management Systems: Innovative

- Production Management Towards Sustainable Growth. Springer International Publishing, Cham, 2015; S. 49–58.
- [Wic17] WICKEL, M. C.: Änderungen besser managen: Eine datenbasierte Methodik zur Analyse technischer Änderungen. Dissertation, Technische Universität München, München 2017.
- [WJB21] WANG, X.; JIANG, Y.; BACH, N.; WANG, T.; HUANG, Z.; HUANG, F.; TU, K.: Automated Concatenation of Embeddings for Structured Prediction. In: Zong, C. et al.: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Stroudsburg, PA, USA, 2021; S. 2643–2660.
- [WMT17] WIESNER, S.; MARILUNGO, E.; THOBEN, K.-D.: Cyber-Physical Product-Service Systems – Challenges for Requirements Engineering. In: International Journal of Automation Technology, 11 2017 1; S. 17–28.
- [WPH15] WIESNER, S.; PERUZZINI, M.; HAUGE, J. B.; THOBEN, K.-D.: Requirements Engineering. In: Stjepandić, J.; Wognum, N.; J.C. Verhagen, W.: Concurrent Engineering in the 21st Century. Springer International Publishing, Cham, 2015; S. 103–132.
- [WRF15] WALDEN, D. D.; ROEDLER, G. J.; FORSBERG, K.: INCOSE Systems Engineering Handbook Version 4: Updating the Reference for Practitioners. In INCOSE International Symposium, 25 2015 1; S. 678–686.
- [WRH97] WILSON, W. M.; ROSENBERG, L. H.; HYATT, L. E.: Automated analysis of requirement specifications. In: Adrion, W. R.: ICSE 97. 19th Annual Conference on Software Engineering. Association for Computing Machinery, New York, Oct. 1997; S. 161–171.
- [ZH17] ZICHLER, K.; HELKE, S.: Ontologiebasierte Abhängigkeitsanalyse im Projektlastenheft. In: Dencker, P. et al.: Automotive - Safety & Security 2017. Sicherheit und Zuverlässigkeit für automobile Informationstechnik. Gesellschaft für Informatik e.V. (GI), Bonn, 2017.
- [Zha22] ZHANG, X.: Entwicklung der Systemarchitektur eines Softwarewerkzeugs zur Extrahierung und Formalisierung von Anforderungen. Bachelorarbeit, Universität Paderborn, Heinz Nixdorf Institut, Paderborn. Betreuer:innen: Prof. Dr.-Ing. Iris Gräßler, Daniel Preuß 2022.
- [ZJ05] ZHU, X.; JIN, Z.: Inconsistency measurement of software requirements specifications: An ontology-based approach: Proceedings, 10th IEEE International Conference on Engineering of Complex Computer Systems. ICECCS 2005, 16-20 June 2005, Shanghai, China. IEEE Computer Society, Los Alamitos, Calif, 2005; S. 402–410.

- [ZLZ14] ZHANG, H.; LI, J.; ZHU, L.; JEFFERY, R.; LIU, Y.; WANG, Q.; LI, M.: Investigating dependencies in software requirements for change propagation analysis. In *Information and Software Technology*, 56 2014 1; S. 40–53.

Zitation aus studentischen Arbeiten

Die nachstehend aufgeführten studentischen Arbeiten wurden im Kontext der vorliegenden Dissertation am Lehrstuhl für Produktentstehung der Universität Paderborn angefertigt. Die Definition der Zielsetzung, die Bearbeitung sowie die Auswertung, Interpretation und Visualisierung von Ergebnissen erfolgten unter wissenschaftlicher Anleitung der Betreuenden Prof. Dr.-Ing. Iris Gräßler und Daniel Preuß. Die erzielten Ergebnisse sind zum Teil in die Dissertation eingeflossen.

- [Ngu23] Nguyen, B. M. A. (Betreuer: Gräßler, I.; Preuß, D.): Abhängigkeitsanalyse von Anforderungen durch aktiv-lernende Künstliche Intelligenz. Universität Paderborn, Heinz Nixdorf Institut – Lehrstuhl für Produktentstehung, unveröffentlichte Masterarbeit, 2023.
- [Sch21] Schawe, L. (Betreuer: Gräßler, I.; Oleff, C., Preuß, D.): Entwicklung der Systemarchitektur einer Softwareunterstützung zur ganzheitlichen Risikoabschätzung und -handhabung von Anforderungsänderungen. Universität Paderborn, Heinz Nixdorf Institut – Lehrstuhl für Produktentstehung, unveröffentlichte Studienarbeit, 2021.
- [Zha22] Zhang, X. (Betreuer: Gräßler, I.; Preuß, D.): Entwicklung der Systemarchitektur eines Softwarewerkzeugs zur Extrahierung und Formalisierung von Anforderungen. Universität Paderborn, Heinz Nixdorf Institut – Lehrstuhl für Produktentstehung, unveröffentlichte Bachelorarbeit, 2022.

Anhang

A1	Ergänzende Unterlagen zu den Grundlagen	141
A2	Unterlagen zu den Vorarbeiten.....	146
A2.1	Wirkkettenmodellierung.....	147
A2.2	Nutzwertanalyse.....	151
A2.3	ARCA Projekt.....	153
A2.4	ProMaRC-Methodik.....	156
A2.5	Bewertung von Änderungsauswirkungen von Anforderungen	165
A2.6	Projektcharakteristika zur Auswahl von Vergleichsprojekten	166
A2.7	Systemmodellierung.....	167
A3	Aktivitäten zur Einbindung von Industrieanwendern	169
A3.1	ARCA: Rahmendaten von Aktivitäten und Unterlagen	169
A3.2	BIKINI: Rahmendaten von Aktivitäten und Unterlagen.....	184
A4	Unterlagen zur Methodikentwicklung	200
A4.1	Anforderungsdiagramme.....	200
A4.2	Use-Case Diagramme.....	204
A4.3	Aktivitätsdiagramme.....	207
A4.4	Paketdiagramme	210
A4.5	Algorithmen zum Extrahieren von Textsegmenten	212
A4.6	Algorithmen zur Klassifizierung von Anforderungen.....	213
A4.7	Algorithmen zur Formalisierung von Anforderungen.....	214
A4.8	Domänenlexika	215
A4.9	Algorithmen zur Abhängigkeitsanalyse von Anforderungen	235
A5	Unterlagen zur Validierung der Methodik.....	236

A1 Ergänzende Unterlagen zu den Grundlagen

Tabelle A-1: Arten von Anforderungsabhängigkeiten [Ole22]

Nr.	Bezeichnung	Beschreibung	Gleichzusetzen mit
1	Funktionale Beschränkung	Zwischen den Anforderungen gibt es Beschränkungen und Restriktionen, die sich auf die Anforderungsfunktion beziehen.	Constraints [Poh96]; Constrain [ZLZ14]; Randbedingung [Rup14]; Realisierungsvorgabe [Rup14]
2	Erfordert	Eine Anforderung kann nicht ohne eine zweite Anforderung funktionieren. Dies gilt nur in eine Richtung.	Must-Exist [KOR97]; Requires [CSL01]; Fachliche Abhängigkeit [Rup14]; And [CSL01]
3	Oder	Lediglich eine der beiden Anforderungen kann implementiert werden und nicht beide.	Cannot-Exist [KOR97]; Or [CSL01]
4	Vorbedingung	Bei der Implementierung muss eine Reihenfolge berücksichtigt werden. Eine der Anforderungen muss daher zeitlich vor der anderen eingeführt werden.	Precondition [Poh96]; Temporal [CSL01]; Temporal [RPV03]; Precede [ZLZ14]
5	Inhalt	Die Anforderungen haben eine Abhängigkeit auf inhaltlicher Ebene.	Content [Poh96];
6	Übereinstimmung	Die Anforderungen sind identisch oder haben große Ähnlichkeiten.	Similar [Poh96]; Structure [RPV03]; Be_similar_to [ZLZ14]
7	Aufgabe	Eine Anforderung beschreibt eine Aufgabe, die von einer anderen Anforderung abhängt.	Task [RPV03]
8	Widerspruch	Zwischen den Anforderungen gibt es Inkonsistenzen.	Contradicts [Poh96];

Nr.	Bezeichnung	Beschreibung	Gleichzusetzen mit
9	Konsequenz	Eine Anforderung beschreibt die Konsequenz einer anderen Anforderung.	Causality [RPV03]
10	Ausnahme	Eine Anforderung beschreibt einen Ausnahmefall, für Störungen des Funktionsablaufs einer anderen Anforderung.	Ausnahme-Abhängigkeit [Rup14]; Be_exception_of [ZLZ14]
11	Weiterentwicklung	Die Anforderung hat sich in Teilen oder als Ganzes weiterentwickelt.	Evolutionary [Poh96]; Evolve_into [ZLZ14]
12	Formulierung	Der Inhalt der ursprünglichen Anforderung wird in dieser Anforderung neu formuliert.	Elaborates [Poh96]; Elaborate [RJ01]; Formalizes [Poh96];
13	Basiert_auf	Der Inhalt einer Anforderung wird aufgrund einer anderen Anforderung geändert.	Based_on [Poh96]
14	Ersetzt	Die Anforderung wurde durch eine andere Anforderung ersetzt.	Replaces [Poh96]
15	Abhängigkeit	Eine Anforderung ist abhängig von einer anderen Anforderung.	Condition [Poh96]; Depend_on [RJ01]; Dependency [KPK02]
16	Verfeinerung	Eine Anforderung wurde detaillierter in anderen Anforderungen definiert.	Refines [Poh96]; Derive [RJ01]; Eltern/Kind-Verbindung [Rup14]; Refine [ZLZ14]; Refinement [KPK02]; Part_of [RJ01]; Abstraction [Poh96]; Generalizes [Poh96]

Nr.	Bezeichnung	Beschreibung	Gleichzusetzen mit

17	Kosten/Wert	Die Relation zwischen Implementierungskosten und dem Kundenwert dieser Anforderung für den Kunden wird durch eine andere Anforderung beeinflusst.	
18	Kostensteigerung	Eine Anforderung erhöht die Implementierungskosten einer anderen Anforderung.	Negative Cost [KOR97]; ICost [CSL01]; Increase/Decrease_cost_of [ZLZ14]
19	Kostensenkung	Eine Anforderung verringert die Implementierungskosten einer anderen Anforderung.	Positive Cost [KOR97]; ICost [CSL01]; Increase/Decrease_cost_of [ZLZ14]
20	Wertsteigerung	Eine Anforderung steigert den Kundenwert einer anderen Anforderung.	Positive Value [KOR97]; CValue [CSL01]; Increase/Decrease_value_of [ZLZ14]
21	Wertminderung	Eine Anforderung mindert den Kundenwert einer anderen Anforderung.	Negative Value [KOR97]; CValue [CSL01]; Increase/Decrease_value_of [ZLZ14]
22	Korrelation	Eine Anforderung beeinflusst eine andere Anforderung.	
23	Positive Korrelation	Ein verbesserter Erfüllungsgrad der einen Anforderung erhöht auch den Erfüllungsgrad der anderen Anforderung.	Positive Correlation [RPV03]; Satisfies [Poh96]
24	Negative Korrelation	Ein verbesserter Erfüllungsgrad der einen Anforderung vermindert den Erfüllungsgrad der anderen Anforderung.	Negative Correlation [RPV03]; Conflicts [Poh96]; Conflict [ZLZ14]
25	Unbekannte Korrelation	Ein veränderter Erfüllungsgrad der einen Anforderung hat einen unbekannten Effekt auf den Erfüllungsgrad der anderen Anforderung.	Unspecified Correlation [RPV03]

Zugrundeliegende Begriffe

Anforderungen

Eine Anforderung wird definiert als [PR21]:

- 5) Ein notwendiges Bedürfnis eines Stakeholders.
- 6) Eine Fähigkeit oder Eigenschaft, die ein System erfüllen muss.
- 7) Eine dokumentierte Repräsentation eines Bedürfnisses, einer Fähigkeit oder Eigenschaft.

In dieser Arbeit werden Anforderungen in funktionale und nicht-funktionale Anforderungen (Anforderungsarten) unterteilt. Die nicht-funktionalen Anforderungen werden unterteilt in Qualitäts- und technologische Anforderungen, Anforderungen an die Benutzungsoberfläche und rechtlich-vertragliche Anforderungen [Rup14].

Lastenheft

„Vom Auftraggeber oder in dessen Auftrag erstellte Zusammenstellung aller Anforderungen des Auftraggebers hinsichtlich Liefer- und Leistungsumfang als Ausschreibungs-, Angebots- und/oder Vertragsgrundlage.“ [Ver14]

Pflichtenheft

„Beschreibung der Realisierung aller Anforderungen des Lastenhefts.“ [Ver14]

Formalisierung (von Anforderungen)

Die Formalisierung natürlicher Sprache bedeutet die Extraktion unstrukturierter, in Texte eingebetteter Informationen in ein strukturiertes und maschinenlesbares Datenformat [JJM21]. Die Formalisierung von Anforderungen bedeutet, dass natürlich-sprachliche Anforderungen maschinenlesbar gemacht werden, indem beispielsweise Schlüssel-Wert-Paare der Anforderung extrahiert werden.

Generative Entwicklung

Generative Entwicklung ist ein Entwicklungsprozess indem das Produkt oder dessen Bauteile auf der Grundlage von quantitativen und qualitativen Daten sowie geometrischen Eingaben automatisiert generiert wird [GGF].

Künstliche Intelligenz

„Künstliche Intelligenz ist die Fähigkeit einer Maschine, menschliche Fähigkeiten wie logisches Denken, Lernen, Planen und Kreativität zu imitieren. KI ermöglicht es technischen Systemen, ihre Umwelt wahrzunehmen, mit dem Wahrgenommenen umzugehen und Probleme zu lösen, um ein bestimmtes Ziel zu erreichen.“ [Eur20]

Daten-Augmentation

Die Daten-Augmentation bezieht sich auf Techniken, die dazu verwendet werden, den Umfang und die Vielfalt eines Datensatzes zu erhöhen, indem vorhandene Daten modifiziert werden. Dies kann durch verschiedene Methoden erreicht werden, je nach Art der Daten (Bild, Audio, Text). [Tan23]

Aktives Lernen

Aktives Lernen ist eine Form des maschinellen Lernens, bei der ein Lernalgorithmus interaktiv ein Orakel (typischerweise einen menschlichen Experten) befragt, um die gewünschte Kennzeichnung für neue Daten zu erhalten [Set09].

A2 Unterlagen zu den Vorarbeiten

In dieser Arbeit werden Ergebnisse aus den Projekten ARCA, BIKINI und ZWM genutzt, um Erkenntnisse für die Entwicklung der Methodik abzuleiten. In ARCA mit einer Laufzeit von 2 Jahren wurde eine Methodik für das proaktive Risikomanagement von Anforderungsänderungen in der Entwicklung komplexer technischer Systeme in Zusammenarbeit mit dem Industriepartner IAV GmbH erforscht [GOP22]. Die Erkenntnisse aus ARCA wurden in dem BMWK-geförderten 3-jährigen Projekt BIKINI aufgegriffen. Hier wurde eine Methodik für die Extrahierung und Formalisierung von Anforderungen und deren Abhängigkeitsanalyse entwickelt [GP23, GPB22, GPB23]. Es wurde primär mit den Industriepartnern Atos Information Technology GmbH und EDAG Engineering GmbH zusammenarbeitet. In dem Projekt ZWM wurde in Zusammenarbeit mit einem Consulting Unternehmen und einem weltweit führenden Automobilhersteller ein Vorgehen zur Modellierung von Wirkketten für die Auswirkungsanalyse von Änderungen an zertifizierungsrelevanten Automobil-Komponenten ausgearbeitet [GWK22]. Steckbriefe zu den Projekten werden in Bild A-1 dargestellt.

Automated Requirement Change Analysis for the development of complex technical systems (ARCA)

Beschreibung: Entwicklung von Methoden (und deren Implementierung) zur Bewertung des Änderungsrisikos von Anforderungen in interdisziplinären Entwicklungsprojekten.
Partner: IAV GmbH
Fördergeber: BMBF Software-Campus
Laufzeit & Volumen: 2 Jahre, 100.000€



Bionik und KI zur nachhaltigen Integration in der Produktentwicklung (BIKINI)

Beschreibung: Extrahierung und Formalisierung sowie Abhängigkeitsanalyse von Anforderungen.
Partner: Atos Information Technology GmbH, EDAG Engineering GmbH, Krause DiMaTec GmbH, RHaug GmbH, Alfred Wegener Institut, Additive Marking GmbH
Fördergeber: BMWK
Laufzeit & Volumen: 3 Jahre, 4.000.000€



Zertifizierungsgerechte Wirkkettenmodellierung (ZWM)

Beschreibung: Modellierung von Abhängigkeiten zwischen Artefakten für die Auswirkungsanalyse von Änderungen.
Partner: Consulting Unternehmen, Automobilhersteller
Fördergeber: bilaterales Forschungsprojekt
Laufzeit & Volumen: 8 Monate, 190.000€

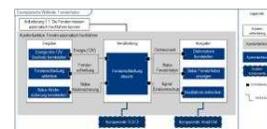


Bild A-1: Projektsteckbriefe

A2.1 Wirkkettenmodellierung

Die Komplexität technischer Systeme führt zu einer zunehmenden Anzahl von Wechselwirkungen zwischen den Systemelementen des Fahrzeugs, die bei der Entwicklung berücksichtigt werden müssen. Zum Beispiel die Interaktion zwischen verschiedenen Teilsystemen des Fahrzeugs. Ein Beispiel ist die Interaktion von Fensterheber und Klimasteuerung mit den Benutzerbefehlen von Geräten wie einem Smart Key. Alle Systeme haben ihre eigenen Entwicklungsaufgaben (Anforderungen, Funktionen, logische Elemente, physikalische Elemente) und sind von zusätzlichen Artefakten des Systemkontextes wie z. B. Regularien umgeben. Aufgrund des komplexen Zusammenspiels haben Änderungen an Systemelementen nicht nur unmittelbare Auswirkungen, sondern pflanzen sich innerhalb des Gesamtsystems und über Systemgrenzen hinweg fort und führen zu steigenden Kosten, Qualitätsveränderungen oder sogar zum Scheitern des Projekts [EEC07]. Die daraus resultierenden Auswirkungen sind ohne systematischen Ansatz und Softwareunterstützung kaum zu bewerten. Die Auswirkungsanalyse kann genutzt werden, um vorhandene Informationen zu analysieren und den Entwicklern zu helfen, die Auswirkungen von auftretenden Änderungswünschen abzuschätzen [GW21]. Die Modellierung von Wirkketten ist eine Möglichkeit, ein solches Informationsmodell zu erstellen. Hierbei werden Artefakte miteinander verknüpft, um eine Nachverfolgbarkeit herzustellen. [GWK22]

Im Rahmen des ZWM-Projekts wurde die „Model-based Effect Chain Analysis“ (MECA) Methode von GRÄBLER, WIECHEL, KOCH, PREUB und OLEFF veröffentlicht [GWK22]. Die Methode ist generisch-anwendbar und besitzt eine flexible Struktur, um die Hauptmerkmale von Wirkketten-Modellierungs- und Analyseansätzen zu verbinden und eine Anpassung zu ermöglichen. Initial wird das individuelle Ziel der Anwendung der Wirkkette definiert. Anhand des Ziels werden Start- und Endpunkt sowie Elemente und Abhängigkeiten der zu modellierenden Betrachtung bestimmt. Die Methode kann in unterschiedlichen Kontexten eingesetzt werden, ohne sich auf bestimmte Artefakte, Disziplinen oder Domänen zu beschränken. Erreicht wird dies durch eine flexible Struktur, aber auch durch die Verwendung von SysML als Modellierungssprache, die für komplexe technische Systeme konzipiert ist. In einem SysML-Profil können organisatorische oder umweltbezogene Artefakte sowie spezifische Ursache-Wirkungs-Beziehungen definiert werden, wobei Stereotypen zur Erfassung der erforderlichen Informationen verwendet werden. Außerdem ist es möglich, Views und Viewpoints zu definieren, um rollenspezifische Informationen zu spezifizieren. Daher hilft die MECA-Methode den Entwicklern, Ursache-Wirkungs-Beziehungen zwischen Artefakten innerhalb eines ausführbaren Modells zu identifizieren und zu modellieren, die dann zur Analyse der spezifischen Auswirkungen verwendet werden können. [GWK22]

Schritt 1: Definition des Ziels der Wirkkettenmodellierung

Zunächst muss das Ziel der Wirkungskettenmodellierung definiert werden. Der betrachtete Modellierungskontext bestimmt den Umfang der benötigten Systemelemente und deren Ursache-Wirkungs-Abhängigkeiten. Das Ziel muss von den typischerweise beteiligten MBSE-Rollen, z. B. dem Modellierungs-Ingenieur oder dem Fachexperten, geklärt werden [GWP21]. Das Ziel muss definiert und für alle interagierenden Personen verfügbar sein, um die bedarfsgerechte Modellierung zu gewährleisten. Beispielhafte Ziele für die Modellierung von Wirkungsketten sind die Erfüllung externer Vorgaben oder die Bewertung der Auswirkungen von technischen Änderungen. [GWK22]

Schritt 2: Identifikation der verfügbaren Informationen

Im zweiten Schritt müssen Abhängigkeiten von Systemelementen innerhalb eines abgegrenzten Systemausschnitts (horizontaler Scope) und über vordefinierte Systemebenen (vertikaler Scope) identifiziert werden. Dabei werden Systemelemente unterschiedlicher Art berücksichtigt. Der horizontale Scope umfasst die Bestimmung der betrachteten Disziplinen und der zugehörigen Informationen aus bestehenden Teilmodellen. Der vertikale Scope bestimmt die Granularitätsebene der zu entwickelnden Wirkkette. Dazu muss ein Set von Modellierungselementen definiert werden. Die Menge kann sowohl typische Entwicklungsartefakte als auch organisatorische oder umweltbezogene Artefakte wie Vorschriften umfassen. In Kombination bestimmen alle Dimensionen die Informationen, die zur Modellierung der Wirkkette erforderlich sind. Der daraus resultierende Output, bzw. die zu modellierenden Informationen, dienen als Input für den nächsten Schritt. [GWK22]

Schritt 3: Modellierung von Wirkketten

Die Wirkkette muss in Abhängigkeit von den vorhandenen Informationen modelliert werden. Im Vergleich zu „Design Structure Matrizen“ oder Domain Mapping Matrizen ermöglicht die Modellierung von Wirkketten die Verwaltung einer großen Anzahl heterogener Entwicklungsartefakte (Vorschriften, Anforderungen, Komponenten, etc.) und verschiedener Arten von Abhängigkeiten zwischen ihnen. Wirkketten enthalten Systemelementinformationen (Knoten) und deren Rückverfolgbarkeitsinformationen (Kanten) einschließlich der kontextspezifischen Ursache-Wirkungs-Beziehungen. Zunächst wird die Modellierungssprache durch die Definition von Stereotypen in einem Profil angepasst (Schritt 3.1). Stereotypen ermöglichen die Abbildung von Informationen, die nicht durch SysML spezifiziert sind, zum Beispiel Zertifizierungsanforderungen als eine Art <<Anforderung>>. Zusätzliche, spezifische Abhängigkeiten können auf der Basis bestehender Abhängigkeiten wie <<Allokation>> definiert werden. Anschließend können alle standardisierten und kundenspezifischen Informationen innerhalb der Wirkkette modelliert werden. Abhängig von der definierten Informationsmenge wird das System auf verschiedene Arten zerlegt (Schritt 3.2). Dazu gehört die Dekomposition von

Anforderungen, Funktionen, logischen Elementen und physischen Elementen. Auf der Grundlage der Dekomposition wird ein Satz strukturorientierter (Schritt 3.3) und verhaltensorientierter (Schritt 3.4) Diagramme festgelegt. Dazu gehören die Strukturdiagramme (Blockdefinitionsdiagramm, internes Blockdiagramm, Paketdiagramm, Parameterdiagramm) und die Verhaltensdiagramme (Aktivitätsdiagramm, Sequenzdiagramm, Zustandsautomatendiagramm und Anwendungsfalldiagramm) der SysML. Bei der MECA-Methode dient die IPO-Logik [Alt12] innerhalb eines internen Blockdiagramms als zentraler funktionaler Teil der Wirkkette. Diesem Diagramm kann mit Hilfe des spezifischen SysML-Profils ein beliebiges Element zugeordnet werden. Im letzten Schritt (Schritt 3.5) werden die standardisierten und kundenspezifischen Abhängigkeiten zwischen diesen Modellelementen modelliert, die zu einer nachvollziehbaren Wirkkette zwischen allen Systemelementen im Scope führen. Es empfiehlt sich die Verwendung von <>Allokation>>, die durch Stereotypen angepasst werden kann. [GWK22]

Schritt 4: Analyse der resultierenden Effekte innerhalb des Modellierungskontextes

Die Wirkketten können mit verschiedenen Ansätzen analysiert werden, z. B. Auswirkungsanalyse im Kontext von Engineering Change Management [GW21, HCW13] oder Requirements Change Management [JLR18]. Auch können bestehende Ansätze an die MECA-Methode und an die unternehmensspezifischen Anforderungen angepasst werden. Insgesamt wird die Auswirkungsanalyse durch den Automatisierungsgrad (nicht automatisiert, teilautomatisiert, vollautomatisiert); die Rückverfolgbarkeit (nicht rückverfolgbar, teilrückverfolgbar, rückverfolgbar) und den Bezugspunkt der Analyse (Qualität, Kosten, Zeit) charakterisiert [GW21]. Die Analyse ist abhängig von der Zielsetzung der Wirkkettenmodellierung (Schritt 1), den verfügbaren Informationen (Schritt 2) und dem resultierenden Informationsmodell (Schritt 3). Die systematischen Ansätze dienen der Ableitung qualitativer oder quantitativer Ergebnisse mit Hilfe von Algorithmen oder Expertenwissen. Der Ablauf der MECA-Methode wird in Bild A-2 dargestellt. [GWK22]

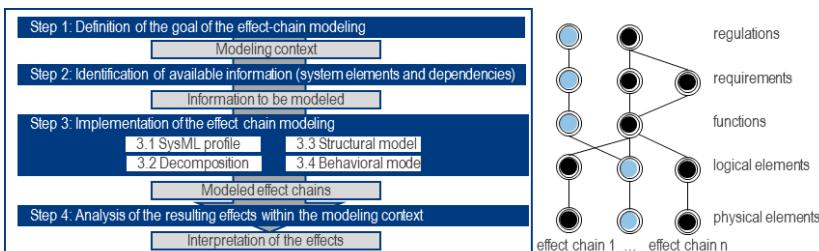


Bild A-2: MECA-Methode zur Wirkkettenmodellierung [GWK22]

Abhängigkeitsanalyse von Anforderungen zur Wirkkettenmodellierung

Eine aktuelle Herausforderung in der Entwicklung von Automobilen besteht darin, dass Fahrzeughersteller aufgrund der UN-ECE R156 Regularie verpflichtet sind, die Auswirkungen von technischen Änderungen auf Softwarekomponenten nachvollziehen zu können. Durch das Informationsmodell, welches mittels Wirkkettenmodellierung erstellt wird, werden Auswirkungen von Änderungen auf Systemelemente bewertbar. Ein Problem in der Wirkkettenmodellierung besteht darin, dass die detaillierte Modellierung bestimmter Artefakte aufwändig ist. Dies betrifft unter anderem das Umgebungsartefakt „Regularien“. In Regularien werden bestimmte Anforderungen an Automobile gestellt. Innerhalb eines Projekts zur Wirkkettenmodellierung mit einem deutschen Automobilhersteller wurde von Praktikern berichtet, dass Regularien aktuell lediglich als Klasse „Regularie“ mit den Instanzen auf Regularien-Ebene – z. B. „UN/ECE R26“ – angelegt werden. Dieser niedrige Detailgrad wurde aufgrund des Aufwands zur Extrahierung und Klassifizierung der Anforderungen aus den Regularien gewählt. Falls eine technische Änderung auftritt, ist nicht direkt ersichtlich, ob die Anforderungen der Regularien weiterhin erfüllt sind. Ein Mehrwert für die Produktentwicklung besteht darin, die Anforderungen, welche in den Regularien dokumentiert sind, direkt mit den Entwicklungsaufgaben zu verknüpfen. Außerdem ist eine Verknüpfung der Anforderungen aus den Regularien mit Entwicklungsaufgaben notwendig, um effizient zu prüfen, ob sämtliche Anforderungen in der Entwicklung des Automobils berücksichtigt worden sind. Die Elemente der Wirkkette entlang RFLPV müssen die Anforderungen erfüllen und sind mit ihnen verknüpft (in UML/SysML: <<satisfy>>). Durch eine Verknüpfung der Anforderungen aus Regularien mit den Anforderungen innerhalb der Wirkkette werden aufgrund der durchgängigen Modellierung die Auswirkungen von technischen Änderungen deutlich. Zur Verknüpfung der Anforderungen innerhalb der Wirkkette mit den Anforderungen aus Regularien müssen Abhängigkeiten dieser Anforderungen bekannt sein. Aufgrund der hohen Anzahl an Regularien und deren Anforderungen sowie der hohen Anzahl an Anforderungen innerhalb der Wirkkette, ist eine manuelle Abhängigkeitsanalyse von Anforderungen aufwändig.

A2.2 Nutzwertanalyse

Tabelle A-2: Nutzwertanalyse zur Auswahl von Ansätzen zur Abhängigkeitsanalyse [GPO20]

Legende

● Erfüllt (1) ○ Teilweise erfüllt (0,5) ○ Nicht erfüllt (0)

ML = Maschinelles Lernen Diff. = Differenziert S. diff. = Semi-differenziert Man. = Manuell

A2.3 ARCA Projekt

1) Problemanalyse und Ermittlung von Anforderungen an die Methode

Aufbauend auf Erkenntnissen aus Projekten des Lehrstuhls für Produktentstehung zum Risikomanagement von Anforderungsänderungen sowie Literaturstudien zum technischen Änderungsmanagement wurden gemeinsam mit acht Industrievertretern der IAV GmbH (Abteilungsleiter, Projektleiter, Teamleiter, Prozess-Manager, Anforderungsingenieur und Entwickler) im Rahmen eines Workshops praxisrelevante Probleme im Kontext von Anforderungsänderungen analysiert. Es wurde bestätigt, dass Anforderungsänderungen häufig unsystematisch gehandhabt werden und Änderungentscheidungen nicht ausreichend unterstützt werden. Außerdem werden Anforderungsabhängigkeiten in der industriellen Praxis aufgrund des hohen Aufwands nicht identifiziert, sodass eine Propagationsanalyse nicht möglich ist [GOP22, GOS20]. Auf Basis der Problemanalyse wurden Anforderungen an die zu entwickelnde Methode ermittelt und mit den Industrievertretern abgestimmt, z. B.: „Die Methode muss die Analyse von Propagationseffekten einer Anforderungsänderung ermöglichen“.

