

# Interaction Room

Eine Methode zur Förderung der Wertorientierung in Planung und  
Requirements Engineering von Informationssystemen

DISSERTATION

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM

(Dr. rer. nat.)

durch Fakultät Wirtschaftswissenschaften  
der Universität Duisburg-Essen, Campus Essen

vorgelegt von

Simon Grapenthin, M. Sc.

geboren am 14. Februar 1983 in Essen

Tag der mündlichen Prüfung: 14.12.2017

1. Gutachter: Prof. Dr. Volker Gruhn  
2. Gutachter: Prof. Dr. Matthias Book

## Zusammenfassung

Software wird immer wichtiger in unserer Gesellschaft, trotzdem dauern IT-Projekte länger und werden teurer, als ursprünglich geplant, sie verfehlen ihre funktionalen und qualitativen Ziele oder werden vorzeitig abgebrochen. Durch inhärente Ungewissheit, Komplexität und mangelnde Wertorientierung wird eine realistische Projektplanung für die Entwicklung von Informationssystemen erschwert. Auf Basis eines unzureichenden Verständnisses über Anforderungen, Risiken und Ziele kann der zu leistende Aufwand nicht realistisch geschätzt werden, die daraufhin allokierten Ressourcen, bereitgestellten monetären Mittel und die geplante Projektlaufzeit reichen für eine erfolgreiche Projektumsetzung nicht aus.

Die initiale Verständnisbildung über Anforderungen und Risiken geschieht in plangetriebenen Vorgehensweisen zu Beginn eines Projekts durch eine umfangreiche Analyse und möglichst vollständige Spezifikation. Das Schreiben vollständiger Spezifikationen ist jedoch unwirtschaftlich, weil soziotechnische Systeme nicht vollständig beschreibbar sind und emergente Anforderungen in erkenntnisgetriebenen Softwareprozessen zu Änderungen führen. In agilen Vorgehensweisen wird das Verständnis durch die gleichen Aktivitäten, jedoch in kurzen Zyklen und ohne vollständige Spezifikation zu Projektbeginn, durch enge Zusammenarbeit zwischen Kunden und Entwicklungsteams hergestellt. Um Projekte erfolgreich planen zu können, müssen die richtigen Anforderungen in angemessenem Detailniveau verstanden und dokumentiert sein. Plangetriebene und agile Vorgehensweisen liefern den organisatorischen Rahmen zur Herstellung eines expliziten und impliziten gemeinsamen Verständnisses. Das Erkennen erfolgskritischer Projekthalte obliegt der Verantwortung einzelner Rolleninhaber in den Vorgehensweisen – eine strukturierte Herangehensweise, um die Aspekte eines Projektes explizit zu machen, die über Erfolg und Misserfolg eines Projektes entscheiden können, existiert nicht.

Um einen Beitrag zum Erfolg von IT-Projekten durch deren realistische Planung zu leisten, wird in dieser Arbeit der Interaction Room vorgestellt: Eine Methode, in der interdisziplinäre Teams pragmatische Modelle über das Verhalten sowie die Struktur eines Informationssystems in moderierten Workshops skizzieren. Sie kennzeichnen Wert-, Aufwands- und Risikotreiber in Modellskizzen, die auf Interaction-Room-Landkarten visualisiert werden. Dabei veranschaulicht die Feature-Landkarte den funktionalen Projektumfang. Die Prozesslandkarte veranschaulicht durch das Informationssystem unterstützte

Prozesse zur Erfüllung der funktionalen Anforderungen. Die Objektlandkarte veranschaulicht erzeugte und verarbeitete Geschäftsobjekte und die Integrationslandkarte veranschaulicht direkte Schnittstellen des Informationssystems zu benachbarten Systemen sowie ausgetauschte Geschäftsobjekte. Die Kennzeichnung von Wert-, Aufwands- und Risikotreiber geschieht mit Hilfe von getypten Annotationen, die von den Stakeholdern des interdisziplinären Teams auf die Landkarten geheftet, im Anschluss diskutiert und dokumentiert werden. Durch die Annotationstypen lassen sich z. B. Werte für Nutzer, Flexibilitätsanforderungen und Ungewissheit kennzeichnen. Die Erkenntnisse aus Interaction-Room-Workshops werden in eine pragmatische Struktur für Anforderungsdokumente überführt und können dadurch unmittelbar in Softwareprozessen weiter verwendet werden.

Zwischen November 2011 und Februar 2017 wurden 23 Analyse-, Neuentwicklungs-, Weiterentwicklungs- und Migrationsprojekte mit Industriepartnern unterschiedlicher Branchen durchgeführt. Auf dieser Datenbasis wurden unterschiedliche Methodenparameter auf Zusammenhänge untersucht. Erwartungsgemäß bestehen positive Zusammenhänge zwischen Metriken der Projektgröße, zwischen dem Vorschlag zur Verwendung von Annotationstypen und deren tatsächlicher Verwendung, sowie zwischen der Menge verwendeter Annotationstypen und der Anzahl Duplikate. Außerdem besitzen die Interaction-Room-Annotationen positiven Einfluss auf die Genauigkeit der Aufwandsschätzung. Die Planung von Entwicklungsiterationen kann durch Interaction-Room-Landkarten unterstützt werden, wodurch eine quantitative Verbesserung bei der Identifikation von Arbeitspaketen beobachtet wurde. Die Verbesserung besitzt positiven Einfluss auf die Fortschrittsmessung während einer Iteration und damit auf ihre Erfolgsprognose. Die Intervention kann in ein übliches Vorgehen zur Planung von Entwicklungsiterationen integriert werden, weil sie von den Planenden nicht als störend wahrgenommen wurde.

## Danksagung

Ich möchte mich bei den Personen bedanken, die mich während dieser Arbeit besonders unterstützt haben. Leider können nicht alle namentlich erwähnt werden, was meinen Dank Ihnen gegenüber allerdings nicht schmälert.

Ein besonderer Dank geht an Volker Gruhn für die Betreuung dieser Arbeit. Seine Erfahrung und sein Wissen haben mir sehr bei der Präzision geholfen und waren wegweisend für die Ausrichtung der Arbeit. Ebenso danke ich ihm für die Unterstützung, den Interaction Room praxisnah entwickeln zu können, was ohne ihn nicht möglich gewesen wäre. Ich danke aktuellen und ehemaligen Kollegen des Lehrstuhls, insbesondere Tobias Brückmann und Thomas Richter, für ihre ermunternde Empathie während des Promotionsprozesses sowie Erik Hebisch und Markus Kleffmann, für den fachlichen Austausch und die wertvollen Diskussionen. Mein außerordentlicher Dank gilt Matthias Book für seine Geduld in zahlreichen Diskussionen sowie sein immer konstruktives und umfangreiches Feedback. Dein kontinuierliches Interesse ist bemerkenswert.

Ich danke allen Freunden, die mich auf diesem Weg begleitet haben, insbesondere Thomas und Kerstin Kölker. Kerstin, die mit meiner Frau Ina mehrfach auf Reisen gegangen ist als ich mit der Dissertationsschrift beschäftigt war. Thomas für die Aufnahme in die WG GH19 während dieser Zeit. Nach langen Tagen am Schreibtisch haben die Abende dazu beigetragen am nächsten Morgen motiviert weiter zu machen.

Ein großer Dank gilt meiner Familie, die mich auf dem Weg zu diesem Ziel von klein auf unterstützt hat. Ina, ich danke Dir für Deine Rücksicht und Nachsicht, den vielen Freiraum den Du mir gelassen und geschaffen hast. Ohne Dich hätte ich die Arbeit vermutlich nicht erfolgreich beendet. Zuletzt möchte ich meinem Vater Horst dafür danken immer mein größtes Vorbild und größter Kritiker gewesen zu sein. Ihm widme ich diese Arbeit.

# Inhaltsverzeichnis

Abkürzungsverzeichnis .....	VIII
Tabellenverzeichnis .....	IX
Abbildungsverzeichnis .....	XI
Formelverzeichnis .....	XIII
1 Einleitung .....	14
1.1 Erfolg von IT-Projekten .....	15
1.2 Ursachen für den Misserfolg von IT-Projekten .....	17
1.3 Ebenen des Erfolgs von IT-Projekten .....	19
1.4 Struktur der Arbeit .....	21
2 Problemstellung .....	23
2.1 Herausforderungen bei der Entwicklung von Informationssystemen.....	23
2.1.1 Ungewissheit.....	24
2.1.2 Komplexität .....	24
2.1.3 Wertorientierung.....	25
2.2 Grenzen plangetriebener und agiler Vorgehensweisen.....	26
2.3 Gemeinsames Verständnis als Grundlage des Projekterfolgs.....	28
3 Zielsetzung, Anforderungen und Vorgehen zur Erarbeitung des Lösungsansatzes .....	31
3.1 Zielsetzung des Lösungsansatzes.....	31
3.2 Anforderungen an den Lösungsansatz .....	32
3.2.1 Pragmatische Modellierung relevanter Perspektiven für die Analyse und Konzeption von Informationssystemen .....	32
3.2.2 Moderierte Kollaboration heterogener Teams .....	33
3.2.3 Fokussierung auf die wesentlichen Anforderungen.....	34
3.3 Vorgehen zur Erarbeitung des Lösungsansatzes.....	34
3.3.1 Methodenentwicklung .....	35
3.3.2 Methodenevolution .....	37
4 Verwandte Ansätze .....	40
4.1 Participatory Design.....	41
4.2 Joint Application Design/Development.....	42
4.3 Soft Systems Methodology .....	44
4.4 Participatory IT-Design .....	47
4.5 Design Thinking.....	49
4.6 Accelerated Solution Environment.....	52
5 Der Interaction Room.....	53
5.1 Überblick.....	54

---

5.1.1	Rollen.....	55
5.1.2	Methodenbaustein Vorbereitung .....	57
5.1.3	Methodenbaustein Interaction-Room-Landkarten.....	58
5.1.4	Methodenbaustein Interaction-Room-Annotationen .....	59
5.1.5	Methodenbaustein Erkenntnisdokumentation .....	59
5.2	Interaction-Room-Landkarten.....	60
5.2.1	Feature-Landkarte.....	61
5.2.2	Prozesslandkarte .....	62
5.2.3	Integrationslandkarte .....	66
5.2.4	Objektlandkarte.....	68
5.2.5	Praxisbeispiel.....	70
5.3	Interaction-Room-Annotationen .....	78
5.3.1	Konzept der Annotationen.....	79
5.3.2	Verwandte Ansätze .....	80
5.3.3	Annotationskategorien.....	82
5.3.4	Herleitung und Beschreibung ausgewählter Annotationstypen.....	85
5.3.5	Verwendung der Annotationen.....	95
5.3.6	Praxisbeispiel.....	99
5.4	Erkenntnisdokumentation .....	107
5.4.1	Struktur von Anforderungsdokumenten .....	108
5.4.2	Abbildung der Interaction-Room-Erkenntnisse auf eine Struktur für Anforderungsdokumente.....	111
6	Evaluation .....	115
6.1	Forschungsfragen.....	117
6.2	Statistik zur Überprüfung der Forschungsfragen.....	121
6.2.1	Korrelation nach Bravais-Pearson .....	121
6.2.2	Kontingenzanalyse mit dem Pearson-Chi-Quadrat-Test .....	123
6.3	Ergebnisse .....	125
6.3.1	Forschungsfrage 1: Besteht ein Zusammenhang zwischen den Metriken der Projektgröße?.....	125
6.3.2	Forschungsfrage 2: Besteht ein Zusammenhang zwischen zur Verwendung vorgeschlagenen und verwendeten Annotationstypen? .....	127
6.3.3	Forschungsfrage 3: Besteht ein Zusammenhang zwischen der Menge verwendeter Annotationen und der Anzahl von Duplikaten?.....	129
6.4	Diskussion.....	132
6.4.1	Konstruktvalidität .....	132
6.4.2	Interne Validität .....	133
6.4.3	Externe Validität.....	134
6.4.4	Fazit .....	134

---

7	Fallstudie: Unterstützung der Aufwandsschätzung durch Interaction-Room-Annotationen .....	136
7.1	Forschungsfragen .....	138
7.2	Methodik zur Überprüfung der Forschungsfragen .....	139
7.2.1	Studienaufbau .....	141
7.2.2	Berechnungen zur Beantwortung der Forschungsfragen.....	142
7.3	Ergebnisse .....	143
7.3.1	Forschungsfrage 1: Wird das gemeinsame Verständnis unter Stakeholdern über Faktoren, die für das Schätzen relevant sind, durch Annotationen unterstützt? .....	145
7.3.2	Forschungsfrage 2: Existiert ein Unterschied in der Schätzgenauigkeit zwischen Schätzungen, die mit Annotationen und Planning Poker durchgeführt werden, im Gegensatz zu Schätzungen, die ohne Annotationen und mit Planning Poker durchgeführt werden? .....	146
7.3.3	Forschungsfrage 3: Existiert ein Unterschied in der Abweichung der Schätzgenauigkeit, wenn Schätzungen mit Annotationen und Planning Poker durchgeführt werden, im Gegensatz zu Schätzungen, die ohne Annotationen und mit Planning Poker durchgeführt werden? .....	147
7.4	Diskussion.....	148
7.4.1	Konstruktvalidität .....	148
7.4.2	Interne Validität .....	148
7.4.3	Externe Validität.....	149
7.4.4	Fazit .....	150
8	Fallstudie: Unterstützung der Sprint-Planung durch Interaction-Room-Landkarten .....	151
8.1	Ausgangssituation .....	151
8.2	Problemstellung .....	153
8.3	Lösungsansatz .....	154
8.4	Ergebnisse .....	156
8.4.1	Quantitative Ergebnisse aus Projekt A .....	156
8.4.2	Quantitative Ergebnisse aus Projekt B .....	158
8.4.3	Qualitative Ergebnisse aus Projekt A .....	159
8.4.4	Qualitative Ergebnisse aus Projekt B .....	160
8.5	Diskussion.....	162
8.5.1	Validität der Ergebnisse.....	162
8.5.2	Fazit .....	163
9	Fazit.....	164
9.1	Beiträge .....	164
9.2	Diskussion.....	165
9.3	Ausblick .....	166

---

Quellenverzeichnis .....	169
Anhang .....	191
A    Überblick aller Methodenbausteine, Prozess- und Produktelemente ....	191
B    Fragebogen zur Vorbereitung eines Interaction-Room-Workshops .....	194
C    Annotationstypen .....	197
C.1    Annotationstyp Business Value .....	197
C.2    Annotationstyp User Value.....	198
C.3    Annotationstyp Innovation .....	199
C.4    Annotationstyp Sicherheit .....	200
C.5    Annotationstyp Hohe Last .....	201
C.6    Annotationstyp Zeitbeschränkung .....	202
C.7    Annotationstyp Korrektheit .....	203
C.8    Annotationstyp Zuverlässigkeit .....	204
C.9    Annotationstyp Benutzbarkeit .....	205
C.10    Annotationstyp Flexibilität .....	206
C.11    Annotationstyp Mobilität.....	207
C.12    Annotationstyp Manuelle Bearbeitung .....	208
C.13    Annotationstyp Automatisierung.....	209
C.14    Annotationstyp Externe Ressource.....	210
C.15    Annotationstyp Randbedingung .....	211
C.16    Annotationstyp Verbesserungsbedarf .....	212
C.17    Annotationstyp Ablösung .....	213
C.18    Annotationstyp Unveränderlichkeit.....	214
C.19    Annotationstyp Komplexität.....	215
C.20    Annotationstyp Ungewissheit.....	216
D    Vorlage zur Dokumentation von Annotationen.....	217



## Abkürzungsverzeichnis

ADM	Außendienstmitarbeiter
ASE	Accelerated Solution Design
BD	Bezirksdirektion
BezDir	Bezirksdirektor
BPMN	Business Process Model and Notation
BRE	Balanced Measure of Relative Error
CATWOE	Customer, Actor, Transformation, Weltanschauung, Owner, Environment
CMS	Content Management System
CRM	Customer Relationship Management
CSV	Comma Separated Value
DT	Design Thinking
DW	Data Warehouse
EPK	ereignisgesteuerte Prozessketten
HAS	Human Activity System
IR	Interaction Room
IR-DC	Interaction-Room-Domänen-Coach
IR-MC	Interaction-Room-Methoden-Coach
JAD	Joint Application Design/Development
KAG	Kapitalanlagegesellschaft
LS	Lead-System
MRE	Measure of Relative Error
PD	Participatory Design
PO	Product Owner
QS	Qualitätssicherung
RE	Requirements Engineering
SPM	Sprint Planning Meeting
SSM	Soft Systems Methodology
UML	Unified Modeling Language
VID	Vertriebsinnendienst
VNR	Versicherungsnummer
VU	Versicherungsunternehmen

## Tabellenverzeichnis

Tabelle 1: Ergebnisse der CHAOS Reports von 1994 bis 2012.....	16
Tabelle 2: Ebenen für die Erfolgsbewertung von IT-Projekten .....	20
Tabelle 3: Sieben Empfehlungen für die Prozessmodellierung.....	63
Tabelle 4: Vorlage zur Beschreibung von Annotationstypen.....	86
Tabelle 5: Annotationstyp User Value .....	87
Tabelle 6: Annotationstyp Hohe Last.....	87
Tabelle 7: Annotationstyp Flexibilität.....	88
Tabelle 8: Annotationstyp Manuelle Bearbeitung.....	90
Tabelle 9: Annotationstyp Randbedingung .....	91
Tabelle 10: Annotationstyp Ablösung.....	92
Tabelle 11: Annotationstyp Komplexität .....	93
Tabelle 12: Überblick über die Annotationstypen.....	94
Tabelle 13: Vorlage zur Erläuterung der Annotationstypen.....	96
Tabelle 14: Ausgangslagen und Ziele durchgeführter Projekte (Auszug) .....	116
Tabelle 15: Anzahl der Landkarten und Annotationen je Landkarte .....	117
Tabelle 16: Korrelation zwischen Workshop-Tagen, Landkarten und Annotationen.....	126
Tabelle 17: Korrelation zwischen Workshop-Tagen, Landkarten und Annotationen exklusive des größten Projekts .....	126
Tabelle 18: Korrelation zwischen zur Verwendung vorgeschlagenen und verwendeten Annotationstypen .....	127
Tabelle 19: Korrelation zwischen Vorschlag und Verwendung je Annotationstyp .....	128
Tabelle 20: Korrelation zwischen der Menge verwendeter Annotationen und der Multimenge verwendeter Annotationen .....	129
Tabelle 21: Korrelation zwischen der Menge verwendeter Annotationen und der Multimenge verwendeter Annotationen exklusive des größten Projekts .	130
Tabelle 22: Erkenntnislose Annotationen aller Projekte .....	131
Tabelle 23: Erkenntnislose Annotationen exklusive des größten Projekts .....	131
Tabelle 24: Aufzeichnung der Schätzungen und tatsächlichen Aufwände (Auszug) ..	142
Tabelle 25: Deskriptive Statistik für komplett implementierte User Stories.....	144
Tabelle 26: Beitrag der Annotationen zum gemeinsamen Verständnis .....	146
Tabelle 27: Einfluss der Annotationen auf die Schätzgenauigkeit.....	146
Tabelle 28: Einfluss der Annotationen auf die Abweichung in Schätzungen .....	147
Tabelle 29: Ergebnisse des Task Breakdowns nach Einführung des Interaction Rooms in Projekt A .....	157
Tabelle 30: Ergebnisse des Task Breakdowns nach Einführung des Interaction Rooms in Projekt B.....	159
Tabelle 31: Annotationstyp Business Value.....	197
Tabelle 32: Annotationstyp User Value .....	198

---

Tabelle 33: Annotationstyp Innovation .....	199
Tabelle 34: Annotationstyp Sicherheit .....	200
Tabelle 35: Annotationstyp Hohe Last .....	201
Tabelle 36: Annotationstyp Zeitbeschränkung .....	202
Tabelle 37: Annotationstyp Korrektheit .....	203
Tabelle 38: Annotationstyp Zuverlässigkeit .....	204
Tabelle 39: Annotationstyp Benutzbarkeit .....	205
Tabelle 40: Annotationstyp Flexibilität .....	206
Tabelle 41: Annotationstyp Mobilität .....	207
Tabelle 42: Annotationstyp Manuelle Bearbeitung .....	208
Tabelle 43: Annotationstyp Automatisierung .....	209
Tabelle 44: Annotationstyp Externe Ressource .....	210
Tabelle 45: Annotationstyp Randbedingung .....	211
Tabelle 46: Annotationstyp Verbesserungsbedarf .....	212
Tabelle 47: Annotationstyp Ablösung .....	213
Tabelle 48: Annotationstyp Unveränderlichkeit .....	214
Tabelle 49: Annotationstyp Komplexität .....	215
Tabelle 50: Annotationstyp Ungewissheit .....	216

## Abbildungsverzeichnis

Abbildung 1: Klassifikation von gemeinsamem Verständnis .....	29
Abbildung 2: Zusammenhang von Metamodell, Methode und Methodeninstanz .....	36
Abbildung 3: Vorgehen zur Erarbeitung des initialen Methodenentwurfs.....	36
Abbildung 4: Vorgehen zur Erarbeitung des Interaction Rooms .....	38
Abbildung 5: Die 5 Phasen von Joint Application Design/Development .....	42
Abbildung 6: Abstrakter Prozess der Soft Systems Methodology .....	45
Abbildung 7: Participatory-IT-Design-Phasen.....	48
Abbildung 8: Prozessphasen des Design Thinking .....	50
Abbildung 9: Metamodell des Interaction Rooms.....	54
Abbildung 10: Metamodell der Interaction-Room-Landkarten.....	60
Abbildung 11: Metamodell der Feature-Landkarte .....	61
Abbildung 12: Metamodell der Prozesslandkarte .....	64
Abbildung 13: Grafische Darstellung der Elemente einer Prozesslandkarte .....	65
Abbildung 14: Metamodell der Integrationslandkarte.....	67
Abbildung 15: Grafische Darstellung der Elemente einer Integrationslandkarte.....	67
Abbildung 16: Metamodell der Objektlandkarte.....	68
Abbildung 17: Grafische Darstellung obligatorischer Elemente einer Objektlandkarte	69
Abbildung 18: Grafische Darstellung optionaler Elemente einer Objektlandkarte.....	69
Abbildung 19: Feature-Landkarte des Praxisbeispiels .....	72
Abbildung 20: Prozesslandkarte „Lead erfassen“ .....	73
Abbildung 21: Prozesslandkarte „Leads zuordnen“ .....	74
Abbildung 22: Prozesslandkarte „Leads nachverfolgen“ .....	75
Abbildung 23: Integrationslandkarte des Praxisbeispiels.....	76
Abbildung 24: Objektlandkarte des Praxisbeispiels.....	77
Abbildung 25: Metamodell der Interaction-Room-Annotationen .....	79
Abbildung 26: Softwarequalitätsmerkmale nach ISO 25010.....	85
Abbildung 27: Annotationsprozess .....	95
Abbildung 28: Annotationstyp diskutieren und dokumentieren .....	98
Abbildung 29: Zur Verwendung vorgeschlagene Annotationstypen für die Prozesslandkarten des Praxisbeispiels .....	100
Abbildung 30: Annotierte Prozesslandkarte „Lead erfassen“ .....	101
Abbildung 31: Annotierte Prozesslandkarte „Leads zuordnen“.....	102
Abbildung 32: Annotierte Prozesslandkarte „Leads nachverfolgen“.....	103
Abbildung 33: Zur Verwendung vorgeschlagene Annotationstypen für die Objekt- und die Integrationslandkarte des Praxisbeispiels .....	104
Abbildung 34: Annotierte Integrationslandkarte des Praxisbeispiels.....	105
Abbildung 35: Annotierte Objektlandkarte des Praxisbeispiels.....	106
Abbildung 36: Beispiele für Ergebnisse des Interaction Rooms in Anforderungsdokumenten.....	108

---

Abbildung 37: Abbildung von Artefakten aus Workshops in eine Dokumentstruktur .....	109
Abbildung 38: Abbildung von Artefakten aus Interaction-Room-Workshops in eine Dokumentstruktur .....	111
Abbildung 39: Durchgeführte Projekte nach Typen und Branchen .....	115
Abbildung 40: Workshop-Tage, Landkarten und Annotationen je Projekt.....	117
Abbildung 41: Zur Verwendung vorgeschlagene und verwendete Annotationstypen je Projekt .....	118
Abbildung 42: Menge und Multimenge verwendeter Annotationen.....	119
Abbildung 43: Grafische Darstellung zur Erläuterung der Herleitung des Bestimmtheitsmaßes .....	122
Abbildung 44: Histogramm des Variationskoeffizienten der Schätzungen .....	145
Abbildung 45: Burndown Charts von zwei Sprints vor der Einführung des Interaction Rooms .....	154
Abbildung 46: Vorgehen zur Skizzierung von Backlog Items für den Task Breakdown .....	155
Abbildung 47: Task Breakdown mit Hilfe von Interaction-Room-Landkarten .....	156
Abbildung 48: Burndown Charts der drei Sprints aus Projekt A nach Einführung des Interaction Rooms.....	157
Abbildung 49: Burndown Charts der drei Sprints aus Projekt B nach Einführung des Interaction Rooms.....	158
Abbildung 50: Methodenbausteine, Prozess- und Produktelemente des Interaction Rooms (1/3) .....	192
Abbildung 51: Methodenbausteine, Prozess- und Produktelemente des Interaction Rooms (2/3) .....	192
Abbildung 52: Methodenbausteine, Prozess- und Produktelemente des Interaction Rooms (3/3) .....	193
Abbildung 53: Fragebogen zur Vorbereitung von Interaction-Room-Workshops (1/3).....	194
Abbildung 54: Fragebogen zur Vorbereitung von Interaction-Room-Workshops (2/3).....	195
Abbildung 55: Fragebogen zur Vorbereitung von Interaction-Room-Workshops (3/3).....	196
Abbildung 56: Vorlage zur Dokumentation von Annotationstypen.....	217

## Formelverzeichnis

Formel 1: Berechnung der Kovarianz .....	121
Formel 2: Korrelationskoeffizient $r$ nach Bravais-Pearson .....	121
Formel 3: Berechnung des Bestimmtheitsmaßes $B^2$ .....	123
Formel 4: Bestimmtheitsmaß eines linearen Zusammenhangs .....	123
Formel 5: Berechnung der relativen Abweichung tatsächlich auftretender und theoretischer Häufigkeiten.....	124
Formel 6: Berechnung von Chi-Quadrat als Vorbereitung zur Bestimmung des Kontingenzkoeffizienten .....	124
Formel 7: Kontingenzkoeffizient nach Pearson .....	124
Formel 8: Korrigierter Kontingenzkoeffizient nach Pearson .....	124
Formel 9: Berechnung des Variationskoeffizienten .....	143
Formel 10: Balancierte relative Abweichung.....	143
Formel 11: Richtung und Größe der balancierten relativen Abweichung.....	143

# 1 Einleitung

„In the future, all companies will be software companies.“

(George F. Colony, CEO Forrester Research [1])

Software ist ein immer wichtiger werdender Bestandteil unseres Alltags [2], [3]. Viele Produkte und Dienstleistungen nutzen Software als Differenzierungsmerkmal, wobei die implementierten Codezeilen zwischen einigen Hundert und mehreren Millionen variieren [4]–[7]. Durch die Erweiterung physischer Objekte mit digitaler Technologie liefern diese Objekte Daten, die in Wertschöpfungsketten verarbeitet werden und dazu führen, dass einzelne Prozesse, Akteure und Komponenten situativ und autonom gesteuert werden. Wesentlicher Innovationstreiber ist nicht die Hardware und Mechanik, sondern die Software, sodass zunehmend in die softwarenahe Forschung investiert wird [3], [8]. Neben der Integration softwareintensiver eingebetteter Systeme werden auch Geschäftsprozesse als Teil vertikaler Wertschöpfungsketten digitalisiert. Daher ist die Konzeption und Entwicklung integrierter Informationssysteme ein zentraler Baustein der vierten industriellen Revolution [9], [10].

