

Sprachliche Kontrolle von Anforderungsdokumenten

Jennifer Krisch

Kontrolle von

Anforderungsdokumenten

Sprachliche

von Anforderungsdok

die Kontrolle

Sprachliche Kontrolle

Impressum

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

Diese elektronische Publikation ist mit der Creative-Commons-Nutzungslizenz BY-NC-ND (Namensnennung – Nicht kommerziell – Keine Bearbeitung) versehen.

Weitere Informationen unter: <http://creativecommons.org/licenses/by-nc-nd/3.0/de>

Universitätsverlag Hildesheim
Universitätsplatz 1
31141 Hildesheim

<https://www.uni-hildesheim.de/bibliothek/forschen-publizieren/universitaetsverlag/>

Erstausgabe Hildesheim 2017
Redaktion und Satz: Jennifer Krisch

Die vorliegende Arbeit ist kostenfrei im Internet abrufbar unter: <http://dx.doi.org/10.18442/737>

Zitierempfehlung:

Krisch, Jennifer (2017). *Sprachliche Kontrolle von Anforderungsdokumenten*.

Hildesheim: Universitätsverlag Hildesheim. E-Publikation (Open Access) <http://dx.doi.org/10.18442/737>

Sprachliche Kontrolle von Anforderungsdokumenten

Vom Fachbereich 3 (Sprach- und Informationswissenschaften)
der Universität Hildesheim zur Erlangung des Grades einer

**Doktorin der Philosophie
(Dr. phil.)**

angenommene Dissertation von
Jennifer Krisch
geboren am 08. Januar 1988 in Nürtingen

Erstgutachter: Prof. Dr. phil. habil. Ulrich Heid
Universität Hildesheim, FB 3

Zweitgutachter: Prof. Dr. rer. nat. Klaus Schmid
Universität Hildesheim, FB 4

Nebenbericht: Dr. rer. nat. Frank Houdek
Daimler AG, RD/FIT, Böblingen

Tag der mündlichen Prüfung: 18. Mai 2017

Danksagung

Eine Dissertation zu schreiben fordert vieles von einem selbst, aber auch von den Menschen, die einem zur Seite stehen. Diesen Menschen möchte ich gerne danken.

Großer Dank gebührt meinem Doktorvater, Herrn Prof. Dr. Ulrich Heid. Schon während meiner Studienzeit konnte ich stets auf seine Unterstützung setzen. Vielen Dank für die umfassende Betreuung, die hilfreichen Ratschläge, den fachlichen Austausch und die Zeit, die Sie für mich investiert haben. Ohne Sie wäre das alles nicht möglich gewesen.

Prof. Dr. Klaus Schmid danke ich herzlich für die Übernahme des Zweitgutachtens, die Betreuung und die hilfreiche Unterstützung. Ein großes Dankeschön richte ich auch an Dr. Frank Houdek und Dr. Matthias Kirchmayr. Danke Frank und Matthias für die Möglichkeit meine Dissertation bei der Daimler AG zu verfassen. Vielen Dank auch für den Glauben an mich und die investierte Zeit. Ihr habt mir etwas ermöglicht, von dem ich nicht mal zu träumen wagte. Danke auch an Stefanie Räther für die Unterstützung bei der technischen Umsetzung meines Projekts.

Ein großes Dankeschön möchte ich an Eva und Tabea aussprechen. Vielen Dank für eure lieben Worte und eure tatkräftige Unterstützung. Ihr habt mich stets motiviert weiterzumachen, nicht aufzugeben und an mich zu glauben. Danke, dass ihr immer für mich da seid. Meinen lieben Freunden Lisa, Christiane, Nadine, Stephanie und Anna möchte ich ebenfalls für ihre Unterstützung danken.

Auch meinen Eltern Andreas und Claudia sowie meinem Bruder Sven und meiner lieben Oma Rotraut möchte ich danken. Ihr seid mein Rückhalt. Danke, dass ich immer auf euch zählen kann und ihr mich bei allem unterstützt.

Doch den allergrößten Dank möchte ich an meinen Partner aussprechen. Dominic, du hast mir immer den Rücken freigehalten, mir stets Mut zugesprochen und auch in den schwierigsten Situationen zu mir gehalten. Ich danke dir von ganzem Herzen und möchte dir sagen, dass ich jederzeit das Gleiche auch für dich tun würde.

Kurzfassung

In der Softwareentwicklung beschreiben Anforderungen, wie sich ein System oder eine Komponente in bestimmten Situationen verhalten soll. Anforderungen stellen also Vorgaben an die Software dar, die entwickelt werden soll. Die natürliche Sprache ist noch immer das vorherrschende Mittel, um Anforderungen zu formulieren. Aus dem Anforderungstext werden Testfälle abgeleitet, mit welchen überprüft wird, ob die in den Anforderungen beschriebenen Funktionen korrekt umgesetzt wurden. Da die Aufwendungen für die Testaktivitäten einen großen Teil der Entwicklungszeit und Entwicklungskosten einnehmen, besteht Bedarf an der sprachlichen Kontrolle von Anforderungstexten. Ist eine Anforderung so formuliert, dass sie beispielsweise aufgrund fehlender Information zu Interpretationsspielraum führt, kann dies dazu führen, dass ein falscher Testfall abgeleitet wird oder eine Funktion falsch entwickelt wird.

Ein Ziel dieser Arbeit ist, zu untersuchen, welche Kriterien eine Anforderung aus sprachwissenschaftlicher Sicht erfüllen muss, um als testbar zu gelten bzw. welche sprachlichen Phänomene dazu führen, dass eine Anforderung nicht testbar ist. Hierfür werden Leitlinien aus der technischen Dokumentation sowie Literatur aus dem Bereich des Requirements Engineering analysiert. Die identifizierten Phänomene werden in einem Qualitätsmodell für testbare Anforderungen zusammengetragen, nach linguistischen Kriterien sortiert und nach Kritikalität bewertet.

Für ausgewählte Phänomene aus dem Qualitätsmodell wird untersucht, inwiefern sich automatisierte Prüfmethode entwickeln lassen, die kritische Instanzen der Phänomene zuverlässig identifizieren, also diejenigen, die Interpretationsspielraum verursachen können. Für die Analyse der Phänomene werden computerlinguistische Analysemethoden eingesetzt und Regeln entwickelt, die auf (morpho-)syntaktische Annotationen zugreifen.

Sprachliche Phänomene, die einen Einfluss auf die Testbarkeit einer Anforderung haben, lassen sich durch die verwendeten computerlinguistischen Analysemethoden automatisiert identifizieren. Für bestimmte Phänomene können zudem Regeln entworfen werden, welche zuverlässig zwischen den kritischen und unkritischen Instanzen eines Phänomens unterscheiden. Die formbasierte Computerlinguistik stößt allerdings an ihre Grenzen, wenn diese Unterscheidung auf Welt- und Expertenwissen beruht.

Die entwickelten Analysemethoden heben sich von existierenden Werkzeugen zur Überprüfung von Texten dadurch ab, dass nicht nur reine Wortlistenabgleiche durchgeführt werden, sondern auch der Satzkontext, in welchem die Phänomene auftreten, in die Analyse miteinbezogen wird. Dieser Ansatz kann die Anzahl der Fehlwarnungen an den Benutzer erheblich reduzieren, sodass der Aufwand bei der Durchsicht der potentiell kritischen Instanzen gesenkt werden kann.

Abstract

In software development, requirements describe how a system or a component is supposed to act in specific situations. Therefore, requirements provide guidelines for the software which is to be developed. Natural language is still the predominant technique used to formulate requirements. Test cases are derived from these requirements to check for correct implementation of the functions described in the requirements. Since the expenses for the test activities make up a huge part of the development time and its costs, there is need for linguistic control of requirements specifications. If a requirement is formulated in a misleading way causing various possibilities of interpretation due to missing information, it can lead to a wrong test case or in developing an incorrect function.

One objective of this dissertation is to investigate which criteria must be met by a requirement from a linguistical point of view for the requirement to be considered testable; respectively which linguistic phenomena lead to untestable requirements. For this purpose, guidelines from the field of technical documentation as well as literature concerning requirements engineering are analyzed. The identified phenomena are gathered in a quality model for testable requirements, arranged from a linguistical point of view and evaluated according to criticality.

For selected phenomena which are integrated in the quality model it is investigated to what extent automated checks which identify the critical instances reliably can be developed; that means those that leave room for interpretation. Computational linguistics analysis methods and rules which access morphosyntactic annotations are applied.

The developed computational linguistics analysis method makes it possible to automatically identify phenomena which have an influence on the testability of a requirement. Furthermore, for specific phenomena rules can be designed which differentiate between critical and uncritical instances. If this decision relies on world and expert knowledge, though, form-based computational linguistics reaches its limit.

The developed analysis methods stand out from existing tools for revising texts insofar as not only pure word lists comparisons are executed, but also the context in which the phenomena occur is included in the analysis. This approach can significantly reduce the number of false warnings to the user reducing the effort involved in viewing the potentially critical instances.

Inhaltsverzeichnis

Abbildungsverzeichnis		X
Tabellenverzeichnis		XII
1 Einleitung		1
1.1 Motivation		1
1.2 Verortung in der Wissenschaft		7
1.3 Beiträge der Arbeit		9
1.4 Rahmen der Arbeit		10
1.5 Struktur und Aufbau der Arbeit		11
2 Textqualität aus der Sicht der Softwareentwicklung		15
2.1 Systematische Softwareentwicklung		15
2.2 Requirements Engineering		18
2.2.1 Arten von Anforderungen		20
2.2.2 Dokumentation von Anforderungen		21
2.3 Testen in der Softwareentwicklung		23
2.3.1 Testbasisanalyse		24
2.3.2 Qualitätskriterien für Anforderungen und Anforderungsdokumente		25
2.4 Zusammenfassung		28
3 Textqualität aus der Sicht der angewandten Sprachwissenschaft		29
3.1 Verständlichkeitsforschung		29
3.1.1 Verständlichkeitsformeln		29
3.1.2 Verständlichkeitsmodelle		31
3.1.2.1 Das Hamburger Verständlichkeitsmodell		31
3.1.2.2 Das Verständlichkeitsmodell nach Groeben		33
3.2 Leitlinien für die Erstellung technischer Dokumentation		35
3.3 Zusammenfassung		36
4 Entwicklung eines Qualitätsmodells für testbare Anforderungen		38
4.1 Testbarkeitskriterium Messbarkeit		39
4.2 Testbarkeitskriterium Verständlichkeit		40
4.2.1 Fehlende Information		41
4.2.1.1 Nominalisierung		41
4.2.1.2 Weak-Word		42
4.2.1.3 Unpersönliche Konstruktion		44
4.2.1.4 Unvollständiger Satz		45
4.2.1.5 Unvollständige Bedingung		45
4.2.1.6 Passiv und Passiversatzkonstruktionen ohne Agens		46
4.2.2 Verbindlichkeit		47
4.2.2.1 Mangelnde Trennung zwischen Anforderung und Information		47
4.2.2.2 Modalverb <i>sollte(n)</i>		47
4.2.2.3 Verallgemeinerung/Universalquantor		48

4.2.3	Mehrdeutigkeit	49
4.2.3.1	Lexikalische Mehrdeutigkeit	49
4.2.3.1.1	Polysemie und Homonymie	49
4.2.3.1.2	Langes Kompositum	50
4.2.3.2	Satzstrukturelle Mehrdeutigkeit	51
4.2.3.2.1	Gemischte Koordinationen	51
4.2.3.2.2	Attribut vs. Adverbial	52
4.2.3.2.3	Koordination + Attribut	52
4.2.3.2.4	Präpositional- und Nominalphrasen	53
4.2.3.3	Satzfunktionale Mehrdeutigkeit	54
4.2.3.3.1	Subjekt vs. Objekt	54
4.2.3.3.2	Genitivus subjectivus vs. objectivus	54
4.2.3.4	Satzsequentielle Mehrdeutigkeit	55
4.2.3.4.1	Pronominaler Bezug	55
4.2.4	Sprachkomplexität	56
4.2.4.1	Verschachtelte Bedingungen	56
4.2.4.2	Verschachtelter/Zu langer Satz	57
4.2.4.3	Langer Klammereinschub	58
4.2.4.4	Aufzählung ohne Liste	58
4.2.4.5	Doppelte Negation	59
4.2.4.6	Pränominales Partizip	60
4.2.4.7	Komplexe Präposition	61
4.2.4.8	Passiv mit Agensangabe	61
4.3	Testbarkeitskriterium Atomarität	62
4.4	Testbarkeitskriterium Widerspruchsfreiheit	63
4.5	Testbarkeitskriterium Inhaltliche Korrektheit	64
4.6	Testbarkeitskriterium Durchführbarkeit	64
4.7	Interpretation des Qualitätsmodells	65
4.8	Vorgehen bei der Auswahl relevanter Phänomene für die geplanten linguistischen Analysen und Regelformulierungen	67
4.9	Zusammenfassung	71
5	Methoden und Werkzeuge zur Sicherstellung der Qualität in Anforderungen	73
5.1	Schreibregeln	73
5.2	Satzschablonen	74
5.3	Kontrollierte Sprachen	79
5.4	Manuelle Review-Techniken	82
5.5	Werkzeuge für die Überprüfung natürlichsprachlicher Texte	84
5.5.1	Werkzeuge aus dem Bereich des Requirements Engineering	84
5.5.1.1	DESIRE®	85
5.5.1.2	ReqCheck	86
5.5.1.3	Requirements Quality Suite	86
5.5.1.4	Requirements Scout	87
5.5.2	Werkzeuge aus dem Bereich der Sprachtechnologie	88
5.5.2.1	LanguageTool	88

5.5.2.2	TextLab	89
5.5.2.3	CLAT	90
5.5.2.4	Acrolinx	91
5.5.3	Testsatzsammlung für die Evaluierung bestehender Textprüfwerkzeuge	91
5.5.3.1	Vorstellung und Aufbau der Testsatzsammlung	92
5.5.3.2	Ergebnisse der Evaluierung anhand der Testsatzsammlung	93
5.5.4	Zusammenfassender Überblick über Lastenheft-Prüfwerkzeuge	98
5.6	Weitere automatisierte Ansätze	99
5.7	Zusammenfassung	102
6	Erstellung und Annotation von Lastenheft-Korpora	104
6.1	Struktur von Lastenheften in der Datenbank <i>DOORS</i>	104
6.2	Export von Lastenheften aus der Datenbank <i>DOORS</i>	105
6.3	Computerlinguistische Annotation der Lastenhefttexte	106
6.3.1	Tokenisierung	107
6.3.2	Lemmatisierung	108
6.3.3	Part-of-Speech-Tagging	108
6.3.4	Morphologie-Tagging	109
6.3.5	Parsing	109
6.3.6	Qualitätsbetrachtung der Annotationswerkzeuge	110
6.4	Zahlen, Daten und Fakten über die erzeugten Korpora	113
6.5	Transformation der Korpora in ein CQP-lesbares Format	113
6.6	Zusammenfassung	114
7	Allgemeines Vorgehen für die Analyse ausgewählter Phänomene	115
7.1	Regelfindung und Regeldefinition	117
7.2	Evaluierung	119
7.3	Zusammenfassung	124
8	Phänomenanalysen	125
8.1	Weak-Words	125
8.1.1	Phänomenbeschreibung	126
8.1.2	Regeln	129
8.1.3	Evaluierung der Regeln	133
8.1.4	Zusammenfassung	140
8.2	Unpersönliche Konstruktionen	140
8.2.1	Phänomenbeschreibung	141
8.2.2	Regeln	143
8.2.3	Zusammenfassung	145
8.3	Passiv und Passiversatzkonstruktionen ohne Agens	146
8.3.1	Phänomenbeschreibung	146
8.3.2	Regeln	149
8.3.3	Auswertung ausgewählter Passivsätze ohne Agens nach Kritikalität	151
8.3.4	Zusammenfassung	152
8.4	Das Modalverb <i>sollte(n)</i>	152

8.4.1	Phänomenbeschreibung	153
8.4.2	Regeln	156
8.4.3	Evaluierung der Regeln	158
8.4.4	Zusammenfassung	162
8.5	Pronominale Bezüge	163
8.5.1	Phänomenbeschreibung	163
8.5.2	Regeln	166
8.5.3	Auswertung der untersuchten pronominalen Bezüge nach Kritikalität	168
8.5.4	Zusammenfassung	170
8.6	Phänomenanalysen für englischsprachige Lastenhefte	171
8.6.1	Weak-Words	172
8.6.2	Passiv ohne Agens	176
8.6.3	Das Modalverb <i>should</i>	177
8.6.4	Zusammenfassung der Ergebnisse der englischen Analysen	181
8.7	Computerlinguistische Bewertung	182
8.8	Zusammenfassung	184
9	Technische Umsetzung und externe Bewertung der entwickelten Analysen	186
9.1	Technische Umsetzung	186
9.2	Externe Bewertung	189
9.3	Zusammenfassung	192
10	Interpretation der empirischen Ergebnisse	194
10.1	Interne Validität	194
10.1.1	Studie 1: Fleiss' Kappa	195
10.1.2	Studie 2: Fragebogen	197
10.2	Externe Validität	199
10.3	Zusammenfassung	199
11	Zusammenfassung und Ausblick	201
11.1	Zusammenfassung der Beiträge und Ergebnisse	201
11.2	Ausblick	204
Anhang		207
Literaturverzeichnis		228

Abbildungsverzeichnis

Abb. 1	Softwareentwicklungsprozess	1
Abb. 2	Techniken und Notationen für die Anforderungsdokumentation laut [AWK13]	4
Abb. 3	Größte Schwächen in Anforderungsdokumenten laut [ASW15]	5
Abb. 4	Ergebnisse einer Befragung von 20 Testexperten zur sprachlichen Qualität von Anforderungen im Automobilkontext	6
Abb. 5	Struktur und Aufbau der Arbeit: Beiträge 1 und 2	11
Abb. 6	Struktur und Aufbau der Arbeit: Beiträge 1-3	12
Abb. 7	Struktur und Aufbau der Arbeit: Beiträge 1-4	14
Abb. 8	Wasserfallmodell nach [Par10]	16
Abb. 9	Allgemeines V-Modell nach [SL12]	17
Abb. 10	Skala des Verständlichkeitsmerkmals Kürze-Prägnanz nach [LST74]	31
Abb. 11	Beurteilungsfenster der Verständlichkeitsmerkmale nach [LST74]	32
Abb. 12	Überblick Qualitätsmodell für testbare Anforderungen	38
Abb. 13	Qualitätskriterium Messbarkeit	39
Abb. 14	Qualitätskriterium Verständlichkeit	41
Abb. 15	Qualitätskriterium Atomarität	62
Abb. 16	Qualitätskriterium Widerspruchsfreiheit	63
Abb. 17	Qualitätskriterium Inhaltliche Korrektheit	64
Abb. 18	Qualitätskriterium Durchführbarkeit	65
Abb. 19	Relevante Phänomene des Testbarkeitskriteriums Verständlichkeit	71
Abb. 20	Satzschablone ohne Bedingung nach [Rup14]	75
Abb. 21	Ausgefüllte Satzschablone ohne Bedingung nach [Rup14]	75
Abb. 22	Satzschablone mit Bedingung nach [Rup14]	76
Abb. 23	Ausschnitt aus der erstellten Testsatzsammlung	93
Abb. 24	Ausschnitt eines Lastenheftes in der Datenbank <i>DOORS</i>	105
Abb. 25	Anforderungen nach dem Export	106
Abb. 26	Tokenisierte Anforderung im CoNLL-2009-Format	108
Abb. 27	Vollständig annotierte Anforderung	109
Abb. 28	Dependenzgraph	110
Abb. 29	Arten von Fehlertypen für die Evaluierung der Annotationsergebnisse	111
Abb. 30	Vorgehen bei der Entwicklung und Umsetzung von Regeln: Regelentwicklung (grün) und Zielprozess (blau)	116
Abb. 31	Ergebnis in CQP zu Query [lemma="Motor"];	118
Abb. 32	Beispiel einer Evaluierung, N=100	121
Abb. 33	Einteilung <i>tp</i> , <i>fp</i> , <i>fn</i> und <i>tn</i> in der Klasse <i>Kritische Anforderung</i>	122
Abb. 34	Einteilung <i>tp</i> , <i>fp</i> , <i>fn</i> und <i>tn</i> in der Klasse <i>Unkritische Anforderung</i>	122
Abb. 35	Evaluierung der Weak-Word-Regeln am Entwicklungskorpus, N=541	134
Abb. 36	Evaluierung der Weak-Word-Regeln am Evaluierungskorpus, N=582	136
Abb. 37	Verschiebung der syntaktischen Funktionen in Aktiv- und Passivsätzen	147
Abb. 38	<i>sollte</i> bildet das Hauptverb des Satzes	157
Abb. 39	<i>sollte</i> ist Regens einer subordinierenden Konjunktion	158
Abb. 40	<i>sollte</i> ist Konjunkt einer Koordination von Verbformen	159
Abb. 41	Evaluierung der Regeln für <i>sollte</i> am Entwicklungskorpus, N=359	159

Abb. 42	Evaluierung der Regeln für <i>sollte</i> am Evaluierungskorpus, N=328 . . .	160
Abb. 43	Evaluierung der Regeln des Weak-Words <i>large</i> am Entwicklungskorpus, N=119	174
Abb. 44	Evaluierung der Regeln des Weak-Words <i>large</i> am Evaluierungskorpus, N=124	175
Abb. 45	Evaluierung der Regeln für <i>should</i> am Entwicklungskorpus, N=620 . .	179
Abb. 46	Evaluierung der Regeln für <i>should</i> am Evaluierungskorpus, N=556 . .	180
Abb. 47	Verarbeitungsschritte im Lastenheft-Analyseassistent	187
Abb. 48	Bewertung von Anforderungen im <i>Lastenheft-Analyseassistenten</i> . . .	188
Abb. 49	Fragebogen: Auswertung der Rückmeldungen zu 17 Anforderungen . .	198

Tabellenverzeichnis

Tab. 1	Herausforderungen in Projekten nach [The]	2
Tab. 2	Gründe für das Scheitern von Projekten nach [The]	3
Tab. 3	Klassifikation von Notationen nach Formalisierungsgraden nach [Mar03]	22
Tab. 4	<i>Reading-Ease-Scores</i> nach [Fle48]	30
Tab. 5	Zuordnung der Phänomene aus dem Qualitätsmodell zu den untersuchten Quellen (RE1: Rupp 2014 [Rup14], RE2: Pohl/Rupp [PR15], RE3: Pohl [Poh07], RE4: Kamsties [Kam01], DAI1= Daimler-interne Schulung [Dai09a], DAI2: Daimler-interne Leitlinie [Dai09b], TD1: tekom [tek11], TD2: Duden [Sch12])	66
Tab. 6	Einstufung der Phänomene aus dem Qualitätsmodell	69
Tab. 7	Überblick, welche Phänomene bei der Verwendung von Satzschablonen aus [Rup14] auftreten können (↑ = Phänomen kann auftreten; ↓ = Phänomen kann nicht auftreten; (R) = Phänomen wird durch Regelwerk ausgeschlossen; (G) = Phänomen wird durch Verwendung eines Glossars ausgeschlossen)	77
Tab. 8	Beispiel für einen Eintrag in einer Wortliste in DESIRE [®] [SSUE09] . .	85
Tab. 9	Vorverarbeitungsschritte der präsentierten Werkzeuge (grün (↑) = ja, gelb (→) = durch Listenabgleich, rot (↓) = nein)	94
Tab. 10	Evaluierungsergebnisse der Analysen bestehender Textprüfwerkzeuge (grün (↑) = ja, gelb (→) = teilweise, rot (↓) = nein)	95
Tab. 11	Evaluierungsergebnisse der Analysen bestehender Textprüfwerkzeuge an ausgewählten Phänomenen aus dem Qualitätsmodell (grün (↑) = ja, gelb (→) = teilweise, rot (↓) = nein)	97
Tab. 12	Evaluierungsergebnisse <i>mate</i> und <i>MarMot</i> (N=1837)	112
Tab. 13	Fehlerquoten der Annotationen (N=1837)	112
Tab. 14	Korpusinformationen zu den Einzelkorpora	113
Tab. 15	Korpusinformationen zu den Gesamtkorpora	113
Tab. 16	Variablen in CQP	118
Tab. 17	Beispielergebnis Precision-, Recall- und F-Measure, N=100	121
Tab. 18	Einteilung der untersuchten Weak-Words in die Klassen der Adjektive und Adverbien	127
Tab. 19	Regeln für die Weak-Words <i>lang</i> , <i>kurz</i> , <i>schnell</i> , <i>breit</i> , <i>hoch</i> und <i>niedrig</i>	130
Tab. 20	Regeln für die Weak-Words <i>groß</i> und <i>klein</i>	131
Tab. 21	Regeln für die Weak-Words <i>maximal</i> und <i>minimal</i>	132
Tab. 22	Regeln für die Weak-Words <i>eigentlich</i> und <i>tatsächlich</i>	132
Tab. 23	Regeln für das Weak-Word <i>mal</i>	132
Tab. 24	Auswertung der Analyse am Entwicklungskorpus, N=541	133
Tab. 25	Precision-, Recall- und F-Measure-Werte der Weak-Word-Regeln am Entwicklungskorpus, N=541	134
Tab. 26	Auswertung der Analyse am Evaluierungskorpus, N=582	135
Tab. 27	Precision-, Recall- und F-Measure-Werte der Weak-Word-Regeln am Evaluierungskorpus, N=582	136
Tab. 28	Einteilung der untersuchten 400 Instanzen von <i>es</i> in die vier definierten Gebrauchsweisen	144

Tab. 29	Gruppen von Passiv und Passiversatz für die Regelentwicklung	149
Tab. 30	Regeln für die Identifikation von Vorgangspassiv ohne Agens	150
Tab. 31	CQP-Regeln für die Identifikation kritischer Verwendungen von <i>sollte(n)</i>	156
Tab. 32	Precision-, Recall- und F-Measure-Werte der Regeln für <i>sollte</i> am Ent- wicklungskorpus, N=359	160
Tab. 33	Precision-, Recall- und F-Measure-Werte der Regeln für <i>sollte</i> am Eva- luierungskorpus, N=328	160
Tab. 34	Nicht behandelte Klassen von Pronomina	167
Tab. 35	Behandelte Klassen von Pronomina	168
Tab. 36	CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger pronominaler Bezüge	169
Tab. 37	Auswertung der pronominalen Bezüge über alle Korpora	169
Tab. 38	Anteil der semantisch mehrdeutigen pronominalen Bezüge über alle Korpora	170
Tab. 39	Korpusinformationen zu den englischen Korpora	172
Tab. 40	Regeln für das Weak-Word <i>large</i>	173
Tab. 41	Precision-, Recall- und F-Measure-Werte des Weak-Words <i>large</i> am Entwicklungskorpus, N=119	174
Tab. 42	Precision-, Recall- und F-Measure-Werte des Weak-Words <i>large</i> am Evaluierungskorpus, N=124	175
Tab. 43	CQP-Regeln für die Identifikation unkritischer Verwendungen von <i>should</i>	178
Tab. 44	Precision-, Recall- und F-Measure-Werte der Regeln für <i>should</i> am Entwicklungskorpus, N=620	179
Tab. 45	Precision-, Recall- und F-Measure-Werte der Regeln für <i>should</i> am Evaluierungskorpus, N=556	180
Tab. 46	Evaluierungsergebnisse der Analysen bestehender Textprüfwerkzeuge an ausgewählten Phänomenen aus dem Qualitätsmodell (grün (↑) = ja, gelb (→) = teilweise, rot (↓) = nein)	189
Tab. 47	CQP-Regeln für die Weak-Words <i>lang, kurz, schnell, breit, hoch</i> und <i>niedrig</i>	212
Tab. 48	CQP-Regeln für die Weak-Words <i>groß</i> und <i>klein</i>	213
Tab. 49	CQP-Regeln für die Weak-Words <i>maximal</i> und <i>minimal</i>	214
Tab. 50	CQP-Regeln für die Weak-Words <i>eigentlich</i> und <i>tatsächlich</i>	214
Tab. 51	CQP-Regeln für das Weak-Word <i>mal</i>	214
Tab. 52	Regeln für die Identifikation von Vorgangspassiv ohne Agens	215
Tab. 53	Regeln für die Identifikation von Zustandspassiv ohne Agens	216
Tab. 54	Regeln für die Identifikation von Passiversatzkonstruktionen in der Form <i>ist ... zu</i>	217
Tab. 55	Regeln für die Identifikation von Passiversatzkonstruktionen in der Form <i>bekommen</i>	218
Tab. 56	Regeln für die Identifikation von Passiversatzkonstruktionen in der Form <i>sich lassen</i>	219
Tab. 57	Regeln für die Identifikation von Passiversatzkonstruktionen in der Form <i>-bar</i>	220
Tab. 58	CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger Personalpronomina	221

Tab. 59	CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger substituierender Demonstrativpronomina	222
Tab. 60	CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger substituierender Relativpronomina	222
Tab. 61	CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger attribuierender Demonstrativpronomina	223
Tab. 62	CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger attribuierender Possessivpronomina	224

1 Einleitung

1.1 Motivation

Viele Experten erwarten, dass die Entwicklung von Software in der Automobilindustrie einen immer höheren Stellenwert bekommt, so zum Beispiel [SZ16]: „Software im Automobil zeigt sich als zentraler Innovationstreiber und das nicht nur in den vergangenen 30 Jahren. Auch in den nächsten 10 bis 20 Jahren sind enorme Innovationsschübe zu erwarten“.

Durch Integration neuer Funktionen, welche den Fahrkomfort und die Fahrsicherheit erhöhen, werden die Anforderungen an Software immer komplexer. Der Anteil an eingebetteten Systemen stellt in einem Fahrzeug bereits 35% der gesamten Wertschöpfung dar [Nör12], [Her06]. Es ist zu erwarten, dass dieser Anteil in Zukunft noch weiter steigen wird.

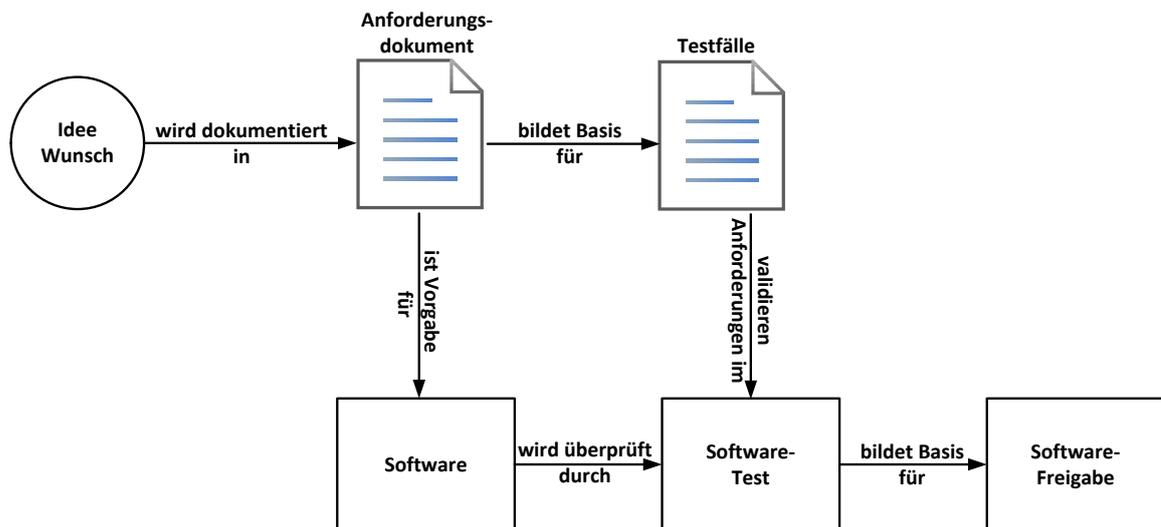


Abbildung 1: Softwareentwicklungsprozess

Um neue Funktionen in einem Fahrzeug zu realisieren, muss zunächst genau beschrieben werden, wie sich die entsprechende Komponente oder das entsprechende System in bestimmten Situationen verhalten muss, um die Funktionen bereitstellen zu können; z.B. wie ein bestimmtes Steuergerät auf bestimmte Eingangssignale zu reagieren hat. Die Definitionen der Funktionen von Komponenten und Systemen werden in der Software-Entwicklung Anforderungen genannt. Diese Anforderungen werden in einem Anforderungsdokument dokumentiert und stellen Vorgaben an die zu entwickelnde Software, die eingehalten werden müssen. Außerdem bilden Anforderungen die Basis für die Ableitung von Testfällen, welche wiederum die Anforderungen validieren (vgl. Abbildung 1). Die Disziplin, welche

sich mit der Ermittlung, Dokumentation, Analyse, Prüfung, Abstimmung und Verwaltung von Anforderungen befasst, ist das Requirements Engineering [Ebe12].

Die Entwicklung von Software-Systemen und -Komponenten wird in der Automobilindustrie immer mehr durch externe Unternehmen durchgeführt, d.h. die Realisierung der in den Anforderungen beschriebenen Funktionen wird von einem externen Lieferanten erbracht. Die Anforderungsdokumente stellen hierbei die Kommunikationsbasis zwischen Auftraggeber und Auftragnehmer dar. Qualitativ gute Anforderungen bilden somit eine wichtige Grundlage bei der Softwareentwicklung und haben einen großen Einfluss darauf, ob Projekte im geplanten Rahmen gelingen. Diverse Studien, welche durchgeführt wurden, um den aktuellen Stand im Requirements Engineering zu analysieren, unterstreichen diesen Aspekt. Ein Beispiel einer solchen Studie ist der *CHAOS* Report der *Standish Group* von 1994 [The]. Laut dieser Studie erreichten lediglich 16,2% der analysierten Projekte ihre Ziele innerhalb der geplanten Zeit, innerhalb des geplanten Budgets und unter Umsetzung aller geplanten Funktionalitäten. 52,7% der Projekte wurden zwar abgeschlossen jedoch nicht im Budgetrahmen, Zeitrahmen oder nicht mit allen geplanten Funktionalitäten. 31,1% der Projekte wurden gar nicht abgeschlossen. Über die Jahre hat sich die Erfolgsquote zwar gesteigert, jedoch ist die Zahl der Projekte, welche scheitern oder nicht im geplanten Rahmen bleiben, noch immer sehr hoch. Die Ergebnisse des *CHAOS* Reports von 2012 zeigen, dass zwar 39% der Projekte erfolgreich abgeschlossen werden, ohne die geplanten Kosten und die eingeplante Zeit zu überschreiten und unter Realisierung aller geplanten Funktionalitäten, jedoch werden laut der Studie noch immer 18% der Projekte abgebrochen. Der Anteil der Projekte, bei welchen die Zeit oder das Budget nicht eingehalten oder weniger Funktionen realisiert wurden, liegt bei 43%. Doch was ist der Grund dafür, dass Projekte nicht im geplanten Rahmen bleiben oder sogar komplett scheitern?

Project Challenged Factors	% of Responses
1. Lack of User Input	12,8%
2. Incomplete Requirements & Specifications	12,3%
3. Changing Requirements & Specifications	11,8%
4. Lack of Executive Support	7,5%
5. Technology Incompetence	7,0%
6. Lack of Resources	6,4%
7. Unrealistic Expectations	5,9%
8. Unclear Objectives	5,3%
9. Unrealistic Time Frames	4,3%
10. New Technology	3,7%
Other	23.0%

Tabelle 1: Herausforderungen in Projekten nach [The]

Im CHAOS Report der Standish Group [The] wird auch dieser Frage nachgegangen. Die Studie zeigt, dass ein Großteil der Herausforderungen, welche in Projekten auftreten, auf die Ermittlung, Dokumentation und Pflege von Anforderungen zurückzuführen ist. In Tabelle 1 sind die Herausforderungen, welche in der Studie ermittelt wurden, aufgelistet. Die Faktoren 1 bis 3 sowie Faktor 8 (= 42,2%) stehen deutlich in Verbindung mit der Erhebung und Dokumentation von Anforderungen.

Der CHAOS Report listet zusätzlich die Faktoren auf, die dazu führen, dass Projekte komplett scheitern. Unvollständige Anforderungen liegen hierbei auf Platz 1 der Liste (vgl. Tabelle 2).

Project Impaired Factors	% of Responses
1. Incomplete Requirements	13,1%
2. Lack of User Involvement	12,4%
3. Lack of Resources	10,6%
4. Unrealistic Expectations	9,9%
5. Lack of Executive Support	9,3%
6. Changing Requirements & Specifications	8,7%
7. Lack of Planning	8,1%
8. Didn't Need It Any Longer	7,5%
9. Lack of IT Management	6,2%
10. Technology Illiteracy	4,3%
Other	9,9%

Tabelle 2: Gründe für das Scheitern von Projekten nach [The]

Die Ergebnisse der Studie zeigen, dass die Qualität von Anforderungen einen immensen Einfluss auf die Qualität eines Projekts hat, und dass Anforderungen sogar einen entscheidenden Einfluss darauf haben, ob Projekte im geplanten Rahmen (Zeit, Kosten, Funktionalitäten) gelingen, diesen Rahmen überschreiten oder ob die Projekte scheitern.

Eine weitere Studie, die diesen Aspekt nochmals unterstreicht, ist der *RE-Kompass*. „Der RE-Kompass ist eine jährliche Befragung, durchgeführt vom Fraunhofer-Institut für Experimentelles Software Engineering IESE und der HOOD GmbH, in der der aktuelle Stand von Requirements Engineering (RE) in Wirtschaft und Industrie im deutschsprachigen Raum (Deutschland, Österreich, Schweiz) erfasst wird“ [AWK13]. In dieser Umfrage wurden die Teilnehmer u.a. mit der Frage konfrontiert, welche Techniken und Notationen für die Anforderungsdokumentation aktuell verwendet werden. Die Verwendung der natürlichen Sprache für die Dokumentation von Anforderungen ist noch immer die vorherrschende Technik (vgl. Abbildung 2). Modellbasierte Dokumentation (z.B. semi-formale Notationen wie *UML*) wird noch nicht so intensiv genutzt wie die Dokumentation mittels natürlicher Sprache. „Trotz vieler Fortschritte in der modellbasierten Entwicklung

und auch im modellbasierten RE haben textbasierte Anforderungen immer noch einen erheblichen Vorsprung“ [AWK13].

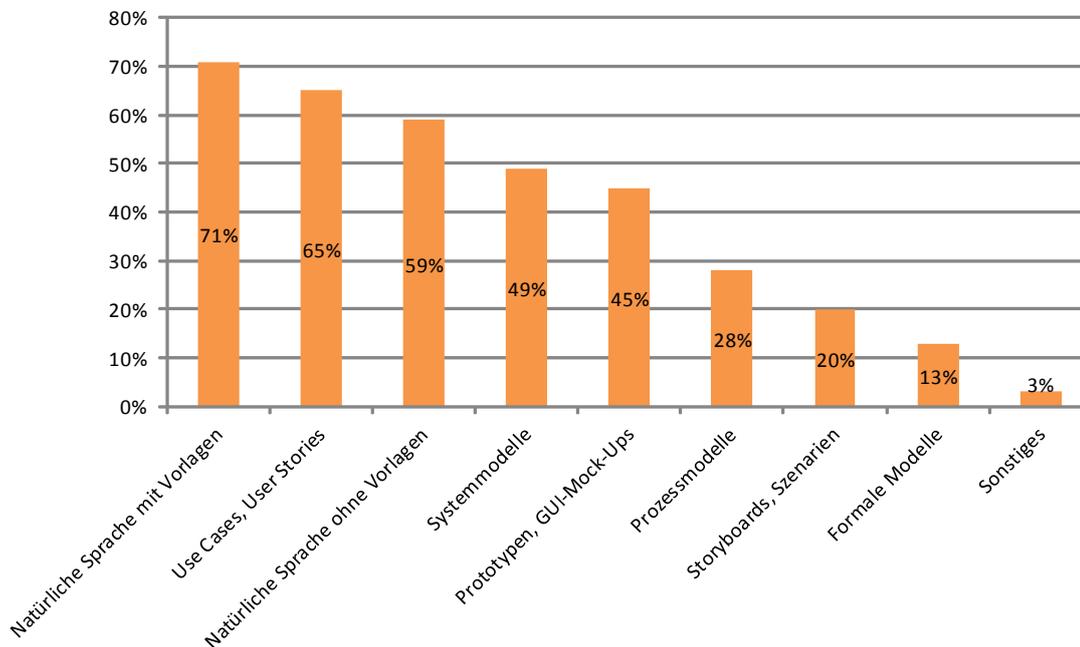


Abbildung 2: Techniken und Notationen für die Anforderungsdokumentation laut [AWK13]

Im RE-Kompass 2014/15 [ASW15] wurde außerdem ermittelt, welche Aspekte die Befragten als größte Schwäche in Anforderungsdokumenten sehen. Die Ergebnisse der Befragung zeigen, dass die größte Schwäche mehrdeutig und unvollständig formulierte Anforderungen und Anforderungsdokumente darstellen (vgl. Abbildung 3).

Auch in einer Studie von 2013, welche durch die *Pennsylvania State University* durchgeführt wurde, ist ein Ergebnis, dass Anforderungen auch heute noch mehrheitlich mittels natürlicher Sprache dokumentiert werden (61%) [KNL14]. Lediglich 33% der Befragten verwenden für die Dokumentation von Anforderungen semi-formale Notationen wie *UML*.

Die durchgeführten Studien zeigen, dass die natürliche Sprache noch immer das vorherrschende Mittel ist, um Anforderungen zu dokumentieren. Ein weiteres Ergebnis ist, dass Anforderungen eine der größten Herausforderungen bei der Durchführung von Projekten darstellen und sogar einer der Hauptgründe dafür sind, wieso Projekte komplett scheitern. Hierbei stellt sich die Frage, wieso die natürliche Sprache eine so große Herausforderung bei der Formulierung von Anforderungen darstellt.

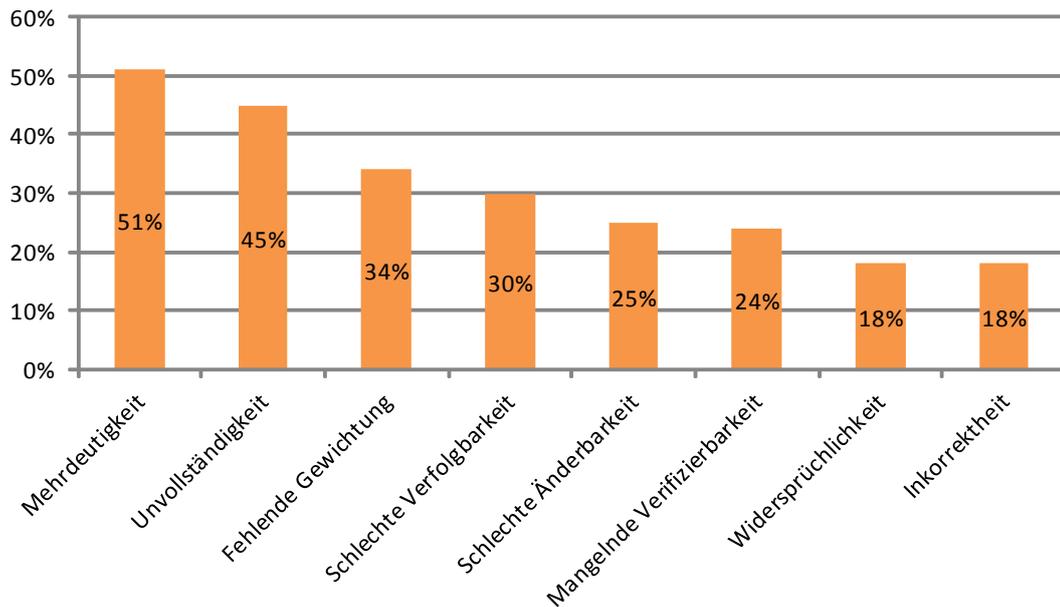


Abbildung 3: Größte Schwächen in Anforderungsdokumenten laut [ASW15]

Die Verwendung der natürlichen Sprache bringt Vor- und Nachteile mit sich. Natürliche Sprache ist universell einsetzbar, flexibel und leicht anwendbar. Da jeder ein intuitives Verständnis für Sprache hat, werden keine Schulungen benötigt, anders als z.B. für das Erstellen und Lesen formaler Modelle. Ein weiterer Vorteil ist, dass mittels natürlicher Sprache alle möglichen Sachverhalte ausgedrückt werden können [Kam01], [Poh07]. Jedoch birgt die natürliche Sprache auch eine große Gefahr: Beim Schreiben von Anforderungen können mehrdeutige Konstruktionen entstehen, die bei den Projektbeteiligten zu unterschiedlichen Interpretationen führen können. Dies hat nicht nur einen Einfluss auf die Entwicklung der Software, sondern auch einen großen Einfluss auf die Testaktivitäten, in welchen überprüft wird, ob eine entwickelte Komponente oder ein entwickeltes System die gewünschten Funktionen korrekt erfüllt. Dieser Phase kommt eine große Bedeutung zu; immerhin nehmen die Aufwendungen für die Testaktivitäten mittlerweile 25%-50% der Entwicklungszeit und Entwicklungskosten ein [PKS00].

Aus den Anforderungen werden Testfälle abgeleitet, durch welche die beschriebenen Funktionalitäten getestet werden können und eine Aussage darüber gemacht werden kann, ob das Ist-Ergebnis dem Soll-Ergebnis entspricht (vgl. Abbildung 1). Eine hohe Qualität der Anforderungen, welche als Grundlage für die Ableitung von Testfällen dienen, ist von großer Wichtigkeit. Wird aufgrund einer mehrdeutigen Anforderung ein falscher Testfall

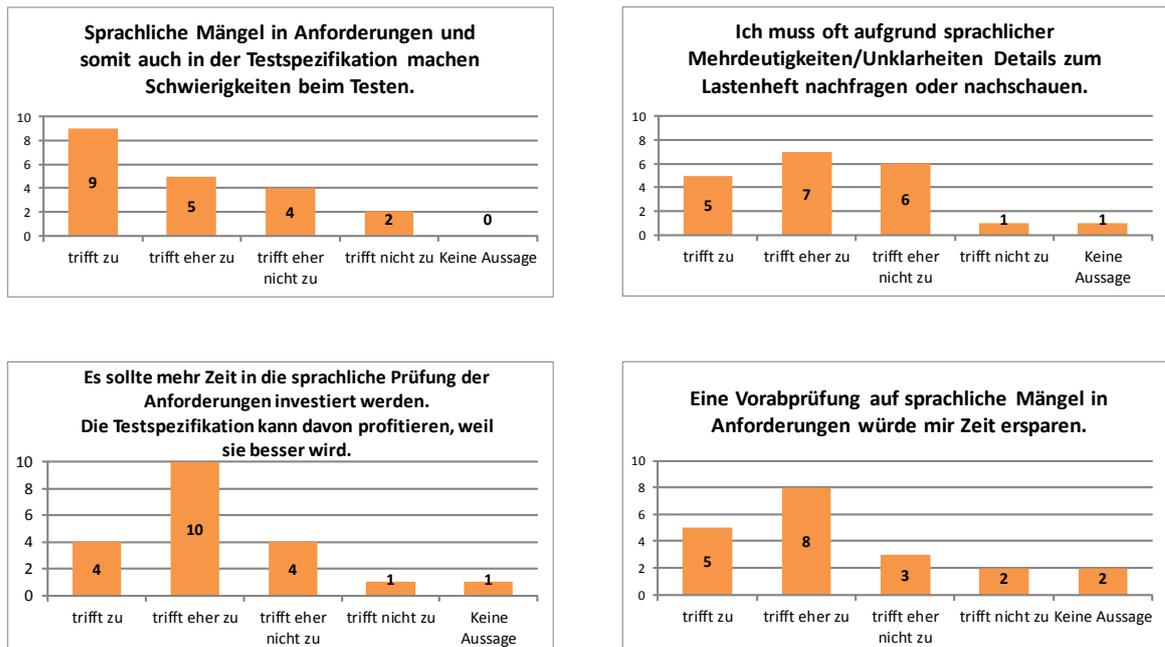


Abbildung 4: Ergebnisse einer Befragung von 20 Testexperten zur sprachlichen Qualität von Anforderungen im Automobilkontext

abgeleitet, kann dies zu Problemen führen. Im schlimmsten Fall wird eine falsch realisierte Anforderung durch einen falschen Testfall als korrekt umgesetzt klassifiziert.

Die vorliegende Arbeit ist in Zusammenarbeit mit der *Daimler AG* entstanden (vgl. Abschnitt 1.4). Auch hier stellen Anforderungen die Kommunikationsgrundlage zwischen Auftraggeber und Auftragnehmer dar: sie stellen Vorgaben an die Entwicklung und bilden die Basis für die Ableitung von Testfällen. Um nochmals zu unterstreichen, dass nicht nur in der Theorie, sondern auch in der Praxis verständliche und testbare Anforderungen gefordert werden, wurde eine Befragung von 20 Personen durchgeführt. Elf der befragten Personen arbeiten bei der *Daimler AG* und testen dort unterschiedliche Systeme, wie beispielsweise die Klimatisierung oder die Schließung im Auto. Neun weitere befragte Personen arbeiten in externen Testhäusern und testen ebenfalls unterschiedliche Systeme und Funktionalitäten in Kraftfahrzeugen. Da die befragten Tester unterschiedliche Systeme des Autos testen, arbeiten sie mit unterschiedlichen Anforderungsdokumenten und Testspezifikationen. Diese Dokumente weisen eine unterschiedlich hohe Qualität auf. Ein weiterer wichtiger Punkt ist, dass die Befragten unterschiedlich lang im Bereich des Softwaretestens arbeiten und bereits unterschiedlich viele Erfahrungswerte gesammelt haben.

Die Befragung hat ergeben, dass der Großteil der Tester (= 14 von 20) der Meinung ist, dass sprachliche Mängel in Anforderungen Schwierigkeiten beim Testen machen (vgl.

Abbildung 4). Lediglich sechs Tester sind der Meinung, dass sprachliche Mängel in den Anforderungen keine oder fast keine Schwierigkeiten beim Testen machen. Vermutlich arbeiten diese Tester mit qualitativ guten Spezifikationen oder haben bereits viel Erfahrung mit den zu testenden Systemen, sodass sprachliche Mängel für diese Personen kein großes Problem darstellen. Ein weiteres Ergebnis der Befragung ist, dass der Großteil der Tester oft aufgrund sprachlicher Mehrdeutigkeiten bzw. Unklarheiten Details zum Lastenheft nachfragen muss (vgl. Abbildung 4). Dies stellt insbesondere bei den externen Testern eine Herausforderung dar, weil sie nicht immer sofort direkt beim Systemverantwortlichen Details nachfragen können. Nahezu alle Tester der externen Testhäuser haben diese Aussage als zutreffend bewertet. Die Bewertung der internen Tester variiert. Weiterhin hat die Umfrage ergeben, dass es für die Tester von Vorteil wäre, wenn mehr Zeit in die sprachliche Prüfung von Anforderungen investiert werden würde und dass eine Vorabprüfung auf sprachliche Mängel in Anforderungen ihnen Zeit ersparen würde (vgl. Abbildung 4).

Diverse Studien, welche den Stand im Requirements Engineering ermitteln, und die oben angesprochene Befragung der Tester ergeben, dass die sprachliche Qualität von Anforderungen einen wesentlichen Einfluss auf den Projekterfolg und auf die folgenden Testaktivitäten hat. Da die Aufwendungen für die Testaktivitäten mittlerweile einen großen Teil der Entwicklungszeit und Entwicklungskosten einnehmen, liegt es nahe, die Anforderungen auf ihre sprachliche Qualität zu prüfen und sie insbesondere dahingehend zu betrachten, ob sie so formuliert sind, dass sie testbar sind. Jedoch stellt sich die Frage, wann eine Anforderung das Kriterium der Testbarkeit erfüllt, bzw. welche sprachlichen Phänomene dazu führen, dass dieses Kriterium verletzt ist.

1.2 Verortung in der Wissenschaft

Anforderungen werden heutzutage noch immer größtenteils mittels natürlicher Sprache verfasst und haben einen Einfluss auf den Erfolg und Misserfolg von Projekten. Da aus den natürlichsprachlichen Anforderungen Testfälle abgeleitet werden und die Testaktivitäten einen Großteil der Entwicklungszeit und Entwicklungskosten einnehmen, ist es sinnvoll die Qualität der Anforderungen im Hinblick auf Testbarkeit zu überprüfen.

Die Verständlichkeitsforschung beschäftigt sich schon seit vielen Jahren mit der Fragestellung was einen Text “gut” lesbar macht. Bereits 1948 wurde die Lesbarkeitsformel von Flesch [Fle48] entwickelt, welche die Kriterien der Satz- und Wortlänge einbezieht. Die Lesbarkeitsformel liefert einen Wert, welcher angibt wie lesbar ein Text ist. Die Formel von Flesch wurde über die Jahre weiterentwickelt. Weitere Formeln berücksichtigen beispielsweise die Komplexität von Konzepten [SC01], Textkohärenz oder das Verhältnis zwischen

“einfachen” und “schwierigen” Konjunktionen und Adverbien [Ben05]. Jedoch zielen diese Formeln lediglich auf eine rein quantifizierende Beurteilung von Texten ab (vgl. Abschnitt 3.1.1).

Das *Hamburger Verständlichkeitsmodell* [LST74] und das Verständlichkeitsmodell von Groeben [Gro72] sind Modelle, die die Beschränkungen, welche die Verständlichkeitsformeln aufweisen, versuchen zu überwinden. Beide Modelle stellen vier Dimensionen der Verständlichkeit auf, mit welchen beurteilt werden soll, wie verständlich ein Text ist. Die Relevanz der einzelnen Dimensionen unterscheidet sich in beiden Modellen. Im Gegensatz zum Hamburger Verständlichkeitsmodell bietet das Modell von Groeben außerdem Vorschläge dazu wie ein Text strukturiert werden sollte, es werden Hinweise zum Schreibstil gegeben und es werden sprachliche Phänomene erwähnt, welche einen Einfluss auf die Verständlichkeit haben können (vgl. Abschnitt 3.1.2).

Die Forschung zur technischen Dokumentation lehnt sich an die Ergebnisse der Verständlichkeitsforschung an. Auch im Bereich der technischen Dokumentation spielt die Verständlichkeit eines Textes eine entscheidende Rolle, da hier das primäre Ziel ist, Bedienungsanleitungen für einen Benutzer eines technischen Geräts verständlich zu machen. Zu diesem Zwecke werden in der Literatur zur technischen Dokumentation Regeln aufgelistet, welche ein Autor einhalten soll, um verständliche Texte zu schreiben. Die *tekom*-Leitlinie für technische Dokumentation [tek11] ist die wohl bekannteste solche Leitlinie. Daran angelehnt wurde ein weiterer Leitfaden durch die Duden-Redaktion erstellt [Sch12]. Beide Empfehlungen stützen sich auf Erkenntnisse aus der Verständlichkeitsforschung (vgl. Abschnitt 3.2).

Das *Requirements Engineering* und das *Requirements Management* befassen sich mit der Formulierung und Verwaltung von Anforderungen. Auch in diesem Bereich existieren Ansätze und Techniken dafür wie ein Autor gute Texte, sprich gute Anforderungen, schreiben kann (vgl. Kapitel 5). Es werden beispielsweise Satzschablonen vorgeschlagen [Rup14], [MW10]. Mit diesen Schablonen kann man bei der Anforderungsformulierung Schritt für Schritt vorgehen und es wird eine bestimmte Struktur und ein eingeschränktes Vokabular vorgegeben. Außerdem existieren Regelwerke dafür wie man gute Anforderungen schreibt [Poh07], [Rup14], [PR15]. Es werden eine Reihe sprachlicher Phänomene angegeben, die dazu führen können, dass kein einheitlicher Konsens über die Anforderungen zwischen allen Projektbeteiligten besteht. Ein Problem kann beispielsweise die mangelnde Verständlichkeit einer Anforderung sein; ein anderes etwa ein inhaltlicher Widerspruch. Unpräzise Adjektive ohne Angabe einer Quantifizierung können ebenso ein Problem darstellen. Auch die Verwendung des Passivs kann problematisch sein, weil der Ausführende

oder das ausführende System bzw. die ausführende Komponente nicht explizit im Text genannt wird.

Der Bereich der Sprachtechnologie und der Bereich des Requirements Engineering stellen Werkzeuge bereit, mit welchen Texte automatisiert geprüft werden können. Hierbei werden nicht nur quantitative Merkmale wie Satz- und Wortlänge untersucht, sondern auch syntaktische Prüfungen wie beispielsweise eine Prüfung auf Passivsätze bereitgestellt.

All diese Ansätze haben den Fokus darauf, eindeutige und leicht lesbare Texte zu produzieren oder Texte auf Verständlichkeit zu überprüfen. Jedoch wird der Aspekt der Testbarkeit durch keinen der Ansätze geprüft. Untersuchungen darüber, wie eine testbare Anforderung aufgebaut sein muss bzw. welche sprachlichen Phänomene einen Einfluss auf die Testbarkeit haben, existieren nicht.

1.3 Beiträge der Arbeit

Ausgehend von den oben überblicksweise identifizierten Herausforderungen und den Grenzen existierender Techniken sind für die vorliegende Arbeit vier Aufgabenstellungen abgeleitet worden, welche die zentralen Beiträge der Arbeit darstellen:

Beitrag 1: *Entwicklung eines Qualitätsmodells für testbare Anforderungen.*

Das erste Ziel dieser Arbeit besteht darin zu untersuchen, welche Eigenschaften eine Anforderung aus sprachwissenschaftlicher Sicht erfüllen muss, damit sie testbar ist. In diesem Zusammenhang wird in dieser Arbeit ein Qualitätsmodell für testbare Anforderungen vorgestellt. Dieses Qualitätsmodell trägt linguistische Phänomene aus der Requirements Engineering-Literatur und aus der Literatur der technischen Dokumentation zusammen, sortiert diese Phänomene nach linguistischen Aspekten und bewertet die Phänomene nach Kritikalität. Mit diesem Qualitätsmodell wird eine Übersicht über die sprachlichen Phänomene gegeben, die potentiell einen Einfluss auf die Testbarkeit einer Anforderung haben können.

Beitrag 2: *Darstellung der Unterschiede und Deckungsgleichheit zwischen den präsentierten Phänomenen in Literatur zur technischen Dokumentation und zum Requirements Engineering.*

Ein weiterer Untersuchungsgegenstand besteht darin abzugleichen, inwiefern sich die in der Requirements Engineering-Literatur aufgeführten Phänomene mit den Phänomenen aus der Literatur zur technischen Dokumentation decken. In der technischen Dokumentation liegt der Fokus darauf, verständliche Texte für Jedermann zu schreiben, im Bereich

des Requirements Engineering werden Texte für Experten geschrieben. Gemeinsamkeiten und Unterschiede der Regelwerke werden in dieser Arbeit herausgearbeitet.

Beitrag 3: *Durchführung einer Analyse dazu inwiefern sich die identifizierten Phänomene mittels computerlinguistischer Korpusanalysemethoden identifizieren und nach Kritikalität klassifizieren lassen.*

Einen weiteren Beitrag liefert die vorliegende Arbeit insofern, als dass auch untersucht wird, inwiefern sich die als relevant identifizierten Phänomene mit Hilfe von computerlinguistischen Korpusanalysemethoden identifizieren und nach Kritikalität klassifizieren lassen. Die Ergebnisse der Analysen werden mit Analysen bereits existierender Werkzeuge abgeglichen und bewertet.

Beitrag 4: *Entwicklung und Evaluierung von automatisierten Prüfmethoden für die Identifikation kritischer Instanzen bezüglich Testbarkeit.*

Für die Phänomene, bei welchen sich die kritischen Instanzen zuverlässig automatisiert identifizieren lassen, werden Prüfmethoden entwickelt, die in ein Daimler-internes Werkzeug einfließen. Hierfür werden die in Beitrag 3 identifizierten Korpusanalysemethoden genutzt.

1.4 Rahmen der Arbeit

Die vorliegende Arbeit ist in Zusammenarbeit mit der *Mercedes-Benz PKW-Entwicklung* der *Daimler AG* entstanden. Die Untersuchungen und Ergebnisse, die in dieser Arbeit durchgeführt und gewonnen werden, basieren auf Anforderungsdokumenten der Daimler AG. Die Anforderungsdokumente werden in einem klassischen Auftraggeber-Auftragnehmer-Verhältnis verwendet: Für die Entwicklung einer Komponente wird ein Lieferant ausgewählt, der die Aufgabe hat, die Funktionen einer Komponente entsprechend den dokumentierten Anforderungen umzusetzen. Die Anforderungen in einem Anforderungsdokument sind daher typischerweise an den Auftragnehmer gerichtet (Prozessanforderungen) oder an die Komponente, die der Auftragnehmer entwickelt (Produktanforderungen). Die Dokumentation von Anforderungen wird von Daimler selbst durchgeführt. Die durch den Lieferanten entwickelte Komponente wird in den meisten Fällen durch Daimler integriert und getestet.

Da jedoch die Entwicklung und auch immer mehr die Testaktivitäten an externe Dienstleister vergeben werden, ist eine hohe Qualität von Anforderungen von großer Wichtigkeit. Eine Untersuchung darüber, welche sprachlichen Eigenschaften eine Anforderung erfüllen muss, um als testbar zu gelten, spielt in der Praxis eine entscheidende Rolle.

1.5 Struktur und Aufbau der Arbeit

Um die Beiträge und deren zugrundeliegenden Forschungsfragen aus Abschnitt 1.3 den einzelnen Kapiteln der Arbeit zuordnen zu können, werden in diesem Abschnitt die Struktur und der Aufbau der Arbeit präsentiert. Die vorliegende Arbeit ist nicht nur einer wissenschaftlichen Disziplin zuzuordnen, sie involviert Forschungen aus unterschiedlichen Bereichen: Ein Teil der Arbeit basiert auf Forschungsergebnissen der Computerlinguistik und verwendet Werkzeuge und Methoden aus diesem Bereich. Die Arbeit stützt sich außerdem auf Forschungsergebnisse und Methoden aus dem Bereich des Requirements Engineering. Weitere Teile der Arbeit sind von Ansätzen und Ergebnissen der angewandten Sprachwissenschaft inspiriert. Die Untersuchungen und Analysen werden an Daimler-internen Dokumenten durchgeführt, welche den Anwendungskontext der Arbeit darstellen. In Abbildung 7 sind die Kapitel der Arbeit aufgelistet und den unterschiedlichen Bereichen zugeordnet. Pfeile markieren, welche Teile der Arbeit miteinander interagieren und welche Bausteine der Arbeit aufeinander aufsetzen.

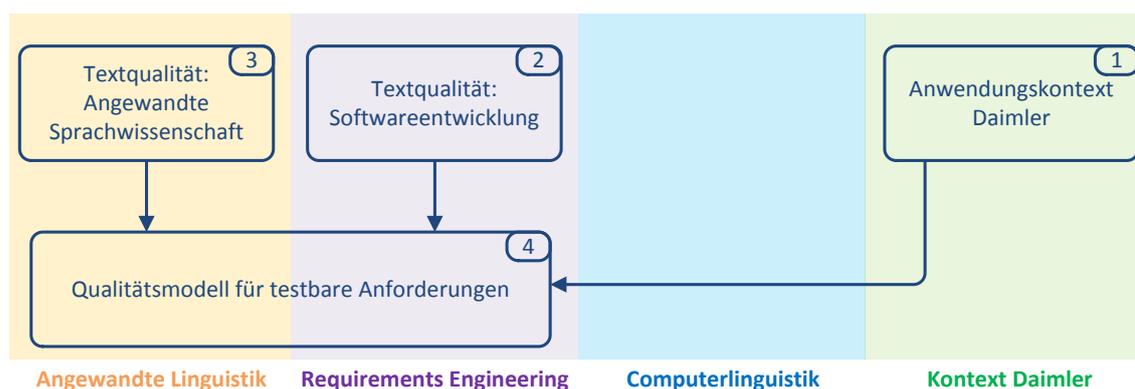


Abbildung 5: Struktur und Aufbau der Arbeit: Beiträge 1 und 2

Beitrag 1 wird in den Kapiteln 1-4 herausgearbeitet (vgl. Abbildung 5). In Kapitel 2 wird zunächst auf die für das Verständnis benötigten Grundlagen der systematischen Softwareentwicklung (Software Engineering) eingegangen, und die Relevanz der Textqualität für diesen Bereich wird herausgearbeitet. Anschließend wird in Kapitel 3 auf die Ergebnisse der angewandt-linguistischen Verständlichkeitsforschung und deren Relevanz für die Probleme im Requirements Engineering eingegangen. Die Ergebnisse dieser Untersuchungen fließen direkt in das in dieser Arbeit entwickelte Qualitätsmodell für testbare Anforderungen ein, welches in Kapitel 4 vorgestellt wird. Das Qualitätsmodell für testbare Anforderungen trägt Phänomene aus der Literatur zum Requirements Engineering und aus der Literatur zur technischen Dokumentation zusammen, sortiert diese Phänomene systematisch und bewertet die aufgelisteten Phänomene nach Kritikalität.

In Kapitel 4 wird außerdem *Beitrag 2* herausgearbeitet. Es wird ein Überblick über die betrachtete Literatur im Bereich des Requirements Engineering und im Bereich der technischen Dokumentation präsentiert, und es wird gezeigt inwiefern sich die in der Literatur präsentierten Phänomene der beiden Bereiche überdecken und wo Unterschiede zu finden sind. In Kapitel 4 wird außerdem beschrieben und motiviert, nach welchen Kriterien die im Detail zu behandelnden Phänomene ausgewählt werden.

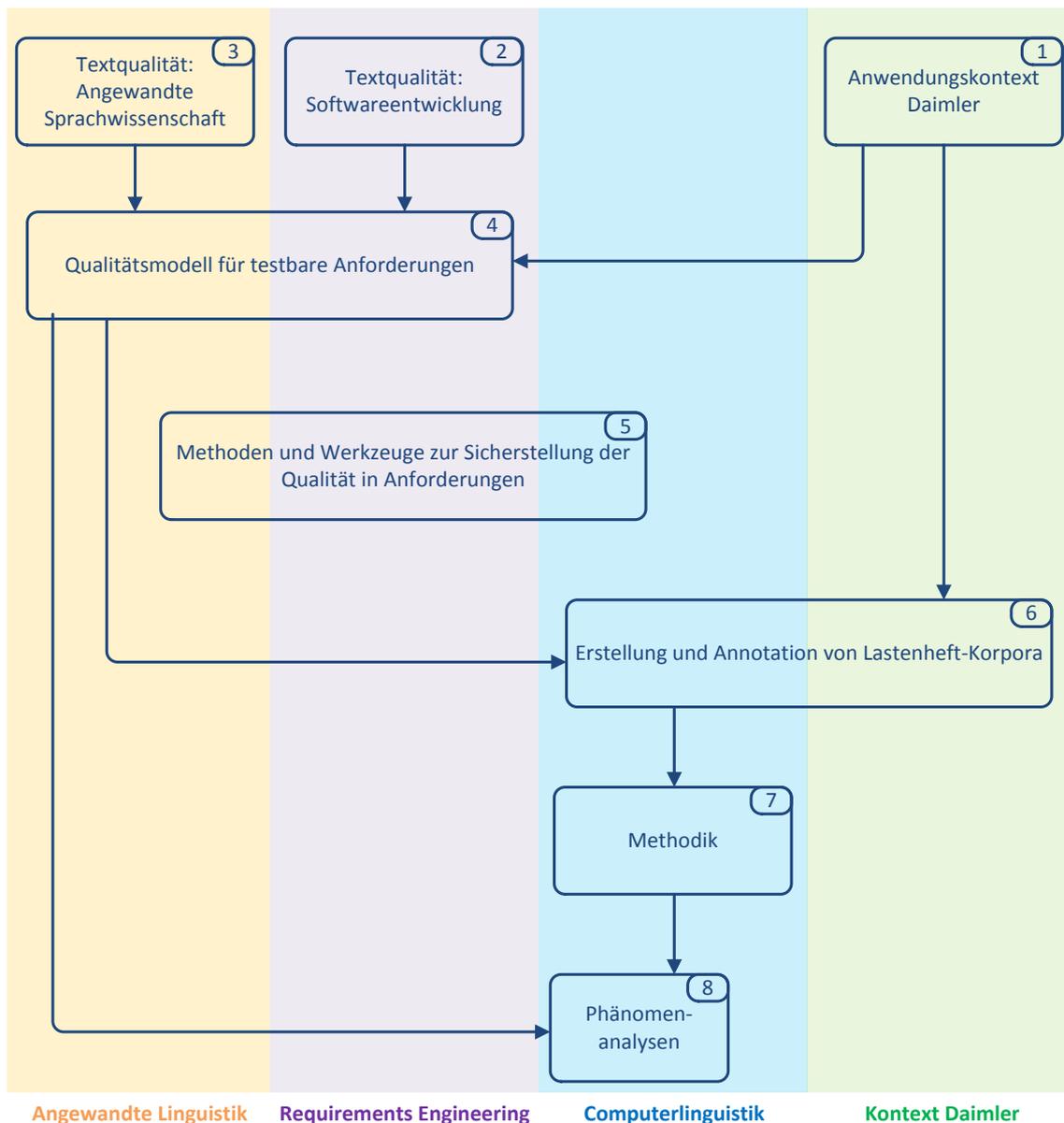


Abbildung 6: Struktur und Aufbau der Arbeit: Beiträge 1-3

Die Kapitel 5-7 analysieren den aktuellen Stand der Technik und dienen der Vorbereitung der in Kapitel 8 präsentierten Phänomenanalysen, welche den dritten Beitrag der Arbeit

zur einschlägigen aktuellen Forschung darstellen (vgl. Abbildung 6). In Kapitel 5 wird zunächst der aktuelle Stand der Technik beschrieben: Es werden Methoden präsentiert, mit welchen bereits beim Schreiben von Anforderungen mehrdeutige Strukturen vermieden werden können. Weiterhin werden Techniken vorgestellt, mit welchen man bestehende Anforderungen automatisiert auf ihre Qualität hin überprüfen kann. Die präsentierten Werkzeuge und Methoden stammen aus dem Requirements Engineering und aus dem Bereich der technischen Dokumentation bzw. der angewandten Sprachwissenschaft und basieren auf Techniken aus der Computerlinguistik.

In Kapitel 6 wird in Vorbereitung der geplanten Phänomenanalysen, welche in Kapitel 8 präsentiert werden, die Erstellung und Annotation von Lastenheft-Korpora beschrieben. Weiterhin wird ein Überblick über die Größe der Textdaten gegeben. In Kapitel 7 wird anschließend die Methodik für die Entwicklung und Evaluierung der in Kapitel 8 präsentierten Regeln zur Prüfung der Textqualität von Anforderungen dargestellt und diskutiert.

In Kapitel 8 werden die durchgeführten Phänomenanalysen vorgestellt. Jedes Phänomen wird aus sprachwissenschaftlicher Sicht beschrieben und falls möglich werden Regeln für die automatische Identifikation der kritischen Instanzen präsentiert. Für die Phänomene, für die Regeln entwickelt werden können, wird abschließend jeweils eine Evaluierung der Regeln durchgeführt. Hierbei wird gezeigt, wie gut die entwickelten Regeln die kritischen von den unkritischen Instanzen unterscheiden können.

Beitrag 4 wird in den Kapiteln 8 und 9 herausgearbeitet (vgl. Abbildung 7). In Kapitel 9 werden die in dieser Arbeit entwickelten Analysen mit Analysemethoden verglichen, welche durch kommerzielle und frei verfügbare Werkzeuge bereitgestellt werden. Außerdem wird ein Werkzeug präsentiert, in welches die entwickelten Analysen einfließen, sodass sie produktiv genutzt werden können.

In Kapitel 10 wird anschließend kritisch auf die empirischen Ergebnisse sowie auf die Gefährdungen der Validität eingegangen. Die Arbeit wird mit einer zusammenfassenden Betrachtung sowie einem Ausblick auf weiterführende Arbeiten abgeschlossen (vgl. Kapitel 11).

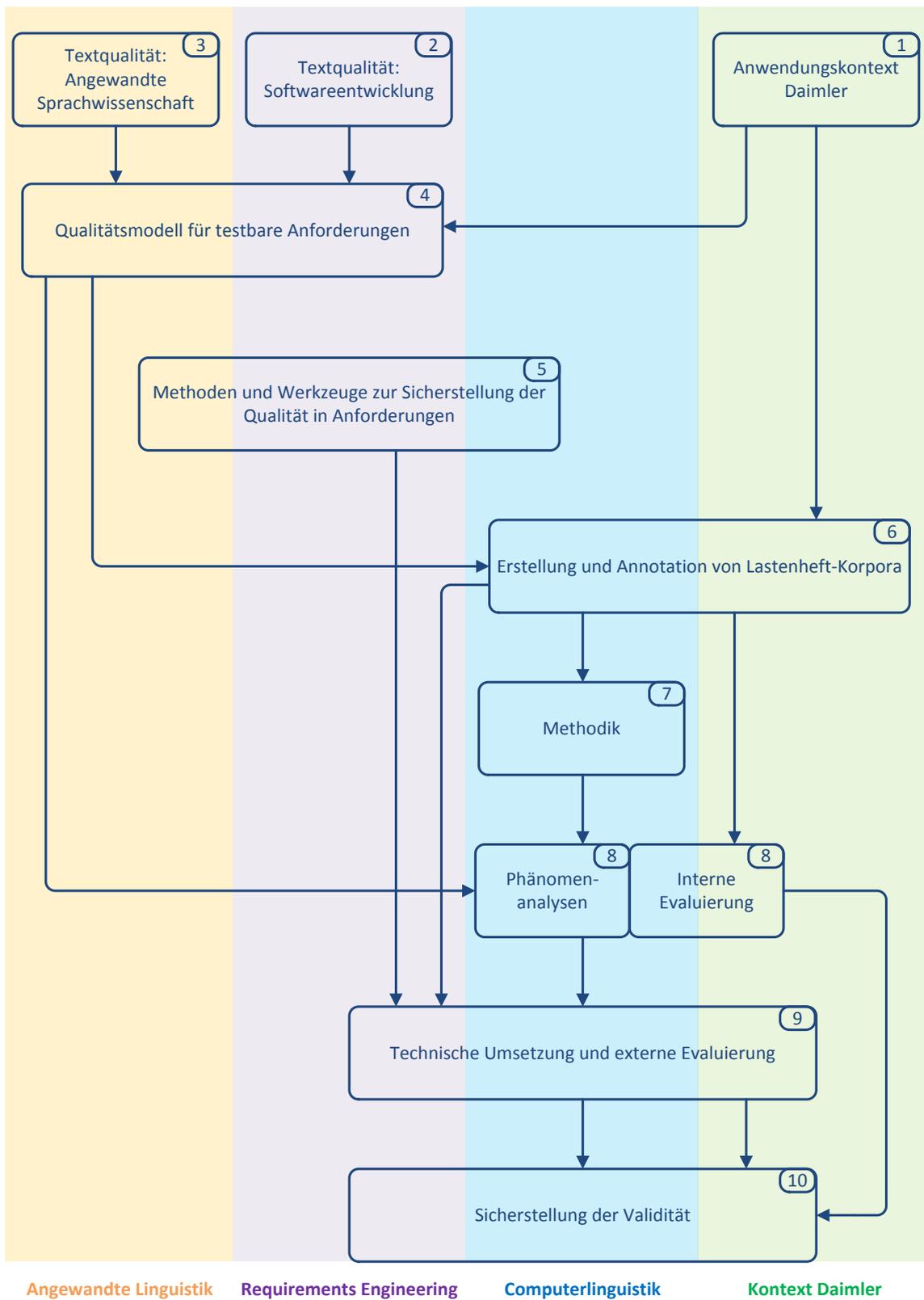


Abbildung 7: Struktur und Aufbau der Arbeit: Beiträge 1-4

2 Textqualität aus der Sicht der Softwareentwicklung

Im vorliegenden Kapitel werden die für das Verständnis der vorliegenden Arbeit relevanten Grundlagen der Softwareentwicklung beschrieben, und die Relevanz der Textqualität in diesem Umfeld wird herausgearbeitet. Zunächst werden die für die Arbeit zentralen Begriffe des Software Engineering und Requirements Engineering definiert und erläutert. In diesem Zusammenhang wird auf die verschiedenen Arten von Anforderungen, auf die Dokumentation von Anforderungen und abschließend auf die Überprüfung von Anforderungen eingegangen. Darauf folgend wird ein kompakter Überblick über die Testaktivitäten in der Software-Entwicklung gegeben und abschließend wird die Relevanz der Eigenschaft der Testbarkeit von Anforderungen beschrieben.

2.1 Systematische Softwareentwicklung

Die Entwicklung und Anwendung von Software wird unter dem Begriff des Software Engineering (= Softwaretechnik) zusammengefasst.

Definition 1 (Software Engineering [Bal09]) *Zielorientierte Bereitstellung und systematische Verwendung von Prinzipien, Methoden und Werkzeugen für die arbeitsteilige, ingenieurmäßige Entwicklung und Anwendung von umfangreichen Softwaresystemen. Zielorientiert bedeutet die Berücksichtigung z.B. von Kosten, Zeit und Qualität.*

„Um eine strukturierte und steuerbare Softwareentwicklung durchzuführen, werden Softwareentwicklungsmodelle bzw. Entwicklungsprozesse eingesetzt“ [SL12]. Über die Jahre wurden zahlreiche Modelle entwickelt. Bekannte Beispiele sind das Wasserfallmodell [Boe81], das allgemeine V-Modell [Boe79] und das Vorgehensmodell-XT des Bundes und der Länder [HH08]. „Weitere sind das Spiralmodell, unterschiedliche inkrementelle und agile oder leichtgewichtige Methoden“ [SL12]. Im Folgenden wird auf das Wasserfallmodell und das allgemeine V-Modell eingegangen. Das Wasserfallmodell stellt neben agilen Methoden das am häufigsten verwendete Vorgehensmodell dar [KNL14].

Beim Wasserfallmodell laufen die einzelnen Aktivitäten der Softwareentwicklung sequentiell getrennt voneinander ab. Die einzelnen Phasen nach [Par10] sind: Analyse und Definition, Entwurf, Implementation, Integration, Test und Abnahme sowie Einsatz und Wartung (vgl. Abbildung 8). Nach Abschluss einer Entwicklungsphase beginnt die nächste Phase. „Der entscheidende Nachteil ist, dass das Testen als einmalige Aktion am Projektende vor der Inbetriebnahme aufgefasst wird“ [SL12]. Dies betrifft aber nicht nur die Phase des Testens, sondern auch alle anderen Aktivitäten. Anforderungen und Rahmenbedingungen (vgl. Abschnitt 2.2.1) können sich während der Entwicklung ändern, wodurch eine Rück-

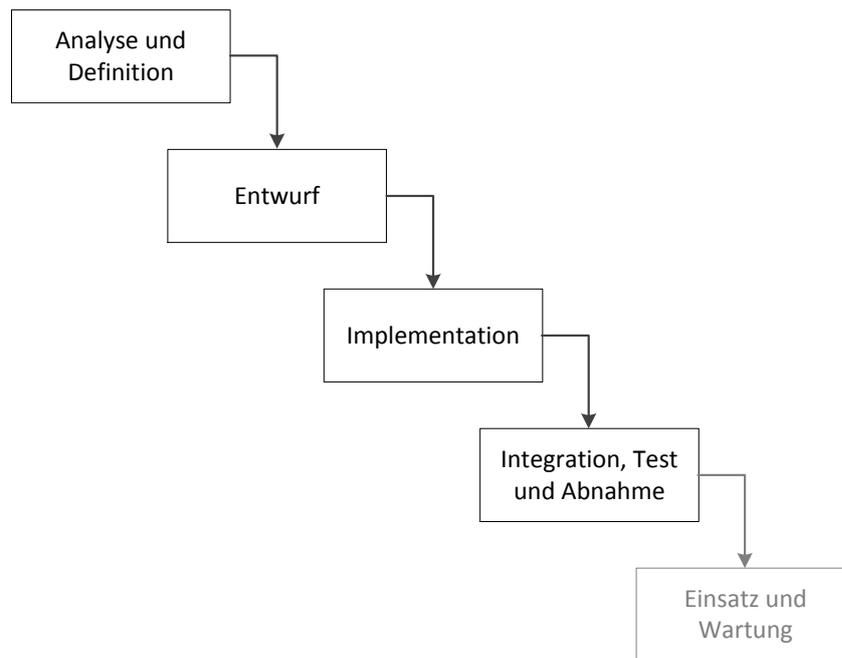


Abbildung 8: Wasserfallmodell nach [Par10]

kopplung der Phasen gewährleistet sein sollte. Sinnvoll ist es auch die Testaktivitäten möglichst parallel zu den Entwicklungsaktivitäten durchzuführen. Diesen Ansatz bietet das V-Modell.

„Die Grundidee des allgemeinen V-Modells ist, dass Entwicklungsarbeiten und Testarbeiten zueinander korrespondierende, gleichberechtigte Tätigkeiten sind“ [SL12]. In Abbildung 9 ist eine Darstellung des allgemeinen V-Modells zu sehen. Im linken Ast sind die Entwicklungsaktivitäten und im rechten Ast die Testaktivitäten abgebildet. Ausgehend von einer lösungsneutralen Spezifikation wird durch Verfeinerung der Anforderungen sukzessive die Umsetzung abgeleitet. Jeder Abstraktionsstufe ist hier eine Testphase zugeordnet. „In jeder Teststufe ist also zu überprüfen, ob die Entwicklungsergebnisse diejenigen Anforderungen erfüllen, die auf der jeweiligen Abstraktionsstufe relevant bzw. spezifiziert sind“ [SL12]. In der Automobilindustrie wird häufig nach dem V-Modell entwickelt. Typisch sind dort mehrere hintereinander ablaufende Instanzen des V-Modells [SZ16].

Es wird zwischen zwei Arten von Überprüfungen unterschieden: Validierung und Verifizierung. Die Validierung ist „der Prozess der Beurteilung eines Systems oder einer Komponente während oder am Ende des Entwicklungsprozesses, mit dem Ziel, festzustellen, ob die spezifizierten Anforderungen erfüllt sind“¹ [IEE12]. Bei der Validierung wird also

¹Original: “The process of evaluating a system or component during or at the end of a development process to determine whether it satisfies specified requirements.”

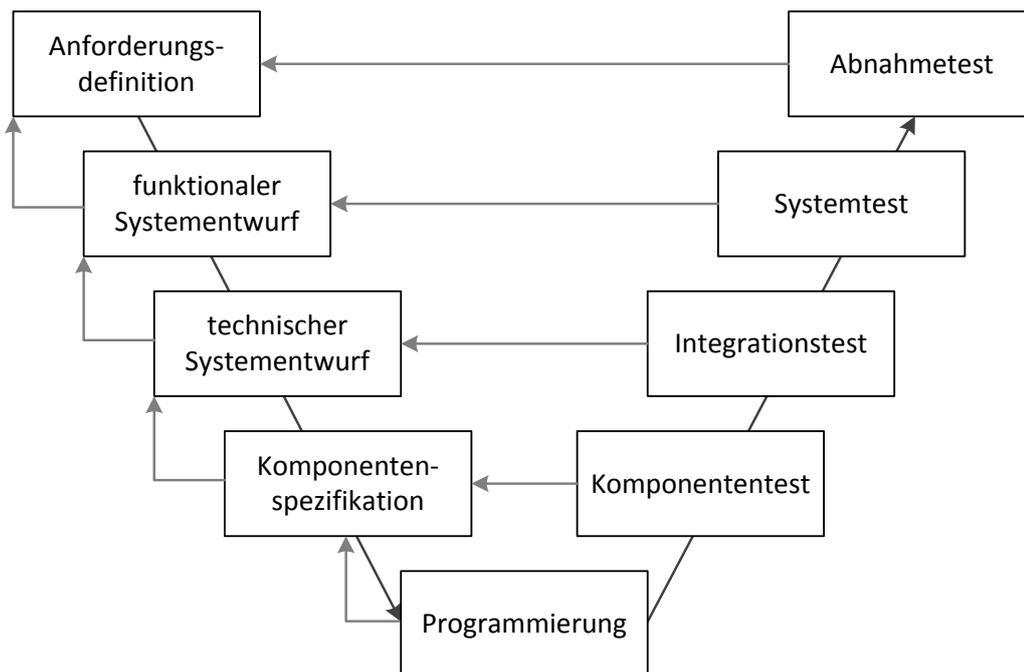


Abbildung 9: Allgemeines V-Modell nach [SL12]

geprüft, ob ein Arbeitsergebnis geeignet ist, die Erwartungen der Endbenutzer zu erfüllen, d.h. es wird geprüft, ob das richtige Produkt entwickelt wird.

Die Verifizierung ist hingegen „der Prozess der Beurteilung eines Systems oder einer Komponente mit dem Ziel, festzustellen, ob die Resultate einer gegebenen Entwicklungsphase den Vorgaben für diese Phase entsprechen“² [IEE12]. Bei der Verifizierung wird also geprüft, ob das Arbeitsergebnis den für den betreffenden Arbeitsschritt gegebenen Vorgaben entspricht, d.h. es wird geprüft, ob das Produkt richtig entwickelt wird.

In der Literatur finden sich noch weitere, zum Teil auch widersprüchliche Definitionen für die Begriffe Validierung und Verifizierung. Spillner [SL12] beispielsweise bezeichnet die Validierung als einen Prozess, in welchem überprüft wird, ob die Entwicklungsergebnisse die Anforderungen erfüllen, die auf der zugeordneten Abstraktionsstufe spezifiziert sind. Verifikation hingegen ist „auf eine einzelne Entwicklungsphase bezogen und soll die Korrektheit und Vollständigkeit eines Phasenergebnisses relativ zu seiner direkten Spezifikation (Phaseneingangsdokument) nachweisen“ [SL12].

²Original: “The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed of the start of that phase.”

In der vorliegenden Arbeit werden die Definitionen des IEEE-Standards 1012TM-2012 [IEE12] verwendet.

Im Folgenden wird nun auf beide Seiten des V-Modells, also auf die Entwicklungs- und auf die Testaktivitäten, eingegangen. Bei den Entwicklungsaktivitäten wird insbesondere die Teildisziplin *Requirements Engineering* beschrieben.