2) Entwicklung einer Methode für das proaktive Management von Anforderungsänderungen

Aufbauend auf den Ergebnissen aus 1) wurde ein Lösungskonzept erarbeitet. Anforderungsbeschreibungen sollen hinsichtlich Qualität geprüft werden, damit nachfolgende Schritte durchgeführt werden können. Anforderungsabhängigkeiten sollen erkannt werden, um eine Propagationsanalyse von Änderungen zu ermöglichen. Die Auswirkungen sowie die Wahrscheinlichkeit einer Änderung sollen bewertet werden. Es sollen risikospezifische Maßnahmen proaktiv ausgewählt werden. Für die einzelnen Schritte wurden existierende Lösungsansätze analysiert und darauf aufbauend kontextspezifische Methoden entwickelt [GOH22a, GOP21, GOP22, GPO20].

Die Qualität von Anforderungsbeschreibungen wird geprüft, indem NLP-Techniken zur Vorverarbeitung eingesetzt werden und anschließend eine lexikalische Prüfung nach bestimmten Begriffen durchgeführt wird. Wenn beispielsweise der Begriff „sollte“ in einer Anforderungsbeschreibung erkannt wird, dann deutet dies auf eine Unsicherheit hin. Zur Erkennung von Anforderungsabhängigkeiten werden KI-Methoden zur Klassifizierung eingesetzt. Untersuchungen haben gezeigt, dass BERT-Modelle in dem Kontext eine hohe Leistungsfähigkeit besitzen [GOH22a]. Die Auswirkungen einer Anforderungsänderung werden mit Hilfe eines Leitfadens bewertet. Hierzu werden einem Experten verschiedene geschlossene Fragen präsentiert. Außerdem werden ihm die abhängigen Anforderungen erster und zweiter Ordnung aufgezeigt, sodass er Propagationseffekte einschätzen kann. Auf Basis der Antworten wird eine Punktzahl berechnet, die proportional zu der Höhe der Auswirkungen einer Änderung ist [GOP21].

Die Wahrscheinlichkeit einer Anforderungsänderung wird in exogene und endogene Änderungswahrscheinlichkeit unterteilt. Zur Berechnung der exogenen Änderungswahrscheinlichkeit (Änderung wird durch Ursachen außerhalb des Anforderungsnetzes verursacht; z. B. Gesetzesänderung) werden durch statistische Informationen einer Änderungshistorie vergangener Projekte ermittelt. Die endogene Änderungswahrscheinlichkeit (Änderung wird aufgrund von Propagationseffekten verursacht) wird berechnet, indem der „PageRank“ als Konnektivitätsmaß für Anforderungen verwendet wird [GTO19]. Auf Basis der Risikokennzahlen Auswirkungen und Wahrscheinlichkeit einer Änderung werden dem Anwender Risikosteuerungsmaßnahmen vorgeschlagen. Je nach Ausprägung der Risikokennzahlen kommen verschiedene Risikostrategien und deren spezifische Maßnahmen in Frage. Beispielsweise wird die Strategie „Risiko transferieren“ und zugehörige Risikosteuerungsmaßnahmen (z. B. „Versicherung oder vertragliche Abwälzung der Änderungskosten auf die Kunden“) bei hohen Änderungsauswirkungen einer Anforderung vorgeschlagen [GPO21].

3) Implementierung der Methoden in einem Software-Prototypen

Die entwickelten Methoden wurden in Python implementiert (siehe Bild A-3). Das grafische User-Interface (GUI) wurde in PyQT realisiert. Parallel zur Implementierung wurden die Methoden kontinuierlich evaluiert. Zur Evaluierung der Abhängigkeitsanalyse wurde ein Datensatz generiert. Der Datensatz bestand aus manuell erstellten Entwicklungsdaten eines intelligenten Knickarmroboters (erstellt auf Basis des Open-Source-Projekts "BCN3D Moveo"¹) und Daten aus vier studentischen Entwicklungsprojekten zu Elektrowerkzeugen wie einem Akkuschrauber oder einem Laserentfernungsmeßgeräte (jeweils etwa 20 Anforderungen und die entsprechende Abhängigkeitsmatrix). Nach internen Tests wurde für jedes Lösungselement ein Expertenfeedback der Industrievertreter eingeholt. Die Testanwendungen der Funktionen des Softwareprototyps, monatliche Austauschtreffen und 8 Workshops mit drei Branchenexperten (je 2 h, 3 Teilnehmer: Abteilungsleiter, Anforderungsingenieur und Prozess-Manager) wurden genutzt, um Feedback von Praktikern einzuholen und die Methoden anwendungsnahe zu entwickeln.

¹ <https://www.bcn3d.com/bcn3d-moveo-the-future-of-learning-robotic-arm/>



Bild A-3: Implementierung der Methoden in einem Software-Werkzeug [GOP22]

4) Validierung der Methode in Zusammenarbeit mit Industrievertretern

Die Methoden wurden anhand eines Bewertungsbogens und zwei darauf aufbauender Experteninterviews mit jeweils zwei Experten validiert. Ein Ergebnis war, dass alle Einflüsse von Anforderungsänderungen auf den Projekterfolg sowie die zur Handhabung erforderlichen Informationsbedarfe ermittelt werden konnten. Ein weiteres Ergebnis waren die Verbesserungspotenziale der initial gewählten Lösungskonzepte, anhand derer die Entwicklung der Methodik auf konkrete Bedürfnisse aus der industriellen Praxis ausgerichtet werden konnte. Es zeigte sich beispielsweise, dass manuelle Aktivitäten, die für jede einzelne Anforderung der Anforderungsliste durchzuführen sind (z. B. Zuordnung einer Anforderung zu einem möglichen Änderungsinitiator), als nicht praxistauglich eingeschätzt werden. Zudem wurde erkannt, dass eine Unterstützung bei der Einbindung des proaktiven Änderungsmanagements in die unternehmensspezifischen Entwicklungsprozesse erforderlich ist. Das dritte Ergebnis waren Rahmenbedingungen für das proaktive Änderungsmanagement aus der industriellen Praxis. *Anforderungen an eine Methodik für das proaktive Risikomanagement von Anforderungsänderungen im Kontext komplexer technischer Systeme sind: Verarbeitbarkeit einer hohen Anzahl an Anforderungen sowie von Anforderungen aus unterschiedlichen Disziplinen (1), Berücksichtigung von Propagationseffekten zwischen Anforderungen (2) und Verfügbarkeit der erforderlichen Daten und Informationen in frühen Entwicklungsphasen (3)*

A2.4 ProMaRC-Methodik

Die Bewertung des Änderungsrisikos bildet den Kern von ProMaRC. Sie folgt einem dreistufigen Ansatz: Analyse der Anforderungsabhängigkeit, Bewertung der Auswirkungen von Änderungen und Bewertung der Änderungswahrscheinlichkeit. Während die Schritte Auswirkungsanalyse und Wahrscheinlichkeitsberechnung für die Risikobestimmung erforderlich sind, wird die Abhängigkeitsanalyse aufgrund ihrer Bedeutung als separater Schritt hinzugefügt. Sie bildet die Grundlage für beide Folgebewertungen und hat einen starken Einfluss auf den Anwendungsaufwand und die Genauigkeit der Ergebnisse. Ein BERT Modell wird für die Abhängigkeitsanalyse von Anforderungen verwendet (siehe Bild A-4). [GOP22]

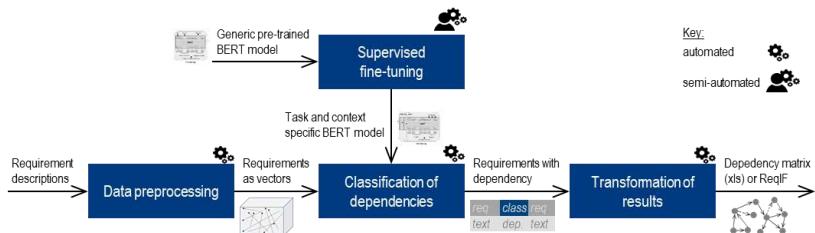


Bild A-4: Abhängigkeitsanalyse von Anforderungen mit BERT [GOP22]

Pre-Training und kontextspezifisches Fine-Tuning

Um Encoder-Decoder-Modelle wie BERT zu trainieren, sind zwei verschiedene Arten des Trainings erforderlich: Pre-Training und kontextspezifisches Fine-Tuning. Das Pre-Training wird durchgeführt mit großen Mengen von Textdaten (z. B. aus Wikipedia und Buchkorpora) durchgeführt. Das Ziel ist es, das Modell zu trainieren, um kontextbezogene Textdaten zu identifizieren. Vortrainierte Modelle sind generisch und können aus Open-Source-Datenbanken (in diesem Fall huggingface) verwendet werden. Nach dem Pre-Training wird das Modell auf die jeweilige Aufgabe und den Kontext abgestimmt. Für Abhängigkeitsanalyse wird eine Klassifizierungsschicht auf die Transformer-Ausgabe gelegt, um damit BERT vordefinierte (Abhängigkeits-)Klassen bestimmen kann. Die Feinabstimmung erfolgt mit gelabelten Textdaten (Fallstudiedaten über einen intelligenten Knickarmroboter). Wie beim Vergleich wurden die Anforderungsdaten durch Zufallsauswahl künstlich angereichert und nach Klassen gewichtet, um Ungleichgewichte im Datensatz auszugleichen. Nach der Feinabstimmung ist ein aufgaben- und kontextspezifisches BERT-Modell für die Abhängigkeitsanalyse verwendbar. [GOP22]

Vorverarbeitung der Daten

Um die Anforderungsdaten in ein geeignetes Datenformat für die Abhängigkeitsanalyse umzuwandeln, ist eine Vorverarbeitung der Daten erforderlich. Die Eingangsdaten (textuelle Anforderungsbeschreibungen) müssen bereinigt und in eine mathematische Darstellung (Vektoren) umgewandelt werden. Zunächst werden die Anforderungen in einen fortlaufenden Text aus zwei Eingabeanforderungen (Textfeld) umgewandelt, dem ein Dummy Label (Label-Feld) ergänzt. Die Dummy-Labels werden später in eine Abhängigkeitsart umgewandelt. Dann wird der Anforderungstext tokenisiert - der Satz wird in einzelne Wörter zerlegt und kategorisiert (Stoppwort vs. Inhaltswort). Hierfür wurde der BERT-Tokenizer verwendet. Zum Schluss, können die Textdaten in eine Vektordarstellung umgewandelt und als Eingabe für das BERT-Modell genutzt werden. [GOP22]

Klassifizierung von Abhängigkeiten

Um Abhängigkeiten zu identifizieren, werden die vorverarbeiteten Eingabedaten mit dem fein abgestimmten BERT-Modell analysiert. Seine Aufgabe ist es, den geschätzten Abhängigkeitstyp zwischen zwei Anforderungen zu bestimmen. Um das Ausbreitungsverhalten zu kennzeichnen, sind die Abhängigkeitstypen "verfeinert", "verfeinert durch", "requires" und "required" von Bedeutung. Aus diesem Grund werden diese vier Typen unterschieden. Die Klassifizierung wird vollautomatisch durchgeführt. Das Ergebnis ist ein Datensatz aus dem Anforderungstext und ihrem Abhängigkeitstyp. [GOP22]

Umwandlung der Ergebnisse

Für die weitere Verwendung der Abhängigkeitsinformationen muss die Ausgabe des BERT-Modells in ein geeignetes Datenformat umgewandelt werden, z. B. xls oder ReqIF. Welches Datenformat am besten geeignet ist, hängt vom Verwendungszweck ab. In diesem Fall wird eine Liste der Anforderungen und ihrer klassifizierten Abhängigkeitstypen im xls-Datenformat erstellt, die für die Auswirkungsabschätzung und das Änderungsmanagement verwendet werden kann. [GOP22]

Bewertung der Auswirkungen von Anforderungsänderungen

Für die Bewertung der Auswirkungen von Anforderungsänderungen müssen die Einflussfaktoren differenziert werden. Ausgangspunkt ist eine erste Anforderungsänderung von einem exogenen Änderungsinitiator (z. B. einem Kunden). Die Umsetzung dieser Änderung führt zu einer Auswirkung, die direkt mit der Änderung der betroffenen Anforderung verbunden ist (lokale Änderungsauswirkung), sendet aber auch nachfolgende Änderungsimpulse an andere Anforderungen aus (konsekutive Änderungsauswirkungen). Diese entstehen aufgrund von Abhängigkeiten zwischen den

Anforderungen und müssen umgesetzt werden, um die Konsistenz und Gültigkeit des Anforderungssatzes zu gewährleisten. Üblicherweise werden Änderungen, die durch konsekutive Effekte verursacht werden, als Änderungspropagation bezeichnet. Durch Aufsummieren der lokalen Änderungsauswirkungen aller Anforderungen auf einem Ausbreitungspfad kann die kollektive Änderungsauswirkung der ursprünglichen Änderungsanforderung bewertet werden. Zur Bewertung der kollektiven Änderungsauswirkungen werden zunächst die aufeinanderfolgenden Auswirkungen ermittelt. Dies erfolgt automatisch auf Basis der Ergebnisse der Abhängigkeitsanalyse. Die Abhängigkeiten werden nach ihrem Typ gewichtet, um das Ausbreitungsverhalten anzuzeigen, und die Anforderungen werden nach ihrer Priorität gewichtet, um die lokalen Änderungsauswirkungen anzuzeigen. In einem zweiten Schritt wird das gewichtete Anforderungsnetz mit Hilfe eines modifizierten PageRank-Algorithmus analysiert, um kritische Anforderungen zu identifizieren. Diese werden ausgewählt und im Hinblick auf kollektive Änderungen bewertet. Auswirkungen manuell von Experten ausgewählt und bewertet. Im Folgenden werden die drei Schritte zur Bewertung der Änderungsauswirkungen im Detail erläutert. [GOP22]

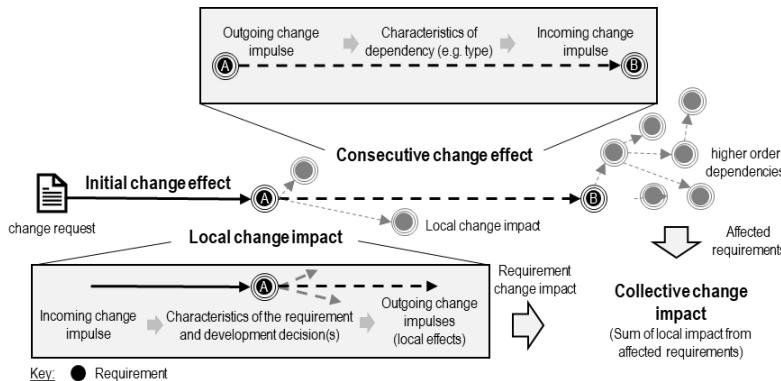


Bild A-5: Analyse von Änderungsauswirkungen [GOP22]

Gewichtung von Abhängigkeiten und Anforderungen

Die Gewichtung der Abhängigkeiten und Anforderungen (siehe Bild A-6) wird in zwei Schritten durchgeführt. Zunächst werden die Abhängigkeiten gewichtet, um die Folgeeffekte einer Änderung der Anforderungen abzuschätzen. Anschließend werden die Anforderungen auf der Grundlage der erwarteten lokalen Auswirkungen gewichtet. Die Bewertung der konsekutiven Auswirkungen einer Anforderungsänderung basiert auf dem Ausbreitungsverhalten Verhalten einer bestimmten Anforderungsabhängigkeit. Das Vorhandensein einer Abhängigkeit ist eine Voraussetzung und bestimmt, ob eine Anforderung von der ursprünglichen Änderung betroffen sein kann. Die Art der Abhängigkeit gibt das Ausbreitungsverhalten zwischen zwei Anforderungen an. Um

Abhängigkeiten zu identifizieren, die zu einer Änderungspropagation führen können, werden Abhängigkeitstypen Typen aus bestehenden Referenzmodellen berücksichtigt. [GOP22]

Es werden drei Kategorien abgeleitet: Zwingende Propagation, mögliche Propagation und keine Propagation werden klassifiziert. Die Unterscheidung basiert auf früheren Studien zum Propagationsverhalten nach Abhängigkeitstypen. Da diese im Kontext der Softwareentwicklung durchgeführt wurden, wurde die Gültigkeit für interdisziplinäre Systeme anhand einer Anforderungsmenge eines Beispielsystems (Drohne) überprüft. Ausgehend von der Annahme, dass alle Abhängigkeiten ein zwingendes Propagationsverhalten haben, wurde versucht, diese Annahme für jede Abhängigkeitsart zu falsifizieren, indem exemplarische Konstellationen ohne obligatorische Propagierung versucht wurden zu finden. Dies wurde wiederholt für nicht obligatorische Propagierung. Die Ergebnisse waren konsistent mit den Erkenntnissen aus der Softwareentwicklung, so dass die folgende Kategorisierung der Abhängigkeitstypen zur Bewertung der aufeinanderfolgenden Auswirkungen einer Anforderungsänderung genutzt wurde [GOP22]:

- Obligatorisches Propagationsverhalten: benötigt, wird benötigt von und wird verfeinert von
- Nicht zwingendes Ausbreitungsverhalten: positive/negative Korrelation
- Kein Ausbreitungsverhalten: verfeinert, Kosten und Wert

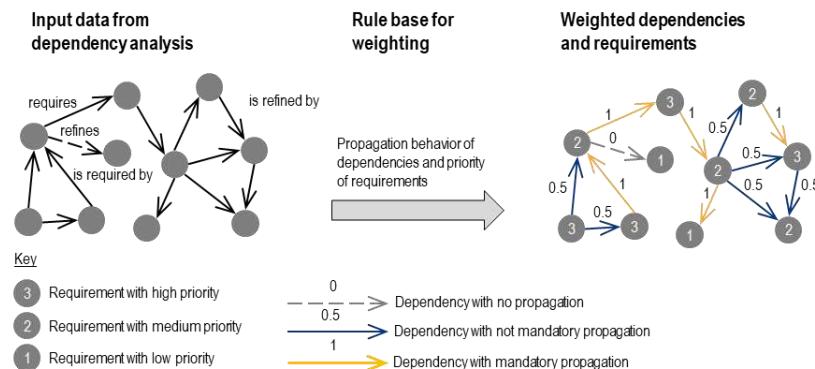


Bild A-6: Gewichtung von Anforderungsabhängigkeiten [GOP22]

Die Gewichtung der Abhängigkeitstypen einer Kategorie basiert auf der Ausbreitungswahrscheinlichkeit. Für die Kategorie obligatorische Propagierung beträgt diese 100% (Gewichtung = 1) und für die Kategorie keine Propagierung 0% (Gewichtung = 0). Die Gewichtung der Kategorie nicht obligatorische Ausbreitung kann kontextspezifisch in Abhängigkeit von der Risikoaffinität oder statischen Daten zur

Ausbreitungswahrscheinlichkeit bestimmt werden. Als Beispiel wird eine Gewichtung von 0,5 verwendet. Die Kategorisierung der Abhängigkeiten nach Ausbreitungsverhalten wird genutzt, um die Kanten im Anforderungsnetz zu gewichten und damit eine Aussage über die Folgewirkungen zu ermöglichen. Im Vergleich zu bestehenden Ansätzen mit binärer Unterscheidung hinsichtlich des Vorhandenseins einer Abhängigkeit erlaubt die Differenzierung der Abhängigkeitstypen eine genauere Bewertung. Das Ergebnis dieses Schrittes sind gewichtete Kanten im Anforderungsnetz als Indikator für konsekutive Auswirkungen einer Anforderungsänderung. [GOP22]

Anhand der gewichteten Kanten (Abhängigkeiten) im Anforderungsnetz können Ausbreitungspfade identifiziert und die darauf liegenden Anforderungen auf lokale Änderungseffekte hin untersucht werden. Vorhersagen über die lokalen Auswirkungen werden zur Gewichtung der Knoten im Anforderungsnetz verwendet. Als Indikator für die lokalen Auswirkungen einer Anforderungsänderung wurde die Anforderungspriorität gewählt. Die Priorität lässt einen Rückschluss auf die Bedeutung für den Projekterfolg zu und hat zwei Vorteile: hohe Verfügbarkeit in den Anforderungsdaten und Fähigkeit zur automatischen Auswertung. Die Alternative ist eine manuelle Bewertung durch Experten. Diese führt vermutlich zu einer höheren Genauigkeit, verursacht aber einen hohen Anwendungsaufwand, da jede Anforderung einzeln bewertet werden muss. Im Rahmen der Vorstudie wurde die manuelle Bewertung von den Anwendern als zu zeitaufwändig für die industrielle Praxis eingeschätzt. Daher wird die Anforderungspriorität verwendet. Je nach unternehmensspezifischem Priorisierungsschema können Gewichtungsfaktoren definiert werden. Da viele Priorisierungsschemata einer dreifachen Struktur folgen (z. B. Begeisterungsmerkmal, Leistungsmerkmal und Basismerkmal), wird beispielhaft eine Differenzierung in hohe (Gewichtung = 3), mittlere (Gewichtung = 2) und niedrige Priorität (Gewichtung = 1) verwendet. Das Ergebnis dieses Schrittes sind gewichtete Knoten im Anforderungsnetz als Indikator für lokale Auswirkungen einer Veränderung. [GOP22]

Expertenbewertung der kollektiven Änderungsauswirkungen einer Anforderung

Die Analyse des Anforderungsnetzes dient der vollständigen Bewertung der Änderungsauswirkungen aller Anforderungen sowie der Auswahl der kritischen Anforderungen (siehe Bild A-7). Diese Auswahl ist notwendig, um den Anwendungsaufwand für die Expertenbewertung der kollektiven Änderungsauswirkungen auf ein praktikables Maß zu reduzieren. Die Bewertung der kollektiven Auswirkungen kann nicht automatisch erfolgen, da eine Betrachtung der Anforderungsinhalte ebenso notwendig ist wie die Einbeziehung kontextspezifischer Einflussgrößen (z. B. Unternehmensstruktur, vertragliche Vereinbarungen oder Restriktionen aus der Produktlebenszyklusphase Realisierung/Produktion). Zur Ermittlung der kollektiven Änderungsauswirkungen einer Anforderung werden sowohl die Ergebnisse der Analyse des Anforderungsnetzes als auch ein zusätzlicher Leitfaden

zur Identifikation betroffener Entwicklungsaktivitäten herangezogen. Der Leitfaden wurde in Anlehnung an die Systems-Engineering-Prozesslandschaft entwickelt und umfasst alle potenziell von einer Änderung betroffenen Aktivitäten. Dies gewährleistet die Vollständigkeit der Analyse und reduziert die Subjektivität der Bewertung. Sowohl die Vollständigkeit als auch die Subjektivität der Bewertung sind Defizite bestehender Methoden, da die Anwender bei ihrer Bewertung nicht unterstützt werden. Der Leitfaden ist in vier Kategorien gegliedert: Architektur und Design, Integration und Implementierung, Verifikation und Validierung und Sonstiges. Da Anforderungsmanagementprozesse immer von einer Anforderungsänderung betroffen sind, wurde diese Kategorie nicht aufgeführt. Jeder Kategorie sind eine übergeordnete Frage ("Wirkt sich eine Änderung dieser Anforderung auf diese Kategorie aus?") und Detailfragen zugeordnet, mit denen sich die Auswirkungen auf bestimmte Entwicklungsaktivitäten differenzieren lassen. Um die allgemeine Praktikabilität des Leitfadens zu gewährleisten, wurde er mit drei Industrievertretern diskutiert und als vollständig und praxistauglich beurteilt. [GOP22]

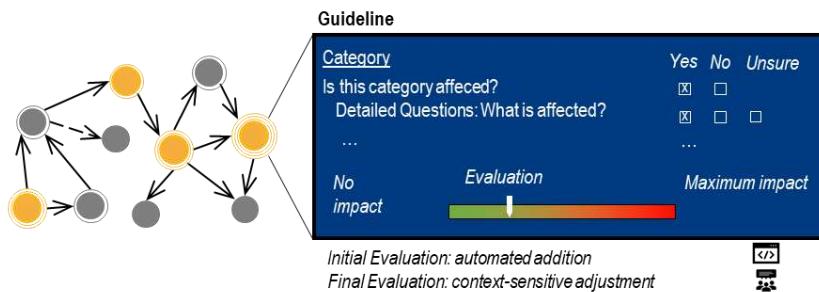


Bild A-7: Bewertung von Änderungsauswirkungen einer Anforderung mittels Leitfaden [GOP22]

Wahrscheinlichkeit der Anforderungsänderung

Die Anforderungsänderungswahrscheinlichkeit setzt sich aus der exogenen und endogenen Änderungswahrscheinlichkeit zusammen. Exogene Änderungsauslöser lösen eine erste Anforderungsänderung aus. Diese haben nichts mit dem Zusammenspiel der Anforderungen zu tun und kommen von außerhalb des Anforderungssatzes. Endogene Änderungen können nur auftreten, nachdem eine initiale Änderung ausgelöst wurde und entstehen durch Ausbreitungseffekte zwischen Anforderungen. [GOP22]

Zur Ermittlung der Anforderungsänderungswahrscheinlichkeit werden exogene und endogene Änderungswahrscheinlichkeiten berechnet und zu einer Gesamtänderungswahrscheinlichkeit zusammengeführt (siehe Bild A-8). Zur Berechnung der exogenen Änderungswahrscheinlichkeit werden die historischen Änderungsdaten analysiert. Um die Gültigkeit der historischen Änderungsdaten für die

vorliegende Anforderungsmenge sicherzustellen, werden diese auf äquivalente Sachverhalte eingegrenzt. Dies geschieht durch die manuelle Zuordnung von relevanten Änderungsinitiatoren sowie Referenzprojekten mit ähnlichen Merkmalen. Die endogene Änderungswahrscheinlichkeit kann auf Basis der Anforderungsabhängigkeiten und deren Ausbreitungsverhalten berechnet werden. Dieser Schritt ist vollständig automatisiert und basiert auf einem modifizierten PageRank-Algorithmus. [GOP22]

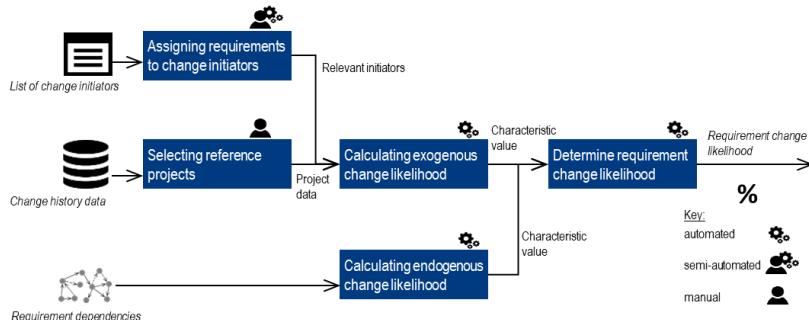


Bild A-8: Bewertung der Änderungswahrscheinlichkeit von Anforderungen [GOP22]

Auswahl proaktiver Risikosteuerungsmaßnahmen

Die Methode zur Auswahl der proaktiven Maßnahmen erfolgt in drei Schritten (siehe Bild A-9). Zunächst werden die Anforderungen auf Basis eines Risikoprofils priorisiert und ein Anforderungsprofil mit Risikoparametern erstellt. Anschließend werden jeder Anforderung eine Handlungsstrategie und zugehörige Maßnahmenvorschläge zugeordnet. Daraus wählen die Anwender die bevorzugte proaktive Maßnahme aus und setzen sie im Rahmen des übergeordneten Change-Managements um. [GOP22]



Bild A-9: Schritte zur Auswahl proaktiver Risikosteuerungsmaßnahmen [GOP22]

Priorisierung der Anforderungen

Zur Priorisierung der Anforderungen wird ein Risikoportfolio mit den Dimensionen Änderungswahrscheinlichkeit und Änderungsauswirkung durch Ausbreitung verwendet (siehe Bild A-10). Ein Risikoportfolio als Priorisierungstechnik wurde gewählt, weil es in der Praxis üblich ist, leicht interpretiert und für die unternehmensweite

Kommunikation genutzt werden kann und einen relativen Vergleich zwischen Anforderungen ermöglicht. [GOP22]

In Abweichung von einem konventionellen Risikoportfolio wurde ein dritter Aspekt hinzugefügt: Die kollektiven Auswirkungen von Änderungen werden durch den Durchmesser einer Anforderung dargestellt. Die dritte Dimension ist erforderlich, um den Anforderungssatz vollständig darzustellen und gleichzeitig alle Ergebnisse der Risikobewertung zu zeigen. Während die Änderungswahrscheinlichkeit für alle Anforderungen ermittelt wird, wird die kollektive Änderungsauswirkung nur für kritische Anforderungen bewertet. Daher wird für die Dimension des Risikoportfolios die Ausbreitungswirkung verwendet. Um die Anforderungen zu priorisieren, wird für jede Portfoliodimension ein Schwellenwert definiert. Auf der Grundlage der Schwellenwerte wird das Portfolio in vier Quadranten unterteilt, denen eine Risikokategorie (hoch, mittel und niedrig) zugeordnet wird. Die Anforderungen werden entsprechend dem zugehörigen Quadranten kategorisiert und priorisiert. Die Schwellenwerte können je nach abhängig von der Risikoaffinität angepasst werden. Als Standard werden Durchschnittswerte verwendet. [GOP22]

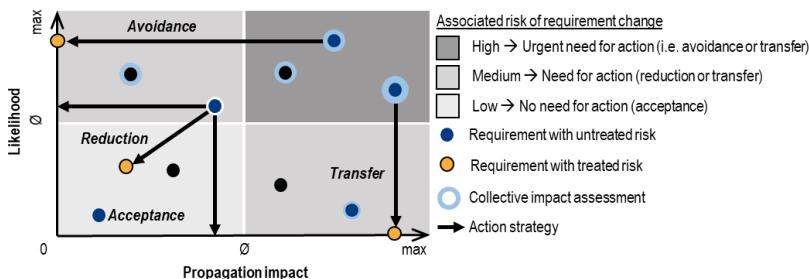


Bild A-10: Risikoportfolio zur Priorisierung von Anforderungen [GOP22]

Definition einer Aktionsstrategie

Auf der Grundlage des Risikoportfolios wird für jede Anforderung eine Aktionsstrategie festgelegt. Es werden vier Handlungsstrategien unterschieden: Vermeidung, Übertragung, Reduzierung und Akzeptanz. [GOP22]

Proaktive Maßnahmen werden je nach beabsichtigter Wirkung einer Handlungsstrategie zugeordnet:

- Vermeidung: Eliminierung der Eintrittswahrscheinlichkeit (z. B. Ablehnung eines Auftrags oder von Änderungswünschen).

- Verringerung: Verringerung der Wahrscheinlichkeit des Auftretens oder der Auswirkungen (z. B. Sicherstellung einer Anforderungsspezifikation mit dem Beteiligten oder Vereinbarung von Aufwandsbeschränkungen)
- Übertragung: Teilen oder Übertragen der Änderungsauswirkungen (z. B. Versicherung oder vertragliche Weitergabe der Änderungskosten an die Kunden).
- Annahme: Keine Aktion und keine Auswirkung.