Nicholas Negroponte hat bereits 1997 die These formuliert, dass alles, was digitalisiert werden kann, langfristig digitalisiert werden wird [11]. Richard Charette hat 2005 in einem Artikel geschrieben: „Software is everywhere“ [12]. Marc Andreessen hat die zunehmende Bedeutung von Software 2011 mit den Worten „Software is eating the world“ [13] beschrieben. Software ist der Innovationstreiber, der Unternehmen und ganze Branchen dazu zwingt, ihre Geschäftsmodelle anzupassen, zu erweitern oder neue zu entwickeln. Verhältnismäßig junge Unternehmen erobern durch softwareintensive Dienste und Produkte bestehende Märkte und greifen dadurch die Marktanteile etablierter Unternehmen an [13], [14]. Der Wettbewerb erzeugt einen Innovationsdruck, der zu kürzer werdenden Innovationszyklen führt, was den Trend vom Produkteigentum zur Servicemiete begünstigt. Durch die Servicemiete sinkt die langfristige Kapitalbindung, wodurch die Möglichkeit steigt, auf veränderte Markt- oder Kundenbedürfnisse reagieren zu können [15].

Die ubiquitäre Verfügbarkeit des Internets ist ein wesentlicher Treiber der zunehmenden Relevanz von Software. Im Jahr 2014 hatten 2,9 Milliarden Menschen einen Zugang zum Internet, im Jahr 2016 waren es bereits 3,19 Milliarden Menschen und für das Jahr 2019 werden 3,9 Milliarden Menschen mit Internetzugang prognostiziert [16]. Die Verfügbarkeit mobiler Internetzugänge ermöglicht Menschen in strukturschwachen Regionen, das Internet zu benutzen,

denn die Abdeckung von 3G-Technologie umfasste bereits im Jahr 2014 192 Länder [17]. Die Verbreitung dieser Technologie hat einen Milliardenmarkt für mobile Anwendungen geschaffen, die unter anderem zu einer höheren Lebensqualität von behinderten, kranken und alten Menschen beitragen können [18]–[20].

Die Kosten von Infrastruktur zur Entwicklung und für den Betrieb von Software sind durch Cloud-Computing gesunken. Dadurch ist auch die Markteintrittsbarriere gesunken und die Geschäftsmodelle heutiger Unternehmen können mit geringerem Kapitalbedarf durch softwareintensive Produkte und Dienstleistungen angegriffen werden [21]. Situativ notwendige automatische Provisionierung und Deprovisionierung von Ressourcen ermöglicht, dass Anwendungen mit zunehmender Marktdurchdringung skalieren, ohne dass ungenutzte Ressourcen vorgehalten werden müssen [22].

Trotz zunehmender Relevanz von Software und Informationstechnologie überschreiten IT-Projekte ihre zeitlichen und monetären Rahmenbedingungen, verfehlen notwendige funktionale und qualitative Anforderungen oder werden vorzeitig abgebrochen [23], weshalb in dieser Arbeit der Interaction Room und sein Beitrag zur erfolgreichen Planung von IT-Projekten vorgestellt wird.

In den folgenden Unterkapiteln wird veranschaulicht, dass die erfolgreiche Umsetzung von IT-Projekten eine aktuelle Herausforderung darstellt (vgl. Kapitel 1.1), erläutert, auf welche Ursachen der Misserfolg zurückzuführen ist (vgl. Kapitel 1.2), und dargelegt, dass der Projekterfolg auf unterschiedlichen Ebenen sowie aus unterschiedlichen Perspektiven bewertet wird (vgl. Kapitel 1.3). Zum Abschluss des ersten Kapitels wird die Struktur dieser Arbeit vorgestellt (vgl. Kapitel 1.4).

Aus Gründen der einfacheren Lesbarkeit wird in dieser Dissertation die männliche Sprachform bei personenbezogenen Substantiven und Pronomen verwendet. Diese sprachliche Vereinfachung ist als geschlechtsneutral zu verstehen und soll keine Benachteiligung des weiblichen Geschlechts darstellen.

## 1.1 Erfolg von IT-Projekten

In vielen Veröffentlichungen wird von einer Häufung fehlgeschlagener IT-Projekte berichtet (siehe z. B. [23]–[26]), wobei einige Autoren die Ergebnisse der CHAOS Reports (siehe z. B. [24], [25], [27], [28]) referenzieren. Die Standish Group hat zwischen 1994 und 2012 sieben



Versionen des CHAOS Reports veröffentlicht [29], [30]. Durch die wiederholte Befragung von Projektverantwortlichen wurden erfolgreiche, herausgeforderte und gescheiterte Projekte identifiziert. Erfolgreiche Projekte haben die monetären und zeitlichen Rahmenbedingungen eingehalten, herausgeforderte Projekte haben diese überschritten, wurden jedoch beendet, und gescheiterte Projekte haben die Rahmenbedingungen überschritten und wurden abgebrochen. Die Ergebnisse dieser Untersuchungen sind in Tabelle 1 zusammengefasst.

Tabelle 1: Ergebnisse der CHAOS Reports von 1994 bis 2012 (angelehnt an [29], [30])

<i>Jahr</i>	<i>erfolgreich (in %)</i>	<i>herausfordernd (in %)</i>	<i>gescheitert (in %)</i>
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2008	32	44	24
2010	37	42	21
2012	39	43	18

Die Standish Group stellt eine Verbesserung über die Zeit fest, wobei 2012 weniger als die Hälfte der untersuchten Projekte erfolgreich abgeschlossen wurde. Die Ergebnisse der CHAOS Reports sind Gegenstand akademischer Diskussion [29], [31]–[33]. Es wird bemängelt, dass der Projektkontext bei der Beurteilung des Erfolgs nicht berücksichtigt wird [33], denn ob eine funktional vollständige und qualitativ hochwertige Software, deren Entwicklung 10 % Mehrkosten verursacht hat, als nicht erfolgreiches Projekt bezeichnet werden kann, ist fraglich [31]. Außerdem wird die Anzahl nicht erfolgreicher Projekte [29], [31] und die Höhe der budgetären Überschreitung [33], [34] in Frage gestellt. Abseits der Diskussion über die Validität der CHAOS Reports besteht jedoch Einigkeit, dass viele Projekte ihre monetären und zeitlichen Rahmenbedingungen verfehlen (siehe z. B. [23], [26], [29], [31]–[33], [35]).

In jüngeren Studien wird von besonders kritischen Projekten berichtet, die ihre monetären und zeitlichen Rahmenbedingungen so weit verfehlen, dass dadurch Firmen, Städte oder ganze Staaten in existenzielle Gefahr geraten [23], [26]. Flyvbjerg und Budzier haben im Rahmen einer weltweiten Studie 1471 IT-Projekte untersucht und festgestellt, dass jedes sechste Projekt

das veranschlagte Budget um mehr als 200 % und die Laufzeit um mehr als 70 % überschritten hat [23]. Bloch, Blumberg und Laartz [26] haben 5400 IT-Projekte untersucht und festgestellt, dass 45 % der Projekte mit einem Budget von mehr als 15 Millionen US-Dollar teurer wurden, 7 % länger dauerten und 56 % weniger Wert lieferten, als erwartet. In Summe kosteten alle untersuchten IT-Projekte 66 Milliarden US-Dollar mehr, als zunächst veranschlagt worden war. 17 % der untersuchten Projekte gefährdeten die Existenz der Firmen, in denen sie durchgeführt wurden.

Ob IT-Projekte eine existentielle Gefahr für Organisationen bedeuten oder nicht, macht keinen Unterschied im Hinblick auf die Ursachen für den Misserfolg von IT-Projekten [36], welche im folgenden Kapitel dargestellt werden.

## 1.2 Ursachen für den Misserfolg von IT-Projekten

Taylor [37] hat durch eine Befragung von 38 Mitgliedern eines britischen Projektmanagement-Verbandes und durch die Analyse von 1027 IT-Projekten Ursachen für gescheiterte Projekte identifiziert. Unklare Anforderungen und Ziele, fehlendes Commitment der Fachbereiche und sich ändernde fachliche Anforderungen zählen zu den Hauptursachen für das Scheitern. Der Autor kommt zu dem Schluss, dass sich das Scheitern nicht auf einzelne Gründe zurückführen lässt.

Robert Glass [38] sieht die Hauptursache für das Scheitern von Projekten im Abwägen zwischen Zeit, Kosten und Qualität. Eine Ursache für die Überschreitung von Zeit und Budget liegt in optimistischen Schätzungen. Er bemängelt, dass Schätzungen auf unzureichender Informationsbasis beruhen und nicht von den späteren Entwicklern vorgenommen werden. Er bemerkt, wie auch Charette [12], dass Schätzungen häufig Wünschen gleichen [39]. Ferner bemängelt Glass, dass keine Adjustierung der Schätzung erfolgt, wenn neue Anforderungen hinzukommen oder Anforderungen sich ändern. Der aus optimistischen Schätzungen resultierende Zeitdruck führt zu einer Vernachlässigung guter Software-Engineering-Praktiken, was wiederum negativen Einfluss auf die Qualität der Ergebnisse besitzt.

Charette [12] führt an, dass Ursachen für das Scheitern meist auf eine Kombination technischer, organisatorischer und fachlicher Entscheidungen zurückzuführen sind, und nennt beispielhaft folgende Ursachen:

- unrealistische und unausgesprochene Ziele

- ungenaue Schätzungen
- schlecht definierte Systemanforderungen
- unbehandelte Risiken
- schlechte Kommunikation zwischen Kunden, Entwicklern und Nutzern
- Verwendung unausgereifter Technologie
- unzureichende Komplexitätsbeherrschung
- ungenügende Entwicklungspraktiken.

Emam und Koru [35] haben festgestellt, dass Anforderungsänderungen und Änderungen des Projektumfangs sowie der mangelnde Einbezug von Entscheidern die häufigsten Ursachen für das Scheitern von Projekten darstellen.

Verner, Sampson und Cerpa [24] haben 57 Ursachen identifiziert, von denen 28 in Kombination für das Scheitern von Projekten verantwortlich sind:

- In 93 % der untersuchten IT-Projekte beeinflusste der Zeitplan die Entwicklungspraktiken negativ.
- In 81 % der IT-Projekte wurde der Aufwand unterschätzt.
- In 76 % der IT-Projekte wurden Risiken nicht ausreichend behandelt.
- In 73 % der IT-Projekte wurde ein Auslieferungszeitpunkt definiert, ohne ausreichende Informationen über die Anforderungen zu besitzen.

Cerpa und Verner nennen in einer Veröffentlichung die inadäquate Ermittlung von Anforderungen und daraus resultierende Folgen als wesentliche Ursache für das Scheitern von Projekten [36].

Nasir und Sahibuddin [40] haben 26 Ursachen identifiziert, die sie in menschliche, prozessbezogene und technische Ursachen klassifizieren. Die Autoren schlussfolgern, dass technische Ursachen eine untergeordnete Rolle spielen und prozessbezogene Ursachen überwiegen. Das größte Risiko entsteht durch unklare Anforderungen und eine unklare Spezifikation, gefolgt von unklaren Zielen und einem nicht ausreichend definierten Projektumfang. Weitere Ursachen sind eine unrealistische Planung und mangelnde Unterstützung von Entscheidern.

Davis [41] hat festgestellt, dass kein einheitliches Verständnis darüber existiert, welche Ursachen für das Scheitern eines Projekts kritisch sind. Die Autorin hat sieben Stakeholder-Gruppen identifiziert und kritische Ursachen aus deren Perspektive analysiert. Die meisten Gruppen nannten mangelnde Kooperation, Kollaboration, Konsultation und Kommunikation als kritische Ursache.

Es gibt also eine Reihe von Ursachen, die Einfluss auf den Erfolg von Projekten besitzen: Das Commitment von Fachbereichen und Entscheidern auf den Umfang eines IT-Projekts hilft, unklare Anforderungen und Ziele einzugrenzen, und kann die Anzahl von Anforderungsänderungen während der Projektlaufzeit reduzieren. Dieses Commitment muss als Resultat effektiver Kommunikation und Kooperation zwischen Stakeholdern festgelegt werden. Auf Basis ausreichender Informationen muss der Aufwand eines IT-Projekts geschätzt werden, um einen realistischen Projektplan definieren zu können.

### 1.3 Ebenen des Erfolgs von IT-Projekten

Häufig wird der Projekterfolg durch die Einhaltung monetärer und zeitlicher Rahmenbedingungen sowie durch die Umsetzung in angemessener Qualität beurteilt [42]. Die ausschließliche Bewertung von Zeit, Kosten und Qualität reicht jedoch nicht aus, denn auch die Effektivität des Ergebnisses (Produkts) trägt zum Projekterfolg bei [43]–[45]. Die Bewertung des Projekterfolgs muss anhand unterschiedlicher Kriterien erfolgen, die aus unterschiedlichen Perspektiven bewertet werden und den Erfolg auf unterschiedlichen Abstraktionsebenen messen [43], [45]–[52].

Auf der Prozessebene wird der Erfolg anhand der eingesetzten Techniken, Methoden und Werkzeuge aus einer technischen Perspektive bewertet. Während Projektmanager beispielsweise die Zweckmäßigkeit eingesetzter Techniken, Methoden und Werkzeuge beurteilen, legen Entwickler bspw. die Wiederverwendbarkeit der gesammelten Erfahrung zu Grunde. Auf der nächsthöheren Ebene wird der Erfolg des Projektmanagements durch die Erreichung der organisatorischen Projektziele bewertet. Budget, Zeit und Erreichen des geplanten Projektumfangs sind relevante Metriken. Kunden und Nutzer bewerten den Erfolg des Projekts über dessen Ergebnisartefakt, das Produkt. Die Vollständigkeit der Implementierung geforderter Anforderungen und die Akzeptanz, Nutzung und der Wert des Produktes sind relevante Metriken. Aus Organisationsperspektive kann der Beitrag des Produkts zur Steigerung des Geschäftswerts, die Erreichung der gesteckten Ziele und die Kapitalrendite als Metriken zu Grunde gelegt werden. Die höchste Ebene bewertet den Einfluss des Projekts auf die Wettbewerbsposition der Organisation. Das Erreichen strategischer Vorteile kann anhand von Marktreaktionen und externer Wahrnehmung gemessen werden. Tabelle 2 gibt einen Überblick über das zuvor Erläuterte.

Tabelle 2: Ebenen für die Erfolgsbewertung von IT-Projekten (angelehnt an [50])

<i>Ebene</i>	<i>Perspektive</i>	<i>Beschreibung der Ebene</i>	<i>Mögliche Metriken</i>
1. Prozess	Technisch	Genutzte technische und Verwaltungsprozesse, Methoden, Werkzeuge und Techniken, um das Projekt abzuschließen.	<ul style="list-style-type: none"> <li>• Prozesse, die ihren Zweck erfüllt haben</li> <li>• Prozesse, die die Erreichung der Projektziele unterstützt haben</li> <li>• Prozesse, die integriert wurden, wo notwendig</li> <li>• Prozesse, die effektiv implementiert wurden</li> </ul>
2. Projektmanagement	Projekt	Die Projektziele und Gestaltungsparameter	<ul style="list-style-type: none"> <li>• Budget nicht überschritten</li> <li>• Zeitplan eingehalten</li> <li>• Geplanten Projektumfang implementiert</li> </ul>
3. Produkt	Kunden/ Nutzer	Hauptsächliche Ergebnisse des Projekts, z. B. ein neu implementiertes System, eine Aktualisierung einer Softwareversion, etc.	<ul style="list-style-type: none"> <li>• Spezifikation erfüllt</li> <li>• Anforderungen erfüllt</li> <li>• Erwartung erfüllt</li> <li>• Akzeptanz</li> <li>• Nutzung des Systems</li> <li>• Zufriedenheit</li> <li>• Nutzen realisiert</li> </ul>
4. Business	Organisation	Die Geschäftsziele, die der Grund für die Investition waren.	<ul style="list-style-type: none"> <li>• Ziele erfüllt</li> <li>• Business Case bestätigt</li> <li>• Geschäftswert realisiert</li> </ul>
5. Strategie	Industrie	Der resultierende, (un-)beabsichtigte Wettbewerbsvorteil	<ul style="list-style-type: none"> <li>• Geschäftsentwicklung ermöglicht</li> <li>• Wahrnehmung externer Stakeholder/der Konkurrenz</li> <li>• Reaktion des Wettbewerbs erzeugt</li> </ul>

Um den Erfolg eines Projektes zu bewerten, reicht also eine eindimensionale Betrachtung nicht aus. Erfolgskriterien müssen personen- und projektabhängig sowie unter aktiver Einbeziehung der Stakeholder definiert werden. Damit die korrekten Erfolgskriterien identifiziert werden können, müssen relevante Perspektiven auf den Projekterfolg einbezogen werden [44], [49], [53].

Der Interaction Room greift die Notwendigkeit multiperspektivischer Erfolgsbewertung auf. Durch Kooperation und Kommunikation interdisziplinärer Teams werden relevante Stakeholder an der Definition der inhaltlichen Projektziele beteiligt. Es wird eine Informationsbasis geschaffen, auf der die Rahmenbedingungen des Projektes (Zeit, Budget, Projektumfang) definiert werden können. Die Methode ist in plangetriebenen und agilen Vorgehensweisen anwendbar, sodass keine Einschränkung bei der Auswahl von Methoden für die Projektdurchführung entsteht.

## 1.4 Struktur der Arbeit

In Kapitel 2 wird die Problemstellung mit Fokus auf Informationssysteme als relevante Systemklasse erarbeitet. Ein Informationssystem ist ein soziotechnisches System, dessen Entwicklung spezielle Herausforderungen birgt (vgl. Kapitel 2.1). Typische Vorgehensweisen zur Entwicklung von Informationssystemen sind plangetrieben oder agil, besitzen jedoch Grenzen, die in Kapitel 2.2 hergeleitet werden. Das Kapitel 2 schließt mit der Einführung von gemeinsamem Verständnis als Grundvoraussetzung für erfolgreiche Projekte (vgl. Kapitel 2.3).

Im dritten Kapitel werden die Zielsetzung (vgl. Kapitel 3.1) und die Anforderungen (vgl. Kapitel 3.2) an den Lösungsansatz Interaction Room formuliert. Außerdem wird das Forschungsvorgehen zur Entwicklung des Interaction Rooms erläutert (vgl. Kapitel 3.3).

Im vierten Kapitel werden die verwandten Ansätze vorgestellt. Der Schwerpunkt liegt auf kollaborativen und kooperativen Methoden des Requirements Engineerings.

In Kapitel 5 wird der Interaction Room eingeführt und detailliert erläutert. Zunächst wird ein Überblick über Rollen und die Methodenbausteine Vorbereitung, Interaction-Room-Landkarten, Interaction-Room-Annotationen sowie Erkenntnisdokumentation gegeben (vgl. Kapitel 5.1). Im Anschluss werden die Interaction Room-Landkarten im Detail beschrieben. Es wird für jede Landkarte eine Syntax, Semantik und Pragmatik definiert. Abschließend wird die pragmatische Verwendung der Landkarten anhand eines Praxisprojekts veranschaulicht (vgl. Kapitel 5.2). In Kapitel 5.3 werden die Interaction-Room-Annotationen eingeführt und detailliert vorgestellt. Zuerst wird das Konzept erörtert, dann von verwandten Ansätzen abgegrenzt. Es werden die Kategorien der Annotationstypen vorgestellt und einzelne Annotationstypen hergeleitet und beschrieben. Das darauffolgende Unterkapitel widmet sich der Verwendung der Interaction-Room-Annotationen, bevor das bereits bekannte Praxisbeispiel aufgegriffen und um Annotationen erweitert wird. Im abschließenden Kapitel 5.4 wird gezeigt, wie die Erkenntnisse aus dem Interaction Room in eine pragmatische Struktur für Anforderungsdokumente überführt werden.

Im sechsten Kapitel wird die praktische Relevanz des Interaction Rooms evaluiert. Die Methode wurde in 23 Praxisprojekten angewendet. Die Projekte umfassen 48 Workshops mit 71 Workshop-Tagen. In den Workshops wurden 195 Landkarten und 1629 Annotationen verwendet. Mit statistischen Analysen wird der Zusammenhang zwischen Metriken der Projektgröße, zwischen zur Verwendung vorgeschlagenen und tatsächlich verwendeten Annotationstypen und zwischen der Menge verwendeter Annotationen und Duplikaten untersucht.

In Kapitel 7 wird der Einfluss der Annotationen auf das gemeinsame Verständnis unter Schätzenden, die Genauigkeit der Aufwandsschätzung und das Über- bzw. Unterschätzen in einer empirischen Studie untersucht.

In Kapitel 8 wird die Unterstützung durch Interaction-Room-Landkarten bei der Planung von Entwicklungsiterationen und die daraus resultierende Auswirkung auf die Fortschrittmessung anhand quantitativer und qualitativer Ergebnisse untersucht.

Im abschließenden Kapitel 9 wird die Arbeit zusammengefasst, es erfolgt eine kritische Betrachtung, ihre Grenzen werden diskutiert und es wird ein Ausblick auf weitere Forschungsfragen gegeben.

## 2 Problemstellung

Software-Ingenieuren stehen viele Methoden zur Verfügung, um einen Softwareprozess zu gestalten [54]. Bei der Verwendung plangetriebener oder agiler Vorgehensweisen können sich die Beteiligten an einem Fundus von Notationen zur Beschreibung von Realweltausschnitten durch Modelle bedienen (z. B. die Unified Modeling Language (UML) [55], die Business Process Model and Notation (BPMN) [56], ereignisgesteuerte Prozessketten [57] oder die i-\* Modeling Language [58]). Außerdem existieren Werkzeuge zur konstruktiven und organisatorischen Gestaltung eines Softwareprozesses (siehe z. B. [59], [60]). Die umfangreichen Hilfsmittel zur Unterstützung eines Softwareprojekts führen jedoch nicht zwangsläufig zum Projekterfolg (vgl. Kapitel 1).

Die betrachtete Systemklasse in dieser Arbeit sind Informationssysteme. Informationssysteme bestehen aus Menschen und Maschinen, die Daten erzeugen und zu Informationen verarbeiten [61]. Sie sind also soziotechnische Systeme, die nicht vollständig beschrieben werden können, da das Verhalten des Menschen indeterministisch ist [62] und emergente Anforderungen im erkenntnisgetriebenen Softwareprozess entstehen. Anforderungen weisen Beziehungen zueinander auf, die konfliktär sein können [63], woraus Abstimmungsbedarf entsteht, denn Konflikte stellen ein Projektrisiko dar [64]. Durch die Notwendigkeit der Integration mit anderen Softwaresystemen, entstehen weitere Anforderungen, die zu berücksichtigen sind [61]. Zusätzlich zur Komplexität durch viele interdependente Anforderungen ist die Entwicklung von Informationssystemen durch Ungewissheit geprägt, die zur wahrgenommenen Komplexität beiträgt.

### 2.1 Herausforderungen bei der Entwicklung von Informationssystemen

Der Erfolg von Entwicklungsprojekten für Informationssysteme hängt vom Erkenntnisgewinn der beteiligten Individuen ab [65], [66]. Auf Basis eines gemeinsamen Verständnisses über die Fachlichkeit eines Projekts wird eine Lösung entworfen und Technologie ausgewählt, die bestmöglich zur Erfüllung der fachlichen Anforderungen beiträgt. Allerdings fällt die initiale Auseinandersetzung mit der Fachlichkeit häufig einer raschen Debatte über Technologiedetails zum Opfer. Die zu frühe Technologiediskussion erschwert den Wissensträgern der Fachdomäne die



aktive Mitgestaltung des Erkenntnisprozesses und kann dazu führen, dass die Auseinandersetzung mit der Fachlichkeit nicht ausreichend geschieht (vgl. zu diesem Abschnitt [67]).

Viele Lösungen zur notwendigen Kollaboration zwischen Fach- und IT-Bereichen setzen auf Werkzeugunterstützung [54], deren Nutzung erlernt werden muss und deren Funktionsweise die pragmatische Äußerung wesentlicher Informationen einschränken kann (z. B. durch den Zwang zu syntaktisch korrekter Modellierung). Um immanente Ungewissheit, Komplexität und mangelnde Wertorientierung bei der Entwicklung von Informationssystemen geeignet behandeln und die Projektplanung erfolgreich gestalten zu können, müssen kognitive, kommunikative und erfahrungsbasierte Probleme adressiert werden [ebenda].

### 2.1.1 Ungewissheit

Bei der Entwicklung von Informationssystemen ist ein gewisses Maß an Ungewissheit inhärent und unausweichlich [68]. Geringes Wissen über die Anwendungsdomäne in der IT und emergente Stakeholder-Anforderungen führen, gemeinsam mit organisatorischen und umweltbedingten Einflüssen, zu dieser Ungewissheit [69]–[72]. Durch die Natur des zu Grunde liegenden Erkenntnisprozesses, sind neue und veränderte Anforderungen immanent: die Ungewissheit sinkt nicht linear [66], [73]–[75]. Um Erkenntnis zu erzielen, erfordert es Kreativität, Kommunikation und Interaktion zwischen Individuen sowie ein gemeinsames Verständnis über die wesentlichen funktionalen und nicht funktionalen Anforderungen eines Projekts [28], [67]. Im Rahmen eines Softwareprozesses fällt diese Aufgabe in die Phase des Requirements Engineerings (RE), denn im RE werden die funktionalen und nicht funktionalen Anforderungen an ein Informationssystem ermittelt, dokumentiert, geprüft und abgestimmt [76]. Auf Basis der Anforderungen werden Metriken für den Umsetzungsaufwand geschätzt und so der monetäre und zeitliche Rahmen des Projektes festgelegt. Zur realistischen Definition dieser Metriken muss ein ausreichendes Wissen über die wesentlichen Anforderungen bestehen [77]. Ungewissheit ist außerdem ein Treiber für wahrgenommene Komplexität eines Projektes [78]–[80], so dass Aufwand nicht nur unter-, sondern auch überschätzt werden kann (vgl. Kapitel 7), was in beiden Fällen zu inkorrektener Ressourcenallokation führt.

### 2.1.2 Komplexität

Ein Grund für das Scheitern von Entwicklungsprojekten für Informationssysteme ist, dass diese weniger komplex wahrgenommen werden, als sie tatsächlich sind, denn es existieren mehr

Anforderungen, als umgesetzt werden können. Komplexe Softwarekomponenten weisen Abhängigkeiten zueinander auf und die Entwicklung wird von Aspekten beeinflusst, die außerhalb der Kontrolle von Projektteams liegen (z. B. technische oder organisatorische Rahmenbedingungen) [80]–[82].

Komplexe Projekte zeichnen sich durch strukturelle Komplexität, Ungewissheit über die zu erreichenden Ziele, Ungewissheit über die Art der Zielerreichung und Umwelteinflüsse aus [83]. Komplexität steigt mit zunehmender Anzahl Stakeholder, mit der auch die Wahrscheinlichkeit einer höheren Anforderungsanzahl steigt. Änderungen von und Beziehungen zwischen Anforderungen erhöhen die Komplexität ebenfalls [79]. Es ist eine Aufgabe im Requirements Engineering, Komplexität zu reduzieren, denn Komplexität ist ein Kostentreiber [84]. Trotz aktiver Komplexitätsreduktion können Anomalien, Variationen und unerwartete Ereignisse während des Erkenntnisprozesses eintreten, sodass die Komplexität im Verlauf eines Projektes zunehmen kann und aktiv behandelt werden muss [85], [86].

Owen und Linger schlagen für Projekte mit hohem Innovationsanteil vor, Projektmanagement als Ansatz des Wissensmanagements zu verstehen [83]. Williams fordert, Komplexität zu modellieren, und führt die iterative Verfeinerung von Modellen zur Komplexitätsreduktion an [79]. Dabei fordert er, dass nicht nur „harte“, quantitative Daten modelliert werden, sondern auch „weiche Ideen“ [ebenda].