2.2 Requirements Engineering

Das Requirements Engineering hat die Aufgabe Anforderungen zu ermitteln, zu dokumentieren, zu überprüfen, abzustimmen und über den gesamten Lebenszyklus eines Systems zu verwalten [Poh96], [PR15]. Das Requirements-Engineering „ist eine Kerndisziplin aller Ingenieurwissenschaften“ [Ebe12]. In schwergewichtigen Vorgehensmodellen wie z.B. dem Wasserfallmodell und dem V-Modell wird das Requirements Engineering als abgeschlossene, zeitlich befristete erste Phase der Entwicklung durchgeführt [PR15]. Im Folgenden wird zunächst der Begriff Requirements Engineering definiert und anschließend werden die für das Verständnis nötigen Begriffe erklärt. In Definition 2 ist die Definition gemäß IREB (*International Requirements Engineering Board*) zu sehen.

Definition 2 (Requirements Engineering [PR15]) *Das Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:*

- (1) Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.*
- (2) Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren sowie die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht.*

Die vier Haupttätigkeiten im Requirements Engineering sind die Folgenden [PR15]:

- 1) Ermitteln,
- 2) Dokumentieren,
- 3) Prüfen und Abstimmen,
- 4) Verwalten.

Beim Ermitteln der Anforderungen (auch: Anforderungserhebung) geht es darum zu erfassen, welche Anforderungen im nächsten Schritt dokumentiert werden sollen. Hilfsmit-

tel sind hierbei beispielsweise Befragungstechniken wie Interviews und Fragebögen. Auch kreative Techniken wie Brainstorming und das Sechs-Hut-Denken (vgl. [DB06]) sowie beobachtende Techniken wie das *Apprenticing* (vgl. [Rup14]) sind Techniken für die Ermittlung von Anforderungen.

In den nächsten Schritten werden die Anforderungen in einer Anforderungsspezifikation dokumentiert, geprüft und abgestimmt. Bei der letzten Tätigkeit handelt es sich um die Anforderungsverwaltung, auch RM (= Requirements Management) genannt. „Die Anforderungsverwaltung [...] geschieht flankierend zu allen anderen Aktivitäten und umfasst alle Maßnahmen, die notwendig sind, um Anforderungen zu strukturieren, für unterschiedliche Rollen aufzubereiten sowie konsistent zu ändern und umzusetzen“ [PR15].

Mit unterschiedlichen Rollen sind diejenigen Personen gemeint, die einen Einfluss auf die Anforderungen eines Systems haben (vgl. Definition 3).

Definition 3 (Stakeholder [PR15]) *Ein Stakeholder eines Systems ist eine Person oder Organisation, die (direkt oder indirekt) Einfluss auf die Anforderungen des betrachteten Systems hat.*

Die Gruppe der Stakeholder kann variieren, d.h. es sind nicht in jedem Projekt dieselben Gruppen vertreten. Mögliche Stakeholderrollen sind: Gesetzgeber, Käufer, Produktdesigner, Entwickler, Tester, usw. Die Stakeholder haben alle einen unterschiedlichen Hintergrund von Fachwissen in unterschiedlichen Fachbereichen. Deshalb ist es umso wichtiger verständliche Anforderungen zu schreiben, um den in Definition 2 beschriebenen Konsens zwischen allen Stakeholdern zu erreichen.

In Definition 2 wurde bereits der Begriff der Anforderung erwähnt. Bisher wurde der Begriff allerdings noch nicht detailliert ausgeführt. In Definition 4 ist die Definition³ des IEEE Standards 610.12-1990 (“IEEE Standard Glossary of Software Engineering Terminology”) aufgeführt.

Definition 4 (Anforderung [IEE90]) *Eine Anforderung ist:*

- (1) *Eine Bedingung oder Fähigkeit, die vom Benutzer (Person oder System) zur Lösung eines Problems oder zur Erreichung eines Ziels benötigt wird.*
- (2) *Eine Bedingung oder Fähigkeit, die ein System oder Teilsystem erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen.*
- (3) *Eine dokumentierte Repräsentation einer Bedingung oder Eigenschaft gemäß (1) oder (2).*

³Übersetzung der englischen Norm in der Fassung von [Rup14] und [PR15].

„Anforderungen wirken unmittelbar auf den Systementwicklungsprozess ein und bilden die Basis für viele weitere Schritte“ [Rup14]. Sie dienen außerdem als Kommunikationsgrundlage zwischen Anforderungsautor und Entwickler. Anforderungen stellen Vorgaben für die Softwareentwicklung dar und bilden die Testbasis, d.h. aus den Anforderungen, die Funktionen beschreiben, werden Testfälle abgeleitet, durch welche die entsprechenden Anforderungen validiert werden können.

Anforderungen können in unterschiedliche Arten eingeteilt werden, je nachdem ob sie eine Funktion beschreiben, die Qualität eines Produktes oder Randbedingungen. Auf die unterschiedlichen Arten von Anforderungen wird in Abschnitt 2.2.1 eingegangen.

2.2.1 Arten von Anforderungen

Laut [PR15], [Ebe12] und [Poh07] kann zwischen drei Arten von Anforderungen unterschieden werden: funktionale Anforderungen, Qualitätsanforderungen und Randbedingungen. Diese drei Arten von Anforderungen werden im Folgenden näher beschrieben.

Definition 5 (Funktionale Anforderung [PR15]) *Eine funktionale Anforderung ist eine Anforderung bezüglich des Ergebnisses eines Verhaltens, das von einer Funktion des Systems bereitgestellt werden soll.*

„Beispiele für solche funktionalen Anforderungen sind konkrete Funktionen, Ablaufbeschreibungen und Szenarien, die angeben, wie ein System auf bestimmte Eingangsgrößen oder Eingaben zu reagieren hat“ [Ebe12].

Beispiel 1 *Wenn Zustand Z eintritt, muss Signal A an System 1 gesendet werden.*

In Beispiel 1 ist eine Anforderung zu sehen, welche in die Kategorie der funktionalen Anforderungen fällt. Es ist eine konkrete Funktion des Systems beschrieben, die Anforderung enthält eine Eingangsgröße und die Reaktion des Systems ist ebenfalls beschrieben.

Definition 6 (Qualitätsanforderung [Poh07]) *Eine Qualitätsanforderung definiert eine qualitative Eigenschaft des gesamten Systems, einer Systemkomponente oder einer Funktion.*

Qualitätsanforderungen beziehen sich in den meisten Fällen „auf die Performanz, die Verfügbarkeit, die Zuverlässigkeit, die Skalierbarkeit oder die Portabilität des betrachteten Systems“ [PR15]. [Ebe12] nennt neben Verfügbarkeit und Zuverlässigkeit auch die folgenden Merkmale einer Qualitätsanforderung: Wartbarkeit, funktionale Sicherheit und Informationssicherheit.

Beispiel 2 *Die Übermittlung der Daten darf nicht länger als 2 Sekunden dauern.*

Ein Qualitätsanforderung ist in Beispiel 2 zu sehen. In diesem Beispiel wird die Performanz des Systems beschrieben, also eine Qualitätseigenschaft des Systems.

Definition 7 (Randbedingung [PR15]) *Eine Randbedingung ist eine Anforderung, die den Lösungsraum jenseits dessen einschränkt, was notwendig ist, um die funktionalen Anforderungen und die Qualitätsanforderungen zu erfüllen.*

„Rahmenbedingungen sind für die Projektbeteiligten schwer oder gar nicht veränderbar. Sie schränken entweder den Entwicklungsprozess oder das zu entwickelnde System ein“ [Poh07]. Sie stellen Vorschriften dar, die beachtet werden müssen. [Ebe12] nennt hierfür Beispiele wie Kosten, Geschäftsprozesse und Gesetze. In Beispiel 3 ist eine Anforderung, die eine Randbedingung darstellt, zu sehen.

Beispiel 3 *Die zu entwickelnde Anwendung muss unter Windows 10 lauffähig sein.*

Eine weitere verbreitete Klassifikation von Anforderungen stellt die Unterscheidung zwischen funktionalen und nicht-funktionalen Anforderungen dar. Jedoch werden diese Begriffe in der Literatur unterschiedlich ausgelegt: Laut [Poh07] und [Ebe12] werden Qualitätsanforderungen auch nicht-funktionale Anforderungen genannt. Laut [Rup14] hingegen stellt jede Anforderung, die keine funktionale Anforderung ist, eine nicht-funktionale Anforderung dar; d.h. Qualitätsanforderungen und Randbedingungen können auch unter dem Begriff der nicht-funktionalen Anforderungen zusammengefasst werden.

2.2.2 Dokumentation von Anforderungen

In Abschnitt 2.2.1 wurden die unterschiedlichen Arten von Anforderungen beschrieben. Diese sollen nicht nur in den Köpfen der Anforderungsautoren existieren, sondern müssen dokumentiert werden, damit der Entwickler des beschriebenen Systems weiß, welche Funktionen er entwickeln soll und damit Testfälle für die Validierung abgeleitet werden können. „Das Ziel der Dokumentationsaktivität ist es, Anforderungen gemäß der Dokumentationsvorschriften zu dokumentieren oder gemäß der Spezifikationsvorschriften zu dokumentieren“ [Poh07].

Anforderungen werden in sogenannten Anforderungsspezifikationen (auch *Anforderungsdokumente* genannt) dokumentiert (vgl. Definition 8).

Definition 8 (Anforderungsspezifikation [PR15]) *Eine Anforderungsspezifikation ist eine systematisch dargestellte Sammlung von Anforderungen (typischerweise für ein System oder eine Komponente), die vorgegebenen Kriterien genügt.*

„Im deutschen Sprachraum haben sich zwei Typen von Anforderungsdokumenten etabliert: das Lastenheft [...] und das Pflichtenheft“ [Poh07]. Lastenhefte werden vom Auftraggeber für den Auftragnehmer geschrieben. Im Lastenheft sind alle Forderungen an das zu entwickelnde System sowie die Randbedingungen enthalten. „In der Regel beschreibt das Lastenheft also *was* und *wofür* etwas gemacht werden soll“ [PR15]. Das Pflichtenheft wird vom Auftragnehmer erstellt und enthält die erarbeiteten Realisierungsvorgaben und beschreibt die Umsetzung des Lastenhefts [DIN97].

Für die Dokumentation von Anforderungen können unterschiedliche Notationsarten verwendet werden, welche sich im Grad der Formalität unterscheiden. Beispiele für Notationsarten sind endliche Automaten, Prädikatenlogik und die natürliche Sprache. In die Kategorie der formalen sowie semi-formalen Notationen fallen die modellbasierten Spezifikationen, die natürliche Sprache stellt eine informelle Notationsart dar (vgl. Tabelle 3). Da die natürliche Sprache und die Dokumentation mittels Modellen die am häufigsten genutzten Dokumentationsarten sind [AWK13], [KNL14], werden deren Vor- und Nachteile im Folgenden detaillierter diskutiert.

Formalisierungsgrad	Beispiel
formal (operational, ausführbar)	endlicher Automat Statechart Petri-Netz
formal (mathematisch)	Prädikatenlogik
semi-formal	Klassendiagramm Datenflussdiagramm Sequenzdiagramm
informell	natürliche Sprache

Tabelle 3: Klassifikation von Notationen nach Formalisierungsgraden nach [Mar03]

Modellbasierte Dokumentation von Anforderungen ist eine weit verbreitete Technik. Laut [KNL14] werden 33% der Anforderungen heutzutage mittels UML spezifiziert. „Die Unified Modeling Language [...] gilt im Software Engineering als Standard-Modellierungssprache“ [Sch11]. Diese semi-formale Notation bringt Vor- und Nachteile mit sich. Semi-formale

Notationen sind kompakter als die natürliche Sprache. „Zudem bieten konzeptuelle Modelle aufgrund ihres höheren Grades an Formalität einen höheren Grad der Eindeutigkeit (d.h. weniger Möglichkeiten zur Interpretation) als natürliche Sprache“ [PR15]. Der Autor benötigt zum Einsatz dieser Technik allerdings Modellierungskennntnisse, die auch der Leser haben muss.

Die Mehrheit der Anforderungen wird auch heute noch mittels natürlicher Sprache formuliert [KNL14]. Der Vorteil der natürlichen Sprache ist, dass die Autoren keine Notation erlernen müssen und Sachverhalte mit natürlicher Sprache vergleichsweise einfach ausgedrückt und verstanden werden können. Die natürliche Sprache ist außerdem flexibel, und man kann mit ihr abstrakte und konkrete Dinge beschreiben [Bal09]. Der Nachteil ist jedoch, dass natürliche Sprache inhärent mehrdeutig ist [Poh07]. Vage Ausdrücke, fehlende Information, Widersprüche und andere sprachliche Defekte können dazu führen, dass der einheitliche Konsens zwischen Auftraggeber und Auftragnehmer nicht gegeben ist.

Bei den Anforderungen, welche in dieser Arbeit analysiert werden, handelt es sich um natürlichsprachliche Anforderungen aus dem Automobilkontext (vgl. hierzu Abschnitt 1.4).

2.3 Testen in der Softwareentwicklung

Die rechte Seite des V-Modells stellt die Testaktivitäten dar. „Beim Testen wird die zu prüfende Software auf einem Rechner zur Ausführung gebracht, und es muss festgestellt werden, ob die Software sich entsprechend den Anforderungen verhält oder nicht“ [SL12]. Hierbei muss grundsätzlich beachtet werden, dass durch Testen keine Fehlerfreiheit nachgewiesen werden kann, sondern lediglich vorhandene Fehler bzw. Fehlerwirkungen aufgedeckt werden können. Testen ist folglich jede Aktivität, durch welche Fehler identifiziert werden können [HJD10].

Der Testprozess kann in einzelne Phasen unterteilt werden [O’R14]:

- (1) Testplanung,
- (2) Testfallanalyse und -design,
- (3) Testdurchführung,
- (4) Testreporting.

Bei der Testplanung wird u.a. festgelegt welche Mitarbeiter für die Testdurchführung benötigt werden, es werden die Testendekriterien definiert und es wird festgelegt auf welcher Testplattform getestet wird. Die entsprechenden Festlegungen sind dann im Testkonzept

zu dokumentieren [SL12]. Ein mögliches Testendekriterium kann beispielsweise so ausgelegt sein, dass jede funktionale und jede Qualitätsanforderung durch mindestens einen Test überprüft werden muss.

Bei der Testfallanalyse wird zunächst die Testbasis analysiert und festgelegt, ob aus den Anforderungen Testfälle abgeleitet werden können. „Testbasis, also Grundlage für die Erstellung der Tests können die Anforderungsdokumente, die Architekturdokumente, die Ergebnisse der Risikoanalyse oder auch weitere Dokumente sein, die im Rahmen der Softwareentwicklung genutzt bzw. erstellt werden“ [SL12]. Dieser Schritt wird in Abschnitt 2.3.1 ausführlich behandelt. Außerdem werden in der zweiten Phase die Testfälle unter Anwendung der im Testkonzept definierten Testmethoden abgeleitet und dokumentiert, typischerweise in einer Testspezifikation.

In der dritten Phase werden die Testfälle ausgeführt, um Fehlerwirkungen nachzuweisen. Hierbei werden die Soll- mit den Ist-Ergebnissen verglichen. „Tritt bei der Testausführung ein Unterschied zwischen dem tatsächlichen und dem vorausgesagten Ergebnis auf, ist bei der Auswertung der Testprotokolle zu entscheiden, ob tatsächlich eine Fehlerwirkung vorliegt“ [SL12]. Eine Abweichung des Ist-Ergebnisses vom Soll-Ergebnis kann unterschiedliche Ursachen haben: entweder ist eine Funktion des Testobjekts fehlerhaft, der Testfall selbst wurde fehlerhaft abgeleitet oder die Vergleichsoperation zwischen Ist- und Soll-Ergebnis ist fehlerhaft. Die Ergebnisse der Tests werden dokumentiert.

In der vierten Phase wird nach Abschluss der Testaktivitäten überprüft, ob die Testendekriterien erreicht sind oder ob weiter getestet werden muss.

Von Relevanz für diese Arbeit ist vor allem die zweite Phase des Testprozesses, in der es darum geht, zunächst die Testbasis zu analysieren und anschließend aus den Anforderungen der Anforderungsspezifikation Testfälle abzuleiten, die die beschriebenen Funktionalitäten validieren.

2.3.1 Testbasisanalyse

Bei der Testbasisanalyse werden die Anforderungen aus den für den Test relevanten Spezifikationen einem Review unterzogen. Die Testbasis ist aus dem Blickwinkel zu analysieren, „ob die zu verwendenden Dokumente ausreichend detailliert sind, um daraus Testfälle abzuleiten“ [SL12]. Die Testbasisanalyse dient also dazu den nächsten Schritt des Testprozesses vorzubereiten.

Testbarkeitsbewertungen werden größtenteils manuell durchgeführt: der Reviewer liest die Anforderungen und bewertet die Anforderungen nach Kriterien, die mit der Testbarkeit zusammenhängen. Er kann beispielsweise mit einer Checkliste arbeiten und für jede An-

forderung angeben, ob diese die geforderten Kriterien erfüllt. Mögliche Kriterien können beispielsweise die Verständlichkeit oder die Messbarkeit einer Anforderung sein.

Manuelle Reviews sind sehr zeitintensiv und garantieren nicht, dass alle Fehler gefunden werden. Es ist sinnvoll, die manuellen Analysen durch werkzeuggestützte Prüfungen zu erweitern. Für die Durchführung statischer Analysen existieren Werkzeuge, welche automatisiert Hinweise auf mangelnde Verständlichkeit liefern. Jedoch fokussiert sich keines der auf dem Markt verfügbaren Werkzeuge speziell auf das Thema Testbarkeit (vgl. Abschnitt 5.5).

Bei der Durchführung der Testbisanalyse müssen die Kriterien, welche einen Einfluss auf die Testbarkeit einer Anforderung haben, klar definiert sein. Jedoch gibt es keine Festlegung dazu, welche Kriterien eine Anforderung erfüllen muss, damit sie als testbar gilt. Es existieren allerdings Qualitätskriterien für Anforderungen, welche stark mit der Testbarkeit in Verbindung stehen. Diese werden in Abschnitt 2.3.2 präsentiert.

2.3.2 Qualitätskriterien für Anforderungen und Anforderungsdokumente

In der Literatur zum Requirements Engineering werden eine große Menge an Qualitätskriterien für Anforderungen und Anforderungsdokumente aufgelistet. Qualitätskriterien geben an, welche Eigenschaften eine Anforderung und/oder ein Anforderungsdokument erfüllen muss, um als qualitativ gut zu gelten.

Die Auflistung von Qualitätskriterien in der Literatur zum Requirements Engineering variiert. In den einzelnen Werken unterscheiden sich die Qualitätskriterien sowohl in der Anzahl als auch in der Benennung. Im IEEE-Standard 830 [IEE98] werden beispielsweise acht Qualitätskriterien definiert: *Bewertbarkeit*, *Eindeutigkeit*, *Konsistenz*, *Korrektheit*, *Modifizierbarkeit*, *Prüfbarkeit*, *Vollständigkeit* und *Verfolgbarkeit*. In [Ebe12] werden hingegen noch drei weitere Qualitätskriterien vorgeschlagen: *Notwendigkeit*, *Realisierbarkeit* und *Verständlichkeit*.

In [Kne06] wurde eine Übersicht über die Qualitätskriterien im Requirements Engineering erstellt. Hierfür wurden 19 Lehrbücher und Veröffentlichungen aus dem Requirements Engineering gesichtet und die Qualitätskriterien in einer Übersicht zusammengefasst. Insgesamt wurden 33 Qualitätskriterien aus den Lehrbüchern und Veröffentlichungen extrahiert. In der Übersicht von [Kne06] werden auch synonyme Benennungen von Qualitätskriterien und deren englische Bezeichnungen aufgeführt. Diese Zusammenfassung stellt eine gute Übersicht über die gängigen Qualitätskriterien im Requirements Engineering dar.

Die Qualitätskriterien sind nicht unabhängig voneinander zu betrachten. In [Kne06] wird darauf hingewiesen, dass zwischen Qualitätskriterien “ursächliche Zusammenhänge” und “tendenzielle Widersprüche” bestehen können. Ein ursächlicher Zusammenhang besteht beispielsweise zwischen den beiden Qualitätskriterien *Testbarkeit* und *Eindeutigkeit*. „Testbarkeit setzt Eindeutigkeit voraus: Anforderungen, die Interpretationsspielraum offen lassen, sind nicht testbar, da die Abnahmekriterien nicht klar ableitbar sind“ [Kne06]. Eine Anforderung kann nur dann testbar sein, wenn sie eindeutig ist. „Ein starker tendenzieller Widerspruch liegt vor, wenn keines der beiden betrachteten Kriterien vollständig erfüllt werden kann, ohne die Erfüllung des jeweils anderen Kriteriums in irgendeiner Weise zu beeinträchtigen“ [Kne06]. Ein Beispiel für einen solchen Widerspruch sind die Qualitätskriterien *Verständlichkeit* und *Redundanzfreiheit*. Wiederholung wichtiger Information steigert die Verständlichkeit, wirkt sich jedoch negativ auf das Qualitätskriterium *Redundanzfreiheit* aus.

Der Fokus der vorliegenden Arbeit liegt auf dem Qualitätskriterium *Testbarkeit*. Es stellt das Oberkriterium dar, welchem weitere Kriterien untergeordnet werden können. Zunächst wird daher das Qualitätskriterium der *Testbarkeit* erläutert.

Qualitätskriterium Testbarkeit „Testbarkeit einer Anforderung bedeutet, dass sich aus der Anforderungsbeschreibung effektive Testfälle ableiten lassen, nach denen eindeutig entschieden werden kann, ob diese Anforderung vom erstellten System erfüllt wird oder nicht“ [Pus02].

Aus dem Qualitätskriterium der *Testbarkeit* lassen sich eine Reihe von Qualitätskriterien ableiten, welche mit dem Qualitätskriterium *Testbarkeit* ursächlich zusammenhängen, d.h. nur wenn diese Kriterien erfüllt sind, ist auch das Kriterium der *Testbarkeit* erfüllt. Ursächliche Zusammenhänge zur Testbarkeit haben die folgenden Qualitätskriterien: *Atomarität*, *Eindeutigkeit*, *Klassifizierbarkeit bezüglich der juristischen Verbindlichkeit*, *Konsistenz/Widerspruchsfreiheit*, *Korrektheit*, *Präzision*, *Verständlichkeit* und *Vollständigkeit*. All diese Qualitätskriterien finden sich in den einzelnen Kriterien und Phänomenen des in Kapitel 4 definierten Qualitätsmodells für testbare Anforderungen wieder.

Es gibt noch weitere Qualitätskriterien, welche in [Kne06] beschrieben sind. Diejenigen, die nicht ursächlich mit dem Qualitätskriterium *Testbarkeit* zusammenhängen, werden in der vorliegenden Arbeit nicht betrachtet.

Für die Qualitätskriterien, die direkt mit dem Qualitätskriterium der Testbarkeit zusammenhängen, werden nun ausgewählte Definitionen aus der Literatur aufgeführt:

Qualitätskriterium Atomarität „Ein Inhaltselement ist hinreichend atomar, wenn es nicht sinnvoll in mehrere Einträge aufgespaltet werden kann“ [Mel00].

Qualitätskriterium Eindeutigkeit „Die Eindeutigkeit ist hoch, wenn die Anforderungen so formuliert sind, daß sie für alle zu erwartenden Leser die gleiche Bedeutung haben“ [Mel00].

Qualitätskriterium Konsistenz/Widerspruchsfreiheit „Eine Anforderung ist in sich widerspruchsfrei, wenn Aussagen innerhalb der Anforderung nicht im Widerspruch zueinander stehen. Eine Anforderung ist widerspruchsfrei, wenn sie zu keiner anderen Anforderung innerhalb des Projekts bzw. im gesamten Unternehmen im Widerspruch steht“ [SS03].

Qualitätskriterium Korrektheit „Unter Korrektheit versteht man, dass das beschriebene Verhalten exakt den Vorstellungen derjenigen Interessenträger entspricht, auf die das Anforderungsdokument zurückgeht bzw. von denen es in Auftrag gegeben wurde“ [Pus02].

Qualitätskriterium Präzision „Die Präzision ist hoch, wenn die Anforderungen keine ungenauen Aussagen enthalten und wenn, soweit möglich, numerische Werte verwendet werden“ [Mel00].

Qualitätskriterium Realisierbarkeit „Eine Anforderung ist dann realisierbar, wenn sie mit dem gegebenen Stand der Technik und den Randbedingungen des Projekts z.B. bezüglich der Ressourcen und Zeitpläne umsetzbar ist“ [SS03].

Qualitätskriterium Verständlichkeit „Eine Anforderung ist verständlich formuliert, wenn der zu erwartende Leserkreis die Aussage der Anforderung in einer der Komplexität der Anforderung angemessenen Zeit und mit den referenzierten Informationen verstehen kann“ [SS03].

Qualitätskriterium Vollständigkeit „Inhaltliche Vollständigkeit eines Anforderungsdokumentes bedeutet, dass alle relevanten Anforderungen, funktionale wie auch nicht-funktionale, jeweils vollständig beschrieben sind“ [Pus02].

Qualitätskriterien geben an, welche Eigenschaften eine Anforderung oder ein Anforderungsdokument erfüllen muss, um als qualitativ gut zu gelten. Wird eines der Qualitätskriterien verletzt, mindert dies die Qualität der Anforderung. In Kapitel 4 wird ein Qualitätsmodell für testbare Anforderungen definiert, welches auf den in diesem Abschnitt beschriebenen Kriterien basiert. Diese Kriterien leiten sich aus dem Qualitätskriterium *Testbarkeit* ab.

2.4 Zusammenfassung

Der Textqualität von Anforderungen in der Softwareentwicklung kommt ein hoher Stellenwert zu, da Anforderungen die Kommunikationsgrundlage zwischen Auftraggeber und Auftragnehmer bilden. Anforderungen müssen zum einen eine hohe sprachliche Qualität haben, damit zwischen Auftraggeber und Auftragnehmer ein einheitlicher Konsens besteht und die Funktionen der zu entwickelnden Komponenten oder des zu entwickelnden Systems korrekt realisiert werden. Die Textqualität von Anforderungen spielt jedoch auch in den Testaktivitäten eine entscheidende Rolle: da die Aufwendungen für die Testaktivitäten mittlerweile 25-50% der Entwicklungszeit und der Entwicklungskosten einnehmen, kommt dieser Phase eine besonders hohe Bedeutung zu. Aus den natürlichsprachlichen Anforderungen werden Testfälle abgeleitet, mit welchen geprüft wird, ob die beschriebenen Funktionalitäten korrekt realisiert wurden. Um die Ableitung der korrekten Testfälle garantieren zu können, müssen Anforderungen bestimmten Kriterien genügen. Diese Kriterien werden in Kapitel 4 der vorliegenden Arbeit herausgearbeitet.

3 Textqualität aus der Sicht der angewandten Sprachwissenschaft

Ausgehend von den Beiträgen und daraus abgeleiteten Forschungsfragen aus Abschnitt 1.3 wird in diesem Kapitel die Problematik der Textverständlichkeit aus der Sicht der angewandten Sprachwissenschaft beschrieben. Im ersten Teil dieses Kapitels wird auf ausgewählte aus der Verständlichkeitsforschung stammende Verständlichkeitsindices und Verständlichkeitsmodelle eingegangen (Abschnitt 3.1). Aus den Verständlichkeitsmodellen der Verständlichkeitsforschung wurden im Bereich der technischen Dokumentation Regeln abgeleitet, welche dem Autor dabei helfen können verständlichere Texte zu produzieren. Eine solche Ableitung von Regeln hat beispielsweise *Der deutsche Fachverband für Technische Kommunikation tekom* vorgenommen und eine Leitlinie für das Verfassen von verständlicher technischer Dokumentation verfasst. Diese und eine weitere Leitlinie werden in Abschnitt 3.2 vorgestellt.

3.1 Verständlichkeitsforschung

In diesem Abschnitt werden Modelle präsentiert, welche im Bereich der Verständlichkeitsforschung entworfen wurden, um einen Text danach zu bewerten, wie verständlich dieser ist. In Abschnitt 3.1.1 wird auf sogenannte Verständlichkeitsformeln eingegangen und deren Rolle im Bezug auf die Textsorte der natürlichsprachlichen Anforderungen beschrieben. In Abschnitt 3.1.2 werden die zwei wohl bekanntesten Verständlichkeitsmodelle präsentiert: das Hamburger Verständlichkeitsmodell und das Verständlichkeitsmodell von Groeben (auch Heidelberger Konstrukt genannt).

3.1.1 Verständlichkeitsformeln

In der Verständlichkeitsforschung werden Verständlichkeitsformeln (= Verständlichkeitsindices) präsentiert, durch welche sich die Verständlichkeit eines Textes anhand von Metriken berechnen lässt. Bereits 1948 wurde in [Fle48] die *Reading-Ease-Formel*, eine der bekanntesten Lesbarkeitsformeln, präsentiert. Die Reading-Ease-Formel setzt sich aus den Merkmalen der durchschnittlichen Satzlänge und Wortlänge (Anzahl der Silben) zusammen und kann auf englischsprachige Texte angewendet werden:

$$\text{R.E. ("reading ease")} = 206,835 - (84,6 * (\text{Silben/Wörter})) - (1,015 * (\text{Wörter/Sätze}))$$

Die Formel liefert einen numerischen Wert, welcher sich durch eine Skala interpretieren lässt. Diese Skala aus [Fle48] ist in Tabelle 4 zu sehen.

Reading-Ease-Score	Lesbarkeit
0-30	Sehr schwierig
30-50	Schwierig
50-60	Ziemlich schwierig
60-70	Normal
70-80	Ziemlich einfach
80-90	Einfach
90-100	Sehr einfach

Tabelle 4: *Reading-Ease-Scores* nach [Fle48]

Eine weitere Lesbarkeitsformel, der *Flesch-Kincaid-Grade-Level*, gibt die Anzahl an Schuljahren an, die ein Leser absolviert haben sollte, um einen gegebenen Text fließend lesen zu können [Sch16]. Auch hier liegt der Fokus auf den Merkmalen der Satz- und Wortlänge.

$$\text{Flesch-Kincaid-Grade-Level} = (0,39 \text{ Wörter/Sätze}) + (11,8 \text{ Silben/Wörter}) - 15,59$$

„Da im Englischen häufiger einsilbige Wörter verwendet werden als im Deutschen, sind die Reading Ease Formel sowie der Flesch-Kincaid-Grade-Level nicht unmittelbar auf deutschsprachige Texte übertragbar“ [Sch16]. Eine Adaptierung der Formeln auf deutschsprachige Texte wird in [Ams78] präsentiert.

Diese Art der Messung der Verständlichkeit eines Textes beruht lediglich auf quantitativen Merkmalen von dessen Oberflächenstruktur. Syntaktische, semantische und inhaltliche Kriterien werden in solchen Formeln vernachlässigt. Die Formeln bieten außerdem keine Anleitung dafür wie man Texte besser schreiben könnte (abgesehen von Satz- und Wortlänge). Verschiedene Erweiterungen solcher Lesbarkeitsformeln wurden seither vorgeschlagen, meist aus der Sicht von Sprachlernern; diese berücksichtigen auch die Komplexität von Konzepten [SC01], Textkohärenz oder das Verhältnis zwischen “einfachen” und “schwierigen” Konjunktionen und Adverbien [Ben05]. All diese Formeln zielen auf eine quantifizierende Beurteilung von Texten ab, bieten jedoch keine direkte Empfehlung zur Verbesserung der Lesbarkeit. Die Verständlichkeit einer Anforderung hängt von vielen Faktoren ab, welche allein durch Lesbarkeitsformeln nicht abgedeckt werden können. Es bedarf einer umfangreicheren Umschreibung der Faktoren, welche einen Einfluss auf die Verständlichkeit und somit auch auf die Testbarkeit einer Anforderung haben können.

3.1.2 Verständlichkeitsmodelle

Das *Hamburger Verständlichkeitsmodell* [LST74] und die Theorie von Groeben [Gro72] sind Modelle, die die Beschränkungen, welche die Verständlichkeitsformeln aufweisen, zu überwinden versuchen. Das Hamburger Verständlichkeitsmodell beschreibt „ein mehr induktives Vorgehen, das von möglichen Textmerkmalen aus die relevanten Dimensionen aufbaut“ [Gro82]. Das deduktive Verfahren von Groeben leitet die Dimensionen der Verständlichkeit aus der Theorie der Verarbeitungsprozesse her. „Dabei wurden aus theoretischen Ansätzen der Sprachpsychologie, kognitiven Lerntheorie und Motivationspsychologie die relevanten Dimensionen hypothetisch postuliert und zugehörige Textmerkmale abgeleitet“ [Gro82]. Die beiden Modelle werden in den folgenden Abschnitten vorgestellt.

3.1.2.1 Das Hamburger Verständlichkeitsmodell

Das *Hamburger Verständlichkeitsmodell* „basierte auf einem empirisch-induktiven Vorgehen, um Dimensionen verständlicher Texte zu generieren. Im Zentrum der Arbeit standen Ratings, in denen Leseexpertinnen und Leseexperten Sachtexte auf der Grundlage von unterschiedlichen Kriterien als mehr oder weniger verständlich einschätzten“ [Sch16]. Aus den Ergebnissen dieser Studie wurden insgesamt vier verschiedene Dimensionen der Verständlichkeit abgeleitet: *Einfachheit*, *Gliederung-Ordnung*, *Kürze-Prägnanz* und *Zusätzliche Stimulanz*.

Zur Veranschaulichung der Messung der Verständlichkeit anhand der genannten Dimensionen, wird für jede Dimension eine Skala festgelegt, welche relevante Beurteilungskriterien enthält. In Tabelle 10 ist die Skala für das Merkmal *Kürze-Prägnanz* zu sehen. Für jedes Merkmal existiert eine solche Skala und für jedes Merkmal ist ein Bereich definiert, welcher als optimal gilt.

Kürze-Prägnanz	+ 2 + 1 0 - 1 - 2	Weitschweifigkeit
zu kurz		zu lang
aufs Wesentliche beschränkt		viel Unwesentliches
gedrängt		breit
aufs Lernziel konzentriert		abschweifend
knapp		ausführlich
jedes Wort ist notwendig		vieles könnte weg

Abbildung 10: Skala des Verständlichkeitsmerkmals Kürze-Prägnanz nach [LST74]

Das Merkmal der *Kürze-Prägnanz* besagt, dass die zu übermittelnde Information mit wenigen, aber dennoch ausreichend Wörtern ausgedrückt werden sollte. Diese Forderung entspricht inhaltlich (und in ihrer Allgemeinheit) der Grice'schen Maxime der Quantität und zu einem gewissen Maße auch der Maxime der Modalität (vgl. [Gri75]).

Um das Merkmal *Einfachheit* zu erfüllen, soll der Autor kurze Sätze und möglichst bekannte Wörter verwenden, Fachwörter erklären und Sachverhalte konkret und anschaulich darstellen. Die Dimension der *Gliederung und Ordnung* umfasst Kriterien, die mit dem Aufbau eines Textes in Zusammenhang stehen. Der Autor soll, um das Merkmal zu erfüllen, darauf achten, dass der Gesamtaufbau seines Textes logisch ist, d.h. gedanklich nachvollziehbar und nicht durcheinander. Wichtige Inhalte sollen von eher unwichtigen Inhalten klar abgegrenzt werden und die äußere Gestaltung soll in sich stimmig sein.

Zusätzliche Anregung wird beispielsweise durch die Verwendung von Graphiken erreicht, aber auch durch konkrete Beispiele, die den Leser persönlich ansprechen.

Einfachheit + 2	Gliederung-Ordnung + 2
Kürze-Prägnanz 0 oder + 1	Zusätzliche Stimulanz 0 oder + 1

Abbildung 11: Beurteilungsfenster der Verständlichkeitsmerkmale nach [LST74]

Für jede Dimension der Verständlichkeit existiert eine Skala wie in Abbildung 10 und für jede Dimension gibt es einen optimalen Bereich, der bei der Erzeugung von Texten beachtet werden sollte (vgl. hierzu Abbildung 11). Die Dimension *Einfachheit* liegt bei einem gut verständlichen Text im Bereich +2, die Dimension *Gliederung-Ordnung* ebenfalls. Das Optimum der Dimension *Kürze-Prägnanz* liegt im mittleren Bereich, da zu knappe und zu weitschweifige Texte jeweils das Leseverständnis erschweren können. Das Optimum der Dimension *Zusätzliche Stimulanz* ist in Abhängigkeit von den anderen Dimensionen zu betrachten. „Für die anregenden Zusätze eines Textes galt außerdem, dass diese nur eine positive Wirkung auf das Textverständnis hatten, wenn der Text ausreichend gegliedert war. Jedoch wird diese Wechselwirkung nicht im Beurteilungsfenster berücksichtigt“ [Sch16]. Die Merkmale *Einfachheit* und *Gliederung-Ordnung* sind nach [LST74] die wichtigsten Dimensionen.

Friedemann Schulz von Thun fasst das Hamburger Verständlichkeitsmodell in eigenen Worten folgendermaßen zusammen: „Verständlichkeit steht nicht auf einem, sondern auf vier Beinen! Erstens, sprich einfach wie ein normaler Mensch – nicht wie ein Universitäts-

gelehrter. Zweitens, mach den Bauplan deiner Ausführung sichtbar – wie gliedert sich das Ganze und wo verläuft der rote Faden? Drittens, bring das Wichtigste auf den Punkt, mit einer prägnanten Formulierung oder einer guten Abbildung. Und viertens, mach das Ganze ein wenig lebendig: mit Metaphern, mit lebensnahen Beispielen, mit allem, was dir einfällt, um den Leser bei Stange zu halten“ [Sch15].

Die Merkmale und Beurteilungskriterien des Hamburger Verständlichkeitsmodells beruhen auf Erfahrungen und Eindrucksurteilen von Leseexperten, die im Zuge einer Studie durch Lesen verschiedener Texte und Textsorten (Schulbuchtexte, wissenschaftliche Veröffentlichungen, Gesetzestexte, usw. [Gro82]) gesammelt werden konnten. Das Modell stützt sich auf wenig Theorie und steht deshalb stark in der Kritik [Gro82], [Sch16]. Ein weiterer Schwachpunkt des Modells ist, dass es lediglich vorgibt, was Autoren beim Schreiben von Texten keinesfalls tun sollten. Alternative Formulierungen oder Regeln, die den Autoren helfen könnten verständlichere Texte zu schreiben, werden nicht bereitgestellt. Die konkrete Anwendung der eher allgemein formulierten Vorschläge ist für einen Autor schwierig umzusetzen. Weiterhin sind bestimmte Ziele des Modells für ein Fachpublikum eher irrelevant, wie beispielsweise die Erklärung von Fachausdrücken.

3.1.2.2 Das Verständlichkeitsmodell nach Groeben

Ein auf verschiedene Theorien gestütztes Modell wird in [Gro72] geliefert, welches auf sprachpsychologischen lerntheoretischen und motivationspsychologischen Ansätzen beruht. Aus diesen theoretischen Ansätzen wurden die relevanten Dimensionen der Verständlichkeit postuliert und die zugehörigen Textmerkmale abgeleitet [Gro82]. Die vier Verständlichkeitsdimensionen sind: *stilistische Einfachheit*, *semantische Redundanz*, *kognitive Strukturierung* und *konzeptueller/kognitiver Konflikt*.

Bei der stilistischen Einfachheit stützt Groeben sich auf Ergebnisse der Stil- und Lesbarkeitsforschung und auf psycholinguistische Modelle, die sich auf die Transformationsgrammatik von Noam Chomsky [Cho65] zurückführen lassen. „Dabei stehen vor allem Phänomene wie einfache ‚Kern‘-Sätze (deklarative Sätze mit Subjekt und Prädikat), aktive oder passive Formulierung, affirmative und negative Formulierung, Nominalisierungen und Adjektivierungen, Parataxen und Hypotaxen sowie deren mögliche Kombination im Zentrum der Untersuchungen“ [Gro82].

Das Merkmal der semantischen Redundanz beruht auf Modellen der Informationsverarbeitung und Informationstheorie. Dieses Merkmal basiert auf der Annahme, dass Texte mit semantischen Wiederholungen dazu beitragen, dass der Leser eines Textes dessen Inhalte einfacher behalten kann. Die Information soll somit nicht nur im Kurzzeitgedächtnis,

sondern auch im Langzeitgedächtnis gespeichert werden. Fokussiert werden Redundanzen von Subjekten, Prädikaten und Objekten [Sch16].

Zur Beschreibung der kognitiven Strukturierung bezieht sich Groeben auf Ergebnisse der kognitiven Lerntheorie von Ausubel (vgl. hierzu [Aus63]). „Nach Ausubel [...] basiert das Lernen aus Texten auf einem sinnorientierten Rezeptionslernen, wobei das sprachlich bedeutungsvolle Material in die kognitive Wissensstruktur der Leserinnen und Leser integriert wird.“ [Sch16]. Das Merkmal der kognitiven Strukturierung umfasst Empfehlungen dazu, dass Autoren viele Beispiele verwenden sollen, Vorstrukturierungen vornehmen (zum Beispiel in Form eines Abstracts) und durch Verwendung von Überschriften und Randbemerkungen die Texte abrunden.

Die vierte Verständlichkeitsdimension ist der kognitive Konflikt und basiert auf der Theorie der Neugiermotivation von Berlyne (vgl. hierzu [Ber74]). Diese Dimension beschreibt die Wichtigkeit des Motivations-Aspekts eines Textes. Ein Text soll laut dieses Merkmals die Leser dazu motivieren, selbständig nach Problemlösungen zu suchen. „Die Gegenstandscharakteristika, die [...] eine Wissensneugier auslösen, nennt Berlyne ‚kollative‘ Variablen (weil sie z.B. mit bestimmten Voreinstellungen, Erwartungen etc. verglichen werden). Die wichtigsten kollativen Variablen sind Neuheit, Inkongruität, Überraschung, Perplexität, Zweifel und Konfusion etc.“ [Sch16].

Nach Groeben sind die Merkmale *kognitive Strukturierung* und *kognitiver Konflikt* in Kombination die wichtigsten Merkmale. Die beiden Merkmale fasst Groeben nochmals unter dem Merkmal der *inhaltlichen Strukturierung* zusammen [Gro82]. Die semantische Redundanz ist nur in Kombination mit der stilistischen Einfachheit verständnisfördernd. Im Hamburger Verständlichkeitsmodell hingegen kommt der stilistischen Einfachheit die größte Bedeutung zu. Insgesamt kommt Groeben zu dem Ergebnis, dass eine mittlere und nicht die maximale Verständlichkeit aus motivations- und kognitionspsychologischen Gründen am besten ist [CG99]. „Man sollte die kognitive Gliederung nicht so weit treiben, dass die Textverarbeitung keinen oder nur einen sehr geringen kognitiven Anspruch an den Rezipienten stellt. Vielmehr ist es sinnvoll, auch die konfliktevozierende Darstellung zu integrieren, allerdings nur so weit, dass eine eindeutige konzeptuelle Strukturierung des Lesers/Lerners aufrechterhalten bleibt“ [Gro82].

In Groebens Verständlichkeitsmodell werden wie auch im Hamburger Verständlichkeitsmodell vier Dimensionen der Verständlichkeit vorgestellt. Die Relevanz der einzelnen Dimensionen unterscheidet sich in beiden Modellen. Im Gegensatz zum Hamburger Verständlichkeitsmodell werden im Verständlichkeitsmodell von Groeben zum Teil sprachliche Phänomene erwähnt, welche einen Einfluss auf die Verständlichkeit eines Textes haben

können. Es werden auch Vorschläge zur Strukturierung eines Textes gegeben und es wird zusätzlich auf den Stil eines Textes eingegangen. Jedoch bietet auch dieses Modell kein konkretes Regelwerk, wie ein Autor verständliche Texte schreiben kann.

3.2 Leitlinien für die Erstellung technischer Dokumentation

Konkrete Regelwerke für verständliche Texte bieten Leitfäden zur *Technischen Dokumentation*. „Technische Dokumentation (=TD) ist ein Sammelbegriff für Unterlagen zu technischen Geräten“ [Juh05]. Es handelt sich im Allgemeinen um Bedienungsanleitungen für einen Endverbraucher. „Das primäre Ziel von TD ist es, dem Leser einen Zugang zu Fachinformationen zu ermöglichen, der ihn in die Lage versetzt, bestimmte Handlungen auszuführen“ [DZ11]. Im Bereich der technischen Dokumentation spielen folglich gut lesbare und eindeutige Texte eine entscheidende Rolle für die Übermittlung von eindeutiger, verständlicher Information.

Es existieren Leitfäden für das Verfassen von technischer Dokumentation. Ein Beispiel für einen solchen Leitfaden ist die *tekomp*-Leitlinie *Regelbasiertes Schreiben – Deutsch für die Technische Kommunikation*, „das umfassendste Regelwerk für die deutsche Sprache“ [SL15]. Ausgehend von den Ergebnissen der Verständlichkeitsforschung leitet die *tekomp*-Leitlinie [tek11] sprachliche Phänomene ab, die einen Einfluss auf die Verständlichkeit eines Textes haben können, und stellt Regeln auf, die beim Verfassen von technischer Dokumentation beachtet werden sollen. Diese Regeln „werden bereits seit vielen Jahren von unterschiedlichen wissenschaftlichen Gruppen entwickelt, basieren auf wissenschaftlich-empirischen Erkenntnissen der Verständlichkeitsforschung und sind in Technischen Redaktionen großer Organisationen seit langem erprobt“ [SL15].

Die *tekomp*-Leitlinie von 2011 umfasst Textregeln, Satzregeln und Wortregeln. Die Textregeln umfassen Empfehlungen dazu, wie Überschriften idealerweise aufgebaut sein sollen (z.B. dass keine ganzen Sätze in Überschriften verwendet werden sollen), sowie Empfehlungen zur Strukturierung von Indizes, Querverweisen und Glossaren. Die Satzregeln lassen sich in fünf Kategorien einteilen: Vermeidung von mehrdeutigen Konstruktionen, Vermeidung von unvollständigen Konstruktionen, Vermeidung von komplexen Konstruktionen, Regeln zur Wortstellung und Abfolge von Satzelementen und stilistische Regeln. Die Wortregeln sind größtenteils auf die Strukturierung von Komposita ausgerichtet, geben aber auch Empfehlungen zur einheitlichen Bildung des Genitivs und Dativs und beschreiben den Umgang mit Abkürzungen.

Die Duden-Leitlinie [Sch12] ist ein weiteres bekanntes Werk, welches Regeln für das Verfassen von technischer Dokumentation vorgibt. Diese Leitlinie lehnt sich an die Ergebnisse

und Regeln der tekomp-Leitlinie an und übernimmt einen Teil der Regeln, integriert jedoch auch eigene Regeln.

Bei der technischen Dokumentation handelt es sich um eine Schnittstellenkommunikation zwischen Experten und Laien [DZ11]. Die produzierten Texte müssen für einen Laien mit nur wenig bis gar keinem Vorwissen verständlich sein. Hilfreich bei der Produktion dieser Art von Texten sind daher Regeln aus der Wissenschaft für leichte Sprache, weil Texte in leichter Sprache schneller verarbeitet werden können und verständlicher sind [Maa15]. Ein Ziel der technischen Dokumentation ist auch mehrdeutige Strukturen zu vermeiden und stets alle für das Verständnis benötigte Information schriftlich wiederzugeben, damit ein einheitliches Verständnis vermittelt werden kann.

Den Lastenheften, welche den Untersuchungsgegenstand dieser Arbeit darstellen, liegt eine Experten-Experten-Kommunikation zugrunde. Das für das Textverständnis benötigte Vorwissen der Leser ist im Bereich des Requirements Engineering in höherem Maße vorhanden als bei Lesern von technischer Dokumentation. Folglich überschneiden sich die in der Literatur zum Requirements Engineering genannten Schreibregeln nur teilweise mit den Regeln, welche in der Literatur zur technischen Dokumentation genannt werden. Die Regeln aus der technischen Dokumentation zielen großteils auf die Verbesserung des Schreibstils und auf die Strukturierung von Texten ab. Auch mehrdeutige Strukturen und Phänomene, die dazu führen, dass wichtige Information ungenannt bleibt, gilt es in der technischen Dokumentation zu vermeiden. Die Regeln aus dem Requirements Engineering betreffen größtenteils Phänomene, die dazu führen, dass wichtige Information ungenannt bleibt, aber auch Phänomene, die zu Mehrdeutigkeiten führen können. Komplexe Strukturen spielen im Gegensatz zur technischen Dokumentation, eine eher untergeordnete Rolle (vgl. hierzu auch Abschnitt 4.7, Tabelle 5). Jedoch ist es sinnvoll, Regeln aus den Leitfäden der technischen Dokumentation auch auf natürlichsprachliche Anforderungen zu übertragen, da auch dort sprachliche Phänomene genannt werden, die zu mehrdeutigen Konstruktionen führen können.

3.3 Zusammenfassung

Im vorliegenden Kapitel wurde die Problematik der Textverständlichkeit aus Sicht der angewandten Sprachwissenschaft beschrieben. Es wurden Verständlichkeitsformeln und Verständlichkeitsmodelle präsentiert, welche verwendet werden können, um einen Text danach zu bewerten, wie verständlich dieser ist. Lesbarkeitsformeln zielen auf eine quantifizierende Beurteilung von Texten ab, bieten jedoch keine direkte Empfehlung zur Verbesserung der Lesbarkeit. Das *Hamburger Verständlichkeitsmodell* und die Theorie von Groeben sind Modelle, die die Beschränkungen, welche die Verständlichkeitsformeln auf-

weisen, zu überwinden versuchen. Im Verständlichkeitsmodell von Groeben werden zwar sprachliche Phänomene erwähnt, die einen Einfluss auf die Verständlichkeit eines Textes haben können, jedoch bietet auch das Modell kein konkretes Regelwerk, wie Autoren verständliche Texte schreiben können.

Konkrete Leitfäden für das Schreiben von verständlichen Texten bieten hingegen Leitfäden zur technischen Dokumentation. Im vorliegenden Kapitel wurde die wohl bekannteste Leitlinie vorgestellt: die *tekomp*-Leitlinie *Regelbasiertes Schreiben – Deutsch für die Technische Kommunikation*. Ausgehend von den Ergebnissen der Verständlichkeitsforschung leitet die *tekomp*-Leitlinie sprachliche Phänomene ab, die einen Einfluss auf die Verständlichkeit eines Textes haben können, und stellt Regeln auf, die beim Verfassen von technischer Dokumentation beachtet werden sollen. Ein weiteres bekanntes Werk ist die Duden-Leitlinie, welche sich an die Ergebnisse und Regeln der *tekomp*-Leitlinie anlehnt.

Da es sich bei der technischen Dokumentation um eine Schnittstellenkommunikation zwischen Experten und Laien handelt und nicht um eine Experten-Experten-Kommunikation, wie sie bei den Lastenheften vorliegt, ist es nur für ausgewählte Regeln aus den Leitfäden zur technischen Dokumentation sinnvoll, diese auf die Lastenhefte zu übertragen.

4 Entwicklung eines Qualitätsmodells für testbare Anforderungen

Basierend auf den in Abschnitt 3.2 beschriebenen Leitlinien aus der technischen Dokumentation, Regeln aus der Literatur zum Requirements Engineering und den mit der Testbarkeit zusammenhängenden Qualitätskriterien aus Abschnitt 2.3.2, werden sprachliche Phänomene identifiziert und in ein Qualitätsmodell für testbare Anforderungen integriert. Jedes der aufgelisteten Qualitätskriterien aus Abschnitt 2.3.2 ist im Qualitätsmodell vorhanden, wobei das Qualitätskriterium *Testbarkeit* die oberste Ebene des Modells bildet. Die verwendeten Literatur-Quellen zielen auf unterschiedliche Textsorten ab und sind unterschiedlicher Herkunft. Es wurden Vorschläge aus dem Bereich des Requirements Engineering betrachtet ([Rup14], [PR15], [Poh07], [Kam01]), zwei Daimler-interne Richtlinien ([Dai09a], [Dai09b]), die technische Redaktionsleitlinie der tekomp [tek11] und die an die tekomp-Leitlinie angelehnten Stil-Empfehlungen des Dudens [Sch12]. Das Qualitätsmodell trägt Phänomene, die einen Einfluss auf die Testbarkeit einer Anforderung haben können, aus den genannten Quellen zusammen, sortiert sie nach linguistischen Aspekten und bewertet sie nach Kritikalität.

In Abschnitt 4.7 sind in Tabelle 5 Zuordnungen zwischen den im Qualitätsmodell enthaltenen Phänomenen und den hier präsentierten Quellen zu sehen. Es werden nur die Zuordnungen, welche für das Qualitätsmodell für testbare Anforderungen relevant sind, aufgeführt. Des Weiteren wird eine Diskussion zu den Quellen und die Interpretation des Qualitätsmodell in Abschnitt 4.7 präsentiert.

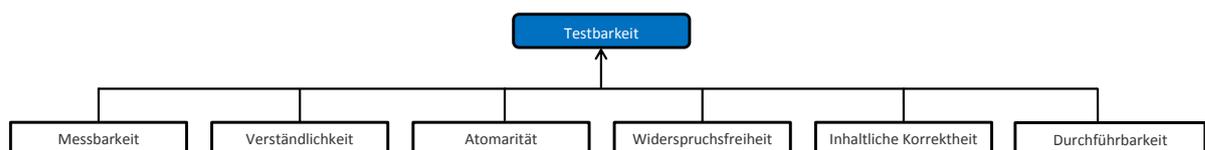


Abbildung 12: Überblick Qualitätsmodell für testbare Anforderungen

Ein Gesamtüberblick über die Struktur des Qualitätsmodells ist in Abbildung 12 zu sehen. Nach diesem Modell setzt sich die Testbarkeit einer Anforderung aus den folgenden Kriterien zusammen: *Messbarkeit*, *Verständlichkeit*, *Atomarität*, *Widerspruchsfreiheit*, *Inhaltliche Korrektheit* und *Durchführbarkeit*. Anlehnend an die Definition des Qualitätskriteriums *Testbarkeit* aus Abschnitt 2.3.2, lässt sich ableiten, dass eine Anforderung genau dann testbar ist, wenn alle Testbarkeitskriterien erfüllt sind und sich somit effektive Testfälle aus der Anforderungsbeschreibung ableiten lassen, nach welchen eindeutig entschieden werden kann, ob diese Anforderung vom erstellten System erfüllt wird.

In Abbildung 12 ist die oberste Ebene des Qualitätsmodells zu sehen. Jedes der fünf Testbarkeitskriterien beinhaltet sprachliche Phänomene.

Phänomene, die einen Einfluss auf die Testbarkeit haben, werden im Qualitätsmodell nach ihrer Kritikalität bewertet. Dabei unterscheiden wir drei Stufen, die über die Anzahl von Fällen bestimmt werden, in denen sie eines der Testbarkeitskriterien verletzen. Ein Phänomen, das bei jedem Auftreten ein gegebenes Testbarkeitskriterium verletzt, gehört zur obersten Kritikalitätsklasse. Ein Phänomen, das nur in bestimmten Kontexten ein gegebenes Testbarkeitskriterium verletzt, gehört zu der mittleren Kritikalitätsklasse. Ein Phänomen, das nur in sehr seltenen Fällen ein gegebenes Testbarkeitskriterium verletzt, gehört zu der untersten Kritikalitätsklasse. Die drei Stufen der Kritikalität werden in der graphischen Darstellung des Qualitätsmodells farblich markiert. Die Phänomene der obersten Kritikalitätsstufe sind rot markiert, die der mittleren gelb und die Phänomene der untersten Kritikalitätsstufe sind grün markiert.

Im Folgenden wird nun auf die einzelnen Testbarkeitskriterien und deren zugeordneten Phänomene detailliert eingegangen.

4.1 Testbarkeitskriterium Messbarkeit

In Abbildung 13 ist das Testbarkeitskriterium *Messbarkeit* mit einem untergeordneten sprachlichen Phänomen zu sehen. Das Testbarkeitskriterium *Messbarkeit* ist genau dann erfüllt, wenn die analysierte Anforderung, sofern nötig, eine quantifizierende Angabe enthält. Dieses Kriterium hängt mit dem sprachlichen Phänomen der *Weak-Words* zusammen, welches in Abschnitt 4.2.1.2 weiter ausgeführt wird.

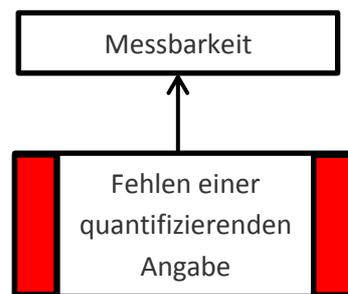


Abbildung 13: Qualitätskriterium Messbarkeit

Das Testbarkeitskriterium *Messbarkeit* setzt eine Quantifizierung oder eine numerische Angabe voraus. Fehlt eine solche Angabe, dann ist der in der entsprechenden Anforderung beschriebene Sachverhalt nicht messbar und damit auch nicht (quantitativ) testbar⁴.

⁴Nicht jede zu testende Anforderung muss messbar sein. Das Testbarkeitskriterium ist nur dann zu erfüllen, wenn eine quantifizierende Angabe für das einheitliche Verständnis nötig ist.

Ein typisches Beispiel für eine Klasse von Kontexten, in denen eine Quantifizierung nötig ist, sind Angaben mit Dimensionsadjektiven (*lang, breit, hoch, ...*) [Bie89], aber auch Grenzadjektive-Adjektive (*maximal, minimal, ...*) [Par97]. Wenn solche Ausdrücke verwendet werden, ohne dass die notwendige Quantifizierung gegeben wird, so liegt ein Verstoß gegen die Messbarkeit vor. Die Adjektive werden in ihrer Verwendung ohne quantifizierende Angaben zu den *Weak-Words* gerechnet. Andere Beispiele sind Ausdrücke wie *wiederholt, regelmäßig* und *öfters*, die für zählbare Kategorien stehen, jedoch keine Zählung angeben (im Gegensatz zu *dreimal, etc.*).

Beispiel 4 *Bei einer langen Schließbetätigung am Schlüssel werden die Dachrollen geschlossen.*

In Beispiel 4 tritt das Dimensionsadjektiv *lang* ohne eine quantifizierende Angabe auf, was einen großen Interpretationsspielraum eröffnet. Autor und Leser können verschiedene Vorstellungen davon haben, was *lang* in dieser Anforderung genau bedeutet. Der Autor muss hier eine temporale Angabe hinzufügen, damit die Anforderung messbar und somit auch testbar wird.

Das Fehlen einer benötigten quantifizierenden Angabe führt in jedem Fall zu einer nicht eindeutig testbaren Anforderung. Das Auftreten dieses Phänomens ist folglich als kritisch im Hinblick auf die Testbarkeit einer Anforderung einzustufen und deshalb in der graphischen Form des Qualitätsmodells rot markiert.

Das Testbarkeitskriterium *Messbarkeit* steht eng in Zusammenhang mit dem Kriterium *Fehlende Information*, welches wiederum dem Testbarkeitskriterium *Verständlichkeit* zugeordnet ist. Die Messbarkeit einer Anforderung stellt einen sehr wichtigen Aspekt dar, im Bezug auf die Testbarkeit von Anforderungen, weshalb das Kriterium separat auf der obersten Ebene des Qualitätsmodells angeordnet ist.

4.2 Testbarkeitskriterium Verständlichkeit

Das Testbarkeitskriterium *Verständlichkeit* beinhaltet die meisten Phänomene. Diese Phänomene wurden in verschiedene Unterkategorien aufgeteilt und dabei nach linguistischen Kriterien sortiert. Es wurden folgende linguistische Kriterien verwendet: *Fehlende Information, Verbindlichkeit, Mehrdeutigkeit* und *Sprachkomplexität*. In Abbildung 14 ist das Testbarkeitskriterium *Verständlichkeit* mit allen linguistischen Kriterien und sprachlichen Phänomenen abgebildet. In den folgenden Abschnitten soll nun auf jedes linguistische Kriterium und dessen zugeordnete sprachliche Phänomene genauer eingegangen werden.

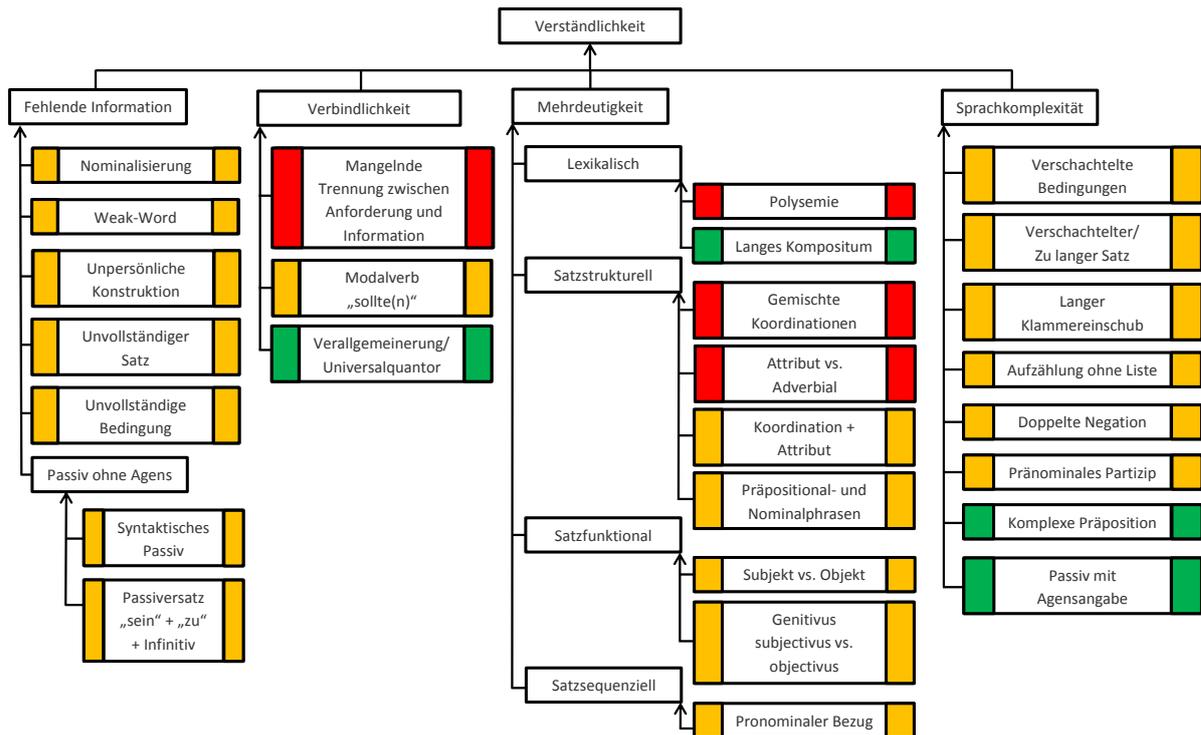


Abbildung 14: Qualitätskriterium Verständlichkeit

4.2.1 Fehlende Information

Das linguistische Kriterium *Fehlende Information* enthält die folgenden Phänomene: *Nominalisierung*, *Weak-Word*, *Unpersönliche Konstruktion*, *Unvollständiger Satz*, *Unvollständige Bedingung* und *Passiv ohne Agens*. *Passiv ohne Agens* gliedert sich wiederum in *Syntaktisches Passiv* und *Passiversatz-Konstruktionen* in der Form *sein + zu + Infinitiv*. Diese sprachlichen Phänomene haben nur in bestimmten Kontexten einen negativen Einfluss auf die Testbarkeit einer Anforderung. Nicht jedes Auftreten ist daher als kritisch zu bewerten. Die Markierung in gelb in der graphischen Darstellung des Qualitätsmodells soll diesen Aspekt graphisch darstellen.

4.2.1.1 Nominalisierung

Eine Nominalisierung (= Substantivierung) ist ein Nomen, das aus einem Stamm einer anderen Wortart (Verben oder Adjektive) abgeleitet wurde. Sind an der Wortbildung der Nominalisierung Affixe beteiligt, wird der Schritt *Derivation* genannt, ohne Beteiligung von Affixen handelt es sich um *Konversion* [EKR09]. Laut [Rup14] können Nominalisierungen wie *Erhöhung* einen kompletten Prozess verzerren, laut [tek11] wirken sie zudem „schwerfällig, führen zu einer großen Informationsdichte und erschweren so die Textverständlichkeit“. Laut [Sch12] werden Sätze, welche Nominalisierungen enthalten, oft umständlich formuliert.

Beispiel 5 *Die Erhöhung des Wertes X muss durchgeführt werden.*

In Beispiel 5 muss aufgrund der Verwendung der Nominalisierung ein neues Verb (*durchführen*) gewählt werden, sodass ein Funktionsverbgefüge entsteht. Außerdem wird nicht spezifiziert, wer die Erhöhung durchführen soll und um welche Einheit der Wert X erhöht werden muss. Anders als bei verbalen Ausdrücken (... *muss den Wert X um Z erhöhen* ...), sind alle Argumente von Nominalisierungen optional, d.h. der Autor kann sie weglassen. Wenn Verbargumente vom Autor nicht explizit genannt werden, kann es dazu kommen, dass für den Leser nicht klar ist, welche Handlungsbeteiligten welchen Prädikaten zugeordnet sind (*wer macht was?*).

Nominalisierungen sind nur in bestimmten Kontexten kritisch. Oft können Informationen, welche aufgrund der Nominalisierung getilgt werden, im Satz- oder Diskurskontext oder durch Expertenwissen erschlossen werden. Dieses Phänomen ist daher mittelkritisch zu bewerten und im Qualitätsmodell gelb markiert⁵.

4.2.1.2 Weak-Word

Weak-Words sind ein weiteres linguistisches Phänomen des Kriteriums *Fehlende Information*. *Weak-Words* sind „Wörter oder Phrasen, deren Benutzung in einem Freitext darauf schließen lässt, dass der Freitext mit hoher Wahrscheinlichkeit unpräzise ist“ [Mel00]. *Weak-Words* werden auch „unscharfe Wörter“ genannt; sie führen in bestimmten Kontexten dazu, dass wichtige Information nicht genannt wird. Im Requirements Engineering ist der Ausdruck *Weak-Word* ein Fachterminus. Aus dem Blickwinkel der Sprachwissenschaft kann die Klasse der *Weak-Words* in Adverbien, unterschiedliche Klassen von Adjektiven und Nominalisierungen der Adjektive unterteilt werden. Im Folgenden soll zunächst die allgemeine Einteilung von Adjektiven aufgeführt werden und dann beschrieben werden in welche der aufgeführten Adjektivklassen die *Weak-Words* fallen.

In [Par97] werden Adjektive in die drei folgenden Klassen unterteilt: skalierbare Adjektive (*schnell, lang*), extreme Adjektive (*brilliant, desaströs*) und Grenzadjektive (*wahr, möglich*). Paradis führt für die Unterscheidung der Klassen vier Kriterien ein:

- Kriterium 1: Das Adjektiv ist steigerbar.
- Kriterium 2: Folgende Konstruktion kann gebildet werden: *Wie X ist es?*
- Kriterium 3: Das Adjektiv kann in einem Exklamativsatz verwendet werden.
- Kriterium 4: Das Adjektiv kann in der semantischen Relation der Antonymie zu einem anderen Adjektiv stehen.

⁵Ein Werkzeug zur Analyse englischer Nominalisierungen wird in [LKT⁺15] präsentiert.

Skalierbare Adjektive erfüllen alle vier Kriterien, extreme Adjektive hingegen erfüllen nur die Kriterien 3 und 4 und Grenzadjektive erfüllen lediglich Kriterium 4.

[Bie89] teilt die skalierbaren Adjektive in zwei weitere Untergruppen ein: Dimensionsadjektive (*lang*, *schnell*) und evaluative Adjektive (*gut*, *schön*).

Die Weak-Words fallen in die Klassen Dimensionsadjektive und Grenzadjektive.

In dieser Arbeit wird anstatt der linguistischen Begriffe Dimensionsadjektive und Grenzadjektive der Begriff aus dem Requirements Engineering (Weak-Words) verwendet. Ein Beispiel für ein Weak-Word ist die Wortform *lange* in Beispiel 6.

Beispiel 6 *Eine lange Betätigung der Taste löst einen Alarm aus.*

Betrachtet man den Satz in Beispiel 6, kann man feststellen, dass die Verwendung der Wortform *lange* dazu führt, dass die Anforderung nicht vollständig spezifiziert ist. Das Wort *lang(e)* ohne quantifizierende Angabe ist subjektiv und kann in seiner Interpretation von Mensch zu Mensch variieren.

Lexikalische Einheiten, die Weak-Word-Lesarten haben, können aber auch andere Lesarten haben, die keinen Interpretationsspielraum aufwerfen (vgl. Beispiel 7).

Beispiel 7 *Eine 3 Sekunden lange Betätigung der Taste löst einen Alarm aus.*

Die Verwendung der Wortform *lange* in Beispiel 7 ist nicht kritisch im Bezug auf die Testbarkeit, da das Wort hier keinen Interpretationsspielraum auslöst. Die Verwendung des Ausdrucks *3 Sekunden lange* hilft in diesem Fall sogar dabei, die Anforderung zu präzisieren. Ob Weak-Words im Hinblick auf die Testbarkeit einer Anforderung, kritisch sind, ist folglich stark von dem Satzkontext abhängig, in welchem das Weak-Word steht. Das Auftreten dieses Phänomens ist daher mittel-kritisch und deshalb im Qualitätsmodell gelb markiert.

Die Klassifikation von lexikalischen Einheiten als Weak-Words liegt also quer zu den Kategorien des vorliegenden Qualitätsmodells. Weak-Word-Verwendungen sind durch das Fehlen (expliziter) Angaben gekennzeichnet; dadurch entsteht Interpretationsspielraum, der in Anforderungen unerwünscht ist. Das Fehlen solcher Angaben kann die Verständlichkeit beeinträchtigen, es kann aber auch die Messbarkeit beeinträchtigen (vgl. Abschnitt 4.1).

4.2.1.3 Unpersönliche Konstruktion

Durch die Verwendung unpersönlicher Konstruktionen, wie z.B. Sätze mit expletivem *es* oder *man* als Subjekt, kann die semantische Rolle Agens eines Satzes unausgedrückt gelassen werden, d.h. das Agens wird nicht textuell wiedergegeben. Probleme können entstehen, wenn sich diese fehlende Information weder durch Welt-, Kontext-, noch durch Expertenwissen herleiten lässt.

Das Pronomen *es* kann in vier unterschiedlichen Gebrauchsweisen in Texten auftreten:

- (i) Referentielles/Phorisches *es*
- (ii) Expletives *es*
- (iii) Korrelat-*es*
- (iv) Vorfeld-*es* bzw. Vorfeldplatzhalter

Die Gebrauchsweisen (i)-(iii) entsprechen dem unpersönlichen *es*. Gebrauchsweise (i) wird in Abschnitt 4.2.3.4.1 in der Kategorie der pronominalen Bezüge näher erläutert.

Beispiel 8 *Es gibt einen hängenden Schalter.*

Beispiel 9 *Die Funktion ermöglicht es, dass [...]*

In Gebrauchsweise (ii) fallen Formulierungen wie in Beispiel 8, bei welchen das Pronomen *es* obligatorisch ist und somit nicht weggelassen werden kann. In Gebrauchsweise (iii) fallen die Verwendungen, in welchen sich das Pronomen *es* auf einen Nebensatz bezieht (vgl. Beispiel 9).

Beispiel 10 *Es ist eine Erkennung für einen hängenden Schalter zu realisieren.*

Beispiel 11 *Es wird das Rollo normiert.*

In Gebrauchsweise (iv) fallen u.a. agenslose Passivsätze wie in Beispiel 11. In diesem Beispiel wird das Agens nicht textuell wiedergegeben. Dies kann zu Problemen führen, wenn der Referent, auf den sich das Subjekt in einem Aktivsatz beziehen würde (also das Agens), vom Leser aus dem Kontext nicht erschlossen werden kann. In Beispiel 10 führt die Verwendung des Expletivs zu keinen Verständnisproblemen, weil aus dem Hintergrundwissen, dass Lastenhefte für den Auftragnehmer geschrieben werden, hergeleitet werden kann, dass der Auftragnehmer die Anforderung realisieren soll, d.h. dass er das Agens des Verbs *realisieren* ist. Anders verhält es sich bei der Anforderung in Beispiel 11. Die Tilgung des Agens führt hier dazu, dass nicht hergeleitet werden kann, wer oder was die *Normierung* durchführt. Die Kategorie der unpersönlichen Konstruktionen liegt

folglich quer zu der Kategorie *Passiv ohne Agens*, beinhaltet jedoch noch weitere Gebrauchsweisen.

Die unpersönliche Konstruktion *man* ist weniger kritisch zu betrachten, da das Pronomen nur in Fällen verwendet wird, in welchen eindeutig das Agens bestimmt werden kann (vgl. Beispiel 12).

Beispiel 12 *Unter Ansteuerung versteht man: [...]*

Unpersönliche Konstruktionen stellen nicht grundsätzlich ein Problem dar. Auch hier muss von Fall zu Fall unterschieden werden, ob die Tilgung des Agens zu einer unerspezifizierten Anforderung führt und somit negativen Einfluss auf die Testbarkeit hat. Dieser Aspekt wird durch die gelbe Markierung im Qualitätsmodell dargestellt.

4.2.1.4 Unvollständiger Satz

Das Weglassen von Wörtern in Sätzen kann ebenfalls dazu führen, dass Informationen, welche für das Verständnis des Lesers wichtig sind, nicht genannt werden. Ein Beispiel für einen unvollständigen Satz ist in Beispiel 13 zu sehen.

Beispiel 13 *Das Signal auf den Wert 1 zurücksetzen.*

In Beispiel 13 wird das Subjekt des Satzes nicht genannt. Somit ist nicht klar spezifiziert wer oder was das Signal zurücksetzen muss. In [tek11] wird empfohlen keine Wörter wegzulassen. Die Leitlinie geht hierbei insbesondere auf Verben ein: „Wenn Sätze Verben enthalten, werden Mehrdeutigkeiten vermieden und Leser können die Informationsarten schneller erfassen.“ Die Kritikalität muss von Fall zu Fall unterschieden werden. Das Weglassen eines Wortes hat nicht immer einen Einfluss auf die Testbarkeit der entsprechenden Anforderung. Dies hängt in hohem Maße von der Wortart und vom Satzkontext ab. Lässt der Autor einen Artikel im Satz weg, hat dies vergleichsweise wenig Einfluss auf die Verständlichkeit einer Anforderung. Fehlt hingegen ein Nomen oder ein Verb, hat dies weit größeren Einfluss auf die Verständlichkeit und somit auch auf die Testbarkeit.

Das Phänomen ist mittelkritisch zu beurteilen und deshalb im Qualitätsmodell gelb markiert.

4.2.1.5 Unvollständige Bedingung

Unvollständige Bedingungen können ebenfalls ein Problem im Hinblick auf die Testbarkeit darstellen. In Beispiel 14, welches sich aus zwei Anforderungen zusammensetzt, ist ein solcher Fall abgebildet.

Beispiel 14 *Wenn Signal A 15 Sekunden mit dem Wert 1 anliegt, setzt B das Signal C auf 0 zurück. Wenn das Signal 20 Sekunden mit dem Wert 1 anliegt, setzt B das Signal C auf den Wert 1.*

In Beispiel 14 wird lediglich beschrieben, was geschehen soll, wenn Signal A genau 15 Sekunden oder genau 20 Sekunden anliegt. Über die Bereiche $T(A) < 15$, $15 < T(A) < 20$ und $T(A) > 20$ wird keine Aussage getroffen. Geht es in einer Anforderung um messbare Bereiche wie in Beispiel 14, müssen alle möglichen und relevanten Bereiche spezifiziert werden. Geschieht dies nicht, werden Bereiche, welche ebenfalls relevant sind, möglicherweise nicht abgetestet. Unvollständige Bedingungen sind nicht immer kritisch im Hinblick auf die Testbarkeit einer Anforderung. Falls in bestimmten Situationen ein bestimmter Bereich oder Zustand gar nicht erreicht werden kann, muss dieser nicht spezifiziert werden, weil er nicht relevant ist. Das Phänomen ist aus diesem Grund als mittelkritisch eingestuft und somit im Qualitätsmodell gelb markiert.

4.2.1.6 Passiv und Passiversatzkonstruktionen ohne Agens

Die Verwendung von Passiv in Anforderungen wird in der Literatur zum Requirements Engineering nicht empfohlen [Kof07], [Rup14], [PR15], [Wun15]. Durch das Passiv kann, wie auch bei den unpersönlichen Konstruktionen, das Agens, also der Handelnde eines Satzes, bewusst weggelassen werden, d.h. er muss textlich nicht wiedergegeben werden. Die semantische Rolle Agens muss aber bei der Verwendung des Passivs nicht zwingend unausgedrückt bleiben. Diese Information kann durch eine *von*- oder eine *durch*-Phrase innerhalb eines Passivsatzes gegeben werden. Passivsätze, bei welchen das Agens in Verbindung mit einer *von*- oder *durch*-Phrase genannt wird, sind als nicht kritisch einzustufen. Passivische Konstruktionen ohne Agensangabe lassen sich nochmals in zwei Fälle aufgliedern, welche im Bezug auf Kritikalität immer im Kontext betrachtet werden müssen: syntaktische Passivkonstruktionen und Passiversatz-Konstruktionen der Form ‘*sein* + *zu* + Infinitiv’.

Beispiele für die beiden Verwendungen sind in 15 und 16 zu sehen.

Beispiel 15 *Der Zustand Unterspannung wird erkannt, wenn die Spannung länger 500 Millisekunden ist.*

Beispiel 16 *Das Bit ist zu setzen, wenn eine Ansteuerung mindestens einer Lüftergruppe vom AB erfolgt.*

In beiden Fällen kann nicht hergeleitet werden, wer die Aktionen *erkennen* und *setzen* durchführen muss. Ohne Experten- oder Kontextwissen kann das Agens nicht hergeleitet werden. In Lastenheften mit einer Auftraggeber-Auftragnehmer-Beziehung ist allerdings

in den meisten Fällen klar, wer das Agens der Anforderung darstellt: der Auftragnehmer oder das zu beschreibende System bzw. die zu beschreibende Komponente. Weitere Informationen können häufig aus dem Kontext erschlossen werden. In der tekomp-Leitlinie wird angemerkt, dass bei automatisch ablaufenden Prozessen die Verwendung des Passivs möglich ist [tek11]. Auch im Duden-Ratgeber wird von der Verwendung des Passivs generell abgeraten, es wird aber auch erwähnt, dass es bei beschreibenden Informationen in Ordnung ist, das Passiv zu verwenden, weil hier nicht die Notwendigkeit herrscht, das Agens explizit zu nennen [Sch12]. Passiv ist daher nicht immer als kritisch zu betrachten und aus diesem Grund im Qualitätsmodell gelb markiert.

4.2.2 Verbindlichkeit

Verbindlichkeit ist ebenfalls ein Teilkriterium des Testbarkeitskriteriums *Verständlichkeit*. Die Verbindlichkeit gibt Aufschluss darüber, ob der Auftragnehmer die in der Anforderung beschriebene Funktionalität realisieren muss, oder ob sie nur eine Möglichkeit darstellt. Dem Kriterium *Verbindlichkeit* sind drei Phänomene untergeordnet: *Mangelnde Trennung zwischen Anforderung und Information*, *Modalverb „sollte(n)“* und *Verallgemeinerung/Universalquantor*.

4.2.2.1 Mangelnde Trennung zwischen Anforderung und Information

Die Etikettierung von Anforderungen nach ihrem Typ ist ebenfalls ein wichtiger Punkt, der beim Schreiben von Anforderungen beachtet werden muss. Wird ein Satz als Anforderung markiert, wird festgelegt, dass die Anforderung potentiell rechtlich bindend ist. Hinzu kommt, dass die entsprechenden Schlüsselwörter (*müssen, hat ... zu ..., ist ... zu*) im Text der Anforderungen gewählt werden müssen (vgl. Abschnitt 4.2.2.2). Erhält ein Satz das Etikett 'Information', ist der beschriebene Sachverhalt in keinem Fall rechtlich bindend, sondern dient lediglich als Zusatzinformation für den Auftragnehmer (eine genaue Beschreibung der möglichen Etikettierungen findet sich in Abschnitt 6.1). Ist ein Satz fälschlicherweise als Information markiert, obwohl es sich um eine zu realisierende Anforderung handelt, steht der Auftragnehmer nicht in der Pflicht die beschriebene Funktionalität zu realisieren. Das Auftreten dieses Phänomens ist aus diesem Grund der höchsten Kritikalitätsstufe zuzuordnen und daher im Qualitätsmodell rot markiert.

4.2.2.2 Modalverb *sollte(n)*

Im Requirements Engineering wird empfohlen für bindende Anforderungen stets das Schlüsselwort *müssen* zu verwenden [Rup14]. Durch dieses Wort wird festgelegt, dass der in der entsprechenden Anforderung beschriebene Sachverhalt vom Auftragnehmer umgesetzt werden muss und es sich nicht um eine freiwillig zu erbringende Leistung handelt.

Vorschläge werden hingegen durch das Schlüsselwort *können* gekennzeichnet. Die in dieser Arbeit analysierten Spezifikationen enthalten eine Vielzahl an Vorkommnissen des Wortes *sollte(n)* (vgl. Abschnitt 8.4). Das Wort *sollte(n)* ist aus linguistischer Sicht ambig zwischen Verpflichtung und Optionalität. Dass *sollte(n)* dennoch häufig in Anforderungen verwendet wird, obwohl diese eigentlich rechtlich bindend sein sollen, lässt vermuten, dass den Autoren das Wort *sollte(n)* höflicher erscheint und sie es aus diesem Grund dem scheinbar etwas unhöflicheren *müssen* vorziehen. Bei der Verwendung von *sollte(n)* bleibt allerdings häufig der Grad der Verbindlichkeit offen. Eine Anforderung, die das Wort *sollte* enthält, ist in Beispiel 17 zu sehen.

Beispiel 17 *Der Abstand zwischen Bauteil 1 und Bauteil 2 sollte 2 cm betragen.*

Die Verwendung des Modalverbs *sollte* in Beispiel 17 bewirkt, dass die Verbindlichkeit nicht eindeutig gegeben ist. Das Wort *sollte* lässt in diesem Fall zwei Lesarten zu. Die erste Lesart ist die deontische Lesart (*sollte* steht hier für *muss*) [Hol10]. Die zweite Lesart ist die dispositionelle Lesart, welche dem Wort *können* entspricht [Hol10] und somit die Anforderung optional erscheinen lässt. Da die dispositionelle Interpretation von *sollte* in Beispiel 17 getroffen werden kann, ist der Auftragnehmer nicht dazu verpflichtet die Anforderung umzusetzen, obwohl der Autor dies intendiert hatte.

Beispiel 18 *Falls es zu einem Abbruch der Aktivierung kommen sollte, wird das System neu gestartet.*

Die Verwendung des Modalverbs *sollte(n)* stellt aber nicht grundsätzlich die Verbindlichkeit (deontische vs. dispositionelle Lesart) in Frage (vgl. Beispiel 18). Wenn statt der Einbettung unter dem Modalverb *sollte* (*zu ... kommen sollte*) der Indikativ des Hauptverbs verwendet wird (*zu ... kommt*), bleibt die Interpretation dieselbe. Die Verwendung von *sollte* ist in diesem Fall durch die Grammatik des Konditionalsatzes bedingt, und es entstehen bei dieser Verwendung von *sollte* nicht die beiden kritischen Lesarten (deontisch und dispositionell). Das Wort *sollte* steht hierbei als Markierung für eine Potentialis-Lesart.

Die Verwendung von *sollte(n)* in Anforderungen führt nur in bestimmten Kontexten dazu, dass die Verbindlichkeit nicht klar gegeben ist. Das Phänomen ist folglich mittelkritisch eingestuft und im Qualitätsmodell gelb markiert.

4.2.2.3 Verallgemeinerung/Universalquantor

Bei dem Phänomen *Verallgemeinerung/Universalquantor* handelt es sich um ein Phänomen, das nur in sehr seltenen Fällen einen Einfluss auf die Testbarkeit einer Anforderung

hat. Hierbei handelt sich um die Verwendung von Wörtern wie *alle*, *jeder*, *immer*, usw. In Beispiel 19 ist ein Satz, welcher den Universalquantor *alle* enthält, zu sehen.

Beispiel 19 *Alle Werte müssen innerhalb des Messbereichs liegen.*

In [Rup14] wird vorgeschlagen, dass bei Auftreten eines Universalquantors geprüft werden soll, ob die durch diesen Universalquantor ausgedrückte Verallgemeinerung wirklich gilt. Bei dem Universalquantor *alle* aus Beispiel 19 soll beispielsweise geprüft bzw. hinterfragt werden, ob denn *wirklich alle Werte* ohne Ausnahme gemeint sind. Da das Auftreten dieses Phänomens nur sehr selten einen Einfluss auf die Testbarkeit der entsprechenden Anforderung hat, ist dieses Phänomen als gering kritisch eingestuft und somit in der Graphik des Qualitätsmodells grün markiert.

4.2.3 Mehrdeutigkeit

Das Auftreten von sprachlichen Mehrdeutigkeiten kann ebenfalls die Testbarkeit einer Anforderung beeinflussen. Im Qualitätsmodell ist die Kategorie *Mehrdeutigkeit* nochmals in vier Unterkategorien untergliedert: *Lexikalische Mehrdeutigkeiten*, *satzstrukturelle Mehrdeutigkeiten*, *satzfunktionale Mehrdeutigkeiten* und *satzsequenzielle Mehrdeutigkeiten*. Diese Unterscheidung beruht auf der Größe der Beschreibungseinheiten (Wörter, Sätze, Satzsequenzen) sowie auf den linguistischen Beschreibungsebenen der (lexikalischen) Semantik, Konstituenz bzw. Dependenz.

4.2.3.1 Lexikalische Mehrdeutigkeit

„Unter lexikalischer Ambiguität wird [...] der Fall verstanden, dass einer lexikalischen Form mehrere konventionalisierte Sinne zugeordnet werden, wobei jede Verbindung einer lexikalischen Form mit einem konventionalisierten Sinn eine distinktive lexikalische Einheit konstituiert“ [Beh93]. Eine lexikalische Mehrdeutigkeit liegt folglich vor, wenn ein Wort mehrere Bedeutungen haben kann. Hierbei kann zwischen Homonymie und Polysemie unterschieden werden. Ein weiteres Phänomen, das in die Kategorie der lexikalischen Mehrdeutigkeiten fällt, sind lange Komposita. Lange Komposita können aufgrund ihrer Wortstruktur ambig sein.