Je nach Risikokategorie einer Anforderung werden die Handlungsstrategien vorgeschlagen. [GOP22]

Auswahl der proaktiven Maßnahme

Da es bisher keine Forschung zu kontextspezifischen proaktiven Maßnahmen für Anforderungsänderungen gibt, wird eine Übersicht über kontextspezifische Maßnahmen erstellt (siehe Bild A-11). Dazu wurden Maßnahmen aus anderen Kontexten (insbesondere Change- und Risikomanagement) übertragen und angepasst. Im Rahmen eines Workshops mit Branchenexperten (2 h; 3 Teilnehmer: Abteilungsleiter, Requirements Engineer und Process Owner) wurden die Ergebnisse auf ihre Eignung überprüft und angereichert. Dennoch ist die Übersicht als Referenzliste zu verstehen, die keinen Anspruch auf Vollständigkeit erhebt und bei Bedarf angepasst werden kann. Zur Differenzierung der Maßnahmen werden Handlungsstrategie und Zielwert herangezogen. Darüber hinaus wird angegeben, ob die Maßnahmen das Änderungsrisiko einer einzelnen Anforderung oder des Anforderungssatzes adressieren. [GOP22]

<i>Avoidance or reduction of change likelihood</i>	<i>Reduction of change impact</i>	<i>Transfer</i>
<p>Exogenous:</p> <ul style="list-style-type: none"> • Requirement freeze with stakeholders' change cost responsibility (R,S) • Application of established development methods and methodologies (for instance engineering approaches, project management or Capability Maturity Model Integration) (S) • Quality assurance (R,S) <p>Endogenous:</p> <ul style="list-style-type: none"> • Risk optimized decision management (A) • Modularization of components (R,S) • Change likelihood reduction of affected requirements (R) 	<ul style="list-style-type: none"> • Front loading, by: <ul style="list-style-type: none"> • In-Process Validation (R,S) • Feasibility studies (R) • Modified development sequence (R) • Uncertainty management measures (R,S) • Application of established development methods and methodologies (for instance engineering approaches, project management or Capability Maturity Model Integration) (S) • Modularization (S) • Post-negotiation with stakeholders • Cost and time constraints (R) • Requirements (delete, change, re-prioritize) (R) • Elaborate fall-back solution or alternative solution (R) • Plan/increase cost and time buffers (R,S) • Terminate/reject request (S) 	<ul style="list-style-type: none"> • Insurance (S) • Outsourcing (R,S) • Contract design: <ul style="list-style-type: none"> • (agreed on) limitation of working hours (R,S) • Requirement freeze with stakeholders' change cost responsibility (R,S)

Key:

(R) = Proactive measure for an individual requirement
(S) = Proactive measure for a requirement set

Bild A-11: Risikosteuerungsmaßnahmen von Anforderungsänderungen [GOP22]

A2.5 Bewertung von Änderungsauswirkungen von Anforderungen

Kategorie 1: Architektur und Design
Müssten neue Systemelemente entwickelt werden?
Müssten Systemelemente überarbeitet werden?
Müsste die Systemgrenze neu definiert werden?
Musste die Funktionsstruktur überarbeitet werden?
Müsste die Wirkstruktur überarbeitet werden?
Müssten interne Schnittstellen ¹ überarbeitet werden?
Müssten externe Schnittstellen ¹ überarbeitet werden?
Müssten neue Flusse ² berücksichtigt werden?
Müssten neue Stoffflüsse berücksichtigt werden?
Kategorie 2: Implementierung und Integration
Müssten Verfahren zur Implementierung verändert werden (z. B. Fertigung)?
Müssten Verfahren zur Integration verändert werden (z. B. Montage)?
Müssten neue Betriebsmittel beschafft werden?
Müssten zusätzliche Ressourcen ³ beschafft werden?
Kategorie 3: Verifizierung und Validierung
Müssten Verifizierungspläne überarbeitet werden?
Müssten Validierungspläne überarbeitet werden?
Müssten neue Simulationen / Analysen durchgeführt werden?
Müsste ein neuer Prototyp entworfen werden?
Müsste ein Prototyp überarbeitet werden?
Müssten neue Testfälle definiert werden?
Müssten Testfälle überarbeitet werden?
Kategorie 4: Weitere
Müsste neues Personal eingestellt werden?
Müssten Schulungen durchgeführt werden?
Müssten Qualitätsstandards neu überprüft werden?
Müssten Sicherheitsstandards neu überprüft werden?
Müssten Make-or-Buy-Entscheidungen überdacht werden?
Hinweise:
¹ interne Schnittstellen: innerhalb des Systems, externe Schnittstellen: zum Umfeld
² Stoffflüsse, Energiefüsse, Informationsflüsse
³ Betriebsstoffe, Hilfsstoffe, Rohstoffe

Bild A-12: Leitfaden zur Bewertung von Änderungsauswirkungen von Anforderungen

A2.6 Projektcharakteristika zur Auswahl von Vergleichsprojekten

Tabelle A-3: Projektcharakteristika

Frage	Antwortmöglichkeit
Wie hoch ist der Innovationsgrad des zu entwickelnden Systems?	niedrig / mittel / hoch
Wie hoch ist die Wahrscheinlichkeit von Änderungswünschen durch den Kunden?	niedrig / mittel / hoch
Wie hoch ist die Sicherheit politischer Verhältnisse, die Einfluss auf das Entwicklungsprojekt haben?	niedrig / mittel / hoch
Wie hoch ist die Komplexität des Projekts?	niedrig / mittel / hoch
Wie hoch ist die Liefertreue des Lieferanten?	niedrig / mittel / hoch
Wie gut sind wir im Vergleich zu unserer Konkurrenz positioniert?	schlechter / gleichwertig / besser
Sind strukturelle Änderungen beim Kunden zu erwarten?	ja / nein
Sind interne strukturelle Änderungen zu erwarten?	ja / nein
Werden Zu- oder Abgänge von Mitarbeitern im Projekt mit technischem "Know-How" erwartet?	ja / nein
Wenden wir strukturiert Anforderungsmanagement-Methoden im Projekt an?	ja / nein
Findet ein disziplinübergreifender Austausch statt?	ja / nein
Sind Kundenanforderungen eindeutig formuliert?	ja / nein

A2.7 Systemmodellierung

Ein Systemmodell ist eine Informationsquelle, die die Konsistenz zwischen verschiedenen Entwicklungsartefakten, wie Anforderungen, Funktionen oder Test-Cases gewährleistet [Est08]. In Abhängigkeitsmatrizen enthält jede Zelle numerische oder binäre Informationen über die Abhängigkeit zwischen dem betrachteten Artefakt der Zeile und Spalte [CSE04]. Dies führt zu einer exponentiell wachsenden Anzahl von Zellen, die pro hinzugefügtem Artefakt gefüllt werden müssen. Ein Ansatz zur Abbildung von Artefakten und zur Erfassung von Abhängigkeiten ist ein Systemmodell. Ein Systemmodell wird unter der Verwendung von drei Komponenten erstellt: Modellierungsmethode (1), Modellierungssprache (2) und Modellierungswerkzeug (3) [Del14]. Verschiedene Methoden beschreiben das Vorgehen zur Modellierung, zum Beispiel OOSEM [Fri14] und SYSMOD [Wei14]. Zusätzlich können verschiedene Modellierungssprachen zur Beschreibung des Systems verwendet werden, zum Beispiel UML [Obj05] und SysML [Obj19]. Verschiedene IT-Anbieter bieten Tools zur Ausführung von SysML an, zum Beispiel IBM Rhapsody oder Cameo Systems Modeler von Dassault Systems. [GK22]

Die System-Architektur ist Teil des Systemmodells. Die Modellierung der System-Architektur zielt darauf ab, dass das entwickelte System, bzw. Software-Werkzeug die Anforderungen erfüllt und robust gegenüber Änderungen ist [FA02]. In Bezug auf die Entwicklung eines Software-Werkzeugs ist die System-Architektur in zwei Bereiche eingeteilt. Die Infrastrukturarchitektur ist eine holistische, operationale Sicht auf das System [FA02]. Dazu zählt die Beschaffenheit des technischen Systems, bzw. die Hardware (zur Ausführung der Software) und beschreibt auch das Management des technischen Systems (Kapazitätsplanung, Softwareverteilung, Datensicherung und Wiederanlauf) [Sch21]. Die Software-Architektur beschreibt die funktionale Sicht auf die Software. Hierzu zählt die Struktur und Aufteilung von Softwarekomponenten, sowie Schnittstellen [FA02]. Relevante Diagramme zur Modellierung der Software-Architektur sind: Anforderungsdiagramme, Use-Case-Diagramme, Aktivitätsdiagramme, Klassendiagramme und Komponentendiagramme [Wei14]. Anforderungsdiagramme werden speziell in SysML verwendet. Es werden die Anforderungen an ein System mit ihren Relationen untereinander (horizontal) sowie zu weiteren Modellelementen (vertikal) dargestellt. Ein Beispiel für eine horizontale Relation ist „Derive Requirement Relationship“: eine Anforderung wird aus einer anderen Anforderung abgeleitet. Im Systemmodell wird sie durch <<deriveRqt>> abgebildet. Ein Beispiel für eine vertikale Relation ist „Verify“: die Anforderung wird durch den verknüpften Test-Case verifiziert. Use Case Diagramme beschreiben das Verhalten eines Systems durch Interaktionen mit einem Akteur. Innerhalb eines Aktivitätsdiagramms werden Elemente, welche Aktivitäten des Systems repräsentieren, mittels Kontroll- und Datenflüssen verknüpft. Klassendiagramme beschreiben die Struktur und das Verhalten von Objekten. Das Komponentendiagramm stellt die Schnittstellen zwischen Komponenten des Systems durch Relationen und Konnektoren dar. [Wei14]

A3 Aktivitäten zur Einbindung von Industrieanwendern

Die Übersicht entwicklungsbegleitender Aktivitäten zur Einbindung von Industrievertretern listet alle explizit durchgeführten Aktivitäten zur Einbindung von Industrievertretern auf, die zur Untersuchung des Anwendungszusammenhangs, der Anforderungserhebung sowie der Entwicklung und Validierung der Methodik durchgeführt wurden. Die Aktivitäten und Unterlagen werden für die Projekte ARCA und BIKINI aufgeführt. Aufgrund von Geheimhaltungsvereinbarungen werden die Aktivitäten und Unterlagen für das ZWM Projekt nicht aufgeführt.

A3.1 ARCA: Rahmendaten von Aktivitäten und Unterlagen

Tabelle A-4: Aktivitäten zur Einbindung von Industrievertretern in ARCA

ID	Rahmendaten
A-W1	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: initiale Anforderungserhebung -Teilnehmer: Abteilungsleiter, Projektleiter, Teamleiter, Prozess- / Tool-Verantwortliche, Anforderungsmanager, Entwickler und Fachreferent -Dauer: 180 Minuten -Zeitpunkt: 02.2020
A-W2	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Spezifizierung und Priorisierung der Anforderungen; Definition von Use-Cases -Teilnehmer: siehe A-W1 -Dauer: ca. 120 Minuten -Zeitpunkt: 04.2020
A-R1	<ul style="list-style-type: none"> -Art: Regelaustausch (ca. monatlich) -Beschreibung: insbesondere Diskussion von Zwischenergebnissen hinsichtlich Eignung und Praxistauglichkeit und Charakterisierung der zu berücksichtigenden Rahmenbedingungen -Teilnehmer: in der Regel Abteilungsleiter, Anforderungsmanagerin, Prozessverantwortlicher und ein bis drei wechselnde Anwender -Dauer: 30-60 Minuten -Zeitpunkt: 01.2020 bis 12.2021
A-W3	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Finalisierung Use-Cases -Teilnehmer: Anforderungsmanagerin und Prozessverantwortlicher (Summe: 2 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 04.2020

ID	Rahmendaten
A-W4	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Diskussion der Praxistauglichkeit des Risikoportfolios -Teilnehmer: Anforderungsmanagerin und Prozessverantwortlicher (Summe: 2 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 09.2020
A-W5	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Diskussion und Anreicherung der Liste mit Risikosteuerungsmaßnahmen und Maßnahmen zur Identifikation von Unsicherheiten -Teilnehmer: Anforderungsmanagerin und Prozessverantwortlicher (Summe: 2 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 11.2020
A-W6	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Erfassung der Ist-Prozesse und Definition der Soll-Prozesse bei Anwendung der Methodik (Änderungs- und Risikomanagement), Erfassung von Software-Landschaft und Definition erforderlicher Schnittstellen Teilnehmer: Abteilungsleiter, Anforderungsmanagerin, Prozessverantwortlicher, Fachreferent und Anwenderin (Summe: 5 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 12.2020
A-W7	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Validierungsplanung (Anforderungen, verfügbare Daten, Rahmenbedingungen) -Teilnehmer: zwei Abteilungsleiter, Anforderungsmanagerin, Prozessverantwortlicher, Fachreferent und Anwenderin (Summe: 6 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 01.2021
A-W8	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Validierung der Methode zur Bewertung der Änderungsauswirkungen und prototypische Umsetzung -Teilnehmer: Anforderungsmanagerin, Prozessverantwortlicher und Anwenderin (Summe: 3 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 03.2021

ID	Rahmendaten
A-W9	<p>-Art: Workshop</p> <p>-Beschreibung: Validierung der Methode zur Bewertung der Änderungswahrscheinlichkeit und prototypische Umsetzung</p> <p>-Teilnehmer: Abteilungsleiter, Anforderungsmanagerin, Prozessverantwortlicher, Fachreferent und Anwenderin (Summe: 5 Teilnehmer)</p> <p>-Dauer: 120 Minuten</p> <p>-Zeitpunkt: 04.2021</p>
A-W10	<p>-Art: Workshop</p> <p>-Beschreibung: Validierung der Methode zur Analyse von Anforderungsabhängigkeiten und dessen prototypische Umsetzung</p> <p>-Teilnehmer: Anforderungsmanagerin und Prozessverantwortlicher (Summe: 2 Teilnehmer)</p> <p>-Dauer: 120 Minuten</p> <p>-Zeitpunkt: 05.2021</p>
A-W11	<p>-Art: Workshop</p> <p>-Beschreibung: Validierung der Methode zur Risikosteuerung und prototypische Umsetzung sowie der vorgesehenen Lernmechanismen</p> <p>-Teilnehmer: Abteilungsleiter, Anforderungsmanagerin und Prozessverantwortlicher (Summe: 3 Teilnehmer)</p> <p>-Dauer: 120 Minuten</p> <p>-Zeitpunkt: 06.2021</p>
A-W12	<p>-Art: Workshop</p> <p>-Beschreibung: Validierung des Referenzprozesses zur Einführung und Anwendung der Methodik</p> <p>-Teilnehmer: Abteilungsleiter, Anforderungsmanagerin und Prozessverantwortlicher (Summe: 3 Teilnehmer)</p> <p>-Dauer: 90 Minuten</p> <p>-Zeitpunkt: 11.2021</p>
A-F1	<p>-Art: Bewertungsbogen</p> <p>-Beschreibung: Validierung der prototypischen Lösung</p> <p>-Teilnehmer: Anforderungsmanagerin, Prozessverantwortlicher, Teamleiter und Entwicklerin (Summe: 4 Teilnehmer)</p> <p>-Dauer: ca. 30 Minuten</p> <p>-Zeitpunkt: 11.2021 bis 12.2021</p>
A-I1	<p>-Art: semi-strukturierte Experteninterviews je Industriepartner (3 Stück)</p> <p>-Beschreibung: Detaillierung und Anreicherung der Bewertungsbogenergebnisse Teilnehmer: siehe A-F1</p> <p>-Dauer: 30-45 Minuten</p> <p>-Zeitpunkt: 12.2021</p>

A-W1

Fragestellung: Wie kann das ARCA Projekt für Sie einen Mehrwert bringen? [Ole22]

- Bei welchen Problemen aus meinem Arbeitsalltag könnte mir das Projekt helfen? (rot – vgl. nachfolgende Tabellen)
- Welche Anforderungen sind zu beachten, damit die Ergebnisse für mich nützlich sind? (grün – vgl. nachfolgende Tabellen)
- Ist das Projekt interessant für mich? Auf welche Art und Weise könnte ich mir vorstellen an dem Projekt mitzuwirken? (orange – vgl. nachfolgende Tabellen)

Position	Abteilungsleiter	Anforderungsmanagerin, Product Ownerin	Teamleiter, Anforderungsmanager	Fachreferent „Software“ Architektur
Name, Vorname	Anonymisiert 1	Anonymisiert 2	Anonymisiert 3	Anonymisiert 4
grüne Karte	offene Schnittstellen (in-export), adaptive Algorithmen die „mitlernen“, anschauliche Darstellungen der Ergebnisse	Prognose für Änderungswahrscheinlichkeit der Anforderungen, Änderungsauswirkungen erfahren (Referenzen grafisch aufarbeiten)	modulare und automatisierbare Schnittstellen, Möglichkeit zur Protokollierung und Reporting des Prozesses	Schnittstelle zu gängigen Anforderungstools, Anwendbarkeit in typischen Projekten
orange Karte	beratend Management- bzw. Unternehmensperspektive „Öl-Kännchen“	Review der Anforderungen an das Tool, bestehende unterstützende bei der Methode, Interesse für Interviews und Fragebögen zur Verfügung stehen	Stakeholder / Endanwender Knowhow zu Software Agilität Anforderungsmanagement	Austausch zu Normen und Unternehmens-Vorgaben
rote Karte	Abhängigkeitsanalyse, Handlungsempfehlungen		Systematik bei Evaluierung von Anforderungen und Änderungen	Umsetzung Standardprozesse in der Praxis (Risikobewertung, Anforderungen)

Bild A-13: Workshopergebnisse Teil 1

Position	funktionale Sicherheit	Toolkette	anforderungsba-siertes Testen	Projektleiter, Product Owner
Name, Vorname	Anonymisiert 5	Anonymisiert 6	Anonymisiert 7	Anonymisiert 8
grüne Karte	Abhängigkeitsanalyse für Mechatronik / Elektronik wie auch Funktion / Software	gute Dokumentation auch des Entscheidungs- / Entwicklungsprozesses, modularisierte Software, bei der wir ggf. später Module erweitern / austauschen könnten, liefert „richtige“ Ergebnisse	-	Einbindung in etablierte ALM-TOOLS (Azure, Jira, ...) möglich
orange Karte	Methoden / Toolgestützte Entscheidungsoptionen Fachliche / Kaufmännische Mechanik / Elektronik	technische Grundlage aus Sicht „Informatik für mich interessant, Schnittstelle interessant, semantische Auswertung durch IT lässt sich auf andere Probleme übertragen“	-	Anforderungen an die Anwendungen erheben Testnutzer
rote Karte	Risikoeinschätzung für Anforderungsabhängigkeit von der Entwicklungsphase	Grundlage für Aufgabenstellung durch Vorgesetzte verbessern (aus Sicht als Implementierer)	Herausfiltern von „schlechten“ Anforderungen (nicht testbar, unvollständig)	Anforderungspflege in laufenden Projekten vereinfachen, #lebende Dokumentation, Risiko- und Change-management in sehr kleinen Projekten ohne Projektleiter todsüchtig ohne „viel“ Ausbildung der MA

Bild A-14: Workshopergebnisse Teil 2

A-W2

Anforderungen an das Projektergebnis

Priorisierung



- Priorisierung nach Balzert, H.: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*, Spektrum Akademischer Verlag, Heidelberg, 2009
- Priorisierung nach dem Kriterium **Notwendigkeit** mit den folgenden Ausprägungen:
 - **Essenziell (essential)**: Die Software wird nicht akzeptiert, wenn die Anforderung nicht in der geforderten Weise realisiert wird. (*Prio. 1*)
 - **Bedingt notwendig (conditional)**: Die Anforderung wertet die Software auf. Wird sie nicht realisiert, dann wird die Software aber nicht unakzeptabel. (*Prio. 2*)
 - **Optional (optional)**: Die Realisierung der Anforderung kann wertvoll sein, muss es aber nicht. Die Anforderung gibt dem Auftragnehmer die Möglichkeit, über die vorhandenen Anforderungen hinauszugehen. (*Prio. 3*)

Bild A-15: Priorisierungskonzept für Anforderungen

Anforderungen an das Projektergebnis

Stakeholder-Anforderungen



Funktionale Anforderungen – Eingaben

A₁: Der Support soll für den Anwender eine Funktion bereitstellen, um die Beweggründe für eine Anforderungsänderung für eine Anforderung zu notieren. (*Prio. 3 - SdF*)

A₂: Anforderungen aller Granularitätsstufen/Hierarchieebenen sollen mit dem Support verarbeitet werden können (ggf. sind bestimmte Vorinformationen zur Hierarchieebene der Anforderung erforderlich). (*Prio. 2 – Neu & SdF*)

A₃: Die Datengrundlage des Supports für die Risikoberechnung soll projektspezifisch vom Anwender angepasst werden können. (*Prio. 2 - Neu*)

Bild A-16: Funktionale Anforderungen an die Eingaben

Anforderungen an das Projektergebnis

Stakeholder-Anforderungen



Funktionale Anforderungen – Verarbeitungsschritte

A₄: Der Support soll unternehmensspezifisches Expertenwissen (tbc) mit einbeziehen. **(Prio. 2 – Neu & SdF)**

A₅: Verfügbare Daten und Informationen aus existierenden Datenbanken (tbc – z. B. Änderungsdaten) sollen genutzt werden. **(Prio. 3 – Neu & SdF)**

A₆: Das Expertenwissen der Anwender soll in der Datengrundlage abgebildet werden. **(Prio. 2 – Neu & SdF)**

A₇: Als unsicher eingeschätzte Ergebnisse sollen dem Anwender nicht für die Entscheidungsfindung bereitgestellt werden. **(Prio. 2 - Neu)**

A₈: Das Änderungsrisiko soll automatisiert vom Support berechnet werden. **(Prio. 1 - Neu)**

A₉: Die Änderungswahrscheinlichkeiten von Anforderungen sollen abgeschätzt werden. **(Prio. 1 - Neu)**

Bild A-17: Funktionale Anforderungen an die Verarbeitung (I/II)

Anforderungen an das Projektergebnis

Stakeholder-Anforderungen



Funktionale Anforderungen – Verarbeitungsschritte

A₁₀: Die Auswirkungen (tbc) von Anforderungsänderungen sollen abgeschätzt werden. **(Prio. 1 – Neu & SdF)**

A₁₁: Der Support soll auf aktuellen wissenschaftlichen Methoden aufbauen. **(Prio. 1)**

A₁₂: Der Support soll die Qualität (tbc) von Anforderungen bewerten. **(Prio. 3 - SdF)**

Bild A-18: Funktionale Anforderungen an die Verarbeitung (II/II)

Anforderungen an das Projektergebnis

Stakeholder-Anforderungen



Funktionale Anforderungen – Ausgaben

A₁₃: Es sollen Risikosteuerungsmaßnahmen vorgeschlagen werden.
(Prio. 1 - SdF)

A₁₄: Der Anwender soll durch den Support bei der Auswahl von Risikosteuerungsmaßnahmen unterstützt werden (z.B. Leitfaden).
(Prio. 2 - Neu)

A₁₅: Die Unterstützung bei der Auswahl von Risikosteuerungsmaßnahmen soll zunehmend genauer werden (Lerneffekt). *(Prio. 3 - Neu)*

A₁₆: Es soll pro ausgegebenes Ergebnis (tbc) angegeben werden, wie verlässlich/genau das Ergebnis vom Support prognostiziert wird. *(Prio. 3 - Neu)*

Bild A-19: Funktionale Anforderungen an die Ausgabe (I/II)

Anforderungen an das Projektergebnis

Stakeholder-Anforderungen



Funktionale Anforderungen – Ausgaben

A₁₇: Umgesetzte Risikosteuerungsmaßnahmen sollen in Relation zu den betroffenen Anforderungen/Änderungen im Support angezeigt werden können.
(Prio. 3 - SdF)

A₁₈: Im Support soll angezeigt werden, welche Person eine Anforderungsänderung vorgenommen hat. *(Prio. 2 - SdF)*

A₁₉: Im Support soll angezeigt werden, welche Person welche Risikosteuerungsmaßnahme für eine Anforderung ausgewählt hat. *(Prio. 3 - SdF)*

A₂₀: Im Support soll eine Änderungshistorie (tbc) für jede Anforderung dokumentiert werden. *(Prio. 2 - SdF)*

A₂₁: Die vom Support bereitgestellte Informationsgrundlage soll bei einer Entscheidung dokumentiert werden. *(Prio. 2 - SdF)*

Bild A-20: Funktionale Anforderungen an die Ausgabe (II/II)

Anforderungen an das Projektergebnis

Stakeholder-Anforderungen



Nicht-Funktionale Anforderungen – Benutzbarkeit

A₂₂: Das Nutzerinterface soll intuitiv bedienbar sein. **(Prio. 2)**

A₂₃: Möglichkeit zur eigenständigen Einarbeitung in den Support soll vorhanden sein. **(Prio. 2)**

A₂₄: Die Ergebnisse des Supports und deren Herleitungen müssen für den Anwender nachvollziehbar dargestellt werden. **(Prio. 1 - Neu)**

Bild A-21: Nicht-funktionale Anforderungen an die Benutzbarkeit

Anforderungen an das Projektergebnis

Stakeholder-Anforderungen



Nicht-Funktionale Anforderungen – Realisierung

A₂₅: Datenaustausch zwischen dem Support und den Tools der Anwender (tbc) soll ohne manuellen Aufwand möglich sein. **(Prio. 2 - Neu)**

A₂₆: Der Support soll projektübergreifend anwendbar sein. **(Prio. 1 - Neu)**

A₂₇: Die Funktionalität des Supports soll unabhängig von Software-Produkten sein (Stand-Alone Lösung). **(Prio. 1)**

Nicht-Funktionale Anforderungen – Einführung, Nutzung und Betreuung

A₂₈: Relativ zum Projektumfang soll der Anwendungsaufwand gering sein (tbc). **(Prio. 1)**

Bild A-22: Nicht-funktionale Anforderungen an die Realisierung

A-W3

Detailierung der Use-Cases (I/III)

Rolle des Entwickler/in

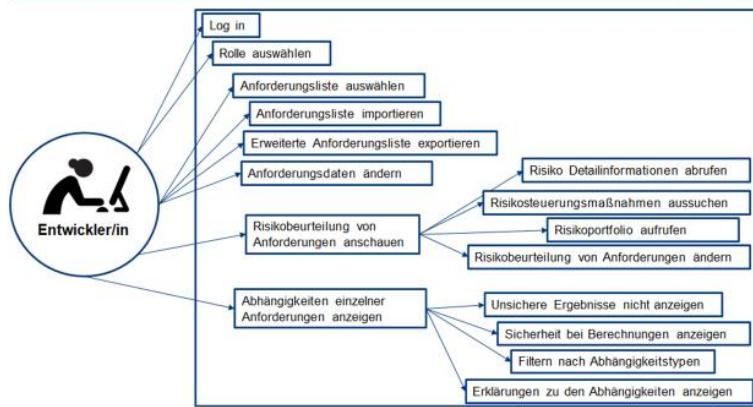


Bild A-23: Use-Case Entwickler/in

Detailierung der Use-Cases (II/III)

Rolle des Anforderungsmanager/in

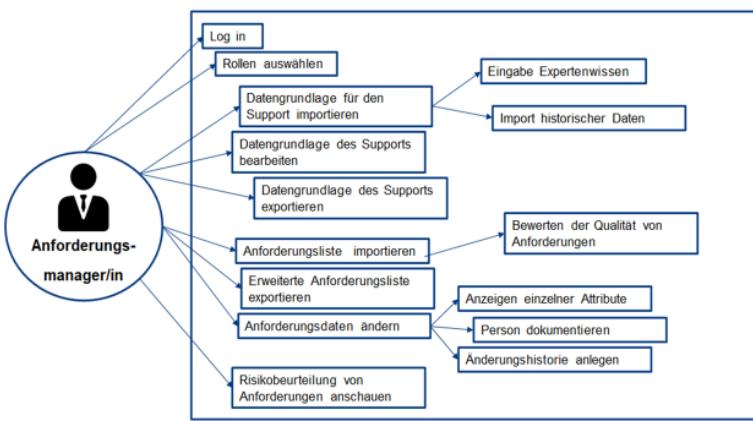
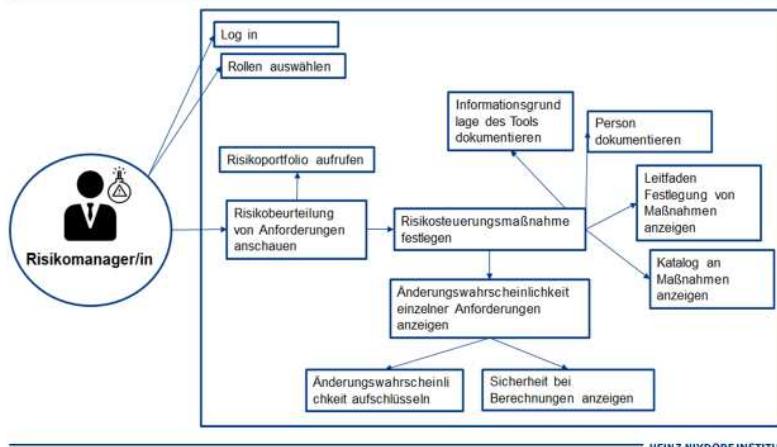


Bild A-24: Use-Case Anforderungsmanager/in

Detailierung der Use-Cases (III/III)

Rolle des Risikomanager/in



13 Lehrstuhl für Produktentstehung

Univ.-Prof. Dr.-Ing. Iris Graßler

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Bild A-25: Use-Case Risikomanager/in

A-W4

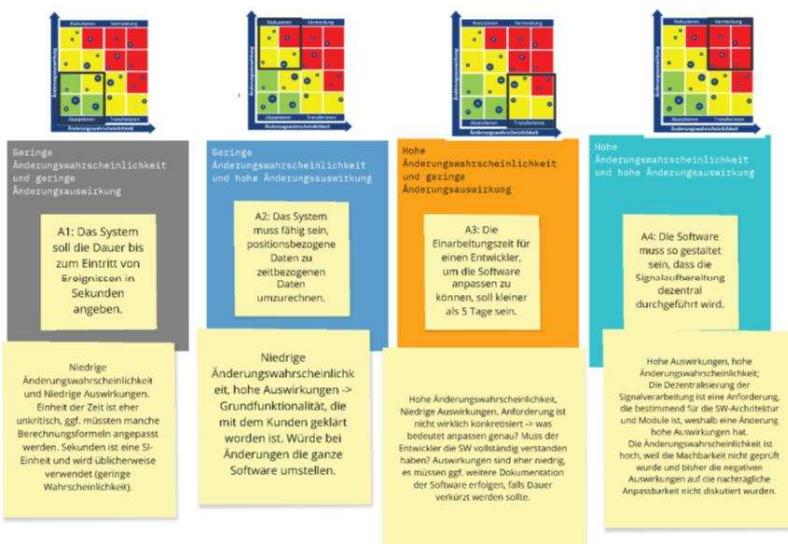


Bild A-26: Auszug aus Workshop-Unterlagen (Miro-Board)

A-W6

Aufgrund von Geheimhaltungsvereinbarungen sind diese Unterlagen nicht öffentlich.