Die optimale Umsetzung eines komplexen Systems ist in vertretbarer Zeit und mit vertretbarem Aufwand nicht realisierbar [87]. Anstatt jeden Aspekt bis ins Detail verstanden zu haben, muss Komplexität handhabbar gemacht werden, was durch Abstraktion, unvollständige Darstellung und durch Unterscheidung von wichtigen und weniger wichtigen Artefakten erfolgen kann [86], [88]. Wegen steigendem Kosten- und Termindruck muss das Kosten-Nutzen-Verhältnis umgesetzter Anforderungen optimiert werden und dabei wertorientiert vorgegangen werden, was Wertorientierung zu einer weiteren Herausforderung für die Entwicklung von Informationssystemen macht [89].

### 2.1.3 Wertorientierung

Die Anzahl der Anforderungen an ein zu entwickelndes Informationssystem sind häufig so hoch, dass die Zeit und/oder das Budget nicht zur Umsetzung aller Anforderungen ausreichen [90]. Um die wichtigsten Anforderungen im zeitlichen und monetären Rahmen umsetzen zu können, muss sichergestellt sein, dass Stakeholder alle relevanten Anforderungen formuliert

haben, diese im Anschluss priorisiert wurden und nach absteigender Priorität umgesetzt werden. Eine Grundlage zur Priorisierung von Anforderungen ist deren Wert (vgl. z. B. [91], [92]). Um den Wertbeitrag einer Anforderung bestimmen zu können, ist Wissen über die Domäne und die verfolgten Geschäftsziele notwendig [65]. Geringe Domänenkenntnisse in der IT können dazu führen, dass Software Engineering wertneutral stattfindet [89]: Jede Anforderung, jeder Anwendungsfall und jeder Testfall wird als gleich wichtig erachtet, Methoden und Praktiken werden ausschließlich als zeitlich-logische Aktivitätsfolgen verstanden und Produktivitäts- sowie Korrektheitsanforderungen werden nicht in Bezug zu ihrem Nutzen bewertet. Dieses wertneutrale Verständnis entspricht jedoch nicht der Realität, denn manche Aktivitäten von Geschäftsprozessen werden häufiger durchgeführt als andere und erfordern daher höhere Qualität. Bestimmte Daten sind zentral für die Wertschöpfung einer Organisation, wohingegen andere unterstützenden Charakter besitzen, sodass die wertschöpfenden Daten ein höheres Maß an Konsistenz und Sicherheit erfordern als unterstützende Daten. Auch Schnittstellen zu benachbarten Systemen unterliegen beispielsweise unterschiedlichen Verfügbarkeits- und Geschwindigkeitsanforderungen.

Um unterschiedliche Kundenanforderungen, Geschäftsanforderungen und technische Notwendigkeiten identifizieren zu können, müssen unterschiedliche Stakeholder in Entscheidungsprozesse einbezogen werden. Daraus entsteht die Chance, möglichst viele relevante Perspektiven in die Entscheidungsfindung einzubeziehen und die wichtigsten Anforderungen zur Erfüllung übergeordneter Ziele zu identifizieren [93], [94]. Dazu müssen Ziele, Strategien, Architekturen und Prozesse zwischen Fachbereichen und der IT abgestimmt werden, um eine einheitliche Ausrichtung von Produkt, Projekt- und Geschäftsentscheidungen sicherstellen zu können [95], [96]. Klassische und agile Vorgehensweisen bieten einen organisatorischen Rahmen zur Abstimmung dieser.

## 2.2 Grenzen plangetriebener und agiler Vorgehensweisen

Unterschiede in der Projektgröße, deren Domäne und rascher technologischer Fortschritt führen zu einer Verteilung relevanten Wissens über Teammitglieder eines Projekts. Während eines Erkenntnisprozesses werden viele unterschiedliche, teils konfliktäre Anforderungen an Informationssysteme identifiziert. Diese Dynamik erfordert zunehmende Geschwindigkeit, Flexibilität, Kommunikation, Kooperation und Abstimmung zwischen Stakeholdern (vgl. zu diesem Absatz [97]).

In plangetriebenen Vorgehensweisen wird zu Beginn eines Projektes eine möglichst umfassende Anforderungserhebung durchgeführt und das gewonnene Wissen in einer Spezifikation dokumentiert. Basierend auf der Spezifikation werden die folgenden Phasen eines Softwareprozesses geplant, weswegen diese Vorgehensweise oft als plangetrieben bezeichnet wird [63], [98], [99]. Plangetriebene Vorgehensweisen definieren sich also über formale Kommunikationsmechanismen wie Dokumente, die an ihre Grenzen geraten, wenn die Umwelt von Dynamik und Ungewissheit geprägt ist [100]. Deswegen werden kontroverse Diskussionen über den notwendigen Detailgrad einer initialen Anforderungserhebung und -dokumentation geführt (vgl. [72], [101], [102]).

Vertreter agiler Vorgehensweisen argumentieren, dass eine vorgelagerte Phase zur Anforderungsanalyse nicht notwendig ist, da Anforderungen erst nahe des Zeitpunkts ihrer Implementierung detailliert verstanden sein müssen. Das Schreiben umfassender Anforderungsdokumente stellt daher eine Verschwendung dar [103]. Außerdem verändern sich Anforderungen durch die Instabilität der System- und Kontextgrenze. Um mit Anforderungsänderungen und Ungewissheit umgehen zu können, ist informale Kommunikation zwischen Stakeholdern notwendig [79], [104]. Agile Vorgehensweisen stellen genau diese informale Kommunikation in den Vordergrund (vgl. z. B. [105]–[107]).

Kritische Betrachtungen agiler Vorgehensweisen bemängeln, dass zu spontan vorgegangen wird und anfangs keine hinreichende Auseinandersetzung mit den wesentlichen Anforderungen stattfindet [72]. Diese Auseinandersetzung ist insbesondere dann sinnvoll, wenn es sich um komplexe Projekte handelt, damit hohe Kosten für die Korrektur von Fehlern und unvorhergesehenen Änderungen vermieden werden können [72], [108]. Um die wesentlichen Anforderungen eines Projektes identifizieren zu können, muss zunächst der Problemraum in voller Breite verstanden werden [77]. Dazu ist eine detaillierte Anforderungsanalyse nicht notwendig, stattdessen muss die Verständnisbildung über die fachlichen und technischen Anforderungen eines Projekts auf einem angemessenen Abstraktionsniveau stattfinden [ebenda]. Durch die Einbeziehung fachlicher, technischer und organisatorischer Stakeholder steigt die Spezialisierung, was zu einer Fokussierung relevanter Aspekte des Problemraums führt [109].

Plangetriebene und agile Vorgehensweisen bieten jedoch kaum Hilfestellung zur inhaltlichen Auseinandersetzung mit der Materie des Problemraums, sondern liefern hauptsächlich einen organisatorischen Rahmen zur Durchführung von Projekten. Damit eine inhaltliche Auseinandersetzung erfolgen kann, ist ein gemeinsames Verständnis unter allen Stakeholdern Grundvoraussetzung. Gemeinsames Verständnis ist in heterogen zusammengesetzten Teams

schwieriger herzustellen als in homogen zusammengesetzten Teams, weil Erfahrung und Wissen Einzelner zu individueller Interpretation des Kontextes führen (vgl. zu diesem Abschnitt [110]–[113]).

### 2.3 Gemeinsames Verständnis als Grundlage des Projekterfolgs

Das Herstellen eines gemeinsamen Verständnisses zwischen fachlichen und technischen Stakeholdern kann durch mehr oder weniger offensichtliche Hürden in der Kommunikation behindert werden. Unterschiede in Erfahrung, Fähigkeiten und Wissen der Teammitglieder erfordern, dass Aspekte, die sich homogen zusammengesetzten Teams aus dem Kontext erschließen, explizit formuliert und erklärt werden müssen.<sup>1</sup> Trotzdem werden häufig, auch in heterogen zusammengesetzten Teams, ausreichend verstandene Aspekte wie Anmeldeverfahren durch Benutzernamen und Passwort oder die Auswahl von Attributen zur Speicherung von Daten natürlicher Personen ausgiebig diskutiert und dokumentiert, statt weniger gut verstandene Aspekte und kritische Erfolgsfaktoren in den Mittelpunkt der Diskussion und Dokumentation zu stellen [67], [114]. Eine weitere Herausforderung für das Herstellen eines gemeinsamen Verständnisses liegt in unterschiedlichen Dokumentationsformen von Anforderungen. Effekte natürlicher Sprache (z. B. Nominalisierung, Universalquantoren und Substantive ohne Bezugsindex), können das gemeinsame Verständnis dokumentierter Anforderungen beeinträchtigen, denn natürliche Sprache ist zwar allgemein verständlich, bietet jedoch Raum zur Interpretation (vgl. z. B. [63]). Die Dokumentation von Anforderungen durch formale Modelle ermöglicht mehr Präzision, erfordert jedoch das Beherrschen der Modellierungssprache, die insbesondere bei nicht technischen Stakeholdern nicht vorausgesetzt werden kann [115]. Daher ist eine Mischform zwischen natürlichsprachlicher und modellbasierter Dokumentation von Anforderungen anzustreben [116]. Durch diese Mischform entstehen sehr umfangreiche und detaillierte Anforderungsspezifikationen, deren Umfang und Tiefe jedoch nicht förderlich für das Erlangen eines initialen Projektüberblicks und für das Erkennen von Zusammenhängen kritischer Anforderungen ist [117].

Um ein gemeinsames Verständnis zwischen Stakeholdern herzustellen, muss das richtige Abstraktionsniveau gewählt und das wirtschaftlich sinnvolle Maß dokumentierter Anforderungen

---

<sup>1</sup> Zum Beispiel besitzt ein deutscher Discounter ein Food- und Non-Food-Sortiment. Zum Food-Sortiment zählen jedoch nicht nur Nahrungsmittel, sondern auch Waschpulver, Shampoo und Zigaretten.

erreicht werden [118]. Glinz und Fricker [104] systematisieren gemeinsames Verständnis in explizites und implizites gemeinsames Verständnis, das sowohl wahr als auch falsch sein kann. In Abhängigkeit des Kontextes kann das gemeinsame Verständnis relevant oder irrelevant und bekannt oder unbekannt sein (vgl. Abbildung 1).

		gemeinsames Verständnis	
		implizit und wahr, aber irrelevant	explizit und wahr, aber irrelevant
relevante, aber bisher nicht bekannte Information	implizit, wahr und relevant	explizit, wahr und relevant	
	implizit und falsch, aber relevant	explizit und falsch, aber relevant	
		implizit, falsch und irrelevant	explizit, falsch und irrelevant

Abbildung 1: Klassifikation von gemeinsamem Verständnis (angelehnt an [104])

Explizites gemeinsames Verständnis basiert auf dokumentierten Artefakten des Softwareprozesses (z. B. Anforderungen, Architekturmodellen, Testfällen). Implizites gemeinsames Verständnis kann durch implizites Wissen, Werte, Annahmen und Meinungen von Stakeholdern sowie durch den Kontext entstehen. Ein gemeinsames Verständnis kann wahr oder falsch sein. Besitzen Stakeholder ein falsches gemeinsames Verständnis, unterliegen sie der Annahme, das Gleiche zu verstehen, tun dies jedoch nicht. Eine dokumentierte Anforderung, die Stakeholder missinterpretierten, ist falsches explizites gemeinsames Verständnis. In einer Diskussion, in der beide Diskussionspartner meinen, dasselbe zu verstehen, dies jedoch nicht tun, besitzen sie ein falsches implizites gemeinsames Verständnis. Maßgeblich für ein wahres gemeinsames Verständnis ist der Kontext. Durch einen Kontext reduziert sich die Notwendigkeit, Wissen zu explizieren, da bereits wahres implizites gemeinsames besteht. Der Kontext wird durch eine Kontextgrenze von der irrelevanten Umgebung getrennt, die sich im Laufe eines Projektes verschieben kann. Durch Reduktion von Ungewissheit können Teile der irrelevanten Umgebung relevant werden und Aspekte des Kontextes nicht mehr relevant für ein Projekt sein [63]. Glinz und Fricker bezeichnen relevante Information, die unbekannt ist, als „dunkle Information“ [104].

Die Entwicklung komplexer Softwaresysteme kann nicht nur auf explizitem gemeinsamem Verständnis beruhen. Denn selbst wenn der Kontext und die Anforderungen stabil sind, ist es unmöglich [62] oder zumindest wirtschaftlich nicht sinnvoll, alle Anforderungen und den

kompletten Kontext zu dokumentieren [104]. Implizites gemeinsames Verständnis ist also von ökonomischer Bedeutung für die Entwicklung von Softwaresystemen, da entfallender Aufwand für die Dokumentation die Kosten eines Projektes senkt. Die Entwicklung komplexer Softwaresysteme kann jedoch auch nicht nur auf implizitem gemeinsamem Verständnis beruhen, weil ein übermäßiger Grad impliziten gemeinsamen Verständnisses ein Projektrisiko darstellt (z. B. durch Erkrankung von Projektmitarbeitern) [ebenda]. Ziel muss also sein, ein angemessenes Maß an explizitem und implizitem gemeinsamen Verständnis durch entsprechende Dokumentationsbreite und -tiefe zu erlangen. Ein angemessener Dokumentationsgrad ist jedoch nicht pauschal definierbar, da dieser von den Rahmenbedingungen des Projektes abhängt (z. B. Größe des Projekts, Art der Software, Fähigkeiten des Projektteams) [ebenda].

## 3 Zielsetzung, Anforderungen und Vorgehen zur Erarbeitung des Lösungsansatzes

Software wird ein immer wichtiger werdender Bestandteil für unseren Alltag und die betriebliche Wertschöpfung. Trotz der zunehmenden Relevanz überschreiten IT-Projekte ihre finanziellen und zeitlichen Rahmenbedingungen. Sie verfehlen ihre Qualitäts- und funktionalen Anforderungen und tragen nicht im gewünschten Maß zum Erreichen von Geschäftszielen bei oder werden abgebrochen. Die Beurteilung des Projekterfolgs geschieht anhand von Kriterien, die von den Stakeholder-Perspektiven und den Bewertungsebenen abhängen. Um möglichst viele Kriterien zur Bewertung des Projekterfolgs abdecken zu können, ist eine multiperspektivische Erarbeitung von Anforderungen an Informationssysteme notwendig.

Die Herausforderungen Ungewissheit, Komplexität und mangelnde Wertorientierung bei der Entwicklung soziotechnischer Informationssysteme werden in plangetriebenen und agilen Vorgehensweisen adressiert. Diese bieten einen organisatorischen Rahmen zur Durchführung von Projekten, liefern jedoch nur geringe Unterstützung bei der inhaltlichen Auseinandersetzung mit der Fachlichkeit eines IT-Projekts. Um Komplexität und Ungewissheit angemessen behandeln und wertorientiert vorgehen zu können, benötigen Stakeholder ein gemeinsames Verständnis über die erfolgskritischen Projekthinhalte. In Abhängigkeit des Projektkontextes muss das gemeinsame Verständnis relevant, wahr und angemessenen implizit bzw. explizit sein.

An dieser Stelle setzt die vorliegende Arbeit an: Der Interaction Room (IR) ist eine Methode, durch die heterogene Teams bei der inhaltlichen Auseinandersetzung mit der Fachlichkeit eines Projektes unterstützt werden.

### 3.1 Zielsetzung des Lösungsansatzes

Mit dem Interaction Room sollen fachliche und technische Stakeholder Anforderungen ermitteln, dokumentieren und abstimmen können. Die Grundlage zur Anforderungserhebung soll ein gemeinsames Verständnis unter den Stakeholdern sein, das durch persönliche Kommunikation, pragmatische Modellierung von Abläufen, Strukturen und Schnittstellen zur Identifikation von Wert-, Aufwands- und Risikotreibern des Projektes beiträgt.



Der Lösungsansatz soll praktische Relevanz besitzen, einen Beitrag zur Planung von IT-Projekten liefern und in plangetriebenen sowie agilen Vorgehensweisen anwendbar sein. Die praktische Relevanz soll für unterschiedliche Branchen und Projekttypen gelten.

## 3.2 Anforderungen an den Lösungsansatz

In den folgenden Unterkapiteln werden Anforderungen an den Interaction Room hergeleitet, indem für die Nützlichkeit pragmatischer Modellierung informaler Modelle und moderierter Kollaboration in heterogenen Teams argumentiert wird. Es werden die relevanten Modellierungsperspektiven für Informationssysteme hergeleitet und es wird für die Nützlichkeit der Kennzeichnung von Wert-, Aufwands- und Risikotreibern in den Modellen argumentiert.

### 3.2.1 Pragmatische Modellierung relevanter Perspektiven für die Analyse und Konzeption von Informationssystemen

Das Modellieren von Aspekten, die Stakeholder im Problemraum einer Domäne beschreiben, führt zu einem expliziten gemeinsamen Verständnis, da mentale Modelle explizit dokumentiert werden [119]. Die Nutzung der vollständigen Syntax einer Modellierungssprache beeinträchtigt die Klarheit eines Modells durch viele Details, wohingegen informale Modelle positiv zum korrekten impliziten gemeinsamen Verständnis beitragen, weil sie aus einer Diskussion heraus entstehen [104], [120]. Informale Modelle werden Modellskizzen genannt, was deren unvollständigen, inkonsistenten und syntaktisch inkorrekten Charakter betont. Modellskizzen besitzen hohe Relevanz für den Softwareprozess: Sie werden zur Kommunikation über und zum Verständnis von Artefakten genutzt, sie werden zur Dokumentation von Diskussionen genutzt, sie werden zur initialen Verständnisbildung eines Projektes genutzt, sie erfordern keine besondere Kompetenz der Modellierenden, sie nutzen nur vereinzelte Syntaxelemente formaler Modellierungssprachen, sie werden präferiert auf analogen Medien erstellt, sie fördern die Akzeptanz aller Stakeholder und senken die Einstiegshürde in einen fachlichen Diskurs, weil keine Werkzeuge und Modellierungsnotationen erlernt werden müssen [115], [121]–[126].

Die daraus resultierende Anforderung an den Interaction Room als Methode lautet, dass syntaktische Modellqualität (Vollständigkeit, Konsistenz, Korrektheit) zugunsten des gemeinsamen Verständnisses vernachlässigt werden darf und Modellskizzen auf analogen Medien erstellt werden sollen.

Damit die Stakeholder ein gemeinsames Verständnis über die modellierte Fachlichkeit erlangen können, müssen Modelle mindestens eine minimale Syntax besitzen und die Stakeholder müssen deren Semantik verstehen [119]. Dass technische Stakeholder mit etablierten Modellierungssprachen und Notationen vertraut sind, ist wahrscheinlicher, als dass fachliche Stakeholder damit vertraut sind [115]. Um den Lernaufwand bei allen Stakeholdern möglichst gering zu halten und den Fokus auf die inhaltliche Auseinandersetzung mit der Fachlichkeit zu legen, sollen die Modellskizzen im Interaction Room eine minimale Syntax besitzen. Die Semantik und die Pragmatik stehen im Vordergrund. Die Güte der Semantik wird durch den Grad der Abdeckung relevanter Aspekte der realen Welt und durch die Angemessenheit der Abbildung bewertet, wobei die Bewertung der Vollständigkeit und Angemessenheit subjektiv durch die Stakeholder erfolgt [127], [128]. Die Güte der Pragmatik drückt die Verständlichkeit eines Modells aus [129]. Sofern Stakeholder bei der Modellierung mitwirken, steigt die semantische und die pragmatische Güte, da die Stakeholder den Kontext der Modellerstellung kennen [128]. Die Diversifikation der Stakeholder-Expertise beeinflusst die semantische Güte positiv [119]. Sofern Modelle in Gruppen erstellt werden, ist das präferierte Medium ein Whiteboard [126]. Deswegen sollen die Modellskizzen im Interaction Room in interdisziplinären Stakeholder-Gruppen auf Whiteboards skizziert werden sollen.

### 3.2.2 Moderierte Kollaboration heterogener Teams

Die heterogene Zusammensetzung von Teams wirkt sich positiv auf die Vollständigkeit eines Projektüberblicks aus, erschwert jedoch das Herstellen eines gemeinsamen Verständnisses im Vergleich zu homogenen Teams [104], [119]. Während kontroverse fachliche Diskussionen ein implizites gemeinsames Verständnis fördern, kann die Asymmetrie von Entscheidungskompetenz und operativem Wissen zu komplementärer Kommunikation führen [28]. Komplementäre Kommunikation entsteht durch unterschiedliche Machtpositionen von Kommunikationspartnern. Im Idealfall ergänzen sich die Kommunikationspartner unterschiedlicher Hierarchieebenen, allerdings besteht die Gefahr von Konformismus niedriger Hierarchieebenen, indem negative Meinungskonvergenz durch sozialen Druck entsteht [119], [130], [131]. Die Kombination von operativen Stakeholdern und Entscheidern ist jedoch ein kritischer Erfolgsfaktor für Projekte [132], sodass die Kooperation und Kollaboration heterogener Teams methodisch durch den Interaction Room unterstützt werden soll.

Um Diskussionen möglichst symmetrisch („auf Augenhöhe“) zu gestalten, komplementäre Gespräche zu steuern und die pragmatische Modellierung zu unterstützen, soll die Durchführung

des Interaction Rooms moderiert werden. Eine Moderation hilft, Diskussionen zu strukturieren, auf sachlicher Ebene zu halten und Redeanteile angemessen zu verteilen [133]. Darüber hinaus kann ein Moderator mit geringen Domänenkenntnissen durch Verständnisfragen dazu beitragen, falsches implizites gemeinsames Verständnis offen zu legen [134]. Durch Moderation bleiben Teammitglieder ausdauernder, zeigen bessere Leistungen und lassen sich auf Lösungsalternativen für ein Problem ein [135]. Des Weiteren achtet ein Moderator auf ein angemessenes Abstraktionsniveau, das allen Stakeholdern ermöglicht, den Gesamtzusammenhang im Problemraum zu verstehen und sich nicht in Detaildiskussionen zu verlieren.

### 3.2.3 Fokussierung auf die wesentlichen Anforderungen

Die pragmatische Modellierung soll aus unterschiedlichen Perspektiven geschehen, die relevante Sichten auf ein Informationssystem darstellen. Durch Informationssysteme wird die Durchführung von Prozessen und die Verarbeitung von Daten unterstützt. Daher soll der Interaction Room je eine Modellierungsperspektive zur Darstellung von Prozessen und Daten beinhalten. Informationssysteme werden im Verbund mit anderen Systemen betrieben, weswegen durch eine Modellierungsperspektive die Schnittstellen zu benachbarten Systemen darstellen werden sollen. Durch die Einordnung des Interaction Rooms in frühe Phasen von Softwareprozessen soll es eine Modellierungsperspektive geben, auf der abstrakte Anforderungen veranschaulicht werden. Das Vorgehen soll „top-down“ zu einem angemessenen Abstraktionsniveau der Modellskizzen führen. Wann ein Abstraktionsniveau als angemessen gilt, soll in der Verantwortung der Stakeholder liegen. Die Modellskizzen sollen die wichtigsten funktionalen Anforderungen an ein Informationssystem dokumentieren und so ein angemessenes explizites gemeinsames Verständnis durch sie erzeugt werden. Da etablierte Modellierungssprachen keine durchgängigen Konzepte vorsehen, um Modellelemente im Hinblick auf Wert, Aufwand und Risiko zu unterscheiden (vgl. Kapitel 5.3.2), soll eine Möglichkeit geschaffen werden, mit der fachliche und technische Stakeholder Elemente der Modellskizzen im Hinblick auf Wert-, Aufwands- und Risikotreiber kennzeichnen können.

## 3.3 Vorgehen zur Erarbeitung des Lösungsansatzes

Der initiale Methodenentwurf des Interaction Rooms wurde 2012 von Book, Grapenthin und Gruhn [67] veröffentlicht. Seit dem dritten Quartal im Jahr 2012 wurde der Interaction Room in 23 Industrieprojekten angewendet. Die Methode wurde nicht immer vollständig eingesetzt,

sondern einzelne Bestandteile aufgrund situativer Notwendigkeit. Die Entwicklung und der situative Einsatz von Methoden finden ihren Ursprung in der Disziplin des Method Engineering. Die Einordnung des Interaction Rooms in das Method Engineering und die Beschreibung des Vorgehens zu ihrer Entwicklung erfolgt in Kapitel 3.3.1. Die Weiterentwicklung der Methode wurde durch einen Action-Research-Prozess [136] gestaltet, der wegen bekannter Schwächen (vgl. z. B. [137], [138]) durch Studien ergänzt wurde. Das Vorgehen der Methodenevolution wird in Kapitel 3.3.2 beschrieben.

### 3.3.1 Methodenentwicklung

Method Engineering wurde 1985 von Bergstra, Jonkers und Obbrink [139] eingeführt und ist auf Mechanical Engineering zurückzuführen. Dessen Ziel ist es, Arbeitsmethoden in Produktionsunternehmen ingenieurmäßig zu konstruieren und weiterzuentwickeln [140], [141]. Kumar und Welke haben die Prinzipien und Konzepte des Mechanical Engineerings auf die Konstruktion und Entwicklung von Informationssystemen übertragen und mit dem Begriff Methodology Engineering eine frühe Version des heutigen Method Engineerings entwickelt [142]. Unter dem Begriff Method Engineering wird die Entwicklung, Anpassung und Evaluation von Methoden für Entwicklungsprojekte von Informationssystemen verstanden [143].

Im Method Engineering liegt der Entwicklung und Anpassung einer Methode ein Metamodell zu Grunde, das Konzepte, Beziehungen und Regeln für die Konstruktion einer Methode spezifiziert [144]. Durch die Instanziierung des Metamodells entsteht eine Methode, die dokumentiert wird. Die dokumentierte Methode beschreibt das Zusammenspiel von Akteuren in bestimmten Rollen, die strukturierte Aktivitäten durchführen, um ein bestimmtes Ziel zu erreichen. Dokumentierte Methoden abstrahieren von unterschiedlichen Projektkontexten, in denen sie angewendet werden können. Um eine Methode in einem spezifischen Projektkontext anzuwenden, kann eine Methode auf die Spezifika des Projektkontextes angepasst werden, woraus eine Methodeninstanz entsteht [145] (vgl. Abbildung 2). Die situationsspezifische Anpassung einer Methode geschieht aufgrund von fachlichen, technischen und organisatorischen Randbedingungen und wird als „Tailoring“ bezeichnet [146].

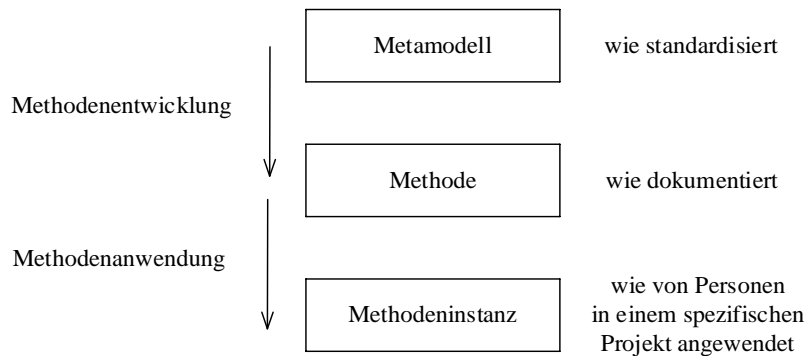


Abbildung 2: Zusammenhang von Metamodell, Methode und Methodeninstanz (angelehnt an [145], [147])

Für den Entwurf der ersten Version des Interaction Rooms wurde sich an der Vorgehensweise von Mayer, Crump, Fernandes, Keen und Painter [148] orientiert (vgl. Abbildung 3).