4.2.3.1.1 Polysemie und Homonymie

„Ein Wort ist [...] polysem, wenn es mehrere Bedeutungen besitzt, die in einem begrifflichen Zusammenhang stehen“ [SK04]. Ein Beispiel hierfür ist das Wort *Pferd*. Dieses Wort kann für ein Tier oder für eine Figur im Schach stehen. Der Name der Schachfigur ist aufgrund der äußerlichen Ähnlichkeit von dem Namen des Tieres abgeleitet. Ein Beispiel für Polysemie aus dem Englischen ist das Wort *regular*, welches die Bedeutung *regelmäßig*

oder *üblich* bzw. *herkömmlich* haben kann. Polysem sind auch Konstruktionen, bei welchen ein Schrägstrich als *und*, *oder* oder *bzw.* fungieren kann. In Beispiel 20 ist ein solcher Fall zu sehen.

Beispiel 20 *In diesem Fall müssen die Start-/Stopfbefehle berücksichtigt werden.*

In Beispiel 20 kann nicht eindeutig abgeleitet werden, welche Art Konjunktion der Schrägstrich darstellt. Intuitiv würde man vermuten, dass es sich um die Konjunktion *und* handelt. Die alternativen Konjunktionen *oder* und *bzw.* können aber nicht mit absoluter Sicherheit ausgeschlossen werden.

Homonyme hingegen sind „lexikalische Grundformen, die zwar dieselbe phonologische (Homophone) oder graphemische Form (Homographe) besitzen, deren verschiedene Bedeutungen aber in keinem semantischen Zusammenhang stehen.“ [SK04] Ein Beispiel für ein Homonym ist das Wort *Bank*. Es kann ein Geldinstitut oder eine Sitzgelegenheit denotieren. Tritt ein Wort in einer Anforderung auf, das mehrere Bedeutungen haben kann, ist dies in jedem Falle als kritisch zu bewerten. Dieses Phänomen ist deshalb in der Graphik des Qualitätsmodells rot markiert.

4.2.3.1.2 Langes Kompositum

Ein weiteres Phänomen, welches den lexikalischen Mehrdeutigkeiten zugeordnet werden kann, ist *Langes Kompositum*. Auch in diesen Fällen bezieht sich die Ambiguität auf ein einzelnes Wort. Lange bzw. „vielgliedrige Komposita sind relativ schwer zu verstehen“ [EKR09]. In Beispiel 21 ist ein mehrgliedriges Kompositum aufgeführt, welches dieselbe Strukturambiguität aufweist wie das in der Literatur häufig aufgeführte Beispiel *Mädchenhandelsschule* von [Clé00].

Beispiel 21 *Heckdeckelfernentriegelungsschalter*

Das Kompositum *Heckdeckelfernentriegelungsschalter* kann auf mehrere Arten interpretiert werden: 1. Interpretation: Es handelt sich um einen Schalter für die Heckdeckelfernentriegelung. 2. Interpretation: Es handelt sich um einen Fernentriegelungsschalter für den Heckdeckel. Durch Weltwissen kann allerdings die richtige Interpretation (Interpretation 1) identifiziert werden. Weitere Interpretationen sind morphologisch möglich, aber unplausibel (z.B. Entriegelungsschalter für die Heckdeckelferne).

Lange Komposita führen nur in sehr seltenen Fällen dazu, dass eine Fehlinterpretation getroffen wird. Welt- und Expertenwissen hilft an dieser Stelle, um die korrekte Interpre-

tation zu treffen. Der Kritikalitätsgrad dieses Phänomens ist daher als niedrig eingestuft und im Qualitätsmodell grün markiert.

4.2.3.2 Satzstrukturelle Mehrdeutigkeit

Satzstrukturelle Mehrdeutigkeiten schließen mehrere Wörter eines Satzes mit ein. Unter diese Kategorie fallen folgende Phänomene: *Gemischte Koordinationen*, *Attribut vs. Adverbial*, *Koordination + Attribut*, *Präpositional- und Nominalphrasen (PP-Attachment)*.

4.2.3.2.1 Gemischte Koordinationen

Ein Phänomen, das in die Kategorie der satzstrukturellen Mehrdeutigkeiten fällt, ist das Phänomen *Gemischte Koordinationen*. Laut [Kam01] führt die Verwendung mehrerer Konjunktionen innerhalb eines Satzes dazu, dass dem Satz mehrere mögliche Lesarten zugeordnet werden können. Diese Aussage ist nur teilweise korrekt, da lediglich gemischte Koordinationen dazu führen, dass ein Satz mehrere mögliche Lesarten hat.

Beispiel 22 *Die Werte werden ermittelt und eingelesen und gespeichert.*

Beispiel 23 *Die Werte werden ermittelt oder eingelesen oder gespeichert.*

In den Beispielen 22 und 23 werden drei Verben (*ermittelt*, *eingelesen*, *gespeichert*) durch jeweils dieselben Konjunktionen koordiniert. Jedoch liegt in beiden Fällen keine Ambiguität aufgrund der Aneinanderreihung mehrerer Koordinationen vor. In Beispiel 22 müssen alle drei Aktionen (*ermitteln*, *einlesen* und *speichern*) durchgeführt werden, in Beispiel 23 lediglich eine der Aktionen. Anders verhält es sich mit gemischten Koordinationen.

Beispiel 24 *Die Werte werden ermittelt oder eingelesen und gespeichert.*

In Beispiel 24 ist eine Anforderung zu sehen, in welcher eine satzstrukturelle Ambiguität aufgrund der Aneinanderreihung mehrerer gemischter Koordinationen vorliegt. Es ist nicht eindeutig, welche der beiden Konjunktionen (*und*, *oder*) in dieser Anforderung stärker bindet. Die Anforderung hat folglich zwei Lesarten, welche in den Beispielen 25 und 26 dargestellt werden.

Beispiel 25 *Die Werte werden ermittelt oder eingelesen. Die Werte werden gespeichert.*

Beispiel 26 *Die Werte werden ermittelt oder eingelesen. Die eingelesenen Werte werden gespeichert.*

In Beispiel 25 ist die Lesart von Beispiel 24 zu sehen, bei welcher die Konjunktion *oder* stärker bindet. In Beispiel 26 hingegen ist die Lesart zu sehen, bei welcher die Konjunktion

und stärker bindet. Es handelt sich hierbei um zwei unterschiedliche Interpretationen, welche auch unterschiedlich getestet werden müssen.

Die Aneinanderreihung gemischter Koordinationen in einer Anforderung führt dazu, dass eine Anforderung mehrere Lesarten hat, d.h. es können unterschiedliche Interpretationen getroffen werden. Das Auftreten dieses Phänomens ist in jedem Falle kritisch zu betrachten im Hinblick auf die Testbarkeit einer Anforderung. Aus diesem Grund ist das Phänomen der höchsten Kritikalitätsstufe zugeordnet und in der graphischen Darstellung des Qualitätsmodells rot markiert.

4.2.3.2 Attribut vs. Adverbial

Die Ambiguität zwischen *Attribut vs. Adverbial* zählt zu den satzstrukturellen Mehrdeutigkeiten. Laut [tek11] sollen „Konstruktionen vermieden werden, bei denen ein Ausdruck sowohl als Attribut zum vorhergehenden Ausdruck als auch als Adverbial aufgefasst werden kann“. In Beispiel 27 ist eine solche Mehrdeutigkeit zu sehen.

Beispiel 27 *Um das Rollo vorn weiter anzusteuern, muss zunächst die Bedienanforderung X anliegen.*

Die Phrase *das Rollo vorn* ist mehrdeutig, weil das Wort *vorn* als Adverbial zum Verb oder als Attribut zum Nomen *Rollo* interpretiert werden kann. Es ist nicht klar, ob sich das Rollo vorne befindet (Attribut-Lesart) oder ob das Rollo an dessen vorderer Seite angesteuert werden soll (Adverbial-Lesart). In der Fachsprache der betrachteten Lastenhefte treten Nominalphrasen von der Art auf, dass Begriffe wie *vorn(e)*, *hinten*, *rechts*, *links*, *oben*, *unten*, etc. dem Nomen nachgestellt als Attribut zum Nomen interpretiert werden können. In der Gemeinsprache tritt dieser Fall eher selten auf.

Die Beispiele 28 und 29 zeigen, wie man die Anforderung aus Beispiel 27 umformulieren kann, um die jeweils gewünschte Lesart eindeutig wiederzugeben.

Beispiel 28 *Um das Rollo an dessen vorderer Seite weiter anzusteuern, muss zunächst die Bedienanforderung X anliegen.*

Beispiel 29 *Um das vordere Rollo weiter anzusteuern, muss zunächst die Bedienanforderung X anliegen.*

Tritt dieses Phänomen in einer Anforderung auf, ist dies im Hinblick auf die Testbarkeit in jedem Falle kritisch zu betrachten. Das Phänomen ist der höchsten Kritikalitätsstufe zugeordnet und somit im Qualitätsmodell rot markiert.

4.2.3.2.3 Koordination + Attribut

Bei dem sprachlichen Phänomen *Koordination + Attribut* handelt es sich um ein weiteres Phänomen der satzstrukturellen Mehrdeutigkeiten. Beispiel 30 zeigt einen solchen Fall.

Beispiel 30 *Die Komponente wird zwischen oberem Längsträger und Dämpferbein verbaut.*

Hier entsteht die Mehrdeutigkeit durch die Koordination der beiden Nomina *Längsträger* und *Dämpferbein* und das Attribut *oberem*. Das Adjektiv *oberem* bezieht sich auf das Nomen *Längsträger*. Es könnte sich aufgrund seiner morphologischen Form auch auf das Nomen *Dämpferbein* beziehen. Ob die Bedeutung von *ober(em)* auf beide Konjunkte verteilt gelesen werden muss, oder nur auf das erste Konjunkt bezogen, wird aus der Anforderung allerdings nicht ersichtlich. Die tekomp-Leitlinie schlägt vor, dass „wenn zu koordinierten Ausdrücken ein Attribut tritt, man prüfen sollte, ob klar ist, auf welche der koordinierten Ausdrücke sich das Attribut bezieht“ [tek11]. In Beispiel 30 gibt es zwei Interpretationen. Entgegenwirken kann man einer solchen Mehrdeutigkeit, indem man das Adjektiv, wenn es sich auf beide Nomina bezieht, wiederholt (vgl. Beispiel 31). Wenn sich das Adjektiv nur auf eines der Nomina bezieht, ist es ratsam, das Nomen mit dem Adjektiv an zweiter Stelle zu nennen (vgl. Beispiel 32).

Beispiel 31 *Die Komponente wird zwischen oberem Längsträger und oberem Dämpferbein verbaut.*

Beispiel 32 *Die Komponente wird zwischen Dämpferbein und oberem Längsträger verbaut.*

Das Phänomen *Koordination + Attribut* ist nur in bestimmten Kontexten kritisch. In vielen Fällen kann durch Weltwissen hergeleitet werden, ob sich das Attribut auf ein Nomen oder auf beide Nomina bezieht. Die Kritikalitätsstufe ist aus diesem Grund die mittlere und somit ist das Phänomen im Qualitätsmodell gelb markiert.

4.2.3.2.4 Präpositional- und Nominalphrasen

Eine strukturelle Mehrdeutigkeit kann auch entstehen, wenn Präpositionalphrasen so auf Nominalphrasen folgen, dass nicht klar ist, ob die Präpositionalphrase Teil der Nominalphrase ist oder syntaktisch selbstständig. Eine Anforderung, die eine solche Aneinanderreihung enthält, ist in Beispiel 33 zu sehen.

Beispiel 33 *Über den Parameter X kann die Drehrichtung der Waschpumpe auf dem Pumpenplatz 2 geändert werden.*

Bei genauerer Betrachtung von Beispiel 33 kann man feststellen, dass nicht eindeutig ist, ob sich die *Waschpumpe* auf dem *Pumpenplatz 2* befindet, oder ob auf dem *Pumpenplatz 2* die *Drehrichtung* geändert werden kann. Die syntaktische Ambiguität (PP-Attachment) führt zu einer semantischen Ambiguität: „Durch die Reihenfolge der Satzglieder ist nicht immer deutlich erkennbar, wie die einzelnen Satzglieder semantisch miteinander zusammenhängen“ [tek11]. Konstruktionen wie diese können einfach in verständlichere Konstruktionen umgewandelt werden und sollten daher vermieden werden. Ambige Strukturen, die durch aufeinanderfolgende Präpositional- und Nominalphrasen ausgelöst werden können, können oft durch Expertenwissen aufgelöst werden. Dies bedeutet, dass das Phänomen der mittleren Kritikalitätsstufe zugeordnet werden kann und im Qualitätsmodell gelb markiert ist.

4.2.3.3 Satzfunktionale Mehrdeutigkeit

Satzfunktionale Mehrdeutigkeiten entstehen durch sprachliche Konstruktionen, bei welchen ein Wort syntaktisch mehrere Satzfunktionen einnehmen kann. Beispiele hierfür sind die Phänomene *Subjekt vs. Objekt* und *Genitivus subjectivus vs. objectivus*. Auf diese soll im Folgenden detaillierter eingegangen werden.

4.2.3.3.1 Subjekt vs. Objekt

Eine satzfunktionale Mehrdeutigkeit kann vorliegen, wenn Kasussynkretismus von zwei Nominalphrasen dazu führt, dass beiden je zwei alternative grammatische Funktionen zugeordnet werden können, beispielsweise wenn das grammatische Subjekt eines Satzes nicht eindeutig identifiziert werden kann, d.h. wenn es mehrere Möglichkeiten für diese Zuordnung gibt. In [tek11] wird empfohlen Konstruktionen dieser Art generell zu vermeiden. In Beispiel 34 ist eine solche Mehrdeutigkeit zu sehen.

Beispiel 34 *Die Beschreibung der Schnittstellen enthält das Kapitel ‘Verknüpfungen’.*

In Beispiel 34 kann aus der Satzstruktur allein nicht erschlossen werden, ob *Beschreibung der Schnittstellen* oder *Kapitel ‘Verknüpfungen’* das Subjekt des Satzes bildet. Genauso ist nicht eindeutig, welche der beiden Nominalphrasen das Objekt bildet. Auch bei diesem Phänomen ist es der Fall, dass das Auftreten nicht immer kritisch ist. Experten- und Kontextwissen helfen häufig dabei diese Art Mehrdeutigkeit aufzulösen. Das Auftreten dieses Phänomens ist mittelkritisch im Hinblick auf die Testbarkeit einer Anforderung zu bewerten und somit im Qualitätsmodell gelb markiert.

4.2.3.3.2 Genitivus subjectivus vs. objectivus

Eine satzfunktionale Mehrdeutigkeit kann auch durch die Verwendung eines missverständlichen Genitivs entstehen. In Beispiel 35 ist eine Anforderung, die dieses Phänomen beinhaltet, zu sehen.

Beispiel 35 *Abschirmung der Elektrik gegen das Eindringen von Spritzwasser.*

Bei dem Genitivattribut *Elektrik* kann allein aufgrund der Satzstruktur nicht entschieden werden, ob es sich um einen Genitivus objectivus (= logisches Objekt) oder einen Genitivus subjectivus (= logisches Subjekt) handelt. Wenn es sich um einen Genitivus objectivus handelt, wird die *Elektrik* durch etwas abgeschirmt. Handelt es sich um einen Genitivus subjectivus, schirmt die *Elektrik* etwas ab. Die Satzfunktion des Genitivattributs kann in diesem Beispiel durch die Satzstruktur allein nicht eindeutig bestimmt werden.

Die tekomp-Leitlinie empfiehlt auf Genitivkonstruktionen dieser Art zu verzichten [tek11]. Durch Beispiel 35 wird allerdings auch ersichtlich, dass die Kritikalität dieses Phänomens vom Weltwissen des Lesers abhängig ist. Aus technischer Sicht ist es für den Experten als Adressaten der Lastenhefte eindeutig, dass die Elektrik abgeschirmt werden soll und nicht, dass die Elektrik selbst das Eindringen von Spritzwasser verhindert. Dieses Phänomen ist folglich mittelkritisch im Hinblick auf die Testbarkeit zu sehen und im Qualitätsmodell daher gelb markiert.

4.2.3.4 Satzsequentielle Mehrdeutigkeit

Satzsequentielle Mehrdeutigkeiten betreffen Mehrdeutigkeiten, die nicht nur innerhalb eines Satzes, sondern in Satzgefügen oder satzübergreifend auftreten können. Ein Beispiel hierfür sind pronominaler Bezüge.

4.2.3.4.1 Pronominaler Bezug

Mehrdeutigkeiten beim Bezug anaphorischer Ausdrücke, wie z.B. von Personal-, Relativ- oder Possessivpronomina, können ebenfalls einen Einfluss auf die Verständlichkeit einer Anforderung haben. Tritt ein Pronomen auf, welches sich aufgrund von Kongruenz in Genus und Numerus auf mehrere Nomina in einem vorhergehenden (Teil-)Satz beziehen kann, ist nicht eindeutig, auf welches Nomen sich das Pronomen bezieht.

Beispiel 36 *Der Blink-Master ist im Außenlicht-Master integriert. Er empfängt alle Signale.*

Das Pronomen *Er* und die beiden Nomina *Blink-Master* und *Außenlicht-Master* in Beispiel 36 kongruieren in Genus (Maskulinum) und Numerus (Singular). Aufgrund dieser Kongruenz ist nicht eindeutig, auf welches Nomen sich das Pronomen bezieht. In [tek11] wird empfohlen, pronominale Bezüge über Satzgrenzen hinweg zu vermeiden und anstatt ein Pronomen zu verwenden das Bezugswort zu wiederholen. In [Sch12] wird geraten „Relativbezüge zwischen Segmenten und Sätzen unbedingt zu vermeiden“.

Fälle wie in Beispiel 37 sind nicht kritisch, da aufgrund der Kongruenz klar entschieden werden kann, dass sich das Pronomen *Sie* nur auf das Nomen *Motorlager* beziehen kann.

Beispiel 37 *Die Motorlager tragen das Eigengewicht des Triebwerkes und definieren so dessen Position im Fahrzeug. Sie stützen Motormomente ab.*

Aber auch Fälle, bei welchen ein Pronomen mit zwei alternativen Nomina kongruiert, müssen nicht unbedingt einen negativen Einfluss auf die Verständlichkeit haben. Oft kann auch hier durch Expertenwissen aufgelöst werden, auf welches Nomen sich das Pronomen bezieht. Dieses sprachliche Phänomen ist folglich nur in bestimmten Kontexten kritisch und daher der mittleren Kritikalitätsstufe zugeordnet. Im Qualitätsmodell ist das Phänomen gelb markiert.

4.2.4 Sprachkomplexität

Das Kriterium *Sprachkomplexität* beinhaltet weitere sprachliche Phänomene, die einen Einfluss auf die Testbarkeit einer Anforderung haben können. In dieser Phänomenklasse werden verschiedene grammatische und texstrukturelle Phänomene zusammengefasst, die Texte „umständlich formuliert“ oder übermäßig lang werden lassen, sodass sie nicht einfach zu verstehen sind. Diese Phänomene leiten sich hauptsächlich aus Stil-Empfehlungen, welche in der Literatur zur technischen Dokumentation zu finden sind, ab. Zu diesen Phänomenen zählen: *Verschachtelte Bedingungen*, *Verschachtelter/Zu langer Satz*, *Langer Klammereinschub*, *Aufzählung ohne Liste*, *Doppelte Negation*, *Pränominales Partizip*, *Komplexe Präposition* und *Passiv mit Agensangabe*.

4.2.4.1 Verschachtelte Bedingungen

Verschachtelte Bedingungen können ein Problem im Hinblick auf die Verständlichkeit und somit auch auf die Testbarkeit einer Anforderung darstellen. In Beispiel 38 ist eine verschachtelte Bedingung zu sehen.

Beispiel 38 *Wenn das System nach einem solchen Fehler gestartet wird bevor die Fehleranzeige im Kombi erfolgt ist – also ohne vorheriges Ein-*

schalten von AB – soll nach dem fehlerlosen neuen Betrieb kein Fehler aus einem alten Betrieb angezeigt werden, wenn nun AC eingeschaltet wird.

Die Anforderung beginnt und endet mit einer Bedingung. Der Leser wird diese Anforderung mehrere Male lesen müssen, um die Anforderung komplett zu verstehen. Das kostet Zeit und es ist nicht garantiert, dass der Leser die Anforderung richtig versteht. Bedingungen sollten daher nicht verschachtelt werden.

In Beispiel 39 ist die Anforderung aus Beispiel 38 zu sehen. Im Unterschied zu Beispiel 38 wurde die zweite Bedingung weiter nach vorne gezogen und im Anschluss an die erste Bedingung gesetzt. Die Anforderung kann somit schneller verstanden werden.

Beispiel 39 *Wenn das System nach einem solchen Fehler gestartet wird bevor die Fehleranzeige im Kombi erfolgt (also ohne vorheriges Einschalten von AB) und wenn AC eingeschaltet ist, soll nach dem fehlerlosen neuen Betrieb kein Fehler aus einem alten Betrieb angezeigt werden.*

Verschachtelte Bedingungen führen nicht grundsätzlich zu Problemen; in jedem Fall kostet es aber Zeit die Anforderung zu verstehen. Es ist also sinnvoll Bedingungen gut zu strukturieren. Da verschachtelte Bedingungen nur in bestimmten Fällen kritisch sind, sind diese im Qualitätsmodell mittelkritisch bewertet und somit gelb markiert.

4.2.4.2 Verschachtelter/Zu langer Satz

Ein weiteres sprachliches Phänomen des Kriteriums *Sprachkomplexität* sind verschachtelte bzw. zu lange Sätze wie in Beispiel 40.

Beispiel 40 *Da die Information über die nicht funktionierende Standheizung mit dem Senden der Botschaft an das Kombi dem Nutzer unübersehbar mitgeteilt wird, ist eine weitergehende Anzeige von Fail in der IB über die Restlaufzeit nicht unbedingt nötig, sodass man bereits zu diesem Zeitpunkt den Systemstatus auf Off ändern kann und die Standheizung so nicht mehr auf evtl. ankommende Abschalt ereignisse achten muss, obwohl das System bereits abgeschaltet ist.*

„Lange Sätze erschweren die Textverständlichkeit, da das Kurzzeitgedächtnis des Lesers überbeansprucht wird“ [tek11]. Die Anforderung in Beispiel 40 enthält eingeschobene Nebensätze und Randinformationen. Für ein besseres Textverständnis muss die Anforderung in mehrere Sätze aufgeteilt werden, wie beispielsweise in Beispiel 41.

Beispiel 41 *Dem Nutzer wird die Information über die nicht funktionierende Standheizung mitgeteilt, indem eine Botschaft an das Kombi gesendet*

wird. Eine weitergehende Anzeige von Fail in der IB über die Restlaufzeit ist deshalb nicht unbedingt nötig. Zu diesem Zeitpunkt kann der Systemstatus auf Off geändert werden. Obwohl das System bereits abgeschaltet ist, muss dann die Standheizung nicht mehr auf evtl. ankommende Abschaltereignisse achten.

Um Anforderungen wie in Beispiel 40 zu verstehen, wird mehr Zeit benötigt als für Anforderungen wie in Beispiel 41. Leser können außerdem schnell zu Fehlinterpretationen kommen und wichtige Informationen übergehen. Das Auftreten des Phänomens führt nicht grundsätzlich dazu, dass eine Anforderung falsch interpretiert wird. Dieses Phänomen ist daher im Qualitätsmodell als mittelkritisch klassifiziert und gelb markiert.

4.2.4.3 Langer Klammereinschub

Auch lange Klammereinschübe erschweren das Verständnis eines Textes, da sie den Satzfluss unterbrechen [tek11]. Ein weiteres Problem von langen Klammereinschüben ist, dass beim Lesen in manchen Fällen nicht klar ist, ob die beschriebene Funktionalität vom Leser realisiert werden muss oder ob sie lediglich eine Information darstellt.

Beispiel 42 *Der Auftragnehmer muss sicherstellen, dass die Aufnahme des Drehsäulenlagers nicht zu Toleranzproblemen (Winkelfehler des Hebels, Wackeln der Achse, großes Spiel oder Lösen der Achse von der Griffmulde) führt.*

In Beispiel 42 ist ein Klammereinschub zu sehen, der den Satzfluss unterbricht. Diese Unterbrechung erschwert die Lesbarkeit des Satzes. In Beispiel 43 ist der Satz aus Beispiel 42 nach der Verteilung in zwei Sätze zu sehen.

Beispiel 43 *Der Auftragnehmer muss sicherstellen, dass die Aufnahme des Drehsäulenlagers nicht zu Toleranzproblemen führt. Toleranzprobleme können sein: Winkelfehler des Hebels, Wackeln der Achse, großes Spiel oder Lösen der Achse von der Griffmulde.*

Die beiden Sätze in Beispiel 43 können schneller vom Leser verarbeitet werden als der Satz in Beispiel 42. Eingeschobene Klammern machen einen Satz unnötig komplex und ‘blähen ihn auf’. Wenn sich der Klammereinschub vermeiden lässt, sollte man die Information erst in einem nachfolgenden Satz angeben. Dieses Phänomen ist nicht immer kritisch im Bezug auf die Testbarkeit einer Anforderung und ist somit als mittelkritisch eingestuft und im Qualitätsmodell gelb markiert.

4.2.4.4 Aufzählung ohne Liste

Das Phänomen *Aufzählung ohne Liste* fällt ebenfalls unter das Kriterium *Sprachkomplexität*. „Aufzählungen, die innerhalb eines Satzes verwendet werden, erschweren das Textverständnis“ [tek11]. Ein Satz, welcher eine Aufzählung ohne Liste enthält, ist in Beispiel 44 zu sehen.

Beispiel 44 „Die aktuelle Situation auf dem Stellenmarkt und die Beschäftigungsentwicklung, Anforderungen an Bewerber, die Entwicklung im Berufsbild sowie die Aus- und Weiterbildung und formale Qualifikation Technischer Redakteure sind zentrale Fragen, deren Beantwortung zu den Zielsetzungen der tekomp gehört“ [tek11].

Der Satz in Beispiel 44 wird durch den Einschub der fünf Aufzählungselemente schwer lesbar. Der Leser benötigt eine bestimmte Zeit um die Menge an Information verarbeiten zu können. Die tekomp [tek11] empfiehlt, ab vier Aufzählungselementen eine Aufzählungsliste zu führen. In Beispiel 45 ist derselbe Inhalt wie in Beispiel 44 dargestellt. Anstatt die Aufzählungselemente in den Satz zu integrieren wird allerdings eine Liste verwendet.

Beispiel 45 „Zentrale Fragen, deren Beantwortung zu den Zielsetzungen der tekomp gehört:

- Die aktuelle Situation auf dem Stellenmarkt
- Die Beschäftigungsentwicklung
- Anforderungen an Bewerber
- Die Entwicklung im Berufsbild
- Aus- und Weiterbildung Technischer Redakteure“ [tek11].

Der Text in Beispiel 45 kann schneller vom Leser verarbeitet werden als der Text in Beispiel 44. Der Leser benötigt mehr Zeit, um einen Satz mit vielen Aufzählungselementen ohne Liste zu verstehen. Weiterhin besteht die Gefahr, dass bei langen Aufzählungen ohne Liste ein Element vom Leser versehentlich übergangen wird. Dieses Phänomen ist daher mittelkritisch und folglich im Qualitätsmodell gelb markiert.

4.2.4.5 Doppelte Negation

Ein weiteres Phänomen, welches in die Kategorie *Sprachkomplexität* fällt, ist die *Doppelte Negation*. In [tek11] wird empfohlen doppelte Verneinungen zu vermeiden. In [Sch12] wird ebenfalls davon abgeraten doppelte Verneinungen zu verwenden, weil sie beim Leser für Verwirrung sorgen.

Beispiel 46 *Der Brennbetrieb darf nicht autonom gestartet werden, wenn das aktive Auslöseereignis nicht angezeigt werden kann.*

In Beispiel 46 ist eine Anforderung, welche eine doppelte Negation enthält, zu sehen. Die beiden Verneinungen führen dazu, dass der Text mehrmals gelesen werden muss, um verstanden zu werden. In vielen Fällen können doppelte Verneinungen in positive Formulierungen umgewandelt werden, wie in Beispiel 47.

Beispiel 47 *Der Brennbetrieb darf nur dann autonom gestartet werden, wenn das aktive Auslöseereignis angezeigt werden kann.*

Die Anforderung in Beispiel 47 ist verständlich und drückt genau dieselbe Information aus wie die Anforderung in Beispiel 46. Doppelte Verneinungen stellen komplexere Strukturen als positiv formulierte Anforderungen dar. Man braucht länger, um sie zu verstehen, es kann auch zu Fehlinterpretationen kommen. Das Auftreten des Phänomens ist aber im Hinblick auf die Testbarkeit der entsprechenden Anforderung nicht immer negativ zu beurteilen. Das Phänomen ist folglich mittelkritisch bewertet und somit im Qualitätsmodell gelb markiert.

4.2.4.6 Pränominales Partizip

Pränominalpartizipien (in [Sch12] als Linkserweiterungen aufgeführt) haben ebenfalls einen Einfluss auf die Satzkomplexität. Das Phänomen der pränominalen Partizipien beschreibt Strukturen, bei welchen zwischen einem Artikel und dem syntaktisch zugehörigen Nomen ein Partizip eingeschoben wird, sowie ggf. auch Phrasen, die dem Partizip syntaktisch untergeordnet sind. In Beispiel 48 ist eine solche Konstruktion zu sehen.

Beispiel 48 *Der auf der Rückseite des Gehäuses über dem Lüftungsgitter angebrachte Reset-Schalter muss betätigt werden.*

Zwischen dem Artikel *Der* und dem syntaktisch zugehörigen Nomen *Reset-Schalter* wurden das Partizip *angebrachte* und die komplexe Präpositionalphrase *auf der Rückseite des Gehäuses über dem Lüftungsgitter* eingeschoben. Konstruktionen wie diese können das Textverständnis merklich erschweren. Konstruktionen dieser Art können in einfacher zu lesende Konstruktionen umgewandelt werden (Relativsätze, mehrere Hauptsätze), wie in Beispiel 49.

Beispiel 49 *Der Reset-Schalter, der auf der Rückseite des Gehäuses über dem Lüftungsgitter angebracht ist, muss betätigt werden.*

In Beispiel 49 wurde anstatt des pränominalen Partizips ein Relativsatz gebildet. Dies erleichtert das Textverständnis erheblich. Das Auftreten des sprachlichen Phänomens *Pränominales Partizip* ist nicht immer problematisch im Hinblick auf die Testbarkeit und ist

daher als mittelkritisch eingestuft und somit in der Graphik des Qualitätsmodells gelb markiert.

4.2.4.7 Komplexe Präposition

Auch das Phänomen der komplexen Präpositionen kann einen Einfluss auf die Sprachkomplexität eines Satzes haben. Eine komplexe Präposition ist in Beispiel 50 zu sehen.

Beispiel 50 *Die Auslegung der Bauteile erfolgt durch den Lieferanten unter Berücksichtigung von allen durch das Gesamtfahrzeug gegebenen Rahmenbedingungen.*

Die komplexe Präposition *unter Berücksichtigung von* führt dazu, dass eine lange Präpositionalphrase in den Satz eingebaut werden muss. Wandelt man die komplexe Präposition in das Verb *berücksichtigen* um, so sinkt der Grad der Satzkomplexität (vgl. Beispiel 51).

Beispiel 51 *Der Lieferant muss bei der Auslegung der Bauteile alle durch das Gesamtfahrzeug gegebenen Rahmenbedingungen berücksichtigen.*

Das Phänomen *Komplexe Präposition* hat einen Einfluss auf die Komplexität eines Satzes. Dieser Einfluss ist allerdings als sehr gering anzusehen. Das Phänomen ist aus diesem Grund als gering kritisch eingestuft und somit im Qualitätsmodell grün markiert.

4.2.4.8 Passiv mit Agensangabe

Das Phänomen *Passiv mit Agensangabe* ist ebenfalls dem Kriterium *Sprachkomplexität* zugeordnet. Ein Passivsatz mit Angabe des Agens ist zwar in jedem Fall besser als ein Passivsatz ohne Angabe des Agens, im Gegensatz zu einem Aktivsatz erhöht er allerdings die Satzkomplexität. In Beispiel 52 ist ein Passivsatz mit Agens zu sehen, in Beispiel 53 ist derselbe Satz im Aktiv zu sehen.

Beispiel 52 *Funktion B wird durch Funktion A aktiviert.*

Beispiel 53 *Funktion A aktiviert Funktion B.*

Aus den beiden Beispielen wird ersichtlich, dass in einem Passivsatz mit Agens im Gegensatz zu einem Aktivsatz die Sprachkomplexität unnötig erhöht wird. Allerdings schadet ein Passivsatz mit Agens nicht dem Verständnis des Lesers. Das Phänomen ist aus diesem Grund im Qualitätsmodell als gering kritisch eingestuft und somit grün markiert.

4.3 Testbarkeitskriterium Atomarität

Das dritte Testbarkeitskriterium im Qualitätsmodell für testbare Anforderungen ist die *Atomarität*. „Eine Anforderung ist atomar, wenn diese Anforderung einen isolierten Sachverhalt beschreibt. Eine Anforderung ist nicht atomar, wenn sie in mehrere Anforderungen untergliedert werden kann“ [Poh07]. Im Requirements Engineering ist es das Ziel, dass eine Aussage pro Anforderung beschrieben wird und somit die Anforderung im Idealfall durch einen Testfall abgedeckt werden kann. In [Rup14] wird daher empfohlen in jeder Anforderung lediglich ein Vollverb zu verwenden. Dem Testbarkeitskriterium *Atomarität* ist das Phänomen *Mehr als eine Handlung pro Satz* zugeordnet (vgl. Abbildung 15).

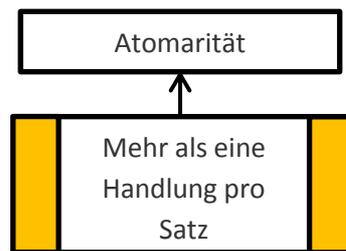


Abbildung 15: Qualitätskriterium Atomarität

Mehrere Handlungen pro Satz können beispielsweise durch das grammatische Mittel der Hauptsatzkoordinationen entstehen, bei welchem zwei Hauptsätze durch eine Konjunktion koordiniert werden. In Beispiel 54 ist eine solche Struktur zu sehen.

Beispiel 54 *Im Fehlerfall muss das System abgeschaltet werden, und das System muss den Fahrer darüber informieren.*

Mehrere Handlungen pro Satz können beispielsweise auch durch die Verwendung von Nebensätzen (vgl. Beispiel 55) oder durch die Verwendung von Infinitiven (vgl. Beispiel 56) entstehen.

Beispiel 55 *Die Funktion muss sicherstellen, dass sich das System abschaltet und dass es den Fahrer darüber informiert.*

Beispiel 56 *Das System muss sich abschalten und den Fahrer darüber informieren.*

In [tek11] wird empfohlen „mehrere nacheinander folgende Handlungsschritte nicht in einem Satz“ zu formulieren. Außerdem wird darauf hingewiesen, dass „eine Kernaussage pro Satz die Textverständlichkeit erleichtert“. In [Sch12] wird ebenfalls empfohlen eine Aussage pro Satz zu formulieren.

Das Auftreten des Phänomens *Mehr als eine Handlung pro Satz* ist mittelkritisch zu beurteilen, da es nur in bestimmten Fällen (wenn nicht klar ist wie viele Testfälle für die

Abdeckung der Anforderung benötigt werden) die Testbarkeit beeinflusst. Das Phänomen ist im Qualitätsmodell daher gelb markiert.

4.4 Testbarkeitskriterium Widerspruchsfreiheit

Das vierte Testbarkeitskriterium im Qualitätsmodell für testbare Anforderungen ist die *Widerspruchsfreiheit*. Dieses Kriterium beinhaltet drei Phänomene: *Globaler Widerspruch*, *Lokaler Widerspruch* und *Bezugsdefizit* (vgl. Abbildung 16). Tritt eines dieser Phänomene auf, ist das Testbarkeitskriterium *Widerspruchsfreiheit* verletzt und somit ist die entsprechende Anforderung nicht korrekt testbar. Die Kritikalitätsstufe der drei Phänomene ist die höchste.

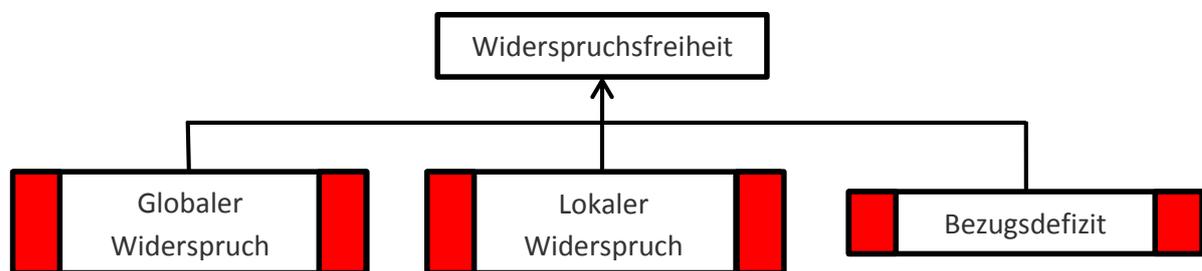


Abbildung 16: Qualitätskriterium Widerspruchsfreiheit

Ein globaler Widerspruch liegt vor, wenn sich der Inhalt einer Anforderung mit einer anderen Anforderung widerspricht. Ein globaler Widerspruch ist in Beispiel 57 dargestellt.

Beispiel 57 *A1: Die Länge der Nachlaufzeit beträgt 4 Sekunden.
A56: 1,5 Sekunden beträgt die Länge der Nachlaufzeit.*

In Beispiel 57 sind zwei Anforderungen zu sehen, welche sich im Inhalt widersprechen. Ein Auftreten eines solchen Widerspruchs ist in jedem Fall als kritisch anzusehen.

Bei lokalen Widersprüchen handelt sich um Widersprüche innerhalb einer Anforderung. Ein solcher Widerspruch ist in Beispiel 58 dargestellt.

Beispiel 58 *Kommt innerhalb von 2 Sekunden eine weitere steigende Flanke von A, wird die Funktion sofort abgebrochen und neu gestartet (D.h. die Paniktaste wird gedrückt. Bei < 1 Sekunde passiert nichts, bei ≥ 1 Sekunde wird die Funktion abgebrochen und neu gestartet).*

Die Zeitangaben in der Klammer stimmen nicht mit den Zeitangaben außerhalb der Klammer überein. Ein Auftreten eines solchen Widerspruchs ist ebenfalls immer als kritisch anzusehen.

Ein Bezugsdefizit tritt auf, wenn beispielsweise in einer Anforderung auf ein bestimmtes Dokument, auf ein bestimmtes Kapitel oder auf einen bestimmten Parameter referiert wird und das Dokument, das Kapitel oder der Parameter nicht auffindbar ist, oder wenn die Referenz auf das falsche Objekt verweist. Ein Bezugsdefizit ist daher ebenfalls immer als kritisch anzusehen.

4.5 Testbarkeitskriterium Inhaltliche Korrektheit

Das fünfte Testbarkeitskriterium im Qualitätsmodell für testbare Anforderungen ist die *Inhaltliche Korrektheit* (vgl. Abbildung 17). Ist eine Anforderung fehlerhaft beschrieben, kann kein eindeutiger bzw. nicht der intendierte Testfall abgeleitet werden. Das Auftreten dieses Phänomens ist daher der höchsten Kritikalitätsstufe zuzuordnen und aus diesem Grund im Qualitätsmodell rot markiert.

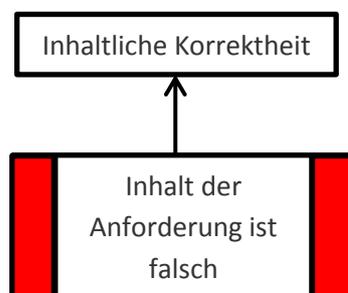


Abbildung 17: Qualitätskriterium Inhaltliche Korrektheit

4.6 Testbarkeitskriterium Durchführbarkeit

Das sechste Testbarkeitskriterium des Qualitätsmodells für testbare Anforderungen stellt das Kriterium der *Durchführbarkeit* dar. Damit eine Anforderung dieses Kriterium erfüllt, muss auf Grundlage der in den Anforderungen beschriebenen Funktionen ein Testfall abgeleitet werden können, welcher effektiv und effizient durchführbar ist. Dem Testbarkeitskriterium *Durchführbarkeit* sind daher zwei Phänomene zugeordnet: *Prüfung der Umsetzung der Anforderung ist nicht effektiv durchführbar* und *Prüfung der Umsetzung der Anforderung ist nicht effizient durchführbar*.

Das Phänomen *Prüfung der Umsetzung der Anforderung ist nicht effektiv durchführbar* tritt auf, wenn der Inhalt einer Anforderung mit den zur Verfügung stehenden Mitteln nicht überprüft werden kann oder die Anforderung aus technischer Sicht überhaupt nicht umsetzbar ist. Dies ist beispielsweise der Fall, wenn in einer Anforderung gefordert wird, dass Messungen durchgeführt werden sollen, die mit dem aktuellen Stand der Technik nicht erbracht werden können.

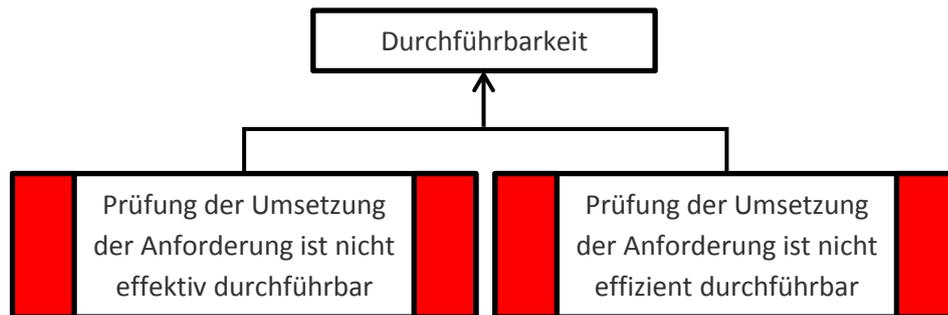


Abbildung 18: Qualitätskriterium Durchführbarkeit

Das Phänomen *Prüfung der Umsetzung der Anforderung ist nicht effizient durchführbar* kann auftreten, wenn beispielsweise der Zeitraum für einen Test zu groß ist. In Beispiel 59 ist eine solche Anforderung abgebildet.

Beispiel 59 *Das System muss 20 Jahre ohne Ausfall laufen.*

Ein effizienter Test für den Nachweis, dass ein System 20 Jahre lang ohne Ausfall läuft, kann nicht durchgeführt werden.

Ist eine Prüfung der Umsetzung einer Anforderung nicht effektiv oder nicht effizient durchführbar, ist die Anforderung auch nicht testbar. Die Phänomene, die dem Testbarkeitskriterium *Durchführbarkeit* zugeordnet sind, fallen in die höchste Kritikalitätsstufe und sind daher in der Graphik des Qualitätsmodell rot markiert.

4.7 Interpretation des Qualitätsmodells

Das präsentierte Qualitätsmodell für testbare Anforderungen wurde durch Sammlung und Strukturierung sprachlicher Phänomene aus unterschiedlichen Quellen entwickelt. Es wurden vier Quellen aus der Requirements Engineering-Literatur ([Rup14], [PR15], [Poh07], [Kam01]), zwei Daimler-interne Richtlinien ([Dai09a], [Dai09b]) und zwei Quellen aus dem Bereich der technischen Dokumentation ([tek11], [Sch12]) analysiert.

In Tabelle 5 sind die Phänomene aus dem Qualitätsmodell und die genannten Quellen aufgeführt. Weiterhin ist die Zuordnung der sprachlichen Phänomene zu den Quellen, in welchen sie genannt werden, dargestellt (+ = Phänomen wird genannt; - = Phänomen wird nicht genannt). Es handelt sich hierbei nur um die Phänomene, die einen Einfluss auf die Testbarkeit einer Anforderung haben können. Die Quellen enthalten teilweise zusätzliche Empfehlungen, wie verständliche Texte geschrieben werden (z.B. Strukturierung von Überschriften), jedoch stehen diese Empfehlungen nicht in direkter Verbindung mit der Testbarkeit einer Anforderung.

Phänomen	RE1	RE2	RE3	RE4	DAI1	DAI2	TD1	TD2
Messbarkeit								
Fehlen einer quantifizierenden Angabe	+	-	-	-	-	-	-	-
Verständlichkeit – Fehlende Information								
Nominalisierung	+	+	-	-	-	+	+	+
Weak-Word	+	+	+	+	+	-	-	-
Unpersönliche Konstruktion	-	-	-	-	-	-	+	-
Unvollständiger Satz	+	+	-	-	+	-	+	-
Unvollständige Bedingung	+	+	-	-	-	-	-	-
Syntaktisches Passiv ohne Agens	+	+	-	-	+	+	+	+
Passiversatz „sein“ + „zu“ + Infinitiv ohne Agens	-	-	-	-	-	-	+	-
Verständlichkeit – Verbindlichkeit								
Mangelnde Trennung zwischen Anf. und Inf.	-	-	-	-	+	+	-	-
Modalverb „sollte(n)“	+	+	-	-	+	+	+	-
Verallgemeinerung/Universalquantor	+	+	-	+	-	+	-	-
Verständlichkeit – Mehrdeutigkeit								
Polysemie	-	-	+	+	-	-	+	+
Langes Kompositum	-	-	-	-	-	-	+	-
Attribut vs. Adverbial	-	-	-	-	-	-	+	-
Gemischte Koordinationen	-	-	-	+	-	-	-	-
Koordination + Attribut	-	-	+	+	-	-	+	-
Präpositional- und Nominalphrasen	-	-	+	+	-	-	+	+
Genitivus subjectivus vs. objectivus	-	-	-	-	-	-	+	-
Subjekt vs. Objekt	-	-	-	-	-	-	+	-
Pronominaler Bezug	-	-	+	+	-	-	+	+
Verständlichkeit – Sprachkomplexität								
Verschachtelte Bedingungen	-	-	-	-	-	-	-	-
Verschachtelter/Zu langer Satz	+	-	-	-	-	-	+	+
Langer Klammereinschub	-	-	-	-	-	-	+	-
Aufzählung ohne Liste	-	-	-	-	-	-	+	-
Doppelte Negation	-	-	-	-	-	-	+	+
Pränominales Partizip	-	-	-	-	-	-	-	+
Komplexe Präposition	-	-	-	-	-	-	-	+
Passiv mit Agensangabe	-	-	-	-	-	-	+	-
Atomarität								
Mehr als eine Handlung pro Satz	+	-	-	-	+	+	+	+
Widerspruchsfreiheit								
Globaler Widerspruch	-	-	-	-	-	-	-	-
Lokaler Widerspruch	-	-	-	-	-	-	-	-
Bezugsdefizit	-	-	-	-	-	-	-	-
Inhaltliche Korrektheit								
Inhalt der Anforderung ist falsch	-	-	-	-	-	-	-	-
Durchführbarkeit								
Prüf. der Ums. der Anf. nicht effektiv durchführbar	+	-	-	-	-	-	-	-
Prüf. der Ums. der Anf. nicht effizient durchführbar	-	-	-	-	-	-	-	-

Tabelle 5: Zuordnung der Phänomene aus dem Qualitätsmodell zu den untersuchten Quellen (RE1: Rupp 2014 [Rup14], RE2: Pohl/Rupp [PR15], RE3: Pohl [Poh07], RE4: Kamsties [Kam01], DAI1= Daimler-interne Schulung [Dai09a], DAI2: Daimler-interne Leitlinie [Dai09b], TD1: tekcom [tek11], TD2: Duden [Sch12])

Die Literatur zum Requirements Engineering nennt hauptsächlich Phänomene, die in den Bereich der fehlenden Information (vorwiegend [Rup14] und [PR15]) oder in den Bereich der Mehrdeutigkeiten (vorwiegend [Poh07] und [Kam01]) fallen. Auch die Literatur zur technischen Dokumentation nennt Phänomene, die in diese Bereiche eingeordnet werden können. Auffällig ist auch, dass sich die Literatur zur technischen Dokumentation stark auf das Thema der Komplexität fokussiert. Die Literatur aus dem Requirements Engineering hingegen ist in diesem Bereich so gut wie gar nicht vertreten. Die einzige Ausnahme bildet [Rup14], wo zu lange Sätze als Problem genannt werden. Vermutlich hat dies mit der

unterschiedlichen Kommunikationssituation zu tun: Im Requirements Engineering werden Anforderungen von Experten für Experten geschrieben, in der technischen Dokumentation hingegen werden Texte von Experten für Laien geschrieben. In der technischen Dokumentation ist es deshalb um so wichtiger weniger komplexe Sätze zu schreiben, welche durch Laien, die sich nicht in der täglichen Arbeit mit einem bestimmten Thema beschäftigen, schnell korrekt interpretiert werden können. Die Leser von Anforderungen haben meist einen speziellen Wissenshintergrund, sogenanntes Domänenwissen. Durch dieses Wissen können komplexe Sachverhalte schneller verstanden werden. Komplexe Strukturen sind im Requirements Engineering daher als weniger kritisch zu betrachten.

Widersprüche und die inhaltliche Korrektheit werden in den Quellen nicht explizit aufgezählt. Vermutlich wird Widerspruchsfreiheit und inhaltliche Korrektheit von den Quellen vorausgesetzt, sodass diese Aspekte nicht explizit genannt werden. Das Kriterium der Durchführbarkeit wird lediglich in [Rup14] erwähnt. Jedoch ist es sinnvoll, diese Kriterien im Qualitätsmodell zu verorten.

Die in der Requirements Engineering-Literatur genannten Phänomene überschneiden sich großteils mit den Phänomenen, welche in der Literatur zur technischen Dokumentation aufgeführt werden. Da die Kommunikationssituation bei der technischen Dokumentation eine andere ist als bei der Dokumentation von Anforderungen, sind die aufgeführten Phänomene aber nicht komplett deckungsgleich.

Das Qualitätsmodell für testbare Anforderungen enthält Phänomene, die in der oben genannten Literatur zum Requirements Engineering und zur technischen Dokumentation aufgeführt werden. Die Vollständigkeit des Qualitätsmodells kann lediglich in Bezug auf die beschriebenen Quellen garantiert werden. Es besteht durchaus die Möglichkeit weitere Phänomene in das Qualitätsmodell zu integrieren und nach Kritikalität zu bewerten.

4.8 Vorgehen bei der Auswahl relevanter Phänomene für die geplanten linguistischen Analysen und Regelformulierungen

Ein Ziel der vorliegenden Arbeit ist es, Regeln zu entwerfen, mit welchen kritische Instanzen ausgewählter Phänomene, die einen Einfluss auf die Testbarkeit haben können, automatisiert identifiziert werden können. Anhand der Kritikalitätseinschätzung der Phänomene im Qualitätsmodell wurde bereits gezeigt, dass die Phänomene eine unterschiedlich hohe Relevanz für die weitere Behandlung aufweisen. Da nur die als relevant identifizierten Phänomene einer tiefergehenden Analyse unterzogen werden sollen, müssen mehrere Auswahlkriterien beachtet werden, welche im Folgenden näher erläutert werden. Diese

Auswahlkriterien sind die Kritikalität, die Häufigkeit, die generelle Erkennbarkeit eines Phänomens und die zuverlässige Erkennbarkeit der kritischen Instanzen eines Phänomens.

Das erste Kriterium betrachtet den Kritikalitätsgrad eines Phänomens. Ist die Kritikalität niedrig eingestuft (= grüne Markierung im Qualitätsmodell), wird das Phänomen keiner näheren Untersuchung unterzogen. Die kritischen Instanzen dieser Phänomene weisen eine nur sehr niedrige Frequenz auf, sodass eine automatische Analyse nicht zielführend ist, weil die kritischen Instanzen nicht gezielt durch eine automatisierte Analyse herausgefiltert werden können. Das Phänomen *Langes Kompositum* ist beispielsweise nur in sehr seltenen Fällen kritisch und hat daher nur einen sehr geringen Einfluss auf die sprachliche Qualität der Lastenhefttexte. Alle Phänomene, die eine niedrige Kritikalität aufweisen, werden für die bevorstehenden Analysen ausgeschlossen.

Das zweite Auswahlkriterium betrachtet die Häufigkeit eines Phänomens in den erstellten Lastenheftkorpora (vgl. Kapitel 6). Um eine grobe Einschätzung über die Häufigkeit der Phänomene zu erhalten, wurden die Phänomene aus dem Qualitätsmodell anhand genereller Regeln in den Korpora identifiziert. Das Phänomen *Pränominales Partizip* kommt beispielsweise vergleichsweise selten in den Korpora vor. Eine detaillierte Untersuchung dieses Phänomens wird daher nicht durchgeführt. Widersprüche, die inhaltliche Korrektheit, unvollständige Bedingungen und weitere Phänomene, können durch reine Korpusabfragen nicht gezählt werden. Aus diesem Grund kann bei diesen Phänomenen keine Aussage über die Häufigkeit getroffen werden. Die Phänomene, bei welchen eine Abschätzung über die Häufigkeit getroffen werden kann und die sehr selten in den Korpora vorkommen, werden von der Analyse ausgeschlossen.

Ein weiteres relevantes Auswahlkriterium ist die automatisierte Erkennbarkeit eines Phänomens. Hierbei muss zwischen zwei Arten der Erkennbarkeit unterschieden werden: die generelle Erkennbarkeit eines Phänomens und die zuverlässige Erkennbarkeit der kritischen Instanzen eines Phänomens. Ein Phänomen ist generell erkennbar, wenn das Phänomen automatisiert mit den in dieser Arbeit verwendeten formbasierten computerlinguistischen Technologien (vgl. hierzu Kapitel 6) erkannt werden kann, unabhängig davon, ob nur die kritischen oder auch die unkritischen Instanzen gefunden werden. Können Regeln entworfen werden, die in den meisten Fällen zuverlässig zwischen kritischen und unkritischen Instanzen eines Phänomens unterscheiden können, ist das Phänomen auch zuverlässig erkennbar. Weak-Words sind beispielsweise generell erkennbar und die kritischen Instanzen können zuverlässig von den unkritischen Instanzen unterschieden werden. Die zuverlässige Erkennbarkeit der kritischen Instanzen konnte bereits durch Vorarbeiten festgestellt werden (vgl. hierzu [Kri13]). Die generelle Erkennbarkeit kann bei allen Phänomenen festgelegt werden. Die zuverlässige Erkennbarkeit der kritischen Instanzen eines

Phänomens hingegen kann ohne tiefgehende Analysen nicht bei allen Phänomenen festgelegt werden. Bei bestimmten Phänomenen kann jedoch ohne Voranalysen die Aussage getroffen werden, dass die kritischen Instanzen jener Phänomene nicht zuverlässig mit der bereitgestellten formbasierten computerlinguistischen Technologie erkannt werden können.

Phänomen	Kritikalität	Häufigkeit	Generelle Erkennbarkeit	Zuverlässige Erkennbarkeit
Messbarkeit				
Fehlen einer quantifizierenden Angabe	↑	↑	→	→
Verständlichkeit – Fehlende Information				
Nominalisierung	→	↑	↑	↓
Weak- Word	→	↑	↑	↑
Unpersönliche Konstruktion	→	↑	↑	-
Unvollständiger Satz	→	↑	→	↓
Unvollständige Bedingung	→	-	→	↓
Syntaktisches Passiv ohne Agens	→	↑	↑	-
Passiversatz „sein“ + „zu“ + Infinitiv ohne Agens	→	↑	↑	-
Verständlichkeit – Verbindlichkeit				
Mangelnde Trennung zwischen Anf. und Inf.	↑	-	↓	↓
Modalverb „sollte(n)“	→	↑	↑	-
Verallgemeinerung/Universalquantor	↓	↑	↑	↓
Verständlichkeit – Mehrdeutigkeit				
Polysemie	↑	-	↓	↓
Langes Kompositum	↓	↑	↑	↓
Attribut vs. Adverbial	↑	↓	↓	↓
Gemischte Koordinationen	↑	↓	↑	-
Koordination + Attribut	→	↑	↑	↓
Präpositional- und Nominalphrasen	→	↑	↓	↓
Genitivus subjectivus vs. objectivus	→	-	↓	↓
Subjekt vs. Objekt	→	-	↓	↓
Pronominaler Bezug	→	↑	↑	-
Verständlichkeit – Sprachkomplexität				
Verschachtelte Bedingungen	→	↓	↑	↓
Verschachtelter/Zu langer Satz	→	↑	↑	↓
Langer Klammereinschub	→	↓	↑	↓
Aufzählung ohne Liste	→	↑	↑	↓
Doppelte Negation	→	↓	↑	↓
Pränominales Partizip	→	↓	↑	-
Komplexe Präposition	↓	↑	↑	↓
Passiv mit Agensangabe	↓	↑	↑	↓
Atomarität				
Mehr als eine Handlung pro Satz	→	-	↑	↓
Widerspruchsfreiheit				
Globaler Widerspruch	↑	-	↓	↓
Lokaler Widerspruch	↑	-	↓	↓
Bezugsdefizit	↑	-	↓	↓
Inhaltliche Korrektheit				
Inhalt der Anforderung ist falsch	↑	-	↓	↓
Durchführbarkeit				
Prüf. der Ums. der Anf. nicht effektiv durchführbar	↑	-	↓	↓
Prüf. der Ums. der Anf. nicht effizient durchführbar	↑	-	↓	↓

Tabelle 6: Einstufung der Phänomene aus dem Qualitätsmodell

Die Phänomene aus dem Qualitätsmodell sowie die beschriebenen Auswahlkriterien sind in Tabelle 6 abgebildet. Für jedes Phänomen ist die Stufe der Kritikalität analog zum Qualitätsmodell angegeben (hoch=↑, mittel=→, niedrig=↓), die Phänomene sind dahingehend markiert, ob sie häufig (=↑) oder eher selten (=↓) vorkommen, eine Angabe darüber, ob die Phänomene generell erkennbar (ja=↑, nein=↓) sind und ob die kritischen Instanzen

zen eines Phänomens zuverlässig automatisiert erkannt werden können (ja= \uparrow , nein= \downarrow), ist ebenfalls gegeben. Für bestimmte Phänomene kann in manchen Fällen keine Aussage über bestimmte Auswahlkriterien getroffen werden. Dies ist beispielsweise der Fall bei dem Auswahlkriterium Häufigkeit des Phänomens *Inhalt der Anforderung falsch*. Metadaten, die Auskunft darüber geben, ob eine Anforderung inhaltlich falsch ist, sind in den Korpora nicht vorhanden, daher ist dieses Phänomen auch nicht zählbar (in Tabelle 6 durch -markiert). Ebenso verhält es sich bei den Phänomenen, welche dem Testbarkeitskriterium *Widerspruchsfreiheit* und dem Testbarkeitskriterium *Durchführbarkeit* zugeordnet sind.

Die zuverlässige Erkennbarkeit der kritischen Instanzen eines Phänomens kann zu diesem Zeitpunkt der Analysen nur bei denjenigen Phänomenen angegeben werden, bei welchen ohne weitere Untersuchungen klar ist, dass eine zuverlässige Analyse mit der formbasierten computerlinguistischen Technologie nicht möglich ist. Bei den anderen Phänomenen kann zu dieser Eigenschaft noch keine Aussage getroffen werden. Eine Ausnahme bilden die Weak-Words, da hierfür schon, wie bereits weiter oben erwähnt, Vorarbeiten geleistet wurden.

Für die weiteren Analysen sind die Phänomene relevant, die eine mittlere oder hohe Kritikalität aufweisen, häufig in den Anforderungstexten vorkommen und für die Sprachprüfungen entworfen werden können, die nicht nur generell die Phänomene identifizieren, sondern die kritischen Instanzen zuverlässig identifizieren. Phänomene, bei welchen ohne weitere Analysen keine Aussage darüber getroffen werden kann, ob die kritischen Instanzen zuverlässig identifiziert werden können und nicht durch die anderen Auswahlkriterien ausgeschlossen werden, werden ebenfalls weiter betrachtet.

Durch die Auswahlkriterien konnten folgende Phänomene als relevant identifiziert werden (in Tabelle 6 fett markiert):

- (1) *Fehlen einer quantifizierenden Angabe,*
- (2) *Weak-Word,*
- (3) *Unpersönliche Konstruktion,*
- (4) *Syntaktisches Passiv ohne Agens,*
- (5) *Passiversatz „sein“ + „zu“ + Infinitiv ohne Agens,*
- (6) *Modalverb „sollte(n)“,*
- (7) *Pronominaler Bezug.*

Da das Phänomen *Fehlen einer quantifizierenden Angabe* zu der Kategorie der *Weak-Words* quer liegt (vgl. Abschnitt 4.2.1.2), werden die beiden Analysen zusammengelegt. Die Passivanalysen werden ebenfalls zu einer Analyse vereint.

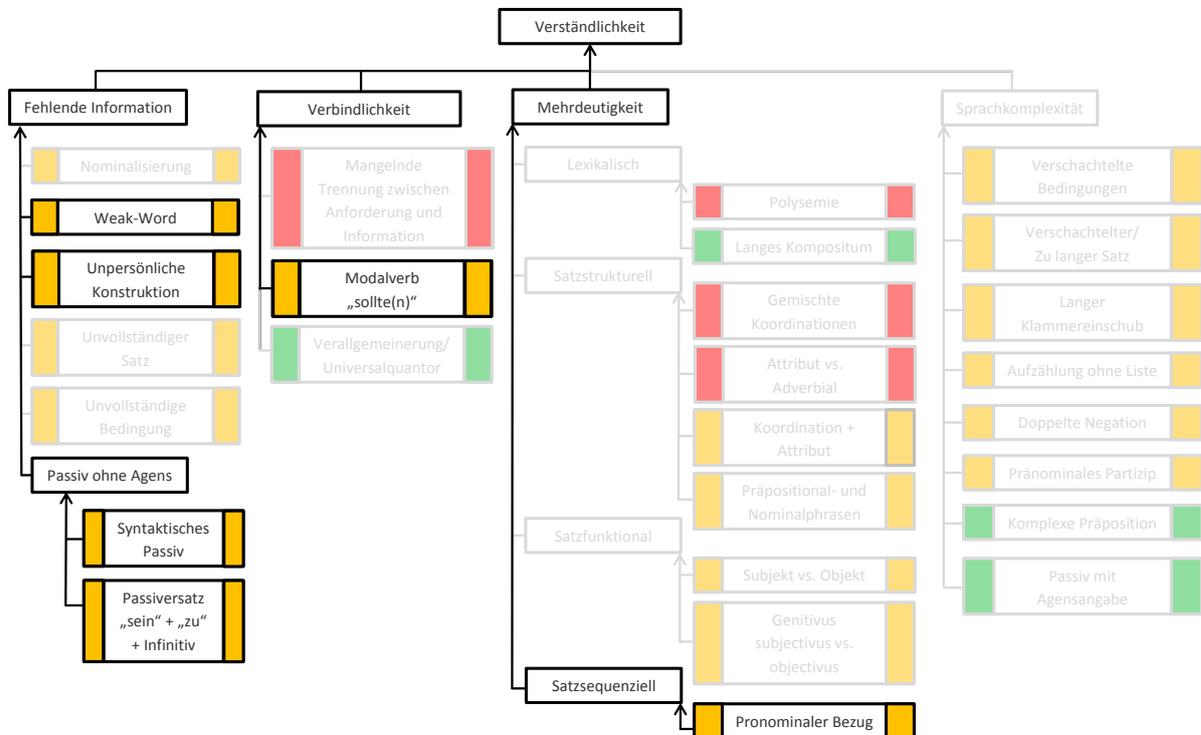


Abbildung 19: Relevante Phänomene des Testbarkeitskriteriums Verständlichkeit

In Abbildung 19 werden die als relevant identifizierten Phänomene des Testbarkeitskriteriums *Verständlichkeit* nochmals graphisch hervorgehoben. In den geplanten linguistischen Analysen werden demnach Phänomene aus dem Bereich der fehlenden Information, der Verbindlichkeit und der Mehrdeutigkeit analysiert.

4.9 Zusammenfassung

Es wurde ein Qualitätsmodell für testbare Anforderungen vorgestellt, welches sprachliche Phänomene, die einen Einfluss auf die Testbarkeit einer Anforderung haben können, aus Literatur zum Requirements Engineering und zur technischen Dokumentation zusammenträgt. Das Qualitätsmodell gliedert sich in sechs Testbarkeitskriterien: *Messbarkeit*, *Verständlichkeit*, *Atomarität*, *Widerspruchsfreiheit*, *Inhaltliche Korrektheit* und *Durchführbarkeit*. Die den Testbarkeitskriterien untergeordneten Phänomene sind im Qualitätsmodell nach linguistischen Aspekten sortiert und nach Kritikalität bewertet.

Für die Auswahl relevanter Phänomene für tiefergehende Analysen wurden Auswahlkriterien vorgestellt. Diese Auswahlkriterien sind der Grad der Kritikalität eines Phänomens,

die Häufigkeit eines Phänomens in den zu analysierenden Korpora sowie die generelle Erkennbarkeit eines Phänomens und die effektive Erkennbarkeit der kritischen Instanzen.

Anhand dieser Klassifikation wurde eine Auswahl darüber getroffen, welche Phänomene in der vorliegenden Arbeit Gegenstand von linguistischen Analysen und Regelformulierungen zur Prüfung von Anforderungen sein sollen.

5 Methoden und Werkzeuge zur Sicherstellung der Qualität in Anforderungen

Es gibt unterschiedliche Ansätze, um gute natürlichsprachliche Anforderungen zu schreiben. In der Literatur findet man viele Leitfäden, in welchen Regeln enthalten sind, an die sich der Autor beim Verfassen von Anforderungen halten soll. Dem Autor werden in solchen Leitfäden generelle Anweisungen gegeben. Eine weitere Technik, bei welcher Satzmuster vorgegeben werden, sind Satzschablonen oder sogenannte Templates. Diese Satzschablonen geben dem Autor Schritt für Schritt vor, welche Satzteile er nacheinander verwenden darf. Auch kontrollierte Sprachen bedienen sich solcher Satzbaumuster; außerdem schlagen sie in der Regel ein eingeschränktes Vokabular vor. Durch die drei genannten Techniken sollen schon beim Schreiben mögliche Ambiguitäten umgangen werden.

Manuelle Reviews dienen der Kontrolle bereits geschriebener Anforderungen und stellen eine weitere Technik für das Schreiben guter Anforderungen dar. Außerdem existieren automatische Analysen, welche bereits geschriebene Anforderungen kontrollieren und dem Autor Rückmeldung darüber geben, welche Anforderungen potentiell mehrdeutig sind.

Diese Techniken, Satzschablonen, kontrollierte Sprachen, Schreibregeln, manuelle Reviews und Werkzeuge zur automatischen Kontrolle bestehender Anforderungen, werden in den folgenden Abschnitten näher beschrieben. Außerdem werden für jede Technik die Vor- und Nachteile beleuchtet.

5.1 Schreibregeln

In der Literatur zum Requirements Engineering finden sich Schreibregeln für die Formulierung guter Anforderungen. Diese Regeln sollen bereits beim Schreiben der Anforderungen helfen, relevante Informationen zu nennen und Ambiguitäten weitestgehend zu vermeiden. Eine der am häufigsten genannten Regeln in dieser Art von Literatur zielt auf die Vermeidung des Passivs bzw. die Verwendung des Aktivs (z.B. in [PR15], [Wun15], [Rup14], [Wie99]), da das Fehlen einer expliziten Nennung des Agens zu Verständnisproblemen führen kann (vgl. Abschnitt 4.2.1.6 und Abschnitt 8.3). Der Autor wird in den Regeln dazu aufgefordert stets im Aktiv zu schreiben. Ein weiteres Problem, das häufig genannt wird, ist die Verwendung von Nominalisierungen, da deren Argumente, anders als bei verbalen Ausdrücken, optional sind und somit nicht genannt werden müssen (vgl. Abschnitt 4.2.1.1). Wann immer möglich, soll der Autor anstatt von Nominalisierungen Verben verwenden. Vage Ausdrücke wie *lang*, *schnell* usw., d.h. Weak-Words (vgl. Abschnitt 4.2.1.2 und Abschnitt 8.1), sollen ebenfalls vermieden werden [Wie99].

In [Rup14] wird ein Regelwerk präsentiert, welches insgesamt 18 Schreibregeln für Anforderungen in deutscher Sprache umfasst. Auch hier wird die Problematik des Passivs und von Nominalisierungen aufgelistet. Hinzu kommen weitere Regeln, welche sich nicht nur auf die Wortebene, sondern auch auf die Satzebene beziehen. Weitere Regelwerke wurden bereits in Kapitel 4 präsentiert.

Ein Regelwerk für das Schreiben von Anforderungen kann dem Autor dabei helfen qualitativ bessere Anforderungen zu schreiben. Bestimmte sprachliche Phänomene, wie beispielsweise die Verwendung des Passivs, können durch diese Technik vermieden werden, und so können die damit ggf. zusammenhängenden Probleme ausgeschlossen werden. Jedoch muss der Autor das Regelwerk erlernen und sich stets daran halten. Fraglich ist auch, ob ein Regelwerk jemals vollständig sein kann und somit jede Formulierung, die Probleme bereiten kann, durch mindestens eine Regel abgedeckt wird. Eine nachträgliche Prüfung der Anforderungen ist in jedem Falle nötig, da die Kontrolle bei der Texterstellung vollständig beim Autor liegt.

5.2 Satzschablonen

Schablonenbasierte Ansätze wie in [Rup14] geben an, wie man Schritt für Schritt dabei vorgehen kann, gute Anforderungen zu schreiben. Die Schablonen geben vor, in welcher Reihenfolge welche Aspekte einer Anforderung beschrieben werden sollen, sie fungieren folglich als ein syntaktischer Bauplan für Anforderungen. Ein Ziel bei der Verwendung von Schablonen ist, dass ein erheblicher Anteil typischer Formulierungsfehler ausgeschlossen werden können [Rup14], da durch die Schablonen eine gewisse Satzstruktur vorgegeben wird. So ist es beispielsweise nicht möglich im Passiv zu formulieren, da das Agens des Verbs stets genannt werden muss. Der Grad der Verbindlichkeit muss ebenfalls für jede Anforderung festgelegt werden. Ein weiteres Ziel der Verwendung von Schablonen ist es, den Anforderungen eine einheitliche Struktur zu verleihen. Aufgrund internationaler Vergaben von Lastenheften müssen diese übersetzt werden. Wenn Anforderungen eine einheitliche Struktur haben, also immer einem bestimmten satzstrukturellen Muster folgen, sinkt auch der Übersetzungsaufwand.

Der Ansatz der SOPHISTen ist der am weitesten verbreitete für deutschsprachige Anforderungen und wird deshalb in dieser Arbeit ausführlich besprochen. In Abbildung 20 ist eine Schablone ohne Bedingung zu sehen [Rup14], [Die13].

Der Autor muss zu Beginn einer Anforderung immer den Handelnden oder die ausführende Komponente bzw. das ausführende System angeben, also deutlich machen, wer die beschriebene Aktion ausführen soll. Hier kann auch der Auftragnehmer genannt werden. Darauf folgend muss der Autor zwischen drei Verben wählen, welche für unterschiedliche

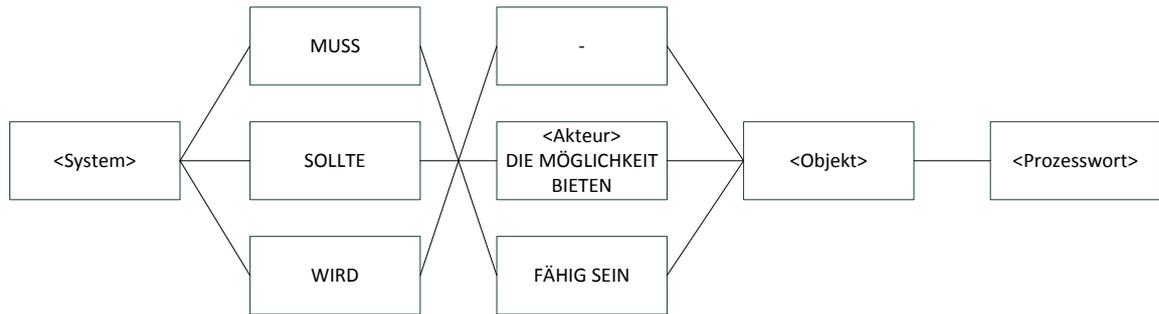


Abbildung 20: Satzschablone ohne Bedingung nach [Rup14]

Grade an Verbindlichkeit stehen. Anforderungen, die das Wort *muss* bzw. *müssen* enthalten, sind rechtlich verbindlich und müssen durch den Auftragnehmer oder durch die vom Auftragnehmer realisierte Komponente bzw. durch das System erfüllt werden. Das Verb *sollte* wird in Anforderungen verwendet, die Wünsche äußern. Die zugehörigen Inhalte müssen vom Auftragnehmer nicht erfüllt werden, da sie nicht rechtlich verbindlich sind. Wenn es das Budget zulässt, kann der Auftragnehmer solche Wünsche zusätzlich zum obligatorischen Leistungsumfang realisieren. Anforderungen mit dem Verb *wird* geben Informationen über Funktionen, die in der Zukunft geplant sind. Diese Anforderungen sind vom Auftragnehmer verpflichtend zu beachten und „helfen in der aktuellen Lösung, Vorbereitungen zu treffen, um Zukünftiges später optimal zu integrieren“ [Rup14]. Nachdem der Autor eines der drei Verben gewählt hat, muss er die Art der Funktionalität angeben. Er muss angeben, ob es sich um eine selbstständige Systemaktivität (obere Option), eine Benutzerinteraktion (mittlere Option) oder um eine Schnittstellenanforderung (untere Option) handelt. Anschließend muss der Autor noch das Objekt, um das es in der Anforderung geht, benennen und ein Prozesswort (= Vollverb) anschließen.

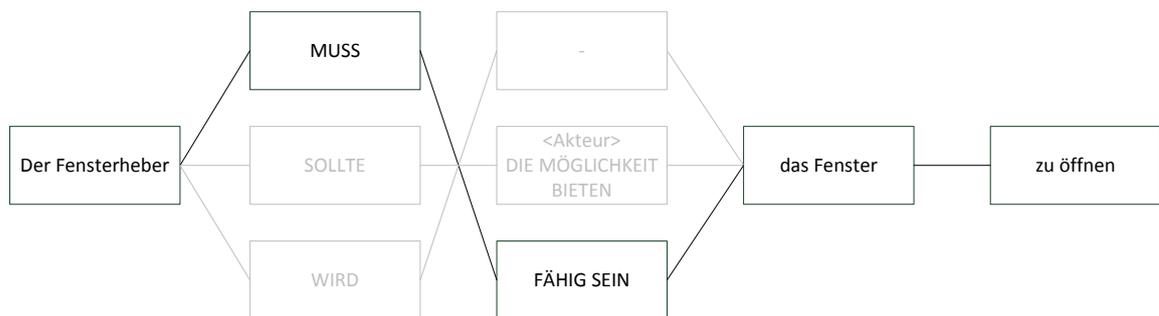


Abbildung 21: Ausgefüllte Satzschablone ohne Bedingung nach [Rup14]

In Abbildung 21 ist eine ausgefüllte Schablone dargestellt. Diese beschreibt eine Funktionalität eines Fensterhebers. Die Schablone kann auf alle möglichen Arten von Anforderungen angewandt werden und ist nicht auf eine Domäne beschränkt.

Für Anforderungen mit Bedingungen gibt es eine erweiterte Schablone (vgl. Abbildung 22). Da die Konjunktion *wenn* als Einleitung einer logischen Bedingung oder als Einleitung einer temporalen Bedingung dienen kann, wird in [Rup14] empfohlen für logische Bedingungen die Konjunktion *Falls* zu verwenden und für temporale Bedingungen die beiden Konjunktionen *Sobald* oder *Solange*.

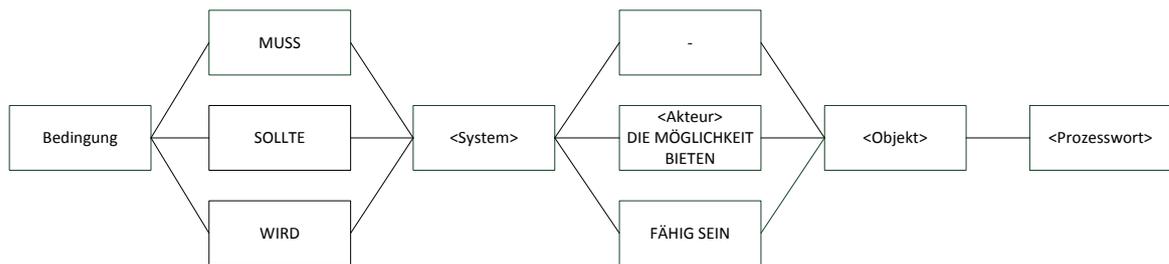


Abbildung 22: Satzschablone mit Bedingung nach [Rup14]

Für nicht-funktionale Anforderungen werden in [Rup14] entsprechende Schablonen aufgeführt, die hier aber nicht weiter ausgeführt werden. Weiterhin wird empfohlen Glossare zu pflegen, in welchen Wörter, deren Bedeutungen und Synonyme erfasst werden.

Durch die Anwendung des Schablonenansatzes, welcher in [Rup14] präsentiert wird, können bestimmte sprachliche Phänomene bei der Formulierung einer Anforderung ausgeschlossen werden. Wie bereits weiter oben erwähnt, ist es nicht möglich im Passiv zu formulieren. Um zu zeigen, welche weiteren Phänomene aus dem Qualitätsmodell (vgl. Kapitel 4) bei der Verwendung dieser Schablonen ausgeschlossen werden können und welche Phänomene weiterhin auftreten können, wurde ein Experiment durchgeführt. Die Ergebnisse sind in Tabelle 7 abgebildet.

In Tabelle 7 ist durch Pfeile (\uparrow = ja, \downarrow = nein) markiert, ob ein bestimmtes Phänomen aus dem Qualitätsmodell bei der Verwendung einer der beiden präsentierten Schablonen auftreten kann. Zusätzlich ist angegeben, ob die Phänomene durch das Regelwerk in [Rup14] (= (R)) oder durch den Einsatz von Glossaren (= (G)) ausgeschlossen werden. Universalquantoren beispielsweise werden durch die Struktur der Schablonen nicht verboten, jedoch beschreibt das Regelwerk den Umgang mit Quantoren. Dieses Phänomen sollte also zu keinen Problemen führen. Pronominale Bezüge können ebenfalls bei der Formulierung verwendet werden und werden nicht durch das Regelwerk abgedeckt. Dieses Phänomen kann folglich zu Problemen führen. Im Regelwerk wird zwar darauf hingewie-

Phänomen	Schablone ohne Bedingung	Schablone mit Bedingung
Messbarkeit		
Fehlen einer quantifizierenden Angabe	↑ (R)	↑ (R)
Verständlichkeit – Fehlende Information		
Nominalisierung	↑ (G,R)	↑ (G,R)
Weak-Word	↑ (G,R)	↑ (G,R)
Unpersönliche Konstruktion	↓	↑
Unvollständiger Satz	↓ (R)	↓ (R)
Unvollständige Bedingung	↓ (R)	↑ (R)
Syntaktisches Passiv ohne Agens	↓ (R)	↑ (R)
Passiversatz „sein“ + „zu“ + Infinitiv ohne Agens	↓	↑
Verständlichkeit – Verbindlichkeit		
Mangelnde Trennung zwischen Anf. und Inf.	↓	↓
Modalverb „sollte(n)“	↓ (G)	↓ (G)
Verallgemeinerung/Universalquantor	↑ (R)	↑ (R)
Verständlichkeit – Mehrdeutigkeit		
Polysemie	↑ (G)	↑ (G)
Langes Kompositum	↑ (G)	↑ (G)
Attribut vs. Adverbial	↑	↑
Gemischte Koordinationen	↑	↑
Koordination + Attribut	↑	↑
Präpositional- und Nominalphrasen	↑	↑
Genitivus subjectivus vs. objectivus	↑	↑
Subjekt vs. Objekt	↓	↑
Pronominaler Bezug	↑	↑
Verständlichkeit – Sprachkomplexität		
Verschachtelte Bedingungen	↓	↑
Verschachtelter/Zu langer Satz	↓	↑
Langer Klammereinschub	↓ (R)	↓ (R)
Aufzählung ohne Liste	↑	↑
Doppelte Negation	↓	↓
Pränominales Partizip	↑	↑
Komplexe Präposition	↑	↑
Passiv mit Agensangabe	↓	↑
Atomarität		
Mehr als eine Handlung pro Satz	↓	↓
Widerspruchsfreiheit		
Globaler Widerspruch	↑	↑
Lokaler Widerspruch	↓	↑
Bezugsdefizit	↑	↑
Inhaltliche Korrektheit		
Inhalt der Anforderung ist falsch	↑	↑
Durchführbarkeit		
Prüf. der Umsetz. der Anf. nicht effektiv durchführbar	↑	↑
Prüf. der Umsetz. der Anf. nicht effizient durchführbar	↑	↑

Tabelle 7: Überblick, welche Phänomene bei der Verwendung von Satzschablonen aus [Rup14] auftreten können (↑ = Phänomen kann auftreten; ↓ = Phänomen kann nicht auftreten; (R) = Phänomen wird durch Regelwerk ausgeschlossen; (G) = Phänomen wird durch Verwendung eines Glossars ausgeschlossen)

sen, dass Weak-Words nicht verwendet werden sollen, jedoch wird keine vollständige Liste aufgeführt. Weak-Words können in den Schablonen verwendet werden und auch dort dazu führen, dass für das Verständnis wichtige Information ungenannt bleibt.

Durch die Schablonen ist das Modalverb *sollte* klar definiert. Es ist eindeutig, dass es sich bei Anforderungen, die das Wort *sollte* enthalten, um optionale Anforderungen handelt. Es besteht hier also keine Gefahr von Missverständnissen durch das Vorhandensein zweier Lesarten, da grundsätzlich die dispositionelle Lesart vorliegt (vgl. Abschnitt 4.2.2.2).

Die Schablone ohne vorausgehende Bedingung schränkt die Phänomene, die potentiell auftreten können, stark ein. Anders verhält es sich bei der Schablone mit Bedingung. Da der Aufbau der Bedingungen nicht exakt vorgegeben ist, können dort viele der im Qualitätsmodell als problematisch identifizierten Phänomene auftreten.

Durch die Anwendung der präsentierten Schablonen und die Anwendung des Regelwerks sowie durch die Arbeit mit Glossaren können bestimmte Phänomene, die im Qualitätsmodell aufgeführt sind und die einen Einfluss auf die Testbarkeit einer Anforderung haben können, ausgeschlossen werden (vgl. ↓ in Tabelle 7). Jedoch können einige der im Qualitätsmodell als problematisch eingestuften Phänomene weiterhin auftreten (vgl. ↑ in Tabelle 7) und dazu führen, dass eine Anforderung falsch interpretiert wird.

Der bekannteste Ansatz für die Formulierung von englischsprachigen Anforderungen mittels Schablonen ist der *EARS*-Ansatz. EARS steht für “Easy Approach to Requirements Syntax” [MW10]. Der EARS-Ansatz liefert fünf Schablonen für die Formulierung von Anforderungen. Im Folgenden wird eine dieser fünf Schablonen genauer diskutiert.

Die Schablone vom Typ *Ubiquitous* (= universell) wird verwendet, wenn der beschriebene Sachverhalt einer Anforderung keine Vorbedingung und keinen Auslöser voraussetzt. Die Anforderungen, die nach dieser Schablone geschrieben sind, gelten als immer aktiv. Die Schablone hat die folgende Form: **The <system name> shall <system response>**. Die Teile, die in spitzen Klammern stehen, sollen vom Autor entsprechend ersetzt werden. Dies könnte wie in Beispiel 60 aussehen.

Beispiel 60 *The control system shall indicate the engine oil quantity to the aircraft. [MW10]*

Die vier weiteren Schablonen werden verwendet, um folgende Sachverhalte zu beschreiben: *Event-Driven* (= ereignisgesteuert), *Unwanted Behaviour* (= ungewolltes Verhalten), *State-Driven* (= zustandsgesteuert) und *Optional Feature* (= optionale Eigenschaft). Die Schablonen können nicht nur einzeln verwendet, sondern auch kombiniert werden, wodurch sie eine komplexere Form annehmen können. In [MW10] wird beschrieben, dass acht Probleme, die durch die natürliche Sprache entstehen können, durch die EARS-Templates reduziert bzw. komplett eliminiert werden können. Folgende sprachliche Probleme werden hierbei aufgelistet: Mehrdeutigkeit, Vagheit, Komplexität, Auslassung, Redundanz, Wortreichtum, Untestbarkeit und Aussagen darüber, wie etwas realisiert werden soll anstatt welche Funktion realisiert werden soll. Weiterhin ist auch bei diesem schablonenbasierten Ansatz zu erwähnen, dass das Agens des Satzes immer explizit genannt wird. Allerdings wird in [MW10] auch erwähnt, dass der EARS-Ansatz nur für *high-level* Anforderungen gilt, also Anforderungen, die keine detaillierten Funktionen einer Software beschreiben.

Für die in dieser Arbeit untersuchten Spezifikationen ist der Ansatz also wegen seiner Allgemeinheit nicht geeignet.

Weitere Beispiele für die Anwendung des Schablonenansatzes sind in [DBRT99] und [Küh10] beschrieben.

Schablonen können dabei helfen strukturierte, einheitlich aufgebaute und sprachlich bessere Anforderungen zu schreiben. Der Autor wird beispielsweise gezwungen im Aktiv zu schreiben und somit das Agens immer explizit zu machen. Weiterhin wird der Grad der Verbindlichkeit durch bestimmte Schlüsselwörter festgelegt. Das Problem der Sprachkomplexität kann reduziert, aber nicht vollständig eliminiert werden. Dadurch, dass Bedingungen sehr frei formuliert werden können, kann eine Anforderung trotz Schablone eine komplexe Form annehmen. Es gibt aber auch potentiell im Kontext von Anforderungen problematische sprachliche Phänomene, die trotz Anwendung von Schablonen weiterhin auftreten können, wie z.B. Weak-Words. Diese können bei schablonenbasierten Ansätzen genauso genutzt werden wie bei der uneingeschränkten Formulierung von Anforderungen. Schablonen müssen außerdem erlernt und vom Autor akzeptiert werden. Diese Akzeptanz zu erlangen ist nicht einfach, da die Schablonen den Autor sehr stark einschränken und er einen Sachverhalt eventuell nicht so darstellen kann, wie er es gerne möchte, sodass dadurch inhaltliche Fehler entstehen. Schablonen sind also im Prinzip eine gute Unterstützung bei der Formulierung von Anforderungen, sprachliche Defekte werden durch die Verwendung von Schablonen allerdings nicht komplett umgangen.