A-W7

Aufgrund von Geheimhaltungsvereinbarungen sind diese Unterlagen nicht öffentlich.

A-W8

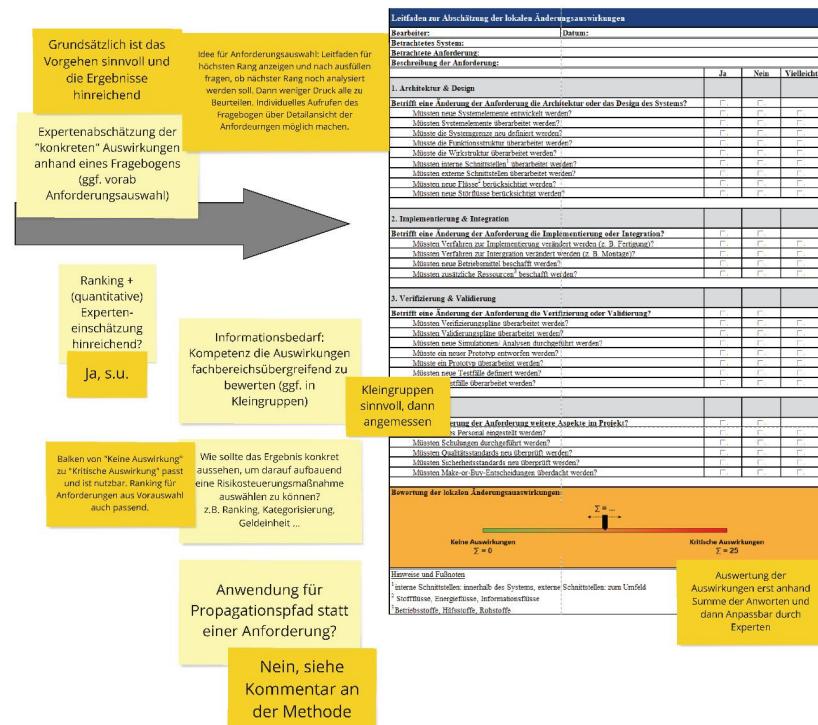


Bild A-27: Auszug aus Workshop-Unterlagen zur Bewertung der kollektiven Auswirkungen einer Anforderungsänderung

A-W9

Problemstellung und Zielsetzung

Um die Wahrscheinlichkeit einer Anforderungsänderung abzuschätzen, werden Informationen benötigt um

- ... die Einflüsse durch exogene Änderungsimpulse zu bestimmen. Diese Änderungsimpulse werden durch Initiatoren wie Politik, Lieferant oder Markt verursacht. **Benötigte Informationen:** Ein Maß, welches die Wahrscheinlichkeit, dass sich eine Anforderung durch einen Änderungsimpuls ändert, in einem definierten Maßstab abbildet.
- ... den Einfluss endogener Ursachen zu bestimmen. Endogene Anforderungsänderungen entstehen aufgrund von Propagationseffekten. **Benötigte Informationen:** Ein Maß, welches die Konnektivität des Anforderungssets abbildet, sodass die Wahrscheinlichkeit einer Änderung aufgrund von Propagationseffekten abgeschätzt werden kann.

Zur Berechnung der Änderungswahrscheinlichkeit müssen diese Informationen (1. Exogene Änderungswahrscheinlichkeit / 2. Endogene Änderungswahrscheinlichkeit / Komplexität) zu einer Gesamt-Berechnung zusammengeführt werden.

Berechnung: Maß für die exogene Änderungswahrscheinlichkeit

Um die exogene Änderungswahrscheinlichkeit abzuschätzen, werden historische Daten von **Referenzprojekten** verwendet.

- Referenzprojekte sind Projekte, die ähnliche Charakteristika wie das aktuelle Projekt besitzen und ähnlich von den Initiatoren einer Änderung beeinflusst werden.
- Die Identifikation von Referenzprojekten wird durch das Ausfüllen eines Fragebogens unterstützt. Zusätzlich kann manuell eine Auswahl getroffen werden.

Berechnung: Maß für die endogene Änderungswahrscheinlichkeit

Es wird ein Maß benötigt, um die Konnektivität der Anforderungen wiederspiegeln, sodass der Einfluss durch Propagationseffekte deutlich wird.

Hierfür wird, analog zur Auswirkungsanalyse, der PageRank verwendet:

- $$PR(r_i) = \frac{1-d}{n} + d \sum_{j \in A(r_i)} \frac{PR(r_j)}{n}$$
- Beispiel aus dem ARCA Workshop vom 23.03.

„Matching“ von Anforderungen und Initiatoren

Es muss ermittelt werden, welche Initiatoren für welche Anforderung relevant sind

Hierzu wird eine Expertenbewertung durchgeführt:

- Es werden die Kapitel-Überschriften in der Anforderungsliste betrachtet und pro Kapitel die relevanten Initiatoren manuell festgelegt
- Als Hilfestellung kann die Tabelle mit Änderungsimpulsen und Initiatoren herangezogen werden: „Welche Änderungsimpulse werden für Anforderungen in diesem Kapitel erwartet?“
- Den untergeordneten Anforderungen werden automatisiert die Initiatoren, die für die Kapitel-Überschrift ausgewählt wurden, zugeordnet

Berechnung: Maß für die exogene Änderungswahrscheinlichkeit

Aus den Referenzprojekten wird eine **projektspezifische Datengrundlage** gebildet, indem die Werte der Änderungsdaten der relevanten Referenzprojekte summiert werden

Pro Anforderung werden die Änderungen für die relevanten Initiatoren summiert

- Relevanz der Initiatoren wird aufgrund der Zuordnung der Anforderung zum Initiatoren festgelegt
- Diese Summe wird durch die Anzahl der gesamten Änderungen dividiert
- Zusammenfassung in einer Formel:
- $$P_{exogen}(r_i) = \frac{\text{Anzahl der Änderungen die für Anforderung } r_i \text{ relevant sind}}{\text{Gesamt Anzahl der Änderungen}}$$
- *: dieser Wert ist ein Indikator für die exogene Änderungswahrscheinlichkeit und keine absolute exogene Änderungswahrscheinlichkeit
- $$0 \leq P_{exogen}(r_i) \leq \frac{\text{Gesamt Anzahl endogener Änderungen}}{\text{Gesamt Anzahl der Änderungen}} \leq 1$$

Berechnung: Maß für die endogene Änderungswahrscheinlichkeit

Gemäß dem „Zufallsurfermodell“ wird der PageRank normiert, um eine Wahrscheinlichkeitsverteilung für die einzelnen Anforderungen zu generieren

- Es wird der PageRank sämtlicher Anforderungen durch den höchsten PageRank dividiert. Zusammenfassung als Formel:
- $$PR'(r_i) = \frac{PR(r_i)}{\max(PR)}$$
- Um den Indikator für die endogene Änderungswahrscheinlichkeit zu berechnen und in ein geeignetes Verhältnis zu $P_{exogen}(r_i)$ zu setzen, wird der normierte PageRank mit dem Anteil der endogenen Änderungen im Verhältnis zu der Gesamtzahl der Änderungen multipliziert:
- $$P_{exogen}(r_i) = PR'(r_i) \cdot \frac{\text{Gesamt Anzahl endogener Änderungen}}{\text{Gesamt Anzahl der Änderungen}}$$
- *: dieser Wert ist ein Indikator für die endogene Änderungswahrscheinlichkeit und keine absolute endogene Änderungswahrscheinlichkeit
- $$0 \leq P_{exogen}(r_i) \leq \frac{\text{Gesamt Anzahl endogener Änderungen}}{\text{Gesamt Anzahl der Änderungen}} \leq 1$$

Bild A-28: Auszug aus Workshopunterlagen zur Bewertung der Änderungswahrscheinlichkeit von Anforderungen

A-W10

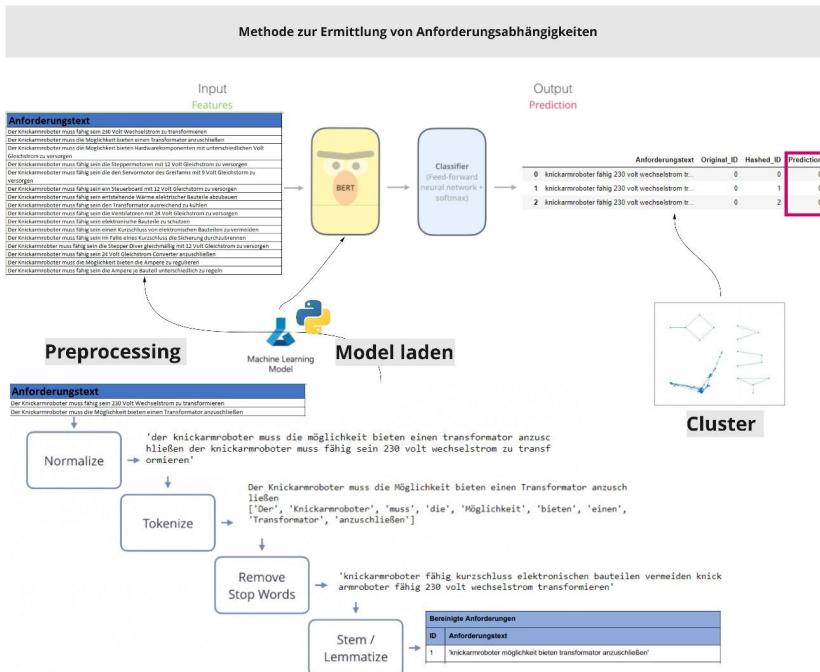


Bild A-29: Auszug aus den Workshopunterlagen zur Abhängigkeitsanalyse von Anforderungen

A-W11

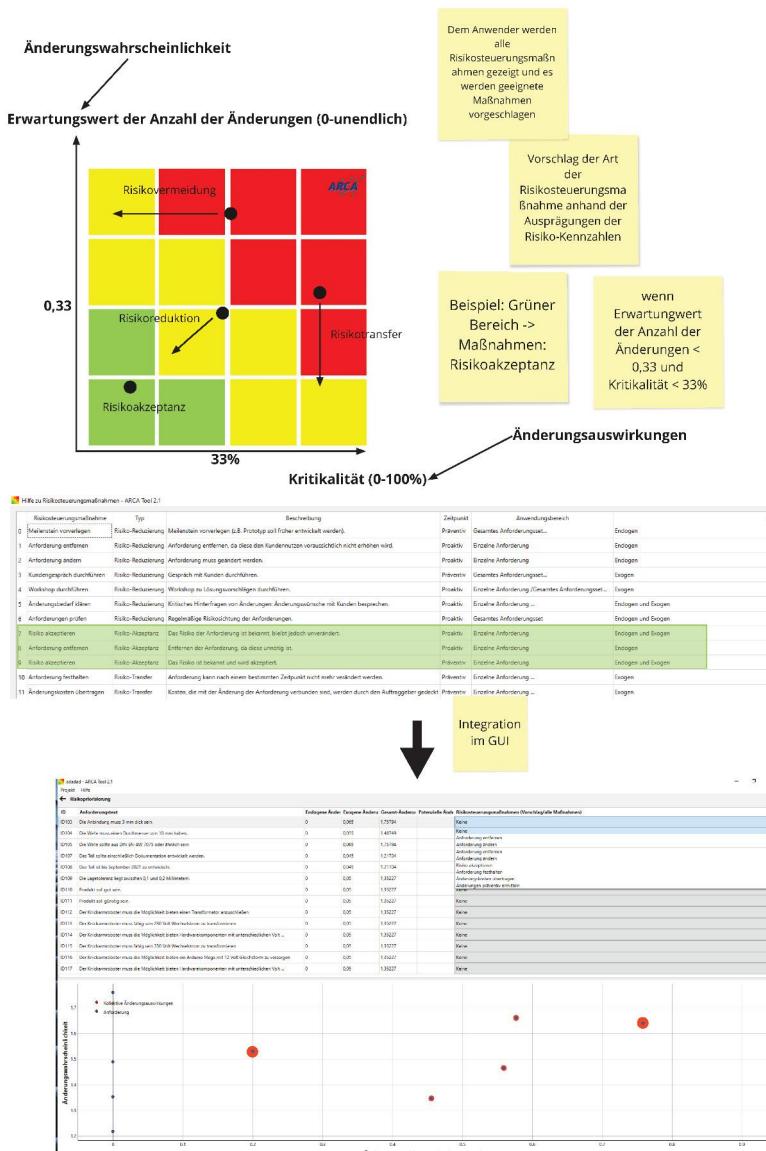


Bild A-30: Auszug aus den Workshopunterlagen zum Risikoportfolio von Anforderungsänderungen

A3.2 BIKINI: Rahmendaten von Aktivitäten und Unterlagen

Tabelle A-5: Aktivitäten zur Einbindung von Industrievertretern in BIKINI

ID	Rahmendaten
B-W1	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Vorstellung des Methodik Konzepts; Vorstellung exemplarischer Lastenhefte und Nutzungsszenarien der Anwendungspartner -Teilnehmer: Projektleiter EDAG, Projektleiter ATOS, Projektleiter Krause DiMaTec GmbH, Projektleiter Additive Marking GmbH Software-Entwickler, Projektleiter RHaug GmbH, UX-Designer, Wissenschaftliche Mitarbeiter (Summe: 21 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 08.2021
B-W2	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: BIKINI Kick Off; Workshop zu Use-Cases der Extrahierung und Formalisierung von Anforderungen -Teilnehmer: Projektleiter EDAG, Projektleiter ATOS, Projektleiter Krause DiMaTec GmbH, Projektleiter Additive Marking GmbH Software-Entwickler, Projektleiter RHaug GmbH, UX-Designer, Wissenschaftliche Mitarbeiter (Summe: 31 Teilnehmer) -Dauer: zwei Tage -Zeitpunkt: 09.2021
B-W3	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Anforderungserhebung an die Methodik mit Projektpartnern -Teilnehmer: Projektleiter EDAG, Projektleiter ATOS, Software-Entwickler, UX-Designer, Wissenschaftliche Mitarbeiter (Summe: 7 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 10.2021
B-W4	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Überarbeitung initialer Anforderungen -Teilnehmer: Projektleiter EDAG, Projektleiter ATOS, Software-Entwickler, Wissenschaftliche Mitarbeiter (Summe: 6 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 11.2021
B-R1	<ul style="list-style-type: none"> -Art: Regeltermin -Beschreibung: Regeltermin mit ATOS zur Besprechung der Implementierung der Methodik zur Extrahierung und Formalisierung -Teilnehmer: Projektleiter ATOS, Software-Entwickler (Summe: üblicherweise 3-6 Teilnehmer) -Dauer: 60 Minuten -Zeitpunkt: Ab 11.2021 bis Projektende

ID	Rahmendaten
B-W5	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: User-Storys der Methodik bei EDAG -Teilnehmer: Projektleiter EDAG, Projektleiter ATOS, Projektleiter Krause DiMaTec GmbH, Projektleiter Additive Marking GmbH Software-Entwickler, Projektleiter RHaug GmbH, UX-Designer, Wissenschaftliche Mitarbeiter (Summe: 15 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 11.2021
B-W6	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Vorstellung des Konzepts der Abhängigkeitsanalyse von Anforderungen -Teilnehmer: Projektleiter EDAG, Projektleiter ATOS, Projektleiter Krause DiMaTec GmbH, Software-Entwickler, UX-Designer, Wissenschaftliche Mitarbeiter (Summe: 8 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 01.2022
B-W7	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Vorstellung des initialen Mockups des Software-Werkzeugs der Methodik -Teilnehmer: Projektleiter EDAG, Projektleiter ATOS, Software-Entwickler, UX-Designer (Summe: 7 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 05.2022
B-W8	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Überarbeitung des initialen Mockups des Software-Werkzeugs der Methodik mit Perspektive Implementierung -Teilnehmer: Projektleiter ATOS, Software-Entwickler, UX-Designer (Summe: 6 Teilnehmer) -Dauer: 120 Minuten -Zeitpunkt: 10.2022
B-W9	<ul style="list-style-type: none"> -Art: Workshop -Beschreibung: Validierung der modellierten Software-Architektur -Teilnehmer: Software-Entwickler (Summe: 3 Teilnehmer) -Dauer: 60 Minuten -Zeitpunkt: 11.2022
B-F1	<ul style="list-style-type: none"> -Art: Bewertungsbogen -Beschreibung: Validierung der prototypischen Lösung -Teilnehmer: Projektleiter EDAG, ATOS, Krause DiMaTec GmbH, Software-Entwickler, UX-Designer (Summe: 10 Teilnehmer) -Dauer: ca. 30 Minuten -Zeitpunkt: 01.2023

B-W1

AP3 Advanced Requirements Engineering

Übersicht Methodik

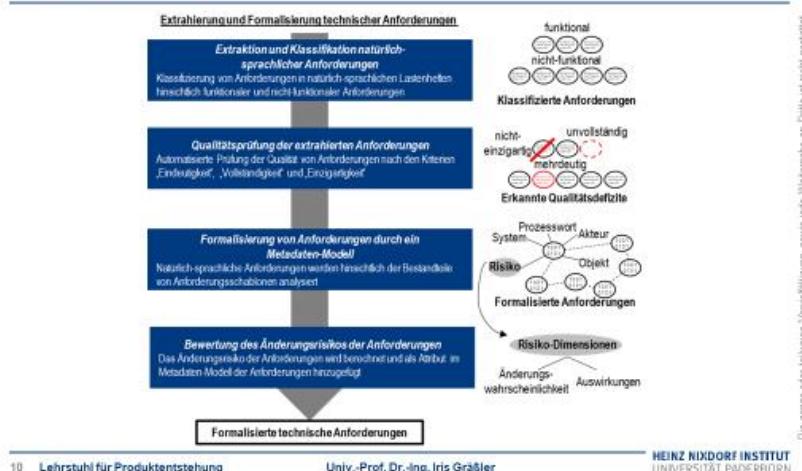


Bild A-31: Initiales Methodikkonzept

AP3 Advanced Requirements Engineering

Input: Extraktion und Klassifikation natürlichsprachlicher Anforderungen

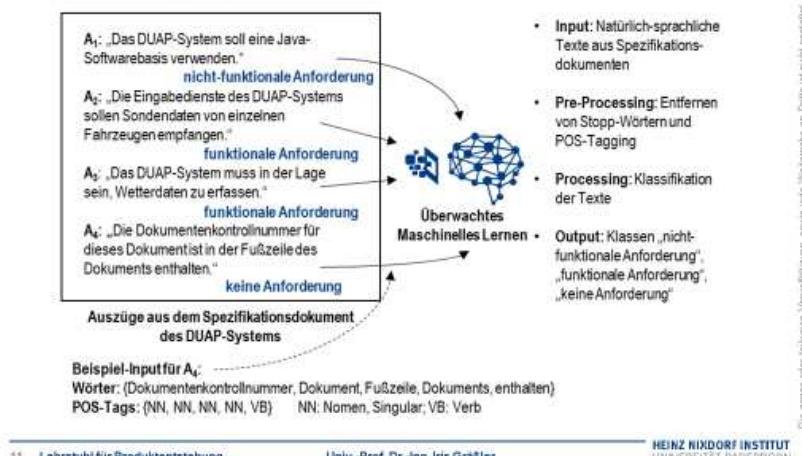


Bild A-32: Extrahierung von Anforderungen Konzept

AP3 Advanced Requirements Engineering

Output: Formalisierung von Anforderungen durch ein Metadaten-Modell

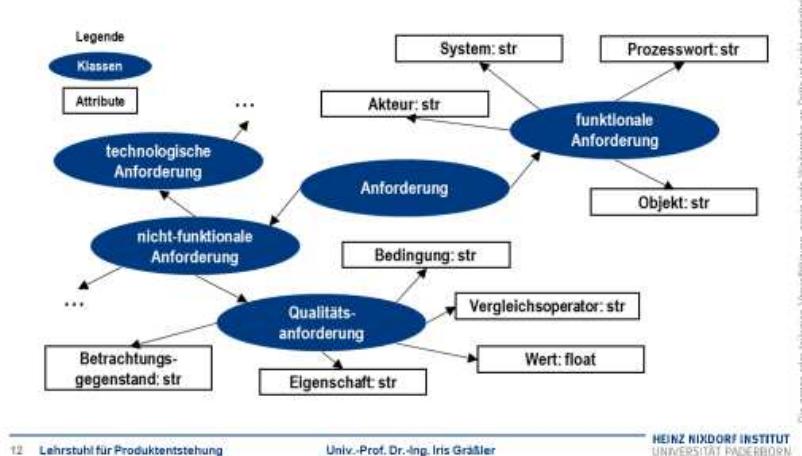


Bild A-33: Initiales Metadatenmodell formalisierter Anforderungen

B-W2

AP3 Advanced Requirements Engineering

Use-Case Partialmodell für die Rolle des/der Anforderungsmanager:in

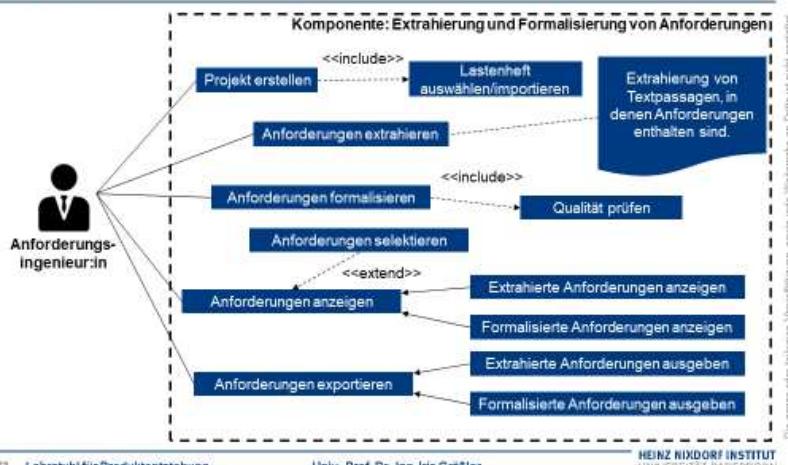


Bild A-34: Use Case Anforderungsmanager

AP3 Advanced Requirements Engineering

Use-Case Partialmodell für die Rolle des/der Risikomanager:in

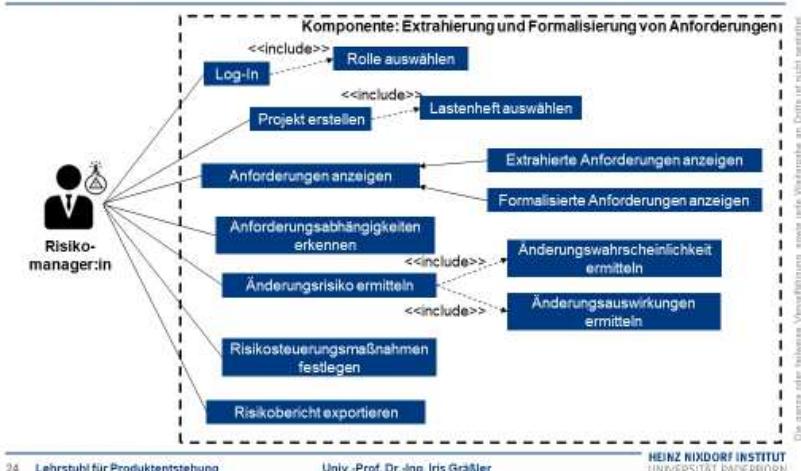


Bild A-35: Use-Case Risikomanager

AP3 Advanced Requirements Engineering

Use-Case Partialmodell für die Rolle des/der Entwicklungsingenieur:in

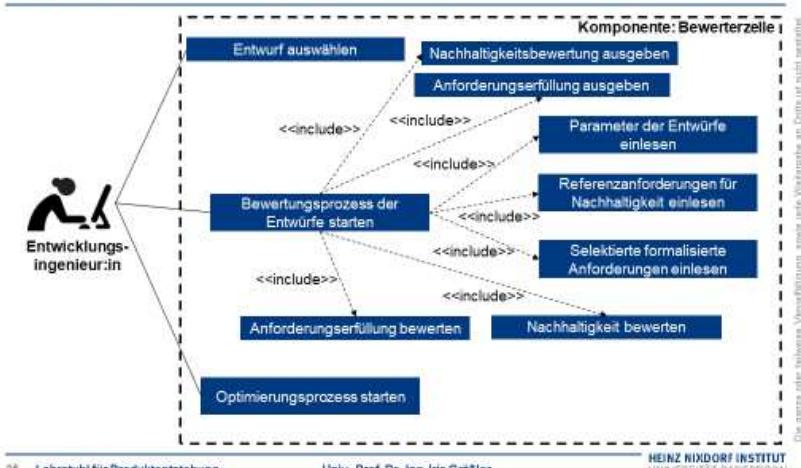


Bild A-36: Use-Case Entwicklungsingenieur

B-W3

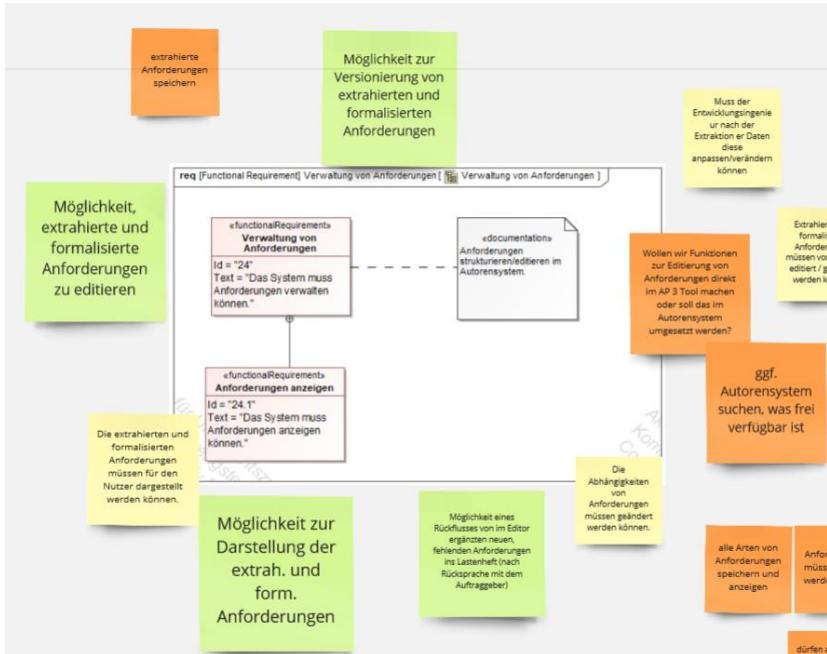


Bild A-37: Auszug aus digitalen Workshop-Unterlagen (Miro-Board)

#	Name	Text
1	1.1.1.1 Eingangsdaten	Die Methodik muss Anforderungsdokumente im Kontext der Entwicklung komplexer interdisziplinärer Systeme (z.B. Mechanik, Elektrik/Elektronik und Softwaretechnik) verarbeiten können.
2	30.1.1 Interdisziplinarität der Anforderungen	Die verarbeiteten Anforderungsdokumente müssen nach einer branchenüblichen Dokumentationsform gestaltet sein.
3	30.1.2 Branchenübliche Dokumentationsform	
4	31. Methoden	
5	31.1. Extrahierung von Anforderungen	Die Extrahierung von Anforderungen muss in natürlichsprachlichen Texten in Anforderungsdokumenten Anforderungen erkennen können.
6	31.2. Formalisierung von Anforderungen	Die Funktion zur Formalisierung von Anforderungen muss Informationen in Anforderungen strukturieren können.
7	31.3. Abhängigkeitsanalyse von Anforderungen	Die Abhängigkeitsanalyse von Anforderungen muss die Abhängigkeitsarten erkennen können, die notwendig sind, um eine Risikoberechnung des Änderungsrisikos von Anforderungen zu ermöglichen.
8	31.4. Änderungsrisiko von Anforderungen	Das Risiko einer Anforderungsänderung muss berechnet werden können.
9	31.5. Selektion von Anforderungen für gene...	Die Anforderungen müssen selektiert werden können hinsichtlich benötigter Anforderungen für die generative Entwicklung (hier Bauteile).
10	31.6. Bewertung von Entwürfen	Die in der generativen Entwicklung generierten Entwürfe müssen hinsichtlich Zielerreichung (hohe Nachhaltigkeit, Nachhaltigkeit und niedrige Kosten) bewertet werden können.
11	32. Anwendung	
12	32.1. Verfügbarkeit benötigter Informationen	Die zur Anwendung der Methodik benötigten Informationen müssen branchenüblich verfügbar sein.
13	32.2. Generische Anwendbarkeit	Die Methodik muss so gestaltet sein, dass eine generische Anwendung in der Entwicklung komplexer technischer Systeme möglich ist.
14	32.3. Spezifische Anwendbarkeit	Die Methoden der Komponente zur Extrahierung und Formalisierung müssen so gestaltet sein, dass die Anwendung in der Projektrealisierung möglich ist.
15	32.4. Akzeptabler Anwendungsaufwand	Die Anwendung der Methodik muss so aufwendig sein, dass Anwender:innen den Anwendungsaufwand in der industriellen Anwendung als „akzeptabel“ beurteilen.
16	32.5. Intergrierbarkeit in den Entwicklungsp...	Die Methodik muss so gestaltet sein, dass sie in den Entwicklungsprozess eines Unternehmens integriert werden kann.
17	33. Ergebnis	
18	33.1. Hervorhebende Genauigkeit	Das Ergebnis der Methodik muss in primär unterstützten Abstufungen hervorhebend genau sein.
19	33.2. Schnittstellen zu Werkzeugen	Die extrahierten Anforderungen müssen in einem gängigen Austauschformat abgebildet werden.