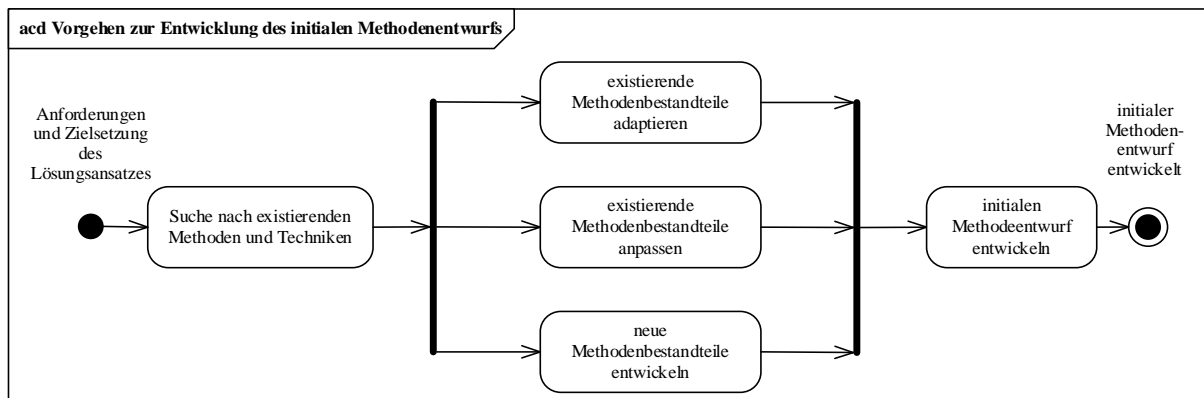


Abbildung 3: Vorgehen zur Erarbeitung des initialen Methodenentwurfs (angelehnt an [148])

Vor dem Hintergrund der Problemstellung, der Zielsetzung und der Anforderungen an den Lösungsansatz wurde nach existierenden Methoden und Techniken zur Analyse und Konzeption von Informationssystemen gesucht. Weil die Methode für den Einsatz in frühen Phasen eines Softwareprozesses geeignet sein soll, wurden primär Techniken aus dem Requirements Engineering verwendet. Daher wurde die Ermittlungstechnik Workshop [63], [133] als Grundlage für die Methode ausgewählt, um die Anforderung „Moderierte Kollaboration heterogener Teams“ (vgl. Kapitel 3.2.2) umzusetzen.

Um der Anforderung „Pragmatische Modellierung relevanter Perspektiven für die Analyse und Konzeption von Informationssystemen“ (vgl. Kapitel 3.2.1) zu genügen, wurden etablierte Notationen zur Dokumentation des Verhaltens und der Struktur von Softwaresystemen ausgewählt und auf deren Basis die Modellierungsperspektiven des Interaction Rooms definiert. Jede Modellierungsperspektive besitzt eine Syntax, jedoch wurde die Menge der Syntaxelemente im

Vergleich zum Ursprung der verwendeten Notationen reduziert und der Zwang syntaktischer Korrektheit wurde zu Gunsten der Pragmatik eliminiert.

Um der Anforderung „Fokussierung der wesentlichen Anforderungen“ (vgl. Kapitel 3.2.3) zu genügen, wurden existierende Ansätze zur Kennzeichnung von Werten, Aufwänden und Risiken in Modellen untersucht (vgl. Kapitel 5.3.2). Da es sich um pragmatische Modelle handelt, viele der untersuchten Ansätze jedoch standardisierte Notationen formal erweitern (wie bspw. das UML Security Profile UMLsec [149]) und nur Teilaspekte abdecken, die für die Kennzeichnung von Wert-, Aufwands- und Risikotreibern notwendig sind, konnten diese nicht für den Interaction Room verwendet werden. Stattdessen wurde mit den Interaction-Room-Annotationen ein neuer Ansatz einwickelt (vgl. Kapitel 5.3), der keine formale Erweiterung einer Notation benötigt und mehr Kriterien zur Kennzeichnung wichtiger Anforderungen abdeckt.

### 3.3.2 Methodenevolution

Der Prozess der Methodenevolution wurde auf der Grundlage von Action Research gestaltet. Action Research ist ein iterativer Prozess, in dem durch Intervention in der Praxis eine Verbesserung oder Lösung eines Problems erreicht werden soll [138]. Während der Intervention werden Daten und Erfahrungen gesammelt, um anschließend daraus zu lernen und die Intervention weiter zu entwickeln [ebenda]. Ein Vorteil von Action Research ist der direkte praktische Bezug, was jedoch den Nachteil mit sich bringt, dass der Forschungsprozess schwer zu kontrollieren ist [150] und die gesammelten Daten verzerrt sein können, denn der Forscher kann das Forschungsobjekt beeinflussen und durch es beeinflusst werden [137]. Des Weiteren sind die gewonnenen Erkenntnisse schwer generalisierbar, weil die gesammelten Daten und die gesammelte Erfahrung kontextabhängig sind [ebenda].

Um die Nachteile von Action Research auszugleichen, jedoch den Vorteil des direkten Praxisbezugs zu behalten, kann Action Research mit Fallstudien und Experimenten kombiniert werden. Mathiassen nennt diese Kombination Collaborative Practice Research [150]: In Experimenten wird der Forschungsprozess kontrolliert und einzelne Variablen werden isoliert, sodass spezifische Fragen präzise beantwortet werden können. Dafür ist der Praxisbezug geringer als bei Action Research. In Studien können unterschiedliche Forschungspraktiken angewendet werden (z. B. Fallstudien, Interviews und Fragebögen). Allerdings ist der Forscher in Studien von der Praxis entkoppelt, denn Ergebnisse entstehen durch Beobachtung und Interpretation [ebenda].

In dieser Arbeit wurde eine Kombination unterschiedlicher Forschungsansätze verfolgt. Action Research bildete die Grundlage der Methodenevolution und wurde gezielt durch Fallstudien und Befragungen mit Hilfe von Interviews und Fragebögen unterstützt. Es wurde eine empirische Studie durchgeführt, um zu untersuchen, ob der Interaction Room einen Beitrag zur Aufwandsschätzung leistet und so zur Planung von IT-Projekten beiträgt (vgl. Kapitel 7). Es wurde eine Fallstudie durchgeführt, um die Unterstützung der Planung von Entwicklungsiterationen durch den Interaction Room zu evaluieren (vgl. Kapitel 8). Die Evolution des Interaction Rooms, von der initialen Version (vgl. [67]) bis zur dokumentierten Methode (vgl. Kapitel 5), wurde durch den Einsatz ihrer Methodenbausteine in 23 Industrieprojekten operationalisiert, die quantitativ untersucht wurden (vgl. Kapitel 6). Die 23 Industrieprojekte wurden nach dem in Abbildung 4 dargestellten Vorgehen durchgeführt.

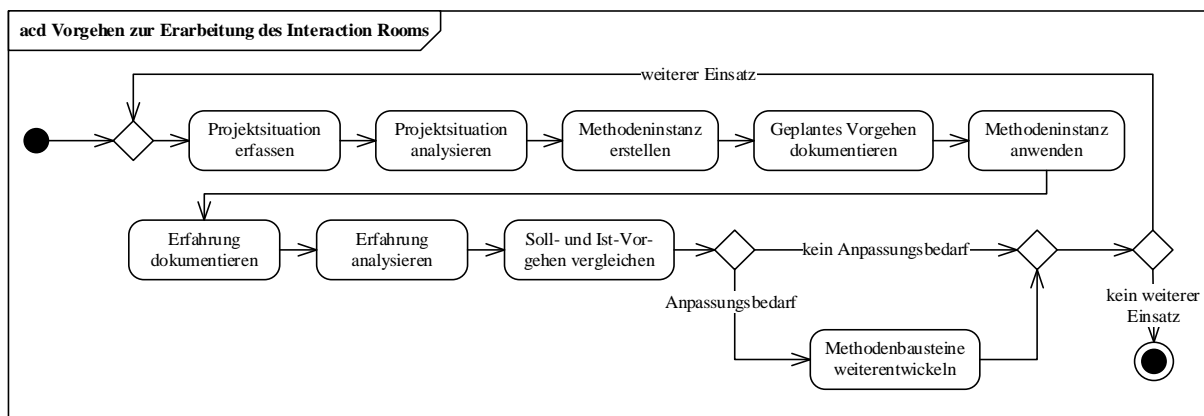


Abbildung 4: Vorgehen zur Erarbeitung des Interaction Rooms

Je Projekt wurde zunächst die Projektsituation erfasst und analysiert. Methodenbausteine wurden entsprechend bestimmter Projektspezifika, wie dem Projekttyp, ausgewählt. Zu den Industrieprojekten zählen unter anderem Migrationsprojekte [151], [152], die Weiterentwicklung eines Legacy-Systems [113] und die Neuentwicklung von Informationssystemen [153]. Das geplante Vorgehen wurde jeweils dokumentiert. Bei der Anwendung gesammelte Erfahrungen wurde ebenfalls dokumentiert. Das geplante Vorgehen wurde mit dem tatsächlichen Vorgehen der Methodeninstanz verglichen. Aus den Erfahrungen und dem Vergleich wurden Handlungsbedarfe abgeleitet, die zur Weiterentwicklung der eingesetzten Methodenbausteine führten.

Ein Beispiel für ein Ergebnis dieses Evolutionsprozesses lässt sich durch den Methodenbaustein Annotationen veranschaulichen. Während bei Book et al. [67] 11 Annotationstypen existierten, wurden in Book, Grapenthin und Gruhn [114] 25 Annotationstypen beschrieben. Von den 11 Annotationstypen aus [67] wurden vier paarweise zusammengefasst, sodass die übrig

gebliebenen neun Annotationstypen um 16 Annotationstypen in [114] erweitert wurden. Von den 25 Annotationstypen aus [114] wurden 15 in dieser Arbeit übernommen, fünf weitere Annotationstypen wurden auf Basis der Erfahrung und Erkenntnisse aus den Praxiseinsätzen des Interaction Rooms entwickelt und hinzugefügt.



## 4 Verwandte Ansätze

Der Interaction Room ist im Softwareprozess in die Phase des Requirements Engineerings einzuordnen. Requirements Engineering ist ein kooperativer, iterativer, inkrementeller Prozess, der in die vier Kernaktivitäten Ermitteln, Dokumentieren, Prüfen und Abstimmen sowie Verwalten von Anforderungen unterteilt wird [63].

In dieser Arbeit werden verschiedene Techniken zur Ermittlung von Anforderungen verwendet, denn die Methode wird als Workshop durchgeführt und mit Interviews sowie Fragebögen vorbereitet. Der Grund für die Auswahl der Technik Workshop als Grundlage des Interaction Rooms liegt in seiner kooperativen Natur, der Möglichkeit, Anforderungen unmittelbar durch Stakeholder abstimmen zu können. Zudem werden Workshops üblicherweise moderiert. Entgegen der Klassifikation aus [63], die Workshops als Kreativitätstechnik verstehen, können Workshops auch als zentrales Element der Anforderungsermittlung eingesetzt werden (vgl. bspw. [133]).

Im Interaction Room werden funktionale und nicht funktionale Anforderungen als Modellskizzen und natürlichsprachlich dokumentiert. Die Konformität zum Dokumentationsformat wird durch die Workshop-Leiter sichergestellt, kann jedoch bedarfsgerecht angepasst werden (vgl. Kapitel 5.2.5). Die Anforderungen werden aus unterschiedlichen Modellierungsperspektiven beschrieben, wobei Beziehungen zwischen Elementen der Perspektiven entstehen und der Zusammenhang mindestens implizit nachvollziehbar bleibt. Durch die Diskussion und Skizzierung von Anforderungen wird ein gemeinsames Verständnis über die Anforderungen erlangt, was als Grundlage zur Abstimmung von Anforderungen genutzt werden kann. Durch den Methodenbaustein Annotationen können strukturiert Konflikte identifiziert werden, die außerhalb des Interaction Rooms durch Techniken des Konfliktmanagements (vgl. z. B. [154]) behandelt werden müssen.

Nach der kurzen Einordnung des Interaction Rooms in die Kernaktivitäten des Requirements Engineerings werden im Folgenden verwandte Methoden vorgestellt, anschließend ein Bezug zum Interaction Room hergestellt sowie eine Abgrenzung vorgenommen. Bei der Suche und Auswahl verwandter Methoden wurde der Fokus auf kooperative Methoden gelegt, die auf Participatory Design zurückzuführen sind.

## 4.1 Participatory Design

Participatory Design (PD) stammt aus Skandinavien und wurde in den 1970er und 1980er Jahren entwickelt. Durch die Einführung von Computern an Arbeitsplätzen, mangelnde Erfahrung in der Konzeption und dem Design von IT-gestützten Arbeitsprozessen sowie der Sorge von Arbeitern, durch Computer ersetzt zu werden, wurden spätere Benutzer von Hard- und Softwaresystemen<sup>2</sup> aktiv in deren Gestaltungsprozess einbezogen [155], [156]. Dadurch sollte das Vorurteil beseitigt werden, Computer werden zur Kontrolle von Arbeitern eingeführt und nicht zur Verbesserung der Arbeitsbedingungen [157], [158]. Da zu diesem Zeitpunkt kein Vorgehen existierte, um Computersysteme kooperativ zu entwerfen, entwickelten Forscher so genannte „language games“, mit dessen Hilfe Softwareentwickler und Arbeiter spielerisch Computersysteme entwerfen und verfeinern konnten. Nach den „language games“ als erste Technik des Participatory Designs wurden weitere Techniken und Werkzeuge entwickelt. Eine Chronologie geben Muller und Kuhn in [159]. Auf Basis dieser Techniken und Werkzeuge wurden verschiedene PD-Methoden entwickelt (vgl. z. B. [160], [161]).

Participatory Design findet in unterschiedlichen Organisationsformen (z. B. Regierungen, Krankenhäusern, Universitäten) und Anwendungskontexten (z. B. Ambient Assistent Living, Software Engineering Psychologie, Soziologie) statt [162]. Dies führt dazu, dass PD in der Literatur unter verschiedenen Synonymen auffindbar ist. Im skandinavischen Raum heißt PD auch Collective Resource Approach [163], Cooperative Design [164], [165] und Cooperative Experimental System Development [166]. In Nordamerika wird PD auch Work-Oriented Design [167], [168] oder Contextual Inquiry [169] genannt. Bedingt durch verschiedene Anwendungskontexte, Hintergründe und beteiligte Disziplinen existiert keine allgemeingültige Definition von Participatory Design [170]. Stattdessen lässt sich subsumieren, dass PD ein kooperativer Lernprozess unter Zuhilfenahme von bestimmten Techniken und Werkzeugen ist. Im Zentrum steht dabei die aktive Mitgestaltung zu entwickelnder Systeme durch die Benutzer. Daher ist PD die Grundlage für im die folgenden diskutierten Methoden und wird auch als Grundlage des Interaction Rooms gesehen.

---

<sup>2</sup> Im Folgenden werden Hard- und Softwaresysteme unter dem Begriff Computersysteme zusammengefasst.

## 4.2 Joint Application Design/Development

Eine erste Version von Joint Application Design/Development (JAD) wurde in den späten 1970er Jahren von Chuck Morris und Tony Crawford von IBM für das Design von computer-gestützten Systemen entwickelt. Es baut auf Business System Planning auf [171]. Ursprünglich wurde mit JAD das Ziel verfolgt, die Interaktion zwischen technischen und nicht technischen Stakeholdern in frühen Phasen eines Softwarelebenszyklus zu unterstützen, indem klassische Besprechungen als Arbeitstreffen (Sessions) organisiert wurden.

Durch die Einführung von JAD konnte die Dauer von Analysephasen in Projekten reduziert werden, weil strukturierte JAD-Sessions effizienter als die Durchführung einzelner Interviews ist [172]. JAD wird in speziell ausgestatteten Räumen durchgeführt, die eine bestimmte Tischanordnung besitzen und mit Flipchart, Whiteboard und Overhead-Projektor ausgestattet sind [173]. Joint Application Design/Development kann sowohl als Technik zur Strukturierung von Arbeitstreffen angesehen werden als auch als Methode, die JAD-Sessions als Kern aller Phasen eines Softwareentwicklungsprozesses versteht [171].

Durch zunehmende praktische Relevanz in den 1980er Jahren wurde JAD auch für andere, nicht technische Zwecke eingesetzt, wie bspw. zur Restrukturierung von Geschäftsprozessen, zur strategischen Planung und zur Restrukturierung von Organisationen [173], [174].

In JAD sind fünf Phasen definiert (vgl. Abbildung 5), die von Wood und Silver [173] ausführlich beschrieben sind und im Folgenden kurz erläutert werden:

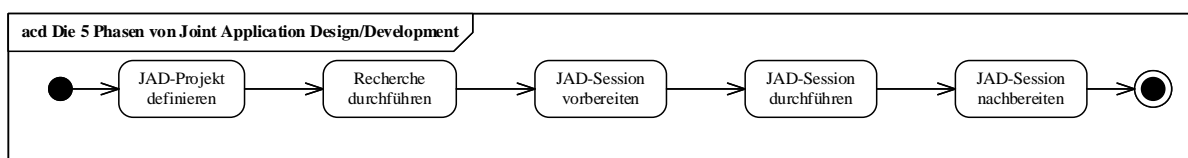


Abbildung 5: Die 5 Phasen von Joint Application Design/Development (angelehnt an [173])

### 1. JAD-Projekt definieren

Auf Basis eines Projektauftrags werden offene Fragen und Annahmen gesammelt und versucht, soweit wie möglich zu beantworten bzw. zu validieren. Ungeachtet der Klärung oder Validierung werden alle Fragen und Annahmen in der JAD-Session diskutiert. Es werden Interviews mit Entscheidern geführt, um den Projektauftrag und die Zielsetzung der Entscheider abzugleichen und den Projektumfang einzuschätzen. Daraus werden benötigte Stakeholder und eine Indikation der Anzahl durchzuführender JAD-Sessions abgeleitet. Das

JAD-Team wird aus allen Stakeholdern zusammengestellt, wobei ein JAD-Team unterschiedliche Rollen besitzt. Es wird zwischen aktiven Teilnehmern, Beobachtern und Sponsoren unterschieden. Ferner werden ein Moderator und eine Person für die Dokumentation ernannt. In einem so genannten „Management Definition Guide“ wird der Zweck, der Umfang und die Ziele des Projekts aus fachlicher Perspektive dokumentiert. Das Dokument dient als Input für die nächste Phase.

## 2. Recherche

Das Ziel dieser Phase ist, eine initiale Analyse des Projekts, anhand eines existierenden Systems und der Fachlichkeit, durchzuführen. Dabei muss der Moderator ein ausreichendes Verständnis erlangen und die wichtigsten Prozesse, Zusammenhänge und Begriffe kennen, um die JAD-Session adäquat moderieren zu können. Darüber hinaus werden erste fachliche Datenmodelle bspw. als Entity-Relationship-Diagramme modelliert. Die Geschäftsprozesse, die durch das Projekt verändert werden, werden identifiziert und modelliert. Außerdem werden Informationen über die Fachanforderungen erhoben und dokumentiert. Abschließend wird eine Agenda für eine JAD-Session erstellt. Die Datenmodelle, Prozessmodelle und die Fachanforderungen bilden die Grundlage für die nachfolgende Phase.

## 3. Vorbereitung der JAD-Session

In dieser Phase wird ein Dokument erstellt, das fortan als Arbeitsgrundlage dient. Darin sind alle bisher ermittelten Informationen aus den Phasen eins und zwei enthalten, die in der JAD-Session detailliert werden. Der Raum für die JAD-Session wird vorbereitet und die dokumentierende Person wird in das Projekt eingeführt. Abschließend erfolgt eine Besprechung, in der das Arbeitsdokument verteilt wird, der Sponsor mit den Zielen der JAD-Session vertraut gemacht und sein Commitment eingeholt wird.

## 4. Durchführung der JAD-Session

In der JAD-Session werden auf Grundlage der Daten- und Prozessmodelle Soll-Anforderungen abgeleitet und modelliert. Außerdem werden getroffene Annahmen aus Phase eins diskutiert und offene Fragen beantwortet. Masken und Dialogflüsse des Systems werden definiert und es wird das Aussehen der Masken durch Prototyping festgelegt. Die Struktur und der Inhalt von Reports werden festgelegt. Der Moderator leitet die Erstellung der zuvor genannten Artefakte und die dokumentierende Person notiert Informationen, sofern der Moderator ihn dazu auffordert.

## 5. Nachbereitung der JAD-Session

In der Nachbereitung der JAD-Session wird das finale Dokument erstellt, indem das Arbeitsdokument und die Notizen aus der JAD-Session aufbereitet und dokumentiert werden. Im Anschluss wird das Dokument unter den Teilnehmern der JAD-Session verteilt und abgestimmt.

Der Interaction Room baut auf JAD auf, denn auch der Interaction Room ist in Phasen unterteilt. Die Durchführung findet ebenfalls in einem speziell ausgestatteten Raum statt. Ein IR-Workshop wird von zwei Interaction-Room-Coaches geleitet und erfordert die Anwesenheit eines Projektverantwortlichen und fachlichen sowie technischen Stakeholders. In der Vorbereitung eines IR werden auch existierende Dokumente und Systeme analysiert, jedoch werden vorab keine Modelle erstellt. Dies erfolgt erst während des Workshops im interdisziplinären Team. Die Modelle des Interaction Rooms bilden den Ist-Zustand oder den Soll-Zustand ab und werden durch Annotationen mit Zusatzinformationen angereichert, die ebenfalls den Ist- oder Soll-Zustand darstellen können. Wie in JAD werden Daten- und Prozessmodelle sowie grobe fachliche Anforderungen im Interaction Room skizziert. Zusätzlich existiert eine Perspektive zur Visualisierung des relevanten Kontextes der Anwendungslandschaft eines Informationssystems. Masken und Berichte sind nicht Bestandteile des IRs. Wie Joint Application Design/Development kann auch der Interaction Room in unterschiedlichen Projekttypen eingesetzt werden. Anders als bei JAD ist der Übergang zwischen einzelnen Phasen weniger formal. Während in JAD ein Dokument fortgeschrieben wird, wird in der Nachbereitung eines Interaction-Room-Workshops ein Dokument angefertigt. Werden in einem Interaction-Room-Projekt mehrere Interaction-Room-Workshops durchgeführt, wird das Anforderungsdokument ebenfalls fortgeschrieben.

### 4.3 Soft Systems Methodology

Die Soft Systems Methodology (SSM) wurde an der Universität Lancaster unter der Leitung von Peter Checkland entwickelt [175]. SSM wurde durch Action Research erarbeitet und verwendet Action Research als Grundlage der Methode. SSM wurde zur Lösung problematischer Situationen<sup>3</sup> entworfen, wobei zwei problematische Situationen nie exakt gleich sind. Daher existieren keine Lösungen für problematische Situationen, sondern diese entstehen in einem kontinuierlichen Lernprozess [176], [177]. Checkland versteht problematische Situationen als

---

<sup>3</sup> Checkland spricht von „wicked problems“ und explizit nicht von Problemen, weil nicht immer ein Problem zu Grunde liegen muss, um SSM anzuwenden[177].

unstrukturierte Entscheidungsprozesse, zu deren Durchführung keine Erfahrung in einer Organisation existiert und für die keine bekannten und expliziten Antworten in einer Organisation vorhanden sind [178]. Stattdessen sind sie durch die Variablen Meinung, Kultur, Politik und Motivation von Stakeholdern geprägt [179].

Das Ziel von SSM ist es, durch eine ganzheitliche Analyse einer problematischen Situation emergente Eigenschaften zu identifizieren, die nicht isoliert von Menschen und seinem Kontext existenzberechtigt sind [176]. So sollen wünschenswerte und kulturell angemessene Verbesserungen erarbeitet werden [177]. Der Begriff System umfasst also nicht nur Softwaresysteme, sondern subsumiert unter dem Begriff soziotechnische Systeme.

SSM geht von zwei Grundannahmen aus:

1. Jede Situation muss aus einer individuellen Perspektive, die eine bestimmte Weltanschauung repräsentiert, betrachtet und analysiert werden.
2. Die Akteure in der problematischen Situation streben danach, zweckmäßige und zielgerichtete Aktivitäten durchzuführen.

Die Soft Systems Methodology wird personen- und situationsspezifisch ausgestaltet. Daher wird SSM in unterschiedlichen Kontexten eingesetzt, wovon einer die Entwicklung von Informationssystemen ist [177].

Das abstrakte Vorgehen von SSM ist in Abbildung 6 dargestellt und zielt darauf ab, einen Transformationsprozess zu strukturieren. Es besteht aus vier Schritten, die im Folgenden erläutert werden.

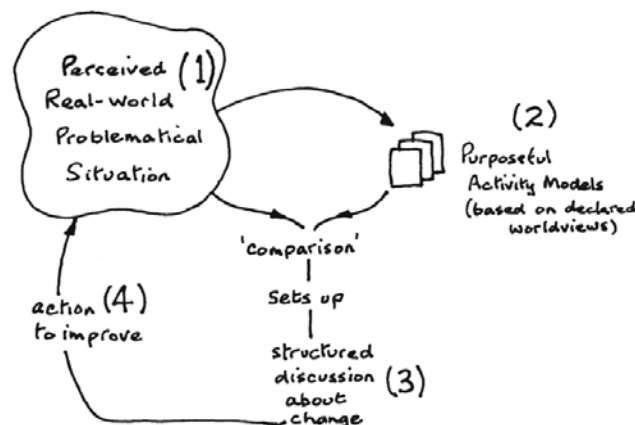


Abbildung 6: Abstrakter Prozess der Soft Systems Methodology [177]

Im ersten Schritt soll die problematische Situation verstanden werden, indem Gespräche und Interviews geführt, an Besprechungen teilgenommen und Dokumente gelesen werden. Daraufhin wird der relevante Realweltausschnitt durch so genannte „Rich Pictures“ beschrieben, um den Wirkungszusammenhang der Bestandteile des Realweltausschnitts darzustellen. Das Ziel ist es, die wesentlichen Entitäten, Strukturen und Perspektiven der Situation, die inhärenten Prozesse und bereits erkannte sowie potentielle Probleme zu verstehen. Rich Pictures besitzen keine Notation, stattdessen schlägt Checkland vor, Bilder zu malen [180], da diese sich besonders zur Beschreibung problematischer Situationen eignen [177].

Im zweiten Schritt werden aus Rich Pictures unterschiedliche Perspektiven abgeleitet. Eine Perspektive repräsentiert ein fiktives System, in dem Menschen zweckmäßige Aktivitäten durchführen [175]. Diese fiktiven Systeme werden auch als Human Activity Systems (HAS) bezeichnet [179]. Ein HAS wird durch die Beantwortung der Fragen des Mnemonic CATWOE<sup>4</sup> analysiert. Die Beantwortung führt zu so genannten „Root Definitions“. Für jedes HAS werden die folgenden Fragen beantwortet:

- Welche Kunden profitieren von der Transformation der problematischen Situation?
- Welche Akteure sind an der Transformation beteiligt?
- Welche Transformation soll durchgeführt werden?
- Wie passt die Transformation zur Weltanschauung des HAS und welche Implikationen ergeben sich daraus?
- Welche Verantwortlichen können die Transformation fördern oder verhindern?
- Welche Umweltfaktoren beeinflussen die Transformation?

Ausgehend von den Root Definitions werden Prozesse mit  $7 \pm 2$  Aktivitäten erstellt, deren Ausführung zur Verbesserung einer Situation führen soll. Für jedes Modell werden Kennzahlen definiert, um die Effektivität der Verbesserung messen und ggf. weitere Maßnahmen zur Verbesserung einleiten zu können.

Im dritten Schritt werden die Modelle mit dem Realweltausschnitt verglichen, um die Wirksamkeit der Modelle einschätzen zu können. Dazu werden Diskussionen, Befragungen oder kollaborative Modellierungstechniken angewendet, sodass eine Struktur entsteht, die darauf abzielt, argumentativ wünschenswerte und kulturell passende Verbesserungen zu identifizieren. Diese genügen den HAS zu Grunde liegenden Weltanschauungen.

---

<sup>4</sup> CATWOE steht für Customer, Actor, Transformation, Weltanschauung [sic], Owner und Environment [177].