5.3 Kontrollierte Sprachen

Eine weitere Methode, um gute Anforderungen zu formulieren, ist die Anwendung einer kontrollierten natürlichen Sprache. „Kontrollierte natürliche Sprache ist eine Art der menschlichen Sprache, welche eingeschränkt ist für bestimmte praktische Anwendungen [...]“⁶ [ISO15]. Eine kontrollierte Sprache ist ein restringiertes Fragment einer natürlichen Sprache, d.h. sie verwendet ein beschränktes Vokabular und ist ebenfalls auf bestimmte Satzbaumuster beschränkt. Dabei handelt es sich um ein für spezielle Anwendungen definiertes Fragment, welches eine Untermenge der natürlichen Sprache ist. Ziel bei der Verwendung einer kontrollierten Sprache ist es, „lexikalische und strukturelle Ambiguitäten aufzulösen“ [ISO15] bzw. sie gar nicht erst entstehen zu lassen. Ein weiteres Ziel ist, die Lesbarkeit von Texten zu erhöhen und somit ein schnelleres Verständnis beim Lesen zu fördern. Kontrollierte Sprache hilft außerdem dabei, den Übersetzungsaufwand zu reduzieren [Bal14], da immer dieselben Satzbaumuster verwendet werden.

⁶Original: “Controlled natural language is a type of human language that is restricted for certain practical purposes.”

Kontrollierte Sprache grenzt sich von leichter Sprache insofern ab, als dass kontrollierte Sprache speziell dafür ausgelegt ist Ambiguitäten weitestgehend zu reduzieren. Leichte Sprache hingegen hat das Hauptziel, Menschen mit eingeschränkten kognitiven oder sensorischen Fähigkeiten das Lesen zu erleichtern [Maa15]. Leichte Sprache ist insofern eine (weitergehend eingeschränkte) Untermenge der kontrollierten Sprache.

Beispiele für kontrollierte Sprachen sind *Caterpillar Fundamental English* [Gla72], *AEC-MA Simplified English* [AEC95], *Simplified Technical English* [ASD07] und *Attempto Controlled English (ACE)* [FKK08]. Als Beispiel für eine kontrollierte Sprache wird ACE im Folgenden detaillierter beschrieben.

„Attempto Controlled English (ACE) ist eine kontrollierte natürliche Sprache, d.h. eine genau definierte Untermenge des Englischen, die automatisch und eindeutig in Prädikatenlogik 1. Stufe übersetzt werden kann“⁷ [FHK⁺05]. ACE wurde ursprünglich für die Definition von Software- und Hardwareanforderungen in der Luft- und Raumfahrtindustrie entworfen. ACE basiert auf der Diskursrepräsentationstheorie (DRT) [KR93] und beinhaltet ein vordefiniertes beschränktes Vokabular, welches jederzeit erweitert werden kann. ACE kann folglich auch auf weitere Textsorten angewandt werden. Außerdem enthält ACE sogenannte *Construction Rules*, nach welchen Sätze nach bestimmten Mustern gebildet werden können. Man kann nicht nur einfache Sätze, die für Ereignisse und Zustände stehen können, konstruieren, sondern man kann auch komplexere Sätze bilden, welche Koordinationen, Subordinationen, Quantifizierungen und Negationen einschließen. Die Sätze können mit der *Attempto Parsing Engine (APE)* (vgl. [FKK08]) in Diskursrepräsentationsstrukturen (DRSen) umgewandelt werden, wobei ca. 20 Interpretationsregeln verwendet werden. Im Folgenden werden zwei Interpretationsregeln, welche in ACE enthalten sind, kurz beschrieben. Enthält ein Satz beispielsweise die Konjunktionen *und* und *oder*, so gilt die Regel, dass *und* stärker bindet als *oder*.

Beispiel 61 *Das System muss einen Wert zurückgeben oder eine Warnung ausgeben und eine Anzeige im Display erscheinen lassen.*

Für den Satz in Beispiel 61 würde dies bedeuten, dass die folgenden zwei Fälle auftreten können: Im ersten Fall wird ein Wert zurückgegeben. Im zweiten Fall erscheint eine Anzeige im Display und zusätzlich zu der Anzeige wird eine Warnung ausgegeben.

Eine weitere Interpretationsregel verwendet ACE zur Auflösung von anaphorischen Referenzen. Tritt beispielsweise ein Personalpronomen auf und im vorausgehenden Satzkontext

⁷Original: “Attempto Controlled English (ACE) is a controlled natural language, i.e. a precisely defined subset of full English that can automatically and unambiguously be translated into full first-order logic.”

gibt es mehrere mögliche Antezedenten, d.h. Nomina oder Nominalphrasen, welche in Genus und Numerus mit dem Pronomen übereinstimmen, so wird als Antezedens immer das Nomen oder die Nominalphrase gewählt, welches dem Pronomen unmittelbar vorausgeht (d.h. im kürzest möglichen Abstand zu ihm steht).

Beispiel 62 *Der Blink-Master ist im Außenlicht-Master integriert. Er empfängt alle Signale.*

In Beispiel 62 wäre dies der *Außenlicht-Master*. Als Kontext eines Satzes, auf welchem Interpretationen getroffen werden können, können nur vorausgehende Sätze verwendet werden.

Laut [FKK08] können durch das eingeschränkte Vokabular die meisten mehrdeutigen Konstruktionen vermieden werden, da jedes Lexem lediglich eine mögliche Lesart hat. Die mehrdeutigen Konstruktionen, die nicht durch das beschränkte Vokabular vermieden werden können, werden durch die Interpretationsregeln aufgelöst. Der Benutzer muss die von ACE definierten Interpretation akzeptieren, oder seinen Satz umformulieren.

Ambiguitäten können durch die Verwendung von ACE größtenteils vermieden werden, da dem Benutzer nach Eingabe einer Anforderung in der *Attempto Parsing Engine* eine paraphrasierte Version seiner Anforderung aufgezeigt wird. Der Benutzer kann sich dann entscheiden, ob diese Interpretation seiner eigenen Intention entspricht. Falls dies nicht der Fall ist, kann der Benutzer die Anforderung nochmals überarbeiten und sich die neue Interpretation ansehen. Das Problem der fehlenden Information wird allerdings nicht komplett gelöst. Weak-Words wie *lang* und *kurz* können ohne quantifizierende Angabe verwendet werden. Es gibt keine Möglichkeit, eine Regel anzulegen, die beispielsweise kritische vs. unkritische Instanzen von Weak-Words überprüft. Die Verwendung von Passiv ist in ACE nicht möglich. Der Autor wird gezwungen im Aktiv zu schreiben, womit gewährleistet wird, dass das Agens des Verbs immer angegeben wird und es somit zu keinen Fehlinterpretationen kommen kann.

Kontrollierte Sprachen können dem Autor dabei helfen Texte zu verfassen. Durch das beschränkte Vokabular und die Satzbildungsregeln, können Mehrdeutigkeiten größtenteils vermieden werden. Hierbei fungieren die Satzbildungsregeln ähnlich wie die Satzschablonen (vgl. Abschnitt 5.2). Der Unterschied liegt im beschränkten Vokabular. Aufgrund der syntaktischen Vorgaben handelt es sich bei den Satzschablonen auch um eine Art der kontrollierten Sprache.

Die kontrollierte Sprache bringt, wie auch schon die Satzschablonen, Nachteile mit sich. Die Sprache muss zunächst erlernt werden [Bal14] und auf Akzeptanz beim Anwender stoßen. Der Autor muss sich an die Satzbildungsregeln halten und wird somit in seinem

Schreiben eingeschränkt. Die Pflege eines Lexikons ist ein weiteres wichtiges Kriterium. Das Lexikon muss immer auf dem neuesten Stand sein. Im Bereich der PKW-Entwicklung ist die Erstellung eines solchen Lexikons komplex und nimmt eine große Menge Zeit in Anspruch. Die Komponenten und Systeme, deren Funktionen in Lastenheften beschrieben werden, sind sehr vielfältig. Es müsste für jedes System im Fahrzeug (Schließung, Innenlicht, Außenlicht, Klimatisierung, usw.) ein eigenes Lexikon aufgebaut werden. Hinzu kommen immer wieder neu zu entwickelnde Systeme, Komponenten und Innovationen, für welche neue Lexika aufgebaut werden müssten. Fehlende Information bleibt auch bei Verwendung einer kontrollierten Sprache weiterhin ein Problem.

Der Einsatz einer kontrollierten Sprache im Bereich des Requirements Engineering ist im Prinzip sinnvoll; jedoch ist der Zeitaufwand für die Bereitstellung und Pflege der Daten zu hoch. Ein Einsatz in der Praxis ist somit kaum realisierbar.

5.4 Manuelle Review-Techniken

Die Durchführung manueller Reviews ist eine Technik, um bereits geschriebene Anforderungen auf ihre Qualität hin zu überprüfen. In [PR15] werden unter dem Oberbegriff des Reviews drei Ausprägungsformen unterschieden: Stellungnahme, Walkthrough und Inspektion. Auf diese drei Ausprägungen soll im Folgenden genauer eingegangen werden.

Bei der Stellungnahme übergibt der Autor die von ihm geschriebenen Anforderungen an eine zweite Person; typischerweise an einen Kollegen. „Diese Person überprüft die Anforderungen dann mit dem Ziel, Qualitätsmängel in den Anforderungen (z.B. Mehrdeutigkeiten oder Fehler) hinsichtlich vorher festgelegter Qualitätskriterien zu identifizieren“ [PR15]. Der Prüfer markiert im Dokument die Anforderungen, welche der Autor seines Erachtens nochmals überarbeiten sollte, und versieht diese mit kurz gehaltenen Kommentaren. Der Autor kann dann die daraus resultierenden Änderungen in die Anforderungen einarbeiten [Rup14]. Die Vorteile einer Stellungnahme sind der geringe Zeitaufwand bei dessen Durchführung und die Tatsache, dass Lücken bei der Anforderungsformulierung einfach aufgedeckt werden können. Die Nachteile der Stellungnahme sind, dass die Qualität des Reviews stark vom Prüfer abhängt. Dabei können die Faktoren verfügbare Zeit, Kompetenz und Motivation eine Rolle spielen. Ein weiterer Nachteil ist, dass Verständnisprobleme während der Prüfung nur sehr schlecht aufgelöst werden können [Rup14].

„Bei einem Walkthrough präsentiert der Autor die von ihm erstellten Artefakte einer Gruppe von Teilnehmern. Der Autor erläutert die Artefakte und weist auf kritische Stellen hin, die die Teilnehmer kommentieren sollen“ [Poh07]. Die Prüfer können Fragen stellen, Verbesserungsvorschläge einbringen und über die präsentierten Anforderungen diskutieren.

Das Ziel dieser Review-Technik ist vor allem ein einheitliche Verständnis der Anforderungen zwischen allen Beteiligten herzustellen. Verständnisprobleme können bei der Sitzung sofort diskutiert und die aus den Diskussionen resultierenden Ergebnisse in die Anforderungen eingearbeitet werden. Der große Vorteil dieser Technik ist, dass sich das Team im persönlichen Gespräch austauschen kann [Rup14]. Ein entscheidender Nachteil dieser Technik besteht jedoch darin, dass in den meisten Fällen der Autor selbst die Sitzung leitet und somit aktiv von den Schwachstellen in der Spezifikation ablenken kann. Sind dem Autor bestimmte Probleme gar nicht bewusst, werden diese in der Sitzung nicht diskutiert und können somit auch nicht beseitigt werden. Für die Durchsprache steht außerdem nur ein begrenzter Zeitraum zur Verfügung, wodurch ggf. nicht alle Anforderungen detailliert diskutiert werden können. Auch bei dieser Technik hängt die Qualität des Reviews stark von den beteiligten Personen ab.

Die Inspektion stellt die dritte Ausprägungsform des Reviews dar. Wie bei der Durchführung eines Walkthroughs nehmen auch bei der Inspektionen mehrere Personen die Rolle des Prüfers ein. Die typischen Phasen für die Validierung von Anforderungen sind hierbei die Planung, die Übersicht, die Fehlersuche und Fehlersammlung [PR15]. In der Planungsphase werden die Inspektionsteams eingewiesen. Hierbei wird festgelegt, welche Anforderungen einem Review unterzogen werden sollen, und die Teilnehmer der Inspektion werden festgelegt. „In der Übersichtsphase erläutert der Autor den anderen Mitgliedern des Inspektionsteams die zu überprüfenden Anforderungen, damit innerhalb des Inspektionsteams ein gemeinsames Verständnis über die Anforderungen entsteht“ [PR15]. In der folgenden Phase führen die Inspektoren die Fehlersuche durch. Diese kann im Team oder von Einzelpersonen durchgeführt werden. Es kann hierbei sinnvoll sein, jedem Inspektor eine Perspektive zuzuweisen, unter welcher er die Anforderungen betrachten soll. Dies kann der Blickwinkel der Testbarkeit sein, die Perspektive des Kunden, usw. In der Phase der Fehlersammlung werden dann die identifizierten Fehler zusammengetragen, konsolidiert und dokumentiert [PR15], [Poh07]. Der entscheidende Vorteil dieser Technik ist, dass eine große Menge an Auffälligkeiten gefunden werden kann. Ein erheblicher Nachteil ist allerdings, dass der Zeitaufwand bei der Durchführung einer Inspektion sehr hoch ist. Auch bei dieser Review-Technik hängt die Qualität der Fehlersuche stark von den Inspektoren ab.

Manuelle Review-Techniken können dabei helfen, Anforderungen früh im Entwicklungsprozess auf ihre Qualität hin zu überprüfen. Hierbei werden drei Ausprägungsformen unterschieden: Stellungnahme, Walkthrough und Inspektion. Jede dieser Techniken bringt Vor- und Nachteile mit sich. Allen Techniken gemeinsam ist, dass die Qualität der Ergebnisse stark von der Verfügbarkeit, Kompetenz und Motivation der Reviewer abhängt.

Manuelle Reviews bringen außerdem einen hohen Zeitaufwand mit sich. Die Menge an zu prüfenden Anforderungen in der Systementwicklung (ca. 1300 Anforderungen pro Systemlastenheft), ist jedoch zu hoch, um alle Anforderungen manuell prüfen zu können. Es ist daher sinnvoll, die manuellen Reviews in Kombination mit einer automatischen Überprüfung der Lastenhefte mittels eines Kontrollwerkzeugs durchzuführen.

5.5 Werkzeuge für die Überprüfung natürlichsprachlicher Texte

Ein weiteres Hilfsmittel, um qualitativ gute Anforderungen zu schreiben, sind Textprüfwerkzeuge. Diese Werkzeuge sollen dem Autor dabei helfen bereits geschriebene Texte oder schreibbegleitend Texte danach zu überprüfen, ob sie mehrdeutige Konstruktionen enthalten und ob für das Verständnis wichtige Information fehlt.

Es existieren frei verfügbare und kostenpflichtige Werkzeuge auf dem Markt, die auf unterschiedliche Textsorten spezialisiert sind. Die in dieser Arbeit betrachteten Werkzeuge sind aus dem Bereich der Sprachtechnologie (bzw. aus dem Bereich der technischen Dokumentation) und aus dem Bereich des Requirements Engineering.

Um den Bedarf eines neuen Werkzeugs für die Prüfung der in dieser Arbeit betrachteten Lastenhefte zu unterstreichen, werden im Folgenden die Funktionen der bestehenden Werkzeuge, welche Sprachprüfungen für deutsche Texte bereitstellen, betrachtet und es wird überprüft, welche Analyseerfolge die Prüfungen haben. Hierfür werden vier Werkzeuge und deren Funktionen aus dem Bereich der Sprachtechnologie und vier Werkzeuge aus dem Bereich des Requirements Engineering vorgestellt. Hierbei soll insbesondere auf die sprachlichen Phänomene eingegangen werden, welche in Abschnitt 4.8 als relevant identifiziert wurden.

Nach einer kurzen Einführung der Werkzeuge (vgl. Abschnitt 5.5.1 und 5.5.2) wird in Abschnitt 5.5.3 eine Testsatzsammlung für die Evaluierung der Funktionen der Werkzeuge vorgestellt. Diese Testsatzsammlung umfasst 140 Anforderungen, in welchen alle Phänomene enthalten sind, die im Qualitätsmodell aufgeführt werden. Es werden, sofern möglich, kritische und unkritische Fälle jedes Phänomens abgebildet. Mithilfe der Testsatzsammlung soll evaluiert werden, welches Textprüfwerkzeug welche Phänomene aus dem Qualitätsmodell analysiert, und es soll insbesondere gezeigt werden welche Qualität die Analyseergebnisse haben.

5.5.1 Werkzeuge aus dem Bereich des Requirements Engineering

Die vier Werkzeuge aus dem Bereich des Requirements Engineering, die in diesem Abschnitt vorgestellt werden, sind *DESIRE*[®], *ReqCheck*, *Requirements Quality Suite* und *Requirements Scout*.

5.5.1.1 DESIRe[®]

DESIRe[®] (= *Dynamic Expert System for Improving Requirements*)⁸ ist ein frei verfügbares Werkzeug aus dem Bereich des Requirements Engineering für die Überprüfung von natürlichsprachlichen Anforderungen. DESIRe[®] ist verfügbar für *MS WORD* und für *DOORS*. „Die Idee von DESIRe[®] zeichnet sich dadurch aus, dass aufgrund von zuvor bestimmten Wörtern, beispielsweise Verben, vordefinierte Fragen und/oder Hinweise an den Autor kommuniziert werden. Dazu bedarf es einer Liste relevanter Wörter inklusive aller möglichen Wortformen und der dazugehörigen Fragen und Hinweise.“ [SSUE09]. Die Herangehensweise von DESIRe[®] beruht auf Wortlistenabgleichen und angepassten Fragen. Für das potentielle Weak-Word *schnell* könnte dies wie in Tabelle 8 dargestellt aussehen.

Wortausprägungen	Fragen	Hinweise
schnell schneller schnellsten schnellere	Wie schnell?	Das Wort schnell ist zu quantifizieren.

Tabelle 8: Beispiel für einen Eintrag in einer Wortliste in DESIRe[®] [SSUE09]

In dieser Weise sollen vom Benutzer jegliche Wörter abgelegt werden, die bei einer Analyse geprüft werden sollen. Sobald ein Wort, welches in der Liste der zu prüfenden Wörter abgelegt ist, in einem Text auftritt, werden dem Autor alle zur Verfügung stehenden Fragen gestellt. Bei der Analyse wird jedoch nicht geprüft, ob Fragen überflüssig sind, weil sie im analysierten Satz bereits beantwortet werden. Zur Vorverarbeitung der Texte wird bei DESIRe[®] lediglich eine Tokenisierung des Textes durchgeführt. Eine computerlinguistische Aufbereitung der Texte durch Lemmatisierung, Tagging oder Parsing ist nicht vorhanden.

Weiterhin wird für die Verwendung von DESIRe[®] empfohlen mit einem Glossar zu arbeiten. In diesem Glossar soll der Benutzer Wörter und ihre Definitionen ablegen. „Dadurch wird das Verständnis eines Begriffes innerhalb des Leserkreises eindeutig kommuniziert [...]“ [SSUE09].

DESIRe[®] setzt die Pflege einer Wortliste und eines Glossars voraus. Da bei der Analyse mit DESIRe[®] lediglich diese Wortlisten abgeprüft werden, entstehen bei der Verwendung des Werkzeugs eine große Menge an Fehlwarnungen. Dies ist beispielsweise der Fall, wenn Weak-Words wie *lang* oder *schnell* geprüft werden sollen, da diese Wörter stets im Kontext betrachtet und bewertet werden müssen (vgl. Abschnitt 4.2.1.2 und Abschnitt 8.1).

⁸<https://www.hood-group.com>

5.5.1.2 ReqCheck

*ReqCheck*⁹ ist ein am *Fraunhofer-Institut für Experimentelles Software Engineering IE-SE* entwickeltes Werkzeug für die Überprüfung von natürlichsprachlichen Anforderungen. Grundlage für die Analysen ist das aus der Sprachtechnologie stammende Werkzeug *CLAT* (= *Controlled Language Authoring Tool*, vgl. Abschnitt 5.5.2.3). *ReqCheck* enthält eine Teilmenge der Regeln, welche in *CLAT* zur Verfügung gestellt werden. Diese Regeln zielen insbesondere auf die Kriterien der Eindeutigkeit und Verständlichkeit ab [KKRH15]. In *ReqCheck* wird eine umfangreiche computerlinguistische Vorverarbeitung und Annotation der Texte durchgeführt: Die Texte werden tokenisiert, lemmatisiert, es wird Part-of-Speech-Tagging und eine morphologische Analyse der Wörter durchgeführt und die zu analysierenden Texte werden geparst.

ReqCheck kann Texte verarbeiten, welche im *MS WORD*-Format vorliegen. Das *WORD*-Dokument wird bei der Analyse zunächst in ein *HTML*-Dokument konvertiert. „Dieses wird im Anschluss vom System analysiert und um Tags angereichert, die die Befunde markieren. Diese Tags enthalten unter anderem auch die Information, welche Regel an dieser Stelle angeschlagen hat und erlaubt es somit, genaue Erläuterungen zum mutmaßlichen Fehler sowie Hinweise, wie dieser behoben werden kann, einzufügen. Diese Informationen werden als *Popup* im *HTML*-Dokument angezeigt, sobald sich die Maus über eine entsprechend markierte Befundstelle bewegt“ [KKRH15]. Alternativ zur Bereitstellung der *HTML*-Version, kann ein *PDF* erzeugt werden, in welchem die Befunde markiert und durch eindeutige Bezeichner am Ende des Dokuments die Fehlermeldungen referenziert werden können.

Auch bei *ReqCheck* wird der Großteil der Analysen mittels Wortlistenabgleichen durchgeführt. Jedoch ist die Grundlage für tiefer gehende Analysen durch die computerlinguistische Vorverarbeitung der Texte gegeben.

5.5.1.3 Requirements Quality Suite

Die *Requirements Quality Suite* (= *RQS*)¹⁰ ist ein Werkzeug, entwickelt von der *The REUSE Company*, welches ebenfalls darauf spezialisiert ist natürlichsprachliche Anforderungen zu überprüfen. *RQS* kann für Anforderungen in deutscher und in englischer Sprache verwendet werden und ist u.a. in *DOORS* und *MS Excel* anwendbar. Im Vordergrund steht bei *RQS* der Gedanke, dass Anforderungen idealerweise mit Satzschablonen (vgl. Abschnitt 5.2) geschrieben werden.

⁹<http://reqcheck.iese.de/>

¹⁰<https://www.reusecompany.com/requirements-quality-suite>

Der Benutzer kann in RQS eigene Satzschablonen anlegen, bestehende bearbeiten und diese gewichten. Die Schablonen werden beim Schreiben der Anforderungen abgeprüft. Ein Ziel hierbei ist, eine einheitliche syntaktische Struktur der Anforderungen zu erreichen. Einige sprachliche Phänomene können durch die Verwendung von Schablonen ausgeschlossen werden, wie beispielsweise die Verwendung des Passivs. Andere sprachliche Phänomene können nicht grundsätzlich ausgeschlossen werden (vgl. hierfür Abschnitt 5.2). Für die Verwendung der Schablonen setzt RQS eine vollständig spezifizierte Terminologie voraus. Ist ein Wort nicht in der Terminologiedatenbank vorhanden, wird das Wort als *Unclassified Noun* kategorisiert. Tritt dieser Fall in einer Anforderung auf, können die definierten Satzmuster nicht greifen.

Der Benutzer hat die Möglichkeit im sogenannten *Knowledge Manager* eine Terminologiedatenbank aufzubauen oder eine bereits bestehende Datenbank zu importieren. Hierfür müssen mehrere Dateien, welche Wörter, die der gleichen Wortart angehören, vom Benutzer bereitgestellt werden. Standardmäßig ist keine umfangreiche Terminologie enthalten.

Regeln werden durch RQS standardmäßig mitgeliefert und der Benutzer kann eigene Regeln hinzufügen. Hierfür können alle Merkmale der Wörter, welche in der Terminologiedatenbank bereitgestellt werden, verwendet werden. Ein integrierter Part-of-Speech-Tagger, ein Morphologie-Tagger und ein Parser sind in RQS nicht vorhanden. Die bereitgestellten Regeln basieren wie bei *DESIRE*[®] und *ReqCheck* größtenteils auf Wortlistenabgleichen.

Für die Verwendung von RQS ist ein sehr hoher Zeitaufwand notwendig, um die Terminologiedatenbank aufzubauen, Regeln zu entwerfen und die Gewichte der Regeln passend zu verteilen. Die Gewichtung der Regeln muss mit großer Sorgfalt durchgeführt werden, damit bei Überschneidung der Regeln die gewünschte Regel greifen kann.

5.5.1.4 Requirements Scout

Requirements Scout [FFWE16] ist ein weiteres Werkzeug, welches darauf spezialisiert ist natürlichsprachliche Anforderungen (in Deutsch und in Englisch) zu überprüfen. Das Werkzeug wurde von dem Unternehmen *Qualicen*¹¹ entwickelt.

Das Werkzeug *Requirements Scout* kann Texte analysieren, welche im Format *MS WORD*, *MS Excel*, *PDF*, *CSV* oder als einfacher Text vorliegen. Auch Lastenhefte, die in der Datenbank *DOORS* abgelegt sind, können analysiert werden. Die Dokumente werden in *Requirements Scout* eingelesen und computerlinguistisch vorverarbeitet und annotiert. Die Texte werden tokenisiert, lemmatisiert, getaggt und geparst. Neben Part-of-Speech-Tagging wird auch eine oberflächliche morphologische Analyse durchgeführt.

¹¹<https://www.qualicen.de/en/>

Die bereitgestellten Sprachprüfungen bedienen sich unterschiedlicher Analysetechniken [FFWE16]. Ein Teil der Prüfungen wird durch Wortlistenabgleiche durchgeführt (z.B. bei der Prüfung auf mehrdeutige Adjektive und Adverbien), bei anderen Prüfungen hingegen wird auch die morphologische Analyse und/oder das Part-of-Speech-Tagging hinzugezogen. Die Erkennung von Komparativ- und Superlativformen wird beispielsweise durch die Kombination von Part-of-Speech-Tagging und morphologischer Analyse durchgeführt. Pronomina werden lediglich durch Part-of-Speech-Tags geprüft.

Das Werkzeug *Requirements Scout* hebt sich von den anderen präsentierten Werkzeugen aus dem Requirements Engineering insofern ab, als dass es sich von reinen Wortlistenabgleichen, so weit dies sinnvoll ist, distanziert. Computerlinguistische Annotationen verbessern die Analysen des Werkzeugs erheblich. Jedoch wird keine tief gehende linguistische Analyse durchgeführt, die das Parsing einbezieht oder eine umfangreiche Kontextanalyse durchführt. Die technischen Voraussetzungen dafür sind jedoch vorhanden.

5.5.2 Werkzeuge aus dem Bereich der Sprachtechnologie

Die vier Werkzeuge aus dem Bereich der Sprachtechnologie, die in diesem Abschnitt vorgestellt werden, sind *LanguageTool*, *TextLab*, *CLAT* und *Acrolinx*.

5.5.2.1 LanguageTool

Das Werkzeug *LanguageTool*¹² ist eine freie Stil- und Grammatikprüfung aus dem Bereich der Sprachtechnologie. „Die mehr als 1400 Regeln für die deutsche Sprache, mit deren Hilfe *LanguageTool* Fehler erkennt, dienen aber nicht nur zum Auffinden von Fehlern in den Kategorien Grammatik, Groß- und Kleinschreibung, Zusammen- und Getrenntschreibung und Zeichensetzung [...]“ [Bre12]. Es werden auch weitere Sprachprüfungen zur Verfügung gestellt (wie beispielsweise eine Prüfung auf die Verwendung von Passiv). Zusätzlich stellt *LanguageTool* eine Prüfung auf *Leichte Sprache* bereit. *Leichte Sprache* hat das Hauptziel, Menschen mit eingeschränkten kognitiven oder sensorischen Fähigkeiten das Lesen zu erleichtern [Maa15]. Das Regelset für die Prüfung auf *Leichte Sprache* enthält beispielsweise Regeln von der Art, dass keine Wortformen im Genitiv benutzt werden und auf Abkürzungen komplett verzichtet werden soll.

„*LanguageTool* kann als eigenständige Anwendung auf jedem Computer, auf dem Java ausgeführt werden kann, benutzt werden“ [Bre12]. Zusätzlich kann es als Erweiterung in *LibreOffice* oder *Apache OpenOffice* angebunden werden.

¹²<https://languagetool.org/de/>

Bevor der Text analysiert werden kann und die Regeln zur Anwendung kommen, wird der Text zunächst in Sätze unterteilt und anschließend tokenisiert. Darauffolgend wird Part-of-Speech-Tagging und morphologisches Tagging durchgeführt. Den Wörtern werden immer alle möglichen verfügbaren Tags, welche in einem Wörterbuch abgelegt sind, zugewiesen; d.h. es wird nicht basierend auf dem Kontext der Wörter disambiguiert. Regeln zur Disambiguierung können durch den Benutzer hinzugefügt werden.

Die Analysen von LanguageTool beruhen nicht nur auf Wortlistenabgleichen. Für die Analysen werden die Part-of-Speech-Tags und die morphologische Analyse miteinbezogen. Der Benutzer kann selbständig Regeln anlegen und hierfür alle zur Verfügung stehenden linguistischen Merkmale in die Regeln miteinbeziehen. Zu jedem Fehler gibt es eine ausführliche Fehlermeldung. „Manche Regeln sind standardmäßig nicht aktiviert, da sie entweder nur für bestimmte Textarten (z.B. Briefe und Emails) geeignet sind, zu viele Fehlalarme erzeugen oder einfach nur ein Scherz sind“ [Bre12].

Der Fokus von LanguageTool liegt auf der Stil- und Grammatikprüfung und auf der Prüfung auf *Leichte Sprache*. Regeln, die auf Ambiguitäten und fehlende Information abzielen, sind kaum zu finden.

5.5.2.2 TextLab

Ein weiteres Werkzeug aus dem Bereich der Sprachtechnologie, welches natürlichsprachliche Texte überprüft, ist *TextLab*¹³. TextLab wurde von der *H&H Communication Lab GmbH* in Zusammenarbeit mit der Universität Hohenheim entwickelt. „TextLab wird [...] von namhaften Versicherungsunternehmen, Banken, Energieversorgern, Technik-Konzernen, Telekommunikationsunternehmen sowie von wissenschaftlichen Einrichtungen und staatlichen Behörden eingesetzt. Also überall dort, wo es darauf ankommt, den Kunden schwierige Sachverhalte einfach und verständlich mitzuteilen“ [Tex].

In TextLab kann die Verständlichkeit eines Textes durch verschiedene Verständlichkeitsindices errechnet werden (vgl. Abschnitt 3.1.1). Diese Verständlichkeitsmessungen kann sich der Benutzer anzeigen lassen, um einen Überblick darüber zu bekommen wie verständlich seine erstellten Texte sind. Zusätzlich kann in TextLab auch Terminologie und Corporate Language überprüft werden. Die in TextLab enthaltenen Sprachprüfungen zielen größtenteils auf die Komplexität einzelner Wörter und Sätze ab (zu lange Wörter, Schachtelsätze, usw.). Bei der Prüfung eines Textes kann zusätzlich angegeben werden, um welche Textsorte es sich handelt (z.B. E-Mail, Broschüre oder Web-Text), damit die

¹³<https://www.text-lab.de/>

Schwellenwerte der Prüfungen entsprechend angepasst und nur die Regeln, die für die jeweilige Textsorte entwickelt wurden, angewandt werden.

In TextLab kommen ein Tokenizer, ein Lemmatisierer und ein Part-of-Speech-Tagger zum Einsatz. Es wird kein Parsing und keine morphologische Analyse durchgeführt. TextLab ist als Online-Version, als Plug-in für *MS WORD*, als Integration in Intranetseiten von Firmen verfügbar und kann als API-Schnittstelle in ein eigenes Schreibprogramm integriert werden.

Die Analysen von TextLab konzentrieren sich hauptsächlich auf die Komplexität von Sätzen und einzelnen Wörtern. Die Identifikation mehrdeutiger Konstruktionen und das Aufdecken fehlender Information stehen nicht im Fokus der Analysen.

5.5.2.3 CLAT

CLAT (= *Controlled Language Authoring Tool*)¹⁴ ist ein am *Institut für Angewandte Informationsforschung* in Saarbrücken entwickeltes Werkzeug, das Texte auf eine einheitliche Terminologie prüft und zusätzlich Sprachprüfungen bereitstellt. Das Werkzeug CLAT ist komplett in den *Congree Authoring Server*¹⁵ von der Firma *Across* integriert, weshalb *Congree Authoring Server* in dieser Arbeit nicht separat vorgestellt wird.

In CLAT ist der Duden-Korrektor eingebunden, welcher eine Rechtschreib- und Grammatikprüfung durchführt. Ein Fokus von CLAT liegt außerdem auf der Sicherstellung einer einheitlichen Terminologie. Hat CLAT Zugriff auf eine Terminologiedatenbank, kann es auf erlaubte und unerlaubte Terme verweisen und Terme aufzeigen, bei welchen potentiell Verwechslungsgefahr mit anderen Termen besteht.

Die zu verarbeitenden Texte werden in CLAT zunächst tokenisiert. Anschließend werden die Tokens nach Wortarten und nach morphologischer Information analysiert und entsprechend getaggt. Die Texte werden außerdem geparst. Laut [Ras06] werden durch CLAT unter anderem lexikalische Probleme, Mehrdeutigkeiten, Ellipsen, und das Problem der Komplexität behandelt. Weiterhin wird eine einheitliche Anordnung der Konstituenten im Satz geprüft (z.B. Prüfung, ob Bedingungssatz am Anfang des Satzes steht).

CLAT stellt für jedes sprachliche Phänomen, das durch eine Prüfung identifiziert werden kann, eine Fehlermeldung zur Verfügung, zusammen mit einem konstruierten Beispiel. Es wird jedoch kein konkreter Vorschlag zu dem vorliegenden Satz gemacht, sondern lediglich die festgelegte Fehlermeldung ausgegeben.

¹⁴<http://www.iai-sb.com/>

¹⁵<http://www.congree.com/>

CLAT stellt eine große Menge an Prüfungen bereit, die sich größtenteils auf die in der tekcom-Leitlinie [tek11] aufgelisteten Phänomene beziehen. Eine Prüfung auf Terminologie ist ebenfalls ein wesentlicher Bestandteil der Prüfungen, welche von CLAT bereitgestellt werden.

5.5.2.4 Acrolinx

*Acrolinx*¹⁶ ist eine Software der *Acrolinx GmbH*, mit welcher natürlichsprachliche Texte überprüft werden können. Acrolinx stellt eine Reihe von Prüfungen und Funktionen bereit. Mit Acrolinx kann der Benutzer beispielsweise Terminologie verwalten und eigene Texte auf eine einheitliche Terminologie überprüfen lassen. Es ist auch eine Prüfung auf Einhaltung des *ASD-STE100*-Standards¹⁷ (*Simplified Technical English*) möglich, und es können Lesbarkeitsindices (vgl. hierzu Abschnitt 3.1.1) angewandt werden, um einen Eindruck zu erhalten, wie verständlich die zu überprüfenden Texte geschrieben sind. Ein Regelset, welches auf der aktuellen tekcom-Leitlinie basiert, ist ebenfalls vorhanden sowie ein Regelset, das branchenübergreifend einsetzbar ist.

Acrolinx bietet neben einer Rechtschreib- und Grammatikprüfung auch weitere Sprachprüfungen an. Hierbei wird die Einhaltung von Sprachregeln, welche in der tekcom-Leitlinie enthalten sind, sowie die Einhaltung weiterer Sprachregeln überprüft. In Acrolinx ist außerdem ein Authoring-Memory integriert. Acrolinx kann dem Benutzer bereits verwendete Textbausteine vorschlagen, um einen einheitlichen Stil bei der Formulierung zu garantieren. Auch ein Translation-Memory ist enthalten, sodass ein Übersetzer beim Schreiben angezeigt bekommt, wie eine bestimmte Textpassage bei einer früheren Übersetzung übersetzt wurde. Der Benutzer kann dann entscheiden, ob er die früher verwendete Übersetzung verwenden möchte oder doch eine eigene Übersetzung wählt.

Acrolinx kann auf unterschiedlichen Datenformaten arbeiten wie *MS WORD*, *MS Excel*, *XML*, *PDF*, uvm. Das Werkzeug enthält außerdem einen Tokenisierer, einen Lemmatisierer, einen POS- und Morphologie-Tagger und einen Parser. Mit Acrolinx können folglich nicht nur reine Wortlistenabgleiche, sondern auch tiefer gehende linguistische Analysen durchgeführt werden.

5.5.3 Testsatzsammlung für die Evaluierung bestehender Textprüfwerkzeuge

Im Folgenden wird eine Testsatzsammlung für die Evaluierung der bereits bestehenden Textprüfwerkzeuge vorgestellt. Diese Testsatzsammlung enthält Anforderungen, in welchen die im Qualitätsmodell präsentierten Phänomene enthalten sind. Anhand dieser Test-

¹⁶www.acrolinx.de

¹⁷<http://www.asd-ste100.org>

satzsammlung sollen die in den Abschnitten 5.5.1 und 5.5.2 vorgestellten Werkzeuge evaluiert werden, d.h. es wird jeweils untersucht, welche Phänomene aus dem Qualitätsmodell durch welches Werkzeug analysiert werden und wie zuverlässig diese Analysen sind. In Abschnitt 5.5.3.1 wird zunächst der allgemeine Aufbau der Testsatzsammlung beschrieben und anschließend in Abschnitt 5.5.3.2 werden die Analyseergebnisse der Werkzeuge präsentiert und evaluiert.

5.5.3.1 Vorstellung und Aufbau der Testsatzsammlung

In den Abschnitten 5.5.1 und 5.5.2 wurden bereits Werkzeuge aus der Sprachtechnologie und aus dem Requirements Engineering vorgestellt, mit welchen natürlichsprachliche Texte auf ihre sprachliche Qualität hin überprüft werden können. Um die Qualität der Analysen der bereits existierenden Werkzeuge bewerten zu können, wurde eine Testsatzsammlung entwickelt. Diese Testsatzsammlung ist ein Dokument, welches Anforderungen enthält, die von den bestehenden Werkzeugen analysiert und bewertet werden sollen. Jedes Phänomen aus dem Qualitätsmodell wird in der Testsatzsammlung durch mindestens zwei Anforderungen repräsentiert. Die Testsatzsammlung enthält nicht nur problematische, sondern auch unproblematische Anforderungen. Die Güte der Analysen der Werkzeuge wird anhand der Testsatzsammlung evaluiert, d.h. es wird ein Überblick darüber gegeben, wie zuverlässig welches Werkzeug die kritischen Instanzen in der Testsatzsammlung identifiziert, und ob viele Fehlalarme durch die Werkzeuge produziert werden. Die Testsatzsammlung wurde als Modul in *DOORS* (vgl. Abschnitt 6.1) angelegt. Da nicht alle Werkzeuge auf Basis von *DOORS* arbeiten, wurden zusätzlich ein *MS WORD*- und ein *MS Excel*-Dokument erstellt.

In Abbildung 23 ist ein Ausschnitt aus der erstellten Testsatzsammlung zu sehen. Die Testsatzsammlung besteht aus den folgenden Spalten:

- *ID*: Jeder Anforderung wird eine eindeutige ID zugeordnet;
- *Anforderung*: Text der Anforderung;
- *Object Type*: Angabe, ob es sich um eine zu prüfende Anforderung handelt (alle Anforderungen nehmen in der Testsatzsammlung den Wert *requirement* an und müssen geprüft werden);
- *Phänomen*: Angabe darüber, welches Phänomen aus dem Qualitätsmodell in der Anforderung repräsentiert wird;
- *Problembeschreibung*: Eine kurze Erläuterung zu den vorhandenen Phänomenen (unabhängig von der Kritikalitätsstufe);

- **Kritikalität:** Angabe der Kritikalitätsstufe.

In der Testsatzsammlung gibt es folgende Kritikalitätsstufen für einzelne Anforderungen: *unkritisch*, *gering*, *gering/mittel*, *mittel/hoch* und *hoch*. Die Einteilung in Zwischenstufen kommt zustande, wenn beispielsweise mehrere Phänomene in einer Anforderung repräsentiert werden und diese unterschiedlichen Kritikalitätsstufen zugeordnet werden. Die Zwischenstufen kommen außerdem zustande, weil in manchen Fällen Kontextinformation dabei hilft, vermeintlich fehlende Information zu ermitteln. Die Testsatzsammlung enthält insgesamt 140 Anforderungen, von welchen 70 Anforderungen in die Kritikalitätsstufen *unkritisch*, *gering* oder *gering/mittel* fallen und 70, die in die Kritikalitätsstufen *mittel/hoch* oder *hoch* fallen.

ID	Anforderung	Object Type	Phänomen	Problembeschreibung x Kontext einer Vortrouerung.	Kritikalität
TSJK-D39	Die Länge der Nachlaufzeit beträgt 4 Sekunden. Die Länge der Nachlaufzeit beträgt 1,5 Sekunden.	requirement	(1) Lokaler Widerspruch	(1) Ein lokaler Widerspruch liegt vor, wenn eine Anforderung sich selbst widerspricht.	hoch
TSJK-D40	Das System leitet den Benutzer nach der Anmeldung zu seiner persönlichen Seite weiter.	requirement	(1) Nominalisierung	(1) Nominalisierungen können wichtige Informationen unterdrücken, da nicht alle Verbargumente des zugrundeliegenden Verbes genannt werden müssen.	gering
TSJK-D41	Das System zeigt die Meldung im Kombiinstrument an.	requirement	(1) Nominalisierung	(1) Nominalisierungen können wichtige Informationen unterdrücken, da nicht alle Verbargumente des zugrundeliegenden Verbes genannt werden müssen.	gering
TSJK-D130	Das System zeigt die Meldung x mal hintereinander an.	requirement	(1) Weak-Word (2) Fehlen einer quantifizierenden Angabe	(1) Weak-Words sind mehrdeutige Wörter, die je nach Kontext zu Interpretationsspielraum führen können. (2) Wann immer möglich müssen Quantifizierungen angegeben werden. Ansonsten wird Interpretationsspielraum eröffnet.	hoch
TSJK-D42	Funktion A ist kürzer als 3 Sekunden aktiv.	requirement	(1) Weak-Word	(1) Weak-Words sind mehrdeutige Wörter, die je nach Kontext zu Interpretationsspielraum führen können.	unkritisch
TSJK-D50	Funktion B ist eine kürzere Zeit als Funktion A aktiv.	requirement	(1) Weak-Word (2) Fehlen einer quantifizierenden Angabe	(1) Weak-Words sind mehrdeutige Wörter, die je nach Kontext zu Interpretationsspielraum führen können. (2) Wann immer möglich müssen Quantifizierungen angegeben werden. Ansonsten wird Interpretationsspielraum eröffnet.	mittel hoch
TSJK-D43	Funktion C ist eine kürzere Zeit aktiv.	requirement	(1) Weak-Word (2) Fehlen einer quantifizierenden Angabe	(1) Weak-Words sind mehrdeutige Wörter, die je nach Kontext zu Interpretationsspielraum führen können. (2) Wann immer möglich müssen Quantifizierungen angegeben werden. Ansonsten wird Interpretationsspielraum eröffnet.	hoch

Abbildung 23: Ausschnitt aus der erstellten Testsatzsammlung

5.5.3.2 Ergebnisse der Evaluierung anhand der Testsatzsammlung

Die Testsatzsammlung wurde von allen Werkzeugen, die in den Abschnitten 5.5.1 und 5.5.2 beschrieben sind, analysiert. Anschließend wurden die Ergebnisse der Analysen manuell gesichtet und es wurde abgeglichen, welche Phänomene aus dem Qualitätsmodell von den Werkzeugen tatsächlich identifiziert werden. Zusätzlich wurde ausgewertet, wie effizient die Analysen der Werkzeuge sind, d.h. ob wirklich nur die kritischen Instanzen eines Phänomens gefunden werden oder ob bei jeder Instanz eines Phänomens eine Warnung ausgegeben wird.

In Tabelle 9 ist abgebildet, welches Werkzeug welche Vorverarbeitungsschritte durchführt. Hierbei stehen die Prozessierungsschritte Tokenisierung, Lemmatisierung, Part-of-Speech-Tagging (= POS-Tagging), Morphologie-Tagging und Parsing im Vordergrund (vgl. hierzu Kapitel 6).

	DESIRE®	ReqCheck	Requirements Quality Suite	Requirements Scout	LanguageTool	TextLab	CLAT	Acrolinx
Tokenisierung	↑	↑	↑	↑	↑	↑	↑	↑
Lemmatisierung	↓	↑	↑	↑	↑	↑	↑	↑
POS-Tagging	↓	↑	↓	↑	→	↑	↑	↑
Morphologie-Tagging	↓	↑	↓	→	→	↓	↑	↑
Parsing	↓	↑	↓	↑	↓	↓	↑	↑

Tabelle 9: Vorverarbeitungsschritte der präsentierten Werkzeuge
 (grün (↑) = ja, gelb (→) = durch Listenabgleich, rot (↓) = nein)

In Tabelle 10 sind alle Phänomene aus dem Qualitätsmodell und alle verwendeten Werkzeuge aufgelistet. Ob ein Werkzeug ein bestimmtes Phänomen analysiert, wird durch eine farbliche Markierung und einen Pfeil gekennzeichnet. Hierbei wird nicht nur auf die generelle Erkennbarkeit der Phänomene geachtet, sondern auch darauf, ob die kritischen Fälle zuverlässig erkennbar sind.

Eine grüne Markierung in der Spalte “Erkennbarkeit generell” bedeutet, dass das entsprechende Phänomen durch das entsprechende Werkzeug geprüft wird. Ist in dieser Kategorie eine Spalte rot markiert, bedeutet dies, dass das Phänomen durch das entsprechende Werkzeug nicht analysiert wird. Eine gelbe Markierung kennzeichnet Phänomene, die nur teilweise abgeprüft werden. Dies ist beispielsweise der Fall, wenn nur eine Teilmenge der möglichen Instanzen geprüft wird. Das Werkzeug *TextLab* prüft beispielsweise nur die unpersönliche Konstruktion *man*. Laut Qualitätsmodell fällt in diese Kategorie auch das unpersönliche *es*, welches durch *TextLab* nicht geprüft wird. Eine gelbe Markierung kennzeichnet aber auch Phänomene, die eher zufällig analysiert werden. Dies ist beispielsweise bei der doppelten Verneinung der Fall. Wenn ein Werkzeug grundsätzlich das Wort *nicht* als kritisch bewertet, wird auch die doppelte Verneinung gefunden. Jedoch nicht mit dem primären Ziel die doppelte Verneinung zu finden.

In der Kategorie “Kritische Fälle zuverlässig erkennbar” kennzeichnet eine grüne Markierung die Phänomene, welche durch das entsprechende Werkzeug gefunden werden und dabei die unkritischen Instanzen dieser Phänomene zuverlässig von den kritischen Instanzen unterschieden werden. Bei den Weak-Words ist dies beispielsweise der Fall, wenn eine Kontextanalyse durchgeführt wird und nicht bei jedem Auftreten eines Weak-Words eine

Phänomen	Erkennbarkeit generell								Kritische Fälle zuverlässig erkennbar							
	DESIRE®	ReqCheck	Requirements Quality Suite	Requirements Scout	LanguageTool	TextLab	CLAT	Acrolinx	DESIRE®	ReqCheck	Requirements Quality Suite	Requirements Scout	LanguageTool	TextLab	CLAT	Acrolinx
Fehlen einer quantifizierenden Angabe	↓	↓	↓	→	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Nominalisierung	↑	↓	↓	↓	↓	↑	↓	↑	↓	↓	↓	↓	↓	↓	↓	↓
Weak-Word	↑	↓	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Unpersönliche Konstruktion	↓	↑	↓	↑	↓	→	↑	→	↓	↓	↓	↓	↓	↓	↓	↓
Unvollständiger Satz	↓	↓	↓	↓	↓	↓	↓	→	↓	↓	↓	↓	↓	↓	↓	↓
Unvollständige Bedingung	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Passiv ohne Agens	↑	↑	↑	↑	↑	↑	↑	↑	↓	↓	→	→	↓	↓	→	→
Information statt Anforderung	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Modalverb „sollte(n)“	↑	↑	↑	↑	↓	↑	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓
Verallgemeinerung/ Universalquantor	↓	↓	→	↓	↓	↓	↓	↓	↓	↓	→	↓	↓	↓	↓	↓
Polysemie	↓	↓	↓	↓	↓	↓	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓
Langes Kompositum	↓	↑	↓	↓	↑	↑	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓
Attribut vs. Adverbial	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Gemischte Koordinationen	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Koordination + Attribut	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Präpositionalphrasen und Nominalphrasen	↓	↓	↓	↓	↓	↓	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓
Genitivus subjectivus vs. Genitivus objectivus	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Subjekt vs. Objekt	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Pronominaler Bezug	↓	↓	↑	↑	↓	↓	↑	→	↓	↓	↓	↓	↓	↓	↓	↓
Verschachtelte Bedingungen	↓	→	→	→	→	→	→	→	↓	→	→	→	→	→	→	→
Verschachtelter/ Zu langer Satz	↓	↑	↑	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↑	↑	↑
Langer Klammereinschub	↓	↓	↓	↓	↓	↓	→	↑	↓	↓	↓	↓	↓	↓	↓	↓
Aufzählung ohne Liste	↓	↓	↓	↓	↓	↓	→	↑	↓	↓	↓	↓	↓	↓	↓	↓
Doppelte Negation	↓	↓	↓	→	↓	↓	→	↑	↓	↓	↓	→	↓	↓	→	→
Pränominales Partizip	↓	↓	↓	↓	↓	↓	↓	↑	↓	↓	↓	↓	↓	↓	↓	↓
Komplexe Präposition	↓	↓	↓	↓	↓	↑	↓	↓	↓	↓	↓	↓	↓	↑	↓	↓
Passiv mit Agens	↓	↓	↓	↑	↓	↓	↑	↑	↓	↓	↓	↑	↓	↓	↑	↑
Mehr als eine Handlung pro Satz	↓	↓	↓	↓	↓	→	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Globaler Widerspruch	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Lokaler Widerspruch	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Bezugsdefizit	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Inhalt der Anforderung ist falsch	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Prüf. der Ums. der Anf. ist nicht effektiv durchführbar	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Prüf. der Ums. der Anf. ist nicht effizient durchführbar	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓

Tabelle 10: Evaluierungsergebnisse der Analysen bestehender Textprüfwerkzeuge (grün (↑) = ja, gelb (→) = teilweise, rot (↓) = nein)

Warnung ausgegeben wird, sondern nur wenn für das Verständnis wichtige Information fehlt.

Gelbe Markierungen kennzeichnen Phänomene, bei welchen das entsprechende Werkzeug zwar die kritischen von den unkritischen Instanzen in manchen Situationen unterscheidet, die Analysen jedoch noch nicht eine ausreichend hohe Güte haben; d.h. die Werkzeuge produzieren eine zu hohe Anzahl an Fehlwarnungen. Hierbei spielt auch das Kontext-, Welt- und Expertenwissen eine Rolle. Grammatikalisch mehrdeutige Fälle oder Fälle, bei welchen eine für das Verständnis wichtige Information fehlt, können teilweise durch die Werkzeuge identifiziert werden. Jedoch können diese Mehrdeutigkeiten durch Kontext-, Welt- oder Expertenwissen aufgelöst oder vermeintlich fehlende Information hergeleitet werden. Dies ist beispielsweise bei dem Phänomen Passiv der Fall. Passivsätze ohne Agens können zwar durch die Werkzeuge gefunden werden, jedoch ist nicht jeder Passivsatz ohne Agens generell als kritisch zu betrachten. Oft kann das Agens durch Kontext-, Welt- oder Expertenwissen hergeleitet werden (vgl. Abschnitt 4.2.1.6 und 8.3).

Rote Markierungen kennzeichnen, dass ein Phänomen entweder gar nicht oder nicht kontextabhängig analysiert wird. Das Werkzeug *Requirements Quality Suite* beispielsweise analysiert Weak-Words. Aus diesem Grund ist das Kästchen in der Kategorie "Erkennbarkeit generell" grün markiert. Da aber keine Kontextanalyse, sondern ein reiner Wortlistenabgleich durchgeführt wird und somit bei jeder Instanz eines Weak-Words gewarnt wird, ist das Phänomen in der Kategorie "Kritische Fälle zuverlässig erkennbar" rot markiert.

Es ist zu beachten, dass die Bewertung allein anhand der Testsatzsammlung durchgeführt wurde und die Werkzeuge hierbei in ihrer Standard-Ausführung getestet wurden. Eine Markierung, dass ein Phänomen nicht geprüft wird, bedeutet nicht gleichzeitig, dass das entsprechende Werkzeug nicht die technischen Voraussetzungen dafür erfüllt, es analysieren zu können. Die Prüfungen werden lediglich zum Zeitpunkt der Evaluierung nicht in der Standardversion angeboten. Weiterhin ist zu beachten, dass die Werkzeuge zusätzliche Sprachprüfungen durchführen, die im Qualitätsmodell nicht durch Phänomene repräsentiert werden. Dabei handelt es sich beispielsweise um Regeln für eine einheitliche Struktur der Anforderungen (z.B. Bedingung an den Anfang des Satzes setzen). Diese Phänomene sind nicht relevant für die Testbarkeit einer Anforderung und wurden deshalb nicht in die Evaluierung mit aufgenommen.

Auffällig ist, dass viele Phänomene, die in die Kategorie der Mehrdeutigkeiten fallen, rot markiert sind und somit nicht geprüft werden. Die untersuchten Werkzeuge behandeln größtenteils Phänomene, bei welchen das Problem der fehlenden Information, oder das

Problem der Komplexität vorliegt. Ebenfalls auffällig ist, dass alle Werkzeuge das Phänomen Passiv analysieren. Jedoch führen nur die Werkzeuge *Requirements Quality Suite*, *Requirements Scout*, *CLAT* und *Acrolinx* eine Analyse dahingehend durch, ob bei der Verwendung des Passivs das Agens angegeben wird. Nur die Instanzen ohne Agens werden von diesen Werkzeugen als kritisch markiert. Wie bereits oben erwähnt sind jedoch nicht alle Fälle von Passivsätzen ohne Agens kritisch.

Alle Werkzeuge behandeln außerdem das Phänomen *Verschachtelter/Zu langer Satz*. Ab einer bestimmten Satzlänge steigt die Komplexität eines Satzes so stark an, dass die Verständlichkeit für den Leser sinkt. Alle Werkzeuge haben hierfür eine eigens definierte Grenze. Da die Satzlänge (= Menge der Wörter) von den Werkzeugen analysiert wird, werden auch verschachtelte Bedingungen von den Werkzeugen als kritisch interpretiert. Da die von den Werkzeugen produzierte Warnung aber nur auf die Länge des Satzes hinweist und nicht direkt auf die verschachtelte Bedingung, ist das Phänomen bei allen Werkzeugen, ausgenommen bei *DESIRE*, gelb markiert.

Widersprüche werden durch keines der Werkzeuge analysiert. Dies war zu erwarten, da eine Analyse auf Widersprüche sehr komplex und kaum zu realisieren ist, da hierfür Domänenwissen benötigt wird.

Phänomen	Erkennbarkeit generell								Kritische Fälle zuverlässig erkennbar							
	DESIRE®	ReqCheck	Requirements Quality Suite	Requirements Scout	LanguageTool	TextLab	CLAT	Acrolinx	DESIRE®	ReqCheck	Requirements Quality Suite	Requirements Scout	LanguageTool	TextLab	CLAT	Acrolinx
Weak-Word	↑	↓	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Unpersönliche Konstruktion	↓	↑	↓	↑	↓	→	↑	→	↓	↓	↓	↓	↓	↓	↓	↓
Passiv	↑	↑	↑	↑	↑	↑	↑	↑	↓	↓	→	→	↓	↓	→	→
Modalverb „sollte(n)“	↑	↑	↑	↑	↓	↑	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓
Pronominaler Bezug	↓	↓	↑	↑	↓	↓	↑	→	↓	↓	↓	↓	↓	↓	↓	↓

Tabelle 11: Evaluierungsergebnisse der Analysen bestehender Textprüfwerkzeuge an ausgewählten Phänomenen aus dem Qualitätsmodell (grün (↑) = ja, gelb (→) = teilweise, rot (↓) = nein)

In Tabelle 11 sind nochmals die Phänomene aus dem Qualitätsmodell separat aufgelistet, welche in Abschnitt 4.8 als relevant identifiziert wurden und in dieser Arbeit genauer untersucht werden. Weak-Words werden nur von den Werkzeugen aus dem Requirements

Engineering behandelt (ausgenommen von *ReqCheck*). Jedoch führt keines der Werkzeuge eine kontextabhängige Analyse durch.

Unpersönliche Konstruktionen werden durch *ReqCheck*, *Requirements Scout* und *CLAT* analysiert und teilweise durch *TextLab* und *Acrolinx*. Keines der Werkzeuge kann die kritischen Instanzen zuverlässig herausfiltern, d.h. es wird bei jeder Instanz gewarnt. Dies ist auch bei den pronominalen Bezügen der Fall. Lediglich vier Werkzeuge widmen sich diesem Phänomen. Jedoch wird auch hier jede Instanz eines Pronomens als kritisch interpretiert, ohne zu prüfen, ob es mehrere Nomina gibt, mit welchen das Pronomen in Genus, Numerus und Person kongruiert. *Acrolinx* analysiert ausschließlich substituierende Demonstrativpronomina. Aus diesem Grund ist das Phänomen bei diesem Werkzeug gelb markiert. Das Phänomen *sollte* wird durch alle Werkzeuge, ausgenommen von *LanguageTool*, behandelt. Jedoch wird auch hier von allen Werkzeugen ein reiner Wortlistenabgleich durchgeführt, ohne den Satzkontext des Wortes *sollte* in die Analyse miteinzubeziehen.

Das Phänomen Passiv wird von allen Werkzeugen behandelt. Wie bereits beschrieben, können die Instanzen von Passivsätzen herausgefiltert werden, welche das Agens explizit benennen (beispielsweise durch eine *von-* oder eine *durch-*Phrase). Eine solche Analyse führen die Werkzeuge *Requirements Quality Suite*, *Requirements Scout*, *CLAT* und *Acrolinx* durch. Jedoch ist es nicht der Fall, dass ein Passivsatz ohne Nennung des Agens immer als kritisch zu bewerten ist. In den meisten Fällen kann durch Kontext-, Welt- oder Expertenwissen das Agens hergeleitet werden. Bei den Werkzeugen, die die Kritikalität eines Passivsatzes danach bewerten, ob ein Agens genannt wird, ist das Phänomen Passiv daher gelb markiert. Die anderen Werkzeuge geben bei der Verwendung von Passiv immer eine Warnung aus, unabhängig davon, ob ein Agens genannt wird oder nicht. Die kritischen Instanzen sind also bei allen Werkzeugen nicht zuverlässig erkennbar.

Lediglich das Werkzeug *Requirements Scout* analysiert ganzheitlich alle in Abschnitt 4.8 als relevant identifizierten Phänomene. Das einzige Phänomen, das durch alle Werkzeuge analysiert wird, ist das Passiv. Das Phänomen *Modalverb sollte* wird durch alle Werkzeuge ausgenommen von *LanguageTool* analysiert. Weak-Words, unpersönliche Konstruktionen und pronominale Bezüge werden nur von einem Teil der Werkzeuge analysiert. Keines der Werkzeuge stellt Analysen bereit, die zuverlässig zwischen den kritischen und den unkritischen Instanzen unterscheidet.

5.5.4 Zusammenfassender Überblick über Lastenheft-Prüfwerkzeuge

In diesem Abschnitt wurden bereits existierende Textprüfwerkzeuge aus dem Bereich der Sprachtechnologie und aus dem Bereich des Requirements Engineering vorgestellt. Die bereitgestellten Sprachprüfungen der Werkzeuge wurden anhand einer Testsatzsammlung,

welche kritische und unkritische Anforderungen enthält, bewertet. Somit konnte untersucht werden, welches Werkzeug welche Phänomene aus dem Qualitätsmodell mit welcher Analysegröße behandelt.

Die Analysen der Werkzeuge sind zum größten Teil auf Phänomene ausgerichtet, welche in die Kategorie der Komplexität oder der fehlenden Information fallen. Eine zuverlässige Unterscheidung zwischen kritischen und unkritischen Instanzen der Phänomene, wird von keinem der Werkzeuge durchgeführt.

Die Unterscheidung zwischen kritischen und unkritischen Instanzen ist ein wichtiger Punkt bei der Entwicklung eines Prüfwerkzeuges. Werden zu viele Anforderungen fälschlicherweise angemahnt, so steigt der Zeitaufwand bei der Durchsicht der zu überarbeitenden Anforderungen und die Akzeptanz seitens der Benutzer sinkt. Da keines der existierenden Werkzeuge eine hinreichende Analysegröße für die in Abschnitt 4.8 identifizierten Phänomene bereitstellt, ist es sinnvoll diese Phänomene und die Satzkontexte, in welchen sie auftreten, im Rahmen dieser Arbeit genau zu analysieren. Hierbei soll festgestellt werden, ob Prüfungen bereitgestellt werden können, die zuverlässig die kritischen von den unkritischen Instanzen unterscheiden können.

Bei der Entwicklung von Prüfungen muss darauf geachtet werden, dass weder zu viele Fehlwarnungen, noch zu viele Fehlentwarnungen entstehen. Werden zu viele Fehlwarnungen produziert, sinkt die Akzeptanz für das Werkzeug bei den Benutzern, da zu viel Nacharbeit investiert werden muss (vgl. hierzu Kapitel 7). Zu viele Fehlentwarnungen haben zwar keinen direkten Einfluss auf die Akzeptanz des Werkzeugs, bedeuten aber, dass Schwachstellen in den Anforderungen nicht identifiziert und erst später im Projekt oder gar nicht auffallen und behandelt werden. Die Fehlwarnungen und Fehlentwarnungen in der Waage zu halten stellt eine Herausforderung bei der Entwicklung von automatisierten Prüfmethoden dar.

5.6 Weitere automatisierte Ansätze

In Abschnitt 5.5 wurden frei verfügbare und kommerzielle Werkzeuge präsentiert, welche natürlichsprachliche Texte auf Verständlichkeit überprüfen. Im vorliegenden Abschnitt werden weitere automatisierte Ansätze präsentiert, bei welchen allerdings kein verfügbares Werkzeug entstanden ist. Diese Ansätze wenden zu einem gewissen Anteil bei der Analyse von sprachlichen Phänomenen computerlinguistische Technologien an, weshalb sie für die vorliegende Arbeit ebenfalls von Relevanz sind. Im vorliegenden Abschnitt wird der Ansatz von [Sto13], der Ansatz von [GCK10] und der Ansatz von [KLM09] präsentiert. Es existieren noch weitere Ansätze, durch welche natürlichsprachliche Anforderungen auf

Verständlichkeit überprüft werden, jedoch werden dort vorwiegend Wortlistenabgleiche durchgeführt und keine Technologien aus der Computerlinguistik genutzt.

In [Sto13] wurden Analysen dazu durchgeführt, wie Techniken aus dem Bereich der Computerlinguistik dazu eingesetzt werden können, natürlichsprachliche Anforderungen zu überprüfen. In der Arbeit werden die folgenden Techniken eingesetzt: Lemmatisierung, Part-of-Speech-Tagging und Parsing (vgl. hierzu Kapitel 6). In [Sto13] werden fünf Prüfungen präsentiert: *Prüfung der Grammatikalität*, *Prüfung der Verbindlichkeit*, *Prüfung der Atomarität*, *Forbidden-Word-Prüfung* und *Weak-Word-Prüfung*.

Bei der *Prüfung auf Grammatikalität* wird zunächst eine Satzenderkennung durchgeführt, „welche auf der Analyse der Interpunktion basiert und mithilfe von regulären Ausdrücken beschrieben werden kann“ [Sto13]. Zusätzlich wird auch überprüft, ob ein Satz grammatikalisch korrekt ist. Hierfür wurde eine kontextfreie Phrasenstrukturgrammatik für deutschsprachige Anforderungen definiert, welche zum Zweck der Satzanalyse eingesetzt wird. Durch die Implementierung einer solchen Grammatik kann laut [Sto13] anhand von Parsing-Algorithmen automatisiert eine Grammatikprüfung durchgeführt werden und somit können ungrammatische Sätze identifiziert werden.

Die *Prüfung der Verbindlichkeit* überprüft, ob Anforderungen bestimmte Schlüsselwörter enthalten, welche die Verbindlichkeit einer Anforderung kennzeichnen (vgl. hierzu auch Abschnitt 4.2.2.2). Enthält eine Anforderung das Lemma *muss* oder *nicht dürfen*, ist die Anforderung als rechtlich bindend markiert und der Autor muss keine Überarbeitung der Anforderung durchführen. Wird eine Infinitivkonstruktion mit *‘sein + zu’* erkannt, wird der Satz automatisch in eine Konstruktion mit *muss* umgewandelt. Somit soll den Sätzen eine einheitliche Struktur bezüglich der verbindlich machenden Verben gegeben werden. Wird keines der definierten Schlüsselwörter in einem Satz gefunden, wird eine Warnung an den Benutzer ausgegeben mit einer Meldung, dass er die Anforderung entsprechend durch die Verwendung von Schlüsselwörtern nach Verbindlichkeit markieren soll.

Bei der *Prüfung der Atomarität* werden Sätze danach überprüft, ob sie mehrere Vollverben enthalten. Hierfür wird auf ein Lexikon, welches eine Liste von Vollverben enthält, zugegriffen. Enthält ein Satz mehrere Vollverben, welche nicht identisch sind, wird an den Benutzer eine Warnung ausgegeben. Diese Prüfung baut auf der *Prüfung auf Grammatikalität* auf. Somit wird auch hier auf Parsing-Information zugegriffen und zusätzlich mit Wortlisten gearbeitet.

Eine *Forbidden-Word-Prüfung* und eine *Weak-Word-Prüfung* werden durch den Ansatz ebenfalls realisiert. *Forbidden-Words* sind verbotene Wörter, die in Anforderungstexten nicht verwendet werden dürfen. *Weak-Words* hingegen sind Wörter, die nur in bestimm-

ten Kontexten nicht verwendet werden dürfen oder um eine Angabe erweitert werden müssen, durch welche die Wörter eindeutig werden (vgl. hierzu Abschnitt 4.2.1.2 und Abschnitt 8.1). Die in [Sto13] entwickelte Analyse führt Wortlistenabgleiche durch. Ist ein *Forbidden-Word* oder ein *Weak-Word*, welches in einer Liste abgelegt ist, im Text vorhanden, wird eine Warnung an den Autor ausgegeben. Für diese Prüfung wird folglich keine computerlinguistische Information einbezogen und es wird somit jede Instanz jedes *Weak-Words* angemahnt.

Die in [Sto13] präsentierten Prüfungen beziehen die bereitgestellten computerlinguistischen Technologien (Lemmatisierung, Part-of-Speech-Tagging und Parsing) nur sehr wenig ein. Lediglich für die Analyse, ob ein Satz grammatikalisch korrekt ist, wird Parsing-Information und Part-of-Speech-Tagging miteinbezogen. Für die weiteren Analysen werden Wortlisten verwendet.

In der vorliegenden Arbeit wird keine Prüfung auf grammatikalisch korrekte Sätze durchgeführt. Dies steht nicht im Fokus der Arbeit. Eine Prüfung auf Atomarität wird ebenfalls nicht bereitgestellt. Der in [Sto13] präsentierte Ansatz ist eine gute Herangehensweise potentiell nicht-atomare Sätze zu erkennen, jedoch kann nicht generell festgelegt werden, dass mehrere Vollverben in einem Satz generell darauf hinweisen, dass ein Satz nicht atomar ist. Eine zuverlässige automatische Erkennbarkeit der kritischen Instanzen ist nicht möglich, weshalb das Phänomen der Atomarität für tiefer gehende Analysen in der vorliegenden Arbeit ausgeschlossen wird.

Eine Prüfung auf die rechtliche Verbindlichkeit im Bezug auf das Modalverb *sollte(n)* wird in der vorliegenden Arbeit durchgeführt, ebenso eine Prüfung auf Weak-Words. Jedoch soll nicht nur geprüft werden, ob bestimmte Wörter in einer Anforderung vorhanden sind, sondern der textuelle Kontext soll in die Analyse miteinbezogen werden, um die kritischen Instanzen zuverlässig herauszufiltern. Der in [Sto13] präsentierte Ansatz produziert in der technischen Umsetzung viele Fehlwarnungen, weil keine tiefgehenden Kontextanalysen durchgeführt werden. Der Fokus der Prüfungen liegt auch nicht auf der Testbarkeit von Anforderungen.

In [GCK10] wird ein weiterer Ansatz vorgestellt, um mehrdeutige Strukturen in natürlichsprachlichen englischen Anforderungen zu identifizieren. Durch diesen Ansatz werden u.a. Weak-Words, pronominale Bezüge, Schrägstriche und Passivsätze behandelt. Lediglich bei der Passivsatzanalyse und der Analyse auf Adjektive und Adverbien kommt eine Technik aus der Computerlinguistik zum Einsatz: Part-of-Speech-Tagging (= Wortarten-Tagging). Für die Passivanalyse wird eine Regel präsentiert, welche Part-of-Speech-Tags in Kombination mit regulären Ausdrücken verwendet, und durch die somit Passivsätze

automatisiert erkannt werden können. Außerdem werden durch den Einsatz von Part-of-Speech-Tags alle vorhandenen Adjektive und Adverbien in einem Text erkannt und dem Autor zur Kontrolle aufgezeigt. Die weiteren Analysen basieren alle auf regulären Ausdrücken, die jedoch keine computerlinguistischen Annotationen miteinbeziehen, sondern lediglich auf Einzelwörtern basieren. Auch durch diesen Ansatz werden computerlinguistische Technologien nicht vollumfänglich für die Analysen genutzt.

In [KLM09] und [Kna12] wird das Werkzeug *HeRA* (*Heuristic Requirements Assistant*) vorgestellt, welches Anforderungen bereits bei der Erstellung analysiert. „Mit Hilfe von heuristischen Kritiken kann ein Rechner auf Basis von Erfahrungen und Best Practices Verbesserungspotential in Anforderungsdokumenten aufzeigen“ [Kna12]. Die in HeRA enthaltenen Regeln basieren auf Erfahrungswerten von Benutzern. Regeln können durch den Benutzer während der Verwendung des Werkzeugs hinzugefügt werden und die hinzugefügten Regeln werden sogleich angewandt [KLM09]. Von Erfahrungswerten unterschiedlicher Benutzer kann hierbei profitiert werden. Weiterhin werden für die Analyse der Anforderungen Glossare verwendet, um ein einheitliches Verständnis bestimmter Begriffe zwischen unterschiedlichen Benutzern herzustellen. In HeRA ist beispielsweise eine Prüfung auf Weak-Words möglich. Jedoch basiert eine solche Analyse auf reinen Wortlistenabgleichen ohne Einbezug des Satzkontextes. Tiefgehende computerlinguistische Analysen werden in diesem Werkzeug nicht durchgeführt.

Die präsentierten Ansätze verwenden für die Analyse von Anforderungstexten zum Teil computerlinguistische Technologien. Jedoch werden diese in ihrer Mächtigkeit nicht voll ausgeschöpft. Der Großteil der Prüfungen basiert auf Wortlistenabgleichen. Ein Ziel der vorliegenden Arbeit liegt darauf, bei den linguistischen Analysen über reine Wortlistenabgleiche hinauszugehen, da durch Wortlistenabgleiche zu viele Fehlwarnungen erzeugt werden und dies einen Einfluss auf die Akzeptanz des Nutzers für das Werkzeug hat (vgl. Kapitel 7). Computerlinguistische Analysen sollen dabei helfen Analysen zur Verfügung zu stellen, bei welchen nur diejenigen Instanzen dem Autor zur Kontrolle zurückgegeben werden, welche wirklich kritisch sind.

5.7 Zusammenfassung

Im vorliegenden Kapitel wurden Methoden und Werkzeuge zur Sicherstellung der Qualität in Anforderungen vorgestellt. Es wurden Techniken präsentiert, die schreibbegleitend durchgeführt werden können und Techniken und Werkzeuge, welche auf bereits existierende Texte angewandt werden können. Schreibregeln, Satzschablonen und die Nutzung kontrollierter Sprachen sind Techniken, die schreibbegleitend durchgeführt werden können. Manuelle Review-Techniken und werkzeuggestützte Prüfungen von natürlichsprachlichen

Texten werden auf bereits existierenden Texten durchgeführt. Alle Techniken haben Vor- und Nachteile, welche in den einzelnen Abschnitten beschrieben wurden.

Es wurden vier Werkzeuge aus dem Bereich des Requirements Engineering und vier Werkzeuge aus dem Bereich der Sprachtechnologie vorgestellt, welche natürlichsprachliche Texte auf Verständlichkeit überprüfen. Anhand einer Testsatzsammlung wurden die Werkzeuge evaluiert, d.h. es wurde analysiert, welche Phänomene aus dem Qualitätsmodell durch welches Werkzeug behandelt werden und wie zuverlässig diese Analysen sind. Keines der Werkzeuge analysiert die in Kapitel 4.8 als relevant identifizierten Phänomene aus dem Qualitätsmodell in der Art, dass die kritischen von den unkritischen Instanzen zuverlässig automatisiert unterschieden werden können und somit nur eine geringe Menge an Fehlwarnungen und Fehlentwarnungen entsteht.

6 Erstellung und Annotation von Lastenheft-Korpora

Für die geplanten sprachlichen Analysen musste zunächst eine Datengrundlage in Form von annotierten Korpora aufgebaut werden. In den folgenden Abschnitten wird beschrieben, wie bei dem Aufbau der Korpora vorgegangen wurde. In Abschnitt 6.1 wird zunächst ein Überblick darüber gegeben, wie die verwendeten Lastenhefte aufgebaut sind, d.h. welche Strukturmerkmale sie aufweisen. Weiterhin wird die Datenbank DOORS beschrieben, in welcher die Lastenhefte abgelegt sind. In Abschnitt 6.2 wird der Export der Lastenhefte aus DOORS beschrieben und das Format, in welches die extrahierten Texte umgewandelt wurden, wird vorgestellt. In Abschnitt 6.3 folgt die Beschreibung der computerlinguistischen Annotation der Lastenhefte. Dabei wird auf die einzelnen Schritte der Annotation eingegangen: Tokenisierung, Lemmatisierung, Part-of-Speech-Tagging, Morphologie-Tagging und Parsing. Außerdem wird eine Evaluation zur Annotationsqualität der Werkzeuge präsentiert. Abschnitt 6.4 gibt anschließend einen Überblick über die Größe der aufgebauten Korpora, d.h. die Anzahl der Sätze und Tokens. In Abschnitt 6.5 wird abschließend beschrieben, wie die annotierten Lastenhefte in ein als computerlinguistisches Korpus verarbeitbares Format umgewandelt wurden.

6.1 Struktur von Lastenheften in der Datenbank *DOORS*

Die ersten Schritte bei dem Aufbau der Korpora waren folgende: die Lastenhefte, welche in der Datenbank *IBM Rational DOORS* abgelegt sind, mussten exportiert werden, und die exportierten Lastenhefte mussten anschließend in ein Format umgewandelt werden, welches durch die computerlinguistischen Werkzeuge in den darauffolgenden Schritten weiterverarbeitet werden kann. Hierfür soll zunächst beschrieben werden, welche Strukturen die Lastenhefte in der Datenbank aufweisen, die für den nachfolgenden Export relevant sind.

In DOORS stellt jedes Lastenheft ein sogenanntes Modul dar. In Abbildung 24 ist ein Ausschnitt eines solchen Moduls in DOORS zu sehen.

Ein Lastenheft besteht aus mehreren Spalten, welche unterschiedliche Attribute darstellen und mit Werten befüllt werden können. In Abbildung 24 sind u.a. die für den Aufbau der Korpora relevanten Attribute zu sehen: *SourceID*, *Anforderungen* und *Object Type*. In der Spalte *Anforderungen* steht der Text einer Anforderung. In der Spalte *SourceID* wird jeder Anforderung eine eindeutige Identifikationsnummer (=ID) zugewiesen. Beginnt eine ID mit dem Kürzel *STLH* oder *SB*, stammt diese Anforderung aus der Lastenheftvorlage. Diese Anforderungen sind bereits auf ihre sprachliche Qualität geprüft und beinhalten keine mehrdeutigen Strukturen. Wurde eine Anforderung hingegen neu von einem Lastenheftautor verfasst, so steht ein anderes Kürzel, und der Anforderungstext muss somit

SourceID	Anforderungen	Object Type	Edit Type
STLH-259A	1.1.2 Logistik-Komponentenlastenheft	heading	don't edit
STLH-260A	Zu jeder Teilefamilie erhält der Auftragnehmer über den Einkäufer des Auftraggebers ein Logistikkonzept. Dieses ist bauteilabhängig, auftragnehmerabhängig und auslieferstandortabhängig.	information	don't edit
STLH-261B	Der Auftragnehmer muss die Anforderungen bezüglich des Logistikkonzeptes gemäß der Beschreibung im separaten Logistik-KLH erfüllen.	requirement	don't edit
STLH-3773D	Bezüglich der allgemeinen Logistikanforderungen muss der Auftragnehmer die Anforderungen im Kapitel ABC-1 "Logistik" im Dokument [STLH-3] einhalten.	requirement	don't edit
STLH-4643A	Die Unterlagen des Logistik-KLH können durch Auftragnehmer über das System „Source Package“ bezogen werden.	information	don't edit

Abbildung 24: Ausschnitt eines Lastenheftes in der Datenbank *DOORS*

kontrolliert werden. In Abbildung 24 sind lediglich Anforderungen zu sehen, die aus der Lastenheftvorlage stammen.

Durch das Attribut *Object Type* kann jeder Anforderung ein Wert zugewiesen werden, welcher Aufschluss darüber gibt, von welcher Art die Anforderung ist, d.h. ob es sich um eine Anforderung handelt, die der Auftragnehmer realisieren muss, oder ob es sich beispielsweise lediglich um eine Zusatzinformation handelt. Das Attribut *Object Type* kann hierbei u.a. folgende Werte annehmen: *requirement*, *information*, *predefinition* und *heading*. Andere Werte sind auch möglich, wenn sie vom Lastenheftautor definiert werden. Mit dem Wert *requirement* werden Anforderungen markiert, die der Auftragnehmer realisieren muss, es handelt sich hierbei nicht um Zusatzinformationen. Zusatzinformationen hingegen können mit dem Wert *information* angegeben werden. Diese Anforderungen sind zwar nicht rechtlich verbindlich, sollten aber nach Möglichkeit ebenfalls keine sprachlichen Mängel enthalten. Mit dem Wert *predefinition* werden diejenigen Anforderungen markiert, die Festlegungen enthalten, und mit dem Wert *heading* werden Überschriften markiert. Die Anforderungen, die mit dem *Object Type* *requirement*, *information* und *predefinition* markiert sind, müssen einer sprachlichen Überprüfung unterzogen werden. Anforderungen, bei welchen der *Object Type* nicht angegeben ist, müssen ebenfalls überprüft werden.

6.2 Export von Lastenheften aus der Datenbank *DOORS*

Für den Aufbau der Korpora wurden aus den Lastenheften nicht nur die Anforderungstexte extrahiert. Pro Anforderung wurden auch die jeweilige *ID* und der *Object Type* übernommen. Auf diese Weise können die Anforderungen auch später noch eindeutig identifiziert werden. Es wurden lediglich diejenigen Anforderungstexte extrahiert, welche den *Object Type* *requirement*, *information* oder *predefinition* besitzen sowie die Texte, bei welchen kein *Object Type* angegeben wurde. Die Anforderungen aus der Lastenheftvorlage sind bereits sprachlich korrigiert und müssen aus diesem Grund nicht in die Korpora auf-

genommen werden. Ein weiterer Grund, wieso diese Anforderungen nicht in die Korpora mitaufgenommen wurden, ist, dass die Standardanforderungen in jedem Lastenheft verwendet werden und somit viele identische Instanzen in den Korpora vorkommen würden. Dies würde die quantitativen Analyseergebnisse beeinflussen.

Jedes Lastenheft wurde in einer Einzeldatei gespeichert und zusätzlich in einer Gesamtdatei abgelegt. Die Texte wurden in ein Format gebracht, welches Grundlage für den verwendeten Tokenizer ist, mit dem die Lastenhefte im nächsten Schritt weiterverarbeitet werden sollten. Dieses Format ist in Abbildung 25 dargestellt.

```
///AGS-37###requirement/// In diesem Zusammenhang müssen auch sämtliche
durch die Funktion des aktiven Gurtschlosses erzeugten Geräusche in
Mercedes-Fahrzeugen hochwertig gestaltet werden.
NEWLINE
///AGS-38###requirement/// Die zu erreichende Geräuschqualität des
aktiven Gurtschlosses soll auf Basis realer Fahrzeugmessungen definiert
werden.
NEWLINE
///AGS-39###requirement/// Neben den verbalen Beschreibungen und
quantitativen Angaben im Lastenheft sollen die festgelegten Zielgeräusche
insbesondere die Qualität und Zusammensetzung der Geräusche beschreiben.
```

Abbildung 25: Anforderungen nach dem Export

Je Anforderung wird zunächst die ID zusammen mit dem *Object Type* extrahiert und zusätzlich der Anforderungstext. Die ID und der *Object Type* werden von Schrägstrichen umfasst und durch Rauten (###) voneinander getrennt. Ist kein *Object Type* angegeben, wird das Wort *none* eingesetzt. Einzelne Anforderungen, d.h. ein oder mehrere Sätze, die in einem Anforderungsobjekt in DOORS stehen, werden hintereinander eingefügt. Beginnt ein neues Anforderungsobjekt, wird ein String *NEWLINE* vor dem neuen Anforderungsobjekt eingefügt. Darauf kann dann die nächste ID mit dem entsprechenden *Object Type* und dem entsprechenden Anforderungstext folgen. Dieses Format bildet die Grundlage für die weitere Verarbeitung der Lastenhefte.

6.3 Computerlinguistische Annotation der Lastenhefttexte

Nach dem Export der Lastenhefte wurden diese zunächst tokenisiert und anschließend mit computerlinguistischen Annotationswerkzeugen weiterverarbeitet. Die computerlinguistische Annotation beinhaltet folgende Schritte: *Lemmatisierung*, *Part-of-Speech-Tagging*, *Morphologie-Tagging* und *Parsing*. Diese einzelnen Annotationsschritte sowie die Tokenisierung werden in den Abschnitten 6.3.1 bis 6.3.5 erläutert und anhand eines durchgängigen Beispiels veranschaulicht. In Abschnitt 6.3.6 wird abschließend eine Evaluierung der verwendeten Annotationswerkzeuge präsentiert.

6.3.1 Tokenisierung

Der erste Schritt der computerlinguistischen Weiterverarbeitung der Lastenhefttexte ist die Tokenisierung und die anschließende Transformation der Anforderungen in das *CoNLL-2009-Format* [HCJ⁺09]. Die Tokenisierung ist „ein Segmentierungsverfahren, bei welchem jedes Wort eines Textes erfasst wird“ [Hal04]. Zusätzlich zu den Wörtern werden außerdem auch Zeichen erfasst. In der vorliegenden Arbeit wurde der Tokenizer aus [Kri13] verwendet, der die folgende Schritte durchführt:

1. Satzendenerkennung;
2. Segmentierung der Anforderungen in Wörter und Zeichen;
3. Transformation in das CoNLL-2009-Format.

Abbildung 26 zeigt, wie eine Anforderung nach der Tokenisierung und der Transformation in das CoNLL-2009-Format aussieht. In der ersten Spalte steht die Satznummer, die Tokennummer und die Anforderungs-ID des jeweiligen Tokens. In diesem Beispiel handelt es sich um den ersten Satz, der in dem Anforderungsobjekt mit der ID *AGS-38* steht. Dies wird durch die 1 am Anfang jeder Zeile dargestellt. Die darauffolgende Zahl gibt die Tokennummer innerhalb eines Satzes an. Beispielsweise ist *Gurtschlosses* das siebte Token in dem dargestellten Satz. Nach der ID folgt ein Tabulator, gefolgt von einem Token. Nach dem Token folgen 13 Tabulatoren mit jeweils einem nachfolgendem Unterstrich. Diese Unterstriche stellen noch nicht instantiierte Variablen dar, welche bei der nachfolgenden Annotation teilweise durch Instanzen ersetzt werden. Welches Annotationswerkzeug welche Variablen ersetzt, hängt von der Position der Variablen ab. Die Unterstriche, die durch die verwendeten Annotationswerkzeuge nicht ersetzt werden, können für die Evaluierung eigens entwickelter Annotationstools an einem Goldstandard genutzt werden. Hierfür muss der Goldstandard an die entsprechenden Stellen im CoNLL-2009-Format eingefügt werden. Diese Möglichkeit wird jedoch in dieser Arbeit nicht genutzt.

Die Transformation in das CoNLL-2009-Format, welches sich zu einem Standardformat in der Computerlinguistik entwickelt hat, wurde durchgeführt, da die verwendeten Annotationswerkzeuge dieses Format für die Weiterverarbeitung der Texte voraussetzen.

In Abbildung 27 ist die Anforderung aus Abbildung 26 zu sehen, nachdem sie alle Annotationschritte durchlaufen hat. Hieran sollen die folgenden Verarbeitungsschritte veranschaulicht werden. Die einzelnen Zwischenrepräsentationen werden nicht graphisch dargestellt. Die Unterstriche, die für eine Evaluation mit einem Goldstandard verwendet werden können und für den Aufbau der Korpora in dieser Arbeit nicht relevant sind, wurden aus der Abbildung aus Übersichtlichkeitsgründen herausgelöscht.