Bild A-38: Initiale Stakeholder-Anforderungen

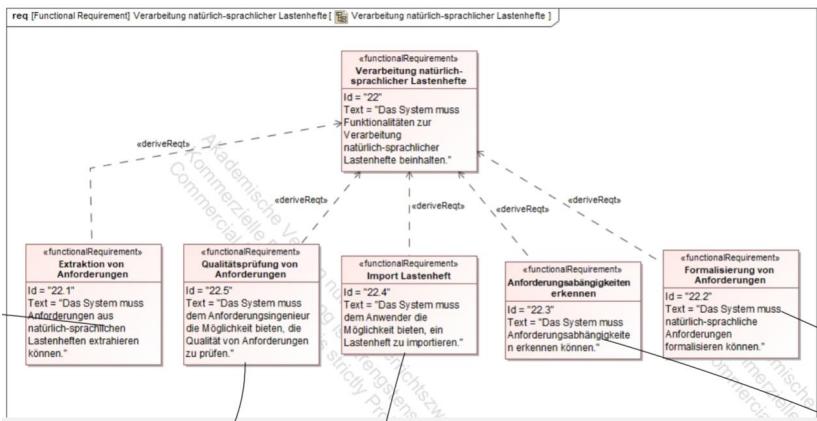


Bild A-39: Initiale funktionale Anforderungen an die Methoden

B-W4

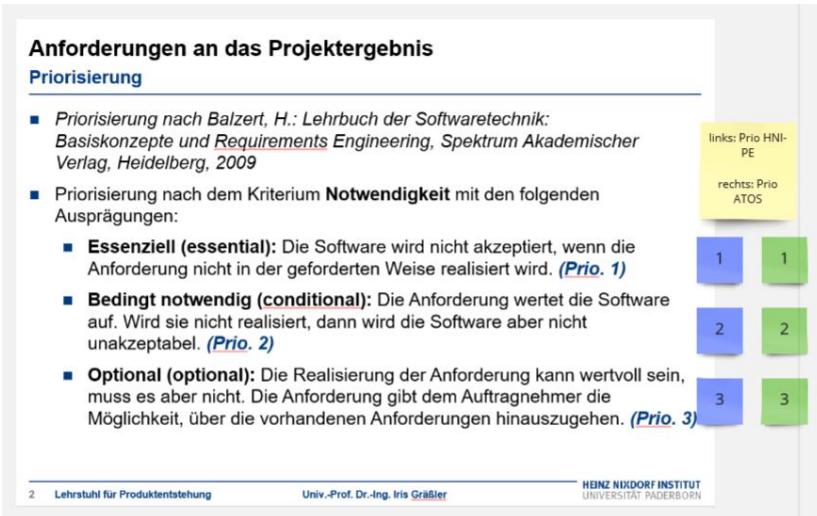


Bild A-40: Priorisierungsschema von Anforderungen

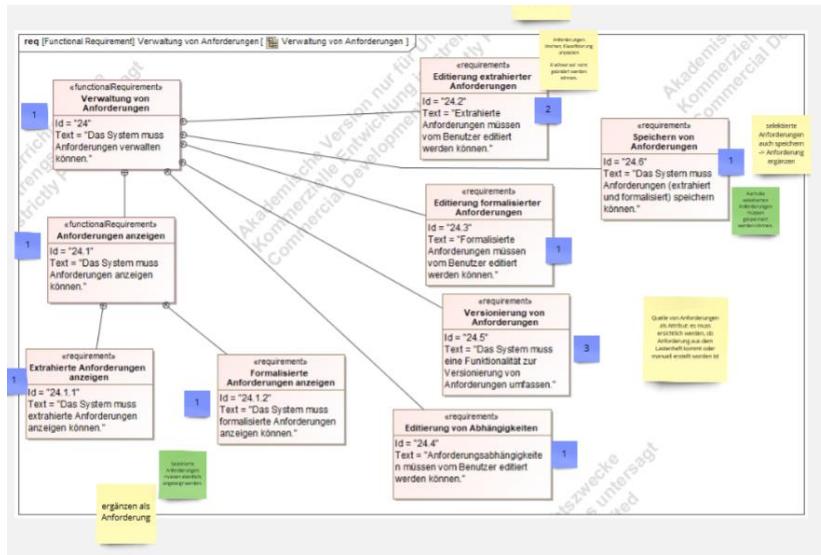


Bild A-41: Exemplarische Priorisierung von Anforderungen (Miro-Board)

B-W5

Aufgrund von Geheimhaltungsvereinbarungen sind diese Unterlagen nicht öffentlich.

B-W6

Abhängigkeitsanalyse

Motivation und Zielsetzung

- Heutige Entwicklungsprojekte sind durch eine hohe Komplexität, Dynamik und Unsicherheit geprägt
 - Kontinuierliche Anpassung der Entwicklungsziele ist unvermeidbar
 - Anforderungsänderungen sind **zentrale Ursache** für Projektfehlschläge
 - Hohe Kosten durch Propagationseffekte

Zielsetzung

- Risikomanagement von Anforderungsänderungen ermöglichen und umsetzen durch die Anwendung einer softwaregestützten Methode
 - Hoher Automatisierungsgrad der Methode zur Anwendbarkeit für umfangreiche Anforderungssets
 - Notwendig hierfür ist eine automatisierte Abhängigkeitsanalyse

Abhängigkeitsanalyse

Vorarbeiten HNI-PE

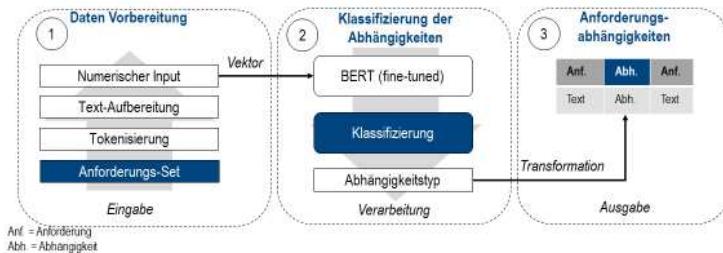


Bild A-43: Vorarbeiten in ARCA

Abhängigkeitsanalyse

Handlungsbedarf

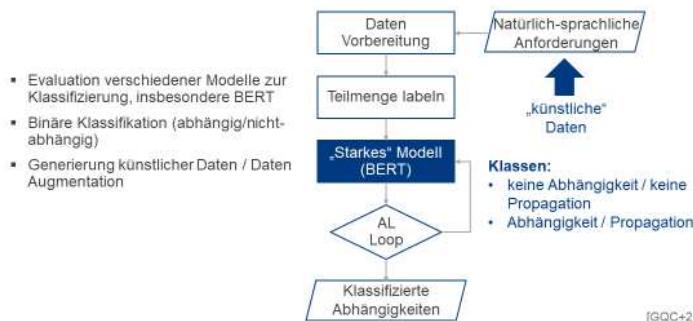


Bild A-44: Initiales Konzept Abhängigkeitsanalyse

B-W7

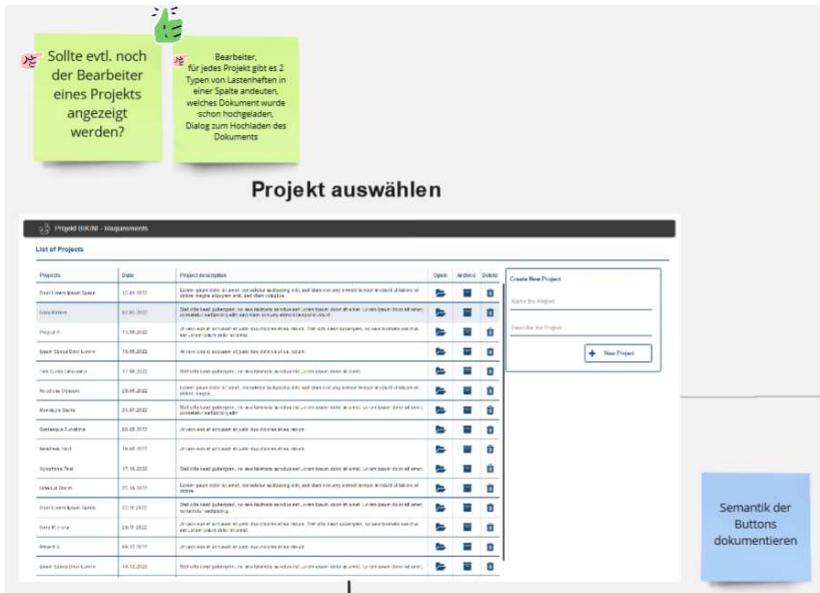


Bild A-45: Initiales Mockup (digitale Workshopunterlagen)



Bild A-46: Digitale Workshopunterlagen zum Einlesen einer Spezifikation (Beispiel)

B-W8

Fehler: jetzt nur nach Sort of Requirements filtern (falls Tags verwendet werden müssen)

ID patername entfernen

Prio 2: KVP hinzufügen

Bild A-47: Auszug aus digitalen Workshopunterlagen (I/III)

ID: ID des Keys und nicht die ID der Anforderung

ID: ID hier nicht anzeigen

Number of Keys -> Number of Values

Für Keys in 2 Anforderungen ist dann müssen beide Werte angezeigt werden

Falls Key in 2 Anforderungen ist, dann müssen beide Werte angezeigt werden

Falls Key in 2 Anforderungen ist, dann müssen beide Werte angezeigt werden

Für Keys ist es nur möglich weitere Values hinzuzufügen

Verified ID und Other ID zu

Falls Spalte 'Verified' nicht angezeigt wird, dann kann es sich um eine Fehler handeln

Bild A-48: Auszug aus digitalen Workshopunterlagen (II/III)

gekennzeichnet

Verified ID und Other ID zu Requirement ID 1 und Requirement ID 2

Dependency-Analyse

Are the following requirements interdependent?

Requirement ID 1: The door handle must remain functional for at least 5 years without any signs of wear!

Requirement ID 2: The instead of the component must be plastic.

Not Interdependent Separate this and Dependency Analysis

Dependent

Finish

Sollte in Abhängigkeiten nur der Key rechts angezeigt werden, dann kann es sich um eine Fehler handeln

Hintergrund z.B. ausgrauen oder Detaillierung nicht anzeigen...

Bild A-49: Auszug aus digitalen Workshopunterlagen (III/III)

B-W9

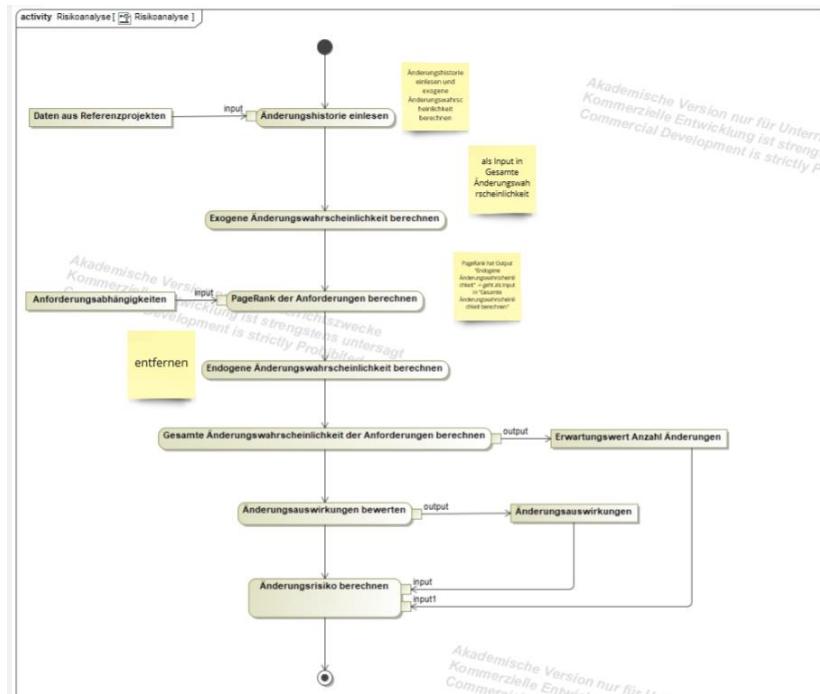


Bild A-50: Exemplarisches Aktivitätsdiagramm aus den digitalen Workshopunterlagen zur Validierung der Software-Architektur

B-F1

Bewertungsbogen zur Validierung

Liebe(r) Teilnehmer(innen),

zur Validierung der Methodik zur Extrahierung und Formalisierung von Anforderungen möchte ich Sie bitten, diesen Bewertungsbogen auszufüllen. Die Validierung ist in die folgenden vier Bereichen unterteilt:

- 1) Thesen zur Ist-Situation
- 2) Thesen zur Anwendbarkeit des Software-Werkzeugs
- 3) Thesen zum Anwendungsaufwand
- 4) Thesen zum Mehrwert der Methodik-Anwendung

Bitte bewerten Sie die Thesen in den darauffolgenden Blöcken nach der folgenden Skala:

- 2: stimme zu
- 1: stimme eher zu
- 0: teils / teils
- -1: stimme eher nicht zu
- -2: stimme nicht zu

Ich möchte mich bereits im Vorfeld herzlich für die Mitarbeit bedanken!

Daniel Preuß

Thesen zur Ist-Situation

- 1) Zur Extrahierung von Anforderungen aus Lastenheften ist ein hoher manueller Aufwand notwendig.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 2) Natürlich-sprachliche Anforderungen sind durch Software-Werkzeuge in der Produktentwicklung üblicherweise nicht direkt interpretierbar. Es besteht ein Bedarf nach formalisierten Anforderungen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 3) Anforderungsänderungen werden in Unternehmen noch nicht systematisch gehandhabt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

Thesen zur Anwendbarkeit des Software-Werkzeugs

- 4) Ich kann mir sehr gut vorstellen, das Software-Werkzeug regelmäßig zu nutzen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 5) Ich empfinde das Software-Werkzeug als unnötig komplex.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 6) Ich denke, dass ich technischen Support brauchen würde, um das Software-Werkzeug zu nutzen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 7) Ich finde, dass es im Software-Werkzeug zu viele Inkonsistenzen gibt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 8) Ich kann mir vorstellen, dass die meisten Leute das Software-Werkzeug schnell zu beherrschen lernen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 9) Ich muss voraussichtlich eine Menge Dinge lernen, bevor ich mit dem Software-Werkzeug arbeiten kann.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 10) Das Werkzeug ist für branchenübliche Lastenhefte anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 11) Das Werkzeug ist generisch für die Entwicklung komplexer technischer Systeme anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 12) Ich kann mir vorstellen, dass die Anwendung des Werkzeugs in unternehmensinterne Entwicklungsprozesse integriert werden kann.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

13) Die Informationen zur Anwendung des Werkzeugs sind in der Praxis frühzeitig (in der Regel ab der Projektinitierungsphase) in Form von Daten oder Expertenwissen verfügbar:

		ja	nein
a)	Natürlich-sprachliche Anforderungsspezifikation, bzw. Lastenheft (Datengrundlage zur Anwendung der Extrahierung und Klassifizierung von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
b)	Wissen zur Definition von Strukturregeln von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
c)	Wissen zur Definition von Textindikatoren von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
d)	Gekennzeichnete Daten zum Training von Named-Entity-Recognition Modellen für Schlüssel-Wert-Paare (KI-Modelle zur Formalisierung von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
e)	Anforderungsabhängigkeiten (natürlich-sprachliche Anforderungspaare sowie die Abhängigkeit zwischen den Anforderungen) für das initiale Training eines KI-Modells (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
f)	Wissen zur Bewertung von Anforderungsabhängigkeiten (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
g)	Anzahl der aufgetretenen Anforderungsänderungen in vergangenen Projekten (Änderungshistorie)	<input type="checkbox"/>	<input type="checkbox"/>
h)	Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen (Risikomanagement von Anforderungsänderungen)	<input type="checkbox"/>	<input type="checkbox"/>

Thesen zum Anwendungsaufwand

14) Den Vorbereitungsaufwand, bevor die Anwendung möglich ist, empfinde ich als angemessen: Erstellen eines Domänenlexikons; Auswahl und Vorbereitung von Anforderungsabhängigkeitsdaten; Zusammenstellung einer Änderungshistorie vergangener Projekte;

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

15) Den Anwendungsaufwand schätze ich als angemessen ein: Import von Lastenheften, Prüfen und Auswahl von Schlüssel-Wert-Paaren, Abhängigkeitsanalyse, Bewertung von Änderungsauswirkungen; Ausgabe der Ergebnisse im CSV-Format

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

Thesen zum Mehrwert der Methodik-Anwendung

16) Durch die automatisierte Extrahierung von Anforderungen aus Lastenheften wird Aufwand reduziert und es wird erleichtert, eine vollständige Anforderungsliste zu generieren.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

17) Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

18) Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

19) Auf Grundlage der Anforderungsabhängigkeiten können Auswirkungen von Anforderungsänderungen besser bewertet werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

20) Auf Grundlage der Informationen zum Änderungsrisiko von Anforderungen können Risiken realistischer abgeschätzt werden als zuvor.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

21) Anhand der teilautomatisierten Risikoanalyse kann ein höherer Anteil an Anforderungen eines Anforderungssets auf Änderungsrisiken untersucht werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

22) Die Anwendung des Software-Werkzeugs trägt zur Effizienzsteigerung in der Entwicklung komplexer technischer Systeme bei.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

23) Durch den CSV-Export können die Anforderungsdaten aufwandsarm in branchenübliche Anforderungsmanagement-Software importiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

A4 Unterlagen zur Methodikentwicklung

A4.1 Anforderungsdiagramme

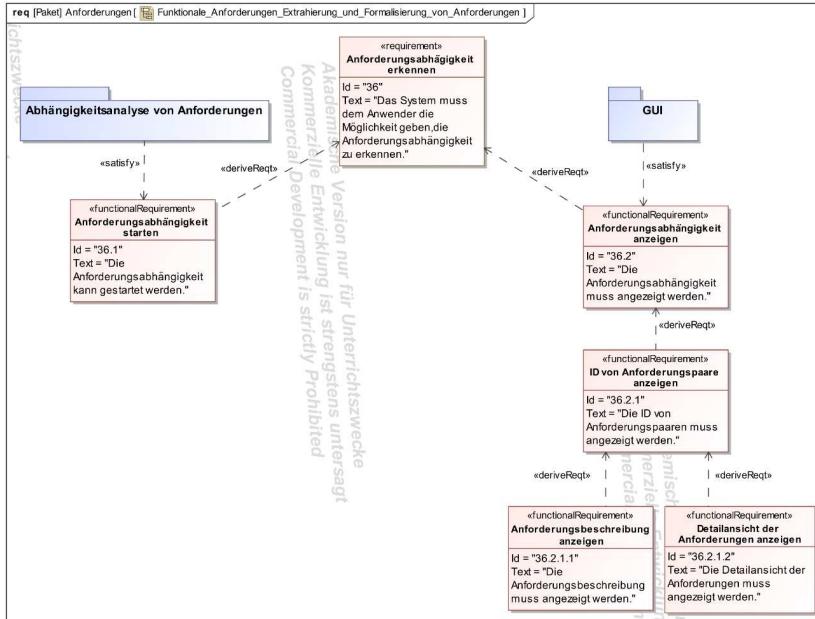


Bild A-51: Anforderungsdiagramm „Anforderungsabhängigkeiten erkennen“ [Zha22]

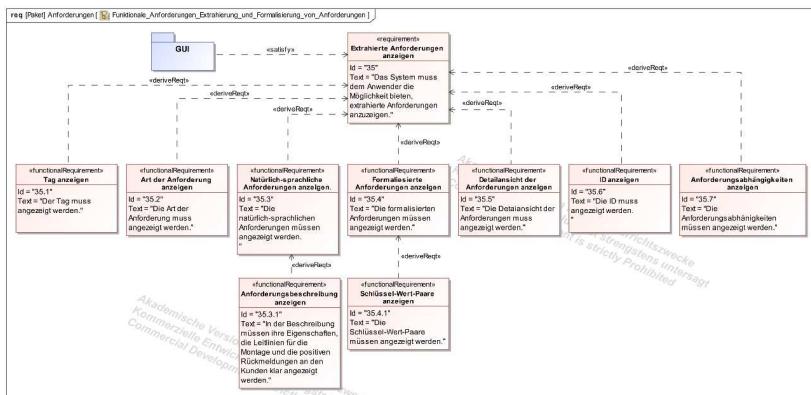


Bild A-52: Anforderungsdiagramm "Extrahierte Anforderungen anzeigen" [Zha22]

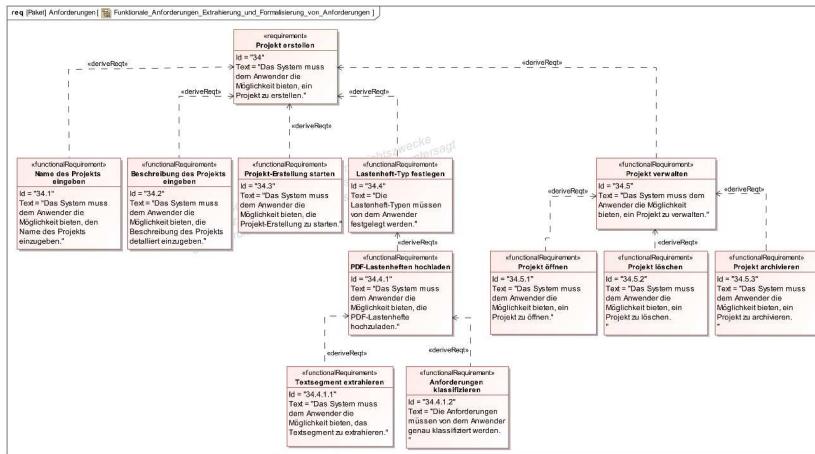


Bild A-53: Anforderungsdiagramm "Projekt erstellen" [Zha22]

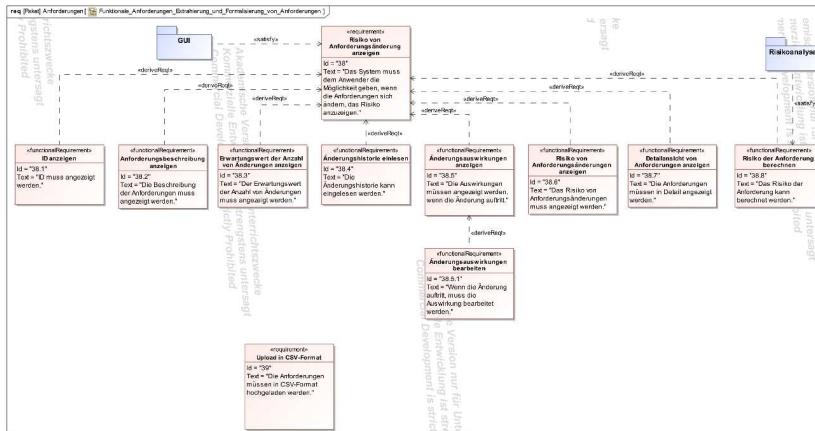


Bild A-54: Anforderungsdiagramm "Risiko von Anforderungsänderungen anzeigen" [Zha22]

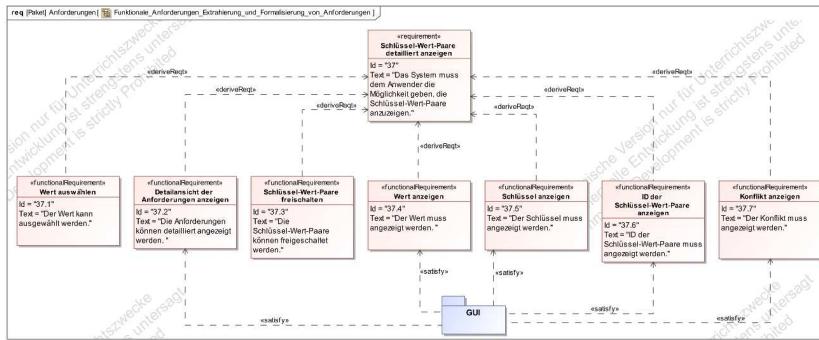


Bild A-55: Anforderungsdiagramm "Schlüssel-Wert-Paare detailliert anzeigen" [Zha22]

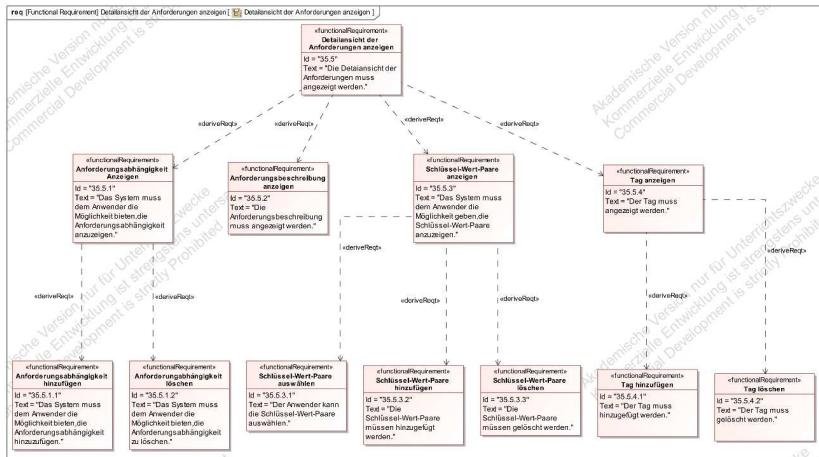


Bild A-56: Anforderungsdiagramm "Detailansicht der Anforderungen anzeigen" [Zha22]

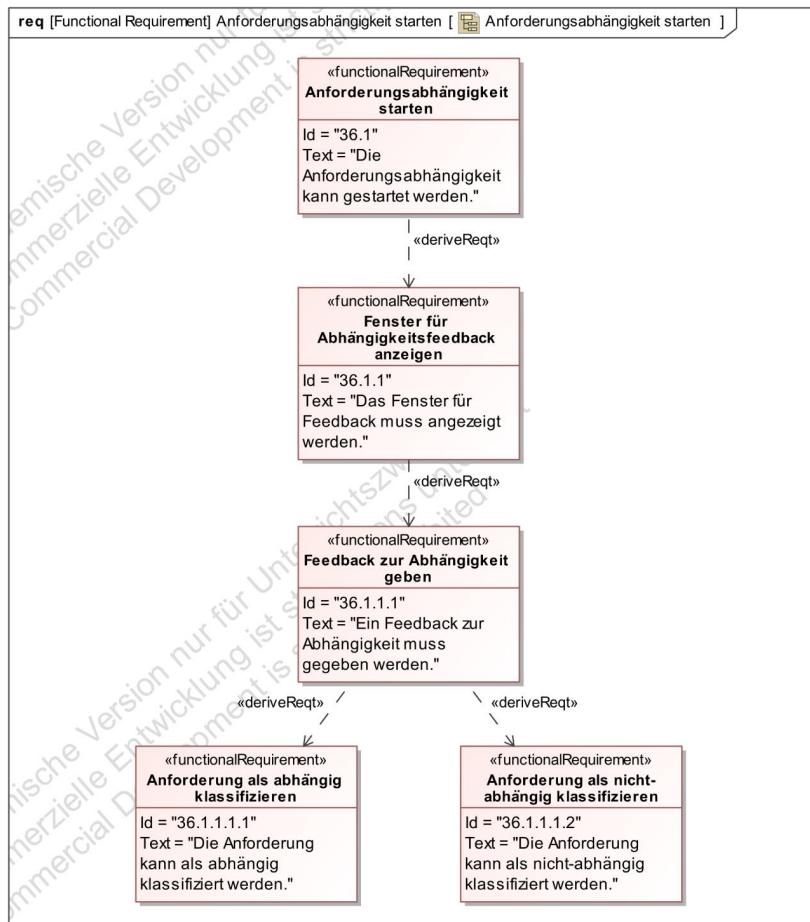


Bild A-57: Anforderungsdiagramm "Anforderungsabhängigkeitsanalyse starten" [Zha22]

A4.2 Use-Case Diagramme

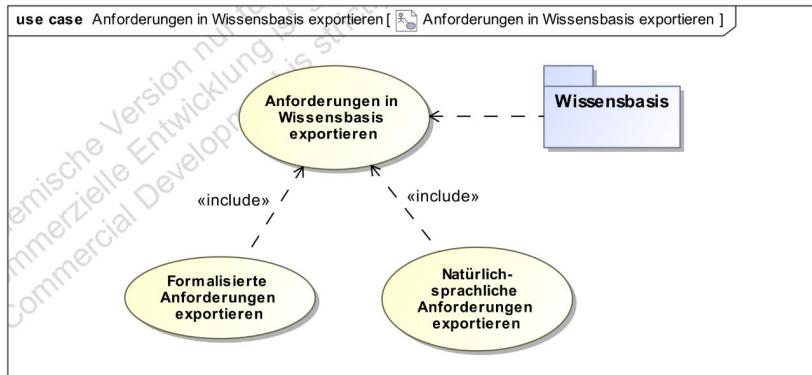


Bild A-58: Use-Case-Diagramm "Anforderungen in Wissensbasis exportieren" [Zha22]

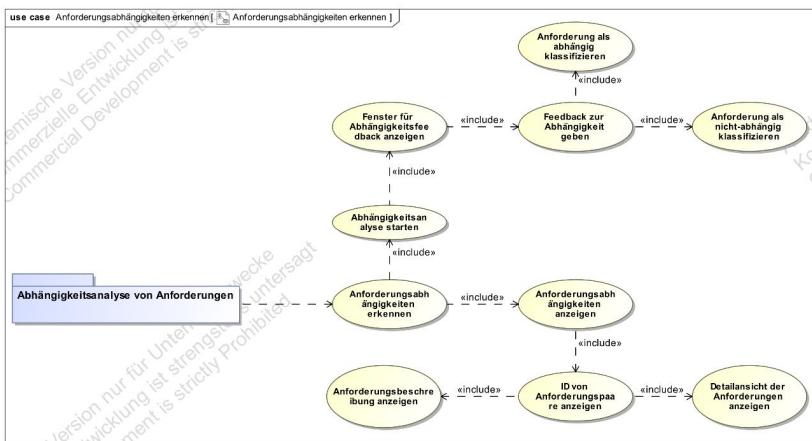


Bild A-59: Use-Case-Diagramm "Anforderungsabhängigkeiten erkennen" [Zha22]

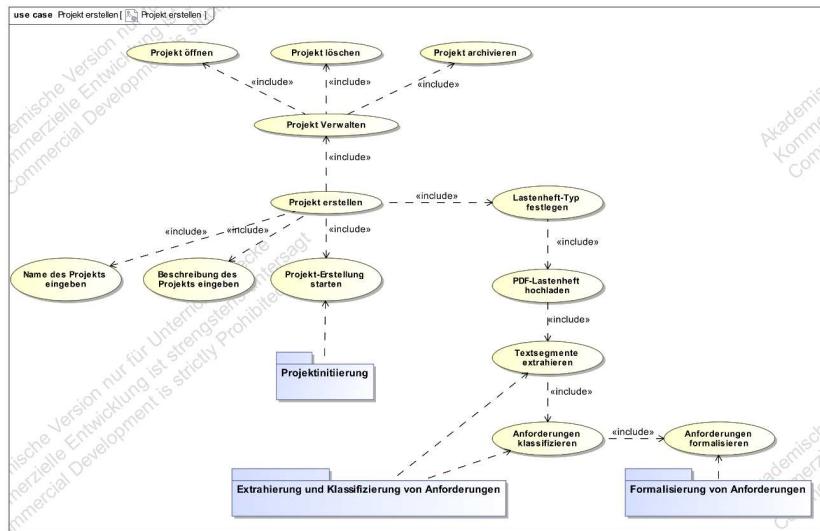


Bild A-60: Use-Case-Diagramm "Projekt erstellen" [Zha22]

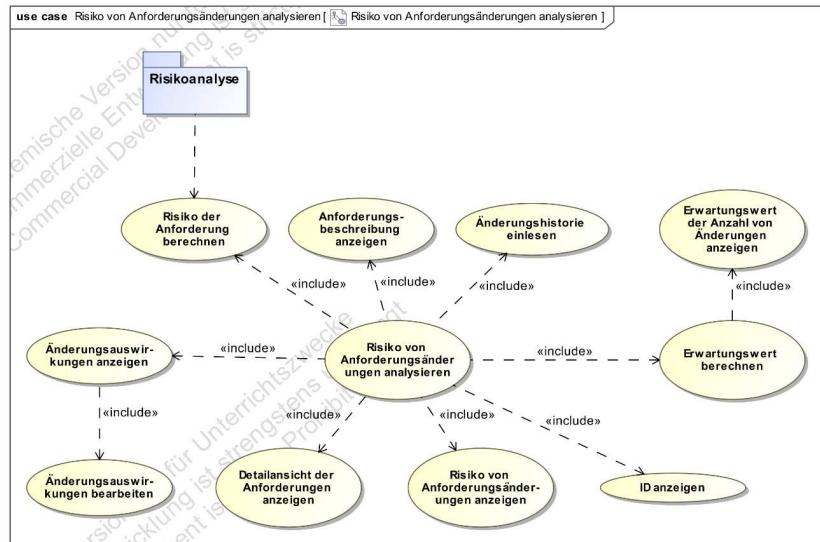


Bild A-61: Use-Case-Diagramm "Risiko von Anforderungsänderungen analysieren" [Zha22]