Im vierten Schritt werden die Aktivitäten, die eine Verbesserung erzielen sollen, angewendet und deren Wirksamkeit überprüft, woraufhin ein neuer Zyklus des Lernprozesses beginnen kann [177].

Die Soft Systems Methodology ist im Gegensatz zum Interaction Room nicht nur auf die Konzeption von Informationssystemen beschränkt, sondern die Konzeption von Informationssystemen stellt nur einen Anwendungsfall für SSM dar. In SSM werden keine vordefinierten Modelle verwendet, wohingegen der Interaction Room vier definierte Modellierungsperspektiven besitzt. Der Interaction Room greift die Grundannahme von SSM auf und berücksichtigt unterschiedliche Perspektiven auf eine Problemstellung, denn an Interaction-Room-Workshops partizipieren interdisziplinär zusammengesetzte Teilnehmer. Auch eine initiale Situationsanalyse wird im Rahmen der Workshop-Vorbereitung für einen Interaction Room durchgeführt, die jedoch im Umfang auf die konkrete Fragestellung des Projekts beschränkt ist.

#### 4.4 Participatory IT-Design

Participatory IT-Design ist eine Methode für das Design von IT-Projekten und ein Resultat des dänischen Forschungsprogramms MUST<sup>5</sup>. Daher ist Participatory IT-Design auch als MUST-Methode bekannt [160]. Die von Kensing, Simonsen und Bødker [181] beschriebene Methode basiert auf der Erfahrung von 13 durchgeführten IT-Projekten und kann in unterschiedlichen Projekttypen eingesetzt werden. Vier Prinzipien werden von den Autoren als Essenz der Methode bezeichnet:

1. Das Prinzip „Coherent Vision for Change“ soll dazu führen, dass ein ausgewogenes Verhältnis zwischen dem Aufwand für die Entwicklung von Systemen, der Organisationsentwicklung und der Entwicklung der Qualifikation von Personen entsteht, um nachhaltigen IT-Einsatz entsprechend der aktuellen Geschäftsziele zu ermöglichen.
2. Durch das Prinzip „Genuine User Participation“ wird betont, dass spätere Nutzer aktiv in die Gestaltung einer IT-Lösung einbezogen werden müssen. Dabei wird davon ausgegangen, dass gegenseitiges Lernen zu besserem Verständnis zwischen Designern und Benutzern führt. Außerdem werden politische Aspekte wie die Mitbestimmungen der Arbeitsumgebung durch das Befolgen des Prinzips adressiert (vgl. auch Kapitel 4.1).

---

<sup>5</sup> MUST ist ein dänisches Akronym für Theorien und Methoden der initialen Analyse und Designaktivitäten in IT-Projekten.



3. Das Prinzip „First Hand Experiences with Work Practice“ zielt darauf ab, implizites Wissen explizit zu machen. Durch Ausprobieren werden implizite Anforderungen offensichtlich und das Problemverständnis steigt.
4. Das Prinzip „Anchoring Visions“ fordert, mittelbar beteiligte oder betroffene Zielgruppen über die Vision, Ziele und Pläne des Designprojekts zu informieren. Insbesondere Änderungen müssen mit Entscheidungsgremien, betroffenen Mitarbeitern und Personen, die die Ergebnisse des Designprojekts umsetzen, abgestimmt werden.

Participatory IT-Design ist eine Methode, die keinem festen Prozess folgt (vgl. Abbildung 7), sondern es sind vier Phasen als Zeiträume zwischen Entscheidungen definiert. In den Phasen werden Aktivitäten durchgeführt, um Informationen zu gewinnen, die in ein oder zwei Berichte je Phase dokumentiert werden. Das Ergebnis jeder Phase bildet die Informationsbasis für die Entscheidungsfindung im Rahmen der Transition zwischen zwei Phasen und dient als Grundlage für die Aktivitäten der folgenden Phase (vgl. 1–8 in Abbildung 7).

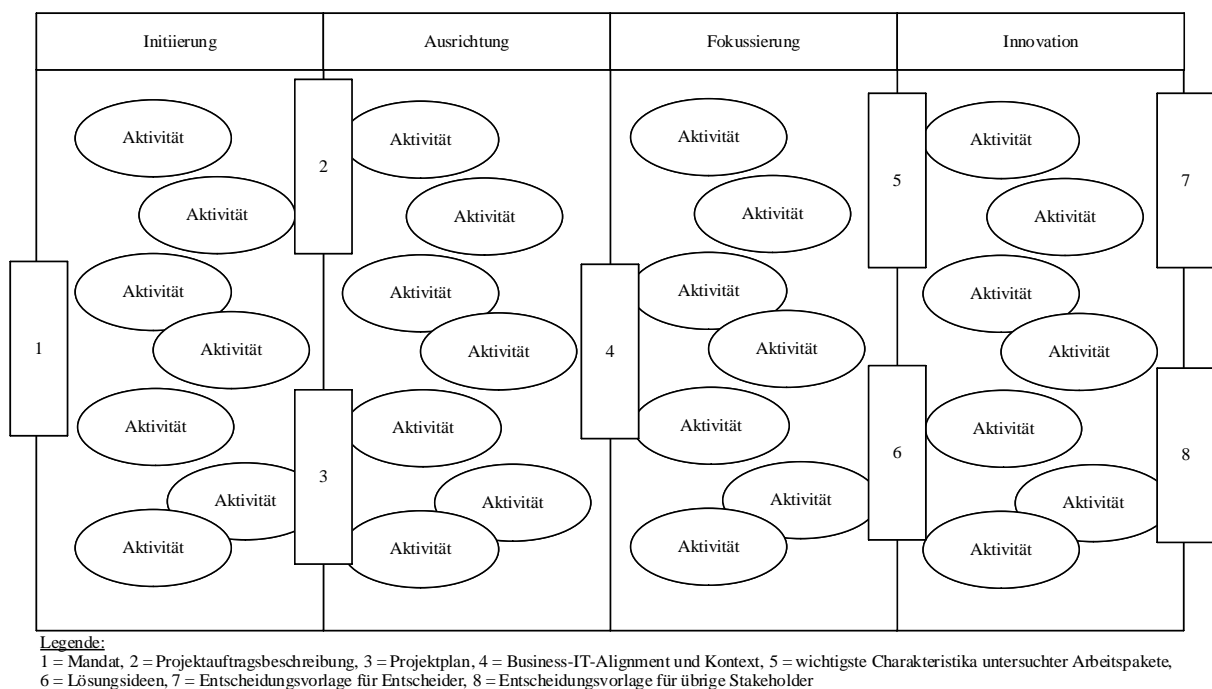


Abbildung 7: Participatory-IT-Design-Phasen (angelehnt an [181])

Die Initiierungsphase basiert auf dem Mandat, ein Projekt durchzuführen (1 in Abbildung 7). Durch die Aktivitäten dieser Phase wird eine Projektauftragsbeschreibung (2 in Abbildung 7) und ein Projektplan (3 in Abbildung 7) erstellt. Auf deren Grundlage wird eine Entscheidung über die Rahmenbedingungen des Projekts und dessen Projektumfang getroffen. Die Aktivitäten der Ausrichtungsphase zielen darauf ab, die Stimmigkeit der Projektinhalte mit den übergeordneten Geschäfts- und IT-Zielen sicherzustellen. Das Resultat ist eine Schärfung der

strategischen Ausrichtung des Projekts und die Kenntnis der einzubeziehenden Unternehmensbereiche (4 in Abbildung 7). In der Phase Fokussierung werden Probleme und Bedürfnisse priorisiert. Die Ergebnisse der Phase Fokussierung sind die wichtigsten Charakteristika untersuchter Arbeitspraktiken, des Weiteren priorisierte Probleme, Bedürfnisse (5 in Abbildung 7) und erste Lösungsideen (6 in Abbildung 7). In der abschließenden Innovationsphase werden Lösungen ausgearbeitet und validiert, um die identifizierten Probleme zu lösen und die Veränderung der Organisation sowie der IT zu gestalten. Die Ergebnisse werden in zwei Berichten festgehalten, die es den Entscheidern (7 in Abbildung 7) und den übrigen Stakeholdern (8 in Abbildung 7) ermöglichen sollen, die Lösungsvorschläge zu beurteilen und über sie zu entscheiden. Der Interaction Room kann ebenfalls in unterschiedlichen Projekttypen verwendet werden und folgt dem Prinzip „Genuine User Participation“ durch die partizipative Arbeit heterogener Teams. Während in der MUST-Methode die Schnittstelle zwischen Phasen durch Dokumentenaustausch realisiert wird, wird der Übergang zwischen den Phasen des Interaction Rooms weniger formal gestaltet. Als Resultat eines Interaction-Room-Workshops wird ein Dokument verfasst, in dem die Erkenntnisse dokumentiert werden (vgl. Kapitel 5.4).

## 4.5 Design Thinking

Design Thinking (DT) wurde von David Kelley, Larry Leifer und Terry Winograd entwickelt [182] und ist auf Participatory Design (vgl. Kapitel 4.1) zurückzuführen [183]. David Kelly ist heute Leiter des Hasso Plattner Institute of Design (sog. d.school) an der Universität Stanford. Seit 2007 wird Design Thinking auch am Hasso Plattner Institut of Design Thinking in Potsdam gelehrt [184]. Design Thinking wird als etablierte Methode für die Entwicklung innovativer Produkte gesehen, deren übergeordnetes Ziel die Befriedigung von Nutzerbedürfnissen ist [185]. Mittlerweile wird DT von Konzernen wie der Swisscom AG, der Deutschen Bank AG und der SAP AG als Innovationsmethode verwendet [186], [187].

Der Begriff Design steht nicht, wie im deutschen Sprachgebrauch üblich, für eine kreative, formgebende Aktivität, sondern umfasst auch die konzeptionellen und technischen Aspekte zur Gestaltung von Objekten und Systemen [186]. Bestandteile von DT sind Techniken, die situativ kombiniert werden, um einen passenden Lösungsweg zu gestalten. Eine feste Reihenfolge bzw. die Notwendigkeit, bestimmte Techniken einzusetzen, besteht nicht [187]. Stattdessen liegt DT ein Prozess zu Grunde, der sich in die Problemanalyse und die Lösungsfindung unterteilen lässt, für die insgesamt sechs Prozessphasen definiert wurden (vgl. Abbildung 8). Die iterative

Verschränkung der Prozessphasen ermöglicht, aus Phasen in vorherige Phasen zurückzukehren oder nachfolgende zu überspringen.

In den jeweiligen Phasen arbeitet ein so genanntes Design-Thinking-Team, das interdisziplinär besetzt ist [188], [189]. Um den Kreativprozess zu unterstützen, findet Design Thinking in variablen Räumlichkeiten statt, bei deren Gestaltung auf die wesentlichen Aspekte der Mobilität, Flexibilität und Einfachheit Wert gelegt wurde [187].

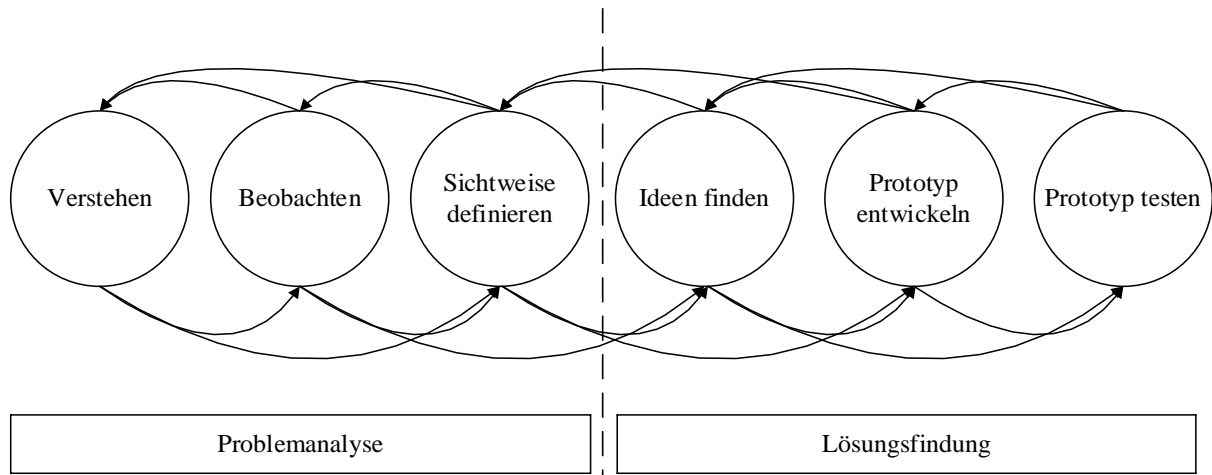


Abbildung 8: Prozessphasen des Design Thinking (angelehnt an [187])

Die erste Prozessphase „Verstehen“ ist der Ausgangspunkt einer jeden Iteration, indem die zu bearbeitende Fragestellung (sog. Design Challenge) im Rahmen einer Problemanalyse formuliert wird [187]. Zur Formulierung der Fragestellung werden Zielgruppen und die Objekte der Fragestellung (Strategien, Produkte, Dienstleistungen, Geschäftsmodelle) identifiziert, um daraufhin den gewünschten Endzustand des Objektes unter Berücksichtigung bestimmter Rahmenbedingungen zu definieren [186]. Bei der Problemanalyse werden immer die Bedürfnisse von Personen der Zielgruppen in den Vordergrund gestellt [187]. Am Ende des Schritts soll der Arbeitsauftrag von allen Teilnehmern verstanden und der Weg der Umsetzung definiert sein [ebenda].

In der zweiten Phase „Beobachten“ werden, mit Hilfe klassischer Techniken zur Anforderungsermittlung (vgl. z. B. [190]), die Bedürfnisse der Zielgruppen in Bezug zum Problem erarbeitet [186]. Durch die Beobachtung und Befragung von potentiellen Nutzern der Zielgruppe werden Bedürfnisse, Wünsche, Erwartungen und Verhaltensweisen der potentiellen Benutzer analysiert, um so viele Daten wie möglich für die Verarbeitung im nächsten Schritt zu sammeln [187].

In der dritten Phase „Sichtweise definieren“ werden die „wichtigen Informationen und notwendigen Bedürfnisse der potentiellen Nutzer herausgefiltert“ [187]. Ziel ist es, die Perspektiven von Repräsentanten der Zielgruppen zu definieren, um darauf basierend Lösungsideen zu entwickeln. Das Ergebnis dieser Phase sind Persona [191], die so genannte Point-of-Views widerspiegeln.

Die Phase „Ideen finden“ ist der erste Schritt der „Lösungsfindung“. Ideen werden durch Kreativitätstechniken (vgl. z. B. [190]) generiert. Um weiterzuverfolgende Ideen zu selektieren, werden die Ideen gruppiert und eine Entscheidung aufgrund eines Gefühls oder einer Erfahrung gefällt [187]. Sofern sich kein einheitliches Gefühl einstellt oder die Erfahrungen divergieren, führt ein Mehrheitsbeschluss zur Entscheidung [192]. Die ausgewählte Idee wird im nächsten Schritt als Prototyp umgesetzt.

Nachdem die theoretische Auseinandersetzung mit der Lösungsidee erfolgt ist, wird die Idee in der fünften Prozessphase prototypisch umgesetzt. Ein Prototyp muss keine Software sein, sondern kann auch aus physischem Material wie Knete oder LEGO® erstellt werden bzw. ein Rollenspiel sein [187]. Durch den Prototyp soll die Lösungsidee greifbar, Stärken und Schwächen erkannt und daraus Erkenntnisse für die weitere Entwicklung gewonnen werden können [ebenda]. Nachdem ein Prototyp erstellt wurde, wird dieser mit Probanden der Zielgruppe getestet.

Um die Stärken und Schwächen der prototypisch umgesetzten Idee identifizieren zu können, wird diese in der sechsten Prozessphase mit potentiellen Nutzern getestet. Ein Prototyp durchläuft die Prozessphasen fünf und sechs mehrfach, sodass Feedback eingearbeitet werden und ein erneuter Test stattfinden kann.

Der Interaction Room ist eine Methode, bei der interdisziplinäre Teams ebenfalls in einem speziellen Raum arbeiten, jedoch nicht mit dem ausschließlichen Ziel, Innovation zu schaffen. Stattdessen können mit dem Interaction Room auch fachliche Analysen [151] oder Iterationsplanungen in Softwareentwicklungsprozessen [113] unterstützt werden. Wie beim Design Thinking wird der Interaction Room durch ein spezielles Team geleitet. Der Interaction Room besteht aus Methodenbausteinen, die situativ selektiert, angepasst und eingesetzt werden können.

## 4.6 Accelerated Solution Environment

Accelerated Solution Environment (ASE) wurde von dem global tätigen Beratungsunternehmen Capgemini Consulting entwickelt und zielt darauf ab, Veränderungsprozesse einzuleiten, indem relevante Stakeholder kollaborativ eine Lösung erarbeiten [193]. Dabei spricht Capgemini davon, dass durch diesen kollaborativen Start einer Initiative „ideale Voraussetzungen für die nachfolgende Umsetzung geschaffen (werden), da alle notwendigen Parteien bereits in die Konzeption und Planung involviert sind“ [ebenda]. Die Literatur zu ASE beschränkt sich auf Quellen, die keinen Peer-Review oder ein Lektorat durchlaufen haben, sondern ausschließlich auf Informationen der Firma Capgemini zurückzuführen sind. Trotzdem wird ASE wegen seiner praktischen Relevanz im Folgenden kurz vorgestellt.

Kern von ASE ist ein moderierter Workshop, der eineinhalb bis drei Tage dauert. Die Vorbereitung des Workshops dauert vier bis sechs Wochen. ASE beinhaltet die drei Phasen Scan, Focus, Act. In der Phase Scan wird der aktuelle Kenntnisstand über das Thema erarbeitet und in der Phase Focus durch parallele und iterative Arbeitsmodule weiterentwickelt. Ziel der Moderatoren ist es, die Stakeholder in den Phasen Scan und Focus zu befähigen, dass sie eigenverantwortlich Ergebnisse umsetzen (Phase Act). Die Phase Act ist also nicht mehr Bestandteil des Workshops.

Das Accelerated Solution Environment wird von einem speziellen ASE-Team durchgeführt, das aus einem ASE-Moderator und Wissensarbeitern besteht. Die Stakeholder eines ASE-Workshops sollen interdisziplinär zusammengesetzt sein.

Der Interaction Room und ASE nutzen beide interdisziplinäre Teams und spezielle Räumlichkeiten, um durchgeführt zu werden. Mit Accelerated Solution Environment werden die Workshops in der Focus-Phase in einer dezidierten Vorbereitungsphase (Scan) vorbereitet. Auch Interaction-Room-Workshops werden speziell vorbereitet. Die Dauer der Interaction-Room- und ASE-Workshops ist vergleichbar und auch der Interaction Room besitzt eine Nachbereitungsphase, in der die Workshop-Ergebnisse analysiert und weiterverarbeitet werden.

## 5 Der Interaction Room

Der Interaction Room ist eine Workshop-Methode, die in einem physisch begehbaren Raum mit großen Whiteboards an den Wänden durchgeführt wird. Auf den Whiteboards werden Modellskizzen abgebildet, die bestimmte Sichten auf ein Informationssystem repräsentieren. Eine Modellskizze heißt Interaction-Room-Landkarte (Landkarte). Der Begriff Landkarte symbolisiert das angestrebte Abstraktionsniveau, denn die Landkarten zeigen relevante fachliche und technische Aspekte eines Informationssystems, bleiben dabei jedoch abstrakt. So genannte Interaction-Room-Coaches (IR-Coaches) moderieren die kooperative Erstellung der Landkarten durch ein heterogen zusammengesetztes Projektteam. Das Ziel der kooperativen Erstellung ist es, einen Überblick über die funktionalen Anforderungen eines Informationssystems herzustellen. Auf den Landkarten werden Wert-, Aufwands- und Risikotreiber identifiziert und mit Interaction-Room-Annotationen (Annotationen) gekennzeichnet. Die Annotationen werden diskutiert und dokumentiert. Die Identifikation und Kennzeichnung von Werttreibern, Aufwands treibern und Risikotreibern durch Annotationen zielt ebenfalls nicht auf Vollständigkeit ab, sondern dient der expliziten Diskussion und Dokumentation von Anforderungen, die ohne Annotationen implizit bleiben würden.

In Kapitel 5.1 wird ein Überblick über den Interaction Room gegeben. Es wird das zu Grunde liegende Metamodell, die Rollen und die Methodenbausteine eingeführt. In Kapitel 5.2 werden die Syntax, die Semantik und die Pragmatik der Interaction-Room-Landkarten definiert. Abschließend werden die Landkarten durch ein Praxisbeispiel veranschaulicht. In Kapitel 5.3 wird das Konzept der Interaction-Room-Annotationen eingeführt, verwandte Ansätze werden diskutiert, Annotationskategorien werden vorgestellt und Annotationstypen hergeleitet sowie beschrieben. Daraufhin wird die Verwendung der Annotationen erläutert. Das Kapitel schließt mit dem bereits bekannten Praxisbeispiel und dessen Ergänzung um Annotationen. Das abschließende Kapitel 5.4 zeigt die Anschlussfähigkeit der Erkenntnisse aus Interaction-Room-Workshops an einen Softwareprozess, indem die Erkenntnisse in ein Anforderungsdokument überführt werden.

## 5.1 Überblick

Dem Interaction Room liegt das in Abbildung 9 dargestellte Metamodell zu Grunde. Es zeigt atomare Methodenbausteine und zusammengesetzte Methodenbausteine. Jeder Methodenbaustein kann einen Vorgänger und beliebig viele Nachfolger besitzen. Ein Methodenbaustein besteht immer mindestens aus einem Prozesselement und einem Produktelement. Ein Prozesselement kann einen Vorgänger und beliebig viele Nachfolger besitzen. Ein Prozesselement erzeugt als Ergebnis ein Produktelement oder benötigt mindestens ein Produktelement als Input. Ein Produktelement kann atomar oder zusammengesetzt sein. Ein zusammengesetztes Produktelement besitzt ein übergeordnetes Produktelement. Produktelemente können durch beliebig viele Produktelemente verfeinert und von beliebig vielen Prozesselementen produziert werden. Ein Produktelement wird von mindestens einem Prozesselement benötigt.

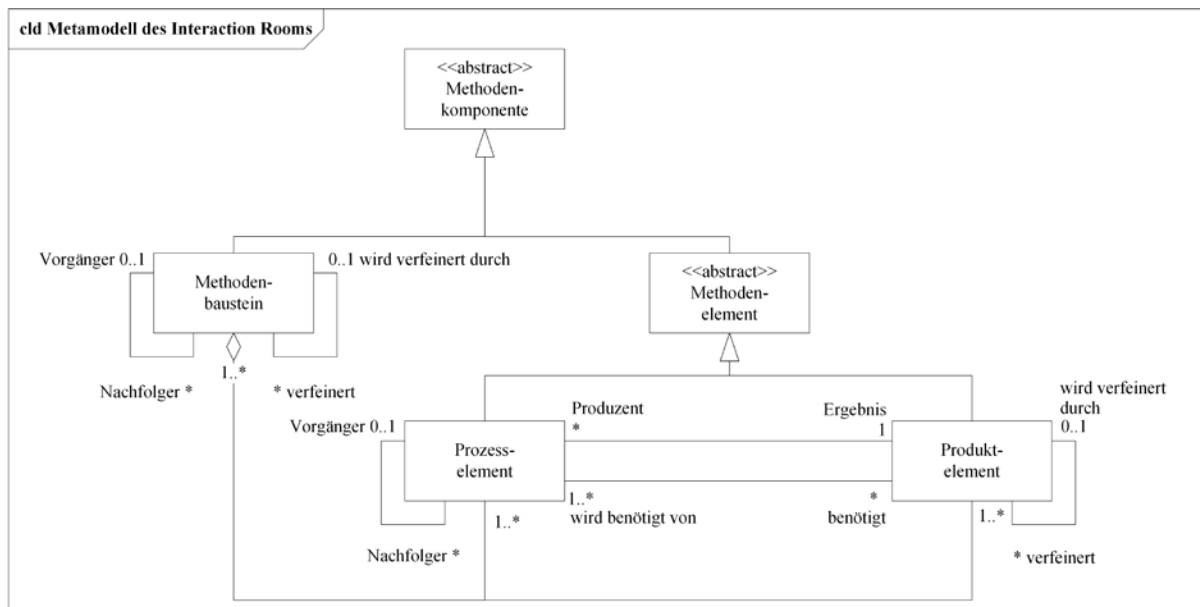


Abbildung 9: Metamodell des Interaction Rooms (angelehnt an [194])

In einem Interaction-Room-Projekt wird mindestens einmal der Methodenbaustein Interaction-Room-Workshop verwendet. Jeder Interaction-Room-Workshop besteht aus den Methodenbausteinen Vorbereitung (vgl. Kapitel 5.1.2), Landkarten (vgl. Kapitel 5.1.3), Annotationen (vgl. Kapitel 5.1.4) und Erkenntnisdokumentation (vgl. Kapitel 5.1.5). In diesem Kapitel wird ein Überblick über alle Methodenbausteine gegeben, wobei die Interaction-Room-Landkarten, die Interaction-Room-Annotationen und die Erkenntnisdokumentation in den Kapiteln 5.2, 5.3 und 5.4 ausführlich beschrieben werden. Eine Tabelle mit allen Methodenbausteinen, Prozess- und Produktelementen sowie deren Dekomposition bzw. Aggregation befindet sich in Anhang A.

Auf die ausführliche Beschreibung notwendiger Raumausstattung wird verzichtet, weil diese sich auf große Whiteboards und ausreichend Bewegungsfreiheit vor den Whiteboards beschränkt. Um die Vor- und Nachbereitung sowie die Durchführung von Interaction-Room-Workshops zu unterstützen, wurden Produktelemente entworfen, die in den folgenden Kapiteln erwähnt, jedoch nicht im Detail vorgestellt werden. Auf die detaillierte Vorstellung wurde verzichtet, weil die Produktelemente zur Anwendung des Interaction Rooms wichtig sind, jedoch nicht zum Verstehen des Interaction Rooms. Diese Produktelemente werden in Anhang A vorgestellt.

Bevor die Methodenbausteine des Interaction Rooms eingeführt werden, werden im folgenden Kapitel die Rollen zur Durchführung von Interaction-Room-Workshops beschrieben.

### 5.1.1 Rollen

Ein Interaction-Room-Workshop wird durch zwei Personen geleitet, die Interaction-Room-Coaches heißen. Ein Methodenexperte (IR-Methoden-Coach, IR-MC) übernimmt die Moderation des Workshops. Ein Domänenexperte (IR-Domänen-Coach, IR-DC) beurteilt geführte Diskussionen, setzt Impulse aufgrund der eigenen Expertise, unterstützt bei der Befüllung der Landkarten und dokumentiert die Diskussion der Annotationen.

Auf die Fähigkeiten eines guten Moderators wird hier nicht weiter eingegangen, weil ausreichend Literatur dazu existiert (vgl. z. B. [195]). Stattdessen wird auf die spezifischen Aufgaben und die dafür notwendigen Fähigkeiten der IR-Coaches eingegangen.

#### 5.1.1.1 Methodenexperte

Der IR-MC ist für das Skizzieren der Landkarten zuständig. Er muss die Syntax, Semantik und Pragmatik der Landkarten kennen (vgl. Kapitel 5.2). Er muss Erfahrung in der Modellierung besitzen, mit der er die Stakeholder bei der Erstellung der Landkarten unterstützt. Er darf keine eigenmächtigen Änderungen an den Landkarten vornehmen, da dies zur Verringerung der Interaktion und zu einem geringeren Commitment der Stakeholder führt [196]. Er muss die Fähigkeit besitzen, den Inhalt von Diskussionen abstrahieren zu können, um die Essenz einer Diskussion auf einer Landkarte darstellen zu können. Dazu muss er irrelevante Details entfernen und den Fokus auf allgemeingültige Konzepte legen [197]. Um den Stakeholdern die Interaction-Room-Annotationen und deren Verwendung erläutern zu können, muss der IR-MC die einzelnen Annotationstypen kennen und diese kontextspezifisch einsetzen können (vgl. Kapitel



5.3.5). Der IR-MC darf geringes Domänenwissen besitzen, weil dies förderlich sein kann, falsches implizites Verständnis zwischen den Stakeholdern durch Verständnisfragen aufzudecken [134]. Durch Domänenwissen sinkt die kognitive Last bei der Moderation, weswegen ein Domänenexperte bei der Durchführung der Methode unterstützt.