1_1_AGS-38	Die	-	-	-	-	-	-	-	-	-	-	-	-	-
1_2_AGS-38	zu	-	-	-	-	-	-	-	-	-	-	-	-	-
1_3_AGS-38	erreichende	-	-	-	-	-	-	-	-	-	-	-	-	-
1_4_AGS-38	Geräuschqualität	-	-	-	-	-	-	-	-	-	-	-	-	-
1_5_AGS-38	des	-	-	-	-	-	-	-	-	-	-	-	-	-
1_6_AGS-38	aktiven	-	-	-	-	-	-	-	-	-	-	-	-	-
1_7_AGS-38	Gurtschlosses	-	-	-	-	-	-	-	-	-	-	-	-	-
1_8_AGS-38	soll	-	-	-	-	-	-	-	-	-	-	-	-	-
1_9_AGS-38	auf	-	-	-	-	-	-	-	-	-	-	-	-	-
1_10_AGS-38	Basis	-	-	-	-	-	-	-	-	-	-	-	-	-
1_11_AGS-38	realer	-	-	-	-	-	-	-	-	-	-	-	-	-
1_12_AGS-38	Fahrzeugmessungen	-	-	-	-	-	-	-	-	-	-	-	-	-
1_13_AGS-38	definiert	-	-	-	-	-	-	-	-	-	-	-	-	-
1_14_AGS-38	werden	-	-	-	-	-	-	-	-	-	-	-	-	-
1_15_AGS-38	.	-	-	-	-	-	-	-	-	-	-	-	-	-

Abbildung 26: Tokenisierte Anforderung im CoNLL-2009-Format

6.3.2 Lemmatisierung

Auf die Tokenisierung folgt die Lemmatisierung. Diese wurde in dieser Arbeit mit den *mate-Tools* [Boh09] durchgeführt. Unter der Lemmatisierung versteht man „die Rückführung von Flexionsformen (Wortformen) auf einen graphematischen Repräsentanten bestimmter gemeinsamer Merkmale“ [Zim72]. Das bedeutet, dass für jedes Token das Lemma, also dessen Grundform, annotiert wird. Das flektierte Verb *soll* erhält beispielsweise das Lemma *sollen*. Das Nomen *Fahrzeugmessungen* erhält das Lemma *Fahrzeugmessung*. Für jede Kategorie von Artikeln gibt es ein Default-Lemma. Für bestimmte Artikel wird beispielsweise das Lemma *der* annotiert.

In jeder Zeile wird der zweite Unterstrich durch das entsprechende Lemma ersetzt. In Abbildung 27 entspricht dies der 3. Spalte, welche in orange dargestellt ist.

„Die Lemmatisierung bietet den Vorteil, dass bei späteren Anfragen an das Korpus nicht alle flektierten Formen eines Wortes einzeln angegeben werden müssen“ [EF04]. Sie dient also, wie auch die weiteren Annotationsverfahren, der Generalisierung und somit einer einfacheren und allgemeineren Suche nach relevanten Instanzen.

6.3.3 Part-of-Speech-Tagging

Das auf die Lemmatisierung folgende Part-of-Speech-Tagging (= POS-Tagging) bzw. Wortarten-Tagging wurde mit dem Werkzeug *MarMot* [MSS13] durchgeführt. Beim POS-Tagging wird für jedes Token dessen Wortart ermittelt und annotiert. MarMot greift

1	2	3	4	5	6	7
1_1_AGS-38	Die	der	ART	case=nom number=sg gender=fem	4	NK
1_2_AGS-38	zu	zu	PTKZU	-	3	PM
1_3_AGS-38	erreichende	erreichend	ADJA	case=nom number=sg gender=fem degree=pos	4	NK
1_4_AGS-38	Geräuschqualität	Geräuschqualität	NN	case=nom number=sg gender=fem	8	SB
1_5_AGS-38	des	der	ART	case=gen number=sg gender=neut	7	NK
1_6_AGS-38	aktiven	aktiv	ADJA	case=gen number=sg gender=neut degree=pos	7	NK
1_7_AGS-38	Gurtschlosses	Gurtschloß	NN	case=gen number=sg gender=neut	4	AG
1_8_AGS-38	soll	sollen	VMFIN	number=sg person=3 tense=pres mood=ind	0	-
1_9_AGS-38	auf	auf	APPR	-	11	MO
1_10_AGS-38	Basis	Basis	NN	case=dat number=sg gender=fem	9	NK
1_11_AGS-38	realer	real	ADJA	case=gen number=pl gender=fem degree=pos	12	NK
1_12_AGS-38	Fahrzeugmessungen	Fahrzeugmessung	NN	case=gen number=pl gender=fem	8	SB
1_13_AGS-38	definiert	definieren	VVPP	-	14	OC
1_14_AGS-38	werden	werden	VAINF	-	8	OC
1_15_AGS-38	.	--	\$.	-	14	--

Abbildung 27: Vollständig annotierte Anforderung

hierfür auf das Stuttgart-Tübingen-Tagset (= STTS) [STST99] zu, mit wenigen Modifikationen (vgl. [Smi03]). Das Stuttgart-Tübingen-Tagset ist im Anhang A der Arbeit aufgeführt. Die Annotation der Wortarten ist in Abbildung 27 in der 4. Spalte in grün dargestellt. Beispielsweise erhält das Token *soll* das Tag *VMFIN*, welches angibt, dass es sich bei *soll* um ein finites Modalverb handelt. Das Token *Fahrzeugmessungen* erhält das Tag *NN*, welches besagt, dass es sich bei dem Token um ein Nomen (common noun) handelt. Bei Satzzeichen wird als Tag eine Zeichenkombination verwendet, die mit *\$.* beginnt (vgl. Anhang A).

6.3.4 Morphologie-Tagging

Nach der Lemmatisierung folgt das Morphologie-Tagging, welches ebenfalls mit MarMot [MSS13] durchgeführt wurde. Beim Morphologie-Tagging werden grammatische Kategorien wie z.B. Genus und Numerus annotiert. Hierbei werden für die unterschiedlichen Wortarten die jeweils relevanten grammatischen (Kongruenz-)Merkmale annotiert. In Abbildung 27 (Spalte 5 in pink) ist beispielsweise zu sehen, dass bei finiten Verben der Numerus, die Person, das Tempus und der Modus annotiert werden. Bei Nomina hingegen Kasus, Numerus und Genus. Bei Adjektiven kommt außerdem noch die Komparationsstufe hinzu.

6.3.5 Parsing

Abschließend wurden die Anforderungen mit den *mate*-Tools [Boh09] geparst. Bei dem verwendeten Parser handelt es sich um einen Dependenzparser. Das heißt, dass neben

Satzfunktionen wie Subjekt, Akkusativobjekt, Relativsatz usw., auch Bezugselemente für grammatische Relationen annotiert werden. Die Bezugselemente werden durch Zahlen (in Abbildung 27 in Spalte 6 rot markiert zu sehen), die grammatischen Funktionen durch spezielle Tags dargestellt (vgl. Abbildung 27 Spalte 7 hellblau markiert).

Die grammatischen Relationen (Dependenzrelationen), bestehen jeweils zwischen dem Kopf (auch Regens genannt) und dem Dependens. Der Kopf ist das regierende Element, das Dependens das vom Kopf abhängige Element der jeweiligen binären Verbindung. Das oberste Regens eines Satzes ist das finite Verb.

Das Hauptverb eines Satzes wird mit der Zahl 0 markiert, in dem Beispiel in Abbildung 27 ist es das Token *soll*. Das Nomen *Geräuschqualität* ist ein Dependens zu dem Verb *soll* (hier mit der grammatischen Funktion Subjekt: *SB*), was durch die Zahl 8 in der sechsten Spalte markiert wird. Das Verb ist hingegen dessen Regens und gleichzeitig das absolute Regens.

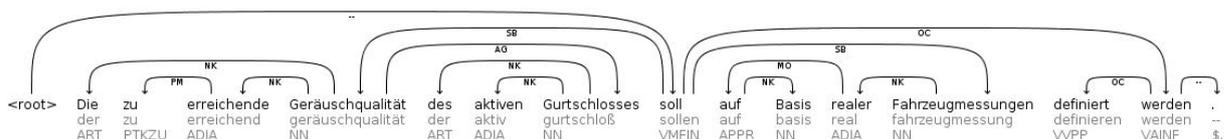


Abbildung 28: Dependenzgraph

In Abbildung 28 ist der annotierte Satz aus Abbildung 27 graphisch dargestellt. Die Abhängigkeiten zwischen den einzelnen Wörtern werden hierbei durch Pfeile dargestellt.

6.3.6 Qualitätsbetrachtung der Annotationswerkzeuge

Für die beschriebenen computerlinguistischen Annotationen wurden die Werkzeuge *mate* und *MarMot* verwendet. Diese beiden Werkzeuge liefern in Kombination die maximal automatisch erzeugbare Abstraktionsstufe von Annotationen und liefern eine gute Analysequalität. Um die Güte der Analysen von *mate* und *MarMot* zu bewerten, wurde eine Evaluierung der Werkzeuge durchgeführt. Hierfür wurden 100 annotierte Anforderungen aus den erstellten Korpora zufällig ausgewählt und alle Annotationen manuell nach ihrer Richtigkeit bewertet. Insgesamt enthalten die untersuchten 100 Anforderungen 1837 Tokens; es wurden also 9185 Attribute überprüft. Die durchschnittliche Satzlänge der untersuchten Anforderungen beträgt 18,37 Tokens.

Bei der Evaluierung der Annotationsergebnisse wurden drei Kategorien von Fehlern gezählt: *Fehler*, *interne Folgefehler* und *externe Folgefehler*. Zusätzlich gibt es die Kategorie *Mängel*. Diese vier Kategorien werden im Folgenden beschrieben.

1	2	3	4	5	6	7
1_1_OHG-38	Die	der	ART	case=nom number=sg gender=fem	2	NK
1_2_OHG-38	Komponente	Komponente	NN	case=nom number=sg gender=fem	3	SB
1_3_OHG-38	darf	dürfen	VMFIN	number=sg person=3 tense=pres mood=ind	0	_
1_4_OHG-38	kein	kein	PIAT	case=nom number=sg gender=neut	5	NK
1_5_OHG-38	Wasser	Wasser	NN	case=nom number=sg gender=neut	3	SB
1_6_OHG-38	in	in	APPR	-	9	MO
1_7_OHG-38	den	der	ART	case=acc number=sg gender=masc	8	NK
1_8_OHG-38	Innenraum	Innenraum	NE	case=acc number=sg gender=masc	6	NK
1_9_OHG-38	lassen	lassen	VVINF	-	3	OC
1_10_OHG-38	.	--	\$.	-	9	--

Abbildung 29: Arten von Fehlertypen für die Evaluierung der Annotationsergebnisse

Abbildung 29 enthält ein Beispiel für einen Mangel und Beispiele für die unterschiedlichen Arten von Fehlern.

Zunächst wird die Kategorie *Fehler* beschrieben. Ein *Fehler* liegt vor, wenn ein Annotationswerkzeug eine falsche Interpretation trifft und dieser Fehler nicht durch einen vorausgehenden Fehler entstanden ist. Dem Token *kein* in Abbildung 29 wurde fälschlicherweise der Kasus Nominativ (*case=nom*) zugewiesen. Die korrekte Annotation wäre *case=acc* für Akkusativ. Dies stellt einen Fehler dar.

Durch Fehler können Folgefehler entstehen, d.h. wenn ein Annotationswerkzeug an einer Stelle eine falsche Interpretation trifft, kann dies dazu führen, dass weitere Fehler entstehen. Diese Fortpflanzung von Fehlern fällt in die Kategorien *interne Folgefehler* und *externe Folgefehler*. Ein externer Folgefehler liegt vor, wenn einem Token eine falsche Annotation zugewiesen wird, die durch eine Fehlannotation innerhalb eines anderen Tokens entstanden ist. In Abbildung 29 ist ein externer Folgefehler in Spalte 5, Zeile 5 zu sehen. Da dem 4. Token *kein* der falsche Kasus zugewiesen wurde, erhält auch das Nomen *Wasser* den falschen Kasus. Ein interner Folgefehler liegt hingegen vor, wenn die Analyse einer Kategorie eines Tokens fehlerhaft ist, und dadurch ein weiterer Fehler im selben Token entsteht. Ein Beispiel hierfür ist in der Abbildung 29 in der 7. Spalte, Zeile 5 zu sehen. Dadurch, dass dem Token *Wasser* der Kasus Nominativ zugewiesen wurde, weist der Parser dem Nomen die Funktion Subjekt zu.

Ein Mangel liegt vor, wenn die Fehlanalyse nicht schwerwiegend ist. Wenn beispielsweise anstatt *NN* (= common noun) das Tag *NE* (= Eigenname) vergeben wurde (vgl. *Innenraum* Zeile 8 in Abbildung 29), ist dies nicht so schwerwiegend als wenn anstatt eines Nomen-Tags ein Tag für eine nicht-nominale Wortart vergeben würde.

	Fehler	Interne Folgefehler	Externe Folgefehler	Mängel	Σ
Lemma	24	0	0	2	26
Part-of-Speech	20	0	0	32	52
Morphologie	96	23	45	0	164
Dependenz	38	12	6	0	56
Gramm. Rel.	23	28	1	0	52
Σ	201	63	52	34	350

Tabelle 12: Evaluierungsergebnisse *mate* und *MarMot* (N=1837)

In Tabelle 12 sind die Ergebnisse der Evaluierung zu sehen. Für jede annotierte Kategorie werden die Fehler und Mängel gezählt. Zusätzlich werden die Summen gebildet. Insgesamt sind 201 Fehler, 63 interne Folgefehler, 52 externe Folgefehler und 34 Mängel in den 100 geprüften Anforderungen vorhanden. Zählt man die Fehler und Mängel zusammen, erhält man eine absolute Fehlerquote von 350.

Kategorie	Fehlerquote
Lemma	1,42%
Part-of-Speech	2,83%
Morphologie	8,93%
Dependenz	3,05%
Gramm. Rel.	2,83%
Gesamt	3,81%

Tabelle 13: Fehlerquoten der Annotationen (N=1837)

In Tabelle 13 sind die Fehlerquoten in Prozent pro Kategorie angegeben. Alle Fehlerquoten liegen im einstelligen Prozentbereich. Auffällig ist, dass im Bereich der Morphologie die meisten Fehler gemacht werden. Da bei der Morphologie-Annotation mehrere Attribut-Wert-Paare vergeben werden, steigt die Anfälligkeit für Fehler. Der leicht erhöhte Wert bei der Kategorie Morphologie ist somit nicht als Schwäche zu sehen. Die gesamte Fehlerquote

über alle Attribute beträgt 3,81%. Dieses Ergebnis zeigt, dass die Werkzeuge *mate* und *MarMot* für die Anwendung auf die in dieser Arbeit untersuchte Textsorte geeignet sind.

6.4 Zahlen, Daten und Fakten über die erzeugten Korpora

Die verwendeten Lastenhefte stammen aus verschiedenen Teilbereichen von Fahrzeugkomponenten: Innenraumelektronik, mechanische Komponenten, Antriebsstrang, Telematik. Tabelle 14 kann entnommen werden, wie viele Sätze und wie viele Tokens in den Korpora jeweils vorhanden sind. Die annotierten Korpora dienen als Entwicklungs- und Evaluierungskorpora. Auf den Korpora 1 und 3 werden Regeln entwickelt und auf Korpus 2 und 4 werden die entwickelten Regeln getestet. Somit kann geprüft werden, ob die entwickelten Prüfmethode nicht nur auf dem für die Regelformulierung verwendeten Textmaterial gute Ergebnisse erzielen, sondern auch auf neuem ungesehenem Textmaterial. In Tabelle 15 sind die Anzahl der Sätze und Tokens des Entwicklungskorpus, des Evaluierungskorpus und des Gesamtkorpus zusammengestellt.

Name	Gebiet	Sätze	Tokens
Korpus 1	Innenraumelektronik	76 243	852 140
Korpus 2	Mechanische Komponenten	68 480	776 321
Korpus 3	Antriebsstrang	25 692	300 252
Korpus 4	Telematik	43 135	573 507

Tabelle 14: Korpusinformationen zu den Einzelkorpora

Name	Zusammensetzung	Sätze	Tokens
Entwicklungskorpus	Korpus 1 + 3	101 935	1 152 392
Evaluierungskorpus	Korpus 2 + 4	111 615	1 349 828
Gesamtkorpus	Korpus 1 + 2 + 3 + 4	213 550	2 502 220

Tabelle 15: Korpusinformationen zu den Gesamtkorpora

6.5 Transformation der Korpora in ein CQP-lesbares Format

Für die Korpusanalysen, auf deren Basis die Regelentwicklung erfolgt, wurde das Werkzeug *The IMS Open Corpus Workbench* [EH11] verwendet. Mit dessen Abfragesprache

CQP (= Corpus Query Processor) können annotierte Korpora gezielt nach Instanzen durchsucht werden, wobei alle annotierten Attribute und Werte miteinbezogen werden können. Das Werkzeug *CQP* benötigt als Eingabe ein bestimmtes Format, welches dem CoNLL-2009-Format sehr ähnlich ist. Die einzige Änderung, die an den annotierten Korpora durchgeführt werden muss, ist, dass das Token immer in die erste Spalte versetzt wird. Um die Korpora dementsprechend anzupassen, wurde ein Programm aus [Kri13] herangezogen, das diesen Schritt automatisiert durchführt.

Die Abfragesprache *CQP* wird in Abschnitt 7.1 detailliert beschrieben.

6.6 Zusammenfassung

Im vorliegenden Kapitel wurde beschrieben, wie die Korpora erstellt wurden, die in Kapitel 8 für die geplanten Phänomenanalysen verwendet werden. Im ersten Schritt wurden Lastenhefte aus der Datenbank DOORS exportiert, anschließend tokenisiert und abschließend in das CoNLL-2009-Format transformiert. Daraufhin konnten die Anforderungstexte durch die computerlinguistischen Annotationswerkzeuge *mate* und *MarMot* annotiert werden. Die Korpora wurden um Lemmata, POS-Tags, Morphologie- und Parsinginformation angereichert. Die Annotationen dienen dazu, generelle Regeln für die Identifikation bestimmter sprachlicher Muster definieren zu können.

Zusätzlich wurde die Güte der Annotationen der beiden Werkzeuge anhand von 100 zufällig ausgewählten Anforderungen bewertet. Die Werkzeuge weisen eine sehr niedrige Fehlerquote auf. Sie eignen sich also gut für die Anwendung auf die spezielle Textsorte der Lastenhefte.

Die vier erstellten Korpora wurden zu zwei Korpora zusammengefasst: zu einem Entwicklungskorpus und einem Evaluierungskorpus. Anhand des Entwicklungskorpus werden die ausgewählten Phänomene untersucht und auf Basis dieses Korpus werden Regeln für die automatische Identifikation kritischer Instanzen entworfen. Das Evaluierungskorpus dient dazu zu überprüfen, ob die entwickelten Regeln auch auf ungesehenen Daten gute Ergebnisse liefern.

7 Allgemeines Vorgehen für die Analyse ausgewählter Phänomene

In Kapitel 8 wird detailliert auf die durchgeführten Phänomenanalysen eingegangen. Wie bei der Durchführung der Phänomenanalysen vorgegangen wird und wie Regeln für die Identifikation kritischer Vorkommen formuliert werden, wird im vorliegenden Kapitel beschrieben.

In Abbildung 30 sind die einzelnen Arbeitsschritte für den Prozess der Regelentwicklung in grün dargestellt. Für die Regelentwicklung werden die Lastenhefte zunächst aus DOORS exportiert und anschließend computerlinguistisch annotiert (vgl. Abschnitt 6.2 und 6.3). Die linguistisch annotierten Korpora werden dann in ein durch die Abfragesprache CQP (= Corpus Query Processor) verarbeitbares Format konvertiert (vgl. Abschnitt 6.5). Anhand von CQP wird in den Korpora nach den ausgewählten Phänomenen gesucht, und deren Satzkontexte werden manuell analysiert. Anschließend werden Regeln zur Identifizierung kritischer Instanzen, also Instanzen, die die Testbarkeit einer Anforderung gefährden können, abgeleitet. Die Regeln werden dann in einen Prototypen¹⁸ eingebaut, um schließlich eine Evaluierung der Regeln mittels Precision, Recall und F-Measure am Entwicklungs- und am Evaluierungskorpus durchführen zu können. Nur wenn die Evaluierung gute Ergebnisse liefert, d.h. die kritischen Instanzen zuverlässig von den unkritischen Instanzen unterschieden werden können, werden die entwickelten Regeln in ein Daimler-internes Werkzeug eingebaut (vgl. Abschnitt 9.1). Die Tauglichkeit für die Praxis wird somit sichergestellt.

In Abschnitt 7.1 wird die Vorgehensweise der Regelentwicklung anhand von CQP detailliert beschrieben. In Abschnitt 7.2 werden anschließend die verwendeten Evaluierungsmethoden aufgezeigt und anhand von Beispielen erklärt. Die Umsetzung in einem Daimler-internen Werkzeug, welches im Folgenden *Lastenheft-Analyseassistent* genannt wird, wird in Abschnitt 9.1 vorgestellt.

Der Zielprozess bei der Prüfung von Anforderungen soll der Folgende sein: Das zu analysierende Lastenheft wird exportiert, computerlinguistisch annotiert und in ein CQP-lesbares Format konvertiert. Die entwickelten CQP-Regeln werden dann auf die annotierten Anforderungen angewandt und die Ergebnisse der Analyse werden dem Benutzer im Lastenheft-Analyseassistent aufgezeigt. Anhand der Ergebnisse kann der Benutzer dann entscheiden, ob die gefundenen potentiell kritischen Konstruktionen geändert werden müssen. In Abbildung 30 ist der Zielprozess blau hinterlegt.

¹⁸Der Prototyp entspricht einer vereinfachten Version des Lastenheft-Analyseassistenten, der für den Zielprozess verwendet wird (vgl. Abschnitt 9.1).

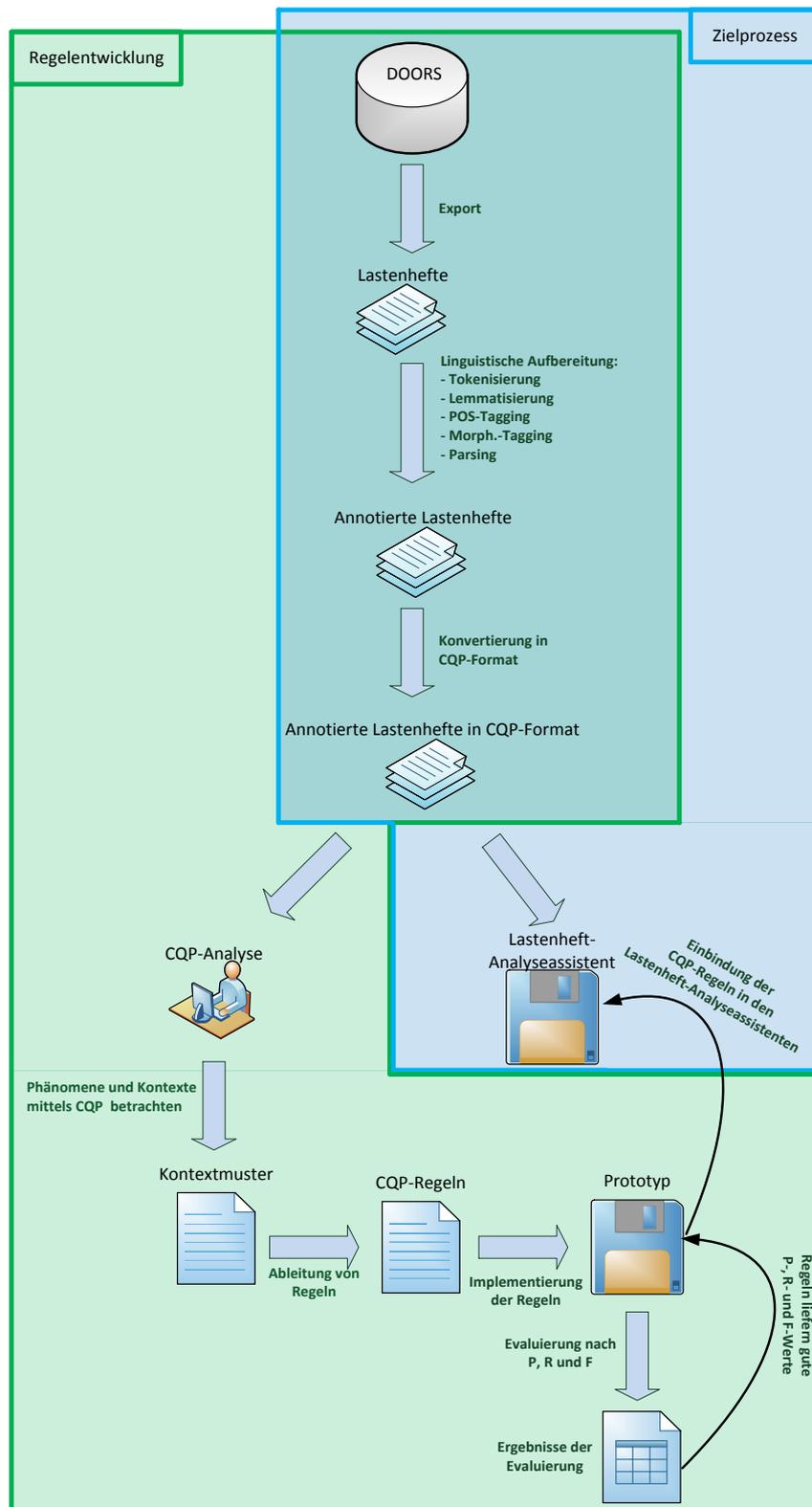


Abbildung 30: Vorgehen bei der Entwicklung und Umsetzung von Regeln: Regelentwicklung (grün) und Zielprozess (blau)

Im Folgenden werden nun die Schritte der Regelentwicklung und der Evaluierung ausführlich beschrieben.

7.1 Regelfindung und Regeldefinition

Um Regeln für die automatische Identifikation kritischer Instanzen entwickeln zu können, werden mittels CQP zunächst die Textstellen im Korpus ermittelt, die potentiell ein bestimmtes Phänomen enthalten könnten. Bei der Analyse des Modalverbs *sollte* werden beispielsweise zunächst die Satzinstanzen gesucht, welche das Wort *sollte(n)* enthalten. Dieses Verfahren ist möglich für Phänomene, die lexikalisch identifizierbar sind. Für die Identifikation von Phänomen, welche nicht durch lexikalische Abgleiche identifiziert werden können, werden die annotierten Informationen der Korpora verwendet. Beispielsweise kann das Phänomen *Passiv ohne Agens* durch Abfolgen von Wörtern, welche bestimmten Wortarten zugeordnet werden können, identifiziert werden.

Die gefundenen Satzinstanzen werden anschließend manuell analysiert und danach bewertet, ob es sich um kritische oder unkritische Instanzen handelt. Die Kriterien für diese Klassifikation werden in Kapitel 8 für jedes Phänomen je einzeln diskutiert. Die kritischen Instanzen lassen sich für bestimmte Phänomene in zwei Kategorien einteilen: *Defect* und *Effect*. Kritische Instanzen, die in die Kategorie *Defect* fallen, entsprechen Instanzen, die bei einer automatischen Analyse dem Autor zurückgeliefert werden müssen, mit dem Hinweis, dass die Textstelle überarbeitet werden muss, da entweder Information fehlt oder eine Mehrdeutigkeit vorliegt. In die Kategorie *Effect* fallen hingegen die Instanzen, bei welchen nicht mit Sicherheit eine Aussage getroffen werden kann, ob Information fehlt bzw. ob eine mehrdeutige Struktur vorliegt. Dies ist der Fall, wenn beispielsweise mit Verweisen gearbeitet wird. In Beispiel 63 ist ein solcher Fall abgebildet.

Beispiel 63 *Ein langer Tastendruck (vgl. Anforderung 123) führt dazu, dass die Funktion abbricht.*

Wenn *Anforderung 123* existiert und dort die Länge des Tastendrucks beschrieben wird, muss die Anforderung nicht geändert werden. Ist dies nicht der Fall, muss die Anforderung geändert werden. Die Kategorie *Effect* dient folglich dazu, dem Autor den Hinweis zu geben zu überprüfen, ob die Information, auf die verwiesen wird, auch wirklich im Anforderungsdokument oder einem beigefügten Dokument genannt wird.

Es gibt folglich drei Kategorien, in welche bestimmte Kontextmuster eingeordnet werden können: *Ok*, *Defect* und *Effect*.

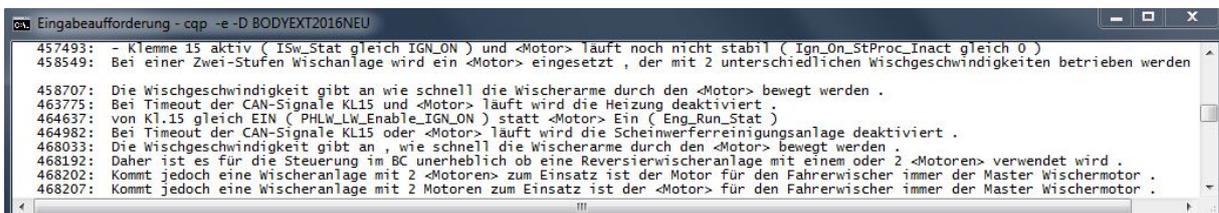
Der nächste Schritt ist, die identifizierten Instanzen in weitere Gruppen zu unterteilen. Die Elemente einer Gruppe folgen hierbei jeweils demselben syntaktischen oder morpho-

logischen Muster, sodass angenommen werden kann, dass sie durch je eine gemeinsame Regel erfasst werden können. Nach dieser Einteilung in Gruppen wird untersucht, wie diese syntaktischen und morphologischen Regelmäßigkeiten innerhalb der Gruppen mittels Regeln identifiziert werden können. Hierbei werden alle Informationen verwendet, welche durch die computerlinguistischen Werkzeuge *mate* und *MarMot* annotiert wurden. Es werden die Lemmata, die POS-Tags, die morphologischen Informationen, die Satzfunktionen, die Abhängigkeiten und die Wortnummern, d.h. Information zur Wort- bzw. Konstituentenreihenfolge, in die Analyse miteinbezogen (vgl. Abschnitt 6.3). Hierfür wird das Werkzeug CWB mit der Abfragesprache CQP verwendet, mit welchem man auf alle im Korpus repräsentierten Attribute und ihre Werte zugreifen kann.

Annotierte Information	Variable in CQP
Wort	word
Lemma	lemma
Wortart	pos
Grammatische Kategorien	morph
Abhängigkeit	dep
Satzfunktion	parse
Wortnummer	wordnr

Tabelle 16: Variablen in CQP

In Tabelle 16 ist zu sehen, mittels welcher Variablen welche Informationen in den annotierten Korpora durch CQP abgerufen werden können. CQP arbeitet mit einer speziellen Syntax, welche auf regulären Ausdrücken basiert. Man kann u.a. nach Lemmata suchen, um alle Okkurrenzen eines Nomens zu identifizieren. Wenn man beispielsweise alle Sätze angezeigt bekommen möchte, die das Nomen *Motor* oder eine Wortform von *Motor* enthalten, kann man folgende Query verwenden: `[lemma="Motor"]`;

Abbildung 31: Ergebnis in CQP zu Query `[lemma="Motor"]`;

In Abbildung 31 ist das Ergebnis von CQP in der Eingabeaufforderung nach Eingabe der Query `[lemma="Motor"]`; zu sehen. Die Sätze, welche das Wort *Motor* oder eine Wortform dieses Wortes enthalten, werden dem Benutzer aufgezeigt. Es ist zu beachten, dass in Abbildung 31 nur eine Untermenge des Gesamtergebnisses zu sehen ist.

Mittels CQP können nicht nur einfache, sondern auch komplexere Querys definiert werden. Möchte man beispielsweise alle Nomina, die auf *-ung* enden, angezeigt bekommen, kann man dies mittels einer komplexeren Query angeben: `[pos="NN" & lemma=".+ung"]`; Die Suche ist nicht nur auf Einzelwörter beschränkt, es kann auch nach bestimmten Wortfolgen gesucht werden. Eine Query, welche Satzteile identifiziert, die einen Artikel mit nachfolgendem Adjektiv und darauffolgendem Nomen enthalten, sieht folgendermaßen aus: `[pos="ART"] [pos="ADJA"] [pos="NN"]`;

Mit CQP können sehr komplexe Regeln definiert und abgefragt werden. Dies ermöglichen reguläre Ausdrücke, welche nicht nur die Tokens, sondern auch die annotierten Eigenschaften miteinbeziehen können. Die Mächtigkeit von regulären Ausdrücken ist in CQP verankert, d.h. CQP implementiert reguläre Ausdrücke im selben Umfang wie beispielsweise die Programmiersprache Perl. Es ist außerdem möglich durch Einbezug der Annotationen generelle Regeln zu formulieren, was die Arbeit mit Wortlisten erheblich reduziert. Bei den Phänomenanalysen in Kapitel 8 werden die Regeln in CQP-Code angegeben. Diese Regeln werden nach der Evaluierung in ein Prüfwerkzeug eingebaut, welches in Abschnitt 9.1 erläutert wird. Nach der Entwicklung der Regeln, müssen diese auf ihre Tauglichkeit in der Praxis evaluiert werden. Das Vorgehen bei der Evaluierung wird in Abschnitt 7.2 erläutert.

7.2 Evaluierung

Die in Kapitel 8 beschriebenen Regeln wurden anhand der Texte des Entwicklungskorpus entwickelt (vgl. Abschnitt 6.4). Um eine Aussage darüber treffen zu können, ob die entwickelten Regeln gute Ergebnisse liefern, wird jeweils eine Evaluierung am Entwicklungskorpus und nachfolgend eine zweite Evaluierung am Evaluierungskorpus durchgeführt. Bei den Texten aus dem Evaluierungskorpus handelt es sich um ungesehene Daten. Anhand der Evaluierung mithilfe des Evaluierungskorpus kann festgestellt werden, ob die Regeln nur für das Korpus gültig sind, an dem sie entwickelt wurden, oder ob sie im Rahmen der benutzten Textsorten (= deutschsprachige Komponenten- und Systemlastenhefte) auch für ungesehene Texte gültig sind.

Um die Qualität der Regeln bewerten zu können, wurden die Ergebnisse, welche die Regeln automatisiert liefern, manuell nach ihrer Richtigkeit bewertet (= Goldstandard). Somit können die Anzahl der Fehlwarnungen und der Fehlentwarnungen festgestellt werden. Die Evaluierungsmaße, die in dieser Arbeit angewandt werden, sind Precision (Präzision), Recall (Vollständigkeit) und F-Measure (F-Maß).

„Mit Präzision [...] bezeichnet man den Anteil der Dokumente am Suchergebnis, die tatsächlich relevant sind in Bezug auf die Gesamtgröße des Ergebnisses“ [DGS04]. Bei der

in dieser Arbeit durchgeführten Evaluierung beziehen sich die Ergebnisse nicht auf gefundene und nicht gefundene Dokumente, sondern auf gefundene bzw. nicht gefundene Satzinstanzen. Bei der Berechnung der Präzision der Klasse *Kritische Anforderung* fallen folglich die Fehlwarnungen ins Gewicht, bei der Klasse *Unkritische Anforderung* die Fehlentwarnungen. Die Formel für die Precision setzt sich folgendermaßen zusammen:

$$Precision = \frac{tp}{tp + fp} \quad (1)$$

Die Abkürzung *tp* in Formel 1 steht für *true positives*, d.h. für die Satzinstanzen die durch die Analysen korrekterweise gefunden werden. *fp* steht für *false positives*. In diese Variable werden die Instanzen eingerechnet, die durch die Analysen fälschlicherweise gefunden werden. Dies sind bei der Klasse *Kritische Anforderung* die Fehlwarnungen und bei der Klasse *Unkritische Anforderung* die Fehlentwarnungen.

„Die Vollständigkeit [...] ist der Anteil der relevanten Dokumente im Suchergebnis in Bezug auf die Menge aller für diese Suche relevanten Dokumente“ [DGS04]. Auch bei der Berechnung des Recall werden anstatt von Dokumenten Satzinstanzen betrachtet. Hierbei fallen bei der Klasse *Kritische Anforderung* die Fehlentwarnungen ins Gewicht, also Instanzen, die dem Autor hätten aufgezeigt werden müssen, aber nicht wurden, bei der Klasse *Unkritische Anforderung* die Fehlwarnungen. Die Formel für den Recall setzt sich folgendermaßen zusammen:

$$Recall = \frac{tp}{tp + fn} \quad (2)$$

Auch in Formel 2 werden relevante gefundene Satzinstanzen in *tp* (true positives) eingerechnet. Bei *fn* (false negatives) handelt es sich um die Satzinstanzen, welche durch die Analysen fälschlicherweise nicht gefunden wurden; dies entspricht bei der Klasse *Kritische Anforderung* den Fehlentwarnungen und bei der Klasse *Unkritische Anforderung* den Fehlwarnungen.

Die Formel für die Berechnung von Precision und Recall liefert Werte, die zwischen 0 und 1 liegen. Jedoch werden die Maße typischerweise mit einer Skala von 0 bis 100 in Prozent angegeben [MRS09].

Das F-Maß ist eine Kombination aus Precision und Recall und gibt einen Gesamtüberblick über die Performanz des Werkzeugs in Bezug auf die einzelnen Klassen. Das F-Maß bildet das gewichtete harmonische Mittel [MRS09] und lässt sich durch Formel 3 berechnen.

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3}$$

Das harmonische Mittel wird dem arithmetischen Mittel vorgezogen, da das harmonische Mittel nur einen hohen Wert zurückliefert, wenn Precision und Recall beide in einem hohen Bereich liegen. Wenn es beispielsweise das Ziel ist alle kritischen Instanzen von *lang* zu identifizieren und für die Analyse ein reiner Wortlistenabgleich durchgeführt wird, liegt der Recall bei der Klasse *Kritische Anforderung* bei 100%, weil alle kritischen Instanzen gefunden werden. Die Precision liegt hingegen in einem vergleichsweise niedrigen Bereich. Wenn die Precision beispielsweise bei 10% liegt und der Recall bei 100%, liefert das arithmetische Mittel einen Wert von 55%. Das harmonische Mittel liefert hingegen einen weitaus niedrigeren Wert von 18,18%.

Die Evaluierungsergebnisse werden in Kapitel 8 wie in Abbildung 32 und Tabelle 17 illustriert für jedes Phänomen angegeben.

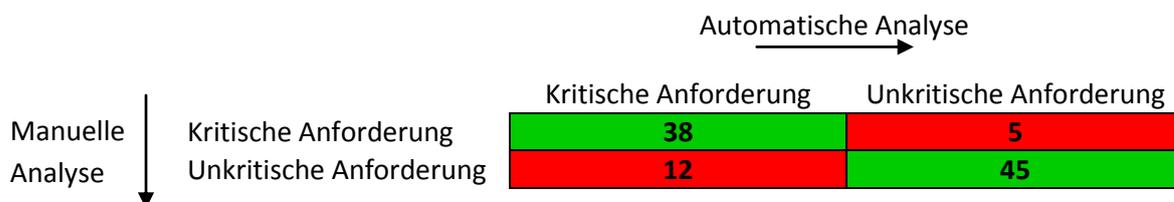


Abbildung 32: Beispiel einer Evaluierung, N=100

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	76%	90%
Recall	88,37%	78,95%
F-Measure	81,72%	84,11%

Tabelle 17: Beispielergebnis Precision-, Recall- und F-Measure, N=100

Die Phänomenanalysen werden automatisiert durchgeführt und anschließend manuell überprüft. In Abbildung 32 sind die Ergebnisse der automatischen und der manuellen Analyse zu sehen. Insgesamt wurden 100 Anforderungen betrachtet. 43 Anforderungen sind kritisch (= 38+5) und 57 Anforderungen unkritisch im Hinblick auf die Testbarkeit (= 45+12). Abbildung 32 kann entnommen werden, wie viele kritische Instanzen korrekt von der automatischen Analyse als kritisch bewertet wurden. Dies sind 38 Instanzen. Weiterhin kann man die korrekt als unkritisch bewerteten Instanzen sehen. In diesem Fall sind dies 45. Man kann der Graphik außerdem die Anzahl der Fehlwarnungen und der Fehlentwarnungen entnehmen. Fehlwarnungen sind Warnungen, die fälschlicherweise an den Benutzer ausgegeben werden. Fehlentwarnungen hingegen sind Warnungen, die

dem Autor fälschlicherweise nicht aufgezeigt werden. In Abbildung 32 entspricht dies 12 Fehlwarnungen und 5 Fehlentwarnungen. Auf diese Weise werden die Ergebnisse bei der Evaluierung der Phänomenanalysen dargestellt. Die Precision-, Recall- und F-Measure-Werte werden in Tabellen dargestellt wie in Tabelle 17. Die Werte werden je einzeln für die beiden Klassen *Kritische Anforderung* und *Unkritische Anforderung* ermittelt. Die Belegung der jeweiligen Werte sind in den Abbildungen 33 und 34 zu sehen.

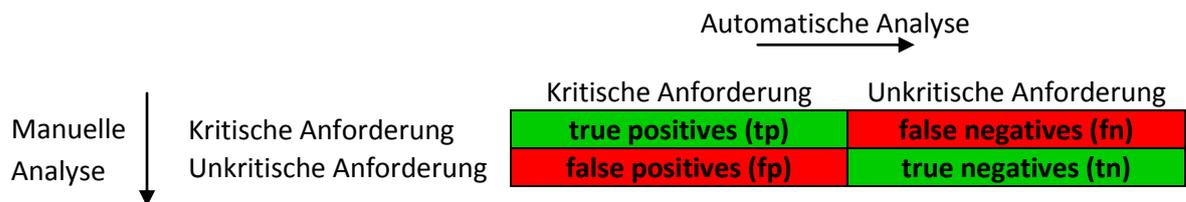


Abbildung 33: Einteilung tp , fp , fn und tn in der Klasse *Kritische Anforderung*

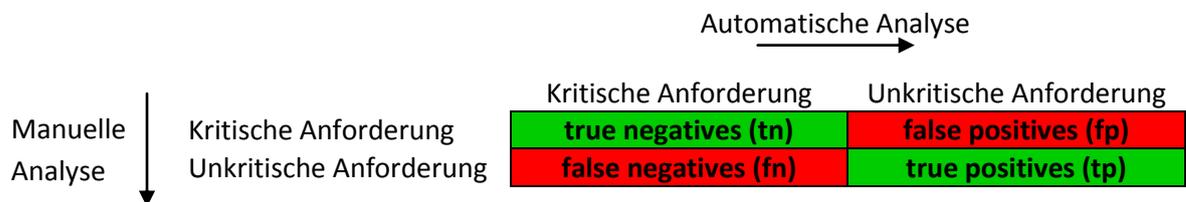


Abbildung 34: Einteilung tp , fp , fn und tn in der Klasse *Unkritische Anforderung*

Die Klasse *Kritische Anforderung* hat einen Precision-Wert von 76% (vgl. Berechnung 4). Hierbei fallen die zwölf Fehlwarnungen ins Gewicht. Der Recall-Wert beträgt 88,37% (vgl. Berechnung 5). Hier fallen die Fehlentwarnungen ins Gewicht.

$$Precision (Kritische Anforderung) = \frac{38}{38 + 12} = 0,76 \quad (4)$$

$$Recall (Kritische Anforderung) = \frac{38}{38 + 5} = 0,8837 \quad (5)$$

Bei der Klasse *Unkritische Anforderung* ist es der umgekehrte Fall: Bei der Berechnung des Precision-Wertes fallen die Fehlentwarnungen ins Gewicht, bei der Berechnung des Recall-Wertes hingegen die Fehlwarnungen. Der Precision-Wert der Klasse *Unkritische Anforderung* beträgt 90% (vgl. Berechnung 6) und der Recall-Wert dieser Klasse beträgt 78,95% (vgl. Berechnung 7).

$$Precision (Unkritische Anforderung) = \frac{45}{45 + 5} = 0,90 \quad (6)$$

$$\text{Recall (Unkritische Anforderung)} = \frac{45}{45 + 12} = 0,7895 \quad (7)$$

Das gewichtete harmonische Mittel (F-Maß) beträgt 81,72% und 84,11% (vgl. Berechnung 8 und 9).

$$F \text{ (Kritische Anforderung)} = 2 * \frac{0,76 * 0,8837}{0,76 + 0,8837} = 0,8172 \quad (8)$$

$$F \text{ (Unkritische Anforderung)} = 2 * \frac{0,9 * 0,7895}{0,9 + 0,7895} = 0,8411 \quad (9)$$

Auf diese Weise werden die Analyseergebnisse in Kapitel 8 jeweils berechnet und die Güte der entwickelten Regeln evaluiert.

In der vorliegenden Arbeit wird eine tokenbasierte Evaluierung durchgeführt. Das bedeutet, dass jeder relevante Satz, auch wenn dieser mehrfach identisch in den Korpora auftritt, in die Bewertung eingerechnet wird. Bei einer typebasierten Evaluierung werden die mehrfach vorkommenden Sätze lediglich einmal bei der Berechnung von Precision, Recall und F-Maß betrachtet. Kommt ein Satz beispielsweise identisch fünfmal in den Korpora vor, würde er bei der typebasierten Evaluierung nur einmal eingerechnet werden. Bei der tokenbasierten Evaluierung wird der Satz hingegen fünfmal eingerechnet. In dieser Arbeit wird die tokenbasierte der typebasierten Evaluierung vorgezogen, da jede Instanz, die kritisch ist, an den Autor zurückgeliefert werden muss, unabhängig davon, ob sie einfach oder mehrfach im zu analysierenden Text auftritt.

Das Ziel der Analysen ist hohe Precision- und Recall-Werte anzustreben. Dies bedeutet, dass bei der Analyse so wenig Fehlwarnungen und Fehlentwarnungen wie möglich entstehen sollen. Zu viele Fehlwarnungen können dazu führen, dass die Akzeptanz des Werkzeugs bei den Benutzern der Analysen sinkt. Ist es für den Benutzer ein zu hoher Mehraufwand die wirklich relevanten Warnungen herauszusuchen, wird er das Werkzeug für zukünftige Analysen vermutlich nicht wieder benutzen. Fehlentwarnungen haben zwar keinen Einfluss auf die Benutzbarkeit des Werkzeugs, jedoch führen sie dazu, dass kritische Instanzen übersehen und nicht überarbeitet werden. Die Anzahl der Fehlwarnungen und der Fehlentwarnungen muss bei einer automatischen Analyse folglich so niedrig wie möglich gehalten werden.

In Abschnitt 3.2 wurde bereits erwähnt, dass bei der Textsorte der Lastenhefte eine spezielle Kommunikationssituation vorliegt: Experten schreiben für Experten. Dies stellt eine andere Kommunikationssituation dar als bei Texten aus dem Bereich der technischen

Kommunikation, bei welchen Experten für Laien schreiben. Mehrdeutigkeiten, welche aufgrund von technischem Fachwissen aufgelöst werden können, sind somit im Kontext der Lastenheftkommunikation nicht als kritisch zu sehen. Bei der Evaluierung der Regeln für die ausgewählten Phänomene wird daher nicht nur betrachtet, ob bestimmte Satzkontexte, in welchen die Phänomene auftreten, mehrdeutig sind, sondern es wird diese besondere Kommunikationssituation bei der Bewertung der Analysen miteinbezogen. Auch der Einsatz von Weltwissen spielt bei der Evaluierung eine wichtige Rolle. Eine morphosyntaktische Mehrdeutigkeit bedeutet nicht immer gleich, dass auch eine semantische Mehrdeutigkeit vorliegt. Des Weiteren ist zu beachten, dass die zu bewertenden Instanzen immer im Kontext betrachtet werden. Das bedeutet, dass nicht nur die Teile des Satzes, welche durch die Regeln identifiziert werden, bewertet werden, sondern, dass zusätzlich überprüft wird, ob im unmittelbaren Satzkontext oder sogar im Gesamtdokument Information enthalten ist, mit welcher die potentielle Mehrdeutigkeit aufgelöst werden kann, oder ob die vermeintlich fehlende Information an einer anderen Stelle genannt wird.

7.3 Zusammenfassung

Es wurde gezeigt welche Methoden für die Regelfindung und Evaluierung verwendet werden. In den annotierten Lastenheften werden zunächst mit der Abfragesprache CQP relevante Textstellen, welche ein bestimmtes sprachliches Phänomen enthalten, identifiziert. Anhand der Ergebnisse werden dann Regeln für die automatische Identifikation kritischer und unkritischer Instanzen aufgestellt. Bei den unkritischen Instanzen wird zwischen zwei Typen unterschieden: *Defect* und *Effect*. Die Regelentwicklung wird am Entwicklungskorpus durchgeführt. Anschließend werden die entworfenen Regeln in einem Prototypen für die Evaluierung implementiert. Die Evaluierung wird zunächst am Entwicklungskorpus und anschließend am Evaluierungskorpus durchgeführt, um zu überprüfen, ob die Regeln nicht nur auf bekannten, sondern auch auf ungesehenen Daten gute Ergebnisse liefern und somit für einen produktiven Einsatz geeignet sind. Für die Evaluierung werden die Evaluierungsmaße Precision, Recall und das harmonische Mittel F-Maß verwendet.

Wenn die Evaluierung gute Ergebnisse liefert, d.h. hohe Precision- und Recall-Werte, werden die entwickelten Prüfungen in ein Daimler-internes Werkzeug eingebaut und stehen für den produktiven Einsatz zur Verfügung. Die praktische Umsetzung in diesem Werkzeug wird in Abschnitt 9.1 beschrieben.

8 Phänomenanalysen

Im vorliegenden Kapitel werden die Phänomene, welche in Abschnitt 4.8 als relevant identifiziert wurden, analysiert. Jedes Phänomen wird zunächst aus sprachwissenschaftlicher Sicht beschrieben und es wird gezeigt, welche Probleme auftreten können, wenn ein bestimmtes Phänomen in einer Anforderung auftritt. In Vorbereitung auf den nächsten Schritt werden zusätzlich zu der allgemeinen Beschreibung der Phänomene auch Kontexte genannt, in welchen das Auftreten eines bestimmten Phänomens einen negativen Einfluss auf die Testbarkeit einer Anforderung haben kann. Im nächsten Schritt werden, soweit möglich, Regeln präsentiert, mit welchen die kritischen Instanzen der entsprechenden Phänomene automatisiert identifiziert werden können. Die Regeln werden in einem anschließenden Schritt evaluiert, d.h. danach bewertet, ob sie auf ungesehenem Textmaterial ebenso gute Ergebnisse liefern wie auf dem Textmaterial, auf dem sie entwickelt wurden. Hiermit soll gezeigt werden, ob ein Einsatz in der Praxis sinnvoll ist. Nicht für jedes Phänomen wurde eine umfassende Evaluierung durchgeführt, da bei bestimmten Phänomenen bei der Analyse schnell ersichtlich wurde, dass durch Regeln zwar die mehrdeutigen Instanzen identifiziert werden können, jedoch durch Welt- und Expertenwissen die potentiell mehrdeutigen Stellen in den meisten Fällen aufgelöst werden können oder die potentiell fehlende Information hergeleitet werden kann.

Die Phänomene, die in Abschnitt 4.8 als relevant identifiziert wurden und in diesem Kapitel ausführlich analysiert werden, sind die Folgenden: *Weak-Word*, *Unpersönliche Konstruktion*, *Passiv und Passiversatzkonstruktionen ohne Agens*, *Modalverb „sollte(n)“* und *Pronominaler Bezug*.

8.1 Weak-Words

Das Phänomen *Weak-Word* ist im Qualitätsmodell der Kategorie *Fehlende Information* zugeordnet. Durch die Verwendung von Dimensionsadjektiven wie beispielsweise *lange* oder *schnell* ohne Angabe eines messbaren Wertes oder eines Vergleichs, kann für den Leser Interpretationsspielraum entstehen. Durch die Verwendung von Grenzadjektiven wie *tatsächlich* oder *eigentlich* kann auch Interpretationsspielraum entstehen. Zwar nicht durch Weglassen einer quantifizierenden Angabe, aber durch das Fehlen von weiterer für das einheitliche Verständnis wichtiger Information.

In diesem Abschnitt soll zunächst auf die Einteilung der Weak-Words in die Adjektivklassen, welche von [Par97] und [Bie89] definiert sind, eingegangen werden. Weiterhin wird beschrieben, inwiefern die Kritikalität von Weak-Words von dem Satzkontext, in welchem

sie stehen, abhängt. Anschließend werden Regeln für eine automatische Identifikation kritischer Instanzen gezeigt und die Evaluierungsergebnisse präsentiert¹⁹.

8.1.1 Phänomenbeschreibung

Weak-Words sind „Wörter oder Phrasen, deren Benutzung in einem Freitext darauf schließen lässt, dass der Freitext mit hoher Wahrscheinlichkeit unpräzise ist“ [Mel00]. Der Begriff *Weak-Word* ist ein feststehender Begriff aus dem Bereich des Requirements Engineering. Ein Beispiel für ein Weak-Word ist das Adjektiv *lang*. Durch die Verwendung eines solchen Weak-Words ohne eine quantifizierende Angabe kann Interpretationsspielraum entstehen. Dies birgt die Gefahr, dass im schlimmsten Fall kein einheitliches Verständnis zwischen Autor und Leser besteht und somit die Ableitung des korrekten Testfalls nicht gewährleistet ist.

In die Klasse der Weak-Words fallen aus sprachwissenschaftlicher Sicht Adverbien und Adjektive, die bestimmten Klassen zugeordnet sind.²⁰ Für die Klassifizierung in die verschiedenen Adjektivklassen wird die Kategorisierung von Paradis [Par97] in Kombination mit der Kategorisierung von Bierwisch [Bie89] verwendet.

In [Par97] werden Adjektive in drei Klassen unterteilt: skalierbare Adjektive (*schnell*, *lang*), extreme Adjektive (*brilliant*, *desaströs*) und Grenzadjektive (*wahr*, *möglich*) (vgl. hierzu auch Abschnitt 4.2.1.2).

Paradis führt für die Unterscheidung der Klassen die folgenden vier Kriterien ein:

- Kriterium 1: Das Adjektiv ist steigerbar.
- Kriterium 2: Folgende Konstruktion kann gebildet werden: *Wie X ist es?*
- Kriterium 3: Das Adjektiv kann in einem Exklamativsatz verwendet werden.
- Kriterium 4: Das Adjektiv kann in der semantischen Relation der Antonymie zu einem anderen Adjektiv stehen.

Skalierbare Adjektive erfüllen alle vier Kriterien, extreme Adjektive hingegen erfüllen nur die Kriterien 3 und 4 und Grenzadjektive erfüllen lediglich das 4. Kriterium.

Eine weitere Unterteilung der skalierbaren Adjektive in zwei Untergruppen nimmt Bierwisch [Bie89] vor. Skalierbare Adjektive können in die Klasse der Dimensionsadjektive (*lang*, *schnell*) oder in die Klasse der evaluativen Adjektive (*gut*, *schön*) fallen.

¹⁹Die Arbeitsweise zur Analyse von Weak-Words wurde u.a. in [KDJH16], [Kri14b] und [Kri14a] publiziert.

²⁰Nominalisierte Weak-Words werden dem Phänomen *Nominalisierung* im Qualitätsmodell zugeordnet und werden deshalb im vorliegenden Abschnitt nicht diskutiert.

Dimensionsadjektive	Grenzadjektive	Adverbien
<i>lang</i>	<i>maximal</i>	<i>mal</i>
<i>kurz</i>	<i>minimal</i>	
<i>schnell</i>	<i>eigentlich</i>	
<i>breit</i>	<i>tatsächlich</i>	
<i>hoch</i>		
<i>niedrig</i>		
<i>groß</i>		
<i>klein</i>		

Tabelle 18: Einteilung der untersuchten Weak-Words in die Klassen der Adjektive und Adverbien

Die Weak-Words, die in dieser Arbeit behandelt werden, fallen in die Klassen der Dimensionsadjektive und der Grenzadjektive. Zusätzlich zu den Adjektiven wird auch ein Adverb (*mal*) analysiert und die von den ausgewählten Adjektiven abgeleiteten Adverbien (z.B. *lange*) werden ebenfalls behandelt. In Tabelle 18 ist die Einteilung der behandelten Weak-Words in die Adjektivklassen und die Adverbklasse zu sehen. In der Literatur finden sich weitere Beispiele für Weak-Words, die potentiell behandelt werden könnten. Die in dieser Arbeit behandelten Weak-Words stellen eine Untermenge dar, welche nach den Kriterien Häufigkeit und Kritikalität im Korpus ausgewählt wurden. Die Herangehensweise, die im Folgenden präsentiert wird, ist übertragbar auf weitere Weak-Word-Kandidaten. Da der Zeitaufwand für die Entwicklung und Evaluierung der Regeln sehr hoch ist, beschränkt sich diese Arbeit auf die in Tabelle 18 präsentierten Weak-Words.

Weak-Words in Anforderungen erhöhen die Wahrscheinlichkeit, dass für das Verständnis wichtige Information für den Leser nicht explizit gemacht wird, d.h. nicht textuell repräsentiert wird. In Beispiel 64 ist ein Fall zu sehen, in welchem das Weak-Word *lange* aufgrund einer fehlenden quantifizierenden Angabe Interpretationsspielraum auslöst. Entweder muss der Testentwickler bzw. der Lieferant bei der Ableitung eines Testfalls aus Erfahrung einen messbaren Wert wählen, einen Wert schätzen oder den Autor nach der fehlenden Information fragen (was nicht immer möglich ist und zusätzliche Zeit kostet). Fälle wie in Beispiel 64, bei welchen für den Leser relevante Information weggelassen wird und dadurch Interpretationsspielraum entsteht, sind im Hinblick auf die Testbarkeit einer Anforderung kritisch.

Beispiel 64 *Das Signal muss lange gesendet werden.*

Jedoch ist die Verwendung von Weak-Words in Anforderungen nicht grundsätzlich als kritisch anzusehen, da sie nicht immer die Vollständigkeit von Aussagen und somit auch nicht grundsätzlich die Testbarkeit negativ beeinflussen (vgl. Beispiel 65).

Beispiel 65 *Das Signal muss 5 Sekunden lang gesendet werden.*

In der Anforderung aus Beispiel 65 wurde *lang* um die quantifizierende Angabe *5 Sekunden* erweitert. In diesem Fall eröffnet das Weak-Word *lang* keinen Interpretationsspielraum. Die Ableitung eines Testfalls kann daher problemlos vorgenommen werden.

Ein weiteres Beispiel für ein Dimensionsadjektiv, das ein Weak-Word sein kann, ist *groß*. Auch bei diesem Adjektiv führt das Weglassen einer quantifizierenden Angabe dazu, dass Interpretationsspielraum entsteht. Ein Beispiel für eine kritische Verwendung von *groß* ist in Beispiel 66 angegeben.

Beispiel 66 *Die sichtbare Lichtleiste ist groß.*

Beispiel 67 *Die sichtbare Lichtleiste ist 410 x 20 mm groß.*

In Beispiel 67 ist die Anforderung aus Beispiel 66 um eine quantifizierende Angabe (*410 x 20 mm*) erweitert. Diese Verwendung von *groß* verursacht keinen Interpretationsspielraum und hat somit keinen Einfluss auf die Testbarkeit.

Die Weak-Words in den bisher diskutierten Beispielen fallen in die Klasse der Dimensionsadjektive. Eine Anforderung, die ein Grenzadjektiv enthält und ebenfalls Interpretationsspielraum aufgrund fehlender Information auslöst, ist in Beispiel 68 zu sehen.

Beispiel 68 *Ein Überhitzen aller im Fahrzeug verbauten Panels über eine maximale Zeit ist zu vermeiden.*

Auch in diesem Fall kann durch Hinzufügen einer quantifizierenden Angabe der Interpretationsspielraum reduziert werden, obwohl das Weak-Word weiterhin verwendet wird (vgl. Beispiel 69).

Beispiel 69 *Ein Überhitzen aller im Fahrzeug verbauten Panels über eine maximale Zeit von 15 Sekunden ist zu vermeiden.*

Die Kritikalität eines Weak-Words hängt folglich stark vom Satzkontext ab, in welchem es verwendet wird. Auch Expertenwissen kann dabei helfen, den durch fehlende Information entstehenden Interpretationsspielraum gering zu halten. Der Ansatz gängiger Textanalysewerkzeuge ist, unabhängig vom Satzkontext, durch reine Wortlistenabgleiche Weak-Words zu identifizieren. Dies bedeutet, dass bei jeder Verwendung eines Weak-Words das System eine Warnung ausgibt. Da nicht jedes Vorkommen eines Weak-Words Interpretationsspielraum auslöst, führt ein reiner Wortlistenabgleich dazu, dass viele Fehlwarnungen produziert werden. Ein Wortlistenabgleich reicht für eine effiziente Analyse folglich nicht aus. Der Satzkontext, in welchem die Weak-Words stehen, muss in die Analyse miteinbe-

zogen werden, um die Anzahl der Fehlwarnungen so niedrig wie möglich halten zu können und dem Autor möglichst nur die kritischen Instanzen zurückzumelden.

Regeln, mit welchen eine kontextbasierte Weak-Word-Analyse durchgeführt werden kann, werden in Abschnitt 8.1.2 präsentiert.

8.1.2 Regeln

Im Folgenden werden die entwickelten Regeln für die Identifikation kritischer und unkritischer Verwendungen der in Abschnitt 8.1.1 aufgeführten Weak-Words präsentiert. Ein Ziel der Regelformulierung ist, nicht für jedes einzelne Wort separate Regeln zu entwerfen. Sofern möglich, soll über bestimmte Wörter, die in dieselbe Klasse von Adjektiven fallen, generalisiert werden. Dies soll den Aufwand der Regelfindung reduzieren und gleichzeitig eine höhere Abdeckung an zu prüfenden Weak-Words ermöglichen.

Durch manuellen Abgleich der Satzkontexte, in welchen die Weak-Words im Entwicklungskorpus stehen, konnten die folgenden fünf Wortgruppen abgeleitet werden:

- 1) *lang, kurz, schnell, breit, hoch, niedrig* (= Dimensionsadjektive)
- 2) *groß, klein* (= Dimensionsadjektive)
- 3) *maximal, minimal* (= Grenzadjektive)
- 4) *eigentlich, tatsächlich* (= Grenzadjektive)
- 5) *mal* (= Adverb)

In einem ersten Schritt wurden Instanzen von Weak-Words und deren Satzkontexte im Entwicklungskorpus betrachtet, anschließend die Kontexte, in welchen die Weak-Words stehen können, nach Kritikalität kategorisiert und darauffolgend Regeln abgeleitet. In Tabelle 19 sind die entwickelten Regeln für die Weak-Words der Gruppe 1) zu sehen. Grün markierte Regeln (R1_lang bis R24_lang) identifizieren unkritische Instanzen der Weak-Words (z.B. *5 Sekunden lang*), d.h. Instanzen von Weak-Words, bei welchen kein Interpretationsspielraum aufgrund fehlender Information auftritt. Rot hinterlegte Regeln (R25_lang bis R29_lang) decken Fälle ab, in welchen die Verwendung des Weak-Words kritisch ist (z.B. *eine längere Zeit*). Da die entwickelten Regeln in CQP-Code teilweise sehr komplexe Formen annehmen, sind die Regeln in einer einfacher lesbaren Form dargestellt. Die CQP-Regeln sind in Anhang C der Arbeit aufgeführt.

In Tabelle 20 ist neben grün (R1_groß bis R24_groß) und rot (R26_groß bis R29_groß) markierten Regeln auch eine gelb (R25_groß) markierte Regel vorhanden. Gelb markierte Regeln fangen Instanzen von Weak-Words ab, bei welchen der Autor prüfen muss, ob die

Regelname	Zusammensetzung	Beispiel
R1_lang	CARD/ein.* NN/NE/ms/sek./... WW	3 Sekunden lang
R2_lang	WW.COMP als? CARD NN/NE/ms/sek/...?	länger als 3 Sekunden
R3_lang	WW.COMP ADJ? als für? best.ART/PDAT	länger aktiv als für die Nachlaufzeit
R4_lang	wie WW	wie lange
R5_lang	([A-Za-z]+_S+ [A-Z][a-z]*[A-Z]+) WW	ABC_MunT lang
R6_lang	WW.COMP V+ als	wird nicht länger betrieben als
R7_lang	WW.COMP als ([A-Za-z]+_S+ [A-Z][a-z]*[A-Z]+)	länger als ABC_MunT
R8_lang	WW NN/NE von weniger/mehr? als? CARD	langes Drücken von weniger als 3 Sekunden
R9_lang	WW.COMP best.ART/PDAT NN/NE	länger der Nachlaufzeit
R10_lang	WW NN/NE nach ISO.*/DIN.*	lange Pulse nach ISO 7636-2
R11_lang	= WW NN/NE	UN = langes Drücken
R12_lang	WW NN/NE : ([A-Za-z]+_S+ [A-Z][a-z]*[A-Z]+)	lange Blockerkennungszeit: t_block_s
R13_lang	WW NN/NE beträgt/ist CARD	die lange Nachlaufzeit beträgt 5 Sekunden
R14_lang	best.ART/PDAT WW.COMP best.ART/PDAT	die längere der beiden Zeiten
R15_lang	genauso/ebenso/so WW	so lange
R16_lang	ART/PDAT/APPR WW.SUP	die längste Strecke
R17_lang	CARD NN/NE/ms/sek/... und/oder WW.COMP	5 cm und länger
R18_lang	best.ART/PDAT WW/COMP	die längere Zeit
R19_lang	CARD .{1-3} / .{1-3} WW	20 m/s schnell
R20_lang	WW.COMP NN/NE als	längere Zeiten als
R21_lang	APPR/APPRART WW.COMP NN/NE	bei längeren Zeiten
R22_lang	APPR best.ART/PDAT WW NN/NE	durch die langen Abstände
R23_lang	WW NN/NE bedeutet	lange Abstände bedeutet
R24_lang	nächst.* WW.COMP NN/NE?	nächst längere Zeit
R25_lang	zu WW	zu lang
R26_lang	unbest.ART WW.COMP ADJ*	eine längere Zeit
R27_lang	WW NN/NE von? ca.	lange Nachlaufzeit ca.
R28_lang	ca. CARD NN/NE /ms/sek/... WW	ca. 5 Sekunden lang
R29_lang	WW.COMP als ca.	länger als ca.

Tabelle 19: Regeln für die Weak-Words *lang*, *kurz*, *schnell*, *breit*, *hoch* und *niedrig*

relevante Information innerhalb des betreffenden Lastenhefts angegeben ist; zum Beispiel wenn auf eine Norm oder einen Parameter verwiesen wird. Nur wenn die Norm oder der Parameter eindeutig definiert sind, ist die Anforderung als unkritisch anzusehen. Die gelb hinterlegten Regeln decken die Kategorie der “Effects” ab (vgl. Abschnitt 7.1). In den Tabellen 19 bis 23 sind die entwickelten Regeln der fünf Wortgruppen abgebildet.

Das Vorgehen bei einer automatischen Analyse soll das Folgende sein: Es werden zunächst alle Sätze, die ein Weak-Word-Lemma enthalten, extrahiert und in einem Subkorpus abgelegt. Anschließend werden die den Weak-Words zugeordneten Regeln auf die einzelnen Sätze angewandt. In einem ersten Schritt werden die Positiv-Regeln angewandt (oben grün hinterlegt), um Instanzen von Sätzen mit Weak-Words auszusortieren, die dem Autor nicht aufgezeigt werden sollen. Die Instanzen, auf die keine Positiv-Regel passt, sind potentiell kritisch und werden dem Autor zur Kontrolle aufgezeigt. Wenn eine Anforderung

nung mehrere kritische Instanzen von Weak-Words enthält, wird mehrmals gewarnt. Die kritischen Weak-Word-Instanzen, für welche nicht die Standard-Fehlermeldung, sondern eine adaptierte Fehlermeldung ausgegeben werden soll, werden durch Negativ-Regeln abgefangen (rot markiert). Bei den Instanzen von Weak-Words, die weder durch eine Positiv- noch durch eine Negativ-Regel abgefangen werden, wird eine Standardmeldung ausgegeben (vgl. hierzu Abschnitt 9.1).

Regelname	Zusammensetzung	Beispiel
R1_groß	WW sein ? als? NN/NE? CARD	größer 18 Volt
R2_groß	best.ART/PDAT WW best.ART/PDAT	die größere der beiden Spannungen
R3_groß	so WW zu V* ,? dass	muss so groß sein, dass
R4_groß	WW NN/NE von CARD	große Spannung von 18 Volt
R5_groß	WW gleich CARD	größer gleich 8 Volt
R6_groß	NN/NE WW best.ART/PDAT	Ist X größer der geforderten Dosiermenge
R7_groß	WW als? ([A-Za-z]+_ \S+ [A-Z][a-z]*[A-Z]+)	größer XAI_mnOT
R8_groß	so WW wie	so groß wie
R9_groß	CARD NN/NE/ms/sek/... WW	25 mm groß
R10_groß	WW VAINF als	größer sein als
R11_groß	WW als best.ART/PDAT NN/NE	größer als die Solltemperatur
R12_groß	NN/NE WW.COMP NN/NE	Motordrehzahl größer Schwelle
R13_groß	(WW)	(groß)
R14_groß	ART/PDAT/APPR? WW.SUP	der größte Wert
R15_groß	genauso/ebenso/so WW	genauso groß
R16_groß	wie WW	wie groß
R17_groß	WW.COMP und/oder gleich	größer oder gleich 60 km/h
R18_groß	CARD NN/NE/ms/sek/... und/oder WW.COMP	5 cm und größer
R19_groß	best.ART/PDAT WW.COMP NN/NE	der größere Wert
R20_groß	WW.COMP NN/NE als	größere Kurven als
R21_groß	APPR/APPRART WW.COMP NN/NE	bei größeren Werten
R22_groß	APPR best.ART/PDAT WW NN/NE	durch die großen Abstände
R23_groß	WW NN/NE bedeutet	große Abstände bedeutet
R24_groß	nächst.* WW.COMP NN/NE?	nächst größere Gang
R25_groß	WW als NN/NE	größer als Solltemperatur
R26_groß	unbest.ART WW.COMP	ein größerer Wert
R27_groß	möglichst WW	möglichst groß
R28_groß	ausreichend WW	ausreichend groß
R29_groß	relevant WW	relevant größer

Tabelle 20: Regeln für die Weak-Words *groß* und *klein*

Regelname	Zusammensetzung	Beispiel
R1_maximal	ART WW NN/NE von CARD	ein maximaler Verbrauch von 500 mA
R2_maximal	WW CARD	maximal 3 Volt
R3_maximal	WW .+ = CARD	maximal I0max = 16
R4_maximal	WW bis	maximal bis
R5_maximal	WW NN/NE !V* beträgt ist sind CARD	maximale Verweildauer beträgt 2 Sek
R6_maximal	WW mit? ART NN/NE von	maximal mit einer Länge von
R7_maximal	best.ART/PDAT/APPR ADJ* WW ADJ* NN/NE	die maximal zulässige Abweichung
R8_maximal	\= WW	= maximal
R9_maximal	WW NN/NE : 5	maximale Eindringtiefe: 5
R10_maximal	WW NN/NE (CARD	R1_maximal maximale PreCon-Dauer (50
R11_maximal	WW NN/NE (gemäß gemaess gemäß laut)	maximale Flankensteilheit gemäß
R12_maximal	WW NN/NE !V* : +/- CARD	maximale Prüftemperatur: + 60
R13_maximal	WW NN/NE V CARD [] V	maximale Dosiermenge muss 1 mg betragen
R14_maximal	WW zu CARD	maximal zu 60%
R15_maximal	<s> WW NN/NE	Druck: Maximale Flächenpressung
R16_maximal	WW jedoch/dennoch/aber/allerdings/um CARD	maximal aber 5
R17_maximal	WW : CARD	maximal: 5
R18_maximal	WW .+mal	maximal einmal
R19_maximal	WW NN/NE bedeutet	maximale Abstände bedeutet
R20_maximal	WW +/-	maximal +
R21_maximal	PPOSAT WW ADJ NN/NE	seinen maximalen zulässigen Wert

Tabelle 21: Regeln für die Weak-Words *maximal* und *minimal*

Regelname	Zusammensetzung	Beispiel
R1_eigentlich	best.ART/KOUS/PIAT/PDAT WW NN/NE/ADJA/ADJD/KON/ADV/APPRART* NN/NE	die eigentliche aktuelle KI30 Versorgung
R2_eigentlich	(WW NN/NE/ADJA/ADJD{0,3} NN/NE	(eigentliche aktuelle Versorgung
R3_eigentlich	PPOSAT WW NN/NE/ADJA/ADJD/KON/ADV* NN/NE	ihre eigentlichen Befestigungen
R4_eigentlich	V* WW	gelten eigentlich nur
R5_eigentlich	WW V*	eigentlich ist es nicht möglich

Tabelle 22: Regeln für die Weak-Words *eigentlich* und *tatsächlich*

Regelname	Zusammensetzung	Beispiel
R1_mal	CARD ADJA ein.* WW	5 mal, beim dritten mal
R2_mal	erst.* WW	das erste mal
R3_mal	([A-Za-z]+_\\S+ [A-Z][a-z]*[A-Z]+) WW	ABC_LUI mal
R4_mal	[0-9]+ten WW	nach dem 3ten mal
R5_mal	CARD - WW	4-mal
R6_mal	noch WW	noch mal
R7_mal	auch WW	auch mal
R8_mal	V* WW	kann mal klemmen

Tabelle 23: Regeln für das Weak-Word *mal*

8.1.3 Evaluierung der Regeln

Die entworfenen Regeln wurden am Entwicklungskorpus und am Evaluierungskorpus evaluiert. Da Weak-Word-Lemmata eine hohe Frequenz in den Lastenheften aufweisen und da jede Weak-Word-Instanz bei der Evaluierung manuell geprüft wird, wurden für die Evaluierung zufällig Lastenhefte aus den beiden Korpora ausgewählt. Eine solche manuelle Evaluierung an allen Lastenheften ist aufgrund der hohen Weak-Word-Frequenz nicht handhabbar.

Aus dem Entwicklungskorpus wurden 7 Lastenhefte zufällig ausgewählt, aus dem Evaluierungskorpus 4 Lastenhefte. Insgesamt beträgt die Anzahl der analysierten Weak-Word-Instanzen im Entwicklungskorpus 541 und im Evaluierungskorpus 582.

Weak-Word	Frequenz	Keine Warnung	Warnung	Fehlwarnung	Fehlentwarnung
<i>lang</i>	33	18	15	0	2
<i>kurz</i>	54	16	38	0	0
<i>schnell</i>	20	3	17	0	0
<i>hoch</i>	53	18	35	7	2
<i>breit</i>	5	1	4	0	0
<i>niedrig</i>	10	2	8	1	0
<i>maximal</i>	74	64	10	8	0
<i>minimal</i>	47	35	12	3	3
<i>eigentlich</i>	7	6	1	0	0
<i>tatsächlich</i>	17	11	6	4	0
<i>groß</i>	121	100	21	2	0
<i>klein</i>	82	71	11	3	0
<i>mal</i>	18	18	0	0	0
Σ	541	363	178	28	7

Tabelle 24: Auswertung der Analyse am Entwicklungskorpus, N=541

In Tabelle 24 sind die Evaluierungsergebnisse am Entwicklungskorpus zu sehen. In der ersten Spalte steht das Weak-Word, die zweite Spalte enthält die Anzahl der Instanzen jedes Weak-Words sowie die Gesamtanzahl aller analysierten Weak-Word-Instanzen. In Spalte drei und vier sind die Interpretationen zu sehen, welche durch die Analyse anhand der entworfenen Regeln getroffen wurden. Diese Zahlen stellen dar, wie viele Warnungen an den Benutzer ausgegeben wurden und wie viele Instanzen von Weak-Words automatisiert als unkritisch interpretiert wurden. Die letzten beiden Spalten zeigen an, wie viele Fehlwarnungen das System produziert und wie viele Warnungen an den Nutzer fälschlicherweise unterdrückt wurden (Fehlentwarnungen), sie vergleichen also mit der manuell erstellten (Gold-)Interpretation. Das Weak-Word-Lemma *lang* beispielsweise ist in den ausgewählten Lastenheften aus dem Entwicklungskorpus 33 mal enthalten. Die automatisierte Analyse hat bei 18 Instanzen von *lang* keine Warnung ausgegeben, und in 15

Fällen wurde eine Warnung ausgegeben. Insgesamt wurden null Fehlwarnungen produziert, jedoch zwei Fehlentwarnungen; d.h. zweimal wurde eine Warnung fälschlicherweise unterdrückt. Aus welchen Gründen Fehlwarnungen und Fehlentwarnungen bei der Analyse entstehen, wird in der Fehlerdiskussion in diesem Abschnitt diskutiert.

Die Gesamtevaluierung ist nochmals in Abbildung 35 zu sehen. Es wurden am Entwicklungskorpus insgesamt 28 Fehlwarnungen ausgegeben und 7 Warnungen fälschlicherweise unterdrückt: diese sieben kritischen Instanzen werden dem Autor vorenthalten, obwohl sie überarbeitet werden müssten. 150 Instanzen von Weak-Words wurden korrekt als kritisch und 356 Instanzen korrekt als unkritisch kategorisiert. Gängige Analysen, die auf Wortlistenabgleichen basieren, würden hier 70,97% Fehlwarnungen produzieren (384 von 541 Warnungen sind Fehlwarnungen). Die in dieser Arbeit vorgestellte Kontextanalyse von Weak-Words produziert am Entwicklungskorpus nur 15,73% Fehlwarnungen (28 von 178 Warnungen sind Fehlwarnungen). Die Precision-, Recall- und F-Measure-Werte sind in Tabelle 25 angegeben.

		Automatische Analyse →	
		Kritische Anforderung	Unkritische Anforderung
Manuelle Analyse	↓	Kritische Anforderung	Unkritische Anforderung
		150	7
		28	356

Abbildung 35: Evaluierung der Weak-Word-Regeln am Entwicklungskorpus, N=541

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	84,27%	98,07%
Recall	95,54%	92,71%
F-Measure	89,55%	95,31%

Tabelle 25: Precision-, Recall- und F-Measure-Werte der Weak-Word-Regeln am Entwicklungskorpus, N=541

Die Ergebnisse am Evaluierungskorpus fallen nur geringfügig schlechter aus (vgl. Tabelle 26 und Abbildung 36). Bei der Analyse der ausgewählten Lastenhefte wurden 58 Fehlwarnungen produziert und 5 Warnungen wurden dem Autor fälschlicherweise vorenthalten. 169 kritische Weak-Word-Instanzen wurden korrekt als solche erkannt und 350 unkritische Instanzen wurden ebenfalls korrekt erkannt. Ein wortlistenbasierter Ansatz liefert 70,1% Fehlwarnungen, der in dieser Arbeit entwickelte Ansatz liefert 25,55% Fehlwarnungen. Die Precision-, Recall- und F-Measure-Werte liegen zwar in einem niedrigeren Bereich als bei der Analyse am Entwicklungskorpus, im Vergleich zu einem wortlistenbasierten Ansatz kann die Anzahl der Fehlwarnungen jedoch stark gesenkt werden (vgl. Tabelle 27).

Die Fehlwarnungen entstehen aus unterschiedlichen Gründen, die im Folgenden aufgelistet werden:

- (1) Rechtschreibfehler,
- (2) Fehlanalysen der Annotationswerkzeuge,
- (3) fehlende Regeln,
- (4) Klammereinschübe und eingeschobene Sätze,
- (5) Weak-Word ist eine Richtungsangabe,
- (6) Weak-Word muss oder kann vom Autor an der analysierten Textstelle noch nicht genau spezifiziert werden.

Weak-Word	Frequenz	Keine Warnung	Warnung	Fehlwarnung	Fehlentwarnung
<i>lang</i>	53	42	11	4	0
<i>kurz</i>	27	4	23	2	0
<i>schnell</i>	18	2	16	3	0
<i>hoch</i>	104	43	61	15	2
<i>breit</i>	0	0	0	0	0
<i>niedrig</i>	20	4	16	2	0
<i>maximal</i>	183	152	31	7	0
<i>minimal</i>	54	36	18	2	0
<i>eigentlich</i>	2	2	0	0	0
<i>tatsächlich</i>	11	10	1	0	0
<i>groß</i>	46	26	20	5	1
<i>klein</i>	47	21	26	14	2
<i>mal</i>	17	13	4	4	0
Σ	582	355	227	58	5

Tabelle 26: Auswertung der Analyse am Evaluierungskorpus, N=582

Die erste Kategorie für das Entstehen von Fehlwarnungen stellen Rechtschreibfehler in Anforderungen dar. In Beispiel 70 ist ein solcher Fall zu sehen. Das koordinierte Kompositums-Erstglied *Hoch-* wurde fälschlicherweise klein und ohne Bindestrich geschrieben. Dieser Rechtschreibfehler führt dazu, dass der Part-of-Speech-Tagger dem Kompositums-Erstglied das Tag Adjektiv annotiert und das Wort somit in der Weak-Word-Analyse betrachtet wird. Der Fall wird durch keine Regel abgefangen und eine Warnung wird fälschlicherweise ausgegeben.

Beispiel 70 *Prüfung der hoch und Tieftemperaturlagerung*

Weitere Fehlwarnungen entstehen durch Fehlanalysen der Annotationswerkzeuge *mate* und *MarMot*. In Beispiel 71 ist eine Anforderung zu sehen, bei welcher eine solche Fehl-

		Automatische Analyse →	
		Kritische Anforderung	Unkritische Anforderung
Manuelle Analyse	↓	169	5
		58	350

Abbildung 36: Evaluierung der Weak-Word-Regeln am Evaluierungskorpus, N=582

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	74,45%	98,59%
Recall	97,13%	85,78%
F-Measure	84,29%	91,74%

Tabelle 27: Precision-, Recall- und F-Measure-Werte der Weak-Word-Regeln am Evaluierungskorpus, N=582

interpretation getroffen wurde. Aufgrund des Synkretismus zwischen der Positiv- und der Komparativform trifft der Morphologie-Tagger die falsche Interpretation: MarMot annotiert die Komparationsform Positiv, obwohl *kleiner* in diesem Fall den Komparativ darstellt. Dies führt dazu, dass die entsprechende Regel für die Identifikation unkritischer Weak-Word-Instanzen nicht greifen kann und eine Fehlwarnung produziert wird.

Beispiel 71 *Abtriebsdrehzahl kleiner Schwelle*

Ein Fall, bei welchem aufgrund einer fehlenden Regel eine Fehlwarnung produziert wird, ist in Beispiel 72 zu sehen. Die Konstruktion *kleiner/gleich 20* wird bisher durch keine Regel abgefangen. Da die Instanz dennoch unkritisch ist, weil kein Interpretationsspielraum aufgrund fehlender Information entsteht, ist es sinnvoll, eine Regel zu entwerfen, die Konstruktionen dieser Art abfängt. Die Regel könnte folgendermaßen aussehen `WW/gleich CARD`.

Beispiel 72 *Es werden keine Schlüsselsuchen auf Betätigung eines kapazitiven Türgriffsensors mehr durchgeführt, bis der Wert wieder kleiner/gleich 20 ist.*

Klammereinschübe sind ein weiterer Grund für die Entstehung von Fehlwarnungen. In Beispiel 73 ist eine Anforderung mit der Weak-Word-Form *hohen* und einem Klammereinschub zu sehen. Ohne den Klammereinschub ist das Weak-Word aufgrund fehlender Information nicht eindeutig. Durch die Information in der Klammer wird das Weak-Word jedoch genauer spezifiziert, wodurch eine Warnung an den Autor überflüssig wird.

Beispiel 73 *Bei sehr hohen Lenkradwinkelgeschwindigkeiten (Richtwert: > 900 °/s) darf diese Anforderung verletzt werden.*

Klammereinschübe bereiten generell Probleme bei der Weak-Word-Analyse. Eine mögliche Lösung für das Problem wäre, eine Regel zu entwerfen, welche prüft, ob in einer Klammer, die einen bestimmten Abstand (= Anzahl Tokens) zu einem Weak-Word hat, ein messbarer Wert angegeben ist und bei Vorhandensein einer solchen Klammer keine Warnung auszugeben. Jedoch bedeutet eine messbare Angabe in einer Klammer nicht notwendigerweise, dass sie sich auf das Weak-Word bezieht. Hierbei besteht die Gefahr zu viele kritische Instanzen von Weak-Words zu übergehen und die Menge der Fehlentwarnungen zu steigern. Da bei der Weak-Word-Analyse nicht zu viele kritische Instanzen übergangen werden sollen, muss mit einer bestimmten Anzahl an Fehlwarnungen aufgrund von Klammereinschüben gerechnet werden.

Eingeschobene Sätze und Appositionen sind ein weiteres Problem bei der Weak-Word-Analyse. In Beispiel 74 ist eine Anforderung mit dem Weak-Word *maximal* und einer Apposition angegeben. Fälle wie diese werden im Normalfall von Regel R1_maximal (vgl. Tabelle 21) abgefangen. Jedoch kann aufgrund des Einschubs die Regel nicht greifen und es wird eine Fehlwarnung ausgegeben. Wie Klammereinschübe, sind auch eingeschobene Sätze ein Problem bei der Weak-Word-Analyse. Je weiter der messbare Wert im Satz von dem Weak-Word, auf welches er sich bezieht, entfernt ist, desto kleiner wird die Wahrscheinlichkeit, dass sich der Wert auf das Weak-Word bezieht. Auch hier muss mit Fehlwarnungen gerechnet werden, damit nicht zu viele Fehlentwarnungen entstehen.

Beispiel 74 *Bei einem Handmoment von 0,5 Nm ist ein maximales Motormoment, bezogen auf das Lenkrad, von 10 Nm erlaubt.*

Die fünfte Fehlerkategorie beschreibt Instanzen, bei welchen Weak-Words als Richtungsangabe fungieren. In Beispiel 75 ist ein solcher Fall zu sehen. Hier steht das Weak-Word *hoch* für eine Bewegungsrichtung des Außenspiegels. Das Wort *hoch* kann unterschiedliche Bedeutungen haben. Die Fälle, in welchen *hoch* für eine Bewegungsrichtung steht, sind nicht als kritisch zu betrachten. Diese semantische Mehrdeutigkeit kann durch die Regeln nicht disambiguiert werden, wodurch Fehlwarnungen entstehen. Eine Möglichkeit, solche Fälle abzufangen, wäre zu prüfen, welche Wörter sich im direkten Kontext des Weak-Words befinden. In Beispiel 75 könnte das Wort *links* ein Hinweis darauf sein, dass *hoch* als Richtungsangabe fungiert. Jedoch kann nicht darüber generalisiert werden, dass Schlagwörter wie *links* grundsätzlich darauf hinweisen, dass es sich bei einem angrenzenden Weak-Word um eine Richtungsangabe handelt.

Beispiel 75 *Spg_LI_H_X "Move left mirror up Aussenspiegel links - hoch"*

Die sechste Fehlerkategorie beinhaltet Fälle, bei welchen das Weak-Word zwar unterspezifiziert ist, jedoch zu dem Zeitpunkt des Schreibens noch nicht konkretisiert werden muss.

Die Anforderung in Beispiel 76 beinhaltet einen solchen Fall. In dieser Anforderung wird darauf hingewiesen, dass ein Maximalwert noch vom Auftraggeber definiert werden muss. Dieser Fall muss bei einer automatischen Analyse dem Autor nicht zur Kontrolle aufgezeigt werden, kann jedoch durch keine Kontextregel abgefangen werden. Somit können bei dieser Fehlerkategorie Fehlwarnungen entstehen.