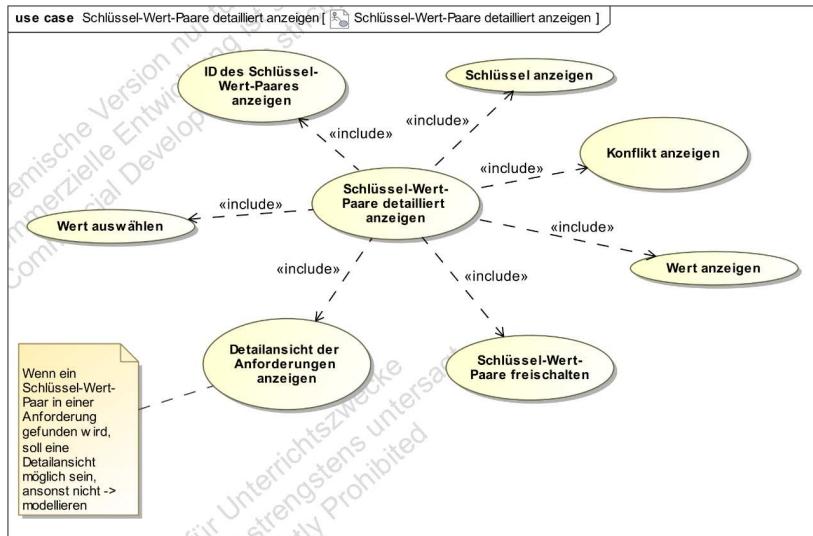


Bild A-62: Use-Case-Diagramm "Schlüssel-Wert-Paare detailliert anzeigen" [Zha22]

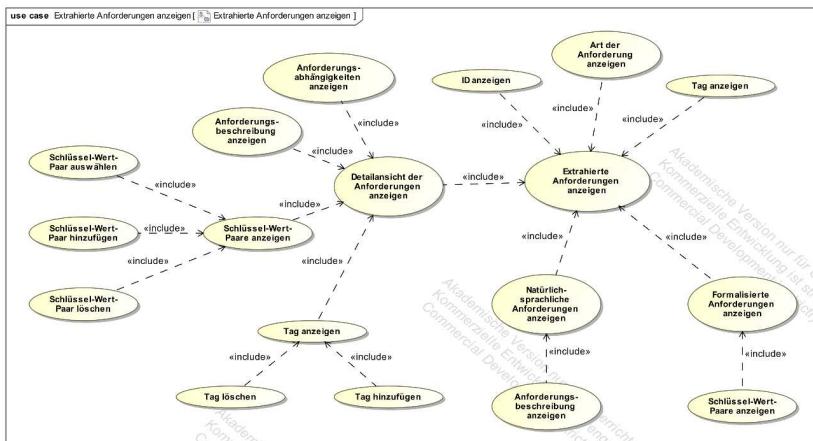


Bild A-63: Use-Case-Diagramm "Extrahierte Anforderungen anzeigen" [Zha22]

A4.3 Aktivitätsdiagramme

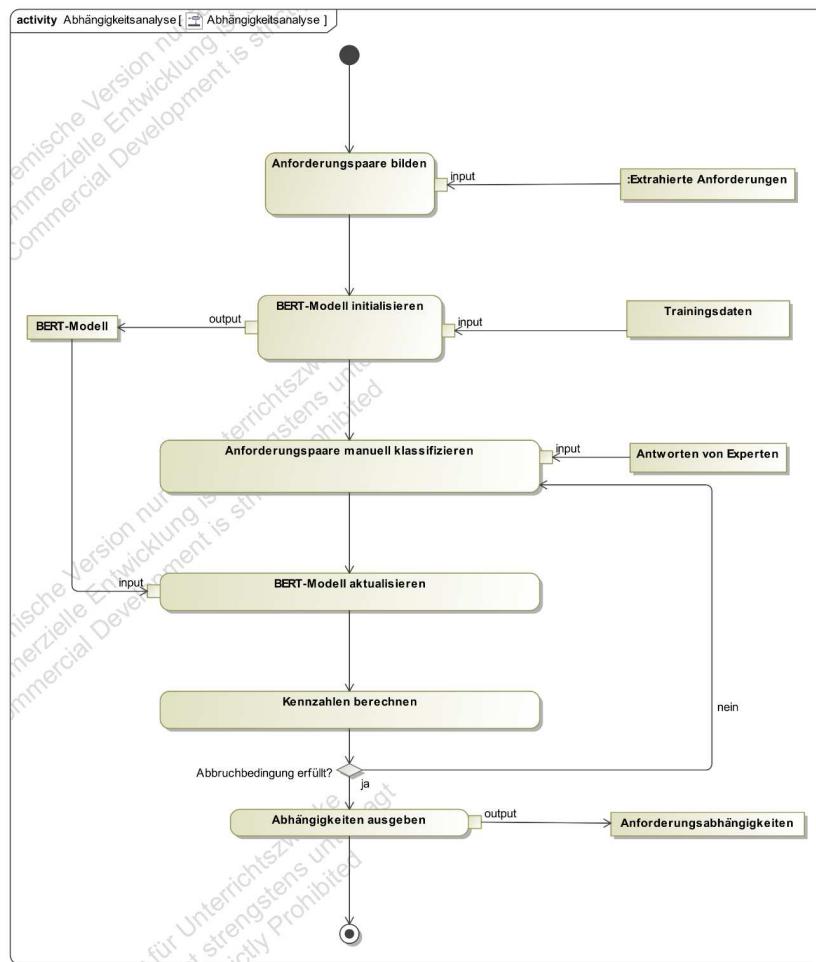


Bild A-64: Aktivitätsdiagramm "Abhängigkeitsanalyse" [Zha22]

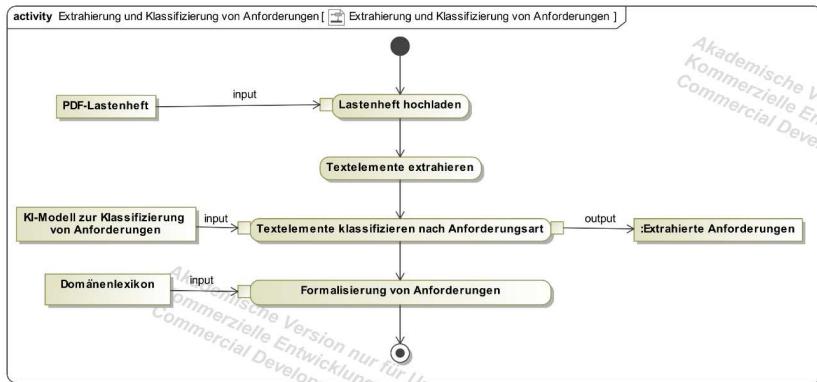


Bild A-65: Aktivitätsdiagramm "Extrahierung und Klassifizierung von Anforderungen" [Zha22]

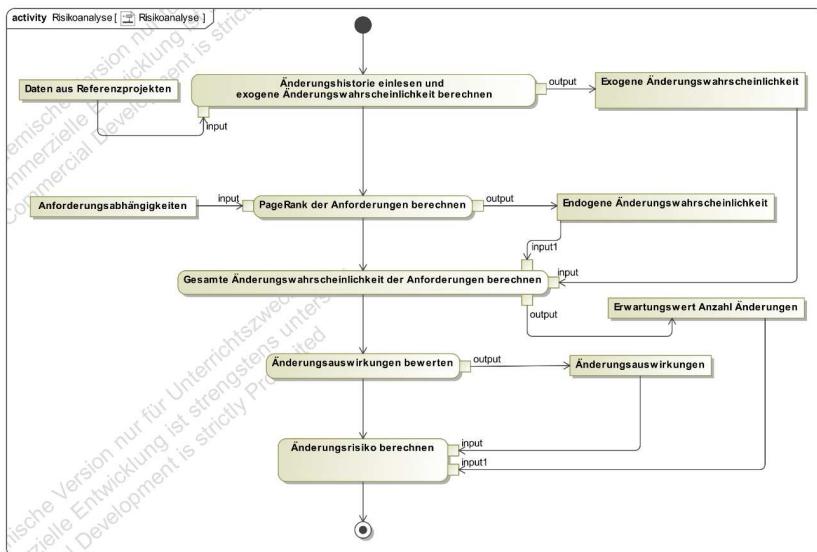


Bild A-66: Aktivitätsdiagramm "Risikoanalyse" [Zha22]

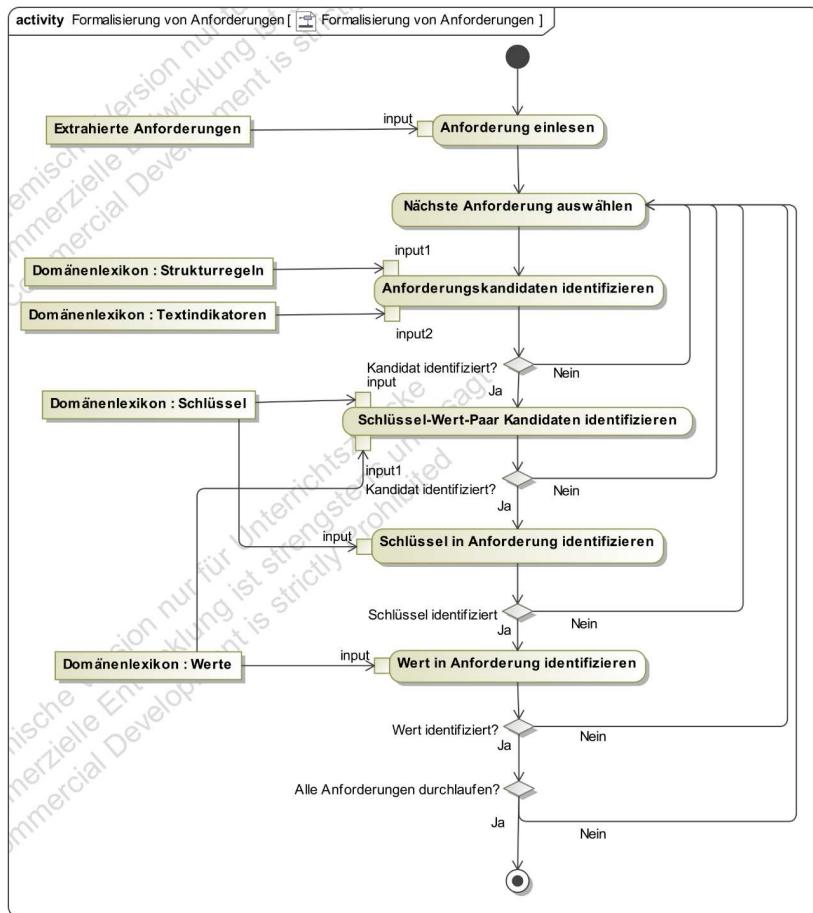


Bild A-67: Aktivitätsdiagramm "Formalisierung von Anforderungen" [Zha22]

A4.4 Paketdiagramme

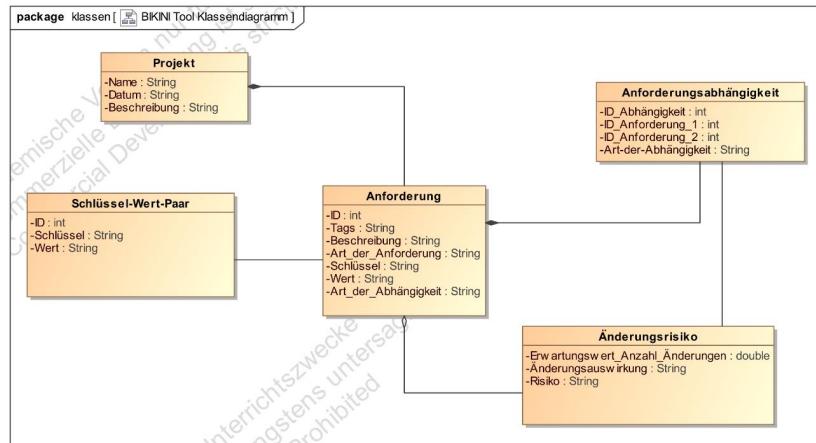


Bild A-68: Klassendiagramm "BIKINI Werkzeug" [Zha22]

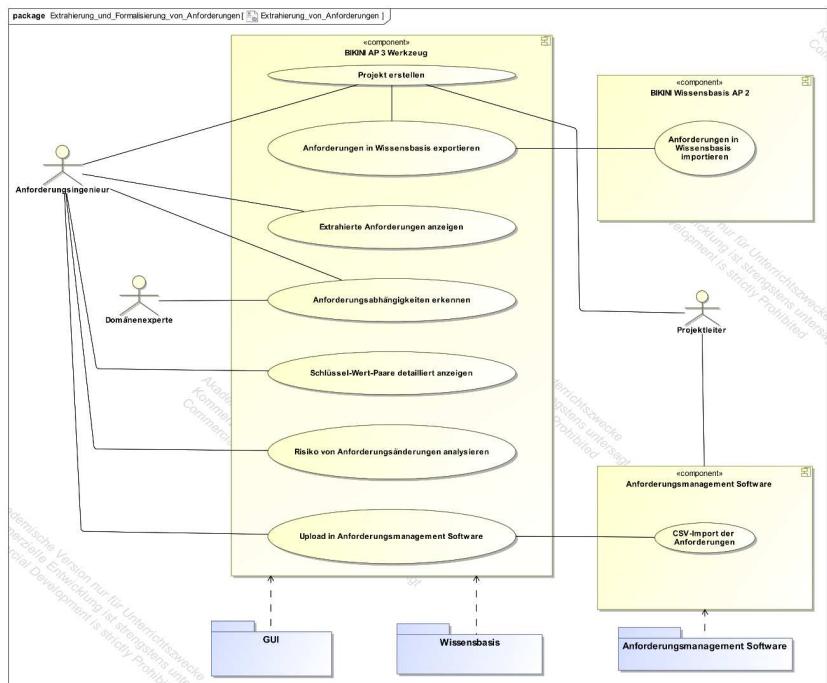


Bild A-69: Use-Cases, Akteure und Software-Komponenten [Zha22]

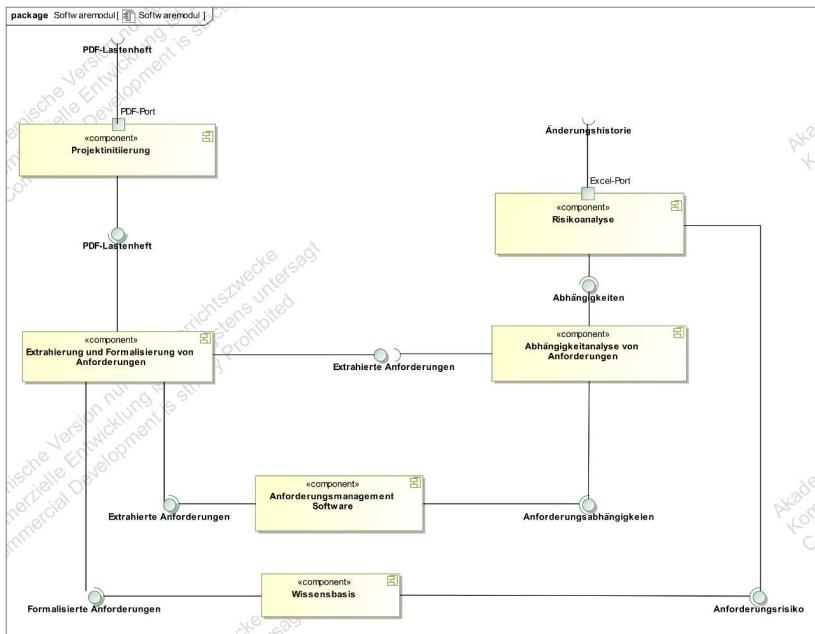


Bild A-70: Komponentendiagramm "Software-Modul" [Zha22]

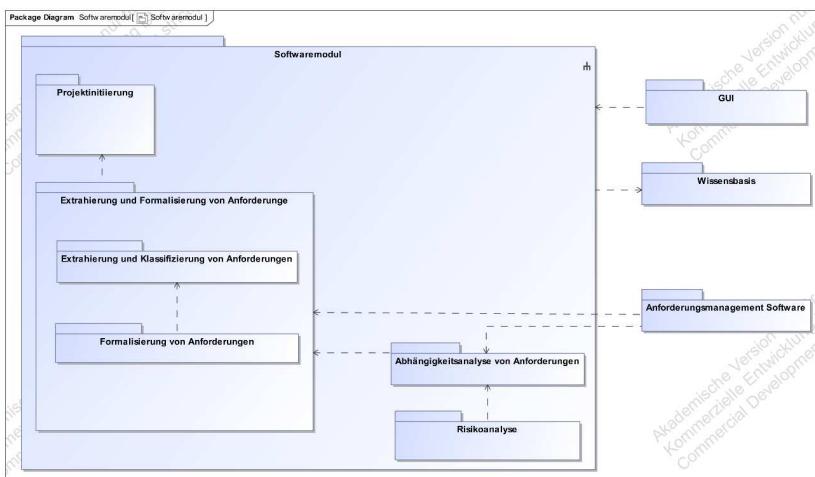


Bild A-71: Paket-Diagramm "Software-Modul" [Zha22]

A4.5 Algorithmen zum Extrahieren von Textsegmenten

```

9  def split_sentences(text):
10    sentences = []
11    if text:
12      for sentence in spacy(text).sents:
13        append = True
14
15        if len(sentences) > 0:
16          """
17          Do not split if:
18          - Sentence 1 ends with a character or symbol that indicates that the sentence continues
19          - Sentence 2 starts with a character or symbol that indicates that the sentence continues
20          - Sentence 2 starts with a lowercase letter
21          """
22
23        current_sentence = sentence.text.strip()
24        previous_sentence = sentences[-1]
25        if (
26          (previous_sentence and previous_sentence[-1] in mid_sentence_chars)
27          or (current_sentence and current_sentence[0] in mid_sentence_chars)
28          or (current_sentence and current_sentence[0].islower())
29        ):
30          append = False
31
32        if append:
33          sentences.append(sentence.text.strip())
34        else:
35          sentences[-1] = f"{sentences[-1]} {sentence.text.strip()}"
36
37    return sentences

```

Bild A-72: Algorithmus zum Trennen von Sätzen (Ausschnitt)

```

10  def check_coreferences(text_1, text_2):
11    root_list = []
12    reference_list = []
13    top_coreference = None
14    resolved_text = ""
15
16    if text_1 and text_2:
17      if text_1 is None:
18        text_1 = ""
19      if text_2 is None:
20        text_2 = ""
21      doc = spacy(f"{{text_1}} {{text_2}}")
22      sentence_break = len(list(spacy(text_1).sents)[0])
23
24      for t in doc:
25        # root is a list of tokens
26        root = doc._.coref_chains.resolve(t)
27        if root:
28          root_word = root[0]
29          # Co-references do have to be across sentences and must not be a reference between the same word twice
30          if (
31            t.text != root_word.text
32            and t.i >= sentence_break
33            and root_word.i < sentence_break
34          ):
35            try:
36              root_nb0r = root_word.nb0r(-1)
37              if root_nb0r.pos_ in identifier_reference_pos_tags:
38                rw = root_nb0r.text_with_ws.lower() + root_word.text_with_ws
39              else:
40                rw = root_word.text_with_ws
41            except IndexError:
42              rw = root_word.text_with_ws
43            reference_list.append(t.text)

```

Bild A-73: Algorithmus zum Auflösen von Co-Referenzen (Ausschnitt)

A4.6 Algorithmen zur Klassifizierung von Anforderungen

```
7  from requirements_pipeline.shared.nlp.models import universal_sentence_encoder
8
9  _classification_model = load("./assets/requirements_type_model.joblib")
10
11
12  _batch_size = 200
13
14
15  def _batcher(requirements, requirements_indices):
16      for i in range(0, len(requirements), _batch_size):
17          requirements_batch = requirements[i : i + _batch_size]
18          text_batch = numpy.array([r.text for r in requirements_batch])
19          index_mapping_batch = requirements_indices[i : i + _batch_size]
20          yield text_batch, index_mapping_batch
21
22
23  def classify_requirement_types(requirements, language, debug_print=False):
24      relevant_requirements = [r for r in requirements if r.text is not None]
25      indices = [i for i, r in enumerate(requirements) if r.text is not None]
26      for requirement_text_batch, index_mapping in _batcher(
27          relevant_requirements, indices
28      ):
29          text_vecs = universal_sentence_encoder(requirement_text_batch)
30          prediction_scores = _classification_model.predict(text_vecs)
31          class_keys = prediction_scores.round().astype(int).astype(str)
32          for i, class_key in enumerate(class_keys):
33              requirements[
34                  index_mapping[i]
35              ].classification = requirement_types_class_mapping[class_key]
36  return requirements
37
```

Bild A-74: Klassifizierung von Anforderungen (Ausschnitt)

A4.7 Algorithmen zur Formalisierung von Anforderungen

```

1  import re
2
3  from requirements_pipeline.shared.nlp.models import bikini_ner, spacy
4
5
6  class RegexTypeExtractor:
7      def __init__(self, regex_pattern, ignore_case=False):
8          self.regex_pattern = regex_pattern
9          self.ignore_case = ignore_case
10
11     def extract_from_text(self, requirement_text):
12         if self.ignore_case:
13             matches = re.findall(
14                 self.regex_pattern, requirement_text, flags=re.IGNORECASE
15             )
16         else:
17             matches = re.findall(self.regex_pattern, requirement_text)
18         return [m.strip() for m in matches]
19
20
21 class StringMatchingExtractor(RegexTypeExtractor):
22     def __init__(self, matching_candidates):
23         regex_pattern = (
24             f"\b({':>|:|'.join([re.escape(c) for c in matching_candidates])})\b"
25         )
26         super(StringMatchingExtractor, self).__init__(regex_pattern)
27
28
29 class NumberWithUnitExtractor(RegexTypeExtractor):
30     def __init__(self, unit_candidates):
31         regex_pattern = f'({":>|:|'.join([f"\d+({':|{'.join([re.escape(c.lower()) for c in unit_candidates])}})"])}\b"
32         super(NumberWithUnitExtractor, self).__init__(regex_pattern, ignore_case=True)
33

```

Bild A-75: Implementierte Ansätze zur Formalisierung, Ausschnitt (I/II)

```

35  class NerExtractor:
36      def __init__(self, ner_function, entity_type):
37          self.entity_type = entity_type
38          self.ner_function = ner_function
39
40      def extract_from_text(self, requirement_text):
41          doc = self.ner_function(requirement_text)
42          entities = [
43              entity.text for entity in doc.ents if entity.label_ == self.entity_type
44          ]
45          return entities
46
47
48  class SpacyBuiltinNerExtractor(NerExtractor):
49      def __init__(self, entity_type):
50          super(SpacyBuiltinNerExtractor, self).__init__(spacy, entity_type)
51
52
53  class BikiniNerExtractor(NerExtractor):
54      def __init__(self, entity_type):
55          super(BikiniNerExtractor, self).__init__(bikini_ner, entity_type)
56
57
58  class LocationExtractor(SpacyBuiltinNerExtractor):
59      def __init__(self):
60          super(LocationExtractor, self).__init__("GPE")

```

Bild A-76: Implementierte Ansätze zur Formalisierung, Ausschnitt (II/II)

A4.8 Domänenlexika

```
  "class_id": "component_authentication",
  "data_type": {
    "type_class": "MATCH",
    "matching_candidates": [
      "authentication",
      "piracy",
      "protection",
      "copy",
      "rights",
      "IP",
      "intellectual property",
      "copyright",
      "trademark",
      "patent",
      "secret",
      "security"
    ]
  },
  "text_indicators": [
    "component",
    "part",
    "element",
    "build",
    "workpiece",
    "specimen",
    "chain",
    "supplychain",
    "lifecycle",
    "product"
  ],
  "structure_rules": [
    {
      "rule_type": "classification_type",
      "required_type": [
        "FUNCTIONAL",
        "NON FUNCTIONAL"
      ]
    }
  ]
}
```

Bild A-77: Domänenlexikon "component_authentication"

```
"class_id": "component_lifespan_range",
"data_type": {
  "type_class": "NUMBER_WITH_UNIT",
  "unit_candidates": [
    "km",
    "kilometers",
    "kilometres",
    "m",
    "meter",
    "metre"
  ]
},
"text_indicators": [
  "lifespan",
  "life span",
  "life-span",
  "lifetime",
  "life-time",
  "life",
  "time span",
  "shelf-life",
  "life expectancy",
  "life cycle",
  "life-cycle",
  "continuance",
  "km",
  "kilometers",
  "kilometres",
  "m",
  "meter",
  "metre"
],
"structure_rules": [
  {
    "rule_type": "classification_type",
    "required_type": [
      "FUNCTIONAL"
    ]
  }
]
```

Bild A-78: Domänenlexikon "component_lifespan_range"

```
1  {
2    "class_id": "component_lifespan_time",
3    "data_type": {
4      "type_class": "LIFESPAN_TIME"
5    },
6    "text_indicators": [
7      "lifespan",
8      "life span",
9      "life-span",
10     "lifetime",
11     "life-time",
12     "life",
13     "time span",
14     "shelf-life",
15     "life expectancy",
16     "life cycle",
17     "life-cycle",
18     "continuance",
19     "durability",
20     "usage",
21     "year",
22     "years"
23   ],
24   "structure_rules": [
25     {
26       "rule_type": "classification_type",
27       "required_type": [
28         "FUNCTIONAL",
29         "NON FUNCTIONAL"
30       ]
31     }
32   ]
33 }
34
```

Bild A-79: Domänenlexikon "component_lifespan_time"

```
1  {
2    "class_id": "component_manufacturing_process",
3    "data_type": {
4      "type_class": "MANUFACTURING_PROCESS"
5    },
6    "text_indicators": [
7      "additive manufacturing",
8      "milling",
9      "drilling",
10     "turning",
11     "injection molding",
12     "casting",
13     "die casting",
14     "deep drawing",
15     "hydroforming",
16     "grinding",
17     "painting",
18     "coating",
19     "powder coating",
20     "producing"
21   ],
22   "structure_rules": [
23     {
24       "rule_type": "classification_type",
25       "required_type": [
26         "FUNCTIONAL",
27         "NON FUNCTIONAL"
28       ]
29     }
30   ]
31 }
```

Bild A-80: Domänenlexikon "component_manufacturing_process"

```
1  {
2      "class_id": "component_manufacturing_quantity",
3      "data_type": [
4          "type_class": "NUMBER_WITH_UNIT",
5          "unit_candidates": [
6              "pieces",
7              "per week",
8              "per day",
9              "per year"
10         ],
11     ],
12     "text_indicators": [
13         "pieces",
14         "per week",
15         "per day",
16         "per year"
17     ],
18     "structure_rules": [
19         {
20             "rule_type": "classification_type",
21             "required_type": [
22                 "FUNCTIONAL",
23                 "NON FUNCTIONAL"
24             ],
25         }
26     ],
27 }
```

Bild A-81: Domänenlexikon "component_manufacturing_quantity"

```
1  {  
2      "class_id": "component_traceability",  
3      "data_type": {  
4          "type_class": "MATCH",  
5          "matching_candidates": [  
6              "track",  
7              "trace",  
8              "retrace",  
9              "reproduce",  
10             "tracking",  
11             "tracing",  
12             "marking",  
13             "ID",  
14             "identification",  
15             "scanning",  
16             "code",  
17             "dpm",  
18             "matrix",  
19             "QR",  
20             "barcode",  
21             "serial",  
22             "UID",  
23             "IUID",  
24             "unique",  
25             "symbol",  
26             "label",  
27             "laser",  
28             "etch",  
29             "emboss",  
30             "engrave",  
31             "sticker",  
32             "dot",  
33             "dot peen",  
34             "ink jet"
```

Bild A-82: Domänenlexikon "component_traceability" (I/II)

```
        ],
    },
    "text_indicators": [
        "component",
        "part",
        "element",
        "build",
        "workpiece",
        "specimen",
        "chain",
        "supplychain",
        "lifecycle",
        "product"
    ],
    "structure_rules": [
        {
            "rule_type": "classification_type",
            "required_type": [
                "FUNCTIONAL",
                "NON FUNCTIONAL"
            ]
        }
    ]
}
```

Bild A-83: Domänenlexikon "component_traceability" (II/II)

```
1  {
2      "class_id": "type_of_marking",
3      "data_type": {
4          "type_class": "MATCH",
5          "matching_candidates": [
6              "code",
7              "dpm",
8              "matrix",
9              "QR",
10             "barcode",
11             "serial",
12             "symbol",
13             "label",
14             "laser",
15             "etch",
16             "emboss",
17             "engrave",
18             "sticker",
19             "dot",
20             "dot peen",
21             "ink jet",
22             "RFID"
23         ]
24     },
25     "structure_rules": [
26     {
27         "rule_type": "classification_type",
28         "required_type": [
29             "FUNCTIONAL",
30             "NON FUNCTIONAL"
31         ]
32     }
33 ]
34 }
```

Bild A-84: Domänenlexikon "type_of_marking"

```
1  {
2      "class_id": "component_max_weight",
3      "data_type": {
4          "type_class": "NUMBER_WITH_UNIT",
5          "unit_candidates": [
6              "tonnes",
7              "t",
8              "kg",
9              "g",
10             "gr",
11             "mg",
12             "miligrams",
13             "tons",
14             "ton",
15             "tonne",
16             "kilogram",
17             "kilograms",
18             "gram",
19             "grams"
20         ]
21     },
22     "text_indicators": [
23         "mass",
24         "weight",
25         "weighs",
26         "weigh",
27         "bulk",
28         "load"
29     ],
30     "structure_rules": [
31         {
32             "rule_type": "classification_type",
33             "required_type": [
34                 "FUNCTIONAL",
35                 "NON FUNCTIONAL"
36             ]
37         }
38     ]
39 }
```

Bild A-85: Domänenlexikon "component_max_weight"

```
2   "class_id": "component_material",
3   "data_type": {
4     "type_class": "MATERIAL"
5   },
6   "text_indicators": [
7     "plastic",
8     "polyamide",
9     "thermoplastics",
10    "thermosoft plastic",
11    "thermosets",
12    "thermosetting polymer",
13    "steel",
14    "stainless steel",
15    "aluminium",
16    "aluminum",
17    "carbonfiber",
18    "glass fibre",
19    "rubber",
20    "titanium",
21    "cast iron",
22    "copper",
23    "sheet",
24    "tailored blank",
25    "silicone",
26    "silicon",
27    "component material",
28    "Titan",
29    "lead",
30    "Lithium"
31  ],
32  "structure_rules": [
33    {
34      "rule_type": "classification_type",
35      "required_type": [
36        "FUNCTIONAL",
37        "NON FUNCTIONAL"
38      ]
39    }
40  ]
41
```