#### 5.1.1.2 Domänenexperte

Der Domänenexperte (IR-Domänen-Coach, IR-DC) unterstützt bei der Wahrung eines erkenntnisbringenden Abstraktionsniveaus und muss mit der Domäne, für die das Informationssystem entwickelt wird, vertraut sein. Auf Basis seines Domänenwissens muss er überprüfen, ob sich die Diskussion zwischen den Stakeholdern auf einem angemessenen Abstraktionsniveau befindet oder Stakeholder irrelevante Details diskutieren. Der IR-DC darf nicht die Rolle des neutralen Moderators einnehmen, sondern muss bewusst Impulse setzen, kritische Fragen stellen und aus seiner eigenen Erfahrung sprechen. Die Trennung von Moderation und Fachwissen besitzt den Vorteil, dass der IR-MC sich auf die Gruppe, die gruppenspezifische Prozesse und die Anwendung der Methode konzentrieren kann, wohingegen sich der IR-DC auf den Inhalt konzentrieren kann.

#### 5.1.1.3 Stakeholder

Die Stakeholder stellen das fachliche und technische Wissen für den Inhalt eines Interaction-Room-Workshops bereit, was die Auswahl der richtigen Stakeholder erfolgskritisch macht [198]. In der Literatur werden verschiedene Ansätze beschrieben, wie Stakeholder identifiziert und ausgewählt werden können (vgl. z. B. [190], [199]–[201]). Um die Vorgehensweise zur Identifikation und Auswahl von Stakeholdern nicht vorzuschreiben, werden im Folgenden notwendige Fähigkeiten und Befugnisse der Stakeholder eines Interaction-Room-Workshops definiert:

1. Die Stakeholder müssen alle relevanten Bereiche eines Projekts repräsentieren. Die Zusammensetzung der Stakeholder muss mindestens fachlich und organisatorisch heterogen sein.
2. Die Stakeholder müssen Experten in ihrem Bereich und in der Lage sein, ihr implizites Wissen vermitteln zu können. Spezialisten, die nicht in der Lage sind, ihr Wissen zu kommunizieren, sind als Stakeholder ungeeignet [190].
3. Die Stakeholder müssen Entscheidungsbefugnis besitzen. Sind nur Personen höher gestellter Hierarchieebenen einer Organisation mit Entscheidungsbefugnis ausgestattet, müssen zusätzlich Stakeholder eingeladen werden, die Produkt-, Domänen- und Marktkenntnisse aus erster Hand besitzen. Ausschließlich Personen auszuwählen, die seit langem keinen

echten Produktbezug mehr besitzen und die Realität nur aus Erzählungen kennen, ist nicht zielführend [190].

4. Mindestens ein Projektverantwortlicher muss während eines Workshops anwesend sein, damit die IR-Coaches spontan notwendige Änderungen des Workshop-Ablaufs abstimmen können.

### 5.1.2 Methodenbaustein Vorbereitung

Ein Interaction-Room-Workshop beginnt mit einer Analyse des Workshop-Kontextes. Mit Hilfe eines Fragebogens (vgl. Anhang B) werden das Projektziel, das Workshop-Ziel und der Projekttyp erhoben. Des Weiteren werden voraussichtlich diskutierte Themen und kritische Aspekte erfragt. Es wird eine Stakeholder-Analyse durchgeführt. In der Regel existieren mehr Stakeholder, als aktiv einbezogen werden können. Aufgabe der Interaction-Room-Coaches ist es, mit den Projektverantwortlichen die relevanten Stakeholder für einen Interaction-Room-Workshop auszuwählen. Hierfür müssen zunächst alle Stakeholder identifiziert werden, die Interesse an dem Projekt und Einfluss auf das Projekt besitzen. Diese werden in rollenhomogene Gruppen eingeteilt und Vertreter der Gruppe entsprechend der in Kapitel 5.1.1.3 ausgewählten Kriterien identifiziert. Die Auswahl der Stakeholder für einen Workshop hängt vom entsprechenden Workshop-Ziel ab. Für potentielle Stakeholder eines Workshops werden deren Ziele in Bezug zum Workshop-Ziel analysiert und auf Zielkonflikte überprüft. Dabei erfolgt unter anderem eine Zuordnung von Stakeholdern zu Themen, die im Workshop diskutiert werden. Die Themen dienen als inhaltlicher Rahmen des IR-Workshops und es muss sichergestellt werden, dass mindestens ein Stakeholder für jedes Thema eingeladen wird. Sofern notwendig, werden Einzelgespräche mit ausgewählten Stakeholdern geführt sowie existierende Dokumente analysiert. So werden Themen und Zielsetzung konkretisiert sowie die Stakeholder-Auswahl reflektiert und ggf. angepasst.

Um die kognitive Last für die IR-Coaches und die Stakeholder handhabbar zu gestalten, jedoch gleichzeitig Erkenntnisgewinn erzielen zu können, sollten an einem Interaction-Room-Workshop zwischen acht und zwölf Stakeholder teilnehmen.

Auf Grundlage der Ergebnisse aus der Kontextanalyse wird die Methode für die Verwendung im jeweiligen Kontext konfiguriert. Es werden die zu verwendenden Landkarten (vgl. Kapitel 5.2) und Annotationstypen (vgl. Kapitel 5.3) ausgewählt, um die Workshop-Ziele erreichen zu

können. Die ausgewählten Annotationstypen je Landkarte heißen Annotationsset, das vollständige Vorgehen zur Erstellung von Annotationssets ist in Kapitel 5.3.5.1 beschrieben.

Es wird die so genannte Feature-Landkarte initial mit den identifizierten Themen befüllt. Das Landkartenelement für Themen heißt Epic. Epics werden auf Karteikarten notiert und dienen als thematischer Rahmen für den Interaction-Room-Workshop.

Mit Hilfe der Daten aus der Kontextanalyse und der Methodenkonfiguration wird eine Workshop-Agenda erstellt, die mit den Projektverantwortlichen abgestimmt wird.

### 5.1.3 Methodenbaustein Interaction-Room-Landkarten

Der Interaction Room besitzt vier Landkarten, auf denen das Verhalten und die Struktur eines Informationssystems skizziert wird. Das Verhalten wird auf einer Feature-Landkarte und einer Prozesslandkarte skizziert, die Informationen, die durch das System verarbeitet werden, sowie deren Beziehungen zueinander werden auf einer Objektlandkarte skizziert. Umsysteme der Anwendungslandschaft mit direkten Schnittstellen zum zentralen Informationssystem und ausgetauschte Daten werden auf einer Integrationslandkarte skizziert. Der Zweck jeder Landkarte ist in der folgenden Auflistung beschrieben, in Kapitel 5.2 werden für jede Landkarte eine Syntax, ihre Semantik und die Pragmatik definiert.

- Auf der Feature-Landkarte werden Epics und Features gesammelt, die das System abstrakt, jedoch funktional vollständig beschreiben.
- Auf der Prozesslandkarte werden Abläufe zur Erfüllung einzelner Features dargestellt.
- Auf der Objektlandkarte werden die Geschäftsobjekte und deren Beziehungen zueinander veranschaulicht. Geschäftsobjekte werden im Rahmen der Abläufe erzeugt, manipuliert und gelöscht.
- Auf der Integrationslandkarte werden direkte Schnittstellen des zentralen Informationssystems zu seinen Umsystemen gezeigt. Die Schnittstellen beinhalten eine Referenz auf ausgetauschte Geschäftsobjekte.

Nachdem eine Landkarte skizziert wurde, werden Wert-, Aufwands- und Risikotreiber mit Hilfe der Annotationstypen des vorgesehenen Annotationssets gekennzeichnet.

#### 5.1.4 Methodenbaustein Interaction-Room-Annotationen

Insgesamt sind 20 Annotationstypen definiert. Ein Annotationstyp wird durch ein Symbol repräsentiert, das auf einen Klebezettel gedruckt ist. Ein Annotationstyp wird auf ein Landkartenelement der Prozess-, Objekt-, und Integrationslandkarte geklebt und im Anschluss diskutiert. Sofern es sich bei einem Annotationstyp auf einem Landkartenelement nicht um eine offensichtliche oder triviale Anforderung handelt, wird das Diskutierte dokumentiert. Mit einer projektweit eindeutigen ID wird die Referenz zwischen dem Annotationstyp auf der Landkarte und der dokumentierten Begründung hergestellt. Ein Annotationstyp mit seiner ID und der dokumentierten Begründung heißt Annotation. Um einen Annotationstyp zu diskutieren, fragt der IR-MC, was durch den Annotationstyp ausgedrückt werden soll, worauf sich der Annotationstyp bezieht und welcher Nutzen durch die Berücksichtigung der Annotation entstehen würde. Der Stakeholder, der das Landkartenelement mit dem Annotationstyp gekennzeichnet hat, heißt Befürworter und beantwortet die Fragen. Die Dokumentation erfolgt über eine standardisierte Vorlage (vgl. Anhang C).

Am Ende eines Interaction-Room-Workshops wird eine obligatorische Ungewissheitsbewertung aller Landkarten mit dem Annotationstyp Ungewissheit durchgeführt. Die isolierte Verwendung des Annotationstyps am Ende eines Workshops dient der Reflexion über die diskutierten Inhalte des Workshops sowie der Kennzeichnung von Projektrisiken durch nicht ausreichend verstandene Aspekte.

#### 5.1.5 Methodenbaustein Erkenntnisdokumentation

Um die Erkenntnisse aus einem Interaction-Room-Workshop zu dokumentieren, werden die Ergebnisse der Kontextanalyse, skizzierte und annotierte Landkarten und die dokumentierten Ergebnisse der Diskussion von Annotationstypen in ein Anforderungsdokument überführt (vgl. Kapitel 5.4). Im Methodenbaustein Landkartendigitalisierung werden Fotos der Whiteboards in einem Modellierungswerkzeug digitalisiert, um diese Modelle im Anschluss zu beschreiben. Die Begründung der Annotationen an Elementen der Landkarten wird in den beschreibenden Text eingearbeitet. Daraufhin wird das Anforderungsdokument erstellt. Die Ergebnisse der Kontextanalyse werden im ersten Kapitel „Vision“ dokumentiert. Im zweiten Kapitel „Überblick“ werden die Objekt- und Integrationslandkarte dokumentiert. Aus den Informationen der Feature- und der Integrationslandkarte wird eine rudimentäre Version eines Use-Case-Diagramms erstellt, die ebenfalls Teil des Kapitels „Überblick“ ist. Im dritten Kapitel „Detaillierte

Anforderungen“ wird jeder Use Case in einem Unterkapitel detailliert beschrieben. Dazu wird mindestens ein Prozess der Prozesslandkarte je Use Case verwendet. Das vierte Kapitel „Anhang“ enthält eine tabellarische Darstellung der Inhalte der Feature-Landkarte und ein Protokoll der dokumentierten Annotationsbegründungen. Das fertige Dokument wird mit den Projektverantwortlichen abgestimmt.

## 5.2 Interaction-Room-Landkarten

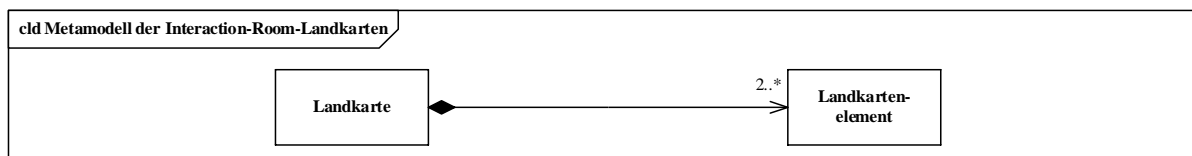


Abbildung 10: Metamodell der Interaction-Room-Landkarten

Abbildung 10 zeigt das Konzept der Interaction-Room-Landkarten. Eine Landkarte besteht aus mindestens zwei Landkartenelementen. Ein Landkartenelement (Element) ist immer Bestandteil einer Landkarte.

Die Syntax und Semantik der Landkarten sind jeweils Teilmengen bereits existierender Notationen, sodass auf die formale Definition von Syntax und Semantik verzichtet wird. Stattdessen wird jede Landkarte durch ein Metamodell beschrieben. Sofern die Semantik einzelner Elemente von dem Äquivalent der Ursprungsnotation abweicht, wird darauf hingewiesen sowie der Unterschied beschrieben.

Die Landkartenelemente der Prozesslandkarte sind eine Teilmenge der Syntaxelemente der Business Process Model and Notation (BPMN). Die Landkartenelemente der Objektlandkarte sind eine Teilmenge der Syntaxelemente von UML-Klassendiagrammen. Die Landkartenelemente der Integrationslandkarte basieren auf einer Notation, die von Hanschke [202] für die Dokumentation von Informationsflüssen zwischen Anwendungen vorgestellt wurde. Die Feature-Landkarte besteht aus Epics, die durch Features detailliert werden. Neben den vier Landkarten kann zur Workshop-Unterstützung ein unstrukturierter Themenspeicher verwendet werden [195]. Ein Themenspeicher im Interaction Room ist eine Sammlung aller Aspekte, die nicht durch die Syntax der Landkarten beschrieben werden können. Der Themenspeicher wird punktuell in den folgenden Kapiteln erwähnt, wo dies zweckmäßig erscheint.

### 5.2.1 Feature-Landkarte

Auf der Feature-Landkarte werden funktionale Anforderungen an das zu realisierende System gesammelt.

#### 5.2.1.1 Syntax und Semantik

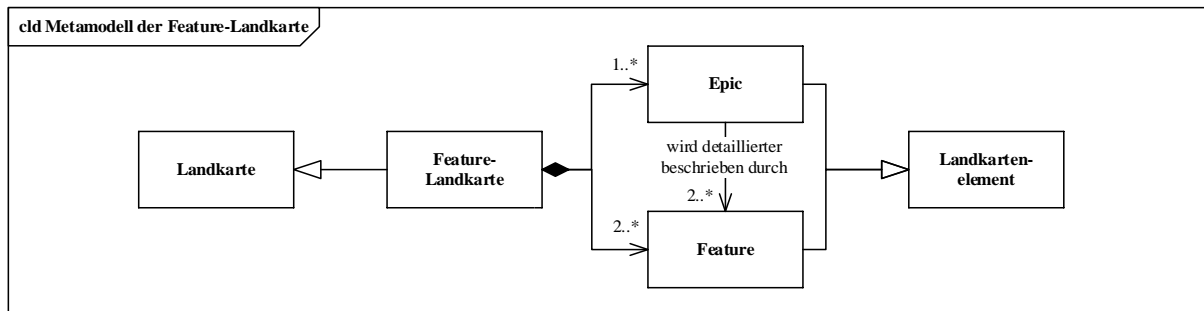


Abbildung 11: Metamodell der Feature-Landkarte

Auf der Feature-Landkarte werden die funktionalen Anforderungen an ein Informationssystem als Epics und Features beschrieben (vgl. Abbildung 11). Die Feature-Landkarte besteht mindestens aus einem Epic und zwei Features, wobei ein Epic durch mindestens zwei Features detaillierter beschrieben wird. Ein Epic bezeichnet eine abstrakte Anforderung, die zu grob und zu wenig verstanden ist, um diese schätzen zu können [203]. Ein Vorteil von Epics ist, dass sich ein System mit wenigen Epics vollständig funktional beschreiben lässt [ebenda]. Epics dienen als funktionaler Ordnungsrahmen für ein Informationssystem. Ein Feature entspricht einer Teilaufgabe eines Epics, die zur Zielerreichung eines Akteurs beiträgt. Es muss zudem einen Abstraktionsgrad besitzen, der den Ablauf zum Erfüllen der funktionalen Anforderung durch eine Menge von Schritten beschreibbar macht. Dabei ist ein Feature nicht mit einem Prozess gleichzusetzen, denn ein Feature beschreibt nur die funktionalen Anforderungen an das Informationssystem und nicht die Aktivitäten, die zum Erfüllen der Anforderung durchgeführt werden müssen.

#### 5.2.1.2 Pragmatik

Um die Feature-Landkarte zu befüllen, werden zunächst die Epics definiert. Dies geschieht in Vorbereitung auf einen Interaction-Room-Workshop durch Absprache mit den Projektverantwortlichen (vgl. Kapitel 5.1.2). Zur Detaillierung der Epics werden folgende Schritte durchgeführt:

1. Der IR-MC stellt den Stakeholdern die Epics vor und fragt, ob das System durch die Epics vollständig beschrieben ist oder ob noch weitere Epics hinzugefügt werden sollen. Gegebenenfalls werden fehlende Epics ergänzt.
2. Der IR-MC erläutert den Zusammenhang zwischen Epics und Features sowie das Ziel der Features. Des Weiteren gibt er Beispiele zur Verdeutlichung des Abstraktionsgrades der Features.
3. Der IR-MC bittet die Stakeholder, die wichtigsten Features je Epic auf Karteikarten zu notieren. Jeder Stakeholder muss sich auf ungefähr zehn Features je Epic beschränken. Dadurch müssen die Stakeholder reflektieren, was für sie die wesentlichen funktionalen Anforderungen sind.
4. Der IR-Coach bittet jeden Stakeholder, seine Features vorzustellen, jedes Feature einem Epic zuzuordnen und auf die Feature-Landkarte zu heften. Die anderen Stakeholder werden aufgefordert Verständnisfragen zu stellen. Duplikate werden durch Übereinanderheften konsolidiert.
5. Während der Vorstellung der Features werden in der Praxis auch nicht funktionale Anforderungen, Wünsche, Ziele, Fragen und Schnittstellen auf den Karteikarten notiert. Die IR-Coaches achten darauf, Karteikarten auszusortieren, die keine funktionalen Anforderungen beschreiben. Schnittstellen und Systeme werden auf der Integrationslandkarte, Geschäftsobjekte auf der Objektlandkarte notiert. Alles, was sich diesen beiden Landkarten nicht zuordnen lässt, wird im Themenspeicher dokumentiert.
6. Nachdem jeder Stakeholder seine Features vorgestellt hat, wird das Abstraktionsniveau der Features durch Konsolidierung oder Dekomposition angeglichen.

### 5.2.2 Prozesslandkarte

Auf einer Prozesslandkarte werden Abläufe skizziert, die durch ein Informationssystem unterstützt werden sollen. Die Syntax, Semantik und Pragmatik folgen den sieben Empfehlungen für die Prozessmodellierung nach Mendling, Reijers und van der Aalst [204] (vgl. Tabelle 3) und werden im Folgenden entsprechend referenziert.

Tabelle 3: Sieben Empfehlungen für die Prozessmodellierung (angelehnt an [204])

<i>ID</i>	<i>Empfehlung (E)</i>
E1	Nutze so wenig Elemente wie möglich. Je größer die Anzahl der Elemente ist, desto schwieriger ist das Modell zu verstehen und desto höher ist die Fehlerwahrscheinlichkeit.
E2	Minimiere die Routing-Pfade (z. B. ausgehende Sequenzflüsse je Entscheidung) pro Element. Das erhöht die Verständlichkeit.
E3	Nutze ein Start- und ein Endereignis je Prozess, da mit zunehmender Anzahl die Fehlerwahrscheinlichkeit steigt.
E4	Modelliere so strukturiert wie möglich. Bspw. zu jeder Verzweigung auch eine Zusammenführung. Das reduziert Fehler und erhöht die Verständlichkeit.
E5	Vermeide das logische Oder, da die Verwendung des logischen Oders zu Fehlern führen kann und daraus Probleme bei der Semantik von Zusammenführungen entstehen.
E6	Nutze den Objekt-Verb-Stil für die Benennung von Aktivitäten, da diese signifikant eindeutiger sind als bspw. die Benennung durch Aktionen gefolgt von Nomen.
E7	Unterteile Modelle, die mehr als 50 Modellelemente besitzen, um die Komplexität zu reduzieren (siehe E1) und die Verständlichkeit zu erhöhen.

### 5.2.2.1 Syntax und Semantik

Abbildung 12 zeigt das Metamodell der Prozesslandkarte. Die Landkartenelemente sind eine Teilmenge der Syntaxelemente von Fluss- und Verbindungsobjekten der BPMN (vgl. [56]). Eine Prozesslandkarte besteht aus genau einem Startereignis, mindestens einem Endereignis, mindestens einer Aktivität, mindestens zwei Sequenzflüssen und kann exklusive sowie parallele Gateways besitzen. Alle Landkartenelemente sind Bestandteil mindestens einer Prozesslandkarte.

Das Landkartenelement Startereignis entspricht dem ungetypten Startereignis der BPMN und wird, Freund, Rücker und Henninger [205] folgend, durch eine Kombination eines Objekts und eines passiv formulierten Verbs benannt.

Das Landkartenelement Endereignis entspricht dem ungetypten Endereignis der BPMN und wird ebenfalls durch ein Objekt und passiv formuliertes Verb benannt.

Das Landkartenelement Aktivität entspricht einer Aktivität der Business Process Model and Notation. Die Aktivität der BPMN ist ein abstraktes Element, das in Prozessmodellen als atomare Aktivität (Task) oder nicht atomare Aktivität (Subprozess) verwendet und in ihrer grafischen Repräsentation unterschieden wird. Die Aktivität auf einer Prozesslandkarte kann atomar oder zusammengesetzt sein, wird jedoch grafisch nicht unterschieden, sondern immer als



Rechteck mit abgerundeten Ecken dargestellt. Anders als in der BPMN ist die Typisierung einer Aktivität nicht zulässig. Eine Aktivität wird nach dem Objekt-Verb-Pattern benannt (vgl. Empfehlung E6 aus Tabelle 3).

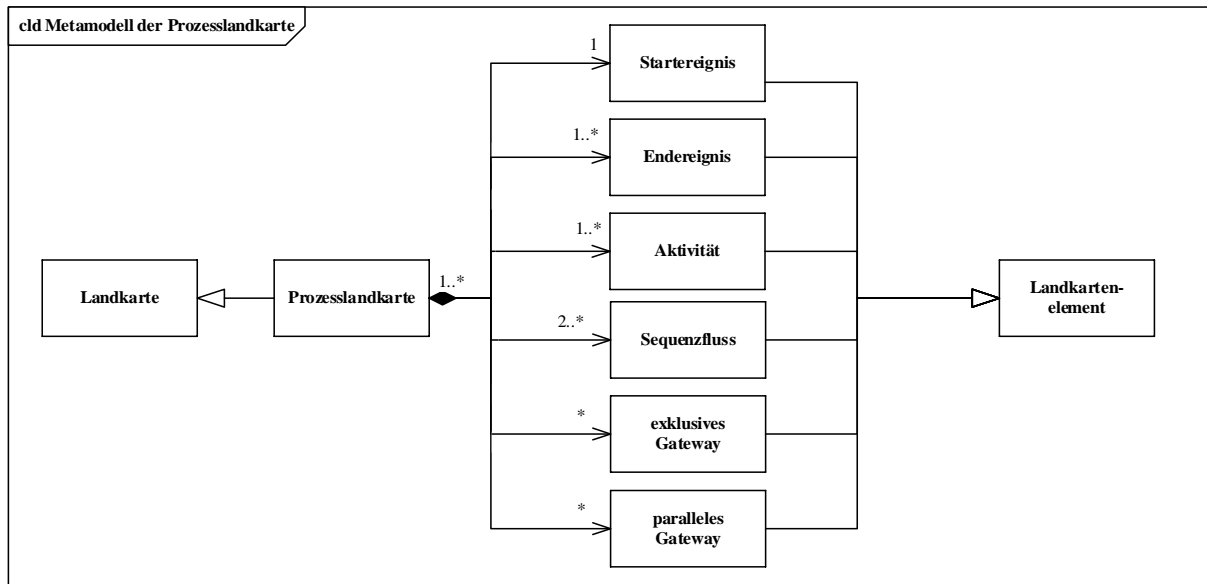


Abbildung 12: Metamodell der Prozesslandkarte

Ein Sequenzfluss verbindet immer genau zwei der anderen Landkartenelemente miteinander und entspricht dem Sequenzfluss der BPMN. Default und Conditional Sequence Flows sind nicht zulässig.

Das parallele Gateway repräsentiert ein logisches Und und entspricht den parallelen Gateways der BPMN. Das exklusive Gateway repräsentiert ein exklusives Oder und entspricht dem exklusiven Gateway der BPMN. Durch beide Gateways können Sequenzflüsse sowohl verzweigt als auch zusammengeführt werden. An den ausgehenden Sequenzflüssen exklusiver Gateways wird eine Bedingung notiert. Eine Synchronisation erfordert ein verzweigendes und zusammenführendes paralleles Gateway. Um Missverständnisse zu vermeiden und die Lesbarkeit zu erhöhen, kann der Empfehlung E4 nach [204] gefolgt werden und immer ein zusammenführendes Gateway skizziert werden. Der Empfehlung E5 nach Mendling et al. [204] folgend, wird auf die Verwendung eines inklusiven Gateways zur Darstellung eines logischen Oders verzichtet.

Abbildung 13 zeigt die grafische Darstellung der Elemente einer Prozesslandkarte.

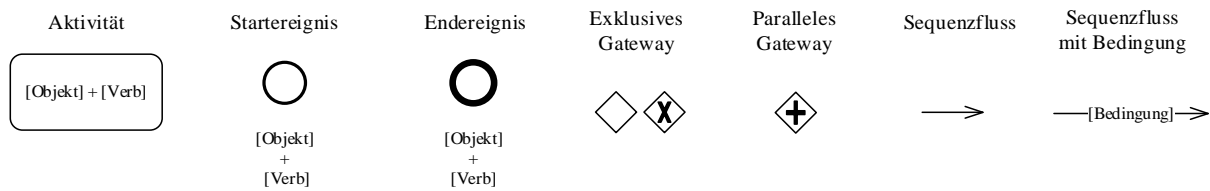


Abbildung 13: Grafische Darstellung der Elemente einer Prozesslandkarte

### 5.2.2.2 Pragmatik

Bei der Befüllung der Feature-Landkarte werden funktionale Anforderungen gesammelt, deren Erfüllung nun durch Skizzieren einer zeitlich-logischen Abfolge von Schritten beschrieben wird. Um die Komplexität einer Prozesslandkarte gering zu halten und so zu besserem Verständnis beizutragen (vgl. Empfehlungen E1, E2 aus Tabelle 3 und [206]), wird der Ablauf zur Erfüllung von circa fünf Features durch eine Prozesslandkarte beschrieben.

Für die Erstellung einer Prozesslandkarte werden die folgenden Schritte durchgeführt:

1. Zu Beginn erläutert der IR-MC den Stakeholdern das Ziel der Prozesslandkarte, den Zusammenhang zwischen Feature und Prozesslandkarte und führt die Landkartenelemente ein. Auf die detaillierte Einführung von Gateways kann zunächst verzichtet werden (vgl. Empfehlungen E1, E2 aus Tabelle 3).
2. Der IR-MC und die Stakeholder identifizieren Features, mit denen begonnen werden soll, und hängen diese temporär von der Feature-Landkarte auf die Prozesslandkarte.
3. Der IR-MC fragt, welches Ereignis eingetreten sein muss, damit der Prozess beginnt, und notiert das Startereignis durch ein Objekt und ein passives Verb (vgl. Empfehlung E3 aus Tabelle 3). Der IR-DC notiert das Objekt als Kandidat für ein Geschäftsobjekt auf der Objektlandkarte.
4. Der IR-MC und die Stakeholder skizzieren den so genannten „Happy Path“ [205]. Der Happy Path ist der ideale Ablauf eines Prozesses ohne Entscheidungen und alternative Ablaufpfade (vgl. Empfehlung E4 aus Tabelle 3). Bei der Benennung von Aktivitäten achtet der IR-MC darauf, dass das Objekt-Verb-Muster eingehalten wird (vgl. Empfehlung E6 aus Tabelle 3). Der Domänen IR-Coach notiert parallel Objekte als Kandidaten für Geschäftsobjekte auf der Objektlandkarte.
5. Erst wenn der Happy Path skizziert ist, beginnt der IR-MC mit den Stakeholdern, alternative und parallele Sequenzflüsse zu skizzieren (vgl. Empfehlung E4 aus Tabelle 3). Dazu werden zunächst die Stellen im Prozess identifiziert, an denen der Sequenzfluss verzweigt.