Beispiel 76 *Der Staudruck, der dabei maximal durch die Komponente verursacht werden darf, ist vom Auftraggeber noch festzulegen.*

Die Entstehung von Fehlentwarnungen ist auf zwei Gründe zurückzuführen:

- (1) Eine Positivregel fängt teilweise auch kritische Instanzen ab.
- (2) Eine Positivregel ist mächtiger als eine Negativregel.

Fünf Fehlentwarnungen im Entwicklungskorpus sind daraus entstanden, dass Positivregeln teilweise auch kritische Instanzen abfangen (Fehlerkategorie 1). Es handelt sich hierbei um die Regeln `R15_maximal` und `R21_lang`.

Regel `R15_maximal` soll Instanzen abfangen, welche Definitionen enthalten. Zum Beispiel, wenn ein Term wie *Druck* definiert wird und keine explizite Quantifizierung nötig ist (vgl. Tabelle 21). Bei der Beschreibung eines Ziels, wie in Beispiel 77, wird aber kein Term definiert. Daher ist hier das nach dem Doppelpunkt folgende Weak-Word als kritisch zu erachten. Im Entwicklungskorpus entstehen aufgrund dieser Regel drei Fehlentwarnungen.

Beispiel 77 *Ziel: minimale Lochgrösse, kein Aufreißen des Stichloches, keine Beeinträchtigung der Nahtfestigkeit.*

Regel `R21_lang` soll Instanzen abfangen, bei welchen vor einem Weak-Word, das im Komparativ steht, eine Präposition vorhanden ist und nach dem Weak-Word ein Nomen steht (z.B.: *bei längeren Zeiten*). Bei Instanzen wie diesen wird davon ausgegangen, dass das Weak-Word einen Vergleich zwischen dem nachfolgenden Nomen und einem Wert darstellt, der in einem vorausgehenden Satz genannt wird. Wenn dieser Vergleich eindeutig zugeordnet werden kann, handelt es sich um eine unkritische Instanz eines Weak-Words. Kann der Vergleich jedoch nicht eindeutig zugeordnet werden, handelt es sich um eine kritische Weak-Word-Instanz. Im Entwicklungskorpus tritt der Fall, dass Regel `R21_lang` aufgrund eines fehlenden Vergleichs eine Fehlentwarnung produziert, zwei mal auf. Eine dieser beiden Anforderungen ist in Beispiel 78 zu sehen.

Beispiel 78 *Der Auftragnehmer darf das Screening auch dann nicht einstellen, wenn über längere Zeit kein Ausfall beobachtet wurde.*

Um Fehlentwarnungen dieser Art zu vermeiden, ist es sinnvoll, Regel R21_lang nicht als Positivregel, sondern als Effect-Regel zu kategorisieren. Somit wird der Autor zukünftig bei Konstruktionen wie *bei längeren Zeiten* dazu aufgefordert, zu überprüfen, ob der Vergleich, welcher mit einem Weak-Word angedeutet wird, auch wirklich vorhanden und eindeutig ist. Ist der Vergleich eindeutig, muss die Anforderung vom Autor nicht geändert werden. Im Falle, dass der Vergleich nicht eindeutig zugeordnet werden kann, muss der Autor die Anforderung nochmals überarbeiten.

Ein weiteres Problem stellt die Reihenfolge bzw. die Mächtigkeit der Regeln dar (Fehlerkategorie 2). Im Entwicklungskorpus wurden zwei Instanzen von der Art, welche in Beispiel 79 dargestellt ist, nicht als Warnung ausgegeben. Im Normalfall sollten diese Instanzen durch Regel R28_lang (= *ca. 10 mm breit*) abgefangen werden und an den Autor als Warnung ausgegeben werden. Da zuerst Regel R1_lang (= *10 mm breit*) auf die Anforderung angewendet wird und somit die Instanz den unkritischen Formulierungen zugeordnet wird, kann Regel R28_lang nicht mehr greifen und es wird fälschlicherweise keine Warnung ausgegeben. Es ist wichtig zu definieren, wie mächtig eine Regel ist und ob die Positiv-Regeln oder die Negativ-Regeln die jeweils andere Kategorie überlagern. Durch eine geeignete Sortierung der Regeln, kann die Zahl der Fehlentwarnungen nochmals gesenkt werden. Im gezeigten Beispiel würde dies bedeuten, dass R28_lang vor R1_lang angewandt werden müsste.

Beispiel 79 *Für die Befestigung ist eine umlaufende Klebefläche (ca. 10 mm breit) angedacht.*

Im Evaluierungskorpus treten die gleichen Kategorien von Fehlentwarnungen auf. Vier Fehlentwarnungen sind der Kategorie (1) zuzuordnen, eine Fehlentwarnung der Kategorie (2).

Durch weitere Überarbeitung der Regeln, durch Hinzufügen neuer Regeln und durch Sortierung nach deren Mächtigkeit, kann die Menge der Fehlwarnungen und Fehlentwarnungen weiter gesenkt werden. Bei näherer Betrachtung der Regeln fällt außerdem auf, dass sich die Weak-Word-Gruppen 1) und 2) sehr ähnlich sind. Viele Regeln sind identisch oder können problemlos auf die andere Gruppe übertragen werden. Dies bedeutet, dass die behandelten Weak-Words nicht wie zunächst angenommen in fünf Gruppen unterteilt werden, sondern sogar in 4 Gruppen zusammengefasst werden können und somit die Überdeckung der möglichen Fälle von Kontexten erweitert wird.

8.1.4 Zusammenfassung

Die Verwendung von Weak-Words in Anforderungen kann dazu führen, dass für das einheitliche Verständnis zwischen Autor und Leser relevante Information nicht explizit angegeben wird. Durch Dimensionsadjektive wie *lang* ohne quantifizierende Angaben kann Interpretationsspielraum entstehen. Weak-Words haben aber nicht grundsätzlich einen negativen Einfluss auf die Verständlichkeit einer Anforderung. Weak-Words in Kombination mit einer Quantifizierung, einem Verweis auf eine Norm oder in Kombination mit Angaben weiterer für das Verständnis wichtiger Information sind nicht als kritisch zu betrachten und können in Anforderungen problemlos verwendet werden. Da die Kritikalität der Weak-Words von dem Satzkontext anhängig ist, in dem sie stehen, muss dieser bei einer automatisierten Analyse einbezogen werden. Eine reine Wortlistensuche reicht folglich nicht aus, um kritische Instanzen automatisiert identifizieren zu können.

Durch die in dieser Arbeit entwickelte Kontextanalyse können kritische Weak-Word-Verwendungen erkannt und die Menge an Fehlwarnungen verglichen mit einem wortlistenbasierten Ansatz stark gesenkt werden. Jedoch treten auch bei der Kontextanalyse eine gewisse Anzahl an Fehlwarnungen und eine geringe Menge an Fehlentwarnungen auf. Weak-Words können in sehr variablen Kontexten verwendet werden, wodurch es hin und wieder vorkommen kann, dass für die Verarbeitung weiterer Anforderungstexte neue Regeln benötigt werden. Jedoch ist zu erwarten, dass der Bedarf an neuen Regeln mit zunehmender Menge an analysierten Lastenheften stark sinkt. Durch weitere Überarbeitung der Regeln, durch Hinzufügen neuer Regeln und durch Sortierung nach Mächtigkeit der Regeln ist zu erwarten, dass die Analyse stabiler wird und immer weniger Aufwand für die Überarbeitung investiert werden muss und dabei gute Ergebnisse geliefert werden.

8.2 Unpersönliche Konstruktionen

Das Phänomen *Unpersönliche Konstruktion* ist im Qualitätsmodell dem Kriterium *Fehlende Information* zugeordnet. Zu den unpersönlichen Konstruktionen zählen manche der Verwendungsweisen des Pronomens *es* sowie Sätze mit dem Pronomen *man*. Durch die Verwendung unpersönlicher Konstruktionen kann der Autor in einem Satz die semantische Rolle des Agens unausgedrückt lassen, d.h. die ausführende Person bzw. die ausführende Komponente oder das ausführende System wird nicht explizit genannt. Die Tilgung des Agens führt aber nicht grundsätzlich zu Problemen, da oft durch Welt-, Kontext- oder Expertenwissen das Agens hergeleitet werden kann. Manche unpersönliche Konstruktionen (z.B. Passiversatz mit *es ist zu ...*, *es muss ... werden*) könnten aber zu Verständnisproblemen führen. Umgekehrt sind aber auch nicht alle Verwendungen von *es* oder *man* problematisch.

In Abschnitt 8.2.1 soll zunächst das Phänomen der unpersönlichen Konstruktionen näher beschrieben werden und die verschiedenen Kontexte, in welchen das Pronomen *es* in den untersuchten Anforderungen auftritt, werden in Gebrauchsweisen eingeteilt. Anschließend wird in Abschnitt 8.2.2 untersucht, ob sich Regeln für die automatische Identifikation kritischer Instanzen entwickeln lassen. Eine ausführliche Untersuchung des Pronomens *man* wird nicht durchgeführt, da dessen Frequenz in den Korpora zu niedrig ist und das Pronomen in den Korpora nur in unkritischen Satzkontexten auftritt (vgl. Abschnitt 8.2.1).

8.2.1 Phänomenbeschreibung

Durch Verwendung unpersönlicher Konstruktionen wie *es* und *man* kann der Autor einer Anforderung die semantische Rolle des Agens eines Satzes unausgedrückt lassen, d.h. das Agens wird nicht explizit im Satz genannt. Dies kann zu Problemen führen, wenn das Agens nicht durch Welt-, Kontext- oder Expertenwissen hergeleitet werden kann. Im Folgenden soll zunächst auf die für die Anforderungen relevanten Konstruktionen mit dem Pronomen *es* eingegangen werden und später im Abschnitt auf die Verwendungen von *man*.

Bei dem Pronomen *es* können insgesamt vier Gebrauchsweisen unterschieden werden (vgl. [EKR09] und [Eis13]):

- (i) Referentielles/Phorisches *es*
- (ii) Expletives *es*
- (iii) Korrelat-*es*
- (iv) Vorfeld-*es* bzw. Vorfeldplatzhalter

Bei Gebrauchsweise (i) handelt es sich um das referentielle bzw. (ana)phorische *es*. Diese Gebrauchsweise umfasst Instanzen von *es*, die auf eine Nominalphrase oder einen anderen sprachlichen Ausdruck referieren und insofern gar nicht unpersönlich sind [EKR09]. Solche pronominalen Bezüge auf vorausgehende Diskursobjekte werden in Abschnitt 8.5 ausführlich betrachtet und im vorliegenden Abschnitt nicht weiter diskutiert.

Bei den Kategorien (ii)-(iv) handelt es sich um das unpersönliche *es*. Das Pronomen ist in diesen Fällen nicht phorisch: „Damit ist es nicht Träger einer semantischen Rolle und in diesem Sinn semantisch leer“ [Eis13]. Diese Gebrauchsweisen von *es* sollen im Folgenden genauer untersucht werden.

Bei der zweiten Gebrauchsweise handelt es sich um das expletive *es*. „Die Funktion des expletiven *es* erschöpft sich darin, die Subjektstelle zu besetzen und so zur Konstruktion

eines vollständigen Subjekt-Prädikat-Satzes beizutragen“ [Eis13]. In Beispiel 80 ist eine Anforderung angegeben, in welcher das Pronomen *es* in der expletiven Gebrauchsweise verwendet wird.

Beispiel 80 *Es gibt eine Umluftfunktion.*

Das Pronomen *es* in Beispiel 80 ist obligatorisch, d.h. es kann nicht weggelassen werden, ohne dass der Satz ungrammatisch wird; es ist Teil einer lexikalisierten verbalen Konstruktion (*es gibt X*). Diese Eigenschaft lässt sich durch die Verschiebeprobe überprüfen [EKR09]. In Beispiel 81 wurde der Satz aus Beispiel 80 ohne das Pronomen *es* formuliert. Der Satz wird durch Weglassen des Pronomens ungrammatisch. Das Pronomen ist in dieser Verwendung und in vergleichbaren idiomatischen Wendungen obligatorisch.

Beispiel 81 *Eine Umluftfunktion gibt.*

Ein weiteres Beispiel dieses Konstruktionstyps mit expletiven *es* findet sich in Beispiel 82. Auch hier ist das Pronomen *es* obligatorisch.

Beispiel 82 *Die LEDs müssen so angesteuert werden, dass es über den geforderten Temperatur- und Spannungsbereich zu keinen sichtbaren Helligkeitsschwankungen kommt.*

Bei Gebrauchsweise (iii) handelt es sich um das Korrelat-*es*. „Wenn das Pronomen *es* auf einen folgenden Nebensatz verweist, spricht man von einem Korrelat“ [EKR09]. In Beispiel 83 ist das Pronomen *es* in der Funktion eines Korrelats zu sehen.

Beispiel 83 *Die Funktion ermöglicht es, dass der Benutzer an seinem Fahrzeug aus einer begrenzten Entfernung die Fenster, das Schiebedach und das Verdeck über Funk mit dem Fahrzeugschlüssel öffnen bzw. schließen kann.*

Das Pronomen *es* ist in diesen Fällen nicht obligatorisch. Wird der Nebensatz vorangestellt, wie in Beispiel 84, wird das Pronomen *es* nicht benötigt, um einen grammatikalisch korrekten Satz zu formulieren.

Beispiel 84 *Dass der Benutzer an seinem Fahrzeug aus einer begrenzten Entfernung die Fenster, das Schiebedach und das Verdeck über Funk mit dem Fahrzeugschlüssel öffnen bzw. schließen kann, ermöglicht die Funktion.*

Bei Gebrauchsweise (iv) handelt es sich um das Vorfeld-*es*, d.h. um einen Vorfeldplatzhalter. „Im Verbzweitsatz erscheint das Subjekt bei unmarkierter Abfolge der Satzglieder in der Regel an erster Stelle im Satz (Vorfeld, d.h. vor dem finiten Verb). Unter bestimmten Bedingungen kann das Subjekt ins Mittelfeld oder Nachfeld rücken, wobei das Vorfeld von *es* besetzt wird“ [Eis13]. In Beispiel 85 ist ein solcher Fall zu sehen. Hier vertritt das Pro-

nomen im Vorfeld das Subjekt. Durch die Verschiebeprobe kann aber gezeigt werden, dass im Gegensatz zur expletiven Gebrauchsweise das Pronomen *es* entfallen kann, wenn die Satzglieder in der unmarkierten Reihenfolge stehen. In Beispiel 86 ist die Anforderung aus Beispiel 85 ohne das Pronomen *es* zu sehen. Das Pronomen ist in dieser Gebrauchsweise folglich nicht obligatorisch.

Beispiel 85 *Es liegt eine fehlerhafte Signalform vor.*

Beispiel 86 *Eine fehlerhafte Signalform liegt vor.*

Die Gebrauchsweise (iv) liegt auch bei agenslosen bzw. unpersönlichen Passivsätzen vor (vgl. Beispiel 87). Auch hier zeigt die Verschiebeprobe, dass das Pronomen nicht obligatorisch ist (vgl. Beispiel 88).

Beispiel 87 *Es wird nur die Position des jeweils betätigten Teilsystems abgespeichert.*

Beispiel 88 *Nur die Position des jeweils betätigten Teilsystems wird abgespeichert.*

Das unpersönliche Personalpronomen *man* tritt im Entwicklungskorpus lediglich 24 mal auf, im Evaluierungskorpus 13 mal. All diese Instanzen von *man* stellen Erklärungen dar. Eine Anforderung aus dem Entwicklungskorpus, die das Pronomen *man* enthält, ist in Beispiel 89 zu sehen. Diese Verwendung von *man* ist für das Verstehen des Anforderungstexts unproblematisch, da hier kein explizites Agens angegeben werden muss, sondern auf die (Fach-)Allgemeinheit verwiesen wird.

Beispiel 89 *Unter Konformität versteht man: [...]*

Da das Pronomen *man* nur sehr selten in den untersuchten Lastenheften verwendet wird und im Entwicklungskorpus nur unkritische Instanzen identifiziert werden konnten, wird diese unpersönliche Konstruktion im Folgenden nicht weiter analysiert. Der Fokus der Analysen liegt auf den optionalen Verwendungsweisen von *es*.

Im folgenden Abschnitt sollen die in den Korpora enthaltenen Instanzen der unpersönlichen Konstruktionen mit *es* untersucht und in die in diesem Abschnitt beschriebenen Gebrauchsweisen eingeteilt werden. Weiterhin soll diskutiert werden, ob sich Regeln zur automatischen Identifikation kritischer Instanzen formulieren lassen.

8.2.2 Regeln

Aufgrund der hohen Frequenz des Pronomens *es* in den untersuchten Korpora (Entwicklungskorpus: 2713 Instanzen, Evaluierungskorpus: 2501 Instanzen) wurden für eine erste

Analyse 400 Anforderungen zufällig aus dem Entwicklungskorpus extrahiert und manuell in die vier in Abschnitt 8.2.1 besprochenen Gebrauchsweisen eingeteilt. Hierbei konnten nicht nur die vier Gebrauchsweisen, sondern auch verschiedene Kontexte definiert werden, in welchen das Pronomen *es* auftreten kann. Die Annotationswerkzeuge *mate* und *MarMot* unterscheiden nicht zwischen einem referentiellen (Kategorie (i)) und einem unpersönlichen *es*, da ein Sprachverarbeitungswerkzeug für diese Unterscheidung semantisches Wissen haben müsste, das aber in beiden Werkzeugen nicht vorliegt. Alle Instanzen von *es* sind in den Korpora durch das Tag PPER (= irreflexives Personalpronomen) gekennzeichnet. Daher sind in den 400 Anforderungen auch referentielle Instanzen von *es* enthalten.

In Tabelle 28 sind die untersuchten Instanzen von *es* in eigens definierte Kategorien eingeteilt. Diese Kategorien stellen Kontextmuster dar, in welchen das Pronomen *es* in der ausgewählten Zufallsstichprobe auftritt. Die Kategorien sind den vier Gebrauchsweisen, welche in Abschnitt 8.2.1 definiert wurden, zugeordnet, und für jede Kategorie wird ein Beispiel gegeben. Zusätzlich wird zu jeder Kategorie die Anzahl der Okkurrenzen in der Stichprobe angegeben.

Gebrauchsweise	Kategorie	Beispiel	Frequenz
(i)	referentiell	<i>Wenn Rollo 1 Position A erreicht hat, stoppt es.</i>	49
(ii)	<i>es gibt</i>	<i>Es gibt eine 4x2-Variante.</i>	41
	<i>es kommt zu</i>	<i>... dass es zu keinen Störungen kommt.</i>	18
	<i>es handelt sich um</i>	<i>Es handelt sich um tauschbare Module.</i>	1
(iii)	<i>ermöglicht es</i>	<i>Die Funktion ermöglicht es, dass ...</i>	1
	<i>es reicht aus</i>	<i>Um ... reicht es aus, nur den Code zu setzen.</i>	1
(iv)	<i>es ist/sind ... zu ... + Verb</i>	<i>Es ist freigegebenes Material zu verwenden.</i>	73
	<i>es wird/werden + Passiv-Verb</i>	<i>Es wird eine Schließung erkannt.</i>	67
	<i>es liegt <NP>vor</i>	<i>Es liegt ein Kurzschluss vor.</i>	48
	<i>es + Modalverb + werden</i>	<i>Es muss ein Fehler abgelegt werden.</i>	37
	<i>es + Modalverb + Verb (aktiv)</i>	<i>Es müssen zwei Durchläufe erfolgen.</i>	21
	<i>es + Modalverb + Negation + Verb (aktiv)</i>	<i>Es dürfen keine Störungen auftreten.</i>	16
	<i>es ist/sind ... <AdjP></i>	<i>Es ist nur der Sensor aktiv.</i>	15
	<i>es gilt/gelten <NP></i>	<i>Es gilt die gemessene Temperatur.</i>	10
	<i>es bleibt</i>	<i>Es bleiben immer dieselben Signale gesetzt.</i>	2

Tabelle 28: Einteilung der untersuchten 400 Instanzen von *es* in die vier definierten Gebrauchsweisen

Die referentielle Gebrauchsweise von *es* kommt in den 400 betrachteten Anforderungen insgesamt 49 mal vor. Hierbei wurde nicht näher untersucht, ob es sich um kritische oder unkritische Instanzen handelt. Eine solche Untersuchung wird in Abschnitt 8.5 diskutiert.

60 Instanzen fallen in die Gebrauchsweise (ii), welche dem expletiven Gebrauch von *es* entspricht. In diesen Fällen ist *es* obligatorisch, d.h. es kann nicht weggelassen werden. Es entstehen keine Probleme im Bezug auf die Verständlichkeit einer Anforderung.

Bei der Durchsicht der 400 Anforderungen wurden lediglich zwei Kategorien mit jeweils einem Vorkommen identifiziert, welche in die Gebrauchsweise (iii) eingeordnet werden können. Hier dient *es* als Korrelat, d.h. das Pronomen verweist auf den folgenden Nebensatz. Auch hier wird durch die Verwendung des Pronomens nicht das Agens eines Verbs

weggelassen: auch durch diese Gebrauchsweise entstehen keine Probleme im Bezug auf die Verständlichkeit.

Gebrauchsweise (iv) umfasst die größte Anzahl an Vorkommnissen. Insgesamt 289 Instanzen konnten dieser Gebrauchsweise zugeordnet werden (= 72,25%). Die in diese Gebrauchsweise fallenden Instanzen konnten in neun Kontexttypen eingeteilt werden, in welchen das Pronomen *es* auftritt. Das Pronomen hat hier die Funktion eines Vorfeld-*es*. Das Pronomen ist in diesen Fällen nicht obligatorisch, was sich durch die Verschiebeprobe (vgl. Abschnitt 8.2.1) nachweisen lässt.

Auffällig bei Gebrauchsweise (iv) ist, dass die meisten Kontexte Ersatzkonstruktionen für Passivsätze darstellen. Lediglich die Kategorien *es liegt* <NP>*vor*, *es ist/sind* ... <AdjP>, *es gilt/gelten* <NP> und *es bleibt* stellen keinen Passiversatz dar. Bei diesen Konstruktionen stellt sich nicht die Frage, ob der Autor durch eine unpersönliche Konstruktion ein Agens unausgedrückt gelassen hat. Bei den fünf weiteren Kategorien, die in die Gebrauchsweise (iv) fallen, besteht dieses Problem potentiell. Diese Kategorien umfassen 214 Instanzen der 400 untersuchten, also 53,5%.

Die Fälle von *es*, welche Passiversatzkonstruktionen darstellen, müssen genauer untersucht werden, um eine Aussage darüber treffen zu können, ob sich Regeln für die automatische Identifikation kritischer Instanzen entwickeln lassen können. Diese Problematik wird im folgenden Abschnitt 8.3 im Zuge der Phänomenanalyse des Passivs mit eingeschlossen. Aus diesem Grund werden im vorliegenden Abschnitt keine Regeln für die automatische Identifikation vorgestellt und evaluiert.

8.2.3 Zusammenfassung

Durch die Verwendung unpersönlicher Konstruktionen wie *es* und *man* kann das Agens eines Satzes unausgedrückt gelassen werden, d.h. es wird nicht textuell wiedergegeben. Probleme können entstehen, wenn sich diese fehlende Information weder durch Welt-, Kontext-, noch durch Expertenwissen herleiten lässt. In diesem Abschnitt wurden 4 Gebrauchsweisen des Pronomens *es* vorgestellt. Lediglich in der Gebrauchsweise als Vorfeld-*es* konnten Kontexte festgestellt werden, in welchen das Problem des getilgten Agens auftreten kann, weil es sich um Passiversatzkonstruktionen handelt. Da in Abschnitt 8.3 eine detaillierte Untersuchung von Passivsätzen durchgeführt wird, werden im vorliegenden Abschnitt keine Regeln für die automatische Identifikation kritischer Instanzen vorgestellt und evaluiert.

Das unpersönliche Pronomen *man* hat eine sehr niedrige Frequenz in den untersuchten Lastenheften. Diese wenigen Vorkommnisse sind allesamt Erklärungen und sind nicht als

kritisch zu betrachten. Das Pronomen *man* wird aus diesen Gründen keiner weiteren Untersuchung unterzogen.

8.3 Passiv und Passiversatzkonstruktionen ohne Agens

Die Kategorie *Passiv ohne Agens* ist im Qualitätsmodell dem Kriterium *Fehlende Information* zugeordnet. Bei der Verwendung von Passivsätzen oder Passiversatzkonstruktionen ohne Agens (z.B. *X ist zu Infinitiv*) besteht potentiell die Gefahr, dass der Leser allein durch Kontext- oder Weltwissen die semantische Rolle Agens, also den Handelnden des Satzes, nicht herleiten kann und dies zu Interpretationsspielraum führt.

Die in dieser Arbeit durchgeführten Analysen zeigen jedoch, dass die betrachteten Passivsätze und Passiversatzkonstruktionen ohne Agens in den meisten Fällen unkritisch sind, da häufig durch Kontextwissen das Agens hergeleitet werden kann, auch wenn dieses im Text nicht explizit genannt wird. Im vorliegenden Abschnitt werden Regeln für die automatische Identifikation agensloser Passivsätze präsentiert und eine stichprobenweise Bewertung der Kritikalität des Phänomens durchgeführt; es findet jedoch keine umfangliche Evaluierung der Regeln statt.

8.3.1 Phänomenbeschreibung

Passivsätze zu vermeiden bzw. stets im Aktiv zu schreiben, ist die wohl am häufigsten genannte Regel in der Requirements Engineering-Literatur für das Schreiben guter Anforderungen (z.B. [Kof07], [Rup14], [PR15] und [Wun15]). Durch die Verwendung des Passivs kann, wie auch bei den unpersönlichen Konstruktionen, das Agens, also der Handelnde eines Satzes, bewusst unausgedrückt gelassen werden, d.h. er muss textlich nicht wiedergegeben werden. Dies kann zu Problemen führen, wenn der Leser selbst durch Kontextwissen das Agens nicht herleiten kann. In Aktivsätzen ist das Agens obligatorisch, d.h. der Autor muss das Agens explizit im Text nennen. In Passivsätzen hingegen ist das Agens optional und kann durch eine Präpositionalphrase mit *durch* oder *von* ausgedrückt werden. Da Passivsätze morphosyntaktisch eine komplexere Struktur als Aktivsätze darstellen [Buß08], ist die Empfehlung stets im Aktiv zu schreiben in jedem Falle sinnvoll, weil damit die Struktur des Satzes weniger komplex ist und das Agens stets genannt wird.

In Beispiel 90 ist eine Anforderung zu sehen, in welcher das Verb (*aktivieren*) das Genus Verbi *Aktiv* besitzt. Die semantische Rolle Agens wird durch das Subjekt ausgedrückt (*Der Benutzer*), die semantische Rolle Patiens durch das Objekt (*die Funktion*). Beide Rollen werden in Beispiel 90 textuell wiedergegeben. Die syntaktischen Funktionen Subjekt und Objekt sind obligatorisch, da sie von der Argumentstruktur des Verbs gefordert werden.

Beispiel 90 *Der Benutzer aktiviert die Funktion.*

Der Sachverhalt des Aktivsatzes aus Beispiel 90 kann ebenfalls durch einen Passivsatz ausgedrückt werden, mit Nennung des Agens in einer Präpositionalphrase (vgl. Beispiel 91) oder ohne (vgl. Beispiel 92).

Beispiel 91 *Die Funktion wird durch den Benutzer aktiviert.*

Beispiel 92 *Die Funktion wird aktiviert.*

Durch Passivsätze können dieselben Sachverhalte ausgedrückt werden wie mit Aktivsätzen. „Aktivsatz und Passivsatz kodieren [...] dieselben semantischen Rollen, aber sie tun es über unterschiedliche syntaktische Funktionen“ [Eis13]. In Abbildung 37 ist zu sehen wie sich die Zuordnung von Satzfunktion und semantischer Rolle in der Umwandlung von einem Aktiv- in einen Passivsatz verschieben. Die Nomina *Benutzer* und *Funktion* nehmen im Aktiv- und im Passivsatz jeweils dieselben semantischen Rollen ein. Jedoch ändert sich die syntaktische Funktion der beiden Nomina. Im Aktivsatz nimmt das Subjekt die semantische Rolle Agens ein, im Passivsatz ist das das Objekt das Agens. Das Patiens bildet im Aktivsatz das Objekt, im Passivsatz das Subjekt. Das Akkusativobjekt im Passivsatz ist optional und kann weggelassen werden.



Abbildung 37: Verschiebung der syntaktischen Funktionen in Aktiv- und Passivsätzen

Warum ein Passivsatz mit Agens dem Aktivsatz vorgezogen wird, hat meist den Grund, dass der Vorgangs- oder Prozesscharakter des Geschehens gegenüber der Dimension des Handelns oder Verursachens in den Vordergrund tritt [EKR09]. Hierbei kann auch die Informationsstruktur eine wichtige Rolle spielen: „Die Kodierung einer Rolle als Subjekt macht sie wichtig, die Nennung in der *von*-Phrase ist beiläufig“ [Eis13].

„Das *werden*-Passiv kommt überwiegend (zu ca. 90%) ohne Agensangabe vor“ [EKR09]. Dies kann zu Problemen führen, wenn der Leser das Agens durch Kontextwissen nicht herleiten kann. Im Falle von Beispiel 92 ist beispielsweise ohne weitere Kontextinformation

nicht eindeutig, wer oder was *die Funktion* aktiviert. Bei der Ableitung eines Testfalls kann dies zu großen Problemen führen, da hier die Angabe des Handelnden vorhanden sein muss. Kann aber diese lokal fehlende Information aus dem Kontext hergeleitet werden, z.B. durch Betrachtung vorausgehender Anforderungen, so ist das Weglassen des Agens nicht kritisch.

In [EKR09] werden drei mögliche Gründe genannt, wieso Agensphrasen in Passivsätzen häufig weggelassen werden:

- 1) Die Identität des Agens geht deutlich aus dem weiteren Zusammenhang hervor.
- 2) Der Satz ist allgemein zu verstehen.
- 3) Dem Sprecher ist die Identität des Agens unbekannt oder unwichtig, oder er will sie nicht verraten.

Bei der in dieser Arbeit betrachteten Textsorte ist Grund 1) der wohl prominenteste. Das Agens ist oft aus dem Kontext ersichtlich und muss deshalb nicht genannt werden. Zu einem gewissen Anteil haben auch die Gründe 2) und 3) einen Einfluss darauf, ob das Agens des Satzes genannt wird, da auch in Anforderungen allgemeine Aussagen zu finden sind und die Identität des Agens dort nicht immer eine wichtige Rolle spielt. Im Allgemeinen ist es jedoch empfehlenswert im Aktiv zu schreiben und somit der Gefahr vorzubeugen, dass eine falsche Interpretation aufgrund der Nicht-Nennung des Agens getroffen wird.

Es wird zwischen unterschiedlichen Arten von Passivkonstruktionen unterschieden. Neben dem syntaktischen Passiv, welches in Vorgangspassiv (auch *werden*-Passiv genannt) und Zustandspassiv (*sein*-Passiv) eingeteilt werden kann, gibt es außerdem Passiversatzkonstruktionen (z.B. das Rezipientenpassiv (*bekommen*-Passiv) oder Konstruktionen der Form *ist ... zu*). Für das Vorgangspassiv, das Zustandspassiv und ausgewählte Passiversatzkonstruktionen werden in Abschnitt 8.3.2 Regeln für die automatische Identifikation präsentiert, welche diejenigen Passivsätze identifizieren, in welchen das Agens nicht genannt wird. Anschließend wird in Abschnitt 8.3.3 eine Untersuchung präsentiert, die die Kritikalität des Passivs herausstellen soll, um zu zeigen, dass das Phänomen in der in dieser Arbeit betrachteten Textsorte kaum kritisch ist.

In Tabelle 29 sind die in dieser Arbeit betrachteten Passivformen aufgelistet, mit je einem Beispiel. In die drei erst genannten Kategorien und in die letzte Kategorie fallen auch die unpersönlichen Passivsätze, welche in Abschnitt 8.2 diskutiert wurden.

Passivform	Beispiel
Vorgangspassiv	<i>Funktion B wird aktiviert.</i>
Zustandspassiv	<i>Funktion B ist aktiviert.</i>
Passiversatzform <i>ist ... zu</i>	<i>Funktion B ist zu aktivieren.</i>
Passiversatzform <i>bekommen</i>	<i>Funktion B bekommt ein Signal mitgeteilt.</i>
Passiversatzform <i>sich lassen</i>	<i>Funktion B lässt sich aktivieren.</i>
Passiversatzform <i>-bar</i>	<i>Funktion B ist parametrierbar.</i>

Tabelle 29: Gruppen von Passiv und Passiversatz für die Regelentwicklung

8.3.2 Regeln

Bei einer ersten manuellen Durchsicht von Passivsätzen hat sich herausgestellt, dass in den Korpora eine hohe Anzahl an Passivsätzen ohne Angabe des Agens vorhanden ist, diese jedoch nur in sehr seltenen Fällen dazu führen, dass das Agens der Anforderung nicht hergeleitet werden kann. Es wurde daher früh ersichtlich, dass keine Regeln entworfen werden können, die zuverlässig zwischen den kritischen und den unkritischen Instanzen agensloser Passivsätze unterscheiden können. Durch Kontextwissen kann die vermeintlich fehlende Information fast immer hergeleitet werden. Trotz der zu erwartenden hohen Anzahl an Fehlwarnungen, die bei einer automatischen Suche nach kritischen Instanzen entstehen können, wurden CQP-Regeln für die automatisierte Identifikation entwickelt, welche im vorliegenden Abschnitt vorgestellt werden. Die Regeln können Passivsätze, in welchen das Agens nicht explizit genannt wird, erkennen. Die wirklich kritischen Fälle können aber nicht zuverlässig identifiziert werden. Die Entscheidung wann die Nicht-Nennung des Agens kritisch ist im Bezug auf die Testbarkeit einer Anforderung, kann nicht automatisiert getroffen werden. Diese Entscheidung muss durch den Benutzer der Analysen getroffen werden. Jedoch können Passivsätze ohne Angabe eines Agens identifiziert und dem Autor vorgezeigt werden; dies kann einen Hinweis darauf liefern, dass an einer bestimmten Stelle im Text Interpretationsspielraum aufgrund fehlender Information entstehen könnte. Ist dies das Ziel, so können die Regeln angewandt werden ohne viel Fehlalarm zu erzeugen.

Es wurden Regeln für die automatische Identifikation der Passivsätze und Passiversatzkonstruktionen ohne Agens, welche in Tabelle 29 aufgelistet sind, entworfen. Passivsätze, in welchen das Agens in einer Präpositionalphrase genannt wird, müssen durch diese Regeln ausgeschlossen werden. Passivkonstruktionen können in Hauptsätzen und in Nebensätzen auftreten sowie in Kombination mit einem Modalverb oder ohne ein Modalverb. Für all diese Fälle wurden Regeln entwickelt.

In Tabelle 30 werden beispielhaft die Regeln für die Identifikation von agenslosen Vorgangspassivkonstruktionen präsentiert. Jede Regel ist durch einen eindeutigen Regeln-

men gekennzeichnet, und für jede Regel wird ein Beispiel präsentiert. Die Regeln für das Zustandspassiv sowie die Regeln für die Identifikation von agenslosen Passiversatzkonstruktionen sind in Anhang D aufgelistet. Auch die Regeln für die Identifikation von agenslosen Vorgangspassivkonstruktionen sind dort nochmals aufgelistet.

Regelname	Beispiel	CQP-Regel
R1_VorgPassiv	<i>Standardkonzepte werden nicht in Form von Dokumenten ausgegeben.</i>	[pos="N(N[E]) PPER"] [* [lemma="werden"]] [* a:[word="vo(n m) durch"] [* b:[pos="VVPP"] :: a.dep!=b.wortnr within s;
R2_VorgPassiv	<i>Die Prüfspannung wird auf Stufe 1 angelegt.</i>	[pos="N(N[E]) PPER"] [* [lemma="werden"] [word!="vo(n m) durch"] [* [pos="VVPP"] within s;
R3_VorgPassiv	<i>..., dass der maximale Heizfeldstrom von 21A nicht überschritten wird.</i>	[pos="KOUS"] [* [pos="N(N[E]) PPER"] [* a:[word="vo(n m) durch"] [* b:[pos="VVPP"] [lemma="werden"] [pos!="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R4_VorgPassiv	<i>..., so dass Änderungen am Prozess dokumentiert werden.</i>	[pos="KOUS"] [* [pos="N(N[E]) PPER"] [word!="vo(n m) durch"] [* [pos="VVPP"] [lemma="werden"] [pos!="VM(FIN INF)"] within s;
R5_VorgPassiv	<i>Clipse müssen im Abstand von 50 mm positioniert werden.</i>	[pos="N(N[E]) PPER"] [* [pos="VM(FIN INF)"] [* a:[word="vo(n m) durch"] [* b:[pos="VVPP"] [lemma="werden"] :: a.dep!=b.wortnr within s;
R6_VorgPassiv	<i>Die Durchflussmenge muss erfüllt werden.</i>	[pos="N(N[E]) PPER"] [* [pos="VM(FIN INF)"] [* [word!="vo(n m) durch"] [* [pos="VVPP"] [lemma="werden"] within s;
R7_VorgPassiv	<i>..., wenn das Zeitsignal in einem Abstand von 3 Sekunden nicht abgespeichert werden kann.</i>	[pos="KOUS"] [* [pos="N(N[E]) PPER"] [* a:[word="vo(n m) durch"] [* b:[pos="VVPP"] [lemma="werden"] [pos="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R8_VorgPassiv	<i>..., dass der Fußgängerschutz erfüllt werden kann.</i>	[pos="KOUS"] [* [pos="N(N[E]) PPER"] [word!="vo(n m) durch"] [* [pos="VVPP"] [lemma="werden"] [pos="VM(FIN INF)"] within s;

Tabelle 30: Regeln für die Identifikation von Vorgangspassiv ohne Agens

Für die Identifikation von Anforderungen, in welchen das Vorgangspassiv ohne die Nennung eines Agens verwendet wird, wurden insgesamt acht Regeln entwickelt. Die Regeln R1_VorgPassiv und R2_VorgPassiv identifizieren Anforderungen, in welchen das Vorgangspassiv in einem Hauptsatz ohne Verwendung eines Modalverbs vorkommt. Durch die Regeln R3_VorgPassiv und R4_VorgPassiv werden hingegen Instanzen identifiziert, bei welchen das Vorgangspassiv in einem Nebensatz ohne Verwendung eines Modalverbs verwendet wird und das Agens dabei ungenannt bleibt. Die Regeln R5_VorgPassiv bis R8_VorgPassiv identifizieren Instanzen von Vorgangspassivkonstruktionen, die in einem Haupt- oder Nebensatz vorliegen, und in denen das Passiv in Verbindung mit einem Modalverb verwendet wird.

Für jede Klasse sind zwei jeweils Regeln angegeben. Durch die Regeln mit ungerader Zahl im Regelnamen werden Instanzen von Vorgangspassivsätzen abgefangen, in welchen zwar eine *vor-* oder eine *durch-* Phrase vorhanden ist, diese aber nicht das Agens des Passivsatzes bildet. Diese Eigenschaft kann anhand der annotierten Parsing-Information überprüft werden. Wenn das Regens der *von-* oder *durch-*Phrase das Partizip ist, handelt es sich um eine Agensangabe. Wenn das Regens hingegen nicht das Partizip ist, handelt es sich um keine Agensangabe. Durch die Prüfung, ob die *von-* oder *durch-*Phrase ein Dependens des Partizips ist, können die agenslosen Sätze gefunden werden. Die Regelvarianten mit einer geraden Zahl suchen nach Instanzen von Vorgangspassivkonstruktionen, bei welchen überhaupt keine *von-* oder *durch-*Phrase vorhanden ist.

8.3.3 Auswertung ausgewählter Passivsätze ohne Agens nach Kritikalität

In Abschnitt 8.3.2 wurde bereits beschrieben, dass in den untersuchten Korpora zwar viele Instanzen von Passivsätzen vorhanden sind, diese jedoch nur in sehr seltenen Fällen dazu führen, dass das fehlende Agens nicht durch Kontextwissen hergeleitet werden kann. In diesem Abschnitt wird deshalb keine Evaluierung der entwickelten Regeln diskutiert, sondern es wird anhand von Strichproben eine Übersicht darüber gegeben, wie sich das Verhältnis zwischen kritischen und unkritischen Instanzen agensloser Passivsätze darstellt.

In einer kleinen Untersuchung wurden 150 zufällig ausgewählte Passiv-Anforderungen aus dem Entwicklungskorpus ausgewählt, bei welchen das Agens nicht durch eine Präpositionalphrase genannt wird. Die Anforderungen wurden zunächst danach bewertet, ob das Agens ohne Kontextwissen hergeleitet werden kann. Ohne Kontextwissen bedeutet, dass die Anforderungen isoliert von den weiteren Anforderungen betrachtet wurden und somit nur die Information in der betrachteten Anforderungen und Wissen zum Diskurskontext für die Interpretation zur Verfügung steht. Bei einer zweiten Analyse wurden die Anforderungen mit Einbezug von Kontextwissen betrachtet, d.h. die umgebenden Anforderungen wurden bei der Bewertung miteinbezogen.

Ohne Einbezug von Kontextwissen konnte bei 111 Anforderungen das Agens hergeleitet werden. Eine solche Anforderung, bei der dies der Fall ist, ist in Beispiel 93 zu sehen. Das Agens des Verbs *abklären* wird in der Anforderung nicht genannt. Mit dem Wissen, dass den betrachteten Lastenheften eine Auftraggeber-Auftragnehmer-Kommunikation zugrunde liegt, kann allerdings hergeleitet werden, dass der Auftragnehmer das Agens des Satzes ist. Weiterhin konnte mit dem Wissen, dass es bei den in den Anforderungen beschriebenen Funktionalitäten um Funktionen der zu entwickelnden Komponente geht, ebenfalls in vielen Fällen das Agens hergeleitet werden.

Beispiel 93 *Die Ausführungen müssen mit dem Auftraggeber abgeklärt werden.*

In 39 der 150 Fälle (= 26% der Stichprobe) konnte die fehlende Information nicht hergeleitet werden. Eine Anforderung, bei der dies der Fall ist, ist in Beispiel 94 aufgeführt.

Beispiel 94 *Die Ansteuerung des Systems kann nur gestartet werden, wenn die folgenden Vorbedingungen erfüllt sind: [...]*

Ohne Einbezug von Kontextwissen ist nicht eindeutig, wer oder was den Start der Ansteuerung durchführt. Das Wissen um die Auftraggeber-Auftragnehmer-Beziehung und das Wissen um die zu entwickelnde Komponente hilft in diesem Falle nicht, das Agens der Anforderung zu ermitteln.

Mit Einbezug von Kontextinformation gestaltet sich das Verhältnis anders: in Kombination mit den umgebenden Anforderungen, konnte bei jeder Anforderung das Agens hergeleitet werden. Es war stets genug Information in den umgebenden Anforderungen oder in beigefügten Tabellen vorhanden, um diese Information zu erhalten.

Die Ergebnisse zeigen, dass die Nicht-Nennung des Agens in Anforderungen zu Problemen führen kann, wenn die Anforderungen isoliert von weiteren Anforderungen betrachtet werden. Mit Einbezug von Kontextwissen stellt das Weglassen des Agens kaum ein Problem dar. Jedoch kann nicht generell behauptet werden, dass das Passiv nie ein Problem darstellt. Es kann durchaus der Fall auftreten, dass bei einem Passivsatz ohne Agens selbst mit Kontextwissen die fehlende Information nicht hergeleitet werden kann. Zusammenfassend lässt sich jedoch sagen, dass der Anteil der wirklich kritischen Instanzen sehr gering ist. Ein ähnliches Verhältnis zeigt sich bei den englischsprachigen Lastenheften (vgl. Abschnitt 8.6).

8.3.4 Zusammenfassung

Bei der Verwendung von Passiv muss die semantische Rolle Agens nicht zwingend angegeben werden, d.h. der Handelnde in einem Satz kann bewusst weggelassen werden. Dies kann zu Problemen führen, wenn das Agens nicht durch Kontextinformation vom Leser hergeleitet werden kann. Die in dieser Arbeit durchgeführten Analysen zeigen, dass Passivsätze problematisch sein können, wenn sie isoliert betrachtet werden, ohne Einbezug von Wissen aus dem umgebenden Text. Jedoch ist zu beachten, dass der Leser im Normalfall Zugriff auf alle Anforderungen hat und somit Kontextinformation in die Interpretation einbeziehen kann. Die Verwendung von Passiv ist daher in den meisten Fällen als unkritisch zu beurteilen.

Es wurden Regeln für die automatische Identifikation von agenslosen Passivsätzen und Passiversatzkonstruktionen präsentiert. Diese Regeln können zwar nicht zuverlässig zwischen den kritischen und den unkritischen Instanzen unterscheiden, jedoch können sie dabei helfen agenslose Passivsätze zu identifizieren. Wenn es das Ziel ist, dass die Autoren stets im Aktiv schreiben, stellen die entwickelten Regeln eine solide Analysebasis dar. So kann sichergestellt werden, dass das Agens in Anforderungen stets genannt wird und somit gar nicht erst Interpretationsspielraum entsteht.

8.4 Das Modalverb *sollte(n)*

Das Phänomen *Modalverb „sollte(n)“* ist im Qualitätsmodell dem Kriterium *Verbindlichkeit* untergeordnet. Werden Formen wie *sollte* oder *sollten* innerhalb einer Anforderung verwendet, kann dies dazu führen, dass der Grad der rechtlichen Verbindlichkeit nicht

eindeutig erkennbar ist. Bei einer automatischen Analyse müssen die kritischen Instanzen identifiziert und dem Autor zur Korrektur aufgezeigt werden. In den folgenden Abschnitten soll nun genauer auf dieses Phänomen eingegangen werden²¹.

8.4.1 Phänomenbeschreibung

Die Angabe der rechtlichen Verbindlichkeit in einer Anforderung ist im Requirements Engineering von großer Wichtigkeit. Sie gibt Auskunft darüber, ob der Auftragnehmer die beschriebene Funktionalität realisieren muss, ob es sich um eine Zusatzfunktion handelt, die er freiwillig realisieren kann, oder ob es sich bei dem zur Rede stehenden Textstück, dem Satz oder dem Teilsatz einer Anforderung lediglich um eine Zusatzinformation handelt. Nur wenn eine Anforderung bindend formuliert ist, steht der Auftragnehmer in der Verantwortung die beschriebenen Funktionen zu realisieren. Aus diesem Grund ist es wichtig zu prüfen, ob die Anforderungen entsprechend durch Schlüsselwörter markiert sind. Rechtlich bindende Anforderungen müssen das Lemma *müssen* enthalten. Die Verneinung wird durch *nicht dürfen* angegeben. Optionale Anforderungen werden mit dem Wort *sollte(n)* markiert.

Bei der Durchsicht der Anforderungen wurde ersichtlich, dass das Wort *sollte* in den untersuchten Korpora häufig verwendet wird. Im Entwicklungskorpus wurden die Konjunktivformen *sollte* und *sollten* zusammen insgesamt 359 mal verwendet. Bei genauer Prüfung wird aber außerdem deutlich, dass viele Anforderungen in Wirklichkeit bindend gemeint sind, jedoch nicht das Lemma *müssen* oder *nicht dürfen* enthalten. Dies soll anhand von Beispiel 95 illustriert werden.

Beispiel 95 *Fangbänder sollten nicht im Gesichtsbereich angeordnet sein.*

Beispiel 95 zeigt, dass der Konjunktiv *sollte* in manchen Fällen von Lastenheftautoren auch dann verwendet wird, wenn eine Anforderung bindend gemeint ist. Da es sich bei der Anforderung in Beispiel 95 um eine sicherheitskritische Anforderung handelt, kann die Anforderung nicht optional sein. Die Vermutung ist hierbei, dass die Autoren das Wort *sollte* verwenden, weil es ihnen höflicher erscheint als das fordernde *muss*.

Bei einer automatisierten Analyse des Wortes *sollte(n)* soll allerdings nicht jede Instanz von *sollte(n)* an den Benutzer zur Überprüfung gegeben werden. Man muss nach Fällen unterscheiden, bei welchen das Wort einen Einfluss auf die Verbindlichkeit hat und nach Fällen, bei welchen das Wort in dieser Hinsicht unkritisch ist. Bei den unkritischen Fällen liegt die sogenannte Potentialislesart vor. In diesen Fällen funktioniert *sollte(n)* wie die Konstruktion *wenn ... ist*. Kritisch hingegen sind die Verwendungen von *sollte(n)*, bei

²¹Die Arbeitsweise zur Analyse des Modalverbs *sollte* wurde in [KDJH16] und [Kri15] publiziert.

welchen allein aus der Formulierung und dem (grammatischen) Kontext im Satz nicht entschieden werden kann, ob es sich um eine bindende Anforderung handelt. In diesen Fällen liegt eine Mehrdeutigkeit zwischen zwei Lesarten vor, zwischen welchen sich der Leser entscheiden muss. Eine Lesart, und somit eine mögliche Interpretation, die der Leser treffen kann, ist die deontische Lesart. Hierbei entspricht *sollte* einer “*müssen-Interpretation*” [Hol10], d.h. die Anforderung muss vom Auftragnehmer realisiert werden. Bei der zweiten Lesart handelt es sich um die dispositionelle Lesart. Diese entspricht der “*können-Lesart*” [Hol10] und bedeutet, dass die Realisierung des beschriebenen Sachverhaltes in der Anforderung optional ist.

Die Fälle von *sollte(n)*, bei welchen die Potentialislesart vorliegt, müssen dem Autor nicht zur Kontrolle aufgezeigt werden, da sie den Grad der Verbindlichkeit nicht beeinflussen. Instanzen von *sollte(n)*, bei welchen der Leser die Wahl hat zwischen der deontischen und der dispositionellen Interpretation und somit die Intention des Autors nicht eindeutig ist, müssen dem Autor zur Kontrolle aufgezeigt werden.

Die drei genannten Lesarten von *sollte(n)* werden nun im Folgenden detaillierter ausgeführt, und es werden Beispiele diskutiert.

Die Potentialislesart liegt vor, wenn *sollte(n)* in einen Konditionalsatz (=Bedingungssatz) eingebettet ist. „Mit einem Konditionalsatz wird behauptet, dass ein bestimmter Sachverhalt unter der Bedingung eintritt, dass ein bestimmter anderer Sachverhalt besteht oder eintritt“ [Eis13]. Konditionalsätze bestehen aus einem Antezedens, welches „den bedingenden Sachverhalt bezeichnet“ [Eis13] und einem Konsequens, in welchem der bedingte Sachverhalt dargestellt wird.

In [Eis13] wird zwischen drei Arten von Konditionalsätzen unterschieden, welche sich in der Form des Antezedens unterscheiden: Das Antezedens kann beispielsweise ein eingeleiteter Stirnsatz sein wie *Gewinnt Anna* in Beispiel 96, eine Partizipialgruppe wie *Angenommen Anna gewinnt* in Beispiel 97 oder ein Konjunktionalsatz wie *Wenn Anna gewinnt* in Beispiel 98.

Beispiel 96 *Gewinnt Anna, verliere ich.*

Beispiel 97 *Angenommen Anna gewinnt, dann verliere ich.*

Beispiel 98 *Wenn Anna gewinnt, verliere ich.*

Die in dieser Arbeit diskutierten Fälle von *sollte(n)*, bei welchen die Potentialislesart vorliegt, funktionieren wie die Konstruktion in Beispiel 96, bei welchen das Antezedens ein Stirnsatz (= Verb-Erst-Satz) ist.

In Beispiel 99 ist eine Anforderung aus dem Entwicklungskorpus zu sehen, welche das Modalverb *Sollte* in der Potentialislesart enthält.

Beispiel 99 *Sollte durch das Steuergerät eine PRE-Situation erkannt werden, so sind interne und externe Daten im Steuergerät abzuspeichern.*

In Beispiel 99 leitet das Wort *Sollte* einen Konditionalsatz ein. Das Antezedens *Sollte durch das Steuergerät eine PRE-Situation erkannt werden* kann in diesem Fall durch einen Konjunktionalsatz mit *Wenn ... wird* ersetzt werden, und die Bedeutung des Satzes wird dabei nicht geändert. In Beispiel 99 liegt somit die Potentialislesart vor, die bei einer automatischen Analyse dem Autor nicht aufgezeigt werden muss. Im Hinblick auf die Verbindlichkeit ist diese Verwendung von *sollte(n)* unkritisch.

Weitere Fälle der Potentialislesart sind in den Beispielen 100 und 101 zu sehen.

Beispiel 100 *Für den Fall, dass ein sofortiges Anhalten des Wischers während eines Wischzyklus notwendig sein sollte, werden beide Wischerrelais sofort abgeschaltet.*

Beispiel 101 *Falls es hinsichtlich der lichttechnischen Anforderungen notwendig sein sollte, müssen die sichtbaren Flächen der Leuchte mit Grundlack lackiert werden.*

In Beispiel 100 ist ein Konditionalsatz, welcher mit der Mehrwort-Konjunktion *Für den Fall, dass* eingeleitet wird, zu sehen. In diesem Fall kann der Konjunktiv *sollte* durch die Indikativform *ist* ersetzt werden, ohne dass die Bedeutung des Satzes sich ändert. In Beispiel 101 (Konjunktion *Falls*) kann ebenfalls der Konjunktiv durch die Indikativform ersetzt werden, und die Bedeutung bleibt erhalten.

Die drei genannten Potentialis-Fälle müssen bei einer automatisierten Konjunktivanalyse von der Korrekturfeedbackausgabe ausgeschlossen werden, d.h. sie dürfen dem Autor nicht aufgezeigt werden. Anders verhält es sich bei Fällen wie in den Beispielen 102, 103 und 104, bei welchen der Leser zwischen der deontischen (*müssen*) und dispositionellen (*können*) Lesart von *sollte(n)* disambiguieren muss.

Beispiel 102 *Die zu erreichende Geräuschqualität des aktiven Gurtschlösses sollte auf Basis realer Fahrzeugmessungen definiert werden.*

Beispiel 103 *Es ist zu beachten, dass die Überprüfungen dieselben Verfahren benutzen sollten.*

Beispiel 104 *Die Drückkontur der Abdeckung muss möglichst großflächig ausgeführt werden und sollte nicht über die EZS-Außenkontur überstehen.*

In Beispiel 102 ist *sollte* des Hauptverb des Satzes. Diese Verwendung von *sollte* ist immer als kritisch anzusehen und muss dem Autor zur Kontrolle aufgezeigt werden, da der Leser in solchen Fällen sonst die Wahl zwischen der deontischen oder der dispositionellen Interpretation hat. In Beispiel 103 hat das Wort *sollte* die Funktion eines Nebensatzprädikates. Auch hier liegen die beiden Lesarten deontisch und dispositionell vor. Diese Fälle sind aus diesem Grund ebenfalls als kritisch anzusehen. In Beispiel 104 wird *sollte* durch die Konjunktion *und* mit dem Hauptverb *muss* koordiniert. Auch diese Fälle müssen dem Autor zur Kontrolle aufgezeigt werden, da trotz der Koordination mit *müssen* lokal zwischen deontischer und dispositioneller Lesart disambiguiert werden muss.

Fälle, bei welchen eine Interpretation zwischen deontischer und dispositioneller Lesart getroffen werden muss, müssen bei einer automatischen Analyse identifiziert und dem Autor zur Kontrolle vorgezeigt werden. Die Potentialis-Fälle sollen hierbei ausgeschlossen werden, da sie keinen Einfluss auf den Grad der Verbindlichkeit haben.

8.4.2 Regeln

Im Folgenden wird nun beschrieben, anhand welcher Regeln die kritischen Instanzen von *sollte* (deontische vs. dispositionelle Lesart) automatisiert identifiziert werden können. Es handelt sich um Regeln, die ausschließlich vom Typ *Effect* sind (vgl. Kapitel 7), d.h. dem Autor wird nicht mitgeteilt, dass er einen Fehler gemacht hat, sondern er wird dazu aufgefordert zu prüfen, ob die gefundenen Anforderungen wirklich optional sind (vgl. hierzu Abschnitt 9.1). Wenn eine Anforderung rechtlich bindend sein soll, soll der Autor das Wort *sollte* durch *muss* ersetzen. Falls die Anforderung wirklich optional ist, muss der Autor keine Änderung vornehmen. Die entwickelten Regeln sind in Tabelle 31 in CQP-Code abgebildet. Jede Regel erhält außerdem einen eindeutigen Regelnamen, welcher ebenfalls in Tabelle 31 abgebildet ist.

Regelname	Regel
R1_sollte	[word="sollten?" & dep="0"];
R2_sollte	[word!="Fall"] [word=","] a:[pos="KOUS" & word="dass"] []* b:[word="sollten?"] :: a.dep=b.wortnr within s;
R3_sollte	[word="sollten?" & parse="CJ"]

Tabelle 31: CQP-Regeln für die Identifikation kritischer Verwendungen von *sollte(n)*

R1_sollte sucht nach Instanzen, die aus dem Wort *sollte* oder *sollten* bestehen und zusätzlich das Hauptverb des Satzes bilden. Die Regel greift auf Wissen aus der Dependenzanalyse der zu betrachtenden Sätze zurück: Wenn der *mate*-Parser die Annotation [dep="0"]; liefert, so ist im Analyseergebnis die so gekennzeichnete Verbform als Hauptverb eines Satzes klassifiziert. Ein Beispiel für eine solche Instanz ist in Abbildung 38 zu sehen, welche dem Beispiel 102 entspricht.

1	2	3	4	5	6	7
1_1_AGS-38	Die	der	ART	case=nom number=sg gender=fem	4	NK
1_2_AGS-38	zu	zu	PTKZU	-	3	PM
1_3_AGS-38	erreichende	erreichend	ADJA	case=nom number=sg gender=fem degree=pos	4	NK
1_4_AGS-38	Geräuschqualität	Geräuschqualität	NN	case=nom number=sg gender=fem	8	SB
1_5_AGS-38	des	der	ART	case=gen number=sg gender=neut	7	NK
1_6_AGS-38	aktiven	aktiv	ADJA	case=gen number=sg gender=neut degree=pos	7	NK
1_7_AGS-38	Gurtschlosses	Gurtschloß	NN	case=gen number=sg gender=neut	4	AG
1_8_AGS-38	sollte	sollen	VMFIN	number=sg person=3 tense=pres mood=ind	0	-
1_9_AGS-38	auf	auf	APPR	-	11	MO
1_10_AGS-38	Basis	Basis	NN	case=dat number=sg gender=fem	9	NK
1_11_AGS-38	realer	real	ADJA	case=gen number=pl gender=fem degree=pos	12	NK
1_12_AGS-38	Fahrzeugmessungen	Fahrzeugmessung	NN	case=gen number=pl gender=fem	10	AG
1_13_AGS-38	definiert	definieren	VVPP	-	14	OC
1_14_AGS-38	werden	werden	VAINF	-	8	OC
1_15_AGS-38	.	--	\$.	-	14	--

Abbildung 38: *sollte* bildet das Hauptverb des Satzes

In Abbildung 38 kann man sehen, dass dem Verb *sollte* das Parse-Tag 0 annotiert wurde (Spalte 6, Zeile 8). R1_sollte greift an dieser Stelle und somit kann die Anforderung dem Benutzer zur Kontrolle aufgezeigt werden. Er kann dann entscheiden, ob er *sollte* durch *muss* ersetzen möchte oder nicht.

Mithilfe von R2_sollte werden Instanzen von *sollte* gesucht, die finite Nebensatzverben bilden und gleichzeitig das syntaktische Regens einer subordinierenden Konjunktion bilden. R2_sollte deckt Fälle wie in Beispiel 103 ab. Die Regel bezieht zusätzlich zu Wortformen auch POS-Annotationen und Parsing-Annotationen ein. Die Regel gleicht ab, ob die Wortnummer von *sollte* der annotierten Dependenz der subordinierenden Konjunktion entspricht. Aussortiert werden durch die ersten beiden Angaben in der Regel Sätze wie in Beispiel 100, die nicht angemahnt werden sollen, weil sie der Potentialislesart entsprechen. Wie man in Abbildung 39 sehen kann, entspricht die Wortnummer von *sollten* (= 12), welche zusammen mit der ID in der ersten Spalte steht, der syntaktischen Dependenz von

dass, welche in der 6. Spalte dargestellt ist. `R2_sollte` greift bei Anforderungen dieser Art und liefert dem Autor die Anforderung zur Kontrolle zurück.

1	2	3	4	5	6	7
1_1_AGS-42	Es	es	PPER	case=nom number=sg gender=neut person=3	2	SB
1_2_AGS-42	ist	sein	VAFIN	number=sg person=3 tense=pres mood=ind	0	--
1_3_AGS-42	zu	zu	PTKZU	--	4	PM
1_4_AGS-42	beachten	beachten	VVINF	--	2	OC
1_5_AGS-42	,	--	\$.	--	4	--
1_6_AGS-42	dass	dass	KOUS	--	12	CP
1_7_AGS-42	die	der	ART	case=nom number=pl gender=fem	8	NK
1_8_AGS-42	Überprüfungen	Überprüfung	NN	case=nom number=pl gender=fem	12	SB
1_9_AGS-42	dieselben	derselbe	PDAT	case=acc number=pl gender=neut	10	NK
1_10_AGS-42	Verfahren	Verfahren	NN	case=acc number=pl gender=neut	11	OA
1_11_AGS-42	benutzen	benutzen	VVINF	--	12	OC
1_12_AGS-42	sollten	sollen	VMFIN	number=pl person=3 tense=past mood=ind	1	RE
1_13_AGS-42	.	--	\$.	--	12	--

Abbildung 39: *sollte* ist Regens einer subordinierenden Konjunktion

`R3_sollte` identifiziert Instanzen von *sollte*, die ein Konjunkt des Hauptverbes bilden. Die Anforderung aus Beispiel 104 ist in Abbildung 40 dargestellt. Koordinierte Hauptverben werden vom *mate*-Parser anhand einer Ketten-Repräsentation (= “chain representation”) dargestellt [KMN09] [TNA11]. Das heißt, die beiden Verben *muss* und *sollte* bilden mit der Konjunktion (in diesem Fall *und*) eine Dependenzkette. Das Hauptverb *muss* ist das erste Glied der Kette und somit der Kopf. Die Konjunktion ist der Kopf des zweiten Hauptverbs (vgl. Spalte 6, Zeile 11). Zusätzlich wird die Funktion des zweiten Hauptverbes durch das Parse-Tag *CJ* (= Konjunkt) angegeben. `R3_sollte` sucht nach solchen Dependenzketten und liefert die Treffer an den Autor zur Kontrolle zurück.

8.4.3 Evaluierung der Regeln

Die Regeln für die automatische Identifikation problematischer Verwendungen von *sollte* wurden anhand des Entwicklungskorpus (vgl. Abschnitt 6.4) entwickelt. Darauf folgend wurden die Regeln zunächst am Entwicklungskorpus und anschließend am Evaluierungskorpus evaluiert.

Die Evaluierungsergebnisse am Entwicklungskorpus sind in Abbildung 41 und Tabelle 32 zu sehen. Insgesamt kommen die Konjunktivformen *sollte* und *sollten* 359 mal im Entwicklungskorpus vor. Durch Anwendung der entwickelten Regeln wurden 123 Instanzen von *sollte*(*n*) korrekt als kritisch klassifiziert und dem Autor zur Kontrolle vorgezeigt. 233 Instanzen von *sollte*(*n*) wurden korrekt als unkritisch klassifiziert und dem Autor nicht

1	2	3	4	5	6	7
1_1_AGS-40	Die	der	ART	case=nom number=sg gender=fem	2	NK
1_2_AGS-40	Drückkontur	Drückkontur	NN	case=nom number=sg gender=fem	5	SB
1_3_AGS-40	der	der	ART	case=gen number=sg gender=fem	4	NK
1_4_AGS-40	Abdeckung	Abdeckung	NN	case=gen number=sg gender=fem	2	AG
1_5_AGS-40	muss	muss	VVFIN	number=sg person=3 tense=past mood=ind	0	_
1_6_AGS-40	möglichst	möglichst	ADV	-	7	MO
1_7_AGS-40	großflächig	großflächig	ADJD	degree=pos	8	MO
1_8_AGS-40	ausgeführt	ausführen	VVPP	-	9	OC
1_9_AGS-40	werden	werden	VAINF	-	5	OC
1_10_AGS-40	und	und	KON	-	5	CD
1_11_AGS-40	sollte	sollen	VMFIN	number=sg person=3 tense=past mood=ind	10	CJ
1_12_AGS-40	nicht	nicht	PTKNEG	-	11	NG
1_13_AGS-40	über	über	APPR	-	16	MO
1_14_AGS-40	die	der	ART	case=acc number=sg gender=fem	15	NK
1_15_AGS-40	EZS-Außenkontur	EZS-Außenkontur	NN	case=acc number=sg gender=fem	13	NK
1_16_AGS-40	überstehen	überstehen	VVFIN	number=pl person=3 tense=pres mood=ind	11	OC
1_17_AGS-40	.	--	\$.	-	16	--

Abbildung 40: *sollte* ist Konjunkt einer Koordination von Verbformen

aufgezeigt. Insgesamt wurden keine Fehlwarnungen produziert, jedoch wurden dem Autor drei kritische Anforderungen vorenthalten. Auf den Grund dafür, wieso dreimal keine Warnung an den Autor ausgegeben wurde, wird später in diesem Abschnitt eingegangen.

Ein wortlistenbasierter Ansatz würde insgesamt 233 Fehlwarnungen liefern, was 64,9% aller Warnungen entsprechen würde. Dieser Wert ist sehr hoch und würde bei einem Einsatz in der Praxis nicht auf Akzeptanz der Nutzer stoßen (vgl. Kapitel 7). Der hier präsentierte regelbasierte Ansatz liefert auf dem Entwicklungskorpus keine Fehlwarnungen, wodurch die Akzeptanz und Benutzbarkeit des Werkzeugs gesteigert wird.

		Automatische Analyse →	
		Kritische Anforderung	Unkritische Anforderung
Manuelle Analyse ↓	Kritische Anforderung	123	3
	Unkritische Anforderung	0	233

Abbildung 41: Evaluierung der Regeln für *sollte* am Entwicklungskorpus, N=359

Die Precision- Recall- und F-Measure-Werte liegen in einem sehr hohen Bereich (vgl. Tabelle 32). Dies bedeutet, dass die Regeln auf dem Entwicklungskorpus gute Ergebnisse liefern.

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	100%	98,73%
Recall	97,62%	100%
F-Measure	98,80%	99,36%

Tabelle 32: Precision-, Recall- und F-Measure-Werte der Regeln für *sollte* am Entwicklungskorpus, N=359

In Abbildung 42 und Tabelle 33 sind die Evaluierungsergebnisse der entwickelten Regeln am Evaluierungskorpus zu sehen. Die Precision-, Recall- und F-Measure-Werte weichen nur geringfügig von den Werten der Evaluierung am Entwicklungskorpus ab. Bei der Anwendung der Regeln auf das Evaluierungskorpus wurden vier Fehlwarnungen an den Autor ausgegeben, und eine Warnung wurde dem Autor fälschlicherweise vorenthalten. 155 kritische Instanzen von *sollte*(n) wurden korrekt als solche identifiziert und 168 unkritische Instanzen wurden ebenfalls korrekt klassifiziert. Eine Wortlistensuche würde 52,43% Fehlwarnungen liefern, der entwickelte Ansatz hingegen liefert 2,33% Fehlwarnungen. Ein Einsatz in der Praxis lohnt sich hierbei, da der Ansatz eine geringe Anzahl an Fehlwarnungen und Fehlentwarnungen liefert.

		Automatische Analyse	
		Kritische Anforderung	Unkritische Anforderung
Manuelle Analyse	Kritische Anforderung	155	1
	Unkritische Anforderung	4	168

Abbildung 42: Evaluierung der Regeln für *sollte* am Evaluierungskorpus, N=328

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	97,48%	99,41%
Recall	99,36%	97,67%
F-Measure	98,41%	98,53%

Tabelle 33: Precision-, Recall- und F-Measure-Werte der Regeln für *sollte* am Evaluierungskorpus, N=328

Die produzierten Fehlwarnungen entstehen alle durch vom Parser nicht erkannte Hauptsätze. Dies hat unterschiedliche Gründe, welche im Folgenden genauer ausgeführt werden sollen.

Eine Fehlentwarnung entstand durch einen Klammereinschub, in welchem ein eigenständiger Hauptsatz vorhanden war (vgl. Beispiel 105); *sollte* stellt das Hauptverb dieses eingeschobenen Satzes dar. Der Parser kann allerdings nur einem Verb die Funktion des

Hauptverbs zuweisen, was in diesem Fall das Wort *ist* des Satzes außerhalb der Klammer ist. Würde der Satz außerhalb der Klammer einen eigenständigen Satz bilden, würde `R1_sollte` greifen, und eine Warnung an den Autor würde ausgegeben werden. In diesem Fall kann die Regel aber nicht greifen, da das Verb *ist* als Hauptverb des gesamten Satzes analysiert wird.

Beispiel 105 *Die Kofferrauminnenabfrage ist weniger zeitkritisch als eine Abfrage zum Entsichern (Entsichern sollte zeitlichen Vorrang haben).*

Eine weitere Fehlentwarnung entstand durch eine Aneinanderreihung zweier Hauptsätze innerhalb eines Satzes. Die betroffene Anforderung ist in Beispiel 106 zu sehen. In dieser Anforderung wurden zwei Hauptsätze aneinandergereiht und durch ein Komma koordiniert. Der Parser kann aber nur ein Verb als Hauptverb analysieren und wählt in diesem Fall korrekterweise *beziehen*. Wären die beiden Hauptsätze durch einen Punkt voneinander getrennt, würde `R1_sollte` auch hier greifen, und eine Warnung würde an den Benutzer ausgegeben werden.

Beispiel 106 *Die aufgeführten Kleinstsprünge beziehen sich auf den DK Bereich mit 12-Bit Auflösung, auch hier sollten sich keine Hysterese Effekte zeigen.*

Eine weitere Fehlentwarnung entstand durch einen Fehler des Tokenizers. Die Anforderung ist in Beispiel 107 dargestellt. Der Tokenizer hat fälschlicherweise nach dem Doppelpunkt keine Satzgrenze eingefügt. Auch hier resultiert daraus das Problem, dass `R1_sollte` nicht greifen kann. Das Verb *wird* wird daher vom Parser als das Hauptverb des Satzes kategorisiert. Wäre der Satz korrekt tokenisiert worden, würde auch hier `R1_sollte` greifen und das Werkzeug würde dem Autor die Anforderung zur Kontrolle anzeigen.

Beispiel 107 *Eine Übersichtsliste über durchgeführte Tests und Status wird übergeben: Diese Übersicht sollte in tabellarischer Form dargestellt sein.*

Die letzte Fehlentwarnung entstand durch einen Fehler des Parsers. Bei Beispiel 108 sollte im Normalfall `R3_sollte` greifen. Dem Konjunktiv *sollte* wurde jedoch fälschlicherweise die grammatische Funktion Objektsatz (= *OC*) annotiert. Korrekt wäre die Satzfunktion Konjunkt (= *CJ*). Durch diesen Parsingfehler kann `R3_sollte` nicht greifen und die Warnung an den Autor wird fälschlicherweise nicht ausgegeben.

Beispiel 108 *Jedoch muss die EM "gleichmäßig" laufen, das heißt die eingestellte Drehzahl sollte konstant bleiben, bis der Motor läuft und die Synchronisation im Ersatzbetrieb auf NW Sensor erfolgt ist.*

Alle Fehlentwarnungen entstanden aufgrund von nicht erkannten Hauptsätzen. Dies liegt zum einen an Fehlern des Tokenizers und des Parsers, zum anderen aber auch an Klammereinschüben mit ganzen Sätzen und an durch Kommata koordinierte Hauptsätze.

Die Fehlwarnungen, die bei der Analyse entstanden sind, resultieren aus nicht erkannten Konditionalsätzen.

Die Anforderung in Beispiel 109 beinhaltet einen eingeschobenen Konditionalsatz (*sollte er der Meinung sein ...*). Durch *sollte* eingeleitete Konditionalsätze sollen dem Autor nicht aufgezeigt werden, weil hier die Potentialislesart vorliegt und keine Ambiguität entsteht zwischen den beiden Lesarten deontisch und dispositionell. Da der Parser aber dem Wort *sollte* die Funktion des Hauptverbs zugewiesen hat und somit `R1_sollte` greift, wird fälschlicherweise eine Warnung an den Autor ausgegeben.

Beispiel 109 *Der Auftragnehmer muss, sollte er der Meinung sein die X-Anforderungen gemäß mitgeltender Unterlagen auch mit einer 2-fach-Leiterplatte erfüllen zu können, dies im Angebot als Option vorsehen.*

Die weiteren drei Fehlwarnungen resultieren aus Aneinanderreihungen von Sätzen, wobei jeweils der zweite Satz mit einem eigenständigen Konditionalsatz beginnt (vgl. Beispiel 110). Der Parser annotiert dem Modalverb *sollte* die Funktion des Hauptverbs. Aufgrund dieser Annotation greift `R1_sollte`, wodurch die Fehlwarnung entsteht.

Beispiel 110 *Die Toleranzen der Einzelteile betragen +/- 1mm, sollte dies aus erklärbaren Gründen nicht umsetzbar sein, ist die Tolerierung der Einzelteile gemäß der gültigen MB Normen durchzuführen.*

Da die entwickelten Regeln sowohl auf dem Entwicklungskorpus als auch auf dem Evaluierungskorpus gute Ergebnisse liefern (wenig Fehl- und Fehlentwarnungen), kann die Aussage getroffen werden, dass die Regeln solide sind und zur Anwendung in der Praxis gebracht werden können.

8.4.4 Zusammenfassung

Das Modalverb *sollte(n)* gibt nicht klar an, ob eine Anforderung rechtlich bindend ist, da oft zwischen deontischer und dispositioneller Lesart disambiguiert werden muss. Allerdings hat nicht jede Verwendung von *sollte(n)* einen Einfluss auf den Grad der Verbindlichkeit einer Anforderung, da Potentialislesarten von *sollte(n)* diesbezüglich unkritisch sind. Es wurden Regeln entwickelt, welche nur die kritischen Instanzen (d.h. die Fälle, bei welchen zwischen deontischer und dispositioneller Lesart disambiguiert werden muss) von *sollte* automatisiert identifizieren und dem Autor zur Kontrolle vorzeigen. Die entwickelten Regeln liefern sowohl am Entwicklungskorpus als auch am Evaluierungskorpus gute

Ergebnisse. Die Anzahl der Fehlwarnungen und Fehlentwarnungen ist sehr gering. Dies bedeutet, dass die Regeln produktiv eingesetzt werden können und die Anwender nicht mit zu viel Mehraufwand bei der Durchsicht der Warnungen belastet werden.

8.5 Pronominale Bezüge

Ambige pronominale Bezüge sind im Qualitätsmodell der Kategorie der satzsequentiel- len Mehrdeutigkeiten zugeordnet. Hat ein Pronomen eine Form, die in Bezug auf Genus, Numerus und Person mit zwei verschiedenen Nomina kongruent ist, kann der pronomina- le Bezug grammatikalisch nicht eindeutig einem Antezedenten zugeordnet werden. Dies kann einen Einfluss auf die Verständlichkeit und somit auf die Testbarkeit einer Anfor- derung haben. Ziel ist es daher, solche kritische Instanzen von pronominalen Bezügen zu identifizieren.

In Abschnitt 8.5.1 soll das Phänomen der pronominalen Bezüge zunächst allgemein be- schrieben werden. Anschließend werden Regeln für die automatische Identifikation kriti- scher Instanzen präsentiert (vgl. Abschnitt 8.5.2). Da die in den untersuchten Korpora grammatikalisch mehrdeutigen pronominalen Bezüge nur selten zu einer semantischen Ambiguität führen, können keine effektiven Regeln für die Identifikation kritischer In- stanzen entwickelt werden. Die in dieser Arbeit präsentierten Regeln dienen lediglich der Identifikation grammatisch mehrdeutiger pronominaler Bezüge. Eine Evaluierung der entwickelten Regeln wird aus diesem Grund nicht durchgeführt. In Abschnitt 8.5.3 soll abschließend ein Überblick darüber gegeben werden, wie sich das Verhältnis zwischen morphosyntaktisch mehrdeutigen pronominalen Bezügen und semantisch mehrdeutigen Bezügen darstellt.

8.5.1 Phänomenbeschreibung

Pronomina wie beispielsweise *er*, *sie*, *dieser*, *jener*, *sein* werden häufig in den untersuchten Lastenheft-Korpora verwendet. „Allen Vertretern gemeinsam ist die semantische Funktion des Verweisens: Im Unterschied zu nominalen Ausdrücken wie Eigennamen, die unabhän- gig von der jeweiligen Redesituation immer dieselben Objekte der außersprachlichen Welt bezeichnen, dienen Pronomen dazu, in Abhängigkeit vom sprachlichen Kontext bzw. der jeweiligen außersprachlichen Realität auf verschiedene Objekte zu referieren [...]“ [Buß08]. Diese Eigenschaft der Pronomina auf verschiedene Objekte in der Realität referieren zu können, führt dazu, dass in bestimmten Situationen Mehrdeutigkeiten in den untersuch- ten Anforderungen entstehen. Kann einem Pronomen nicht eindeutig ein Antezedens, d.h. ein Nomen oder eine Nominalgruppe, auf das es sich bezieht, zugeordnet werden, kann dies dazu führen, dass der Bezug nicht eindeutig aufgelöst werden kann und die Anfor-

derung somit potentiell mehrdeutig ist. Ein Beispiel für einen solchen problematischen pronominalen Bezug ist in Beispiel 111 zu sehen.

Beispiel 111 *Der Blink-Master ist im Außenlicht-Master integriert. Er empfängt alle Signale.*

Der erste Satz in Beispiel 111 beinhaltet zwei Nomina: *Blink-Master* und *Außenlicht-Master*. Diese beiden Nomina weisen dieselben morphologischen Merkmale auf. Sie haben beide dasselbe Genus (Maskulinum) und denselben Numerus (Singular). Das Pronomen *Er* im zweiten Satz besitzt ebenfalls diese Merkmale, d.h. beide Nomina kongruieren morphologisch mit dem Pronomen, und dies bedeutet wiederum, dass es zwei mögliche Antezedenten gibt, auf die sich das Pronomen *Er* potentiell beziehen kann. Da aufgrund der morphosyntaktischen Eigenschaften der Nomina und des Pronomens beide Nomina als potentielle Antezedenten für *Er* in Frage kommen, ist es ohne semantische Information nicht möglich, dem Pronomen *Er* das korrekte Nomen zuzuordnen, auf das es tatsächlich referiert.

Ein weiteres Beispiel eines mehrdeutigen pronominalen Bezugs ist in Beispiel 112 zu sehen. Hier entsteht die Mehrdeutigkeit nicht durch einen satzübergreifenden Bezug wie in Beispiel 111, sondern durch einen satzinternen.

Beispiel 112 *Ist die Prüfspannung kleiner der Bordnetzspannung, so wird sie mittels einer Pulsweitenmodulation begrenzt.*

Auch hier stimmen die beiden Nomina *Prüfspannung* und *Bordnetzspannung* mit dem Pronomen *sie* in Numerus (Singular) und Genus (Femininum) überein. Es ist somit nicht eindeutig, auf welches Nomen das Pronomen referiert.

Kongruiert nur ein Nomen morphologisch mit einem Pronomen, ist die Referenz eindeutig und es entstehen keine Mehrdeutigkeiten auf der Ebene des pronominalen Bezugs. In Beispiel 113 ist ein solcher Fall zu sehen.

Beispiel 113 *Die Motorlager tragen das Eigengewicht des Triebwerkes und definieren so dessen Position im Fahrzeug. Sie stützen Motormomente ab.*

In Beispiel 113 kongruiert lediglich das Nomen *Motorlager* mit dem Pronomen *Sie*, da es das einzige Nomen im unmittelbaren Satzkontext ist, welches im Plural steht.

Nicht nur Personalpronomina können zu ambigen Strukturen führen; weitere Pronomina wie beispielsweise substituierende Demonstrativpronomina besitzen ebenfalls die Eigenschaft der Anaphorik. In Beispiel 114 kongruieren die zwei Nominalgruppen *Signalwert A* und *Signalwert B* morphologisch mit dem Demonstrativpronomen *Dieser*. Auch hier ist

aufgrund der Kongruenz zwischen den beiden Nomina und dem Pronomen nicht eindeutig, auf welches Nomen das Pronomen referiert.

Beispiel 114 *Signalwert A muss kleiner als Signalwert B sein. Dieser tritt vorzugsweise im Leerlauf auf.*

Weitere ambige Referenzen können beispielsweise durch die Verwendung von substituierenden Relativpronomina (vgl. Beispiel 115), attribuierenden Possessivpronomina (vgl. Beispiel 116) und attribuierenden Demonstrativpronomina (in definiten Deskriptionen, vgl. Beispiel 117) entstehen.

Beispiel 115 *Die Ventilsteuerung löst die Steuerungsanfrage der Regelungsaktuatorik aus, die dadurch den Hashwert des Sensors um eins hochsetzt.*

Beispiel 116 *Die Wellen stützen die Getriebestangen aufgrund ihrer Festigkeit.*

Beispiel 117 *Parameter HEA_FB und Parameter IO_PLI müssen gesetzt sein. Dieser Parameter wird auch im nächsten Durchlauf gesetzt.*

Die genannten Beispiele in diesem Abschnitt, ausgenommen Beispiel 113, sind alle hinsichtlich der pronominalen Bezüge morphosyntaktisch mehrdeutig. Jedoch bewirkt ein grammatikalisch mehrdeutiger Bezug nicht gleich, dass auch eine semantische Mehrdeutigkeit vorliegt. Häufig kann durch Experten-, Kontext- oder Weltwissen das richtige Antezedens ermittelt werden. Die durch das Personalpronomen *Sie* entstehende Ambiguität in Beispiel 118 kann beispielsweise durch Experten- und Weltwissen aufgelöst werden.

Beispiel 118 *Diese Prüfung dient der Absicherung einer seriensicheren Systemauslegung. Sie simuliert den Einfluss langer Standzeiten unter Last und hohen Temperaturen.*

Das Personalpronomen *Sie* kongruiert zwar morphologisch mit mehreren Nomina (*Prüfung*, *Absicherung* und *Systemauslegung*) und referiert somit nicht eindeutig auf ein Nomen, jedoch kann der Leser durch Anwendung von Welt- oder Expertenwissen das vom Autor intendierte Antezedens des Pronomens korrekt bestimmen. Durch das Verb *simulieren* wird eindeutig, dass sich das Pronomen *Sie* nur auf *Prüfung* beziehen kann.

Welt-, Kontext- und Expertenwissen kann dabei helfen, bei potentiell mehrdeutigen pronominalen Bezügen die richtige Zuordnung zum intendierten Nomen herzustellen. Da ein morphosyntaktisch mehrdeutiger Bezug nicht grundsätzlich einen semantisch mehrdeutigen Bezug darstellt, müssen bei den pronominalen Bezügen folglich drei Gruppen unterschieden werden: (a) eindeutige Bezüge, (b) morphosyntaktisch mehrdeutige Bezüge und

(c) morphosyntaktisch und semantisch mehrdeutige Bezüge. Regeln für die automatische Identifikation der semantisch mehrdeutigen pronominalen Bezüge können nicht entworfen werden, es können lediglich die grammatikalisch mehrdeutigen Instanzen ermittelt werden.

8.5.2 Regeln

Bei der manuellen Durchsicht der pronominalen Bezüge im Entwicklungskorpus hat sich früh herausgestellt, dass dort eine hohe Anzahl an morphosyntaktisch mehrdeutigen pronominalen Bezügen vorhanden ist, diese jedoch nur in sehr seltenen Fällen dazu führen, dass auch eine semantische Mehrdeutigkeit vorliegt. Es wurde daher früh ersichtlich, dass keine Regeln entworfen werden können, die zuverlässig zwischen den kritischen und den unkritischen Instanzen pronominaler Bezüge unterscheiden können. In den meisten Fällen kann durch Welt-, Kontext- oder Expertenwissen die durch die morphosyntaktische Mehrdeutigkeit potentiell vorhandene semantische Mehrdeutigkeit aufgelöst werden. Trotz der schätzungsweise hohen Anzahl an Fehlwarnungen, die bei einer automatischen Suche nach kritischen Instanzen pronominaler Bezüge entstehen können, wurden CQP-Regeln für die automatisierte Identifikation entwickelt. Diese können zwar die morphosyntaktisch mehrdeutigen Instanzen identifizieren, die semantisch mehrdeutigen Instanzen können jedoch nicht gezielt gefunden werden. Die Entscheidung wann nicht nur eine morphosyntaktische, sondern auch eine semantische Mehrdeutigkeit vorliegt, kann nicht automatisiert getroffen werden. Das für die Identifikation semantisch mehrdeutiger Zuordnungsalternativen nötige Wissen lässt sich nicht ohne sehr großen Aufwand für ein automatisches Prüfsystem modellieren.

Weiterhin wurde bei der manuellen Durchsicht festgestellt, dass nicht für jede Pronomen-Klasse Regeln entwickelt werden können bzw. dass dies nicht bei jeder Klasse zielführend ist. Die Auswahl der behandelten Klassen von Pronomina sowie die Regeln sind im Zusammenhang einer Masterarbeit [War15] entstanden, an die sich die hier präsentierten Regeln und Ergebnisse anlehnen.

Die Auswahl der zu behandelnden Pronomen-Klassen wurde anhand von vier Ausschlusskriterien getroffen: 1) nicht anaphorisch, 2) pronominale Bezüge sind Quantoren, 3) pronominale Bezüge sind Satzanaphern, 4) niedrige Frequenz. Auf diese vier Kriterien wird im Folgenden näher eingegangen.

Das erste Kriterium schließt die Pronomen-Klassen von der Analyse aus, welche nicht anaphorisch gebraucht werden können. Diese Eigenschaft liegt bei den reflexiven Personalpronomen, bei den attributierenden Interrogativpronomen und bei den attributierenden Indefinitpronomen mit Determinator vor.

In das zweite Ausschlusskriterium fallen die Klassen von pronominalen Bezügen, welche ebenfalls in die Kategorie der Quantoren fallen. Dies sind die substituierenden Indefinitpronomina und die attribuierenden Indefinitpronomina ohne Determinator. Bei diesen Pronomen-Klassen ist die Verständlichkeit nicht durch einen ambigen Bezug gefährdet. Wie in Abschnitt 4.2.2.3 beschrieben, liegt bei den Quantoren das Problem vor, dass geprüft werden muss, ob alle Elemente, die durch den Quantor beschrieben sind, auch tatsächlich gemeint sind. Dieses Phänomen wird im Zusammenhang der Pronomina nicht behandelt.