Bild A-86: Domänenlexikon "component_material"

```
1  {
2      "class_id": "component_manufacturing_site",
3      "data_type": [
4          "type_class": "LOCATION"
5      ],
6      "text_indicators": [
7          "production",
8          "produce",
9          "producing",
10         "make",
11         "manufacturing",
12         "manufacture",
13         "craft",
14         "fabricate",
15         "come",
16         "smelted"
17     ],
18     "structure_rules": [
19         {
20             "rule_type": "classification_type",
21             "required_type": [
22                 "FUNCTIONAL",
23                 "NON FUNCTIONAL"
24             ]
25         }
26     ]
27 }
28
```

Bild A-87: Domänenlexikon "component_manufacturing_site"

```
1  {
2      "class_id": "component_redesign",
3      "data_type": {
4          "type_class": "MATCH",
5          "matching_candidates": [
6              "adaptation design",
7              "adaption construction",
8              "adaptive design",
9              "adaptive construction",
10             "adaption engineering",
11             "adaptive engineering",
12             "redesign",
13             "new construction",
14             "new design",
15             "new engineering"
16         ]
17     },
18     "text_indicators": [
19         "engineering",
20         "construction",
21         "design",
22         "redesign"
23     ],
24     "structure_rules": [
25         {
26             "rule_type": "classification_type",
27             "required_type": [
28                 "FUNCTIONAL",
29                 "NON FUNCTIONAL"
30             ]
31         }
32     ]
33 }
34 }
```

Bild A-88: Domänenlexikon "component_redesign"

```
1  {  
2      "class_id": "component_volume",  
3      "data_type": {  
4          "type_class": "NUMBER_WITH_UNIT",  
5          "unit_candidates": [  
6              "meters3",  
7              "metres3",  
8              "meter3",  
9              "metre3",  
10             "cubic meters",  
11             "cubic metres",  
12             "cubic meter",  
13             "cubic metre",  
14             "milimeters3",  
15             "milimetres3",  
16             "milimeter3",  
17             "milimetre3",  
18             "cubic milimeters",  
19             "cubic milimetres",  
20             "cubic milimeter",  
21             "cubic milimetre",  
22             "m",  
23             "m3",  
24             "cm3",  
25             "liters",  
26             "litres",  
27             "liter",  
28             "litre",  
29             "l"  
30         ]  
31     },  
32 }
```

Bild A-89: Domänenlexikon "component_volume" (I/II)

```
32     "text_indicators": [
33         "volume",
34         "capacity",
35         "dimension",
36         "size",
37         "mass",
38         "bulk",
39         "magnitude",
40         "cubic measure",
41         "liter",
42         "litre"
43     ],
44     "structure_rules": [
45         {
46             "rule_type": "classification_type",
47             "required_type": [
48                 "FUNCTIONAL",
49                 "NON FUNCTIONAL"
50             ]
51         }
52     ]
53 }
54 }
```

Bild A-90: Domänenlexikon "component_volume" (II/II)

```
1  [
2    "class_id": "delivery_date",
3    "data_type": {
4      "type_class": "DATE"
5    },
6    "text_indicators": [
7      "delivery",
8      "distribution",
9      "deliver",
10     "distribute",
11     "shipment"
12   ],
13   "structure_rules": [
14     {
15       "rule_type": "classification_type",
16       "required_type": [
17         "FUNCTIONAL",
18         "NON FUNCTIONAL"
19       ]
20     }
21   ]
22 ]
23
```

Bild A-91: Domänenlexikon "delivery_date"

```
1  {
2      "class_id": "supplier_selected",
3      "data_type": {
4          "type_class": "SUPPLIER_SELECTED"
5      },
6      "text_indicators": [
7          "production",
8          "produced",
9          "manufacturing",
10         "manufacturer",
11         "company",
12         "supplied",
13         "manufactured",
14         "provided"
15     ],
16     "structure_rules": [
17         {
18             "rule_type": "classification_type",
19             "required_type": [
20                 "FUNCTIONAL",
21                 "NON FUNCTIONAL"
22             ]
23         }
24     ]
25 }
26 |
```

Bild A-92: Domänenlexikon "supplier_selected"

```
1  {
2      "class_id": "component_energy_type",
3      "data_type": {
4          "type_class": "MATCH",
5          "matching_candidates": [
6              "electricity",
7              "fuel",
8              "efuels",
9              "e-fuels",
10             "electrical",
11             "gasoline",
12             "natural gas",
13             "natural gasoline",
14             "hydrogen",
15             "hydrogen fuel cell",
16             "liquefied gas",
17             "liquid gas",
18             "liquefied petroleum gas",
19             "Liquefied Petroleum Gas",
20             "LPG",
21             "compressed natural gas",
22             "CNG"
23         ],
24     },
```

Bild A-93: Domänenlexikon "component_energy_type" (I/II)

```
25 "text_indicators": [  
26     "energy_type",  
27     "electricity",  
28     "fuel",  
29     "efuels",  
30     "e-fuels",  
31     "electrical",  
32     "gasoline",  
33     "natural_gas",  
34     "natural_gasoline",  
35     "hydrogen",  
36     "hydrogen_fuel_cell",  
37     "liquefied_gas",  
38     "liquid_gas",  
39     "liquefied_petroleum_gas",  
40     "Liquefied_Petroleum_Gas",  
41     "LPG",  
42     "compressed_natural_gas",  
43     "CNG"  
44 ],  
45 "structure_rules": [  
46     {  
47         "rule_type": "classification_type",  
48         "required_type": [  
49             "FUNCTIONAL",  
50             "NON_FUNCTIONAL"  
51         ]  
52     }  
53 ]  
54 }  
55 }
```

Bild A-94: Domänenlexikon "component_energy_type" (II/II)

```
2  "class_id": "marking_encoding",
3  "data_type": {
4      "type_class": "MATCH",
5      "matching_candidates": [
6          "UID",
7          "IUID",
8          "GTIN",
9          "EAN",
10         "GLN",
11         "SSCC",
12         "GRAI",
13         "GIAI",
14         "GSRN",
15         "GDTI",
16         "GSIN",
17         "GINC",
18         "GCN",
19         "CPID",
20         "UPC",
21         "IAN",
22         "JAN",
23         "CAS",
24         "OID",
25         "PMID",
26         "UUID",
27         "GUID",
28         "LOT",
29         "REF",
30         "SN",
31         "UDI"
32     ]
33 },
34 "structure_rules": [
35     {
36         "rule_type": "classification_type",
37         "required_type": [
38             "FUNCTIONAL",
```

Bild A-95: Domänenlexikon "marking_encoding"

```
1  {
2      "class_id": "shipment",
3      "data_type": {
4          "type_class": "SHIPMENT"
5      },
6      "text_indicators": [
7          "shipping"
8      ],
9      "structure_rules": [
10         {
11             "rule_type": "classification_type",
12             "required_type": [
13                 "FUNCTIONAL",
14                 "NON FUNCTIONAL"
15             ]
16         }
17     ]
18 }
```

Bild A-96: Domänenlexikon "shipment"

A4.9 Algorithmen zur Abhängigkeitsanalyse von Anforderungen

```
179 for i in range(num_queries):
180     # ...where each iteration consists of labelling 20 samples
181     indices_queried = active_learner.query(num_samples=20)
182
183     print(indices_queried)
184
185     # Simulate user interaction here. Replace this for real-world usage.
186     #y = train.y[[indices_queried]]
187     y = test.y[indices_queried]
188
189     print(y)
190
191     #z = test.y
192     # Return the labels for the current query to the active learner.
193     active_learner.update(y)
194
195     indices_labeled = np.concatenate([indices_queried, indices_labeled])
196
197     #print(indices_labeled)
198
199     print('-----')
200     print(f'Iteration #{i} ({len(indices_labeled)} samples)')
201     #results.append(evaluate(active_learner, train[indices_labeled], test))
202     results.append(evaluate(active_learner, train, test[indices_labeled]))
203
204     stopping_criterion_response = stopping_criterion.stop(predictions=active_learner.classifier.predict(train))
205     print(f'Stop: {stopping_criterion_response}')
206     stopping_history.append(stopping_criterion_response)
207     #if stopping_criterion_response == 'True':
208     #    break
```

Bild A-97: Active Learning Loop (Auszug)

A5 Unterlagen zur Validierung der Methodik

Anforderungsdaten

Aufgrund von Geheimhaltungsvereinbarungen können die Anforderungsdaten der Produkte „Montage Riegel Haube“, „Haubenanpassung Stöpsel“ und „Kofferraumrollo“ nicht dargestellt werden. Es werden die gekennzeichneten Anforderungsdaten der Open-Source Lastenhefte sowie die künstlich-generierten Anforderungsdaten dargestellt, die zur Validierung der Extrahierung von Anforderungen genutzt worden sind. Außerdem werden die Anforderungsdaten des „Knickarmroboters“, der „elektrischen Gewürzmühle“ und des „Toasters“ dargestellt, die für die Validierung der Abhängigkeitsanalyse von Anforderungen verwendet worden sind.

Open-Source Lastenhefte

A	B	C	D
1 Code Description and content Status Support ers:COE 88 Target specie(s)	0	0	fishing
2 Fixed format defined by the pattern X (X = single character alphabetical s)	0	0	fishing
3 List of accepted codes to be found at the EC ERS web site.	0	0	fishing
4 (http://ec.europa.eu/fisheries/cfp/control_enforcement/ers_en.htm)	0	0	fishing
5 (HH = Hours, MM = Minutes.	0 X	0 X	fishing
6 Values must conform to UTC standards.)	0 X	0 X	fishing
7 ers:COE 89b Transzonal attribute GBRTZ Indicates when a vessel is engaged	0	0	fishing
8 Fixed format defined by the pattern Y or N. (Y = vessel has engaged in tra	0	0	fishing
9 ers:COE 88a	0	0	fishing
10 Directed species attribute GBRDS Directed species.	0	0	fishing
11 Must conform to a single FAO (Fisheries and Agricultural Organisation) sp	0	0	fishing
12 Supported as a Norwegian requirement.	0	0	fishing
13 Code Description and content Status Support ers:COX 92 Start of Effort de	0	0	fishing
14 Used to declare exit out of a stock recovery area or Western Waters.	0	0	fishing
15 COX 93 Date attribute DA	0	0	fishing
16 The date of the exit being reported.	0	0	fishing
17 Fixed format defined by the pattern: YYYY- MM-DD (YYYY =	0 X	0 X	fishing
18 Year, MM = Month, DD = Date.	0 X	0 X	fishing
19 Values must conform to UTC standards.)	0 X	0 X	fishing
20 4.4.3.6 Exit from Zone Declaration (COX)	0	0	fishing
21 Copyright (c) 2009,2010 UK Fisheries Administrations All Rights Reserved	0	0	fishing
22 PROTECT - COMMERCIAL [when completed]	0	0	fishing
23 Code Description and content Status Support COX 95a Fishing effort zone	0	0	fishing
24 List of accepted codes to be found at the UK Fisheries web site.	0	0	fishing
25 (http://www.fishregister.gov.uk/schema/ers/v1)	0	0	fishing
26 ers:COX 95 Target specie(s) attribute TS Species targeted within zone.	0	0	fishing
27 Fixed format defined by the pattern X (X = single character alphabetical s	0	0	fishing
28 List of accepted codes to be found at the EC ERS web site.	0	0	fishing
29 (http://ec.europa.eu/fisheries/cfp/control_enforcement/ers_en.htm) C	0	0	fishing
30 Fixed format defined by the pattern: HH:MM (HH = Hours, MM = Minutes.	0 X	0 X	fishing
31 Values must conform to UTC standards.)	0 X	0 X	fishing
32 COX 97a Transzonal attribute GBRTRZ Indicator of whether a vessel has ei	0	0	fishing
33 Fixed format defined by the pattern Y or N. (Y = vessel has engaged in tra	0	0	fishing
34 Code Description and content Status Support Copyright (c) 2009,2010 UK I	0	0	fishing
35 Illegit to make a transzonal fishing declaration	0	0	fishing

Bild A-98: Auszug aus einem Open-Source Lastenheft

Künstliche Anforderungsdaten

1	In the meantime, it stores values in \$seconds_left and displays a text saying you have left 10:45 minutes after which you can reset as usual by	2
2	regulations issued in conjunction with these subparts, and be at least inches high x inches wide to accommodate a passport photo for United	2
3	cm or less while 100 percent will still view it at 50 cm.	2
4	meters and 100% will require it at 40 metres.	2
5	-50m, with 40-60 degrees FOV and 100 degree horizontal F	2
6	feet, including text and symbols.	2
7	After that point, a monthly subscription will apply for the	2
8	, including through the use of a token provided with the device upon enrollment.	2
9	, including authentication through biometric means (e.g fingerprint) or the use of software-	2
10	, except when the identification information is altered or incorrect.	2
11	(see definition 2, section 4 "access system" 4 - No person may install or make	2
12	The best products of this sort are going to be based around GPS which indicates that you'll find them in a position to offer details like distance	2
13	Therefore, it is difficult for you to monitor all your equipment continuously and instantly when monitoring by a touch on device which has low	2
14	It is a must that your website offers users an easy way of getting what they want and keeping them interested and visiting you again, which is	2
15	The graphs must show trends in terms of change over time, but it is also acceptable to do comparisons between specific variables at different	1
16	The report can be printed from any location with no requirement to purchase software or hardware, since it uses PDF files as opposed to TIFF	1
17	Clicking an event will highlight it and open it using the Event-ID: protocol, e.g.: "savadata#event:save_project	1
18	Each event must have an ID assigned when it is added to the database and each Event has start date and end dates, which can be set from year	1
19	When the user presses on an Event name it should take them to another view with more details about that event (e-mail addresses/dates of i	1
20	A given event may have an end date (if you want to show it before its beginning) and another begin date if we wish to see its progress over t	1
21	All events occurring on that same day will be displayed to show a more complete view of what happened during those days than just listing all	1
22	The following activities and events have been provided by a local government..	1
23	Events shall be defined as any set of time-based data about one or more entities (people, places, computers) that have occurred within a spec	1
24	The primary purpose of this system is to provide a means for displaying such things as Event Tickets, Program Information, Schedules and anyt	1
25	The Event and Activity system can be accessed by the users.	1
26	which will be displayed on this screen, and it will move left when an application calls for that movement.	1
27	(mean scores with standard error) of standardized residuals.	1
28	obtained from mice aged 13-16 weeks.	1
29	This will allow for a more efficient implementation, as not all registers will need to store an entire frame buffer (the full screen size) in memc	1
30	The display region contains a first area (sequential) in which at any given instant, an object or objects can be displayed on one half of the displ	1
31	These are denoted by T1 to T4, which have distinct visual properties depending on their status within a sequence of frames or events (for more	1
32	When an event is successfully executed, this action resets the timer associated with that column back to zero, and sets it again for its next invi	1
33	Events in the bottom three quarters may be held simultaneously.	1
34	The first quarter will be used for initializing things like game state and level logic, then we have half where all of our gameplay is happening u	1
35	The bottom 3 quarters holds the list.	1
36	I am currently using a timestamp column on the top 1-3 rows which is indexed in the data.	1
37	In the case of an event occurring outside it's expected range, it should be flagged as "not normal i.e not blue) and show a numerical deviation	1
38	An event that has already occurred is given the color light yellow, and an event for which the next occurrence can be predicted with 100% cert	1

Bild A-99: Auszug aus den künstlichen Anforderungsdaten

Knickarmroboter

1 Anforderungstext	Dependency
2 The articulated arm robot must offer the possibility to connect a transformer The articulated arm robot must be capable of transforming 230 volts alternating current	1
3 The jointed-arm robot must be able to supply hardware components with different volts of direct current The articulated arm robot must be capable of transforming 2	1
4 The articulated arm robot must be able to supply the stepper motors with 12 volt direct current The articulated arm robot must be capable of transforming 230 volts al	0
5 The jointed-arm robot must be able to supply a control board with 12 volts DC. The articulated arm robot must be capable of transforming 230 volts alternating current	0
6 The jointed-arm robot must be able to dissipate heat generated by electrical components The articulated arm robot must be capable of transforming 230 volts alterna	0
7 The jointed-arm robot must be able to cool the transformer sufficiently The articulated arm robot must be capable of transforming 230 volts alternating current	0
8 The jointed-arm robot must be able to supply the fans with 24 volts direct current The articulated arm robot must be capable of transforming 230 volts alternating curr	0
9 The jointed-arm robot must be able to protect electronic components The articulated arm robot must be capable of transforming 230 volts alternating current	0
10 The jointed-arm robot must be able to avoid short-circuiting of electronic components The articulated arm robot must be capable of transforming 230 volts alternating	0
11 The articulated arm robot must be able to blow the fuse in the event of a short circuit The articulated arm robot must be capable of transforming 230 volts alternating	0
12 The articulated arm robot must be able to supply the Stepper Diver evenly with 12 volts direct current. The articulated arm robot must be capable of transforming 230	0
13 The articulated arm robot must be able to connect a 24 volt DC converter. The articulated arm robot must be capable of transforming 230 volts alternating current	0
14 The articulated arm robot must be able to regulate the amperage The articulated arm robot must be capable of transforming 230 volts alternating current	0
15 The articulated arm robot must be able to regulate the amperes differently for each component The articulated arm robot must be capable of transforming 230 volts al	0
16 The jointed-arm robot must be able to connect stepper drivers with ampeire converter The articulated arm robot must be capable of transforming 230 volts alternating	0
17 The jointed-arm robot must be able to protect the human health The articulated arm robot must be capable of transforming 230 volts alternating current	0
18 The jointed-arm robot must be able to supply an Arduino Mega with 12 volts DC The articulated arm robot must be capable of transforming 230 volts alternating curr	0
19 The articulated arm robot must be able to protect the cabling The articulated arm robot must be capable of transforming 230 volts alternating current	0
20 The articulated arm robot must be able to be deactivated with an emergency stop switch The articulated arm robot must be capable of transforming 230 volts alternati	0
21 The articulated arm robot must be able to be completely deactivated by means of a mushroom button emergency switch The articulated arm robot must be capable of	0
22 The jointed-arm robot must be able to supply power to hardware components The articulated arm robot must be capable of transforming 230 volts alternating current	0
23 The articulated arm robot must be able to prevent hardware components from overheating The articulated arm robot must be capable of transforming 230 volts altern	0
24 The jointed-arm robot must be able to cool hardware components permanently with fans The articulated arm robot must be capable of transforming 230 volts alternat	0
25 The jointed-arm robot must be able to accept input information The articulated arm robot must be capable of transforming 230 volts alternating current	0
26 The jointed-arm robot must be able to store programs on an internal data carrier The articulated arm robot must be capable of transforming 230 volts alternating cur	0
27 The articulated arm robot must be able to be controlled with a control unit The articulated arm robot must be capable of transforming 230 volts alternating current	0
28 The jointed-arm robot must be able to control the individual stepper motors separately The articulated arm robot must be capable of transforming 230 volts alternati	0
29 The articulated arm robot must be able to communicate with an external computer The articulated arm robot must be capable of transforming 230 volts alternating cu	0
30 The articulated arm robot must be able to process the control signals of the external computer The articulated arm robot must be capable of transforming 230 volts alt	0
31 The jointed-arm robot must be able to interconnect hardware components The articulated arm robot must be capable of transforming 230 volts alternating current	0
32 The jointed-arm robot must be able to send a control pulse to the individual stepper motors The articulated arm robot must be capable of transforming 230 volts alter	0
33 The jointed-arm robot must be able to convert the digital communication of the computer into an analog signal The articulated arm robot must be capable of transfor	0
34 The jointed-arm robot must be able to change the direction of rotation of the stepper motors The articulated arm robot must be capable of transforming 230 volts alte	0
35 The jointed-arm robot must offer the possibility to control the position of the gripper arm The articulated arm robot must be capable of transforming 230 volts alterna	0
36 The jointed-arm robot must offer the possibility to open and close the gripper arm The articulated arm robot must be capable of transforming 230 volts alternating cur	0
37 The jointed-arm robot must offer the possibility to send output signals The articulated arm robot must be capable of transforming 230 volts alternating current	0
38 The jointed-arm robot must be able to send the signals of the surveillance camera The articulated arm robot must be capable of transforming 230 volts alternating cur	0
39 The articulated arm robot must be able to communicate with a user interface The articulated arm robot must be capable of transforming 230 volts alternating current	0

Bild A-100. Auszug aus den Anforderungsdaten des Knickarmroboters

Elektrische Gewürzmühle

A	B	Dependency
1 Anforderungstext		
2 The diameter of the housing should be a maximum of 40mm The housing should have a maximum height of 150mm	1	1
3 The diameter of the housing should be a maximum of 40mm The capacity for the ground material should be at least 100cm ³	1	1
4 The diameter of the housing should be a maximum of 40mm The upper and middle parts of the case shall be made of polished 18/10 stainless steel	0	0
5 The diameter of the housing should be a maximum of 40mm The middle part of the housing should be made of glass	0	0
6 The diameter of the housing should be a maximum of 40mm The grinder should be made of ceramic	0	0
7 The diameter of the housing should be a maximum of 40mm Whole peppercorns should be ground into fine crumbs of pepper	0	0
8 The diameter of the housing should be a maximum of 40mm The pepper/salt mill is to be operated electrically with 4 AA type batteries	1	1
9 The diameter of the housing should be a maximum of 40mm The weight of the pepper/salt mill should not exceed 250g without grinding material and batteries	1	1
10 The diameter of the housing should be a maximum of 40mm The grinder should work as frictionless as possible	0	0
11 The diameter of the housing should be a maximum of 40mm The motor and the grinder should generate as little heat as possible	1	1
12 The diameter of the housing should be a maximum of 40mm In the open state for filling with ground material, the grinding mechanism should be decoupled from the power supply for safety reasons	0	0
13 The diameter of the housing should be a maximum of 40mm For hygienic reasons, when not in use, the ground material and the grinder should be protected by a cover	0	0
14 The diameter of the housing should be a maximum of 40mm The electric pepper/salt mill should be operable by a pressure switch.	0	0
15 The diameter of the housing should be a maximum of 40mm It should be possible to read the filling level of the grinding material on the device	0	0
16 The diameter of the housing should be a maximum of 40mm It should be possible to adjust the fineness of the ground material via an adjustment screw on the grinder	0	0
17 The diameter of the housing should be a maximum of 40mm The case should offer a high-quality feel for the user	1	1
18 The diameter of the housing should be a maximum of 40mm An indicator of the available energy should be installed to control the battery level	0	0
19 The diameter of the housing should be a maximum of 40mm An LED should be installed for a better overview of the filling level of the chamber for the ground material	0	0
20 The diameter of the housing should be a maximum of 40mm Accessibility to the battery compartment should be ensured as effortlessly as possible	1	1
21 The diameter of the housing should be a maximum of 40mm Thanks to a split housing, Cleaning of the grinder and the chamber of the ground material should be possible	0	0
22 The diameter of the housing should be a maximum of 40mm In use, the electric pepper/salt mill should operate as quietly as possible	0	0
23 The diameter of the housing should be a maximum of 40mm The ceramic grinder should be operational for at least 100 hours of use	0	0
24 The diameter of the housing should be a maximum of 40mm The maximum allowable manufacturing cost per unit should be less than €9.50	1	1
25 The diameter of the housing should be a maximum of 40mm The delivery date should be no later than 01.01.2018	0	0
26 The diameter of the housing should be a maximum of 40mm An indicator of the available energy should be installed to control the battery level	1	1
27 The housing should have a maximum height of 150mm The capacity for the ground material should be at least 100cm ³	0	0
28 The housing should have a maximum height of 150mm The upper and middle parts of the case shall be made of polished 18/10 stainless steel	0	0
29 The housing should have a maximum height of 150mm The middle part of the housing should be made of glass	1	1
30 The housing should have a maximum height of 150mm Whole peppercorns should be ground into fine crumbs of pepper	0	0
31 The housing should have a maximum height of 150mm The pepper/salt mill is to be operated electrically with 4 AA type batteries	1	1
32 The housing should have a maximum height of 150mm The weight of the pepper/salt mill should not exceed 250g without grinding material and batteries	1	1
33 The housing should have a maximum height of 150mm The grinder should work as frictionless as possible	0	0
34 The housing should have a maximum height of 150mm The motor and the grinder should generate as little heat as possible	0	0
35 The housing should have a maximum height of 150mm In the open state for filling with ground material, the grinding mechanism should be decoupled from the power supply for safety reasons	0	0
36 The housing should have a maximum height of 150mm For hygienic reasons, when not in use, the ground material and the grinder should be protected by a cover	0	0
37 The housing should have a maximum height of 150mm The electric pepper/salt mill is to be operated by a pressure switch.	0	0
38 The housing should have a maximum height of 150mm It should be possible to read the filling level of the grinding material on the device	0	0
39 The housing should have a maximum height of 150mm It should be possible to adjust the fineness of the ground material via an adjustment screw on the grinder	0	0
40 The housing should have a maximum height of 150mm The case should offer a high-quality feel for the user	1	1
41 The housing should have a maximum height of 150mm An indicator of the available energy should be installed to control the battery level	0	0

Bild A-101: Auszug aus den Anforderungsdaten der elektrischen Gewürzmühle

Toaster

A	B	Dependency
1 Anforderungstext		
2 For even browning of both sides of the toast, a bread slice centering device should be installed. The housing should not exceed a height of 20 cm at any point.	1	1
3 For even browning of both sides of the toast, a bread slice centering device should be installed. A roasting attachment is to be integrated into the housing.	1	1
4 For even browning of both sides of the toast, a bread slice centering device should be installed. 2 parallel extra long slots to toast four loaves at the same time.	2	2
5 For even browning of both sides of the toast, a bread slice centering device should be installed. The case should have a length of 35 cm.	2	2
6 For even browning of both sides of the toast, a bread slice centering device should be installed. The width of the housing should be 13 cm.	2	2
7 For even browning of both sides of the toast, a bread slice centering device should be installed. The case should be detachable to remove breadcrumbs.	0	0
8 For even browning of both sides of the toast, a bread slice centering device should be installed. Maximum power of 1000 W.	0	0
9 For even browning of both sides of the toast, a bread slice centering device should be installed. Three adjustable power levels / browning levels.	0	0
10 For even browning of both sides of the toast, a bread slice centering device should be installed. Heat insulating stainless steel housing.	0	0
11 For even browning of both sides of the toast, a bread slice centering device should be installed. The maximum weight of the toaster without contents must be less than 1.7 kg.	1	1
12 For even browning of both sides of the toast, a bread slice centering device should be installed. Do not allow deformation of the housing material due to heat.	1	1
13 For even browning of both sides of the toast, a bread slice centering device should be installed. Stop button for manual termination.	0	0
14 For even browning of both sides of the toast, a bread slice centering device should be installed. Stop button for manual termination.	0	0
15 For even browning of both sides of the toast, a bread slice centering device should be installed. Lift function for easy removal of the loaves.	0	0
16 For even browning of both sides of the toast, a bread slice centering device should be installed. Light up during operation.	0	0
17 For even browning of both sides of the toast, a bread slice centering device should be installed. No sharp edges on the lift lever.	0	0
18 For even browning of both sides of the toast, a bread slice centering device should be installed. A multifunctional knob for controlling and turning on the toaster.	0	0
19 For even browning of both sides of the toast, a bread slice centering device should be installed. Set the upper edge of the housing higher than the heating plate and round it inward to counteract possible deformation.	0	0
20 For even browning of both sides of the toast, a bread slice centering device should be installed. Set the upper edge of the housing higher than the heating plate and round it inward to counteract possible deformation.	0	0
21 For even browning of both sides of the toast, a bread slice centering device should be installed. Stop button for manual termination.	0	0
22 For even browning of both sides of the toast, a bread slice centering device should be installed. End of development: 31.1.2018.	2	2
23 For even browning of both sides of the toast, a bread slice centering device should be installed. Start of series production: February 2018.	0	0
24 For even browning of both sides of the toast, a bread slice centering device should be installed. Delivery time: 12 weeks after production start.	0	0
25 For even browning of both sides of the toast, a bread slice centering device should be installed. Delivery time: 12 weeks after production start.	0	0
26 The housing should not exceed a height of 20 cm at any point. For even browning of both sides of the toast, a bread slice centering device should be installed.	1	1
27 The housing should exceed a height of 20 cm at any point. A roasting attachment is to be integrated into the housing.	1	1
28 The housing should not exceed a height of 20 cm at any point. The case should have a length of 35 cm.	1	1
29 The housing should not exceed a height of 20 cm at any point. The width of the housing should be 13 cm.	1	1
30 The housing should not exceed a height of 20 cm at any point. The bottom of the case should be detachable to remove breadcrumbs.	1	1
31 The housing should not exceed a height of 20 cm at any point. Maximum power of 1000 W.	0	0
32 The housing should not exceed a height of 20 cm at any point. Three adjustable power levels / browning levels.	0	0
33 The housing should not exceed a height of 20 cm at any point. Heat insulating stainless steel housing.	0	0
34 The housing should not exceed a height of 20 cm at any point. Do not allow deformation of the housing material due to heat.	2	2
35 The housing should not exceed a height of 20 cm at any point. No sharp edges on the lift lever.	0	0
36 The housing should not exceed a height of 20 cm at any point. No excessive spring stiffness when inserting the breads.	0	0
37 The housing should not exceed a height of 20 cm at any point. Stop button for manual termination.	1	1

Bild A-102: Auszug aus den Anforderungsdaten des Toasters

Ausgefüllte Bewertungsbögen

Bewertungsbogen 1

- 1) Zur Extrahierung von Anforderungen aus Lastenheften ist ein hoher manueller Aufwand notwendig.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 2) Natürlich-sprachliche Anforderungen sind durch Software-Werkzeuge in der Produktentwicklung üblicherweise nicht direkt interpretierbar. Es besteht ein Bedarf nach formalisierten Anforderungen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 3) Anforderungsänderungen werden in Unternehmen noch nicht systematisch gehandhabt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 4) Ich kann mir sehr gut vorstellen, das Software-Werkzeug regelmäßig zu nutzen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 5) Ich empfinde das Software-Werkzeug als unnötig komplex.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

- 6) Ich denke, dass ich technischen Support brauchen würde, um das Software-Werkzeug zu nutzen.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

- 7) Ich finde, dass es im Software-Werkzeug zu viele Inkonsistenzen gibt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	----------------------------

- 8) Ich kann mir vorstellen, dass die meisten Leute das Software-Werkzeug schnell zu beherrschen lernen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

- 9) Ich muss voraussichtlich eine Menge Dinge lernen, bevor ich mit dem Software-Werkzeug arbeiten kann.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

- 10) Das Werkzeug ist für branchenübliche Lastenhefte anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

11) Das Werkzeug ist generisch für die Entwicklung komplexer technischer Systeme anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

12) Ich kann mir vorstellen, dass die Anwendung des Werkzeugs in unternehmensinterne Entwicklungsprozesse integriert werden kann.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