6. Für jede Verzweigung erfragt der IR-MC, wie viele unterschiedliche ausgehende Sequenzflüsse benötigt werden, ob es sich um alternative oder parallele Abläufe handelt und unter welcher Bedingung der Prozess mit welchem Sequenzfluss fortgeführt wird. Häufig ist es notwendig, die Bedingungen zu gruppieren, dadurch die Anzahl alternativer Pfade zu reduzieren und so das Abstraktionsniveau angemessen zu halten (vgl. Empfehlungen E1, E2, E4 aus Tabelle 3).
7. Der IR-MC notiert die ausgehenden Sequenzflüsse und ggf. deren Bedingungen. Für jeden ausgehenden Sequenzfluss wird erst der Happy Path skizziert, dann erfolgt wieder die sukzessive Verfeinerung durch Gateways (vgl. Empfehlung E4 aus Tabelle 3). Hierbei achten die IR-Coaches darauf, dass die Komplexität handhabbar bleibt, oder teilen die Prozesslandkarte in mehrere Prozesslandkarten auf (vgl. Empfehlung E7 aus Tabelle 3).
8. Der IR-DC notiert die Features, deren Erfüllung durch die Prozesslandkarte beschrieben wird, und hängt die Features zurück auf die Feature-Landkarte.

In der Praxis hat sich gezeigt, dass Stakeholder statt der Formulierung abstrakter Aktivitäten zur Beschreibung eines Ablaufs den Ablauf auf Instanzebene wiedergeben. Sie nennen zusätzlich zu den Aktivitäten ausführende Rollen, Systeme und Masken, auf denen die Rolleninhaber arbeiten. Diese Information wird direkt an den entsprechenden Landkartenelementen notiert. Die Dokumentation erfolgt aus zwei Gründen: Erstens bedeutet die vollständige Aufnahme der Information eines Stakeholders eine Wertschätzung seiner Äußerung. Zweitens kann aus dieser Information zusätzliche Erkenntnis gewonnen werden: Durch das Notieren von Systemen können Schnittstellen und ausgetauschte Daten identifiziert und die Integrationslandkarte kann validiert werden. Mit Hilfe der Masken kann ein Dialogfluss der Anwendung erstellt werden und Rollen und Aktivitäten liefern Erkenntnisse für ein Rollen- und Berechtigungskonzept sowie für ein Use-Case-Diagramm. Die Dokumentation der Ergebnisse des Interaction Rooms in ein Anforderungsdokument wird in Kapitel 5.4 beschrieben. Dabei wird auch auf die Dokumentation der zuvor beschriebenen Zusatzinformationen eingegangen.

### 5.2.3 Integrationslandkarte

Auf der Integrationslandkarte werden die Schnittstellen zwischen dem zentralen Informationssystem und Umsystemen skizziert. An den Schnittstellen werden Namen von Geschäftsobjekten notiert, die zwischen den Systemen ausgetauscht werden.

## 5.2.3.1 Syntax und Semantik

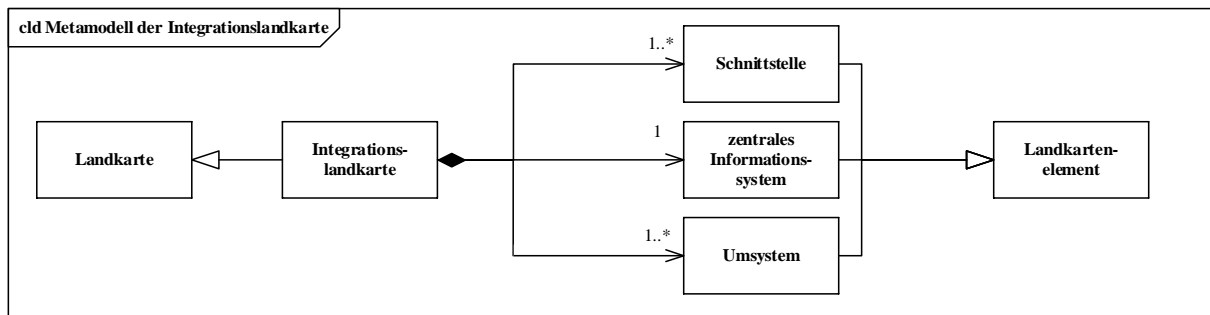


Abbildung 14: Metamodell der Integrationslandkarte

Abbildung 14 zeigt das Metamodell der Integrationslandkarte. Die Landkartenelemente sind an die Syntax der Informationsfluss-Grafik nach Hanschke angelehnt [202]. Hanschke verfolgt mit der Informationsfluss-Grafik das Ziel, Schnittstellen, Datenfluss, logische Komponenten und Daten von Informationssystemen zu visualisieren. Eine Integrationslandkarte verfolgt das Ziel, den Austausch von Geschäftsobjekten über Schnittstellen zwischen Systemen zu visualisieren. Sie besteht aus einem zentralen Informationssystem, mindestens einem Umsystem und mindestens einer Schnittstelle. Das zentrale Informationssystem ist der Gegenstand des Projekts und wird durch die übrigen Landkarten aus anderen Perspektiven beschrieben. Umsysteme sind Systeme mit direkten Schnittstellen zum zentralen Informationssystem. Ein Umsystem und das zentrale Informationssystem sind über mindestens eine Schnittstelle verbunden. Über eine Schnittstelle wird mindestens ein Geschäftsobjekt ausgetauscht. Der Name des ausgetauschten Geschäftsobjektes wird an einer Schnittstelle notiert und ist eine Referenz auf ein Geschäftsobjekt der Objektlandkarte. Schnittstellen zwischen Umsystemen sind nicht zulässig.

Das zentrale Informationssystem und die Umsysteme werden als Rechteck dargestellt (vgl. Abbildung 15), eine Schnittstelle als Pfeil. Auf dem Pfeil steht das ausgetauschte Geschäftsobjekt.

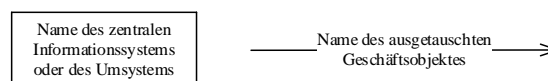


Abbildung 15: Grafische Darstellung der Elemente einer Integrationslandkarte

### 5.2.3.2 Pragmatik

1. Der IR-MC erläutert den Stakeholdern die Landkartenelemente der Integrationslandkarte und skizziert das zentrale Informationssystem in die Mitte der Integrationslandkarte.
2. Während der Befüllung der Feature-Landkarte und der Prozesslandkarte wurden ggf. bereits Umsysteme identifiziert. Die Stakeholder werden gebeten, die Umsysteme zu vervollständigen. Diese werden ebenfalls auf der Integrationslandkarte notiert.
3. Der IR-MC fragt die Stakeholder für jedes System, welche Geschäftsobjekte zwischen den Systemen ausgetauscht werden und in welche Richtung der Austausch stattfindet. Die entsprechende Schnittstelle wird skizziert und das Geschäftsobjekt an der Schnittstelle notiert. Dabei werden mehrere technische Schnittstellen nach fachlicher Zugehörigkeit zusammengefasst.
4. Der IR-DC überprüft, ob das Geschäftsobjekt bereits auf der Objektlandkarte notiert ist und fügt es hinzu, sofern es nicht bereits auf der Objektlandkarte notiert ist.

### 5.2.4 Objektlandkarte

Auf einer Objektlandkarte werden Geschäftsobjekte und deren Beziehungen skizziert.

#### 5.2.4.1 Syntax und Semantik

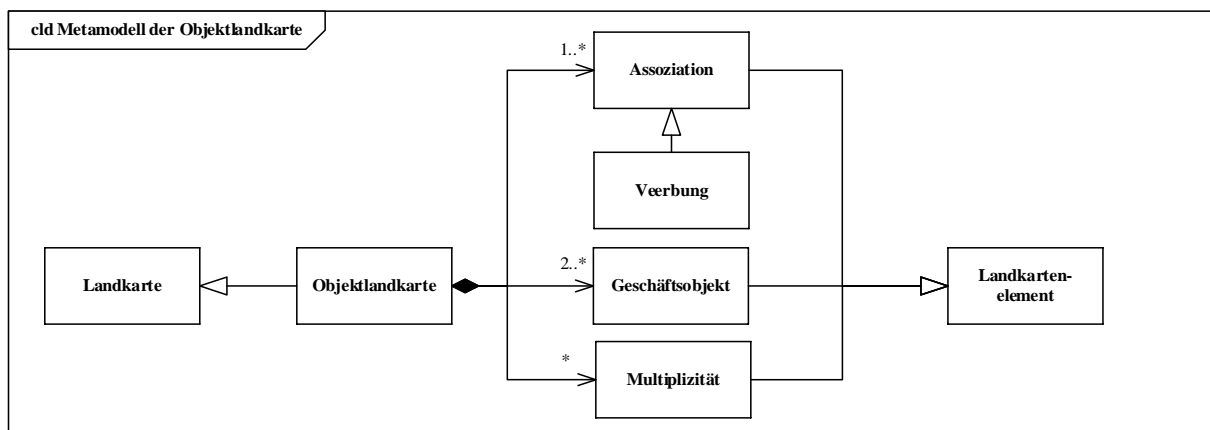


Abbildung 16: Metamodell der Objektlandkarte

Abbildung 16 zeigt das Metamodell der Objektlandkarte. Die Landkartenelemente sind eine Teilmenge der Syntaxelemente von UML-Klassendiagrammen. Eine Objektlandkarte besteht aus Landkartenelementen, deren Verwendung obligatorisch ist, und weiteren Landkartenelementen, deren Verwendung optional ist.

Die obligatorisch zu verwendenden Landkartenelemente sind Geschäftsobjekte und Assoziationen. Ein Geschäftsobjekt entspricht einer Klasse der UML, jedoch ohne Methoden und Attribute. Eine Assoziation entspricht der Assoziation von UML-Klassendiagrammen.

Ein Geschäftsobjekt wird als Rechteck dargestellt, der Name des Geschäftsobjektes wird im Rechteck notiert. Eine Beziehung zwischen zwei Geschäftsobjekten wird durch eine Assoziation dargestellt, die zur ihrer Konkretisierung benannt und gerichtet sein kann (vgl. Abbildung 17).

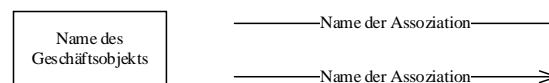


Abbildung 17: Grafische Darstellung obligatorischer Elemente einer Objektlandkarte

Die optionalen verwendbaren Landkartenelemente sind eine Vererbungsbeziehung und Multiplizitäten. Beide Landkartenelemente entsprechen ihren namensgleichen Pendanten der UML-Klassendiagramme. Anders als in der UML bedeutet das Fehlen einer Multiplizität nicht, dass es sich um eine eindeutige Beziehung handelt, sondern dass die Multiplizität an dieser Stelle nicht definiert ist (vgl. Abbildung 18).

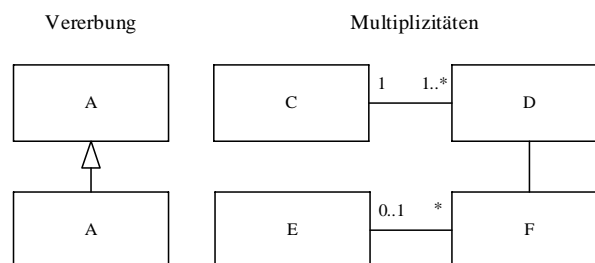


Abbildung 18: Grafische Darstellung optionaler Elemente einer Objektlandkarte

#### 5.2.4.2 Pragmatik

Während des Skizzierens der Prozesslandkarten wurden bereits Kandidaten für Geschäftsobjekte identifiziert, die auf der Objektlandkarte notiert wurden. Während des Skizzierens der Integrationslandkarte können weitere Geschäftsobjekte identifiziert worden sein, die auf der Objektlandkarte notiert wurden. Nun werden die Geschäftsobjekte und deren Beziehungen zueinander sukzessive vervollständigt.

1. Der IR-MC führt die obligatorischen Landkartenelemente der Objektlandkarte ein. Auf die Erläuterung der optionalen Landkartenelemente wird zunächst verzichtet.
2. Durch Konsolidierung, Aggregation oder Eliminierung der Kandidaten für Geschäftsobjekte werden Geschäftsobjekte erstellt.
3. Der IR-MC und die Stakeholder identifizieren die zentralen Geschäftsobjekte und positionieren sie in der Mitte der Objektlandkarte. Zentrale Geschäftsobjekte sind in der Regel Bestandteil mehrerer Prozesslandkarten.
4. Daraufhin setzen der IR-MC und die Stakeholder die Geschäftsobjekte in Beziehung zueinander. Begonnen wird mit den Beziehungen zwischen den zentralen Geschäftsobjekten, um danach sukzessive die übrigen Geschäftsobjekte durch Assoziationen zu verbinden. Tragen Multiplizitäten zum Verständnis bei, werden diese eingesetzt. Ebenso die Vererbungsbeziehung.

#### 5.2.5 Praxisbeispiel

Zur Veranschaulichung der eingeführten Landkarten und deren pragmatischer Handhabung wird der folgende Auszug eines Praxisprojekts dargestellt, das 2016 mit einer deutschen Versicherung durchgeführt wurde.<sup>6</sup> Das Ziel war die Erstellung eines Grobkonzepts für die Entwicklung einer vertriebsunterstützenden Anwendung. Das Grobkonzept wurde zur Ausschreibung der Implementierung der Anwendung verwendet. In den folgenden Absätzen werden die Ausgangssituation und die Zielsetzung des Projekts erläutert, um im Anschluss den Methodenbaustein Vorbereitung zu beschreiben und einen Auszug der verwendeten Landkarten zu zeigen. In Kapitel 5.3.6 wird das Beispiel um die verwendeten Annotationstypen ergänzt.

Der Vertrieb des Versicherungsunternehmens (VU) ist dezentral über Bezirksdirektionen organisiert. Jede Bezirksdirektion (BD) ist einem Postleitzahlengebiet zugeordnet und hat einen Bezirksdirektor (BezDir). Einer BD sind selbständige Außendienstmitarbeiter (ADM) zugeordnet. Das Versicherungsunternehmen generiert Verkaufschancen (Leads) über unterschiedliche Kanäle (z. B. telefonisch, postalisch, über die eigene Webseite, über Landing-Pages zu bestimmten Aktionen). Ein Lead wird erfasst und über seine Postleitzahl (PLZ) in den Lead-Pool der passenden BD weitergeleitet. Daraufhin verteilt der BezDir das Lead manuell an ADMs. Der ADM zahlt 2,5 Euro für das Lead.

---

<sup>6</sup> Zum Zeitpunkt des Verfassens dieser Arbeit lag keine Erlaubnis zur namentlichen Nennung vor, daher wurden Bezeichnungen und Begriffe verfremdet.

Basierend auf dieser Ausgangslage hat das VU folgende Zielsetzung formuliert:

- Die Verteilung der Leads an ADMs soll transparenter werden. Alle ADMs sollen die gleichen Chancen besitzen, einen Lead zu kaufen.
- Die Wertigkeit der Leads (möglicher Umsatz und Höhe der Provision) soll besser zur Geltung kommen, weswegen Leads kategorisiert und jede Kategorie zu einem bestimmten Preis verkauft werden soll.
- Der Zeitraum zwischen dem Eingang und der Bearbeitung von Leads soll verkürzt und Medienbrüche sollen reduziert werden.
- Das Controlling vertrieblicher Aktivitäten sowie der Marketingaktivitäten (z. B. Erfolgsquoten und Wirksamkeit von Rabattaktionen) soll verbessert werden, um die Vertriebs- und Marketingsteuerung optimieren zu können.

Die automatische Schlüsselung der Leads zu BDs sowie das automatische Einstellen der Leads in die Lead-Pools waren bereits in einem produktiven Lead-System implementiert. Wegen umfangreicher zusätzlicher Funktionalität (u. a. sollte eine mobile Anwendung für den Erwerb der Leads durch die ADMs entwickelt werden), wurde das Projekt als Neuentwicklungsprojekt klassifiziert. Des Weiteren existierte die Rahmenbedingung, das neue Lead-System (LS) in die Anwendungslandschaft des VUs zu integrieren. Zu Projektbeginn war bereits bekannt, dass das LS mit dem Customer-Relationship-Management (CRM)-System, dem Agentursystem zur Verwaltung von Angeboten und Verträgen durch den ADM sowie mit dem Data Warehouse (DW) integriert werden musste.

In der Stakeholder-Analyse des Vorabfragebogens wurde ein Unternehmensarchitekt, ein Entwickler, ein Mitarbeiter des Vertriebsinnendienstes, ein Mitarbeiter aus dem Marketing, ein Kundenbetreuer, der technische und der fachliche Projektleiter als Teilnehmer des Interaction Rooms vorgeschlagen. Nach einer Abstimmung mit den Projektverantwortlichen wurden zusätzlich die Rollen BezDir und ADM eingeladen, die durch eine Person repräsentiert wurden.

Im Rahmen der Kontextanalyse wurde ein existierendes Dokument mit Anforderungen an das neue System analysiert. In dem Dokument waren fachliche, technische und organisatorische Anforderungen sowie Ziele beschrieben. Es wurden Lead-Kategorien definiert, über die der Preis eines Leads festgelegt werden sollte. Ein Verkaufszeitraum für Leads war definiert, es sollte außerdem ein dezidiertes Lead-Konto zur Verrechnung von Kauf und Rückerstattung



wegen Reklamation geben. Es wurde definiert, dass der Status eines Leads nachverfolgbar sein soll und dass ADMs Postleitzahlen im PLZ-Gebiet ihrer BD buchen können sollen.

Die Kontextanalyse führte dazu, dass die Feature-Landkarte als Ordnungsrahmen für die Fachlichkeit des Workshops, die Prozesslandkarte zur Detaillierung der Feature-Landkarte, die Integrationslandkarte zur Analyse technischer Abhängigkeiten und die Objektlandkarte zur Analyse der fachlichen Struktur des Systems ausgewählt wurden.

### 5.2.5.1 Feature-Landkarte

Aus den Ergebnissen der Kontextanalyse wurden Epics und Features abgeleitet, um dadurch den thematischen Rahmen des Workshops zu definieren (vgl. Abbildung 19). Nach einer Einleitung wurde die Feature-Landkarte vorgestellt und durch die Stakeholder validiert. Neben der Validierung sollten die Stakeholder fehlende Epics und Features ergänzen. Die rot umrandeten Epics und Features wurden nachträglich hinzugefügt, jedoch nicht ausschließlich in der aktuellen Workshop-Phase, sondern auch während der Diskussion über andere Landkarten.

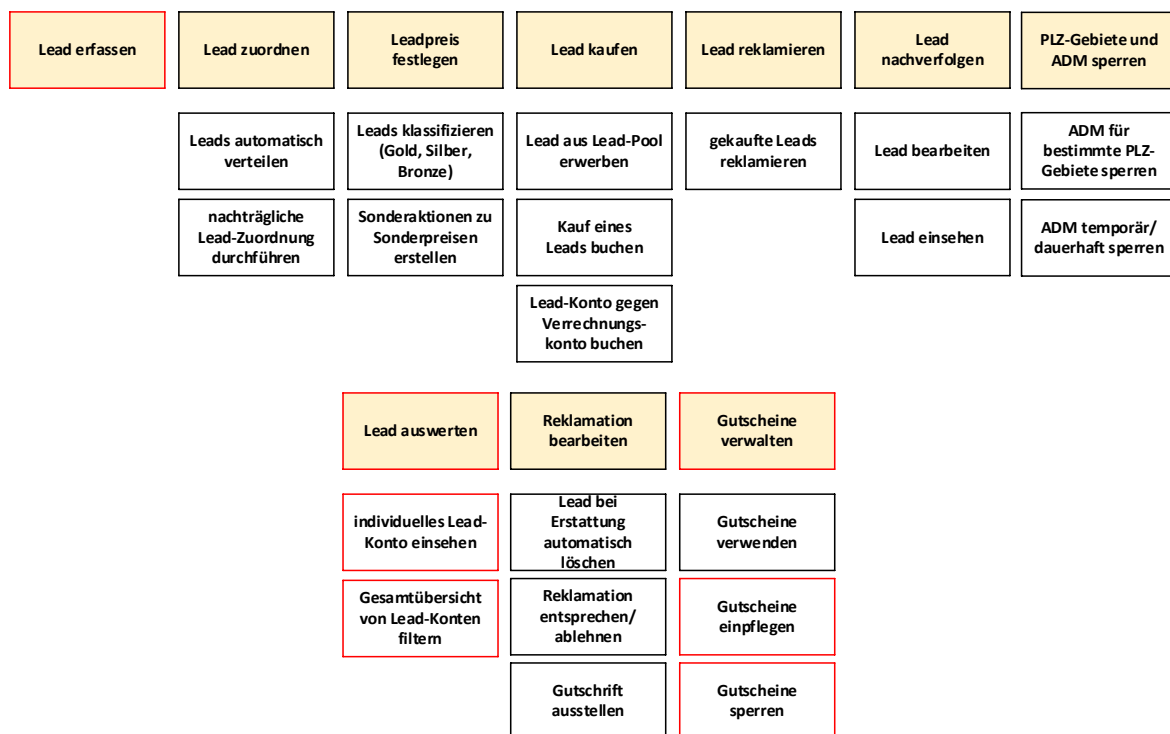


Abbildung 19: Feature-Landkarte des Praxisbeispiels

Die Epics und Features treffen keine Aussage darüber, welche Rollen die Funktionalität ausführen sollen, jedoch wurde aus der Diskussion der Feature-Landkarte unmittelbar ersichtlich, dass mehrere Rollen Berechtigung auf Funktionen des Systems benötigen (z. B. wird ein Lead

vom ADM reklamiert, aber die Reklamation nicht von ihm bearbeitet). Daraus ergab sich die Notwendigkeit, Rollen und Berechtigungen im Workshop zu diskutieren, was teilweise auf den Prozesslandkarten und mit Hilfe der Objektlandkarte geschehen ist.

### 5.2.5.2 Prozesslandkarten

Im Anschluss an die Validierung und Ergänzung der Feature-Landkarte wurde mit dem Skizzieren der Prozesslandkarten fortgefahren. Dazu wurde die zeitlich-logische Abfolge des Auftretens der einzelnen Epics und Features in den Geschäftsprozessen verwendet. Die Prozesse wurden mittels der Prozesslandkarten beschrieben. Begonnen wurde mit dem Epic „Lead erfassen“ (vgl. Abbildung 20).

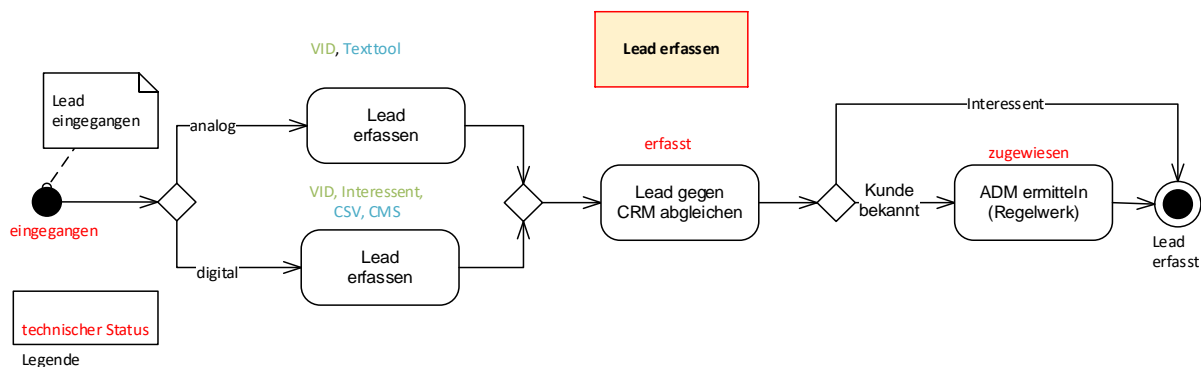


Abbildung 20: Prozesslandkarte „Lead erfassen“

Eingehende Leads sollen den technischen Status „eingegangen“ erhalten. Sofern ein Lead analog eingeht, soll es vom Vertriebsinnendienst (VID) in einem Texttool erfasst werden. Falls es digital eingeht, sollen die Daten vom VID per CSV-Datei eingelesen oder direkt über Landing Pages des Content-Management-Systems (CMS) von Interessenten in ein Webformular eingetragen werden. Im Anschluss soll gegen CRM geprüft werden, ob das Lead bekannt ist. Danach wird der Lead-Status auf „erfasst“ gesetzt. Sofern das Lead bekannt ist, soll der betreuende ADM über ein Regelwerk ermittelt und das Lead in den Zustand „zugewiesen“ überführt werden. Andernfalls soll das Lead im Zustand „erfasst“ verbleiben. In beiden Fällen soll der Prozess danach beendet sein.

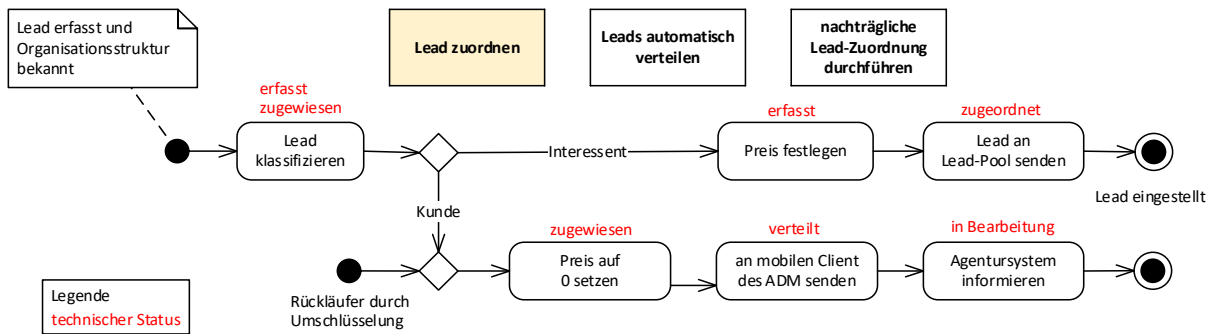


Abbildung 21: Prozesslandkarte „Leads zuordnen“

Nachdem ein Lead erfasst wurde, soll dieses weiterverarbeitet werden (vgl. Abbildung 21), wofür die Organisationsstruktur aus dem Provisionssystem vorliegen muss. Im ersten Schritt soll das Lead klassifiziert werden, indem es anhand des Produktinteresses in eine Kategorie eingeteilt wird. Die Klassifikation des Leads soll sowohl für Interessenten als auch für Kunden erfolgen. Sofern das Lead ein Interessent ist, soll der Preis entsprechend des Ergebnisses der Klassifikation ermittelt werden. Nachdem der Preis festgelegt ist, soll das Lead in den zugeordneten Lead-Pool gestellt werden. Ab diesem Zeitpunkt ist das Lead im Status „zugeordnet“. Handelt es sich bei dem Lead um einen Kunden, soll der Preis automatisch auf null gesetzt werden. Es soll eine Benachrichtigung an eine mobile Anwendung gesendet werden, durch die der zuständige ADM Kenntnis über die Anfrage des Kunden erhalten soll. Durch das Versenden der Nachricht soll sich der Status des Leads auf „verteilt“ ändern, wodurch der Hauptverwaltungsdienst erkennen kann, dass das Lead verteilt wurde, falls der ADM aus technischen Gründen keine Benachrichtigung erhalten haben sollte. Nachdem die Benachrichtigung an die mobile Anwendung gesendet wurde, soll das Agentursystem benachrichtigt werden, woraufhin die Kundenanfrage durch den ADM bearbeitet werden kann. Das Lead soll anschließend in den Zustand „in Bearbeitung“ überführt werden.