Das dritte Ausschlusskriterium schließt Satzanaphern von der Analyse aus. „Mit CQP sind substituierende und attribuierende Pronomina abfangbar, jedoch nicht Interrogativpronomina. Dies liegt daran, dass sich Interrogativpronomina in der Regel nicht auf einen einzigen Term beziehen, sondern auf Satzinhalte, die aus dem Zusammenhang inferiert werden müssen“ [War15]. Satzanaphern sind durch CQP nicht zuverlässig identifizierbar. In die Klasse der Satzanaphern fallen die substituierenden Interrogativpronomina und die adverbialen Interrogativ- und Relativpronomina. Auch die Pronominaladverbien werden durch dieses Kriterium von der Analyse ausgeschlossen.

Bei dem vierten Auswahlkriterium handelt es sich um das Kriterium der zu niedrigen Frequenz. Bestimmte Klassen von pronominalen Bezügen kommen nur sehr selten oder gar nicht in den untersuchten Korpora vor. Substituierende Possessivpronomina beispielsweise kommen in den Lastenheften nicht vor. Attribuierende Relativpronomina kommen hingegen nur sehr selten vor. Für die genannten Pronomenklassen werden im Folgenden keine Regeln entwickelt.

In Tabelle 34 sind die nicht behandelten Pronomen-Klassen aufgelistet, einschließlich des jeweiligen Ausschlusskriteriums und eines Beispiels. Die Beispiele sind dem Stuttgart-Tübingen-Tagset [STST99] entnommen.

POS-Tag	Pronomen-Klasse	Beispiel	Ausschlusskriterien
PRF PWAT PIDAT	reflexive Personalpronomina attribuierende Interrogativpronomina attribuierende Indefinitpronomina mit Determinator	<i>sich, dich, mir</i> <i>welche Farbe, wessen Hut</i> <i>ein wenig Wasser</i>	nicht anaphorisch
PIS PIAT	substituierende Indefinitpronomina attribuierende Indefinitpronomina ohne Determinator	<i>keiner, viele, man, niemand</i> <i>kein Mensch, irgendein Glas</i>	Quantoren
PWAV PWS PROAV	adverbiales Interrogativ- oder Relativpronomina substituierende Interrogativpronomina Pronominaladverbien	<i>warum, wo, wann, worüber, wobei</i> <i>wer, was</i> <i>dafür, dabei, deswegen, trotzdem</i>	Satzanaphern
PPOSS PRELAT	substituierende Possessivpronomina attribuierende Relativpronomina	<i>meins, deiner</i> <i>der Mann, dessen Hund</i>	Niedrige Frequenz

Tabelle 34: Nicht behandelte Klassen von Pronomina

Die behandelten Klassen von Pronomina, für welche Regeln entworfen werden können, sind in Tabelle 35 aufgelistet. Die Beispiele sind ebenfalls aus dem Stuttgart-Tübingen-Tagset entnommen.

POS-Tag	Pronomen-Klasse	Beispiel
PPER	(irreflexive) Personalpronomina	<i>ich, er, ihm, mich, dir</i>
PDS	substituierende Demonstrativpronomina	<i>dieser, jener</i>
PRELS	substituierende Relativpronomina	<i>der Hund, der</i>
PDAT	attribuierende Demonstrativpronomina	<i>dieser Hund</i>
PPOSAT	attribuierende Possessivpronomina	<i>mein Buch, deine Mutter</i>

Tabelle 35: Behandelte Klassen von Pronomina

In Tabelle 36 sind die Regeln zu sehen, welche für die Identifikation morphosyntaktisch mehrdeutiger pronominaler Bezüge verwendet werden können. Zu sehen ist immer die Variante, in welcher das Pronomen und die Nomina das Genus Maskulin haben. Für jede Regel gibt es drei weitere Varianten: 1) Variante mit Genus Feminin und Numerus Singular 2) Variante mit Genus Neutrum und Numerus Singular 3) Variante mit Numerus Plural. Bei der zuletzt genannten Variante findet keine Unterscheidung nach Genus statt, da die Pluralformen bezüglich des Genus synkretistisch sind. Die vollständige Regelsammlung befindet sich in Anhang E der Arbeit.

Pronominale Bezüge, die satzintern geprüft werden, sind im Regelnamen mit dem Kürzel *si* gekennzeichnet. Somit fängt Regel *PPER_si* Instanzen wie in Beispiel 112 ab. Satzübergreifende Prüfungen sind durch das Kürzel *sü* im Regelnamen gekennzeichnet. Regel *PDS_sü* fängt Instanzen ab wie in Beispiel 114. Für die Klasse der substituierenden Relativpronomina wurden lediglich Regeln für satzinterne Bezüge entwickelt, da bei dieser Klasse keine satzübergreifenden Bezüge auftreten.

Die entwickelten Regeln dienen lediglich der automatischen Identifizierung morphosyntaktisch mehrdeutiger pronominaler Bezüge. Semantische Mehrdeutigkeiten können nicht automatisiert identifiziert werden. Die Unterscheidung, ob ein morphosyntaktisch mehrdeutiger pronominaler Bezug auch semantisch mehrdeutig ist, muss der Leser selbst treffen. Dass sich ein produktiver Einsatz der Regeln aufgrund einer zu hohen Anzahl an Fehlwarnungen nicht lohnt, wird in Abschnitt 8.5.3 näher ausgeführt.

8.5.3 Auswertung der untersuchten pronominalen Bezüge nach Kritikalität

In Abschnitt 8.5.2 wurde bereits beschrieben, dass in den untersuchten Korpora zwar viele Pronomina mit morphosyntaktisch mehrdeutigen Bezügen vorhanden sind, diese jedoch nur sehr selten zu einer semantischen Mehrdeutigkeit führen. In diesem Abschnitt wird

Regelname	Regel
PPER_si	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PPER"& morph=".*sg.*masc.*"] within s;
PPER_sü	[pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* [pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* "\." [word!="\."]* [pos="PPER"& morph=".*sg.*masc.*"];
PDS_si	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PDS"& morph=".*sg.*masc.*"] within s;
PDS_sü	[pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* [pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* "\." [word!="\."]* [pos="PDS"& morph=".*sg.*masc.*"];
PRELS_si	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PPER"& morph=".*sg.*masc.*"] within s;
PDAT_si	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PDAT"& morph=".*sg.*masc.*"] within s;
PPOSAT_si	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PPOSAT"& morph=".*sg.*masc.*"] within s;

Tabelle 36: CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger pronominaler Bezüge

daher keine Evaluierung der entwickelten Regeln diskutiert, sondern es wird eine Übersicht darüber gegeben, wie sich das Verhältnis zwischen morphosyntaktisch mehrdeutigen Bezügen und semantisch mehrdeutigen Bezügen darstellt bzw. wie oft semantisch mehrdeutige Bezüge tatsächlich in den untersuchten Korpora vorkommen.

Korpus	Pronominale Bezüge	Behandelte pronominaler Bezüge	Morphosyntaktisch mehrdeutige pronominaler Bezüge	Semantisch mehrdeutige pronominaler Bezüge
1	18026	9364	6199	120
2	18153	9640	6171	61
3	6798	4172	2913	42
4	12692	6783	4315	35
Σ	55669	29959	19598	258

Tabelle 37: Auswertung der pronominalen Bezüge über alle Korpora

Tabelle 37 gibt einen Überblick über die vier Korpora und über die Häufigkeit der dort enthaltenen pronominalen Bezüge. Insgesamt sind in den vier Korpora 55669 pronominaler Bezüge vorhanden. Von diesen 55669 Okkurrenzen wurden insgesamt 29959 in der

Auswertung näher betrachtet. Hierbei handelt es sich um die in Tabelle 35 aufgelisteten Pronomen-Klassen. Durch Anwendung der entwickelten Regeln und manuelle Durchsicht der Ergebnisse konnte festgestellt werden, dass in den Korpora insgesamt 19598 morphosyntaktisch mehrdeutige pronominaler Bezüge der als relevant identifizierten Klassen vorkommen. In dieser Zahl sind sowohl die satzinternen als auch die satzübergreifenden pronominalen Bezüge zusammengefasst. Durch manuelle Prüfung der morphosyntaktisch mehrdeutigen Bezüge konnte weiterhin festgestellt werden, dass lediglich 258 Instanzen der untersuchten pronominalen Bezüge auch semantisch mehrdeutig sind. Bei den anderen Fällen konnte durch Welt-, Kontext- oder Expertenwissen das korrekte Antezedens zum Pronomen ermittelt werden.

Korpus	Anteil semantisch mehrdeutiger pronominaler Bezüge bezogen auf alle pronominalen Bezüge	Anteil semantisch mehrdeutiger pronominaler Bezüge bezogen auf alle morphosyntaktisch mehrdeutigen pronominalen Bezüge
1	1,28%	1,94%
2	0,63%	0,99%
3	1,01%	1,44%
4	0,52%	0,81%
Gesamt	0,86%	1,32%

Tabelle 38: Anteil der semantisch mehrdeutigen pronominalen Bezüge über alle Korpora

Tabelle 38 zeigt wie sich der Anteil der semantisch mehrdeutigen pronominalen Bezüge zu den morphosyntaktisch mehrdeutigen pronominalen Bezügen darstellt. Zusätzlich wird das Verhältnis semantisch mehrdeutiger pronominaler Bezüge in Bezug auf die Gesamtmenge der behandelten pronominalen Bezüge dargestellt. Bezogen auf die Gesamtmenge der behandelten pronominalen Bezüge übersteigt der Anteil der semantisch mehrdeutigen Bezüge in keinem der Korpora 1,28%. Der Gesamtanteil über alle Korpora beträgt 0,86%.

Bezogen auf die Menge der morphosyntaktisch mehrdeutigen pronominalen Bezüge, ist der Prozentsatz nur geringfügig höher. Der Gesamtanteil über alle Korpora liegt hier insgesamt bei 1,32%.

Der Anteil der semantisch mehrdeutigen pronominalen Bezüge ist so niedrig, dass sich ein produktiver Einsatz der entwickelten Regeln nicht lohnt. Die Zahl der Fehlwarnungen, die eine Analyse liefern würde, wäre zu hoch.

8.5.4 Zusammenfassung

Die Verwendung von Pronomina kann einen Einfluss auf die Verständlichkeit einer Anforderung haben, da Pronomina auf verschiedene Objekte in der außersprachlichen Realität

referieren können. Kongruiert ein Pronomen in Genus und Numerus mit mehreren Nomina oder Nominalgruppen, die potentielle Antezedenten sein können, so kann der Bezug des Pronomens nicht eindeutig zugeordnet werden. Jedoch bedeutet ein morphosyntaktisch mehrdeutiger pronominaler Bezug nicht gleichzeitig, dass dieser auch semantisch mehrdeutig ist. Die Evaluierung in Abschnitt 8.5.3 hat gezeigt, dass nur ein sehr geringer Anteil der morphosyntaktisch mehrdeutigen Bezüge auch tatsächlich semantisch mehrdeutig ist. In vielen Fällen hilft Welt-, Kontext- und Expertenwissen dabei, die potentielle Ambiguität aufzulösen, d.h. dem Pronomen das korrekte nominale Antezedens zuzuweisen.

Es wurden Regeln entwickelt, mit welchen morphosyntaktisch mehrdeutige Bezüge automatisiert identifiziert werden können. Jedoch kann allein durch die Regeln keine Entscheidung darüber getroffen werden, ob die gefundenen Instanzen auch semantisch mehrdeutig sind. Ein produktiver Einsatz der Regeln in einem Werkzeug zur sprachlichen Qualitätskontrolle ist daher nicht sinnvoll.

8.6 Phänomenanalysen für englischsprachige Lastenhefte

Die in den Abschnitten 8.1 bis 8.5 präsentierten Phänomenanalysen wurden auf deutschsprachigen Anforderungen durchgeführt. Um zu überprüfen, ob das Vorgehen, welches bei der Analyse deutschsprachiger Anforderungen verwendet wird, auch auf weitere Sprachen übertragen werden kann, wurden ausgewählte Analysen auch auf englischen Lastenheften durchgeführt. Hierfür wurden die Phänomene *Weak-Word*, *Passiv ohne Agens* und *Modalverb „should“* analysiert. Auf diese Phänomene wird in den folgenden Abschnitten im Einzelnen eingegangen. Ein weiteres Ziel neben der Überprüfung, ob das Vorgehen der deutschen Analysen auf englischsprachige Lastenhefte übertragen werden kann, ist zu untersuchen, ob sich die entwickelten Regeln, aufgrund der Ähnlichkeit der Sprachen innerhalb der germanischen Sprachfamilie, 1:1 übertragen lassen.

Für die Analyse der Phänomene wurden weitere Korpora erstellt (vgl. Tabelle 39). Die in den Korpora vorhandenen Anforderungen wurden zunächst tokenisiert und anschließend um Lemma-, Wortart-, Morphologie- und Parsing-Information erweitert. Die Annotationen entsprechen in diesen Korpora nicht dem in Kapitel 6 präsentierten Stuttgart-Tübingen-Tagset, sondern dem Tagset, welches im *Penn Treebank Project*²² der *University of Pennsylvania* verwendet wurde [San90], mit geringen Modifikationen. Das Part-of-Speech-Tagset, welches relevant ist für die in den folgenden Abschnitten vorgestellten Analysen, ist in Anhang B abgebildet.

²²vgl. hierzu auch <https://catalog ldc.upenn.edu/LDC2015T13>

Name	Sätze	Tokens
Entwicklungskorpus	86 004	1 152 920
Evaluierungskorpus	78 558	1 087 919
Gesamtkorpus	164 562	2 240 839

Tabelle 39: Korpusinformationen zu den englischen Korpora

Wie für die deutschen Phänomenanalysen, wurden auch für die Analyse zum Englischen ein Entwicklungskorpus und ein Evaluierungskorpus erstellt. Das Entwicklungskorpus wird dazu verwendet, erste Analysen zu einem Phänomen durchzuführen und Regeln für die Identifikation kritischer Instanzen zu entwerfen. Außerdem werden am Entwicklungskorpus die entwickelten Regeln evaluiert. Das Evaluierungskorpus dient ebenfalls und primär zur Evaluierung der entworfenen Regeln. Es handelt sich um ungesehene Daten, durch welche die Validität des Ansatzes gesichert werden soll. Für die Phänomene *Weak-Word* und *Modalverb „should“* wurde eine Evaluierung am gesamten Entwicklungskorpus und am gesamten Evaluierungskorpus durchgeführt. Für das Phänomen *Passiv ohne Agens* wurde, wie bei den deutschen Analysen, nur eine Teilmenge der Lastenhefte betrachtet, da die Anzahl der Passivsätze ohne Agens für eine Evaluierung an der Gesamtmenge zu umfangreich ist, sodass eine Evaluierung anhand von Stichproben ausreicht.

In den Abschnitten 8.6.1 bis 8.6.3 wird nun auf die Phänomene *Weak-Word*, *Passiv ohne Agens* und *Modalverb „should“* eingegangen.

8.6.1 Weak-Words

In den untersuchten englischsprachigen Korpora treten Weak-Words wie beispielsweise *long*, *small* oder *large* häufig auf und führen in bestimmten Satzkontexten zu Interpretationsspielraum, da für das Verständnis wichtige Information ungenannt bleibt.

Beispiel 119 *For a velocity equal or larger than 266 km/h this bit is set to ‘1’.*

Beispiel 120 *The specified maximum and minimum value shall not be exceeded during transmission, especially in situations with very strong bending of the predicted ego vehicle path or a large slip angle of the vehicle.*

In Beispiel 119 ist eine Anforderung zu sehen, welche das Weak-Word *large* enthält. Das Weak-Word ist als unkritisch zu beurteilen, da es durch die quantifizierende Angabe *266*

km/h eindeutig spezifiziert wird, und somit kein Interpretationsspielraum aufgrund fehlender Information entsteht. In Beispiel 120 hingegen ist eine Instanz von *large* enthalten, welche unterspezifiziert ist. Auch Kontextwissen hilft in diesem Fall nicht, den Interpretationsspielraum bezüglich der fehlenden Information zu reduzieren. Weiterhin ist in Beispiel 120 das Weak-Word *strong* enthalten. Da auch zu diesem Weak-Word quantifizierende Angaben fehlen, entsteht hier ebenfalls Interpretationsspielraum aufgrund fehlender Information.

Die Beispiele zeigen, dass der Grad der Kritikalität von Weak-Words im Englischen ebenfalls stark von dem Satzkontext abhängig ist, in welchem sie stehen. Für ausgewählte Weak-Words wurden Regeln entwickelt, mit welchen eine Kontextanalyse, vergleichbar mit den Analysen zu den deutschen Weak-Words, durchgeführt werden kann. Dies soll im Folgenden beispielhaft an dem Weak-Word *large* präsentiert werden.

Regelname	Zusammensetzung	Beispiel
R1_large	best.DT large	the larger value
R2_large	[]+ Large	the Large Controller
R3_large	<s> Large [A-Z].*	Large Parameter Test
R4_large	LARGE	LARGE
R5_large	large []{0,7} .*[0-9].*./._+/'"/e.g./i.e./case/(/)/:[a-z].*\n*/example/contractor/chapter/table/CD/JJS/...	large distances (see chapter 5)
R6_large	WW.COMP/SUP []{0,7} (/than/as/CD	larger than 2 meters
R7_large	as WW as	as large as
R8_large	WW .? as/enough/that	large enough to

Tabelle 40: Regeln für das Weak-Word *large*

In Tabelle 40 sind Regeln für die Identifikation unkritischer Instanzen des Weak-Word-Lemmas *large* zu sehen. R1_large identifiziert Instanzen von *large*, in welchen dem Weak-Word ein bestimmter Artikel vorangestellt wird. Wie im Deutschen, wird bei Vorhandensein eines bestimmten Artikels angenommen, dass das nachfolgende Weak-Word und das Nomen bereits bekannt sind und das Weak-Word somit nicht kritisch ist. R2_large, R3_large und R4_large identifizieren Instanzen von *large*, in welchen das Weak-Word ein definierter Fachterminus ist. Instanzen, bei welchen *large* als Fachterminus oder als Teil eines solchen fungiert, werden somit den Lastenheftautoren nicht zur Korrektur vorgelegt. Regel R5_large findet Kontexte von *large*, in welchen das Weak-Word durch einen Verweis bzw. eine Referenz näher spezifiziert wird. Dies kann beispielsweise durch einen Verweis auf eine Norm oder Vereinbarung, ein bestimmtes Kapitel, eine Tabelle oder eine bestimmte Anforderung geschehen. Auch die Abkürzungen *e.g.* und *i.e.* liefern einen Hinweis darauf, dass das Weak-Word im Text noch näher spezifiziert wird.

R6_large identifiziert Instanzen, in welchen *large* in einen Vergleich gesetzt wird. Weiterhin werden durch diese Regel Instanzen von *large* gefunden, in welchen das Weak-Word im Superlativ steht. Diese Instanzen von *large* sind ebenfalls als unkritisch zu beurteilen.

Durch die Regel R7_large werden Instanzen vom Typ *as large as* erkannt. Regel R8_large identifiziert hingegen Instanzen, welche die Struktur *large enough* besitzen. Auch diese Verwendungen des Weak-Words *large* sind als unkritisch zu betrachten.

Die Instanzen von *large*, die durch keine Regel identifiziert werden, werden als kritisch interpretiert und dem Autor zur Kontrolle aufgezeigt.

		Automatische Analyse →	
Manuelle Analyse ↓	Kritische Anforderung	72	4
	Unkritische Anforderung	5	38

Abbildung 43: Evaluierung der Regeln des Weak-Words *large* am Entwicklungskorpus, N=119

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	93,51%	90,48%
Recall	94,74%	88,37%
F-Measure	94,12%	89,41%

Tabelle 41: Precision-, Recall- und F-Measure-Werte des Weak-Words *large* am Entwicklungskorpus, N=119

Die Precision- und Recall-Werte am Entwicklungs- und am Evaluierungskorpus liegen in einem hohen Bereich (vgl. Tabelle 41 und Tabelle 42). Es entsteht bei der Analyse nur eine geringe Anzahl an Fehlwarnungen und Fehlentwarnungen.

Im Entwicklungskorpus sind insgesamt 119 Instanzen von *large* enthalten (vgl. Abbildung 43). 43 dieser Instanzen wurden bei der manuellen Analyse als unkritisch identifiziert, 76 Instanzen hingegen als kritisch. Der entwickelte Ansatz identifiziert 72 kritische Instanzen korrekt als kritisch und 38 unkritische korrekt als unkritisch. Bei der Analyse werden auch fünf Fehlwarnungen produziert und vier mal wird fälschlicherweise keine Warnung ausgegeben. Im Evaluierungskorpus verhalten sich die Zahlen ähnlich: Es wurden sieben Fehlwarnungen produziert und drei mal wurde fälschlicherweise keine Warnung ausgegeben (vgl. Abbildung 44). Auf die Gründe für die Fehlwarnungen und Fehlentwarnungen wird im Folgenden näher eingegangen.

		Automatische Analyse →	
		Kritische Anforderung	Unkritische Anforderung
Manuelle Analyse ↓	Kritische Anforderung	67	3
	Unkritische Anforderung	7	47

Abbildung 44: Evaluierung der Regeln des Weak-Words *large* am Evaluierungskorpus, N=124

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	90,54%	94,00%
Recall	95,71%	87,04%
F-Measure	93,05%	90,39%

Tabelle 42: Precision-, Recall- und F-Measure-Werte des Weak-Words *large* am Evaluierungskorpus, N=124

Beispiel 121 [...] *for example concrete borders with large reflectors having 60 m distance difference.*

Eine Anforderung, bei welcher fälschlicherweise keine Warnung an den Benutzer ausgegeben wurde, ist in Beispiel 121 zu sehen. Die Anforderung enthält neben dem Weak-Word *large* auch eine quantifizierende Angabe (*60 m*). Regel **R5_large** identifiziert u.a. Instanzen von *large*, welche von einer quantifizierenden Angabe gefolgt werden. Die Anforderung aus Beispiel 121 wird folglich durch diese Regel identifiziert. Jedoch ist zu beachten, dass sich die Angabe *60 m* in diesem Fall nicht auf das Adjektiv *large* bezieht. An dieser Stelle wird dem Benutzer fälschlicherweise eine Warnung vorenthalten.

Beispiel 122 *The resulting time distance to the object shall always be large enough for system interaction or comfortable driver interaction.*

Fehlwarnungen entstanden größtenteils durch Rechtschreibfehler. Eine Anforderung, welche aufgrund eines Rechtschreibfehlers fälschlicherweise als kritisch identifiziert wurde, ist in Beispiel 122 zu sehen. Regel **R8_large** erkennt Instanzen vom Typ *large enough*. Da die Anforderung in Beispiel 122 einen Rechtschreibfehler enthält (*enough* anstatt *enough*), kann die Regel nicht greifen und es wird fälschlicherweise eine Warnung an den Benutzer ausgegeben.

Insgesamt kann man feststellen, dass Weak-Words auch in englischsprachigen Lastenheften ein Problem darstellen. Kann eine nicht textuell repräsentierte Information nicht durch Kontext- oder Expertenwissen hergeleitet werden, führt dies zu Verständnisproblemen, da Interpretationsspielraum entsteht. Die Analysen haben gezeigt, dass sich die

Methode für die Behandlung von Weak-Words auch auf das Englische übertragen lässt. Jedoch lassen sich die Regeln nicht 1:1 vom Deutschen übernehmen. Für eine zuverlässige Analyse von Weak-Words, bei welcher nur wenige Fehlwarnungen und Fehlentwarnungen entstehen sollen, müssen sprachspezifische Analysen durchgeführt werden.

8.6.2 Passiv ohne Agens

In den untersuchten Korpora sind neben Aktivsätzen auch eine große Anzahl an Passivsätzen zu finden. Analog zur Passivverwendung im Deutschen, kann das Weglassen der semantischen Rolle Agens dazu führen, dass Interpretationsspielraum entsteht. Dies ist der Fall, wenn das Agens weder durch Welt- noch durch Kontextwissen hergeleitet werden kann. Im Englischen kann das Agens durch eine Präpositionalphrase mit *by* im Passivsatz genannt werden. Diese Angabe ist – wie deutsche *durch*-Präpositionalphrasen – nicht obligatorisch und kann deshalb weggelassen werden.

Auch im Englischen ist es der Fall, dass die semantischen Rollen Agens und Patiens in Aktiv- und Passivsätzen unterschiedlichen syntaktischen Funktionen zugeordnet werden. Im Aktivsatz ist das Subjekt das Agens, im Passivsatz hingegen nimmt das Subjekt die Rolle des Patiens ein.

Um die Kritikalität des Phänomens *Passiv ohne Agens* herauszustellen, wurde für englischsprachige Anforderungen dasselbe Experiment durchgeführt, welches bei der Analyse deutscher Passivsätze bereits präsentiert wurde (vgl. Abschnitt 8.3): Hierfür wurden 711 englische Passiv-Anforderungen ohne Agens zufällig aus dem Entwicklungskorpus ausgewählt und danach bewertet, ob das Weglassen des Agens dazu führt, dass der Handelnde durch den Leser nicht hergeleitet werden kann. Die Bewertung wurde aus zwei Perspektiven vorgenommen: In einer ersten Analyse wurden die 711 Anforderungen unabhängig von den ihnen im Lastenheft jeweils vorausgehenden und nachfolgenden Anforderungen bewertet, d.h. weitere Information konnte für die Herleitung des Agens nicht hinzugezogen werden. In einer zweiten Analyse wurden die umgebenden Anforderungen in die Interpretation miteinbezogen.

Die Ergebnisse verhalten sich sehr ähnlich zu den Ergebnissen, welche sich bei der Analyse deutscher Passivsätze ohne Agens ergeben haben. Im Englischen ist es ebenfalls der Fall, dass beim Einbezug von Kontextwissen die Verwendung des Passivs kaum Probleme macht. In den meisten Fällen kann die vermeintlich fehlende Information aus dem Kontext hergeleitet werden, d.h. durch weitere Anforderungen oder durch Tabellen. Ohne Einbezug von Kontextwissen sind 236 von 711 stichprobenweise bewerteten Anforderungen (33,2%) diesbezüglich kritisch. Mit Einbezug von Kontextwissen sind es nur 2 Anforderungen (= 0,28%). Diese Zahlen verhalten sich sehr ähnlich zu den Zahlen, welche im Bezug auf die

deutsche Passivanalyse vorgestellt wurden (ohne Kontext: 26%, mit Kontext: 0%). Das Phänomen *Passiv ohne Agens* ist demnach in den betrachteten Texten unproblematisch. Zu beachten ist, dass sich die Ergebnisse lediglich auf die betrachtete Textsorte beziehen. Es kann nicht angenommen werden, dass sich die Verwendung von Passivkonstruktionen ohne Agens bezogen auf jede Textsorte unkritisch verhält. In den betrachteten Anforderungsdokumenten werden Funktionen für einzelne Komponenten beschrieben. Wenn es in Anforderungen um verteilte Systeme geht, ist die Nennung des Agens wichtig und die Nicht-Nennung kann zu Interpretationsspielraum führen. In den hier betrachteten Anforderungen kann der Leser aufgrund seines Gegenstandsbereichswissens und/oder aufgrund von Angaben im Kontext die nicht ausgedrückte Agens-Funktion erschließen und eindeutig zuordnen, sodass die Textgestalt kein Problem darstellt²³.

In dieser Arbeit werden keine Regeln für die Identifikation englischer agensloser Passivsätze präsentiert. Es ist jedoch denkbar Regeln zu entwickeln. Im Englischen wird das Passiv mit dem *past participle* gebildet. Dieses kann durch das POS-Tag *VBN* unter der Verwendung des Tagsets des *Penn Treebank Project* identifiziert werden. Ob ein Agens genannt wird, kann durch Suche nach einer Präpositionalphrase mit *by* geprüft werden.

8.6.3 Das Modalverb *should*

Bei der Verwendung des deutschen Modalverbs *sollte(n)* besteht potentiell die Gefahr, dass zwei Interpretationen bezüglich der Verbindlichkeit von *sollte(n)* vom Leser getroffen werden können. In diesen Fällen liegen zwei mögliche Lesarten vor: die deontische Lesart (= *müssen*) und die dispositionelle Lesart (= *können*). Bei der Verwendung des englischen Modalverbs *should* besteht ebenfalls die Gefahr, dass Interpretationsspielraum aufgrund Vorhandenseins dieser beiden Lesarten entsteht. Können beide Interpretationen getroffen werden, bedeutet dies, dass nicht klar angegeben ist, ob die beschriebene Funktion von einem Auftragnehmer realisiert werden muss oder ob es sich um eine freiwillig zu erbringende Leistung handelt (vgl. Beispiel 123). Auch Potentialisfälle von *should* sind in den englischen Korpora zu finden (vgl. Beispiel 124). Diese Instanzen sind unkritisch, da hier keine Ambiguität zwischen der *müssen*- und *können*-Lesart besteht.

Beispiel 123 *The characteristic of loss control should be adjustable by parameters.*

Beispiel 124 *Should the safety semiconductor appear to be faulty, the SRS warning light is to be activated.*

²³Detailliertere Analysen hierzu finden sich in [KH15].

Bei einer automatischen Analyse müssen, wie auch bei der Analyse des Wortes *sollte*, die Instanzen an den Autor zurückgeliefert werden, bei welchen eine potentielle Ambiguität zwischen der deontischen und der dispositionellen Lesart vorliegt. Bei den deutschen Analysen wurden die kritischen Instanzen von *sollte* gesucht und die Treffer dem Benutzer zurückgeliefert. Bei den englischen Analysen hat es sich gezeigt, dass es von Vorteil ist, nach den unkritischen Instanzen zu suchen und die Instanzen, die durch keine Regel identifiziert werden, dem Benutzer zur Kontrolle zurückzuliefern.

Nach Durchsicht der Instanzen im Entwicklungskorpus konnten drei Regeln abgeleitet werden, durch welche unkritische Instanzen von *should* identifiziert werden können. Diese sind in Tabelle 43 zu sehen.

Regelname	Regel
R1_should	a:[word="[I i]f"] []* b:[word="should"] [{0,2} c:[pos="V.+"] :: a.dep=c.wortnr & b.dep=c.wortnr within 1s;
R2_should	a:[word="[I i]f"] []* b:[word="should"] "be"[!pos="V.+"]* c:[pos="NN"] :: a.dep=c.wortnr & b.dep=c.wortnr within 1s;
R3_should	<s> [word="Should"];

Tabelle 43: CQP-Regeln für die Identifikation unkritischer Verwendungen von *should*

Durch die Regel R1_should können Instanzen wie in Beispiel 125 und Beispiel 126 identifiziert werden. In diesen Anforderungen ist *should* jeweils in der Potentialislesart in einen Bedingungssatz eingebettet.

Beispiel 125 *If this case should arise, the V0's take responsibility for the coordination between all V3's.*

Beispiel 126 *If this condition should not be fulfilled, level 1 has to abort the start.*

Auch in Beispiel 127 ist *should* in einen Konditionalsatz eingebettet und es liegt keine Ambiguität zwischen der deontischen und der dispositionellen Lesart vor. Instanzen wie in Beispiel 127 können durch Regel R2_should identifiziert werden.

Beispiel 127 *If there should be not enough space in some cars, a solution shall be agreed with Daimler.*

In Beispiel 128 leitet *Should* einen Konditionalsatz ein. Das Antezedens (*Should the supply voltage for an X-module fall below permitted levels*) entspricht in diesem Fall einem Konjunktionalsatz mit *If*. Auch in Beispiel 128 liegt folglich die Potentialislesart vor, die bei einer automatischen Analyse dem Autor nicht aufgezeigt werden muss. Im Hinblick auf die Verbindlichkeit der Anforderung ist diese Verwendung von *Should* unkritisch. Dieser Fall verhält sich analog zum Deutschen. Verwendungen wie diese werden durch Regel R3_should identifiziert.

Beispiel 128 *Should the supply voltage for an X-module fall below permitted levels, the data in this X will be in an undefined state after the control unit is restarted.*

Die Regeln wurden anhand des Entwicklungskorpus entwickelt und am Entwicklungs- und am Evaluierungskorpus evaluiert. Die Evaluierungsergebnisse am Entwicklungskorpus sind in Abbildung 45 und Tabelle 44 zu sehen.

		Automatische Analyse	
		Kritische Anforderung	Unkritische Anforderung
Manuelle Analyse	Kritische Anforderung	594	0
	Unkritische Anforderung	1	25

Abbildung 45: Evaluierung der Regeln für *should* am Entwicklungskorpus, N=620

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	99,83%	100%
Recall	100%	96,15%
F-Measure	99,91%	98,04%

Tabelle 44: Precision-, Recall- und F-Measure-Werte der Regeln für *should* am Entwicklungskorpus, N=620

Das Entwicklungskorpus enthält insgesamt 620 Instanzen von *should*. 594 der Instanzen sind kritisch, da hier Interpretationsspielraum bezüglich der deontischen und dispositiven Lesart entstehen kann. Lediglich 26 Instanzen sind unkritisch bezüglich dieser Eigenschaft. Durch den entwickelten Ansatz konnten alle kritischen Instanzen von *should* im Entwicklungskorpus identifiziert werden. Allerdings wurde auch eine überflüssige Warnung ausgegeben. Die Gründe wieso Fehlwarnungen bei der Analyse auftreten, werden weiter unten im Text genannt. Die Precision- und Recall-Werte liegen in einem sehr hohen Bereich und zeigen, dass die entwickelten Regeln auf dem bekannten Datensatz solide Ergebnisse liefern.

		Automatische Analyse →	
Manuelle Analyse ↓	Kritische Anforderung	541	0
	Unkritische Anforderung	8	7

Abbildung 46: Evaluierung der Regeln für *should* am Evaluierungskorpus, N=556

	Klasse: Kritische Anforderung	Klasse: Unkritische Anforderung
Precision	98,54%	100%
Recall	100%	46,66%
F-Measure	99,26%	63,63%

Tabelle 45: Precision-, Recall- und F-Measure-Werte der Regeln für *should* am Evaluierungskorpus, N=556

Die Ergebnisse am Evaluierungskorpus sind in Abbildung 46 und Tabelle 45 zu sehen. Das Evaluierungskorpus enthält 556 Instanzen von *should*. 541 dieser Instanzen sind kritisch und 15 unkritisch. Durch den entwickelten Ansatz wurden sieben Instanzen korrekt als unkritisch klassifiziert. Die 541 kritischen Instanzen wurden allesamt korrekt als kritisch klassifiziert. Dies bedeutet, dass keine Fehlentwarnungen bei der Analyse entstanden sind. Jedoch wurden acht Fehlwarnungen produziert. Der Recall-Wert der Klasse *Unkritische Anforderung* ist daher verhältnismäßig niedrig.

Die Entstehung von Fehlwarnungen ist auf zwei Gründe zurückzuführen, welche im Folgenden diskutiert werden.

Beispiel 129 *This test is to be conducted with the crossover capacitor, should one exist in the production part.*

In Beispiel 129 leitet *should* einen Konditionalsatz ein (*should one exist in the production part*), welchem das Konsequens vorausgeht. Da der Bedingungssatz, welcher mit *should* eingeleitet wird, nicht am Satzanfang steht, wird dieser nicht durch die Regeln identifiziert (vgl. oben Regel R3_should).

Beispiel 130 *Therefore the current powernet state and the C-state are evaluated to determine if the ventilation should be active or not.*

Weiterhin kamen Fehlwarnungen bei der Analyse zustande, weil bisher noch nicht alle unkritischen Fälle von *should* durch Regeln abgebildet wurden. In Beispiel 130 ist eine Anforderung zu sehen, welche eine unkritische Verwendung von *should* enthält. Auch hier ist *should* in einen Bedingungssatz eingebettet. In diesem Fall liegt ebenfalls die Poten-

tialislesart vor. Fälle wie diese können durch Hinzufügen einer neuen Regel identifiziert werden. Die Regel kann folgendermaßen aussehen:

```
a:[word="[I|i]f"] []* b:[word="should"] []0,2 c:"be" d:[pos="JJ"]
:: a.dep=d.wortnr & b.dep=d.wortnr & c.dep=d.wortnr within 1s;
```

Durch Hinzufügen dieser Regel kann die Anzahl der Fehlwarnungen noch weiter reduziert werden. Im Falle des Evaluierungskorpus sind dies zwei Fehlwarnungen.

Es hat sich gezeigt, dass durch die Verwendung des Modalverbs *should* Interpretationsspielraum bezüglich der Verbindlichkeit einer Anforderung entstehen kann, wenn zwei mögliche Lesarten von *should* vorliegen: deontisch vs. dispositionell. Das Modalverb *should* verhält sich diesbezüglich wie das deutsche Modalverb *sollte(n)*. Durch die in diesem Abschnitt präsentierten Regeln können die Instanzen von *should* identifiziert werden, bei welchen die Potentialislesart vorliegt, d.h. die unkritischen Verwendungen von *should*. Bei den Anforderungen, die durch keine Regel identifiziert werden (Komplementmenge), liegt potentiell eine Ambiguität zwischen der deontischen und der dispositionellen Lesart vor. Diese Instanzen von *should* müssen dem Autor zur Kontrolle aufgezeigt werden..

8.6.4 Zusammenfassung der Ergebnisse der englischen Analysen

Im vorliegenden Abschnitt wurde untersucht, ob sich die für die Behandlung der deutschen Phänomene verwendeten computerlinguistischen Technologien und Methoden auch auf englischsprachige Anforderungen anwenden lassen. Folgende Phänomene wurden hierfür beispielhaft betrachtet: *Weak-Word*, *Passiv ohne Agens* und *Modalverb „should“*.

Weak-Words sind in den englischen Korpora häufig vorhanden. Die Kritikalität von Weak-Words ist wie im Deutschen stets im Kontext zu beurteilen. Tritt beispielsweise ein Dimensionsadjektiv ohne quantifizierende Angabe auf, entsteht Interpretationsspielraum, was wiederum einen negativen Einfluss auf die Testbarkeit der entsprechenden Anforderung hat. Für englische Weak-Words können unter Verwendung von Part-of-Speech-Tags, Lemma-Information und Morphologie-Information Regeln abgeleitet werden, mit welchen je nach Kontext entschieden werden kann, ob ein Weak-Word in einer bestimmten Verwendung kritisch ist. Das Fehlen einer quantifizierenden Angabe ist bei dem Phänomen *Weak-Word* sprachübergreifend kritisch zu beurteilen. Auf der Ebene der Semantik funktionieren englische und deutsche Weak-Words parallel. Jedoch lassen sich die Regeln nicht 1:1 vom Deutschen auf das Englische übertragen, da die Sprachen morphosyntaktisch nicht komplett deckungsgleich sind. Für jedes Weak-Word müssen die Kontexte, in welchen es auftreten kann, intensiv untersucht werden.

Das Phänomen *Modalverb „should“* verhält sich ähnlich zum deutschen Modalverb *sollte(n)*. Existieren zwei mögliche Lesarten von *should* (deontisch vs. dispositionell), ist die Verbindlichkeit der Anforderung nicht eindeutig festgelegt. Auf der Ebene der Semantik verhält sich das Modalverb *should* folglich parallel zum deutschen Modalverb *sollte(n)*. Auch auf der Ebene der Syntax verhalten sich die beiden Phänomene ähnlich. Um die kritischen Instanzen von *should* identifizieren zu können, wurden Regeln entwickelt, welche die Potentialisfälle von *should* herausfiltern.

Das Phänomen *Passiv ohne Agens* ist in der betrachteten Textsorte als nicht kritisch zu beurteilen. In den untersuchten Korpora treten zwar häufig Passivsätze ohne Agens auf, jedoch hilft in den meisten Fällen Kontextinformation dabei, die fehlende Information herzuleiten. Dieses Phänomen verhält sich ebenfalls analog zum Deutschen.

Insgesamt lässt sich feststellen, dass sich die Analysemethoden, welche an deutschsprachigen Lastenheften entwickelt wurden, auch auf englischsprachige Lastenhefte übertragen lassen. Auf der semantischen Ebene funktionieren die englischen Phänomene gleich wie die deutschen Phänomene. Da die Phänomene auf morphosyntaktischer Ebene geprüft werden, lassen sich die Regeln nicht 1:1 übertragen und müssen für jedes Phänomen einzelsprachlich angepasst werden.

8.7 Computerlinguistische Bewertung

Für die Durchführung der Phänomenanalysen, welche in den Abschnitten 8.1 bis 8.6 beschrieben sind, wurden Methoden aus der formbasierten Korpuslinguistik verwendet: Für die Analysen standen die annotierten Informationen (Lemmata, Wortarten, Morphologieinformationen und Parsinginformation) bereit, auf welche durch die Korpusabfragesprache CQP unter Verwendung regulärer Ausdrücke für die Analysen zugegriffen wurde. Die Analysen einzelner Phänomene liefern unterschiedlich gute Ergebnisse in Bezug auf die Identifikation kritischer Instanzen. Für manche Phänomene ist eine zuverlässige automatische Unterscheidung zwischen kritischen und unkritischen Instanzen möglich, für andere Phänomene hingegen nicht, da der Einsatz von Welt- und Expertenwissen hier eine erhebliche Rolle spielt.

Die verwendete Technologie liefert gute Ergebnisse für die Phänomene *Weak-Word* und *Modalverb sollte(n)*, da hier für die Unterscheidung zwischen kritischen und unkritischen Instanzen der Einsatz von Welt- und Expertenwissen eine vergleichsweise kleine Rolle spielt. Durch die Verwendung von Weak-Words kann das Problem auftreten, dass für das einheitliche Verständnis wichtige Information fehlt. Ein häufiger Grund hierfür ist das Fehlen einer quantifizierenden Angabe. Fehlt eine solche Angabe eines messbaren Werts,

kann selbst durch Welt- oder Expertenwissen die fehlende Information in den meisten Fällen nicht hergeleitet werden.

Die Ambiguität zwischen der deontischen und der dispositionellen Lesart des Modalverbs *sollte(n)* kann ebenfalls nicht durch Welt- oder Expertenwissen aufgelöst werden. Das heißt, es kann auch durch Einsatz dieser Art von Wissen nicht festgelegt werden, ob es sich bei einer Anforderung, in welcher das Modalverb *sollte(n)* auftritt, um eine optional zu realisierende Anforderung handelt, oder ob der Auftragnehmer in der Pflicht steht, die geforderte Funktionalität zu realisieren. Bei der Formulierung von Regeln hierfür ist der Einsatz von Lemma-Information, Part-of-Speech-Tags, morphologischer Information und Parsinginformation ausreichend, um die kritischen von den unkritischen Instanzen zuverlässig zu unterscheiden.

Für die Analyse der Phänomene *Pronominaler Bezug*, *Unpersönliche Konstruktion* und *Passiv ohne Agens* gestaltet sich das Ergebnis etwas anders. Morphosyntaktisch mehrdeutige pronominale Bezüge können mit der beschriebenen Technologie zuverlässig automatisiert identifiziert werden. Jedoch spielt bei diesem Phänomen der Einsatz von Welt- und Expertenwissen eine erhebliche Rolle: ein morphosyntaktisch mehrdeutiger Bezug ist zwar ein Indiz dafür, dass auch ein semantisch mehrdeutiger Bezug vorliegen könnte, jedoch haben die Analysen gezeigt, dass ein morphosyntaktisch mehrdeutiger Bezug in den hier benutzten Korpora nur in sehr seltenen Fällen auch zu einem semantisch mehrdeutigen Bezug führt. Durch Einsatz von Welt- und Expertenwissen kann der morphosyntaktisch mehrdeutige Bezug in den meisten Fällen aufgelöst werden. Hier stößt die formbasierte Computerlinguistik an ihre Grenzen: Wissen, das auf Welt- oder Expertenwissen zurückzuführen ist, kann durch den in dieser Arbeit verfolgten automatisierten Ansatz nicht bereitgestellt werden. Bei den Passivanalysen stößt man auf dieselbe Problematik. Die generelle Erkennung von Passivsätzen ist mit den beschriebenen Mitteln möglich. Passivsätze treten verhältnismäßig oft in den Korpora auf, und die Verwendung des Passivs kann dazu führen, dass das Agens der Handlung nicht explizit im Text genannt wird. Jedoch kann durch Welt-, Experten- oder Kontextwissen in den meisten Fällen das Agens ermittelt werden. Prüfungen, die zuverlässig zwischen kritischen und unkritischen Instanzen unterscheiden, können mit den Mitteln der formbasierten Korpuslinguistik für diese Phänomene nicht entwickelt werden. Es können lediglich Hinweise geliefert werden, an welchen Stellen Information fehlen könnte oder ein potentiell mehrdeutiger Bezug vorliegt.

Auch bei Phänomenen, welche dem Testbarkeitskriterium *Widerspruchsfreiheit* zugeordnet sind, stößt die formbasierte Korpuslinguistik an ihre Grenzen. Da die natürliche Sprache sehr flexibel ist und es viele Möglichkeiten gibt ein und denselben Sachverhalt auszu-

drücken (unterschiedliche Satzstruktur, Verwendung von Synonymen), können unter der Verwendung von Methoden aus der formbasierten Korpuslinguistik für diese Phänomene keine zuverlässigen Analysen durchgeführt werden. Der Einsatz von Terminologiedatenbanken und Ontologien könnten ein Mittel sein, potentielle Widersprüche zu identifizieren. Jedoch können auch durch Einsatz dieser Technologien nicht alle Widersprüche zuverlässig identifiziert werden.

Die inhaltliche Korrektheit eines Satzes sowie die effiziente und effektive Durchführbarkeit kann durch einen automatisierten Ansatz ebenfalls nicht überprüft werden. Hierfür wird menschliches Wissen benötigt.

Die Analysen, die auf deutschsprachigen Lastenheften durchgeführt wurden, lassen sich auf englischsprachige Lastenhefte übertragen (vgl. Abschnitt 8.6). Die Phänomene *Weak-Word* und *Modalverb „should“* liefern ähnlich gute Ergebnisse wie im Deutschen. Auch hier kann zuverlässig zwischen kritischen und unkritischen Instanzen unterschieden werden. Bei dem Phänomen *Passiv ohne Agens* verhält es sich ebenfalls wie im Deutschen: In den meisten Fällen kann durch Einsatz von Welt- und Expertenwissen die potentiell fehlende Information hergeleitet werden. Die entwickelten Regeln können lediglich zwischen Passivsätzen mit Agens und ohne Agens unterscheiden. Die Entscheidung darüber, ob durch das Fehlen des Agens eine kritische Konstruktion entsteht, weil das Agens auch nicht durch Welt- oder Expertenwissen hergeleitet werden kann, liegt in der Hand des Lesers.

Bei der Analyse von Phänomenen, bei welchen der Einsatz von Welt- und Expertenwissen eine vergleichsweise große Rolle spielt, stößt die formbasierte Korpuslinguistik an ihre Grenzen. Diese Art von Wissen kann durch ein Werkzeug nicht bereitgestellt werden. Die entwickelten Analysen können jedoch Hinweise auf grammatikalisch mehrdeutige Instanzen oder auf potentiell fehlende Information liefern. Für die Phänomene, für welche der Einsatz von Welt- und Expertenwissen eine vergleichsweise kleine Rolle spielt, können die kritischen von den unkritischen Instanzen zuverlässig durch automatische Analysen unterschieden werden. Das bereitgestellte Wissen reicht in diesen Fällen aus.

8.8 Zusammenfassung

In diesem Kapitel wurden die Phänomene, die in Abschnitt 4.8 als relevant identifiziert wurden, analysiert. Jedes Phänomen wurde zunächst aus sprachwissenschaftlicher Sicht beschrieben und wenn möglich, wurden Regeln für die automatische Identifikation des Phänomens präsentiert.

Bei den Phänomenen *Weak-Word* und *Modalverb „sollte(n)“* können durch die präsentierten Regeln die kritischen Instanzen zuverlässig identifiziert werden. Die Analysen weisen nur eine geringe Menge an Fehlwarnungen auf und sind somit für den Einsatz in der Praxis geeignet.

Für die Phänomene *Unpersönliche Konstruktion*, *Passiv ohne Agens* und *Pronominaler Bezug* können Regeln entworfen werden, durch welche die Phänomene generell identifiziert werden können. Jedoch ist eine zuverlässige Erkennbarkeit der kritischen Fälle nicht möglich, da bei diesen Phänomenen der Einsatz von Welt- und Expertenwissen eine erhebliche Rolle spielt. Potentiell fehlende Information kann durch dieses Wissen in den meisten Fällen hergeleitet und potentiell mehrdeutige Bezüge können aufgelöst werden. Lediglich die morphosyntaktisch mehrdeutigen pronominalen Bezüge können bei einer automatischen Analyse zuverlässig identifiziert werden. Die Entscheidung, ob die Bezüge auch semantisch mehrdeutig sind, muss der Leser treffen und davon abhängig muss er entscheiden, ob die Anforderung umformuliert werden muss. Ebenso kann nur der Leser entscheiden, ob das Weglassen des Agens in einem Passivsatz als kritisch zu betrachten ist.

Die Weak-Word-Analyse und die Analyse bezüglich des Modalverbs *sollte(n)* (im Englischen *should*) lassen sich auf englischsprachige Anforderungen übertragen. Auch hier können die kritischen Instanzen zuverlässig identifiziert werden. Für die Identifikation von Passivsätzen ohne Agens können ebenfalls Regeln bereitgestellt werden. Jedoch spielt auch im Englischen der Einsatz von Welt- und Expertenwissen eine erhebliche Rolle bei der Ermittlung des Agens. Das Weglassen führt nur in sehr wenigen Fällen dazu, dass Interpretationsspielraum durch fehlende Information entsteht.

9 Technische Umsetzung und externe Bewertung der entwickelten Analysen

In Kapitel 8 wurde untersucht, für welche Phänomene sich Regeln entwickeln lassen, die zuverlässig zwischen den kritischen und unkritischen Instanzen eines Phänomens unterscheiden können. Da für die Phänomene *Weak-Word* und *Modalverb „sollte(n)“* zuverlässige Analysen bereitgestellt werden können, wurden diese in einem Daimler-internen Werkzeug, welches in der Datenbank *DOORS* vorhanden ist, integriert. Die einzelnen Verarbeitungsschritte, welche bei der Analyse durchgeführt werden, sowie die graphische Benutzeroberfläche des Werkzeugs werden in Abschnitt 9.1 beschrieben.

In Abschnitt 9.2 werden die in dieser Arbeit entwickelten Sprachprüfungen mit den in Abschnitt 5.5 präsentierten Analysen bereits existierender Werkzeuge in Vergleich gesetzt, um zu zeigen, inwiefern sich die entwickelten Analysen von den bereits existierenden abheben.

9.1 Technische Umsetzung

Die Phänomene, für welche Regeln entwickelt werden konnten, welche zuverlässig zwischen den kritischen und unkritischen Instanzen dieser Phänomene unterscheiden, wurden in ein Daimler-internes Werkzeug integriert, das im Folgenden *Lastenheft-Analyseassistent* genannt wird. Dies sind die Phänomene *Weak-Word* und *Modalverb „sollte(n)“*. Die entwickelten CQP-Regeln sind in einem XML-Dokument abgelegt und können so durch den Lastenheft-Analyseassistenten eingelesen und verarbeitet werden (mehr dazu unten).

Die einzelnen Verarbeitungsschritte, welche im Lastenheft-Analyseassistent bei der Analyse eines Phänomens durchgeführt werden, sind in Abbildung 47 zu sehen. Die zu analysierenden Anforderungen eines Lastenhefts werden zunächst aus der Datenbank *DOORS* exportiert. Anschließend werden die Anforderungen linguistisch aufbereitet, d.h. tokenisiert, lemmatisiert, getaggt und geparkt. Hierfür werden, wie in Kapitel 6 beschrieben, die Werkzeuge *MarMot* und *mate* sowie der Tokenisierer, welcher in [Kri13] entwickelt wurde, verwendet. Der nächste Schritt, welcher für die Aufbereitung der Daten dient, ist die Konvertierung der Daten in ein durch CQP-prozessierbares Format (vgl. Abschnitt 6.5). Auf die zu analysierenden Anforderungen werden anschließend die in Kapitel 8 präsentierten Regeln angewandt. Diese sind in einem XML-Dokument abgelegt und danach markiert, ob sie kritische oder unkritische Instanzen eines Phänomens identifizieren. Durch Anwendung der Regeln können die Anforderungen folglich dahingehend klassifiziert werden, ob sie kritisch oder unkritisch im Bezug auf das zu analysierende Phänomen sind. Anforde-

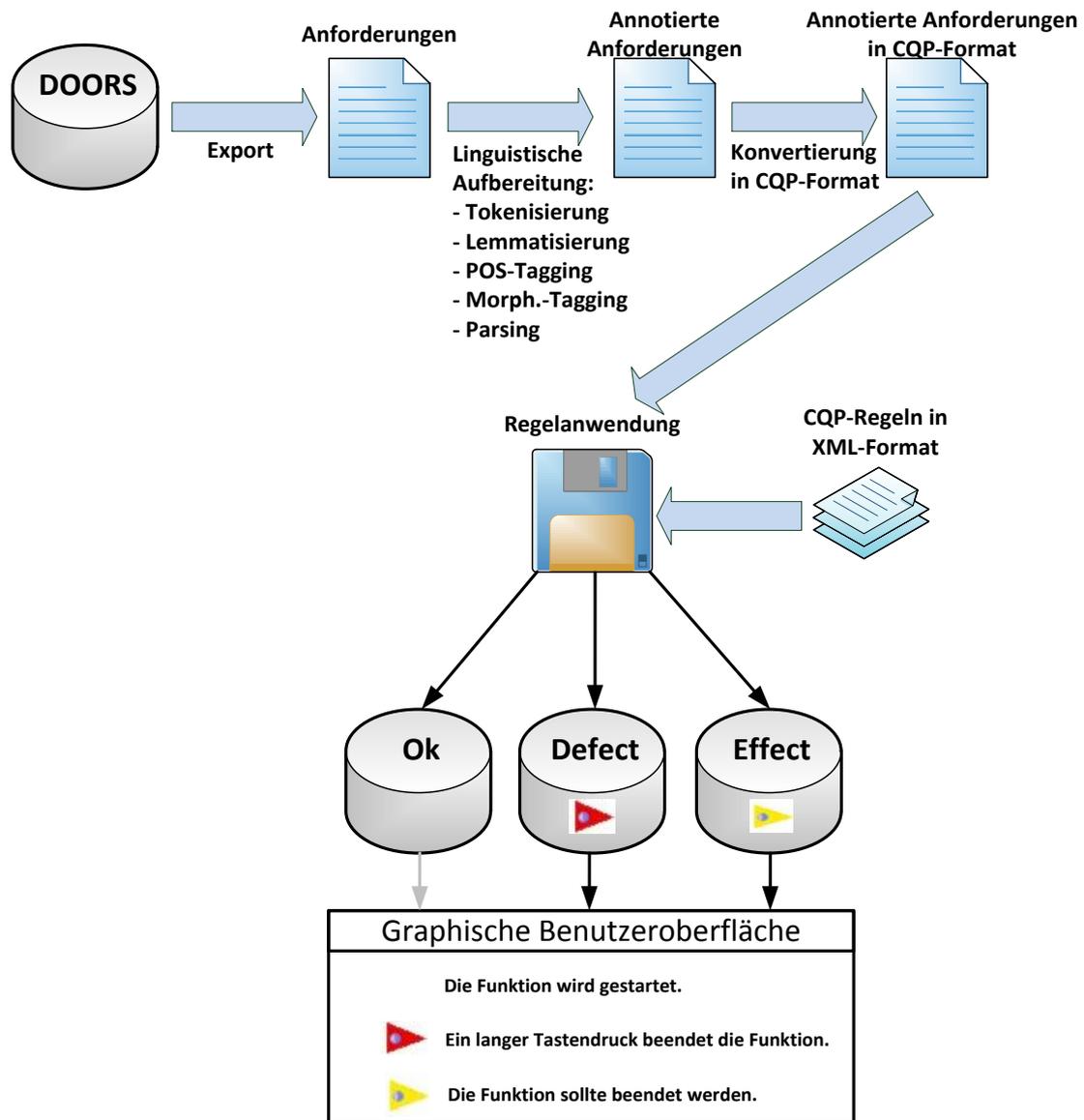


Abbildung 47: Verarbeitungsschritte im Lastenheft-Analyseassistent

rungen, welche unkritisch sind, werden der Kategorie *Ok* zugeordnet. Anforderungen, die kritisch sind, fallen in die Kategorie *Defect* oder *Effect*.

Kritische Instanzen, die in die Kategorie *Defect* fallen, entsprechen Instanzen, die bei einer automatischen Analyse dem Autor zurückgeliefert werden müssen, mit dem Hinweis, dass die Textstelle überarbeitet werden muss, da entweder Information fehlt oder eine Mehrdeutigkeit vorliegt. In die Kategorie *Effect* fallen hingegen die Instanzen, bei welchen nicht mit Sicherheit eine Aussage getroffen werden kann, ob Information fehlt bzw. ob eine mehrdeutige Struktur vorliegt (vgl. Kapitel 7). Die Kategorie *Effect* dient dazu, dem

Autor den Hinweis zu geben, er möge überprüfen, ob die Information, auf die verwiesen wird, auch wirklich im Anforderungsdokument oder einem beigelegten Dokument genannt wird.

Dem Benutzer werden die Analyseergebnisse in einer graphischen Benutzeroberfläche präsentiert, welche in Abbildung 48 zu sehen ist.

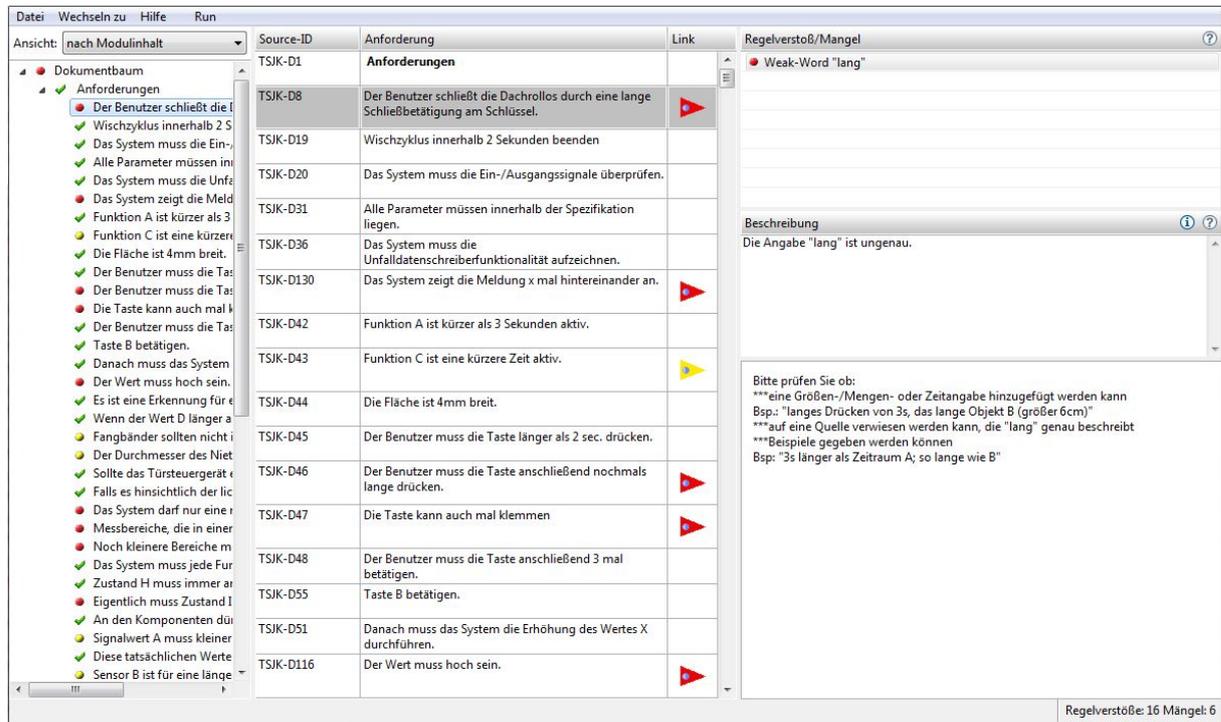


Abbildung 48: Bewertung von Anforderungen im *Lastenheft-Analyseassistenten*

Die graphische Benutzeroberfläche besteht aus drei Teilen. Auf der linken Seite werden die Anforderungen aufgelistet und es ist farblich markiert, welcher Kategorie (*Ok*, *Defect*, *Effect*) sie zugeordnet werden. In der Mitte sind die Anforderungen vollständig zu sehen. Die Anforderungen, die der Klasse *Defect* zugeordnet werden, sind durch rote Pfeile markiert. Die Anforderungen, die in die Klasse *Effect* fallen, hingegen durch gelbe Pfeile. Anforderungen, die in die Klasse *Ok* fallen, werden nicht markiert.

Zu jeder Anforderung, die als *Defect* oder *Effect* klassifiziert ist, wird eine Warnmeldung ausgegeben, die auf das jeweilige Problem angepasst ist. Diese Warnmeldung wird auf der rechten Seite der graphischen Benutzeroberfläche dargestellt. In Abbildung 48 ist eine Fehlermeldung, welche bei kritischen Instanzen des Weak-Word-Lemmas *lange* verwendet wird, zu sehen. Die Warnmeldung enthält zum einen eine Erklärung dazu, wieso die Verwendung des Weak-Words problematisch ist. Weiterhin werden Beispiele gezeigt, in welchen Kontexten die Verwendung des Wortes unkritisch ist. Dies soll dem Autor hel-

fen, die entsprechende Anforderung so anzupassen, dass der Gebrauch des Weak-Words unkritisch ist. Weitere Warnmeldungstexte sind in Anhang F aufgelistet.

Im Lastenheft-Analyseassistent kann eine Sortierung der Anforderungen nach Kategorie vorgenommen werden: Der Benutzer kann sich so gesammelt alle *Defects* und/oder *Effects* anzeigen lassen und die unkritischen Anforderungen werden ausgeblendet. Somit kann der Benutzer einen Überblick darüber erhalten, welche Anforderungen nochmals überarbeitet werden müssen.

Im Lastenheft-Analyseassistent werden die Phänomene *Weak-Word* und *Modalverb „sollte(n)“* analysiert. Die Integration weiterer Analysen kann problemlos vorgenommen werden. Die Voraussetzung hierfür ist, dass die Regeln in CQP-Format vorliegen und die Regeln nur Attribute enthalten, welche durch die Annotationswerkzeuge annotiert werden und somit abgefragt werden können. So können bei Bedarf beispielsweise die Regeln zur Passiverkennung in das Werkzeug integriert werden. Auch weitere Analysen sind denkbar.

9.2 Externe Bewertung

In Abschnitt 5.5 wurden vier Werkzeuge aus dem Bereich des Requirements Engineering und vier Werkzeuge aus dem Bereich der Sprachtechnologie vorgestellt, welche natürlichsprachliche Texte auf Verständlichkeit überprüfen. Im vorliegenden Abschnitt wird die Güte der in dieser Arbeit entwickelten Analysen mit der Güte der Analysen der präsentierten Werkzeuge verglichen.

Phänomen	Relevanz	Erkennbarkeit generell								Kritische Fälle zuverlässig erkennbar									
		Entwickelte Analysen	DESIRE®	ReqCheck	Requirements Quality Suite	Requirements Scout	LanguageTool	TextLab	CLAT	Acrolinx	Entwickelte Analysen	DESIRE®	ReqCheck	Requirements Quality Suite	Requirements Scout	LanguageTool	TextLab	CLAT	Acrolinx
Weak-Word	hoch	↑	↑	↓	↑	↑	↓	↓	↓	↓	↑	↓	↓	↓	↓	↓	↓	↓	↓
Unpersönliche Konstruktion	niedrig	↑	↓	↑	↓	↑	↓	→	↑	→	→	↓	↓	↓	↓	↓	↓	↓	↓
Passiv	niedrig	↑	↑	↑	↑	↑	↑	↑	↑	↑	→	↓	↓	→	→	↓	↓	→	→
Modalverb „sollte(n)“	hoch	↑	↑	↑	↑	↑	↓	↑	↑	↑	↑	↓	↓	↓	↓	↓	↓	↓	↓
Pronominaler Bezug	niedrig	↑	↓	↓	↑	↑	↓	↓	↑	→	→	↓	↓	↓	↓	↓	↓	↓	↓

Tabelle 46: Evaluierungsergebnisse der Analysen bestehender Textprüfwerkzeuge an ausgewählten Phänomenen aus dem Qualitätsmodell (grün (↑) = ja, gelb (→) = teilweise, rot (↓) = nein)

In Tabelle 46 sind, wie bereits in Abschnitt 5.5.3.2 präsentiert, die untersuchten Werkzeuge aufgelistet. Für jedes Werkzeug ist durch eine farbliche Markierung der Zellen angegeben, welche Phänomene generell analysiert werden, und ob die kritischen Fälle der Phänomene auch zuverlässig erkannt werden. Zusätzlich wird aufgeführt, wie sich die in dieser Arbeit entwickelten Analysen im Vergleich darstellen. In Tabelle 46 wird zusätzlich die Relevanz jedes Phänomens markiert. Die Analysen zu den Phänomenen *Unpersönliche Konstruktion*, *Passiv* und *Pronominaler Bezug* haben ergeben, dass diese Phänomene nur in sehr seltenen Fällen kritisch sind. Die potentiell fehlende Information bzw. die potentielle Mehrdeutigkeit kann in den meisten Fällen durch Welt-, Experten- oder Kontextwissen aufgelöst werden. Aus diesem Grund wird diesen Phänomenen eine niedrigere Relevanz zugeordnet. Den Phänomenen *Weak-Word* und *Modalverb „sollte(n)“* wird eine höhere Relevanz zugeordnet, weil bei diesen Phänomenen Experten- und Weltwissen zur Beseitigung eventueller Interpretationsprobleme nicht ausreicht.

Die kritischen Instanzen des Phänomens *Weak-Word* werden durch die in dieser Arbeit entwickelten Analysen zuverlässig erkannt. Es entstehen bei der Anwendung der präsentierten Regeln nur wenige Fehlwarnungen und Fehlentwarnungen. Die Werkzeuge aus dem Requirements Engineering, ausgenommen von *ReqCheck*, analysieren ebenfalls dieses Phänomen. Jedoch werden die Analysen nicht kontextabhängig durchgeführt, es handelt sich lediglich um Wortlistenabgleiche. Da aber nicht jede Verwendung eines Weak-Words als kritisch zu bewerten ist, liefert ein solcher Ansatz eine für Anwender sehr störend hohe Anzahl an Fehlwarnungen.

Keines der Werkzeuge aus der Sprachtechnologie analysiert das Phänomen *Weak-Word*. Die potentielle Gefahr des Gebrauchs von Weak-Words stellt ein bekanntes Problem im Bereich des Requirements Engineering dar. Im Bereich der technischen Dokumentation hat die Behandlung dieses Phänomens vermutlich eine geringere Relevanz, und das Phänomen wird daher nicht behandelt.

Das Phänomen *Unpersönliche Konstruktion* (Instanzen der Pronomina *es* und *man*) wird durch die Werkzeuge *ReqCheck*, *Requirements Scout* und *CLAT* behandelt. *TextLab* und *Acrolinx* behandeln lediglich das Pronomen *man* (= gelbe Markierung). Die Werkzeuge erzeugen bei jeder Verwendung eines dieser Wörter eine Warnung an den Benutzer. Es handelt sich auch hier um reine Wortlistenabgleiche. In Abschnitt 8.2 wurde jedoch gezeigt, dass lediglich agenslose Passivsätze kritische Instanzen darstellen. Auch hier reicht ein wortlistenbasierter Ansatz nicht aus, um die kritischen Instanzen zuverlässig erkennen zu können, ohne zu viel Fehlalarm zu erzeugen. Durch die in dieser Arbeit entwickelten Analysen können die agenslosen Passivsätze automatisiert erkannt werden. Jedoch haben die Analysen zum Passiv gezeigt, dass dieses Phänomen nur in sehr seltenen Fällen

kritisch ist und die potentiell fehlende Agens-Information in den meisten Fällen durch Welt-, Experten- und Kontextwissen hergeleitet werden kann. Die in der vorliegenden Arbeit entwickelten Analysen unterscheiden sich von den existierenden Analysen insofern, als dass nur die Sätze als Warnung ausgegeben werden, welche das Pronomen *es* in der Verwendung eines Vorfeld-*es* in einem agenslosen Passivsatz enthalten. Die kritischen Instanzen können jedoch nicht zuverlässig erkannt werden, da der Großteil der Instanzen des unpersönlichen *es* unkritisch ist.

Das Phänomen *Passiv* wird von allen Werkzeugen behandelt. Insgesamt vier Werkzeuge (*Requirements Quality Suite*, *Requirements Scout*, *CLAT* und *Acrolinx*) analysieren Passivsätze auch dahingehend, ob ein Agens im Satz genannt wird und geben nur dann eine Warnung aus, wenn das Agens ungenannt bleibt. Die Werkzeuge *DESIRE*[®], *ReqCheck*, *LanguageTool* und *TextLab* geben bei jeder Instanz eines Passivsatzes eine Warnung aus, unabhängig davon, ob das Agens genannt wird oder nicht. Der in dieser Arbeit entwickelte Ansatz unterscheidet ebenfalls zwischen Passivsätzen mit und ohne Agens. Da aber das Weglassen des Agens im Bereich der Lastenhefttexte nur in sehr seltenen Fällen dazu führt, dass die potentiell fehlende Information nicht durch Welt-, Experten- oder Kontextwissen hergeleitet werden kann, können Analysen, die die kritischen Fälle zuverlässig erkennen, nicht entworfen werden. Es können lediglich die Passivsätze ohne Agens herausgefiltert werden und dem Benutzer zur Überprüfung zurückgegeben werden. Der entwickelte Ansatz hebt sich hier nicht von den bereits existierenden Analysen ab.

Das Phänomen *Modalverb „sollte(n)“* wird durch alle Werkzeuge, ausgenommen von *LanguageTool* behandelt. Allerdings handelt es sich bei den Analysen der Werkzeuge um reine Wortlistenabgleiche, d.h. bei jeder Instanz von *sollte(n)* wird eine Warnung ausgegeben. Da aber nur Instanzen kritisch sind, bei welchen eine Mehrdeutigkeit hinsichtlich der deontischen und der dispositionellen Lesart vorliegt und die Potentialisfälle von einer automatischen Analyse als unproblematisch kategorisiert werden müssen, reicht ein wortlistenbasierter Ansatz nicht aus. Die Werkzeuge behandeln zwar das Phänomen, jedoch werden die kritischen Instanzen nicht zuverlässig von den unkritischen Instanzen abgegrenzt. Der in dieser Arbeit entwickelte Ansatz identifiziert die Verwendungen von *sollte(n)*, bei welchen eine Ambiguität zwischen der deontischen und der dispositionellen Lesart vorliegt und schließt die Potentialisfälle aus. Die entwickelten Analysen identifizieren zuverlässig die kritischen und unkritischen Instanzen und heben sich in diesem Punkt von den existierenden Werkzeugen ab.

Pronominale Bezüge werden durch die Werkzeuge *Requirements Quality Suite*, *Requirements Scout*, *CLAT* und *Acrolinx* analysiert. Bis auf *Acrolinx*, welches lediglich substituierende Demonstrativpronomina analysiert, behandeln die genannten Werkzeuge alle

Klassen von pronominalen Bezügen. Die Werkzeuge führen auch bei der Analyse dieses Phänomens Wortlistenabgleiche durch und führen keine Analyse dahingehend durch, ob es mehrere mögliche Nomina gibt, auf welche sich das Pronomen beziehen kann, weil es morphologisch mit mehreren Nomina kongruiert. Der in dieser Arbeit entwickelte Ansatz verwendet Regeln, welche in Abschnitt 8.5 präsentiert wurden. Anhand dieser Regeln wird geprüft, ob ein morphosyntaktisch mehrdeutiger Bezug vorliegt. Jedoch bedeutet ein morphosyntaktisch mehrdeutiger Bezug nicht gleich, dass auch ein semantisch mehrdeutiger Bezug vorliegt, da in vielen Fällen Welt- und Expertenwissen dabei hilft die potentielle Ambiguität aufzulösen. Der entwickelte Ansatz hebt sich insofern von den bereits existierenden Werkzeugen ab, als dass nicht bei jedem Auftreten eines Pronomens eine Warnung ausgegeben wird, sondern nur bei den morphosyntaktisch mehrdeutigen pronominalen Bezügen. Eine zuverlässige Analyse der kritischen Instanzen kann aber auch hier nur teilweise gewährleistet werden.

Für die fünf als relevant identifizierten Phänomene konnten Analysen entwickelt werden, welche die Phänomene generell erkennen. Für die Phänomene *Weak-Word* und *Modalverb sollte(n)* konnten zudem Regeln formuliert werden, welche die kritischen Fälle zuverlässig erkennen und dabei wenig Fehlalarm produzieren. Die entwickelten Regeln heben sich von den Analysen der bereits existierenden Werkzeuge insofern ab, als dass alle als relevant identifizierten Phänomene behandelt werden und zudem, wo möglich, zuverlässige Analysen bereitgestellt werden.

9.3 Zusammenfassung

Im vorliegenden Kapitel wurde das Werkzeug präsentiert, in welches die in dieser Arbeit entwickelten Sprachprüfungen integriert wurden. Es handelt sich um ein Daimler-internes Werkzeug, welches Lastenhefte, die in der Datenbank DOORS abgelegt sind, analysieren kann. Der Text aus einem zu analysierenden Lastenheft wird zunächst exportiert, tokenisiert und linguistisch annotiert. Die Texte werden anschließend in ein durch CQP prozessierbares Format umgewandelt, auf welches die Regeln angewandt werden können. Der Autor bekommt eine Übersicht mit den potentiellen Schwachstellen in seiner Spezifikation gezeigt und kann darüber entscheiden, ob er die betroffenen Anforderungen ändern möchte. Dem Benutzer werden nicht nur die zu überarbeitenden Anforderungen zurückgeliefert, es wird auch eine Warnmeldung ausgegeben, welche auf das jeweilige Problem angepasst ist.

In diesem Kapitel wurde weiterhin gezeigt, inwiefern sich die in dieser Arbeit entwickelten Analysen von Analysen bereits existierender Werkzeuge abheben. Die Analysen der bereits existierenden Werkzeuge basieren größtenteils auf Wortlistenabgleichen. Ein wort-

listenbasierter Ansatz liefert jedoch eine große Anzahl an Fehlwarnungen. Die in dieser Arbeit entwickelten Analysen gehen über Worlistenabgleiche hinaus und führen eine kontextabhängige Analyse durch. Für die fünf als relevant identifizierten Phänomene konnten Analysen entwickelt werden, welche die Phänomene generell erkennen. Für die Phänomene *Weak-Word* und *Modalverb sollte(n)* konnten zudem Regeln formuliert werden, welche die kritischen Fälle zuverlässig von den unkritischen Fällen unterscheiden. Die Fehlwarnungen können hier deutlich reduziert werden. Der in dieser Arbeit präsentierte Ansatz hebt sich von den Analysen bereits existierender Werkzeuge insofern ab, dass zum einen alle relevanten Phänomene analysiert werden und wo möglich effektive Analysen bereitgestellt werden.

10 Interpretation der empirischen Ergebnisse

Die Evaluierungsergebnisse aus Kapitel 8 haben gezeigt, dass für ausgewählte Phänomene Analysen entworfen werden können, welche zuverlässig die kritischen und die unkritischen Instanzen eines Phänomens identifizieren. Für andere Phänomene hat sich wiederum ergeben, dass zwar die grammatikalisch mehrdeutigen Instanzen automatisiert identifiziert werden können, jedoch Welt-, Experten- und Kontextwissen in den meisten Fällen dabei helfen die potentiell fehlende Information herzuleiten oder die potentielle Mehrdeutigkeit aufzulösen.

Die empirischen Ergebnisse unterliegen allerdings gewissen Gefährdungen der Validität. Diese potentiellen Gefährdungen werden im vorliegenden Kapitel beschrieben, und es wird gezeigt, wie diesen Gefährdungen entgegengewirkt wird.

10.1 Interne Validität

Eine Gefährdung der internen Validität stellen Störeinflüsse, wie zum Beispiel die Fehleranfälligkeit der Annotationswerkzeuge, aber auch Grammatik- und Rechtschreibfehler in den Anforderungen dar. Diese Faktoren beeinflussen die Ergebnisse zu den präsentierten Phänomenanalysen. Ist eine Analyse aufgrund einer derart "fehlerhaften" Instanz fehlgeschlagen, wurde dies in die Ergebnisse als Fehler der Gesamtanalyse mit eingerechnet. Dies bedeutet, dass die Analysen auf Grundlage von grammatikalisch korrekten Sätzen und durchweg korrekten Annotationen noch bessere Ergebnisse liefern würden.

Eine weitere Gefährdung der internen Validität ist die Subjektivität der durchgeführten Bewertungen. Bei der Durchführung der Phänomenanalysen wurden die relevanten Textstellen automatisiert und manuell bewertet. Die manuelle Bewertung wurde durch die Autorin selbst vorgenommen. Hierbei kann der Eindruck entstehen, dass die Bewertungen einen subjektiven Charakter haben, weil die bewertende Person einerseits sehr kulant sein kann oder sehr streng. Gängige Praxis der Computerlinguistik ist die Bewertung von Analysen durch mehrere Personen und die Ermittlung eines Inter-Rater-Agreements. Angesichts der quantitativen Dimension der Datenbewertungen (in den Arbeiten zu Kapitel 8 wurden insgesamt 23.688 Belege analysiert) erwies sich dieser Ansatz für die vorliegende Arbeit als nicht realisierbar. Um zu zeigen, dass die Subjektivität der Bewertung einen verhältnismäßig kleinen Einfluss auf die interne Validität der Forschungsergebnisse hat, wurden zwei stichprobenweise Studien durchgeführt. In der ersten Studie wurde mittels Fleiss' Kappa errechnet, wie gut sich die Bewertungen der Autorin mit anderen Bewertern überdeckt. Die zweite Studie wurde im Zuge des bereits in Abschnitt 1.1 präsentierten Fragebogens durchgeführt, in welchem Tester neben den allgemeinen Fragen

natürlichsprachliche Anforderungen nach ihrer Qualität bewerten sollten und die Ergebnisse mit der Einschätzung der Autorin abgeglichen wurden. Auf die beiden Studien wird im Folgenden einzeln eingegangen.

10.1.1 Studie 1: Fleiss' Kappa

Im Zusammenhang mit den Passivanalysen für englischsprachige Lastenhefte (vgl. Abschnitt 8.6) wurde ein Experiment durchgeführt, um Fleiss' Kappa [Fle71] zu berechnen. Fleiss' Kappa ist ein Maß, mit welchem man berechnen kann, ob mehrere Beurteiler eine hohe Bewertungsüberdeckung haben, d.h. ob die Personen in den meisten Fällen bei der Bewertung bestimmter Instanzen der selben Meinung sind (= Inter-Rater-Agreement). Dieses Experiment wurde mit drei Personen durchgeführt. Eine dieser drei Personen ist die Autorin, die die in dieser Arbeit präsentierten Analysen bewertet. Die beiden anderen Beurteiler sind Personen, die im Alltag mit natürlichsprachlichen Anforderungen arbeiten. Die Bewerber haben 150 Anforderungen, welche Passivkonstruktionen ohne Angabe eines Agens enthalten, dahingehend bewertet, ob die Verwendung des Passivs zu Verständnisproblemen führt, weil das Agens nicht hergeleitet werden kann. In der ersten Phase wurden die Anforderungen ohne Einbezug von Kontextinformation bewertet, d.h. die Anforderungen wurden isoliert von den restlichen Anforderungen betrachtet. In der zweiten Phase durften die Personen den Kontext, d.h. die anderen Anforderungen, im Anforderungsdokument vorhandene Tabellen, Bilder etc. miteinbeziehen und die Anforderungen dementsprechend bewerten. Für jede Anforderung mussten die Bewerber eine Entscheidung treffen, ob die Verwendung des Passivs dazu führt, dass das Agens des Satzes nicht hergeleitet werden kann (Kategorie: *kritisch*), oder ob die Verwendung des Passivs in dieser Hinsicht unproblematisch ist (Kategorie: *unkritisch*).

Bei der Betrachtung der Anforderungen ohne Einbezug von Kontextwissen, wurden 85 von 150 Anforderungen von allen Bewertern als *unkritisch* kategorisiert. Dass das Agens in diesen Anforderungen ungenannt bleibt, wurde von den Bewertern nicht als kritisch empfunden. 46 Anforderungen wurden von allen Bewertern als *kritisch* kategorisiert. Acht Anforderungen wurden von zwei Personen als *kritisch* und von einer Person als *unkritisch* bewertet. Außerdem wurden elf Anforderungen von einer Person als *kritisch* bewertet und von den beiden anderen Personen als *unkritisch*.

Die Anforderungen, die als kritisch bewertet wurden, waren in den meisten Fällen Anforderungen, in welchen es um Signaltransformationen oder Signalübertragungen geht. In Beispiel 131 und Beispiel 132 sind solche Anforderung zu sehen.

Beispiel 131 *The CAN signals from engine and alternator are converted to some bitwise information for the other functions.*

Beispiel 132 *If during that time the car is unlocked again, X is set to 1 and the function waits for the car to be locked again.*

Ohne zusätzliche Information (Tabellen, andere Anforderungen) ist es nicht möglich herzuleiten, ob der Auftragnehmer in der Pflicht steht die Anforderung zu realisieren oder ob es sich bei der Anforderung lediglich um eine Zusatzinformation handelt. Weiterhin wird keine Aussage darüber gemacht, ob die Signale produziert oder konsumiert werden.

Eine weitere von allen Beurteilern als kritisch bewertete Anforderung ist in Beispiel 133 zu sehen. Auch hier kann nicht hergeleitet werden, ob der Auftragnehmer in der Pflicht steht, die in der Anforderung beschriebene Funktionalität zu realisieren, oder ob es sich lediglich um eine Zusatzinformation handelt.

Beispiel 133 *The information if the car is locked (inverted signal U) is stored in a flag, needing 1 Bit of memory.*

Eine Anforderung, bei welcher die Bewertungen nicht einheitlich waren, ist in Beispiel 134 zu sehen. Für zwei Bewerter war es offensichtlich, dass diese Anforderung durch den Auftragnehmer realisiert werden muss. Für den dritten Bewerter war dies nicht so offensichtlich.

Beispiel 134 *All controls of the user interface shall be in a position where they are not touched accidentally when putting the headphones on or off.*

Zählt man die Übereinstimmungen und Nicht-Übereinstimmungen der drei Bewerter zusammen, ist das Ergebnis wie folgt: In 131 Fällen stimmen die Bewerter miteinander überein, in 19 Fällen gehen die Meinungen auseinander. Anhand dieser Ergebnisse konnte Fleiss' Kappa berechnet werden. Der Kappa-Wert beträgt 0.81, was einer sehr hohen Übereinstimmung entspricht [LK77].

Mit Einbezug von Kontextinformation ist die Überdeckungsrate noch höher. Lediglich eine Anforderung wurde von zwei Bewertern als *kritisch* bewertet. Diese Anforderung ist in Beispiel 135 zu sehen.

Beispiel 135 *Corrosion tests are carried out according to the standards which are used.*

Im gesamten Anforderungsdokument, in welchem diese Anforderung verortet ist, konnte keine Information darüber gefunden werden wer die *Korrosionstests* ausführen muss. Eine Person bewertete die Anforderung als *unkritisch*, weil die vorausgehenden Anforderungen alle an den Auftragnehmer gerichtet waren. Für die anderen beiden Personen war

dies nicht offensichtlich. Die weiteren 149 Anforderungen wurden von allen Bewertern als *unkritisch* kategorisiert.

Das Experiment zu Fleiss' Kappa zeigt, dass die Analysen der Autorin mit denen von anderen Bewertern zum Großteil übereinstimmen. Somit ist der Einfluss auf die interne Validität der empirischen Ergebnisse sehr gering.

10.1.2 Studie 2: Fragebogen

Ein weiteres Experiment, welches ebenfalls durchgeführt wurde, um zu zeigen, dass die subjektive Bewertung der Autorin nur einen geringen Einfluss auf die Validität der Ergebnisse hat, ist der bereits in Abschnitt 1.1 eingeführte Fragebogen. Zusätzlich zu den in Abschnitt 1.1 präsentierten Fragen, wurden den Testern 17 Anforderungen vorgelegt, welche sie bewerten sollten. Für die Bewertung standen ihnen vier Kategorien zur Verfügung:

- ⊕₂ *Diese Formulierung wird für den Test keine Probleme machen.*
- ⊕₁ *Die Anforderung ist zwar mehrdeutig, es weiß aber jeder Tester wie dieser Sachverhalt getestet werden muss.*
- ⊖₁ *Die Anforderung ist zwar mehrdeutig, aber ich weiß aufgrund von Erfahrung wie ich das testen muss. Das weiß aber nicht jeder Tester.*
- ⊖₂ *Diese Formulierung wird für den Test Probleme machen. Ich weiß nicht wie ich das testen soll ohne nachzufragen oder nachzuschauen.*

Die Autorin hat ebenfalls diese Bewertung vorgenommen. Die Gesamtergebnisse sind in Abbildung 49 zu sehen.

In Abbildung 49 ist die durchschnittliche Bewertung der Tester pro Anforderung zu sehen (jeweils markiert in Spalte "T") und die Einschätzung der Autorin (jeweils markiert in Spalte "R"). Die Durchschnittswerte der Tester und die Einschätzung der Autorin sind durch das Zeichen \oplus markiert. Die Einschätzungen der beiden Versuchsgruppen liegen in den meisten Fällen nah beieinander, lediglich bei den Anforderungen A5 und A10 gehen die Meinungen stärker auseinander. Auf die Gründe wird weiter unten eingegangen. Zusätzlich zu den Durchschnittswerten ist die Streuung der Bewertungen zu sehen. Die farbigen Boxen stellen die Interquartilsbereiche dar, d.h. innerhalb dieser Box liegen die mittleren 50% der Bewertungen der Tester. Die Striche, die an den Boxen anliegen, sind der obere und untere Whisker. Diese gehen bis zum maximalen bzw. minimalen angegebenen Wert, jedoch nicht weiter als bis zur 1,5-fachen Länge des Interquartilbereichs. Die Werte, die außerhalb der Whisker liegen, stellen Ausreißer dar.