13) Die Informationen zur Anwendung des Werkzeugs sind in der Praxis frühzeitig (in der Regel ab der Projektinitierungsphase) in Form von Daten oder Expertenwissen verfügbar:

	<i>ja</i>	<i>nein</i>
a) Natürlich-sprachliche Anforderungsspezifikation, bzw. Lastenheft (Datengrundlage zur Anwendung der Extrahierung und Klassifizierung von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
b) Wissen zur Definition von Strukturregeln von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
c) Wissen zur Definition von Textindikatoren von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
d) Gekennzeichnete Daten zum Training von Named-Entity-Recognition Modellen für Schlüssel-Wert-Paare (KI-Modelle zur Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
e) Anforderungsabhängigkeiten (natürlich-sprachliche Anforderungspaare sowie die Abhängigkeit zwischen den Anforderungen) für das initiale Training eines KI-Modells (Abhängigkeitsanalyse von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
f) Wissen zur Bewertung von Anforderungsabhängigkeiten (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
g) Anzahl der aufgetretenen Anforderungsänderungen in vergangenen Projekten (Änderungshistorie)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
h) Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen (Risikomanagement von Anforderungsänderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

14) Den Vorbereitungsaufwand, bevor die Anwendung möglich ist, empfinde ich als angemessen: Erstellen eines Domänenlexikons; Auswahl und Vorbereitung von Anforderungsabhängigkeitsdaten; Zusammenstellung einer Änderungshistorie vergangener Projekte;

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

15) Den Anwendungsaufwand schätze ich als angemessen ein: Import von Lastenheften, Prüfen und Auswahl von Schlüssel-Wert-Paaren, Abhängigkeitsanalyse, Bewertung von Änderungsauswirkungen; Ausgabe der Ergebnisse im CSV-Format

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

16) Durch die automatisierte Extrahierung von Anforderungen aus Lastenheften wird Aufwand reduziert und es wird erleichtert, eine vollständige Anforderungsliste zu generieren.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

17) Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

18) Auf Grundlage der Anforderungsabhängigkeiten können Auswirkungen von Anforderungsänderungen besser bewertet werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

19) Auf Grundlage der Informationen zum Änderungsrisiko von Anforderungen können Risiken realistischer abgeschätzt werden als zuvor.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

20) Anhand der teilautomatisierten Risikoanalyse kann ein höherer Anteil an Anforderungen eines Anforderungssets auf Änderungsrisiken untersucht werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

21) Auf Grundlage des Änderungsrisikos und der darauf aufbauenden Vorschläge zu Risikosteuerungsmaßnahmen können effizienter Maßnahmen definiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

22) Die Anwendung des Software-Werkzeugs trägt zur Effizienzsteigerung in der Entwicklung komplexer technischer Systeme bei.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

23) Durch den CSV-Export können die Anforderungsdaten aufwandsarm in branchenübliche Anforderungsmanagement-Software importiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

Bewertungsbogen 2

- 1) Zur Extrahierung von Anforderungen aus Lastenheften ist ein hoher manueller Aufwand notwendig.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 2) Natürlich-sprachliche Anforderungen sind durch Software-Werkzeuge in der Produktentwicklung üblicherweise nicht direkt interpretierbar. Es besteht ein Bedarf nach formalisierten Anforderungen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 3) Anforderungsänderungen werden in Unternehmen noch nicht systematisch gehandhabt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 4) Ich kann mir sehr gut vorstellen, das Software-Werkzeug regelmäßig zu nutzen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 5) Ich empfinde das Software-Werkzeug als unnötig komplex.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 6) Ich denke, dass ich technischen Support brauchen würde, um das Software-Werkzeug zu nutzen.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 7) Ich finde, dass es im Software-Werkzeug zu viele Inkonsistenzen gibt.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 8) Ich kann mir vorstellen, dass die meisten Leute das Software-Werkzeug schnell zu beherrschen lernen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 9) Ich muss voraussichtlich eine Menge Dinge lernen, bevor ich mit dem Software-Werkzeug arbeiten kann.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 10) Das Werkzeug ist für branchenübliche Lastenhefte anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 11) Das Werkzeug ist generisch für die Entwicklung komplexer technischer Systeme anwendbar.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

12) Ich kann mir vorstellen, dass die Anwendung des Werkzeugs in unternehmensinterne Entwicklungsprozesse integriert werden kann.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

13) Die Informationen zur Anwendung des Werkzeugs sind in der Praxis frühzeitig (in der Regel ab der Projektinitierungsphase) in Form von Daten oder Expertenwissen verfügbar:

		ja	nein
a)	Natürlich-sprachliche Anforderungsspezifikation, bzw. Lastenheft (Datengrundlage zur Anwendung der Extrahierung und Klassifizierung von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
b)	Wissen zur Definition von Strukturregeln von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
c)	Wissen zur Definition von Textindikatoren von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
d)	Gekennzeichnete Daten zum Training von Named-Entity-Recognition Modellen für Schlüssel-Wert-Paare (KI-Modelle zur Formalisierung von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
e)	Anforderungsabhängigkeiten (natürlich-sprachliche Anforderungspaare sowie die Abhängigkeit zwischen den Anforderungen) für das initiale Training eines KI-Modells (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
f)	Wissen zur Bewertung von Anforderungsabhängigkeiten (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input type="checkbox"/>
g)	Anzahl der aufgetretenen Anforderungsänderungen in vergangenen Projekten (Änderungshistorie)	<input type="checkbox"/>	<input type="checkbox"/>
h)	Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen (Risikomanagement von Anforderungsänderungen)	<input type="checkbox"/>	<input type="checkbox"/>

14) Den Vorbereitungsaufwand, bevor die Anwendung möglich ist, empfinde ich als angemessen: Erstellen eines Domänenlexikons; Auswahl und Vorbereitung von Anforderungsabhängigkeitsdaten; Zusammenstellung einer Änderungshistorie vergangener Projekte;

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

15) Den Anwendungsaufwand schätze ich als angemessen ein: Import von Lastenheften, Prüfen und Auswahl von Schlüssel-Wert-Paaren, Abhängigkeitsanalyse, Bewertung von Änderungsauswirkungen; Ausgabe der Ergebnisse im CSV-Format

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

16) Durch die automatisierte Extrahierung von Anforderungen aus Lastenheften wird Aufwand reduziert und es wird erleichtert, eine vollständige Anforderungsliste zu generieren.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

17) Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

18) Auf Grundlage der Anforderungsabhängigkeiten können Auswirkungen von Anforderungsänderungen besser bewertet werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

19) Auf Grundlage der Informationen zum Änderungsrisiko von Anforderungen können Risiken realistischer abgeschätzt werden als zuvor.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

20) Anhand der teilautomatisierten Risikoanalyse kann ein höherer Anteil an Anforderungen eines Anforderungssets auf Änderungsrisiken untersucht werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

21) Auf Grundlage des Änderungsrisikos und der darauf aufbauenden Vorschläge zu Risikosteuerungsmaßnahmen können effizienter Maßnahmen definiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

22) Die Anwendung des Software-Werkzeugs trägt zur Effizienzsteigerung in der Entwicklung komplexer technischer Systeme bei.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

23) Durch den CSV-Export können die Anforderungsdaten aufwandsarm in branchenübliche Anforderungsmanagement-Software importiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

Bewertungsbogen 3

- 1) Zur Extrahierung von Anforderungen aus Lastenheften ist ein hoher manueller Aufwand notwendig.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 2) Natürlich-sprachliche Anforderungen sind durch Software-Werkzeuge in der Produktentwicklung üblicherweise nicht direkt interpretierbar. Es besteht ein Bedarf nach formalisierten Anforderungen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 3) Anforderungsänderungen werden in Unternehmen noch nicht systematisch gehandhabt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 4) Ich kann mir sehr gut vorstellen, das Software-Werkzeug regelmäßig zu nutzen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 5) Ich empfinde das Software-Werkzeug als unnötig komplex.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 6) Ich denke, dass ich technischen Support brauchen würde, um das Software-Werkzeug zu nutzen.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

- 7) Ich finde, dass es im Software-Werkzeug zu viele Inkonsistenzen gibt.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 8) Ich kann mir vorstellen, dass die meisten Leute das Software-Werkzeug schnell zu beherrschen lernen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 9) Ich muss voraussichtlich eine Menge Dinge lernen, bevor ich mit dem Software-Werkzeug arbeiten kann.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 10) Das Werkzeug ist für branchenübliche Lastenhefte anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 11) Das Werkzeug ist generisch für die Entwicklung komplexer technischer Systeme anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

12) Ich kann mir vorstellen, dass die Anwendung des Werkzeugs in unternehmensinterne Entwicklungsprozesse integriert werden kann.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

13) Die Informationen zur Anwendung des Werkzeugs sind in der Praxis frühzeitig (in der Regel ab der Projektinitierungsphase) in Form von Daten oder Expertenwissen verfügbar:

		ja	nein
a)	Natürlich-sprachliche Anforderungsspezifikation, bzw. Lastenheft (Datengrundlage zur Anwendung der Extrahierung und Klassifizierung von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
b)	Wissen zur Definition von Strukturregeln von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
c)	Wissen zur Definition von Textindikatoren von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
d)	Gekennzeichnete Daten zum Training von Named-Entity-Recognition Modellen für Schlüssel-Wert-Paare (KI-Modelle zur Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
e)	Anforderungsabhängigkeiten (natürlich-sprachliche Anforderungspaare sowie die Abhängigkeit zwischen den Anforderungen) für das initiale Training eines KI-Modells (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
f)	Wissen zur Bewertung von Anforderungsabhängigkeiten (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
g)	Anzahl der aufgetretenen Anforderungsänderungen in vergangenen Projekten (Änderungshistorie)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
h)	Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen (Risikomanagement von Anforderungsänderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>

14) Den Vorbereitungsaufwand, bevor die Anwendung möglich ist, empfinde ich als angemessen: Erstellen eines Domänenlexikons; Auswahl und Vorbereitung von Anforderungsabhängigkeitsdaten; Zusammenstellung einer Änderungshistorie vergangener Projekte;

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

15) Den Anwendungsaufwand schätze ich als angemessen ein: Import von Lastenheften, Prüfen und Auswahl von Schlüssel-Wert-Paaren, Abhängigkeitsanalyse, Bewertung von Änderungsauswirkungen; Ausgabe der Ergebnisse im CSV-Format

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

16) Durch die automatisierte Extrahierung von Anforderungen aus Lastenheften wird Aufwand reduziert und es wird erleichtert, eine vollständige Anforderungsliste zu generieren.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

17) Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

18) Auf Grundlage der Anforderungsabhängigkeiten können Auswirkungen von Anforderungsänderungen besser bewertet werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

19) Auf Grundlage der Informationen zum Änderungsrisiko von Anforderungen können Risiken realistischer abgeschätzt werden als zuvor.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

20) Anhand der teilautomatisierten Risikoanalyse kann ein höherer Anteil an Anforderungen eines Anforderungssets auf Änderungsrisiken untersucht werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

21) Auf Grundlage des Änderungsrisikos und der darauf aufbauenden Vorschläge zu Risikosteuerungsmaßnahmen können effizienter Maßnahmen definiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

22) Die Anwendung des Software-Werkzeugs trägt zur Effizienzsteigerung in der Entwicklung komplexer technischer Systeme bei.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

23) Durch den CSV-Export können die Anforderungsdaten aufwandsarm in branchenübliche Anforderungsmanagement-Software importiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

Bewertungsbogen 4

- 1) Zur Extrahierung von Anforderungen aus Lastenheften ist ein hoher manueller Aufwand notwendig.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 2) Natürlich-sprachliche Anforderungen sind durch Software-Werkzeuge in der Produktentwicklung üblicherweise nicht direkt interpretierbar. Es besteht ein Bedarf nach formalisierten Anforderungen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 3) Anforderungsänderungen werden in Unternehmen noch nicht systematisch gehandhabt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 4) Ich kann mir sehr gut vorstellen, das Software-Werkzeug regelmäßig zu nutzen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 5) Ich empfinde das Software-Werkzeug als unnötig komplex.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

- 6) Ich denke, dass ich technischen Support brauchen würde, um das Software-Werkzeug zu nutzen.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

- 7) Ich finde, dass es im Software-Werkzeug zu viele Inkonsistenzen gibt.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 8) Ich kann mir vorstellen, dass die meisten Leute das Software-Werkzeug schnell zu beherrschen lernen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 9) Ich muss voraussichtlich eine Menge Dinge lernen, bevor ich mit dem Software-Werkzeug arbeiten kann.

<input checked="" type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
--	-----------------------------	----------------------------	----------------------------	----------------------------

- 10) Das Werkzeug ist für branchenübliche Lastenhefte anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 11) Das Werkzeug ist generisch für die Entwicklung komplexer technischer Systeme anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

12) Ich kann mir vorstellen, dass die Anwendung des Werkzeugs in unternehmensinterne Entwicklungsprozesse integriert werden kann.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

13) Die Informationen zur Anwendung des Werkzeugs sind in der Praxis frühzeitig (in der Regel ab der Projektinitierungsphase) in Form von Daten oder Expertenwissen verfügbar:

		ja	nein
a)	Natürlich-sprachliche Anforderungsspezifikation, bzw. Lastenheft (Datengrundlage zur Anwendung der Extrahierung und Klassifizierung von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
b)	Wissen zur Definition von Strukturregeln von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
c)	Wissen zur Definition von Textindikatoren von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
d)	Gekennzeichnete Daten zum Training von Named-Entity-Recognition Modellen für Schlüssel-Wert-Paare (KI-Modelle zur Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
e)	Anforderungsabhängigkeiten (natürlich-sprachliche Anforderungspaare sowie die Abhängigkeit zwischen den Anforderungen) für das initiale Training eines KI-Modells (Abhängigkeitsanalyse von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
f)	Wissen zur Bewertung von Anforderungsabhängigkeiten (Abhängigkeitsanalyse von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
g)	Anzahl der aufgetretenen Anforderungsänderungen in vergangenen Projekten (Änderungshistorie)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
h)	Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen (Risikomanagement von Anforderungsänderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

14) Den Vorbereitungsaufwand, bevor die Anwendung möglich ist, empfinde ich als angemessen: Erstellen eines Domänenlexikons; Auswahl und Vorbereitung von Anforderungsabhängigkeitsdaten; Zusammenstellung einer Änderungshistorie vergangener Projekte;

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

15) Den Anwendungsaufwand schätze ich als angemessen ein: Import von Lastenheften, Prüfen und Auswahl von Schlüssel-Wert-Paaren, Abhängigkeitsanalyse, Bewertung von Änderungsauswirkungen; Ausgabe der Ergebnisse im CSV-Format

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

16) Durch die automatisierte Extrahierung von Anforderungen aus Lastenheften wird Aufwand reduziert und es wird erleichtert, eine vollständige Anforderungsliste zu generieren.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

17) Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

18) Auf Grundlage der Anforderungsabhängigkeiten können Auswirkungen von Anforderungsänderungen besser bewertet werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

19) Auf Grundlage der Informationen zum Änderungsrisiko von Anforderungen können Risiken realistischer abgeschätzt werden als zuvor.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

20) Anhand der teilautomatisierten Risikoanalyse kann ein höherer Anteil an Anforderungen eines Anforderungssets auf Änderungsrisiken untersucht werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

21) Auf Grundlage des Änderungsrisikos und der darauf aufbauenden Vorschläge zu Risikosteuerungsmaßnahmen können effizienter Maßnahmen definiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

22) Die Anwendung des Software-Werkzeugs trägt zur Effizienzsteigerung in der Entwicklung komplexer technischer Systeme bei.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

23) Durch den CSV-Export können die Anforderungsdaten aufwandsarm in branchenübliche Anforderungsmanagement-Software importiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

Bewertungsbogen 5

1) Zur Extrahierung von Anforderungen aus Lastenheften ist ein hoher manueller Aufwand notwendig.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

2) Natürlich-sprachliche Anforderungen sind durch Software-Werkzeuge in der Produktentwicklung üblicherweise nicht direkt interpretierbar. Es besteht ein Bedarf nach formalisierten Anforderungen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

3) Anforderungsänderungen werden in Unternehmen noch nicht systematisch gehandhabt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

4) Ich kann mir sehr gut vorstellen, das Software-Werkzeug regelmäßig zu nutzen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

5) Ich empfinde das Software-Werkzeug als unnötig komplex.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

6) Ich denke, dass ich technischen Support brauchen würde, um das Software-Werkzeug zu nutzen.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

7) Ich finde, dass es im Software-Werkzeug zu viele Inkonsistenzen gibt.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

8) Ich kann mir vorstellen, dass die meisten Leute das Software-Werkzeug schnell zu beherrschen lernen.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

9) Ich muss voraussichtlich eine Menge Dinge lernen, bevor ich mit dem Software-Werkzeug arbeiten kann.

<input type="checkbox"/> -2	<input checked="" type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	--	----------------------------	----------------------------	----------------------------

10) Das Werkzeug ist für branchenübliche Lastenhefte anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

11) Das Werkzeug ist generisch für die Entwicklung komplexer technischer Systeme anwendbar.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	---------------------------------------	----------------------------	----------------------------

12) Ich kann mir vorstellen, dass die Anwendung des Werkzeugs in unternehmensinterne Entwicklungsprozesse integriert werden kann.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

13) Die Informationen zur Anwendung des Werkzeugs sind in der Praxis frühzeitig (in der Regel ab der Projektinitierungsphase) in Form von Daten oder Expertenwissen verfügbar:

		ja	nein
a)	Natürlich-sprachliche Anforderungsspezifikation, bzw. Lastenheft (Datengrundlage zur Anwendung der Extrahierung und Klassifizierung von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
b)	Wissen zur Definition von Strukturregeln von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
c)	Wissen zur Definition von Textindikatoren von Schlüssel-Wert-Paaren (Domänenlexikon: Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
d)	Gekennzeichnete Daten zum Training von Named-Entity-Recognition Modellen für Schlüssel-Wert-Paare (KI-Modelle zur Formalisierung von Anforderungen)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
e)	Anforderungsabhängigkeiten (natürlich-sprachliche Anforderungspaare sowie die Abhängigkeit zwischen den Anforderungen) für das initiale Training eines KI-Modells (Abhängigkeitsanalyse von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
f)	Wissen zur Bewertung von Anforderungsabhängigkeiten (Abhängigkeitsanalyse von Anforderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
g)	Anzahl der aufgetretenen Anforderungsänderungen in vergangenen Projekten (Änderungshistorie)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
h)	Wissen zur Bewertung von Auswirkungen von Anforderungsänderungen (Risikomanagement von Anforderungsänderungen)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

14) Den Vorbereitungsaufwand, bevor die Anwendung möglich ist, empfinde ich als angemessen: Erstellen eines Domänenlexikons; Auswahl und Vorbereitung von Anforderungsabhängigkeitsdaten; Zusammenstellung einer Änderungshistorie vergangener Projekte;

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 15) Den Anwendungsaufwand schätze ich als angemessen ein: Import von Lastenheften, Prüfen und Auswahl von Schlüssel-Wert-Paaren, Abhängigkeitsanalyse, Bewertung von Änderungsauswirkungen; Ausgabe der Ergebnisse im CSV-Format

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 16) Durch die automatisierte Extrahierung von Anforderungen aus Lastenheften wird Aufwand reduziert und es wird erleichtert, eine vollständige Anforderungsliste zu generieren.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input checked="" type="checkbox"/> 1	<input type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	---------------------------------------	----------------------------

- 17) Durch die Formalisierung von Anforderungen werden für die Entwicklung von Bauteilen relevante Parameter aus den Anforderungen extrahiert, was zu einer Reduzierung des Aufwands führt.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 18) Auf Grundlage der Anforderungsabhängigkeiten können Auswirkungen von Anforderungsänderungen besser bewertet werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 19) Auf Grundlage der Informationen zum Änderungsrisiko von Anforderungen können Risiken realistischer abgeschätzt werden als zuvor.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 20) Anhand der teilautomatisierten Risikoanalyse kann ein höherer Anteil an Anforderungen eines Anforderungssets auf Änderungsrisiken untersucht werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 21) Auf Grundlage des Änderungsrisikos und der darauf aufbauenden Vorschläge zu Risikosteuerungsmaßnahmen können effizienter Maßnahmen definiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 22) Die Anwendung des Software-Werkzeugs trägt zur Effizienzsteigerung in der Entwicklung komplexer technischer Systeme bei.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

- 23) Durch den CSV-Export können die Anforderungsdaten aufwandsarm in branchenübliche Anforderungsmanagement-Software importiert werden.

<input type="checkbox"/> -2	<input type="checkbox"/> -1	<input type="checkbox"/> 0	<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 2
-----------------------------	-----------------------------	----------------------------	----------------------------	---------------------------------------

Erklärung zur Zitation von Inhalten aus studentischen Arbeiten

In Ergänzung zu meinem Antrag auf Zulassung zur Promotion in der Fakultät für Maschinenbau der Universität Paderborn erkläre ich gemäß §11 der Promotionsordnung und unter Beachtung der Regelung zur Zitation studentischer Arbeiten:

Die von mir vorgelegte Dissertation habe ich selbstständig verfasst, und ich habe keine anderen als die dort angegebenen Quellen und Hilfsmittel benutzt. Es sind Inhalte studentischen Ursprungs (studentische Arbeiten) in dieser Dissertation enthalten.

Ich habe die verwendeten Arbeiten entsprechend der Regelung „Zitation aus studentischen Arbeiten in Dissertationen“ zitiert.

Paderborn, 30.05.2024

Das Heinz Nixdorf Institut – Interdisziplinäres Forschungszentrum für Informatik und Technik

Das Heinz Nixdorf Institut ist ein Forschungszentrum der Universität Paderborn. Es entstand 1987 aus der Initiative und mit Förderung von Heinz Nixdorf. Damit wollte er Ingenieurwissenschaften und Informatik zusammenführen, um wesentliche Impulse für neue Produkte und Dienstleistungen zu erzeugen. Dies schließt auch die Wechselwirkungen mit dem gesellschaftlichen Umfeld ein.

Die Forschungsarbeit orientiert sich an dem Programm „Dynamik, Vernetzung, Autonomie: Neue Methoden und Technologien für die intelligenten technischen Systeme von morgen“. In der Lehre engagiert sich das Heinz Nixdorf Institut in Studiengängen der Informatik, der Ingenieurwissenschaften und der Wirtschaftswissenschaften.

Heute wirken am Heinz Nixdorf Institut acht Professoren/in mit insgesamt 120 Mitarbeiterinnen und Mitarbeitern. Pro Jahr promovieren hier etwa 15 Nachwuchswissenschaftlerinnen und Nachwuchswissenschaftler.

Heinz Nixdorf Institute – Interdisciplinary Research Centre for Computer Science and Technology

The Heinz Nixdorf Institute is a research centre within the University of Paderborn. It was founded in 1987 initiated and supported by Heinz Nixdorf. By doing so he wanted to create a symbiosis of computer science and engineering in order to provide critical impetus for new products and services. This includes interactions with the social environment.

Our research is aligned with the program “Dynamics, Networking, Autonomy: New methods and technologies for intelligent technical systems of tomorrow”. In training and education the Heinz Nixdorf Institute is involved in many programs of study at the University of Paderborn. The superior goal in education and training is to communicate competencies that are critical in tomorrow's economy.

Today eight Professors and 120 researchers work at the Heinz Nixdorf Institute. Per year approximately 15 young researchers receive a doctorate.

Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

- Bd. 402 Wu, L.: Ultrabreitbandige Sampler in SiGe-BiCMOS-Technologie für Analog-Digital-Wandler mit zeitversetzter Abtastung. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 402, Paderborn, 2021 – ISBN 978-3-947647-21-7
- Bd. 403 HILLEBRAND, M.: Entwicklungssystematik zur Integration von Eigenschaften der Selbstheilung in Intelligente Technische Systeme. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 403, Paderborn, 2021 – ISBN 978-3-947647-22-4
- Bd. 404 OLMA, S.: Systemtheorie von Hardware-in-the-Loop-Simulationen mit Anwendung auf einem Fahrzeugachsprüfstand mit parallelkinematischem Lastsimulator. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 404, Paderborn, 2022 – ISBN 978-3-947647-23-1
- Bd. 405 FECHTELPETER, C.: Rahmenwerk zur Gestaltung des Technologietransfers in mittelständisch geprägten Innovationsclustern. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 405, Paderborn, 2022 – ISBN 978-3-947647-24-8
- Bd. 406 OLEFF, C.: Proaktives Management von Anforderungsänderungen in der Entwicklung komplexer technischer Systeme. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 406, Paderborn, 2022 – ISBN 978-3-947647-25-5
- Bd. 407 JAVED, A. R.: Mixed-Signal Baseband Circuit Design for High Data Rate Wireless Communication in Bulk CMOS and SiGe BiCMOS Technologies. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 407, Paderborn, 2022 – ISBN 978-3-947647-26-2
- Bd. 408 DUMITRESCU, R., KOLDEWEY, C.: Daten-gestützte Projektplanung. Fachbuch. Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 408, Paderborn, 2022 – ISBN 978-3-947647-27-9
- Bd. 409 PÖHLER, A.: Automatisierte dezentrale Produktionssteuerung für cyber-physische Produktionssysteme mit digitaler Repräsentation der Beschäftigten. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 409, Paderborn, 2022 – ISBN 978-3-947647-28-6
- Bd. 410 RÜDDENKLAU, N.: Hardware-in-the-Loop-Simulation von HD-Scheinwerfer-Steuergeräten zur Entwicklung von Lichtfunktionen in virtuellen Nachtfahrten. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 410, Paderborn, 2023 – ISBN 978-3-947647-29-3
- Bd. 411 BIEMELT, P.: Entwurf und Analyse modell-prädiktiver Regelungsansätze zur Steigerung des Immersionsempfindens in interaktiven Fahrsimulationen. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 411, Paderborn, 2023 – ISBN 978-3-947647-30-9
- Bd. 412 HAAKE, C.-J., MEYER AUF DER HEIDE, F., PLATZNER, M., WACHSMUTH, H., WEHRHEIM, H. (Eds.): On-The-Fly Computing – Individualized IT-Services in dynamic markets, Collaborative Research Centre 901 (2011 - 2023), Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 412, Paderborn, 2023 – ISBN 978-3-947647-31-6
- Bd. 413 DUMITRESCU, R.; HÖLZLE, K. (Hrsg.): Vorausschau und Technologieplanung. 17. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 14./15. September 2023, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 413, Paderborn, 2023 – ISBN 978-3-947647-32-3

Zuletzt erschienene Bände der Verlagsschriftenreihe des Heinz Nixdorf Instituts

- Bd. 414 ABUGHANNAM, S.: Low-power Direct-detection Wake-up Receiver at 2.44 GHz for Wireless Sensor Networks. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 414, Paderborn, 2024 – ISBN 978-3-947647-33-0
- Bd. 415 REINHOLD, J.: Systematik zur musterbasierten Transformation von Wertschöpfungssystemen für Smart Services. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 415, Paderborn, 2024 – ISBN 978-3-947647-34-7
- Bd. 416 YANG, X.: Eine Methode zur Unterstützung von Entscheidungen bei der Entwicklung modularer Leichtbauprodukte. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 416, Paderborn, 2024 – ISBN 978-3-947647-35-4
- Bd. 417 GRÄLER, M.: Entwicklung adaptiver Einrichtungsinstanzsysteme für Produktionsprozesse. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 417, Paderborn, 2024 – ISBN 978-3-947647-36-1
- Bd. 418 RÖSSMANN, D.: Menschenzentrierte Montageplanung und -steuerung durch fähigkeitsorientierte Aufgabenzuordnung. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 418, Paderborn, 2024 – ISBN 978-3-947647-37-8
- Bd. 419 BAHMANIAN, M.: Optoelectronic Phase-Locked Loop, Theory and Implementation. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 419, Paderborn, 2024 – ISBN 978-3-947647-38-5
- Bd. 420 HEIHOFF-SCHWEDE, J.: Spezifikationstechnik zur Analyse, Gestaltung und Bewertung von Engineering-IT-Architekturen. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 420, Paderborn, 2024 – ISBN 978-3-947647-39-2
- Bd. 421 MEYER, M.: Systematik zur Planung und Verwertung von Betriebsdaten-Analysen in der strategischen Produktplanung. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 421, Paderborn, 2024 – ISBN 978-3-947647-40-8
- Bd. 422 MALENA, K.: Konzipierung, Analyse und Realumsetzung eines mehrstufigen modellpraktiven Lichtsignalanlagenregelungssystems. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 422, Paderborn, 2024 – ISBN 978-3-947647-41-5
- Bd. 423 GÖTTE, R.-S.: Online-Schätzung von Modellgenauigkeiten zur automatischen Modelladaption unter Beibehaltung einer physikalisch-technischen Interpretierbarkeit. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 423, Paderborn, 2024 – ISBN 978-3-947647-42-2
- Bd. 424 PIERENKEMPER, C.: Systematik zur Entwicklung Leistungsstufen-basierter Industrie 4.0-Strategien. Dissertation, Fakultät für Elektrotechnik, Informatik und Mathematik, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 424, Paderborn, 2024 – ISBN 978-3-947647-43-9
- Bd. 425 DUMITRESCU, R.; HÖLZLE, K. (Hrsg.): Vorausschau und Technologieplanung. 18. Symposium für Vorausschau und Technologieplanung, Heinz Nixdorf Institut, 5./6. Dezember 2024, Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 425, Paderborn, 2024 – ISBN 978-3-947647-44-6
- Bd. 426 HESSE, M.: Interaktive Inbetriebnahme von Steuerungen und Regelungen für partiell bekannte dynamische Systeme mittels Gauß-Prozess-Regression. Dissertation, Fakultät für Maschinenbau, Universität Paderborn, Verlagsschriftenreihe des Heinz Nixdorf Instituts, Band 426, Paderborn, 2024 – ISBN 978-3-947647-45-3

Bezugsadresse:

Heinz Nixdorf Institut
Universität Paderborn
Fürstenallee 11
33102 Paderborn

Unvollständige Anforderungslisten, mehrdeutige Anforderungen und Anforderungsänderungen zählen zu den Hauptgründen für das Scheitern von Entwicklungsprojekten. Die manuelle Analyse von natürlich-sprachlichen Anforderungsdokumenten, um die Vollständigkeit der Anforderungen zu erhöhen, ist zeitaufwändig, fehleranfällig und teuer. Daher besteht ein Bedarf an automatisierten Methoden zur Extrahierung und Formalisierung von Anforderungen, um Aufwände und Kosten zu reduzieren. Natürliche Sprache ist mehrdeutig und schwer direkt von Software-Werkzeugen interpretierbar. Anforderungen müssen formalisiert werden, um in der Produktentwicklung genutzt werden zu können. Zur Handhabung von Anforderungsänderungen wird ein frühzeitiges Risikomanagement benötigt. Die Methodik bietet Lösungen zur Extraktion, Formalisierung und Handhabung von Änderungen. KI-basierte Verfahren ermöglichen vollständige und eindeutige Anforderungslisten und erleichtern die Erkennung von Anforderungsabhängigkeiten. Zudem wird die Berechnung von Änderungsrisiken in der industriellen Praxis vereinfacht.