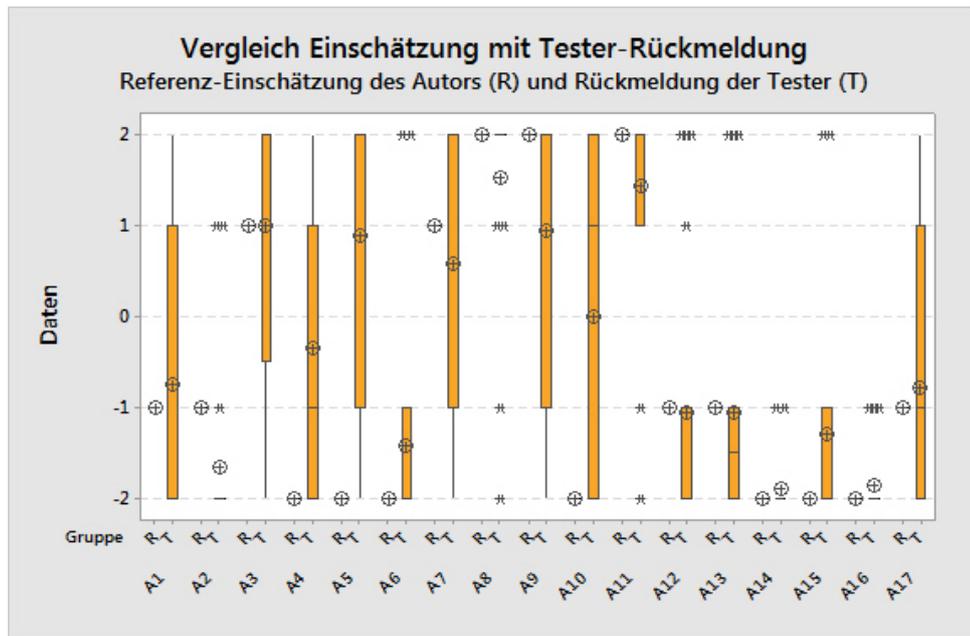


Abbildung 49: Fragebogen: Auswertung der Rückmeldungen zu 17 Anforderungen

Bei Anforderung 5 handelt es sich um den bereits aus Abschnitt 4.4 bekannten Widerspruch aus Beispiel 58, welcher in Beispiel 136 nochmals dargestellt ist.

Beispiel 136 *Kommt innerhalb von 2 Sekunden eine weitere steigende Flanke von A, wird die Funktion sofort abgebrochen und neu gestartet (D.h. die Paniktaste wird gedrückt. Bei < 1 Sekunde passiert nichts, bei ≥ 1 Sekunde wird die Funktion abgebrochen und neu gestartet).*

Nur wenige der Tester haben den lokalen Widerspruch erkannt und die Anforderung dementsprechend bewertet (mit -1 oder -2). Der Widerspruch kann beim schnellen Lesen leicht übersehen werden. Möglicherweise kommen so die Bewertungen der Tester zustande. Anforderung 10 ist in Beispiel 137 dargestellt.

Beispiel 137 *Ist die Prüfspannung kleiner der Bordnetzspannung, so muss das System sie mittels einer Pulsweitenmodulation begrenzen.*

Die Autorin hat die Anforderung als problematisch kategorisiert und somit mit -2 bewertet, weil für die Autorin nicht eindeutig war, auf welches Nomen (*Prüfspannung* oder *Bordnetzspannung*) sich das Pronomen *sie* bezieht. Ein großer Teil der Tester hat wohl vor der Fragebogenaktion schon einmal das fragliche System getestet oder besitzt das notwendige Technikwissen und hatte somit mit der Auflösung des Pronomens kein Problem.

Insgesamt kann man sehen, dass die Einschätzung der Autorin größtenteils mit der Einschätzung der Tester korrespondiert. Somit kann die Aussage, dass die Subjektivität der

Bewertungen kaum einen Einfluss auf die interne Validität der Analysen hat, nochmals unterstrichen werden.

10.2 Externe Validität

Es existieren neben den internen auch externe Gefährdungen der Validität. Die Untersuchungen in der vorliegenden Arbeit wurden an Daimler-internen deutschen und zum Teil an englischen Anforderungsdokumenten in natürlicher Sprache durchgeführt. Die Aussage, dass die Ergebnisse auf alle Arten von Anforderungsdokumenten übertragen werden können und dabei dieselbe Analysequalität erreicht wird, kann nicht generell getroffen werden. Jedoch kann die Aussage getroffen werden, dass bei ähnlichen Bedingungen die Analysen gute Ergebnisse liefern sollten. Mit ähnlichen Bedingungen ist beispielsweise die Tatsache gemeint, dass Anforderungen von einem Auftraggeber für einen Auftragnehmer geschrieben werden. Die kontextabhängige Weak-Word-Analyse ist unabhängig von der Textsorte, denn Wörter wie beispielsweise *lang* ohne Angabe einer Quantifizierung haben immer einen subjektiven Charakter, unabhängig von der betrachteten Textsorte.

Über die Übertragbarkeit auf andere Sprachen kann ebenfalls keine Aussage getroffen werden. Die Analysen basieren auf deutschen und zum Teil englischen Texten. Ob die Analysen auch auf andere Sprachen übertragen werden können und dabei eine entsprechende Analysequalität erreicht wird, kann nicht garantiert werden. Für weitere Sprachen müssten neue Untersuchungen dahingehend durchgeführt werden, ob generell dieselben Phänomene zu Problemen führen oder ob andere Phänomene prominenter sind. Weiterhin müsste analysiert werden, ob sich die Regeln für deutsche Lastenhefte auf andere Sprachen ohne Änderung übertragen lassen, oder ob hierfür neue Regeln entwickelt werden müssen.

10.3 Zusammenfassung

Es wurden interne und externe Gefährdungen für die Validität identifiziert. Eine Gefährdung der internen Validität stellen die Fehleranfälligkeit der Annotationswerkzeuge und mögliche Rechtschreib- und Grammatikfehler in den Anforderungen dar. Liegen grammatikalisch korrekte Sätze ohne Rechtschreibfehler vor und sind die annotierten Informationen der Werkzeuge durchgehend korrekt, können die präsentierten Analysen noch bessere Ergebnisse liefern als die berichteten Ergebnisse.

Eine weitere Gefährdung der internen Validität ist die Subjektivität der Bewertungen. Die Analysen in Kapitel 8 wurden von einer Einzelperson durchgeführt. Um zu zeigen, dass die Subjektivität der Bewertungen nur einen sehr geringen Einfluss auf die Validität der empirischen Ergebnisse hat, wurden zwei Studien durchgeführt. In der ersten Studie wurde Fleiss' Kappa berechnet, wobei die Bewertungen der Autorin mit zwei weiteren

Bewertern abgeglichen wurden. Die errechneten Werte sind sehr hoch und entsprechen einer fast perfekten Übereinstimmung. In der zweiten Studie haben 20 Tester und die Autorin 17 Anforderungen nach ihrer Qualität bewertet. Auch hier zeigt sich, dass die Einschätzung der Autorin nahe an der Einschätzung der Tester liegt. Beide Ansätze zur Sicherstellung der internen Validität zeigen, dass die Subjektivität der Bewertungen eine vergleichsweise kleine Rolle spielt.

Die Ergebnisse der Phänomenanalysen können nicht problemlos auf jede Art natürlichsprachlicher Texte angewandt werden. Hierbei muss es sich um deutsche natürlichsprachliche Anforderungen handeln, welche von einem Auftraggeber für einen Auftragnehmer geschrieben werden. Die Analyse zu den Weak-Words lässt sich auf nahezu jede Art von Text übertragen, da Weak-Words nicht nur im Kontext von natürlichsprachlichen Anforderungen zu Verständnisproblemen führen können. Um sicherstellen zu können, dass die Analysen auch auf andere Sprachen übertragen werden können, müssten weitere Untersuchungen durchgeführt werden.

11 Zusammenfassung und Ausblick

In diesem Kapitel werden die Ergebnisse und die Hauptbeiträge der Arbeit zur Forschungsdiskussion nochmals zusammengefasst (vgl. Abschnitt 11.1), und ein Ausblick auf mögliche weiterführende Arbeiten wird präsentiert (vgl. Abschnitt 11.2).

11.1 Zusammenfassung der Beiträge und Ergebnisse

Die natürliche Sprache ist noch immer das vorherrschende Hilfsmittel, um Anforderungen zu formulieren. Anforderungen definieren die Funktionen von Komponenten und Systemen in der Softwareentwicklung und bilden die Kommunikationsbasis zwischen Auftraggeber und Auftragnehmer. Sie stellen zum einen Vorgaben an die Software, welche entwickelt werden soll, und bilden zum anderen die Grundlage für die Ableitung von Testfällen. Durch Testfälle kann überprüft werden, ob ein entwickeltes System oder eine entwickelte Komponente die in den Anforderungen geforderten Funktionalitäten erfüllt, d.h. ob das Ist-Ergebnis dem in den Anforderungen beschriebenen Soll-Ergebnis entspricht. Da die Aufwendungen für die Testaktivitäten mittlerweile 25%-50% der Entwicklungszeit und Entwicklungskosten einnehmen, kommt dieser Aktivität eine besonders große Bedeutung zu. Die Anforderungen bilden die Grundlage für die Ableitung von Testfällen. Deshalb ist es besonders wichtig, dass die Anforderungen eine hohe Qualität aufweisen und zudem testbar formuliert sind.

Den ersten Beitrag zur Forschung liefert diese Arbeit insofern, als dass untersucht wurde, welche Eigenschaften eine Anforderung erfüllen muss, um als testbar zu gelten. Hierfür wurde ein Qualitätsmodell für testbare Anforderungen entwickelt, welches sprachliche und inhaltliche Phänomene aus der Literatur zum Requirements Engineering und Phänomene aus der Literatur zur technischen Dokumentation zusammenträgt. Die Phänomene sind in diesem Modell nach linguistischen Aspekten sortiert und nach Kritikalität bewertet. Laut diesem Modell setzt sich die Testbarkeit einer Anforderung aus sechs Teilkriterien zusammen: *Messbarkeit*, *Verständlichkeit*, *Atomarität*, *Widerspruchsfreiheit*, *Inhaltliche Korrektheit* und *Durchführbarkeit*. Diesen Kriterien sind die Phänomene aus der Literatur zugeordnet, sodass ein Überblick darüber gegeben wird, welches Phänomen dazu führen kann, dass ein bestimmtes Kriterium nicht erfüllt wird somit die Testbarkeit nicht sichergestellt ist.

Einen weiteren Beitrag liefert die vorliegende Arbeit dahingehend, dass untersucht wurde, inwiefern die Phänomene, welche in der Literatur zum Requirements Engineering und in der Literatur zur technischen Dokumentation genannt werden, sich überschneiden und in welchen Bereichen Unterschiede identifiziert werden können. Die Literatur zum Requirements Engineering nennt hauptsächlich Phänomene, die in den Bereich der fehlenden

Information oder in den Bereich der Mehrdeutigkeiten fallen. Auch die Literatur zur technischen Dokumentation nennt Phänomene, die in diese Bereiche fallen. Jedoch liegt der Fokus in der technischen Dokumentation zusätzlich auf dem Bereich der sprachlichen Komplexität. Phänomene, welche in den Bereich der Komplexität fallen, werden in der betrachteten Literatur zum Requirements Engineering kaum genannt.

Die in der Requirements Engineering-Literatur genannten Phänomene überschneiden sich großteils mit den Phänomenen, welche in der Literatur zur technischen Dokumentation aufgeführt werden. Jedoch ist zu beachten, dass in den beiden Bereichen jeweils eine unterschiedliche Kommunikationssituation vorliegt und die Phänomene aus diesem Grund nicht komplett deckungsgleich sind. In technischer Dokumentation werden Texte von Experten für Laien geschrieben. Information soll von einem Laien, der sich nicht in seiner täglichen Arbeit mit einem bestimmten Thema beschäftigt, schnell verstanden werden. Daher liegt der Fokus unter anderem darauf, möglichst kurze Sätze zu schreiben und Strukturen zu vermeiden, welche die Komplexität erhöhen. Im Requirements Engineering hingegen schreiben Experten für Experten.

Einen dritten Beitrag liefert die Arbeit insofern, als dass untersucht wurde, ob sich die als relevant klassifizierten Phänomene mittels computerlinguistischer Korpusanalysemethoden identifizieren lassen und somit automatisierte Prüfmethode bereitgestellt werden können. Die Auswahl der relevanten Phänomene wurde anhand von vier Kriterien durchgeführt: Kritikalität, Häufigkeit in den als Quelle genutzten Korpora, generelle Erkennbarkeit eines Phänomens und zuverlässige Erkennbarkeit der kritischen Instanzen eines Phänomens. Die Phänomene, deren Kritikalitätsstufe im Qualitätsmodell als mittel oder hoch eingestuft ist, die häufig in den erstellten Korpora vorkommen und generell durch Regeln erkennbar sind, sind für die weiteren Untersuchungen relevant. Die Phänomene, bei welchen ohne weiterführende Analysen bereits die Aussage getroffen werden konnte, dass keine Regeln entworfen werden können, welche die kritischen Instanzen zuverlässig von den unkritischen Instanzen unterscheiden können, wurden von den Analysen ausgeschlossen. Anhand der Auswahlkriterien wurden sieben Phänomene als relevant identifiziert: *Fehlen einer quantifizierenden Angabe*, *Weak-Word*, *Unpersönliche Konstruktion*, *Syntaktisches Passiv ohne Agens*, *Passiversatz „sein“ + „zu“ + Infinitiv ohne Agens*, *Modalverb „sollte(n)“* und *Pronominaler Bezug*. Die Phänomene *Fehlen einer quantifizierenden Angabe* und *Weak-Word* wurden zu einer Analyse zusammengefasst. Die beiden Phänomene, die das Passiv beschreiben, wurden ebenfalls zusammengefasst.

Für die Phänomene *Weak-Word* und *Modalverb „sollte(n)“* konnten Regeln abgeleitet werden, welche die kritischen Instanzen zuverlässig identifizieren, d.h. bei den Analysen werden nur wenige Fehlwarnungen produziert. Die Evaluierung der Regeln zeigt, dass hohe

Precision- und Recall-Werte erzielt werden können, was diesen Aspekt nochmals unterstreicht. Bei den Phänomenen *Weak-Word* und *Modalverb „sollte(n)“* spielt der Einsatz von Welt- und Expertenwissen eine eher untergeordnete Rolle. Möglicher Interpretationsspielraum kann nur sehr selten durch Einsatz dieses Wissens eingeschränkt werden. Bei den Phänomenen *Passiv*, *Unpersönliche Konstruktion* und *Pronominaler Bezug* stellt sich dies anders dar.

Morphosyntaktisch mehrdeutige pronominale Bezüge können mit der beschriebenen computerlinguistischen Technologie zuverlässig automatisiert identifiziert werden. Jedoch spielt bei diesem Phänomen der Einsatz von Welt- und Expertenwissen eine erhebliche Rolle. Ein morphosyntaktisch mehrdeutiger Bezug ist zwar ein Indiz dafür, dass auch ein semantisch mehrdeutiger Bezug vorliegen könnte, jedoch haben die Analysen gezeigt, dass ein morphosyntaktisch mehrdeutiger Bezug nur in sehr seltenen Fällen auch zu einem semantisch mehrdeutigen Bezug führt. Durch Einsatz von Welt- und Expertenwissen kann der morphosyntaktisch mehrdeutige Bezug in den meisten Fällen disambiguiert werden. Hier stößt die formbasierte Computerlinguistik an ihre Grenzen: Wissen, das auf Welt- oder Expertenwissen zurückzuführen ist, kann durch einen automatisierten Ansatz auf der Basis morphosyntaktischer und syntaktischer Merkmale nicht bereitgestellt werden. Bei den Passivanalysen stößt man auf dieselbe Problematik. Die generelle Erkennung von Passivsätzen ohne Agens ist mit den beschriebenen Mitteln möglich. Passivsätze treten verhältnismäßig oft in den Korpora auf. Die Analysen haben gezeigt, dass durch Welt-, Experten- und Kontextwissen in den meisten Fällen das Agens ermittelt werden kann. Prüfungen, die zuverlässig zwischen kritischen und unkritischen Instanzen unterscheiden, können mit den Mitteln der formbasierten Korpuslinguistik für diese Phänomene nicht entwickelt werden. Es können lediglich Hinweise geliefert werden, an welchen Stellen Information fehlen könnte oder ein potentiell mehrdeutiger Bezug vorliegt. Die generelle Erkennbarkeit kann durch die Regeln jedoch gewährleistet werden.

Die entwickelten Analysen heben sich von gängigen Analysemethoden dadurch ab, dass nicht nur reine Wortlistenabgleiche durchgeführt werden, sondern auch der Satzkontext, in welchem die Phänomene auftreten, in die Analyse miteinbezogen wird. Dies stellt insofern eine erhebliche Verbesserung der Analysequalität dar, als dass erheblich weniger Fehlwarnungen produziert werden als bei bloßen Wortlistenabgleichen.

Die *Weak-Word-Analyse* und die *Analyse auf problematische Verwendungen des Modalverbs sollte(n)* wurden in ein Daimler-internes Werkzeug integriert. Das Werkzeug exportiert zunächst die zu analysierenden Anforderungen. Anschließend werden die Texte tokenisiert, lemmatisiert, getaggt und geparst. Die Anforderungen können dann durch die *The IMS Open Corpus Workbench* prozessiert werden. Die entworfenen CQP-Regeln

werden auf die annotierten Sätze angewandt und die Ergebnisse der Analysen an den Benutzer zurückgeliefert. Der Benutzer bekommt nicht nur die Stellen zurückgeliefert, an welchen eine potentielle Schwachstelle vorliegt, es wird auch eine an das Problem angepasste Fehlermeldung angezeigt, damit der Benutzer schnell verstehen kann, wieso eine Anforderung potentiell problematisch ist.

Die Validität der empirischen Ergebnisse wird insofern sichergestellt, als dass Experimente durchgeführt wurden, die unterstreichen, dass die subjektive Einschätzung der Autorin die Validität kaum beeinflusst. Zunächst wurde eine Untersuchung durchgeführt, in welcher Fleiss' Kappa berechnet wurde. Die Autorin und zwei weitere Personen, die in ihrer täglichen Arbeit mit Anforderungen arbeiten, haben 150 Passivsätze danach bewertet, ob die explizite Nichtnennung des Agens zu Verständnisproblemen führt. Die Einschätzung der Autorin deckt sich größtenteils mit der Einschätzung der beiden anderen Bewerter. In einem zweiten Experiment wurde die Einschätzung der Autorin mit der Einschätzung von Daimler-internen Testern und Testern aus externen Testhäusern verglichen. Auch hier korrespondiert die Einschätzung der Autorin größtenteils mit der Einschätzung der Tester. Die beiden Experimente zeigen, dass die subjektive Einschätzung der Autorin kaum einen Einfluss auf die Validität der Ergebnisse hat.

Insgesamt hat sich gezeigt, dass sich sprachliche Phänomene, die einen Einfluss auf die Testbarkeit einer Anforderung haben, durch computerlinguistische Analysemethoden automatisiert identifizieren lassen. Für bestimmte Phänomene können Regeln entworfen werden, welche zuverlässig zwischen den kritischen und den unkritischen Instanzen unterscheiden. Diese Analysen weisen eine geringe Anzahl an Fehlwarnungen auf, wodurch sie sich von gängigen Analysemethoden stark abheben. Jedoch stößt die formbasierte Computerlinguistik an ihre Grenzen, wenn der Einsatz von Welt- und Expertenwissen notwendig ist, um zwischen kritischen und unkritischen Instanzen zu unterscheiden. Für diese Phänomene können zwar Regeln entworfen werden, welche die morphosyntaktisch mehrdeutigen Instanzen identifizieren, jedoch können die semantisch mehrdeutigen Instanzen nicht zuverlässig identifiziert werden. Ebenso stößt die formbasierte Computerlinguistik an ihre Grenzen, wenn es darum geht, inhaltliche Probleme zu identifizieren. Widersprüche können mit der bereitgestellten Technologie nicht automatisiert identifiziert werden. Die inhaltliche Korrektheit einer Anforderung und die effektive und effiziente Durchführbarkeit der Umsetzung einer Anforderung kann ebenfalls nicht überprüft werden.

11.2 Ausblick

Die präsentierten Phänomenanalysen wurden auf deutschen Lastenheften durchgeführt. Vergleichbare Analysen wurden zu einem gewissen Anteil auch auf englischsprachige Las-

tenhefte übertragen. Für weitergehende Untersuchungen von englischsprachigen Lastenheften wäre es sinnvoll, zunächst wiederum ein Qualitätsmodell für testbare Anforderungen aufzustellen, um die Relevanz einzelner Phänomene herauszustellen. In den durchgeführten Phänomenanalysen wurde gezeigt, dass einige Phänomene, die für deutschsprachige Lastenhefte identifiziert wurden, analog auch im Englischen vorliegen. Jedoch ist zu beachten, dass in den englischsprachigen Lastenheften vermutlich auch andere Phänomene als im Deutschen von Relevanz für die Behandlung sein werden.

Da Deutsch und Englisch nicht die einzigen Sprachen sind, welche für die Spezifikation von Anforderungen genutzt werden können, wäre es weiterhin auch interessant zu untersuchen, welche Phänomene in weiteren Sprachen relevant sind im Hinblick auf die Testbarkeit einer Anforderung. Die Deckungsgleichheit und die Unterschiede zwischen den verschiedenen Sprachen könnten somit herausgearbeitet werden.

Das präsentierte Werkzeug in Abschnitt 9.1, in welches die entwickelten Prüfungen integriert wurden, analysiert bereits geschriebene Texte. Eine simultane Prüfung während des Schreibens könnte den Autor noch besser unterstützen, weil mögliche Schwachstellen in den Anforderungen sofort angezeigt werden würden. Der Benutzer müsste somit nicht im Nachgang alle Schwachstellen auf einmal bearbeiten. Eine solche Erweiterung ist in dem präsentierten Daimler-internen Werkzeug aus technischen Gründen nicht möglich. Die Entwicklung einer solchen Funktion wäre aber in jedem Falle sinnvoll.

Einen weiteren interessanten Untersuchungsgegenstand stellt die Übertragbarkeit der Analysen auf weitere Textsorten dar. Anforderungen werden zum einen für die Entwicklung genutzt, da sie beschreiben wie sich eine Komponente oder ein System in bestimmten Situationen verhalten soll, zum anderen dienen sie aber auch für die Ableitung von Testfällen. Die Dokumentation von Testfällen kann ebenfalls in natürlicher Sprache erfolgen. Im Rahmen dieser Arbeit wurde daher eine kleine Studie durchgeführt, in welcher betrachtet wurde, ob sich die entwickelten Methoden auf Testspezifikationen übertragen lassen. Liegen die Testfälle in den Testspezifikationen in ganzen Sätzen oder in Halbsätzen vor, können die Analysen angewandt werden. Jedoch hat sich herausgestellt, dass die in Anforderungen beobachteten linguistischen Phänomene in der Testphase der Softwareentwicklung kaum noch von Relevanz sind. Weak-Words und ihre Kontexte können zwar durch die entwickelten Prüfungen automatisiert identifiziert werden, jedoch sind in den Testspezifikationen kaum Weak-Words ohne quantifizierende Angaben zu finden. Unklarheiten aufgrund fehlender Information wurden vermutlich mit dem zuständigen Fachbereich geklärt. Das Vorhandensein eines unterspezifizierten Weak-Words hat zwar in diesem Fall nicht dazu geführt, dass der falsche Testfall abgeleitet wurde, jedoch stellt es einen erhöhten Zeitaufwand für den Tester dar, wenn fehlende Information erst erfragt

werden muss. Im schlimmeren Fall wird aufgrund subjektiver Einschätzung ein möglicher Testfall abgeleitet, ohne davor mit dem verantwortlichen Fachbereich zu sprechen. In diesem Falle ist die Wahrscheinlichkeit hoch, dass ein falscher Testfall abgeleitet wird.

Die Analyse bezüglich des Modalverbs *sollte(n)* ist in der Testphase nicht mehr relevant. Eine Entscheidung darüber, ob die Anforderung realisiert werden muss oder nicht, wurde in dieser Phase bereits getroffen. Entweder wurde ein entsprechender Testfall entwickelt oder es wurde die Entscheidung gefällt, dass die Anforderung nicht verpflichtend zu realisieren ist. In den Testfällen ist das Modalverb *sollte(n)* daher ebenfalls kaum zu finden.

Weiterhin könnte untersucht werden, inwiefern sich die entwickelten Analysen noch auf andere Bereiche außerhalb der Softwareentwicklung übertragen lassen, z.B. auf Betriebs- und Wartungsliteratur. Zu erwarten ist, dass sich auch hier die entwickelten Analysen anwenden lassen, jedoch muss die Relevanz der Phänomene für die jeweilige Textsorte zunächst herausgearbeitet werden.

Für die manuelle Erstellung der in dieser Arbeit präsentierten Regeln mussten intensive Analysen an Lastenhefttexten durchgeführt werden. Die Klassifikation der Kontexte, in welchen bestimmte Wörter vorkommen, und die manuelle Ableitung von Regeln, stellt einen großen Zeitaufwand dar. Für zukünftige Arbeiten wäre es denkbar, für die Klassifikation von Anforderungen, maschinelles Lernen einzusetzen. Liegen klassifizierte Daten vor, d.h. Anforderungen, die als kritisch und unkritisch markiert sind, könnte anhand dieser Daten ein Klassifizierungswerkzeug trainiert werden, welches dann die Bewertung automatisiert an unbekanntem Textdaten durchführen kann. Dies könnte den Aufwand der Regelentwicklung erheblich reduzieren.

Anhang

A Stuttgart-Tübingen-Tagset

A.1 Part-of-Speech-Tagset

ADJA	adjective, attributive	[das] große [Haus]
ADJD	adjective, adverbial or predicative	[er fährt] schnell, [er ist] schnell
ADV	adverb	schon, bald, doch
APPR	preposition; circumposition left	in [der Stadt], ohne [mich]
APPRART	preposition with article	im [Haus], zur [Sache]
APPO	postposition	[ihm] zufolge, [der Sache] wegen
APZR	circumposition right	[von jetzt] an
ART	definite or indefinite article	der, die, das, ein, eine, ...
CARD	cardinal number	zwei [Männer], [im Jahre] 1994
FM	foreign language material	[Er hat das mit " A big fish]" übersetzt]
ITJ	interjection	mhm, ach, tja
KOUI	subordinate conjunction with <i>zu</i> and infinitive	um [zu leben], anstatt [zu fragen]
KOUS	subordinate conjunction with sentence	weil, daß, damit, wenn, ob
KON	coordinate conjunction	und, oder, aber
KOKOM	comparative conjunction	als, wie
NN	common noun	Tisch, Herr, [das] Reisen
NE	proper noun	Hans, Hamburg, HSV
PDS	substituting demonstrative pronoun	dieser, jener
PDAT	attributive demonstrative pronoun	jener [Mensch]
PIS	substituting indefinite pronoun	keiner, viele, man, niemand
PIAT	attributive indefinite pronoun without determiner	kein [Mensch], irgendein [Glas]
PIDAT	attributive indefinite pronoun with determiner	[ein] wenig [Wasser], [die] beiden [Brüder]
PPER	non-reflexive personal pronoun	ich, er, ihm, mich, dir
PPOSS	substituting possessive pronoun	meins, deiner
PPOSAT	attributive possessive pronoun	mein [Buch], deine [Mutter]
PRELS	substituting relative pronoun	[der Hund ,] der
PRELAT	attributive relative pronoun	[der Mann ,] dessen [Hund]
PRF	reflexive personal pronoun	sich, dich, mir

PWS	substituting interrogative pronoun	wer, was
PWAT	attributive interrogative pronoun	welche [Farbe], wessen [Hut]
PWAV	adverbial interrogative or relative pronoun	warum, wo, wann, worüber, wobei
PAV	pronominal adverb	dafür, dabei, deswegen, trotzdem
PTKZU	<i>zu</i> before infinitive	zu [gehen]
PTKNEG	negative particle	nicht
PTKVZ	separable verbal particle	[er kommt] an, [er fährt] rad
PTKANT	answer particle	ja, nein, danke, bitte
PTKA	particle with adjective or adverb	am [schönsten], zu [schnell]
SGML	SGML markup	<turnid=n022k_TS2004>
SPELL	letter sequence	S-C-H-W-E-I-K-L
TRUNC	word remnant	An- [und Abreise]
VVFIN	finite verb, full	[du] gehst, [wir] kommen [an]
VVIMP	imperative, full	komm [!]
VVINFIN	infinitive, full	gehen, ankommen
VVIZU	Infinitive with <i>zu</i> , full	anzukommen, loszulassen
VVPP	perfect participle, full	gegangen, angekommen
VAFIN	finite verb, auxiliary	[du] bist, [wir] werden
VAIMP	imperative, auxiliary	sei [ruhig !]
VAINFIN	infinitive, auxiliary	werden, sein
VAPP	perfect participle, auxiliary	gewesen
VMFIN	finite verb, modal	dürfen
VMINFIN	infinitive, modal	wollen
VMPP	perfect participle, modal	gekonnt, [er hat gehen] können
XY	non-word containing non-letter	3:7, H2O, D2XW3
\$,	comma	,
\$.	sentence-final punctuation mark	. ? ! ; :
\$(other sentence-internal punctuation mark	- [,] ()

A.2 Parsing-Tagset

AC	adpositional case marker
ADC	adjective component
AG	genitive attribute
AMS	measure argument of adjective
APP	apposition
AVC	adverbial phrase component
CC	comparative complement
CD	coordinating conjunction
CJ	conjunct
CM	comparative conjunction
CP	complementizer
CVC	collocational verb construction (Funktionsverbgefüge)
DA	dative
DH	discourse-level head
DM	discourse marker
EP	expletive es
HD	head
JU	junctor
MNR	postnominal modifier
MO	modifier
NG	negation
NK	noun kernel element
NMC	numerical component
OA	accusative object
OA	second accusative object
OC	clausal object
OG	genitive object
OP	prepositional object
PAR	parenthetical element
PD	predicate
PG	phrasal genitive
PH	placeholder
PM	morphological particle
PNC	proper noun component
RC	relative clause
RE	repeated element

RS	reported speech
SB	subject
SBP	passivised subject (PP)
SP	subject or predicate
SVP	separable verb prefix
UC	unit component
VO	vocative

A.3 Morphologie-Tagset

Attribut	mögliche Werte	verwendet bei
gender	masc, fem, neut	ADJA, ART, APPRART, NE, NN, PDS, PDAT, PIAT, PIS (teilweise), PPER, PPOSAT, PPOSS, PRELS, PRELAT, PWAT, PWS
case	nom, gen, dat, acc	ADJA, ART, APPRART, NE, NN, PDAT, PDS, PIAT, PIS, PPER, PPOSAT, PPOSS, PRELS, PRELAT, PRF, PWAT, PWS
number	sg, pl	ADJA, ART, APPRART, NE, NN, PDAT, PDS, PIAT, PIS, PPER, PPOSAT, PPOSS, PRELS, PRELAT, PRF, PWAT, PWS, V.FIN, V.IMP
degree	pos, comp, sup	ADJA, ADJD
person	1, 2, 3	VVFIN, VAFIN, VMFIN, PPER, PRF
tense	pres, past	VVFIN, VAFIN, VMFIN
mood	ind, subj	VVFIN, VAFIN, VMFIN

B Part-of-Speech-Tagset der Penn Treebank

CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP\$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP\$	Possessive wh-pronoun
WRB	Wh-adverb

C CQP-Regeln für die Analyse von Weak-Words

Regelname	CQP-Regel
R1_lang	[pos='CARD' word='ein.*'] [pos='NN NE' word=(ms) (sek\.) (mm) (m) (cm) (km) (s) (sec\.)'] [lemma='lang'];
R2_lang	[lemma='lang' & morph=.*COMP.*] 'als'? [pos='CARD'] [pos='NN NE' word=(ms) (sek\.) (mm) (m) (cm) (km) (s) (sec\.)']?;
R3_lang	[lemma='lang' & morph=.*COMP.*] [pos='ADJ.+']? 'als' [word='für']? [pos='(ART PDAT)' & word='d.+'];
R4_lang	'[W w]ie' [lemma='lange?'];
R5_lang	'[A-Za-z]+' '_' '\S+)'([A-Z][a-z]*[A-Z]+)' [lemma='lange?'];
R6_lang	[lemma='lang' & morph=.*COMP.*] [pos='V.+']+ 'als';
R7_lang	[lemma='lang' & morph=.*COMP.*] 'als' '[A-Za-z]+' '_' '\S+)'([A-Z][a-z]*[A-Z]+);
R8_lang	[lemma='lange?'] [pos='NN NE'] 'von' '(weniger) (mehr)'? 'als'? [pos='CARD'];
R9_lang	[lemma='lang' & morph=.*COMP.*] [pos='(ART PDAT)' word='d.+'] [pos='NN NE'];
R10_lang	[lemma='lange?'] [pos='NN NE'] 'nach' '(ISO.*) (DIN.*)';
R11_lang	'\=' [lemma='lange?'] [pos='NN NE'];
R12_lang	[lemma='lange?'] [pos='NN NE'] ':' '[A-Za-z]+' '_' '\S+)'([A-Z][a-z]*[A-Z]+);
R13_lang	[lemma='lange?'] [pos='NN NE'] '(ist) (beträgt)' [pos='CARD'];
R14_lang	[pos='(ART PDAT)' & word='[D d].+'] [lemma='lang' & morph=.*COMP.*] [pos='(ART PDAT)' & word='d.+'];
R15_lang	'([G g]enauso) ([E e]benso) ([S s]o)' [lemma='lange?'];
R16_lang	[pos='ART PDAT APPR']? [lemma='lang' & morph=.*SUP.*];
R17_lang	[pos='CARD'] [word=(ms) (sek\.) (mm) (m) (cm) (km) (s) (sec\.)' pos='(NN) (NE)']? 'und oder' [lemma='lang' & morph=.*COMP.*];
R18_lang	[pos='ART PDAT' & word='[D d].+'] [lemma='lang' & morph=.*COMP.*];
R19_lang	[pos='CARD'] '{1,3}' '/' '{1,3}' [lemma='lange?'];
R20_lang	[lemma='lange?' & morph=.*COMP.*] [pos='NN NE'] 'als';
R21_lang	[pos='APPR APPRART'] [lemma='lang' & morph=.*COMP.*] [pos='NN NE'];
R22_lang	[pos='APPR'] [pos='ART PDAT' & word='d.+'] [lemma='lang'] [pos='NN NE'];
R23_lang	[lemma='lang'] [pos='NN NE'] 'bedeutet';
R24_lang	'nächst.*' [lemma='lang' & morph=.*COMP.*] [pos='NN NE']?;
R25_lang	'[Z z]u' [lemma='lang'];
R26_lang	[pos='ART' & word='[E e].+'] [lemma='lange?' & morph=.*COMP.*] [pos='ADJA ADJD']*;
R27_lang	[lemma='lang'] [pos='NN NE'] 'von'? 'ca\.'?;
R28_lang	'ca\.'? [pos='CARD'] [word=(ms) (sek\.) (mm) (m) (cm) (km) (s) (sec\.)' pos='(NN) (NE)'] [lemma='lang'];
R29_lang	[lemma='lang' & morph=.*COMP.*] 'als'? 'ca\.'?;

Tabelle 47: CQP-Regeln für die Weak-Words *lang*, *kurz*, *schnell*, *breit*, *hoch* und *niedrig*

Regelname	CQP-Regel
R1_groß	[lemma='groß' & word='größer'] 'sein'? 'als'? [pos='NN NE']? [pos='(CARD) (NN) (NE)'];
R2_groß	[pos='ART PDAT' & word='[D d].+'] [lemma='groß'] [pos='ART PDAT' & word='[D d].+'];
R3_groß	'so' [lemma='groß'] 'zu'? [pos='V.+']+ ','? 'dass';
R4_groß	[lemma='groß'] [pos='NN NE'] 'von' [pos='CARD'];
R5_groß	[lemma='groß'] 'gleich' [pos='(CARD) (NN) (NE)'];
R6_groß	[pos='NN NE'] [lemma='groß'] [pos='ART PDAT' & word='[D d].+'];
R7_groß	[lemma='groß'] 'als'? '[A-Za-z]+_\S+ [A-Z][a-z]*[A-Z]+';
R8_groß	'so' [lemma='groß'] 'wie';
R9_groß	[pos='CARD'] [word='(ms) (sek\.) (mm) (m) (cm) (km) (s) (sec\.)' pos='(NN) (NE)'] [lemma='groß'];
R10_groß	[lemma='groß'] [pos='VAINF'] 'als';
R11_groß	[lemma='groß'] 'als' [pos='ART PDAT' & word='[D d].+'];
R12_groß	[pos='NN NE'] [lemma='groß' & morph='.*COMP.*'] [pos='NN NE'];
R13_groß	[word='.*\(.*)' [lemma='groß'] [word='.*\).*'];
R14_groß	[pos='ART PDAT APPR']? [lemma='groß' & morph='.*SUP.*'];
R15_groß	'([G g]enauo) ([E e]benso) ([S s]o)' [lemma='groß'];
R16_groß	'[W w]ie' [lemma='groß'];
R17_groß	[lemma='groß' & morph='.*COMP.*'] 'und oder' 'gleich';
R18_groß	[pos='CARD'] [word='(ms) (sek\.) (mm) (m) (cm) (km) (s) (sec\.)' pos='(NN) (NE)']? 'und oder' [lemma='groß' & morph='.*COMP.*'];
R19_groß	[pos='(ART PDAT)' & word='[D d].+'] [lemma='groß' & morph='.*COMP.*'] [pos='NN NE'];
R20_groß	[lemma='groß' & morph='.*COMP.*'] [pos='NN NE'] 'als';
R21_groß	[pos='APPR APPRART'] [lemma='groß' & morph='.*COMP.*'] [pos='NN NE'];
R22_groß	[pos='APPR'] [pos='ART PDAT' & word='d.+'] [lemma='groß'] [pos='NN NE'];
R23_groß	[lemma='groß'] [pos='NN NE'] 'bedeutet';
R24_groß	'nächst.*' [lemma='groß' & morph='.*COMP.*'] [pos='NN NE']?;
R25_groß	[lemma='groß'] 'als' [pos='NN NE'];
R26_groß	[pos='ART' & word='[E e].+'] [lemma='groß' & morph='.*COMP.*'];
R27_groß	'möglichst' [lemma='groß'];
R28_groß	'ausreichend' [lemma='groß'];
R29_groß	'relevant' [lemma='groß'];

Tabelle 48: CQP-Regeln für die Weak-Words *groß* und *klein*

Regelname	CQP-Regel
R1_maximal	[lemma='maximal'] [pos='NN NE'] 'von' [pos='CARD'];
R2_maximal	[lemma='maximal'] [pos='CARD'];
R3_maximal	[lemma='maximal'] [] & '=' [pos='CARD'];
R4_maximal	[lemma='maximal'] 'bis';
R5_maximal	[lemma='maximal'] [pos='NN NE'] [pos!='V.+']* 'beträgt ist sind' [pos='CARD'];
R6_maximal	[lemma='maximal'] 'mit'? [pos='ART'] [pos='NN NE'] 'von';
R7_maximal	[(pos='ART PDAT' & word='[D d].+') pos='APPR'] [pos='ADJ.+']* [lemma='maximal'] [pos='ADJ.+']* [pos='NN NE'];
R8_maximal	'\=' [lemma='maximal'];
R9_maximal	[lemma='maximal'] [pos='NN NE'] '\:' [pos='CARD'];
R10_maximal	[lemma='maximal'] [pos='NN NE'] '.*' [pos='CARD'];
R11_maximal	[lemma='maximal'] [pos='NN NE'] 'gemäß gemäss gemaess laut';
R12_maximal	[lemma='maximal'] [pos='NN NE'] [pos!='V.+']* '\:' '\+ \-' [pos='CARD'];
R13_maximal	[lemma='maximal'] [pos='NN NE'] [pos='V.+'] [pos='CARD'] [] [pos='V.+'];
R14_maximal	[lemma='maximal'] 'zu' [pos='CARD'];
R15_maximal	<s> [lemma='maximal'] [pos='NN NE'];
R16_maximal	[lemma='maximal'] '(jedoch) (dennoch) (aber) (allerdings) (um)' [pos='CARD'];
R17_maximal	[lemma='maximal'] ':' [pos='CARD'];
R18_maximal	[lemma='maximal'] '+mal';
R19_maximal	[lemma='maximal'] [pos='NN NE'] 'bedeutet';
R20_maximal	[lemma='maximal'] '\+ \-';
R21_maximal	[pos='PPOSAT'] [lemma='maximal'] [pos='ADJA ADJD']? [pos='NN NE'];

Tabelle 49: CQP-Regeln für die Weak-Words *maximal* und *minimal*

Regelname	CQP-Regel
R1_eigentlich	[(pos='ART PDAT' & word='[D d].+') pos='KOUS PIAT'] [lemma='eigentlich'] [pos='NN NE ADJA ADJD KON ADV APPRART' word='.*.*']* [pos='NN NE'];
R2_eigentlich	'.*\(..*' [lemma='eigentlich'] [pos='NN NE ADJA ADJD' word='.*.*']{0,3} [pos='NN NE'];
R3_eigentlich	[pos='PPOSAT'] [lemma='eigentlich'] [pos='NN NE ADJA ADJD KON ADV' word='.*.*']* [pos='NN NE'];
R4_eigentlich	[pos='V.+'] [lemma='eigentlich'];
R5_eigentlich	[lemma='eigentlich'] [pos='V.+'];

Tabelle 50: CQP-Regeln für die Weak-Words *eigentlich* und *tatsächlich*

Regelname	CQP-Regel
R1_mal	[pos='CARD' pos='ADJA' word='ein.*'] [lemma='mal'];
R2_mal	'erst.+ ' [lemma='mal'];
R3_mal	'[A-Za-z]+' '_' '(\S+) ([A-Z][a-z]*[A-Z]+)' [lemma='mal'];
R4_mal	'[0-9]+ten' [lemma='mal'];
R5_mal	[pos='CARD'] '-' [lemma='mal'];
R6_mal	'noch' [lemma='mal'];
R7_mal	'auch' [lemma='mal'];
R8_mal	[pos='V.+'] [lemma='mal'];

Tabelle 51: CQP-Regeln für das Weak-Word *mal*

D CQP-Regeln für die Analyse von Passivsätzen ohne Agens

Regelname	Beispiel	CQP-Regel
R1_VorgPassiv	<i>Standardkonzepte werden nicht in Form von Dokumenten ausgegeben.</i>	[pos="N(N E) PPER"] []* [lemma="werden"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] :: a.dep!=b.wortnr within s;
R2_VorgPassiv	<i>Die Prüfspannung wird auf Stufe 1 angelegt.</i>	[pos="N(N E) PPER"] []*[lemma="werden"] [word!="vo(n m) durch"]* [pos="VVPP"] within s;
R3_VorgPassiv	<i>..., dass der maximale Heizfeldstrom von 21A nicht überschritten wird.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="werden"] [pos!="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R4_VorgPassiv	<i>..., so dass Änderungen am Prozess dokumentiert werden.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="werden"] [pos!="VM(FIN INF)"] within s;
R5_VorgPassiv	<i>Clipse müssen im Abstand von 50 mm positioniert werden.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="werden"] :: a.dep!=b.wortnr within s;
R6_VorgPassiv	<i>Die Durchflussmenge muss erfüllt werden.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="werden"] within s;
R7_VorgPassiv	<i>..., wenn das Zeitsignal in einem Abstand von 3 Sekunden nicht abgespeichert werden kann.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="werden"] [pos="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R8_VorgPassiv	<i>..., dass der Fußgängerschutz erfüllt werden kann.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="werden"] [pos="VM(FIN INF)"] within s;

Tabelle 52: Regeln für die Identifikation von Vorgangspassiv ohne Agens

Regelname	Beispiel	CQP-Regel
R1_ZustPassiv	<i>Der Einsatz ist X unabhängig von Y geplant.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] a.dep!=b.wortnr within s;
R2_ZustPassiv	<i>Die Oberfläche des Zierteils ist verchromt.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] [word!="vo(n m) durch"]* [pos="VVPP"] within s;
R3_ZustPassiv	<i>..., dass im Akkubetrieb eine Alarmauslösung für die geforderte Anzahl von Alarmzyklen gewährleistet ist.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="sein"] [pos!="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R4_ZustPassiv	<i>..., dass Fehlmontagen ausgeschlossen sind.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="sein"] [pos!="VM(FIN INF)"] within s;
R5_ZustPassiv	<i>Das Aufnahmegewinde muss minimal mit einer Länge von 5 mm in den XYZ eingesteckt sein.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="sein"] :: a.dep!=b.wortnr within s;
R6_ZustPassiv	<i>Die Absicherung muss sichergestellt sein.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="sein"] within s;
R7_ZustPassiv	<i>..., dass im Akkubetrieb eine Alarmauslösung für die geforderte Anzahl von Alarmzyklen gewährleistet sein muss.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"]* []* b:[pos="VVPP"] [lemma="sein"] [pos="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R8_ZustPassiv	<i>..., dass der Fußgängerschutz erfüllt sein kann.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="sein"] [pos="VM(FIN INF)"] within s;

Tabelle 53: Regeln für die Identifikation von Zustandpassiv ohne Agens

Regelname	Beispiel	CQP-Regel
R1_ist_zu	<i>Es ist im Windkanal bei Frontalanströmung ein Wert von X zu erreichen.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] []* a:[word="vo(n m) durch"] []* [pos="PTKZU"] b:[pos="VVINF"] :: a.dep!=b.wortnr within s;
R2_ist_zu	<i>Der Öffnungswinkel ist zu ermitteln.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] [word!="vo(n m) durch"]* [pos="PTKZU"] [pos="VVINF"] within s
R3_ist_zu	<i>..., dass eine Lichtstärke von mindestens X Lux zu messen ist.</i>	[pos="KOUS"] []*[pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* [pos="PTKZU"] b:[pos="VVINF"] [lemma="sein"] :: a.dep!=b.wortnr within s;
R4_ist_zu	<i>..., dass Rastverbindungen einfach zu betätigen sind.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="PTKZU"] [pos="VVINF"] [lemma="sein"] within s;
R5_ist_zu	<i>Zur Seitenwand ist ein Abstand von 2 mm einzuhalten.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] []* a:[word="vo(n m) durch"] []* b:[pos="VVIZU"] :: a.dep!=b.wortnr within s;
R6_ist_zu	<i>Auf den Teilen sind Abdeckkappen vorzusehen.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] [word!="vo(n m) durch"]* [pos="VVIZU"] within s;
R7_ist_zu	<i>..., dass ein Mindestabstand von 10 mm zwischen A und B einzuhalten ist.</i>	[pos="KOUS"] []*[pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVIZU"] [lemma="sein"] :: a.dep!=b.wortnr within s;
R8_ist_zu	<i>..., ob die Reparatur teile- oder modulbezogen festzulegen ist.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVIZU"] [lemma="sein"] within s;

Tabelle 54: Regeln für die Identifikation von Passiversatzkonstruktionen in der Form *ist ... zu*

Regelname	Beispiel	CQP-Regel
R1_bekommen	<i>Der Sensor bekommt die vom Fahrer ausgewählte Stufe mitgeteilt.</i>	[pos="N(N E) PPER"] []* [lemma="bekommen"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] :: a.dep!=b.wortnr within s;
R2_bekommen	<i>Der Sensor bekommt die ausgewählte Stufe mitgeteilt.</i>	[pos="N(N E) PPER"] []* [lemma="bekommen"] [word!="vo(n m) durch"]* [pos="VVPP"] within s;
R3_bekommen	<i>..., dass der Sensor die vom Fahrer ausgewählte Stufe mitgeteilt bekommt.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="bekommen"] [pos!="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R4_bekommen	<i>..., dass der Sensor die ausgewählte Stufe mitgeteilt bekommt.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="bekommen"] [pos!="VM(FIN INF)"] within s;
R5_bekommen	<i>Der Sensor muss die vom Fahrer ausgewählte Stufe mitgeteilt bekommen.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="bekommen"] :: a.dep!=b.wortnr within s;
R6_bekommen	<i>Der Sensor muss die ausgewählte Stufe mitgeteilt bekommen.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="bekommen"] within s;
R7_bekommen	<i>..., dass der Sensor die vom Fahrer ausgewählte Stufe mitgeteilt bekommen muss.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVPP"] [lemma="bekommen"] [pos="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R8_bekommen	<i>..., dass der Sensor die ausgewählte Stufe mitgeteilt bekommen muss.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVPP"] [lemma="bekommen"] [pos="VM(FIN INF)"] within s;

Tabelle 55: Regeln für die Identifikation von Passiversatzkonstruktionen in der Form *bekommen*

Regelname	Beispiel	CQP-Regel
R1_lassen	<i>Die Türen lassen sich in der Vor- und in der Hauptraste in Zeitabständen von 3 Sekunden verriegeln.</i>	[pos="N(N E) PPER"] []* [lemma="lassen"] []* a:[word="vo(n m) durch"] []* b:[pos="VVINF"] :: a.dep!=b.wortnr within s;
R2_lassen	<i>Die Türen lassen sich in der Vor- und in der Hauptraste verriegeln.</i>	[pos="N(N E) PPER"] []*[lemma="lassen"] [word!="vo(n m) durch"]* [pos="VVINF"] within s;
R3_lassen	<i>..., dass sich die Türen in der Vor- und in der Hauptraste in Zeitabständen von 3 Sekunden verriegeln lassen.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVINF"] [lemma="lassen"] [pos!="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R4_lassen	<i>..., dass sich die Türen in der Vor- und in der Hauptraste verriegeln lassen.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVINF"] [lemma="lassen"] [pos!="VM(FIN INF)"] within s;
R5_lassen	<i>Die Türen müssen sich in der Vor- und Hauptraste in Zeitabständen von 3 Sekunden verriegeln lassen.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] []* a:[word="vo(n m) durch"] []* b:[pos="VVINF"] [lemma="lassen"] :: a.dep!=b.wortnr within s;
R6_lassen	<i>Die Türen müssen sich in der Vor- und Hauptraste verriegeln lassen.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] [word!="vo(n m) durch"]* [pos="VVINF"] [lemma="lassen"] within s;
R7_lassen	<i>..., dass sich die Türen in der Vor- und in der Hauptraste in Zeitabständen von 3 Sekunden verriegeln lassen müssen.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[pos="VVINF"] [lemma="lassen"] [pos="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R8_lassen	<i>..., dass sich die Türen in der Vor- und in der Hauptraste verriegeln lassen müssen.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [pos="VVINF"] [lemma="lassen"] [pos="VM(FIN INF)"] within s;

Tabelle 56: Regeln für die Identifikation von Passiversatzkonstruktionen in der Form *sich lassen*

Regelname	Beispiel	CQP-Regel
R1_bar	<i>Das hintere Rollo ist über Bedienelemente unabhängig vom vorderem Rollo bedienbar.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] []* a:[word="vo(n m) durch"] []* b:[word="."+bar" & pos="ADJD"] :: a.dep!=b.wortnr within s;
R2_bar	<i>Die Komponente ist flashbar.</i>	[pos="N(N E) PPER"] []* [lemma="sein"] [word!="vo(n m) durch"]* [word="."+bar" & pos="ADJD"] within s;
R3_bar	<i>..., dass ein Additivvolumen von 15g absaugbar ist .</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[word="."+bar" & pos="ADJD"] [lemma="sein"] [pos!="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R4_bar	<i>..., dass die Blende montierbar ist.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [word="."+bar" & pos="ADJD"] [lemma="sein"] [pos!="VM(FIN INF)"] within s;
R5_bar	<i>Es muss eine Mindestleistung von 18Wp erreichbar sein.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] []* a:[word="vo(n m) durch"] []* b:[word="."+bar" & pos="ADJD"] [lemma="sein"] :: a.dep!=b.wortnr within s;
R6_bar	<i>Die Wischerarme müssen abklappbar sein.</i>	[pos="N(N E) PPER"] []* [pos="VM(FIN INF)"] [word!="vo(n m) durch"]* [word="."+bar" & pos="ADJD"] [lemma="sein"] within s;
R7_bar	<i>..., falls nicht alle Signale innerhalb von 2 Sekunden erkennbar sein müssen.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] []* a:[word="vo(n m) durch"] []* b:[word="."+bar" & pos="ADJD"] [lemma="sein"] [pos="VM(FIN INF)"] :: a.dep!=b.wortnr within s;
R8_bar	<i>..., falls Signale erkennbar sein müssen.</i>	[pos="KOUS"] []* [pos="N(N E) PPER"] [word!="vo(n m) durch"]* [word="."+bar" & pos="ADJD"] [lemma="sein"] [pos="VM(FIN INF)"] within s;

Tabelle 57: Regeln für die Identifikation von Passiversatzkonstruktionen in der Form *-bar*

E CQP-Regeln für die Analyse pronominaler Bezüge

Regelname	CQP-Regel
PPER_si_masc	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PPER"& morph=".*sg.*masc.*"] within s;
PPER_si_fem	[pos="NN"& morph=".*sg.*fem.*"] []* [pos="NN"& morph=".*sg.*fem.*"] []* [pos="PPER"& morph=".*sg.*fem.*"] within s;
PPER_si_neut	[pos="NN"& morph=".*sg.*neut.*"] []* [pos="NN"& morph=".*sg.*neut.*"] []* [pos="PPER"& morph=".*sg.*neut.*"] within s;
PPER_si_pl	[pos="NN"& morph=".*pl.*"] []* [pos="NN"& morph=".*pl.*"] []* [pos="PPER"& morph=".*pl.*"] within s;
PPER_sü_masc	[pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* [pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* "\." [word!="\."]* [pos="PPER"& morph=".*sg.*masc.*"];
PPER_sü_fem	[pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* [pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* "\." [word!="\."]* [pos="PPER"& morph=".*sg.*fem.*"];
PPER_sü_neut	[pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* [pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* "\." [word!="\."]* [pos="PPER"& morph=".*sg.*neut.*"];
PPER_sü_pl	[pos="NN"& morph=".*pl.*"] [word!="\."]* [pos="NN"& morph=".*pl.*"] [word!="\."]* "\." [word!="\."]* [pos="PPER"& morph=".*pl.*"];

Tabelle 58: CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger Personalpronomina

Regelname	CQP-Regel
PDS_si_masc	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PDS"& morph=".*sg.*masc.*"] within s;
PDS_si_fem	[pos="NN"& morph=".*sg.*fem.*"] []* [pos="NN"& morph=".*sg.*fem.*"] []* [pos="PDS"& morph=".*sg.*fem.*"] within s;
PDS_si_neut	[pos="NN"& morph=".*sg.*neut.*"] []* [pos="NN"& morph=".*sg.*neut.*"] []* [pos="PDS"& morph=".*sg.*neut.*"] within s;
PDS_si_pl	[pos="NN"& morph=".*pl.*"] []* [pos="NN"& morph=".*pl.*"] []* [pos="PDS"& morph=".*pl.*"] within s;
PDS_sü_masc	[pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* [pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* "\." [word!="\."]* [pos="PDS"& morph=".*sg.*masc.*"];
PDS_sü_fem	[pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* [pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* "\." [word!="\."]* [pos="PDS"& morph=".*sg.*fem.*"];
PDS_sü_neut	[pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* [pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* "\." [word!="\."]* [pos="PDS"& morph=".*sg.*neut.*"];
PDS_sü_pl	[pos="NN"& morph=".*pl.*"] [word!="\."]* [pos="NN"& morph=".*pl.*"] [word!="\."]* "\." [word!="\."]* [pos="PDS"& morph=".*pl.*"];

Tabelle 59: CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger substituierender Demonstrativpronomina

Regelname	CQP-Regel
PRELS_si_masc	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PRELS"& morph=".*sg.*masc.*"] within s;
PRELS_si_fem	[pos="NN"& morph=".*sg.*fem.*"] []* [pos="NN"& morph=".*sg.*fem.*"] []* [pos="PRELS"& morph=".*sg.*fem.*"] within s;
PRELS_si_neut	[pos="NN"& morph=".*sg.*neut.*"] []* [pos="NN"& morph=".*sg.*neut.*"] []* [pos="PRELS"& morph=".*sg.*neut.*"] within s;
PRELS_si_pl	[pos="NN"& morph=".*pl.*"] []* [pos="NN"& morph=".*pl.*"] []* [pos="PRELS"& morph=".*pl.*"] within s;

Tabelle 60: CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger substituierender Relativpronomina

Regelname	CQP-Regel
PDAT_si_masc	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PDAT"& morph=".*sg.*masc.*"] within s;
PDAT_si_fem	[pos="NN"& morph=".*sg.*fem.*"] []* [pos="NN"& morph=".*sg.*fem.*"] []* [pos="PDAT"& morph=".*sg.*fem.*"] within s;
PDAT_si_neut	[pos="NN"& morph=".*sg.*neut.*"] []* [pos="NN"& morph=".*sg.*neut.*"] []* [pos="PDAT"& morph=".*sg.*neut.*"] within s;
PDAT_si_pl	[pos="NN"& morph=".*pl.*"] []* [pos="NN"& morph=".*pl.*"] []* [pos="PDAT"& morph=".*pl.*"] within s;
PDAT_sü_masc	[pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* [pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* "\." [word!="\."]* [pos="PDAT"& morph=".*sg.*masc.*"];
PDAT_sü_fem	[pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* [pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* "\." [word!="\."]* [pos="PDAT"& morph=".*sg.*fem.*"];
PDAT_sü_neut	[pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* [pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* "\." [word!="\."]* [pos="PDAT"& morph=".*sg.*neut.*"];
PDAT_sü_pl	[pos="NN"& morph=".*pl.*"] [word!="\."]* [pos="NN"& morph=".*pl.*"] [word!="\."]* "\." [word!="\."]* [pos="PDAT"& morph=".*pl.*"];

Tabelle 61: CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger attribue-
render Demonstrativpronomina

Regelname	CQP-Regel
PPOSAT_si_masc	[pos="NN"& morph=".*sg.*masc.*"] []* [pos="NN"& morph=".*sg.*masc.*"] []* [pos="PPOSAT"& morph=".*sg.*masc.*"] within s;
PPOSAT_si_fem	[pos="NN"& morph=".*sg.*fem.*"] []* [pos="NN"& morph=".*sg.*fem.*"] []* [pos="PPOSAT"& morph=".*sg.*fem.*"] within s;
PPOSAT_si_neut	[pos="NN"& morph=".*sg.*neut.*"] []* [pos="NN"& morph=".*sg.*neut.*"] []* [pos="PPOSAT"& morph=".*sg.*neut.*"] within s;
PPOSAT_si_pl	[pos="NN"& morph=".*pl.*"] []* [pos="NN"& morph=".*pl.*"] []* [pos="PPOSAT"& morph=".*pl.*"] within s;
PPOSAT_sü_masc	[pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* [pos="NN"& morph=".*sg.*masc.*"] [word!="\."]* "\." [word!="\."]* [pos="PPOSAT"& morph=".*sg.*masc.*"];
PPOSAT_sü_fem	[pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* [pos="NN"& morph=".*sg.*fem.*"] [word!="\."]* "\." [word!="\."]* [pos="PPOSAT"& morph=".*sg.*fem.*"];
PPOSAT_sü_neut	[pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* [pos="NN"& morph=".*sg.*neut.*"] [word!="\."]* "\." [word!="\."]* [pos="PPOSAT"& morph=".*sg.*neut.*"];
PPOSAT_sü_pl	[pos="NN"& morph=".*pl.*"] [word!="\."]* [pos="NN"& morph=".*pl.*"] [word!="\."]* "\." [word!="\."]* [pos="PPOSAT"& morph=".*pl.*"];

Tabelle 62: CQP-Regeln für die Identifikation morphosyntaktisch mehrdeutiger attribuierender Possessivpronomina

F Warnmeldungen

lang (Defect)

Die Angabe "lang" ist ungenau.

Bitte prüfen Sie ob:

*** eine Größen-/Mengen- oder Zeitangabe hinzugefügt werden kann

Bsp.: „langes Drücken von 3s; das lange Objekt B (größer 6cm)“

*** auf eine Quelle verwiesen werden kann, die "lang" genau beschreibt

*** Beispiele gegeben werden können

Bsp: „3s länger als Zeitraum A; so lange wie B“

lang (Effect)

Die Angabe "lang" ist möglicherweise ungenau.

Bitte prüfen Sie ob:

*** die Formulierung eindeutig auf eine Quelle verweist, die "lang" definiert

*** eine Größen-/Mengen- oder Zeitangabe hinzugefügt werden kann

Bsp.: „langes Drücken von 3s; das lange Objekt B (größer 6cm)“

*** Beispiele gegeben werden können

Bsp: „3s länger als Zeitraum A; so lange wie B“

groß (Defect)

Die Angabe "groß" ist ungenau.

Bitte prüfen Sie ob:

*** "groß" in Relation zu einer festen Einheit gesetzt werden kann

Bsp.: „Leitungen größer 2m; ein Wert größer als XY“

*** auf eine Quelle verwiesen werden kann, die "groß" genau beschreibt

Bsp.: „einen großen Parametertest gemäß Kapitel 3 des Anhangs X“

groß (Effect)

Die Angabe "groß" ist möglicherweise ungenau.

Bitte prüfen Sie ob:

*** die Formulierung eindeutig auf eine Quelle verweist, die "groß" definiert.

Bsp.: „einen großen Parametertest gemäß Kapitel 3 des Anhangs X“

*** "groß" in Relation zu einer festen Einheit gesetzt werden kann

Bsp.: „Leitungen größer 2m; ein Wert größer als XY“

maximal (Defect)

Die Angabe "maximal" ist ungenau.

Bitte prüfen Sie ob:

*** eine Größen-/Mengen- oder Zeitangabe hinzugefügt werden kann

Bsp.: „eine maximale Drehzahl von 7000rpm“

*** auf eine Quelle verwiesen werden kann, die "maximal" genau beschreibt

Bsp.: „maximale Spannungsdynamik gemäß Tabelle A; bei maximaler Last (ABC, XYZ)“

maximal (Effect)

Die Angabe "maximal" ist möglicherweise ungenau.

Bitte prüfen Sie ob:

*** die Formulierung eindeutig auf eine Quelle verweist, die "maximal" definiert.

Bsp.: „maximale Spannungsdynamik gemäß Tabelle A; bei maximaler Last (ABC, XYZ)“

*** eine Größen-/Mengen- oder Zeitangabe hinzugefügt werden kann

Bsp.: „eine maximale Drehzahl von 7.000rpm“

eigentlich (Defect)

Die Angabe "eigentlich" ist ungenau.

Bitte prüfen Sie ob:

*** Beispiele gegeben werden können

Bsp.: „bis zum Erreichen des eigentlichen Sollwerts AB_CD“

*** auf eine Quelle verwiesen werden kann, die "eigentlich" genau beschreibt

Bsp.: „der eigentliche Wert (Messung AB)“

eigentlich (Effect)

Die Angabe "eigentlich" ist möglicherweise ungenau.

Bitte prüfen Sie ob:

*** die Formulierung eindeutig auf eine Quelle verweist, die "eigentlich" definiert

Bsp.: „der eigentliche Wert (Messung AB)“

*** Beispiele gegeben werden können

Bsp.: „bis zum Erreichen des eigentlichen Sollwerts AB_CD“

mal (Defect)

Die Angabe "mal" ist ungenau.

Bitte prüfen Sie ob:

*** eine Größen-/Mengen- oder Zeitangabe hinzugefügt werden kann

Bsp.: „5 mal durchführen; nach dem 3ten mal“

sollte (Effect)

Sie haben das Verb “sollte” verwendet.

Das Verb “sollte” markiert optionale Anforderungen.

Bitte prüfen Sie ob:

*** die Anforderung bindend ist

Wenn ja, ersetzen Sie “sollte” durch “muss” bzw. “müssen”.

Für die Verneinung benutzen Sie “darf nicht” bzw. “dürfen nicht”.

Literaturverzeichnis

- [AEC95] AECMA: *A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language*. European Association of Aerospace Industries, 1995
- [Ams78] AMSTAD, Toni: *Wie verständlich sind unsere Zeitungen?*, Universität Zürich, unv. Dissertation, 1978
- [ASD07] ASD-STE 100: *ASD Simplified Technical English. Specification ASD-STE 100. International specification for the preparation of maintenance documentation in a controlled language*. 2007
- [ASW15] ADAM, Sebastian ; SEYFF, Norbert ; WÜNCH, Christian: 2014/15 RE Kompass Ergebnisbericht. Fraunhofer IESE, 2015. – Forschungsbericht
- [Aus63] AUSUBEL, David P.: *The Psychology of Meaningful Verbal Learning*. Grune & Stratton, 1963
- [AWK13] ADAM, Sebastian ; WÜNCH, Christian ; KOCH, Matthias: Ergebnisbericht RE-Kompass. Fraunhofer IESE, 2013. – Forschungsbericht
- [Bal09] BALZERT, Helmut: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. Spektrum, Akademischer Verlag, 2009
- [Bal14] BALLSTAEDT, Steffen-Peter: Perfekte Sprache gesucht. In: *technische kommunikation* 6 (2014), S. 66
- [Beh93] BEHRENS, Leila: Lexikalische Ambiguität und Disambiguierung im Kontext der maschinellen Übersetzung. In: *Studien zur Wortfeldtheorie/Studies in Lexical Field Theory* (1993), S. 251–268
- [Ben05] BENNÖHR, Jasmine: *A Web-based Personalised Textfinder for Language Learners*, University of Edinburgh, unv. Masterarbeit, 2005
- [Ber74] BERLYNE, Daniel E.: *Konflikt, Erregung, Neugier: Zur Psychologie der kognitiven Motivation*. Klett, 1974
- [Bie89] BIERWISCH, Manfred: The Semantics of Gradation. In: BIERWISCH, MANFRED ; LANG, EWALD (Hrsg.): *Dimensional Adjectives: Grammatical Structure and Conceptual Interpretation*. Springer-Verlag, 1989
- [Boe79] BOEHM, Barry W.: Guidelines for Verifying and Validating Software Requirements and Design Specification. In: SAMET, Paul A. (Hrsg.): *Proceedings of the*

- European Conference of Applied Information Technology (Euro IFIP)*, 1979, S. 711–719
- [Boe81] BOEHM, Barry W.: *Software Engineering Economics*. Prentice-Hall, 1981
- [Boh09] BOHNET, B.: Efficient Parsing of Syntactic and Semantic Dependency Structures. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task* Association for Computational Linguistics, 2009, S. 67–72
- [Bre12] BRENNEIS, Markus: Language Tool – Freie Stil- und Grammatikprüfung. In: *freiesMagazin* (2012), Nr. 08, S. 24–28
- [Buß08] BUSSMANN, Hadumod: *Lexikon der Sprachwissenschaft*. 4. Auflage. Alfred Kröner Verlag, 2008
- [CG99] CHRISTMANN, Ursula ; GROEBEN, Norbert: Psychologie des Lesens. In: FRANZMANN, BODO ; HASEMANN, KLAUS ; LÖFFLER, DIETRICH ; SCHÖN, ERICH (Hrsg.): *Handbuch Lesen*. K. G. Saur Verlag, 1999, S. 145–223
- [Cho65] CHOMSKY, Noam: *Aspects of the Theory of Syntax*. MIT Press, 1965
- [Clé00] CLÉMENT, Danièle: *WV studium*. Bd. 173: *Linguistisches Grundwissen: Eine Einführung für zukünftige Deutschlehrer*. Vandenhoeck & Ruprecht, 2000
- [Dai09a] DAIMLER AG: *Anforderungen schreiben, aber richtig!*, 2009. – Interne Schulung
- [Dai09b] DAIMLER AG: *Anforderungsregeln*, 2009. – Daimler-interne Leitlinie
- [DB06] DE BONO, Edward: *De Bono's Thinking Course*. BBC Active, 2006
- [DBRT99] DURÁN, Amador ; BERNÁRDEZ, Beatriz ; RUIZ, Antonio ; TORO, Miguel: A Requirements Elicitation Approach Based in Templates and Patterns. In: *Proceedings of the 2nd Workshop on Requirements Engineering* Universidad de Sevilla, 1999, S. 17–29
- [DGS04] DÖRRE, J. ; GERSTL, P. ; SEIFFERT, R.: Volltextsuche und Text Mining. In: *Computerlinguistik und Sprachtechnologie. Eine Einführung*. Bd. 2. Spektrum Akademischer Verlag, 2004, S. 479–495
- [Die13] DIE SOPHISTEN: *Schablonen für alle Fälle*. SOPHIST GmbH, 2013
- [DIN97] DIN 66 905: *DIN 66 905 - Projektwirtschaft - Projektabwicklung - Begriffe*. 1997

- [DZ11] DREWER, Petra ; ZIEGLER, Wolfgang: *Technische Dokumentation: eine Einführung in die übersetzungsgerechte Texterstellung und in das Content-Management*. Vogel Verlag, 2011
- [Ebe12] EBERT, Christof: *Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten*. dpunkt.verlag, 2012
- [EF04] EVERT, Stefan ; FITSCHEN, Arne: Textkorpora. In: *Computerlinguistik und Sprachtechnologie. Eine Einführung*. Bd. 2. Spektrum Akademischer Verlag, 2004, S. 406–413
- [EH11] EVERT, Stefan ; HARDIE, Andrew: Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In: *Proceedings of the Corpus Linguistics Conference 2011* University of Birmingham, 2011
- [Eis13] EISENBERG, Peter: *Der Satz. Grundriss der deutschen Grammatik*. Bd. 2. J.B. Metzler'sche Verlagsbuchhandlung, 2013
- [EKR09] EISENBERG, Peter ; KUNKEL-RAZUM, Kathrin: *Duden – Die Grammatik: unentbehrlich für richtiges Deutsch;[umfassende Darstellung des Aufbaus der deutschen Sprache vom Laut über das Wort und den Satz bis hin zum Text und zu den Merkmalen der gesprochenen Sprache; mit zahlreichen Beispielen, übersichtlichen Tabellen und Grafiken sowie ausführlichem Register]*. Dudenverlag, 2009
- [FFWE16] FEMMER, Henning ; FERNÁNDEZ, Daniel M. ; WAGNER, Stefan ; EDER, Sebastian: Rapid Quality Assurance with Requirements Smells. In: *Journal of Systems and Software* (2016), Nr. 123, S. 190–213
- [FHK⁺05] FUCHS, Norbert E. ; HÖFLER, Stefan ; KALJURAND, Kaarel ; RINALDI, Fabio ; SCHNEIDER, Gerold: Attempto Controlled English: A Knowledge Representation Language Readable by Humans and Machines. In: EISINGER, NORBERT ; MALUSZYŃSKI, JAN (Hrsg.): *Reasoning Web: First International Summer School 2005*, 2005, S. 213–250
- [FKK08] FUCHS, Norbert E. ; KALJURAND, Kaarel ; KUHN, Tobias: Attempto Controlled English for Knowledge Representation. In: BAROGLIO, CRISTINA ; BONATTI, PIERO A. ; MALUSZYŃSKI, JAN ; MARCHIORI, MASSIMO ; POLLERES, AXEL ; SCHAFFERT, SEBASTIAN (Hrsg.): *Reasoning Web: 4th International Summer School 2008*, 2008, S. 104–124

- [Fle48] FLESCHE, Rudolf: A New Readability Yardstick. In: *Journal of Applied Psychology* 32 (1948), Nr. 3, S. 221–233
- [Fle71] FLEISS, Joseph L.: Measuring nominal scale agreement among many raters. In: *Psychological Bulletin* 76 (1971), Nr. 5, S. 378–382
- [GCK10] GLEICH, Benedikt ; CREIGHTON, Oliver ; KOF, Leonid: Ambiguity Detection: Towards a Tool Explaining Ambiguity Sources. In: WIERINGA, ROEL ; PERSOON, ANNE (Hrsg.) ; Springer (Veranst.): *International Working Conference on Requirements Engineering: Foundation for Software Quality* Springer, 2010, S. 218–232
- [Gla72] GLASENAPP, Brent W.: Caterpillar Fundamental English. In: *Proceedings of the 19th International Technical Communication Conference* Society for Technical Communication, 1972, S. 81–85
- [Gri75] GRICE, Herbert P.: Logic and Conversation. In: COLE, PETER ; MORGAN, JERRY L (Hrsg.): *Syntax and Semantics 3: Speech Acts*, Academic Press, 1975, S. 41–58
- [Gro72] GROEBEN, Norbert: *Die Verständlichkeit von Unterrichtstexten: Dimensionen und Kriterien rezeptiver Lernstadien*, Universität Münster, Dissertation, 1972
- [Gro82] GROEBEN, Norbert: *Leserpsychologie: Textverständnis – Textverständlichkeit*. Aschendorff Münster, 1982
- [Hal04] HALAMA, Andre: Flache Satzverarbeitung. In: *Computerlinguistik und Sprachtechnologie. Eine Einführung*. Bd. 2. Spektrum Akademischer Verlag, 2004, S. 218–231
- [HCJ+09] HAJIČ, Jan ; CIARAMITA, Massimiliano ; JOHANSSON, Richard ; KAWAHARA, Daisuke ; MARTÍ, Maria A. ; MÀRQUEZ, Lluís ; MEYERS, Adam ; NIVRE, Joakim ; PADÓ, Sebastian ; ŠTĚPÁNEK, Jan et al.: The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task* Association for Computational Linguistics, 2009, S. 1–18
- [Her06] HERTEL, Guido: *Mercer-Studie Autoelektronik - Elektronik setzt die Impulse im Auto*. Press Release, 2006
- [HH08] HÖHN, Reinhard ; HÖPPNER, Stephan: *Das V-Modell XT: Grundlagen, Methodik und Anwendungen*. Springer-Verlag, 2008

- [HJD10] HULL, Elizabeth ; JACKSON, Ken ; DICK, Jeremy: *Requirements Engineering*. Springer Science & Business Media, 2010
- [Hol10] HOLL, Daniel: *Studia grammatica*. Bd. 71: *Modale Infinitive und dispositionelle Modalität im Deutschen*. Walter de Gruyter, 2010
- [IEE90] IEEE STD 610.12-1990: *IEEE Standard Glossary of Software Engineering Terminology*. 1990
- [IEE98] IEEE STD 830-1998: *IEEE Recommended Practice for Software Requirements Specifications*. 1998
- [IEE12] IEEE STD 1012TM-2012: *IEEE Standard for System and Software Verification and Validation*. 2012
- [ISO15] ISO/TS 24620-1: *Language resource management – Controlled natural language (CNL) – Part 1: Basic concepts and principles*. 2015
- [Juh05] JUHL, Dietrich: *Technische Dokumentation: Praktische Anleitungen und Beispiele*. Springer, 2005
- [Kam01] KAMSTIES, Erik: *Surfacing Ambiguity in Natural Language Requirements*, Universität Kaiserslautern, Dissertation, 2001
- [KDJH16] KRISCH, Jennifer ; DICK, Melanie ; JAUCH, Ronny ; HEID, Ulrich: A Lexical Resource for the Identification of “Weak Words” in German Specification Documents. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, European Language Resources Association (ELRA), 2016, S. 2846–2850
- [KH15] KRISCH, Jennifer ; HOUDEK, Frank: The Myth of Bad Passive Voice and Weak Words – An Empirical Investigation in the Automotive Industry. In: *23rd International Requirements Engineering Conference (RE)* Institute of Electrical and Electronics Engineers, 2015, S. 344–351
- [KKRH15] KLAUS, Alexander ; KOCH, Matthias ; REUTHER, Ursula ; HENSBERG, Gernot: ReqCheck: Automatisierte Qualitätssicherung von Anforderungsdokumenten mit Hilfe linguistischer Regeln. Fraunhofer IESE, 2015. – Forschungsbericht
- [KLM09] KNAUSS, Eric ; LÜBKE, Daniel ; MEYER, Sebastian: Feedback-Driven Requirements Engineering: The Heuristic Requirements Assistant. In: *Proceedings of the International Conference on Software Engineering* Institute of Electrical and Electronics Engineers, 2009, S. 587–590

- [KMN09] KÜBLER, Sandra ; MCDONALD, Ryan ; NIVRE, Joakim: Dependency Parsing. In: *Synthesis Lectures on Human Language Technologies 1* (2009), S. 1–127
- [Kna12] KNAUSS, Eric: Erfahrungsbasierte Verbesserung der Dokumentation von Anforderungen auf Basis von heuristischem Feedback. In: *Softwaretechnikrends 32* (2012), Nr. 4, S. 23–24
- [Kne06] KNEE, Cornelia: *Zielorientierte Überprüfung von natürlichsprachigen Anforderungsdokumenten*, Universität Ulm, unv. Diplomarbeit, 2006
- [KNL14] KASSAB, Mohamad ; NEILL, Colin ; LAPLANTE, Phillip: State of practice in requirements engineering: contemporary data. In: *Innovations in Systems and Software Engineering 10* (2014), Nr. 4, S. 235–241
- [Kof07] KOF, Leonid: Treatment of Passive Voice and Conjunctions in Use Case Documents. In: KEDAD, ZOUBIDA ; LAMMARI, NADIRA ; MÉTAIS, ELISABETH ; MEZIANE, FARID ; REZGUI, YACINE (Hrsg.): *Natural Language Processing and Information Systems*. Springer, 2007, S. 181–192
- [KR93] KAMP, Hans ; REYLE, Uwe: *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht, 1993
- [Kri13] KRISCH, Jennifer: *Identifikation kritischer Weak-Words aufgrund ihres Satzkontextes in Anforderungsdokumenten*, Universität Stuttgart, unv. Diplomarbeit, 2013
- [Kri14a] KRISCH, Jennifer: Kontextbasierte lexikalische Kontrolle von Anforderungsdokumenten. In: ABEL, ANDREA ; VETTORI, CHIARA ; RALLI, NATASCIA (Hrsg.): *Proceedings of the XVI EURALEX International Congress*, 2014, S. 647–656
- [Kri14b] KRISCH, Jennifer: Weak-Words und ihre Auswirkung auf die Qualität von Anforderungsdokumenten. In: *Softwaretechnikrends 34* (2014), Nr. 3, S. 14–15
- [Kri15] KRISCH, Jennifer: Regeln zur Erkennung problematischer Konjunktive in Anforderungen. In: *Softwaretechnikrends 36* (2015), Nr. 3, S. 28–29
- [Küh10] KÜHL, Christian: *Formalisierung von Requirements durch Nutzung von Templates*, Freie Universität Berlin, unv. Bachelorarbeit, 2010
- [LK77] LANDIS, J. R. ; KOCH, Gary G.: The Measurement of Observer Agreement for Categorical Data. In: *Biometrics 33* (1977), Nr. 1, S. 159–174

- [LKT⁺15] LANDHÄUSSER, Mathias ; KÖRNER, Sven ; TICHY, Walter ; KEIM, Jan ; KRISCH, Jennifer: DeNom: A Tool to Find Problematic Nominalizations using NLP. In: *2nd International Workshop on Artificial Intelligence for Requirements Engineering (AIRE 2015)* Institute of Electrical and Electronics Engineers, 2015, S. 9–16
- [LST74] LANGER, Inghard ; SCHULZ VON THUN, Friedemann ; TAUSCH, Reinhard: *Verständlichkeit in Schule, Verwaltung, Politik und Wissenschaft: Mit einem Selbsttrainingsprogramm zur verständlichen Gestaltung von Lehr- und Informationstexten*. Reinhardt, 1974
- [Maa15] MAASS, Christiane: *Leichte Sprache: Das Regelbuch*. LIT Verlag, 2015
- [Mar03] MARTEL, Elena: *Requirements Engineering*, Wilhelm Büchner Hochschule, unv. Seminararbeit, 2003
- [Mel00] MELCHISEDECH, Ralf: *Verwaltung und Prüfung natürlichsprachlicher Spezifikationen*, Universität Stuttgart, unv. Dissertation, 2000
- [MRS09] MANNING, Christopher D. ; RAGHAVAN, Prabhakar ; SCHÜTZE, Hinrich: *Introduction to Information Retrieval*. Cambridge University Press, 2009
- [MSS13] MUELLER, Thomas ; SCHMID, Helmut ; SCHÜTZE, Hinrich: Efficient Higher-Order CRFs for Morphological Tagging. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2013, 322–332
- [MW10] MAVIN, Alistair ; WILKINSON, Philip: Big Ears (The Return of “Easy Approach to Requirements Syntax”). In: *Proceedings of the 18th IEEE International Requirements Engineering Conference (RE’10)* IEEE Computer Society, 2010, S. 277–282
- [Nör12] NÖRENBERG, Ralf: *Effizienter Regressionstest von E/E-Systemen nach ISO 26262*, Karlsruher Institut für Technologie, Dissertation, 2012
- [O’R14] O’REGAN, Gerard: *Introduction to Software Quality*. Springer, 2014
- [Par97] PARADIS, Carita: *Lund Studies in English*. Bd. 92: *Degree modifiers of adjectives in spoken British English*. Lund University Press, 1997
- [Par10] PARTSCH, Helmuth: *Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme*. Springer-Verlag, 2010

- [PKS00] POL, Martin ; KOOMEN, Tim ; SPILLNER, Andreas: *Management und Optimierung des Testprozesses*. dpunkt.verlag, 2000
- [Poh96] POHL, Klaus: *Process-Centered Requirements Engineering*. John Wiley & Sons, Inc., 1996
- [Poh07] POHL, Klaus: *Requirements Engineering: Grundlagen, Prinzipien, Techniken*. dpunkt.verlag, 2007
- [PR15] POHL, Klaus ; RUPP, Chris: *Basiswissen Requirements Engineering. Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. dpunkt.verlag, 2015
- [Pus02] PUSCHNIG, Alexander: *Systematische Qualitätsanalyse natürlich-sprachiger Software-Anforderungsdokumente*, Universität Ulm, unv. Diplomarbeit, 2002
- [Ras06] RASCU, Ecaterina: A Controlled Language Approach to Text Optimization in Technical Documentation. In: BUTT, Miriam (Hrsg.): *Proceedings of KONVENS*, 2006, S. 107–114
- [Rup14] RUPP, Chris: *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil*. Carl Hanser Verlag, 2014
- [San90] SANTORINI, Beatrice: Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision). University of Pennsylvania, 1990. – Forschungsbericht
- [SC01] SI, Luo ; CALLAN, Jamie: A statistical model for scientific readability. In: PAQUES, HENRIQUE ; LIU, LING ; GROSSMAN, DAVID (Hrsg.): *Proceedings of the 10th International Conference on Information and Knowledge Management*, 2001, S. 574–576
- [Sch11] SCHWINN, Hans: *Requirements Engineering: Modellierung von Anwendungssystemen*. Oldenbourg Verlag, 2011
- [Sch12] SCHLENKHOFF, Andreas ; DUDENREDAKTION (Hrsg.): *Duden-Ratgeber Technische Dokumentation: Beschreibende und anleitende Texte erstellen*. Dudenverlag, 2012
- [Sch15] SCHÄFER, Gregor: Ein bisschen dumm stellen – Interview mit Friedemann Schulz von Thun. In: *technische kommunikation* 03 (2015), S. 18–20
- [Sch16] SCHMITZ, Anke: *Verständlichkeit von Sachtexten – Wirkung der globalen Textkohäsion auf das Textverständnis von Schülern*, Bergische Universität Wuppertal, unv. Dissertation, 2016

- [SK04] SCHIEHLEN, Michael ; KLABUNDE, Ralf: Semantik. In: *Computerlinguistik und Sprachtechnologie. Eine Einführung*. Bd. 2. Spektrum Akademischer Verlag, 2004, S. 276–332
- [SL12] SPILLNER, Andreas ; LINZ, Tilo: *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB-Standard*. dpunkt.verlag, 2012
- [SL15] SIEGEL, Melanie ; LIESKE, Christian: Beitrag der Sprachtechnologie zur Barrierefreiheit: Unterstützung für Leichte Sprache. In: *trans-kom* 8 (2015), S. 40–78
- [Smi03] SMITH, George: A brief introduction to the TIGER treebank, Version 1. Universität Potsdam, 2003. – Forschungsbericht
- [SS03] STÖCKEL, Frank ; STERR, Uwe: Anforderungen an Anforderungen. (2003). <http://www.hood-group.com>. – Zuletzt kontrolliert: 13. Januar 2017
- [SSUE09] STÖCKEL, Frank ; STOLZ, Philip ; UDDIN, Ifthaker ; ENDRISS, Larissa: DESIRE[®]: Dynamic Expert System for Improving Requirements. (2009). <http://www.hood-group.com>. – Zuletzt kontrolliert: 13. Januar 2017
- [Sto13] STOYANOVA, Nadya: *Verbesserung der Qualität natürlich-sprachlicher Spezifikationen*, Universität Stuttgart, unv. Dissertation, 2013
- [STST99] SCHILLER, Anne ; TEUFEL, Simone ; STÖCKERT, Christine ; THIELEN, Christine: Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). Version: 1999. <http://www.sfs.uni-tuebingen.de/resources/stts-1999.pdf>. Institut für Maschinelle Sprachverarbeitung Universität Stuttgart und Seminar für Sprachwissenschaft Universität Tübingen, 1999. – Forschungsbericht
- [SZ16] SCHÄUFFELE, Jörg ; ZURAWKA, Thomas: *Automotive Software Engineering: Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 6. Springer Vieweg, 2016
- [tek11] TEKOM (Hrsg.): *Regelbasiertes Schreiben – Deutsch für die Technische Kommunikation*. 1. Auflage. tekomp, 2011
- [Tex] *TextLab*. <https://www.text-lab.de/>. – Zuletzt kontrolliert: 13. Januar 2017
- [The] THE STANDISH GROUP: *The CHAOS Report 1994*. http://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf. – Zuletzt kontrolliert: 13. Januar 2017

- [TNA11] TSARFATY, Reut ; NIVRE, Joakim ; ANDERSSON, Evelina: Evaluating Dependency Parsing: Robust and Heuristics-Free Cross-Annotation Evaluation. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing* Association for Computational Linguistics, 2011, S. 385–396
- [War15] WARMERS, Alexandra: *Potentiell mehrdeutige pronominale Bezüge in Lastenheften*, Universität Hildesheim, unv. Masterarbeit, 2015
- [Wie99] WIEGERS, Karl: *Software Requirements*. Microsoft Press, 1999
- [Wun15] WUNDENBERG, Sven-Michael: *Requirement Engineering for Knowledge-Intensive Processes: Reference Architecture for the Selection of a Learning Management System*. Springer Gabler, 2015
- [Zim72] ZIMMERMANN, Harald H.: Zur Konzeption der automatischen Lemmatisierung von Texten. In: *SFB Elektronische Sprachforschung: Aspekte der Automatischen Lemmatisierung, Bericht 10-72* (1972), S. 4–10. – Manuskript