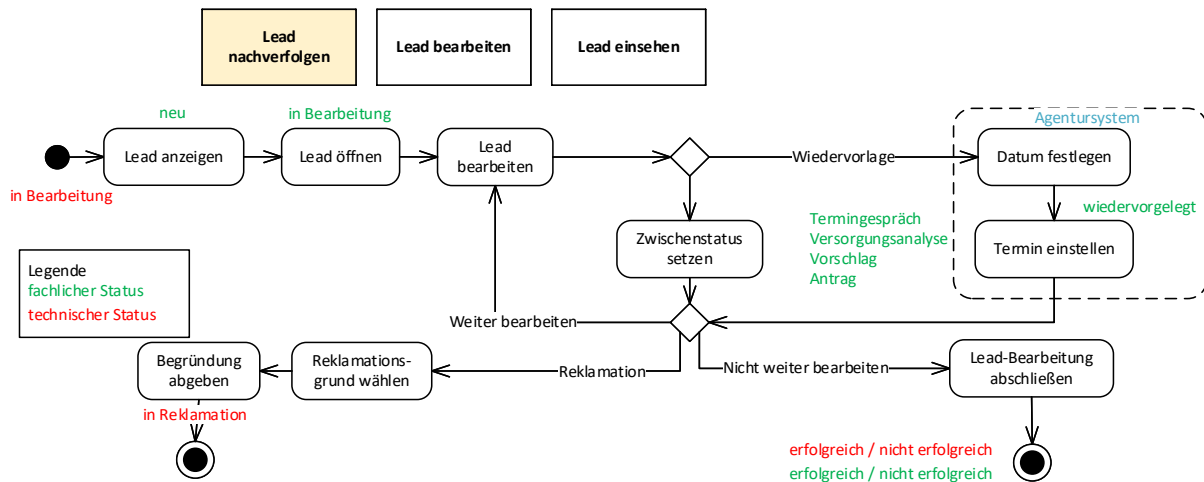


Abbildung 22: Prozesslandkarte „Leads nachverfolgen“

Damit ein ADM einen Lead bearbeiten und nachvollziehen kann, in welchem Bearbeitungsstatus sich das Lead befindet, soll es zusätzlich zu den technischen Status auch fachliche Status geben. Aktuell wird der fachliche Status im Agentursystem gepflegt: Neue Leads werden durch den fachlichen Zustand „neu“ als solche gekennzeichnet. Sobald der ADM das Lead öffnet, wechselt das Lead in den fachlichen Zustand „in Bearbeitung“. Von dort aus, in Abhängigkeit der fachlichen Tätigkeiten des ADM, kann das Lead unterschiedliche fachliche Zwischenstatus einnehmen („Termingespräch“, „Versorgungsanalyse“, „Vorschlag“, „Antrag“). Der fachliche Zwischenstatus eines Leads kann sich beliebig oft wiederholen. Ein besonderer Status eines Leads ist die Wiedervorlage. Um ein Lead auf Wiedervorlage zu legen, soll der ADM weiterhin die Oberfläche des Agentursystems benutzen: Er wählt das Datum der Wiedervorlage und das Agentursystem erstellt die Wiedervorlage. Bis der Zeitraum der Wiedervorlage abgeschlossen ist oder der fachliche Status vom ADM verändert wird, verweilt das Lead im fachlichen Status „wiedervorgelegt“. Damit die Bearbeitung des Leads abgeschlossen werden kann, muss entweder ein Vertrag zu Stande gekommen sein oder es wird entschieden, dass das Lead nicht weiterbearbeitet werden soll. Wenn ein Vertrag geschlossen wurde, muss der ADM, zusätzlich zum Wechsel des Status in „erfolgreich“, die Versicherungsnummer (VNR) der geschlossenen Verträge angeben. Durch den Status und die Zuordnung zu Verträgen soll eine Erfolgsauswertung möglich werden. Nach dem Abschieken des Status und der VNR wechselt das Lead in den technischen Status „erfolgreich“. Wenn ein Lead nicht weiterbearbeitet wird, setzt der ADM den fachlichen auf „nicht erfolgreich“. Danach wird der technische Status auf „nicht erfolgreich“ gesetzt und das Lead wird gelöscht. Ein ADM kann einen Lead reklamieren, dazu ist ein

Reklamationsgrund erforderlich sowie eine zusätzliche Begründung durch den ADM, der technische Status ändert sich auf „in Reklamation“.

### 5.2.5.3 Integrationslandkarte

In der Ausgangslage und am Beispiel der Prozesslandkarte wurden bereits erste Schnittstellen identifiziert. Auf Grundlage aller skizzierten Prozesse wurde im Workshop eine Integrationslandkarte erstellt, die in Abbildung 23 dargestellt ist.

Das Lead System soll aus zwei Teilen bestehen: Eine Webanwendung soll für einen Großteil der Funktionalität implementiert werden und ein Backend zur Administration bieten (z. B. zur Verwaltung von Sperren von Postleitzahlen für ADMs). In der mobilen Anwendung soll der Lead-Kauf implementiert werden. Leads werden in einer externen Komponente erfasst und danach in das LS überspielt. Der Abgleich zwischen Leads und Kunden soll zwischen der Erfassungskomponente und dem CRM geschehen. Um den Datenbestand des CRMs konsistent zu halten, muss ein Abgleich gelöschter Leads mit dem LS implementiert werden. Sofern ein Lead nicht als Kunde klassifiziert wurde, wird er als Interessent vom CRM an das Agentursystem überspielt. Das LS und das Agentursystem halten Leads synchron. Leads werden in das Data Warehouse überspielt und der Saldo des Lead-Kontos wird im Batch mit dem Verrechnungskonto verrechnet. Die Organisationsstruktur wird vom Provisionssystem zugeliefert. Gutscheine werden extern erzeugt und in das LS importiert.

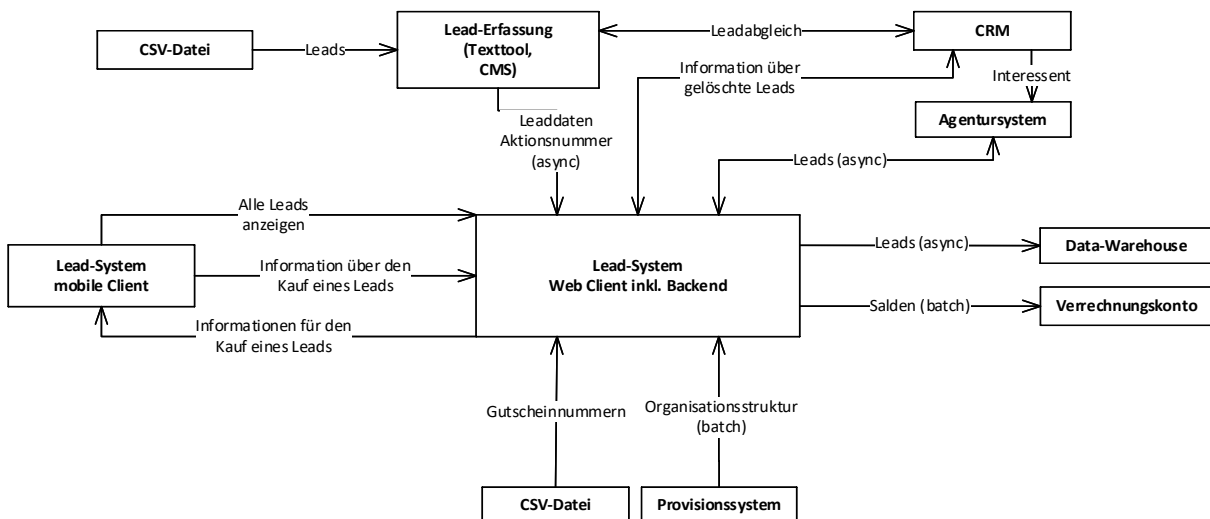


Abbildung 23: Integrationslandkarte des Praxisbeispiels



von Aktivitäten. Durch das Notieren von Systemen für Aktivitäten können technische Schnittstellen zwischen Systemen abgeleitet werden, die wiederum in der Integrationslandkarte explizit veranschaulicht werden. Darüber hinaus kann über Rollen an Aktivitäten ein Teil eines Berechtigungskonzeptes abgeleitet werden und es können Akteure an Anwendungsfällen identifiziert werden.

Außerdem wurden technische und fachliche Status des zentralen Geschäftsobjekts immer dann an Elementen der Prozesslandkarte vermerkt, wenn sich die Status verändert haben. Dieser Umstand wurde durch die Anforderung, Leads nachverfolgen zu können, ausgelöst. Die Nachverfolgung ist fachlich motiviert, die ADMs sollen die Bearbeitung über Status steuern können. Zur Beauskunftung und Anwendungssteuerung wurden technische Status der Leads über alle Prozesse identifiziert, die den Entwurf einer zustandsbasierten Prozess- und Anwendungssteuerung ermöglichen.

Auf der Integrationslandkarte stellt nicht jedes Umsystem ein Informationssystem dar. Leads können in das Texttool und Gutscheine in das zentrale Informationssystem importiert werden. Die Darstellung dieser Imports wurde der Vollständigkeit halber dokumentiert. Zudem stellt der Import in das Texttool eine Schnittstelle dar, die nicht zwischen dem zentralen Informationssystem und einem Umsystem besteht, sondern zwei Umsysteme miteinander verbindet. Dies ist syntaktisch nicht korrekt (vgl. Kapitel 5.2.3.1), hat in diesem Fall jedoch durch die Vollständigkeit des Systemkontextes zum gemeinsamen Verständnis beigetragen und wurde deswegen zugelassen. Gleiches gilt für die Schnittstelle zwischen dem CRM und dem Agentursystem sowie für die Lead-Erfassung und das CRM.

### 5.3 Interaction-Room-Annotationen

Auf den Interaction-Room-Landkarten werden die funktionalen Anforderungen an ein Informationssystem dokumentiert. Die Landkarten bauen auf etablierten Notationen auf und ermöglichen die Beschreibung von Abläufen und Strukturen. Um die Herausforderungen bei der Entwicklung von Informationssystemen (vgl. Kapitel 2.1) zu kennzeichnen, werden die Interaction-Room-Annotationen (kurz: Annotationen) eingesetzt. Mit den Annotationen kann ausufernder Diskussion bereits ausreichend verstandener Details vorgebeugt [207] und der Fokus auf die erfolgskritischen Aspekte eines Projektes gelegt werden. Durch das Kennzeichnen von Werttreibern, Aufwandstreibern (Qualitätsanforderungen und Rahmenbedingungen) und Risikotreibern können wichtige von weniger wichtigen funktionalen Anforderungen unterschieden

werden. Ziel ist es nicht, alle Werte, Qualitätsanforderungen, Rahmenbedingungen und Risiken zu identifizieren, sondern nur überdurchschnittliche Herausforderungen, die andernfalls implizit geblieben wären. Überdurchschnittlich bedeutet in diesem Zusammenhang, dass die Herausforderung nicht offensichtlich, nicht trivial und im Vergleich zu anderen Anforderungen mehr Relevanz für den Projekterfolg besitzt. Die Beurteilung überdurchschnittlicher Herausforderungen obliegt der Verantwortung von Stakeholdern eines Interaction-Room-Workshops.

In Kapitel 5.3.1 wird das Konzept der Interaction-Room-Annotationen eingeführt. Ausgehend vom Konzept der Annotationen, werden verwandte Ansätze diskutiert (vgl. Kapitel 5.3.2) und die Annotationskategorien vorgestellt (vgl. Kapitel 5.3.3). Ausgewählte Annotationstypen werden hergeleitet, beschrieben (vgl. Kapitel 5.3.4) und das Vorgehen zu deren Verwendung wird erläutert (vgl. Kapitel 5.3.5). In Kapitel 5.3.6 wird das Praxisbeispiel aus Kapitel 5.2.5 aufgegriffen und um Annotationen ergänzt.

### 5.3.1 Konzept der Annotationen

Abbildung 25 zeigt das Metamodell der Interaction-Room-Annotationen.

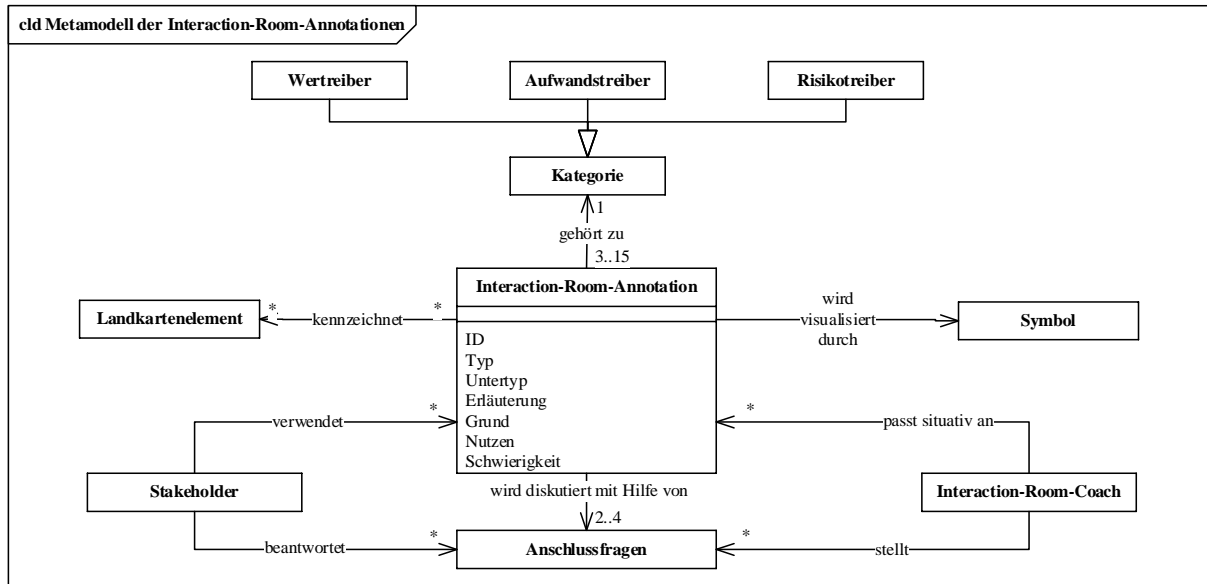


Abbildung 25: Metamodell der Interaction-Room-Annotationen

Eine Annotation besitzt eine ID, einen Typ, einen Untertyp, eine Erläuterung, einen Grund, sowie einen Nutzen und eine Schwierigkeit. Der Annotationstyp wird durch ein Symbol visualisiert. Der Annotationstyp, die ID und der Grund definieren eine Instanz einer Interaction-Room-Annotation, die Annotation heißt. Ein Annotationstyp ist entweder der Kategorie Wertreiber, Aufwandstreiber oder Risikotreiber zugeordnet und wird von einem Stakeholder zur



Kennzeichnung eines Landkartenelements verwendet. Ein Stakeholder beantwortet Anschlussfragen, um den Grund einer Annotation und die daraus entstehende Anforderung zu formulieren. Anschlussfragen werden vom Interaction-Room-Coach gestellt. Die ID ist die Referenz zwischen Annotationstyp auf einer Landkarte und dem dokumentierten Grund. Die Erläuterung eines Annotationstyps kann durch einen IR-Coach situationsspezifisch an ihren Einsatzzweck angepasst werden. Ein Annotationstyp kann durch einen Untertyp konkretisiert werden. Der Nutzen einer Annotation tritt ein, sofern die gestellte Anforderung umgesetzt wird. Dieser Nutzen wird in T-Shirt-Größen qualifiziert, also in die Kategorien small, medium, large oder extra large (S, M, L, XL). Die Eigenschaft Schwierigkeit drückt die erwartete Schwierigkeit aus, die bei der Umsetzung der geforderten Anforderung entstehen würde und wird ebenfalls in T-Shirt-Größen qualifiziert. Die Höhe der zu bewältigenden Schwierigkeit macht einen Teil des zu leistenden Aufwandes für die Umsetzung einer Anforderung aus. Auf die Frage nach der Höhe des Aufwandes wird jedoch verzichtet, um den Anchoring-Effekt [208] einer spontanen Schätzung zu vermeiden.

### 5.3.2 Verwandte Ansätze

Die Idee, Modelle mit Zusatzinformationen zu versehen, ist nicht neu. Einige Ansätze konzentrieren sich darauf, bestimmte Qualitätsmerkmale zu modellieren. Bocciarelli und D'Ambrogio [209] erweitern die BPMN durch die Qualitätsmerkmale Performanz und Zuverlässigkeit. Saeedi, Zhao und Falcone Sampaio [210] erweitern die BPMN, um Zeit, Kosten und Zuverlässigkeit an Aktivitäten modellieren zu können, um geeignete Services auszuwählen und Service Level Agreements erstellen können. Gagne und Trudel [211] erweitern die BPMN zur Modellierung temporaler Eigenschaften von Geschäftsprozessen. Rodriguez, Fernandez-Medina und Piattini erweitern die BPMN und UML-Aktivitätsdiagramme, um Sicherheitsanforderungen modellieren zu können [212], [213]. Korherr und List [214] erweitern ereignisgesteuerte Prozessketten (EPKs), die BPMN und UML-Aktivitätsdiagramme, um die durchschnittlichen und maximalen Kosten, die durchschnittliche und maximale Durchlaufzeit und die durchschnittliche und maximale Anzahl Beschwerden an Funktionen, Knoten und Aktivitäten quantifizieren zu können. Dadurch messen sie die Performanz und die Benutzerzufriedenheit je Notationselement. Jensen und Feja [215] erweitern EPKs, um die Sicherheitsaspekte Zugriffskontrolle, Vertraulichkeit und Integrität modellieren zu können.

Alle zuvor genannten Ansätze erweitern mindestens eine zu Grunde liegende Modellierungssprache um spezifische Aspekte (Performanz, Zuverlässigkeit, Sicherheit,

Benutzerzufriedenheit), lassen andere interessante Aspekte jedoch außer Acht. Die Erweiterungen sind für eine bestimmte Notation entworfen, erweitern deren Metamodelle und sind ausschließlich für Ablaufdiagramme konzipiert, Strukturdiagramme werden nicht berücksichtigt. Die Interaction-Room-Annotationen folgen den zuvor genannten Ansätzen, da mit Annotationen Anforderungen an die Performanz, die Zuverlässigkeit, die Sicherheit und die Benutzerzufriedenheit in Ablaufdiagrammen gestellt werden können. Zusätzlich lassen sich weitere Anforderungen mit Hilfe der Annotationen formulieren, sie können nicht nur in Ablaufdiagrammen, sondern auch in Strukturdiagrammen verwendet werden und erfordern für ihren Einsatz keine Erweiterung einer Notation, sondern sind für die Verwendung in informellen Modellskizzen konzipiert. Das Konzept der Annotationen ermöglicht, diese auch mit der BPMN, der UML, EPKs und anderen Notationen zu verwenden.

Zou und Pavlovski zeigen mit Control Case einen Ansatz, um nicht funktionale Anforderungen an die Dienstgüte von Anwendungsfällen zu stellen. Die nicht funktionalen Anforderungen werden durch Ausführungsbedingungen und einem dazugehörigen Control Case erfasst. Die Ausführungsbedingungen (z. B. Last, Bandbreite oder Netzwerkverfügbarkeit) definieren unter welchen Parametern die Funktionalität des Systems operieren können muss. Ein zugehöriger Control Case beschreibt das Risiko, das durch Verfehlen des Qualitätsziels entsteht, und dessen Eintrittswahrscheinlichkeit. Um Ausführungsbedingungen und Control Cases an Anwendungsfällen zu modellieren erweitern die Autoren das Metamodell der UML-Use-Case-Diagramme (vgl. zum Vorherigen [216], [217]). Die gleichen Autoren wenden ihren Control-Case-Ansatz auf die BPMN an, indem sie eine prozessrelevante Teilmenge von Ausführungsbedingungen verwenden. Um Aktivitäten mit Ausführungsbedingungen zu annotieren, verwenden die Autoren das Konzept der Assoziation der BPMN [218].

Der Control-Case-Ansatz ermöglicht die Identifikation und Dokumentation von nicht funktionalen Anforderungen und Risiken bei Anwendungsfällen und in Geschäftsprozessen. Durch die Verwendbarkeit in Use-Case-Diagrammen kann dieser Ansatz in mehr Diagrammtypen eingesetzt werden als die zuvor genannten Ansätze. Die Interaction-Room-Annotationen sind nicht explizit für die Verwendung in Use-Case-Diagrammen konzipiert, sie könnten jedoch auch zur Kennzeichnung überdurchschnittlicher Herausforderungen in Use Cases verwendet werden. Außerdem werden Erkenntnisse aus dem Interaction Room in Use-Case-Diagramme überführt (vgl. Kapitel 5.4.2). Während Zou und Pavlovski den Control-Case-Ansatz zur Spezifikation der Dienstgüte konzipiert haben, sind die Annotationen nicht nur für die Spezifikation von Ausführungsbedingungen konzipiert, sondern decken mehr Qualitätsanforderungen und

Rahmenbedingungen ab. Control Cases werden, im Gegensatz zu den Annotationen, nicht in Strukturdiagrammen verwendet.

Hogganvik und Stølen präsentieren eine eigene grafische Notation, um Bedrohungen zu identifizieren, zu diskutieren und zu dokumentieren. Die Syntax baut auf dem UML-Profil zur Risikobewertung [219] auf. Die Pragmatik beinhaltet fünf Schritte. Im ersten Schritt werden die schützenswerten Artefakte identifiziert, um daraufhin durch Brainstorming in interdisziplinären Teams Bedrohungen im zweiten Schritt zu identifizieren. Im dritten Schritt werden die Frequenz des Auftretens der Bedrohungen und die resultierenden Konsequenzen in den Kategorien klein, mittel und groß geschätzt. Anschließend werden die Bedrohungen im vierten Schritt modelliert, um einen Überblick der Bedrohungen und ihrer Interdependenzen zu veranschaulichen. Im abschließenden fünften Schritt werden Maßnahmen zur Behandlung der Bedrohungen identifiziert und modelliert. Dies geschieht durch ein weiteres Modell, in dem Behandlungsoptionen dargestellt werden. Die Optionen werden im Hinblick auf ihre Wirksamkeit beurteilt, um daraus eine Strategie zu entwickeln, die Bedrohungen Kosten-Nutzen-optimiert zu behandeln (vgl. zum Absatz [220]).

Die Interaction-Room-Annotationen folgenden dem Ansatz von Hogganvik und Stølen in Teilen, denn die Annotationstypen werden grafisch durch Symbole repräsentiert. Die Kennzeichnung, Dokumentation und Kommunikationsunterstützung erfolgt ebenfalls in interdisziplinären Teams. Wie beim Ansatz von Hogganvik und Stølen wurde für die Annotationen eine Vorgehensweise zu deren Verwendung definiert (vgl. Kapitel 5.3.5). Im Ansatz der beiden Autoren werden Risiken durch das Eintreten von Bedrohungen in klein, mittel oder groß qualifiziert. Für die Annotationen werden der Nutzen und die Schwierigkeit jeweils in small, medium, large oder extra large qualifiziert. Anders als beim Ansatz von Hogganvik und Stølen können mit den Annotationen mehr Aspekte identifiziert und gekennzeichnet werden als Bedrohungen.

### 5.3.3 Annotationskategorien

Die Interaction-Room-Annotationen sind in die drei Kategorien Werttreiber, Aufwandstreiber und Risikotreiber unterteilt. Jede Kategorie adressiert mindestens eine der Herausforderungen für die Entwicklung von Informationssystemen (vgl. Kapitel 2.1).

### 5.3.3.1 Werttreiber

Zunehmender Wettbewerbsdruck, dynamische Märkte und technologischer Fortschritt sind Herausforderungen, denen sich Unternehmen stellen müssen, um im globalen Wettbewerb bestehen zu können. Dazu ist eine Maximierung der Wertschöpfung durch die Befriedigung von Markterfordernissen und Kundenbedürfnissen essentiell. Damit Werttreiber identifiziert werden können, müssen Kundenanforderungen, Geschäftsanforderungen und technologische Möglichkeiten analysiert und bewertet werden. Daraufhin müssen entsprechende Entscheidungen getroffen werden, die größten Werttreiber umzusetzen und so die Wertschöpfung zu maximieren (vgl. zum Absatz [94]).

Eine zentrale Aufgabe im Requirements Engineering ist die Ermittlung von Anforderungen, die die Wertschöpfung einer Organisation maximieren [95]. Kernprozesse weisen größeren Anteil an der Wertschöpfung einer Organisation auf als Unterstützungsprozesse: ein Rechenkern für die Simulation von Marktszenarien einer Bank liefert einen größeren Beitrag zur Wertschöpfung als eine Komponente zur Verwaltung von Räumlichkeiten. Oberflächen, die regelmäßig von vielen Nutzern bedient werden, müssen die Erwartungen der Nutzer befriedigen, um die Akzeptanz des Systems nicht zu gefährden, wohingegen der Wertbeitrag softwareergonomisch optimierter Masken, die selten genutzt werden, geringer ausfällt. Um wertschöpfende Anforderungen zu identifizieren wird eine multiperspektivische Betrachtung der Anforderungen benötigt, denn unterschiedliche Stakeholder besitzen unterschiedliche Blickwinkel. Ergebnis dieser Betrachtung muss eine kritische Auseinandersetzung mit Anforderungen sein, die zu einer entsprechenden Priorisierung führen muss.

Die Annotationstypen der Kategorie Werttreiber werden zur Kennzeichnung von Anforderungen verwendet, die einen größeren Beitrag zur Wertschöpfung leisten als andere. Werttreiber können direkten oder indirekten Nutzen für eine Organisation besitzen (z. B. durch die Identifikation von Schlüsselanforderungen oder durch die Identifikation von imagefördernden Aspekten).

### 5.3.3.2 Aufwandstreiber

Aufwandstreiber kennzeichnen Aspekte, deren Umsetzung einen Mehraufwand gegenüber der ausschließlichen Umsetzung funktionaler Anforderungen bedeutet [221]. Deshalb sollte deren Nutzen größer als der zu leistende Mehraufwand sein. Die Identifikation von Aufwandstreibern ist eine weitere zentrale Aufgabe im Requirements Engineering, die insbesondere zu Beginn eines Projektes wichtig ist, da sie die Softwarearchitektur und das Systemdesign beeinflussen

[221], [222]. Aufwandstreiber sind schwieriger zu identifizieren, abzustimmen, zu klassifizieren und zu dokumentieren als funktionale Anforderungen, sie werden oft nicht explizit gemacht, was weitreichende Folgen auf Projekterfolg haben kann [217], [223], [224]. Aufwandstreiber setzen sich aus Qualitätsanforderungen und Rahmenbedingungen zusammen.

Ebert stellt fest, dass Qualitätsanforderungen häufig übersehen und nicht ausreichend spezifiziert werden, da diese „sehr viel schwieriger“ [225] zu implementieren sind als funktionale Anforderungen. Ein qualitativ hochwertiges System, welches funktional unvollständig ist, führt zu einer größeren Akzeptanz der Nutzer als funktionale Vollständigkeit mit mangelhafter Qualität [12], [222].

Abbildung 26 zeigt die acht Qualitätsmerkmale und zugehörigen Teilmerkmale einer Software nach ISO/IEC 25010, die auf dem ISO/IEC-Standard 9126 aufbauen [226]. Die Qualitätsmerkmale bzw. Teilmerkmale werden nicht vollständig durch die Annotationstypen abgedeckt, denn die Intention der Annotationen ist nicht deckungsgleich mit den Qualitätsmerkmalen nach ISO 25010. Während der ISO-Standard darauf abzielt, ein Softwareprodukt im Hinblick auf die Konformität zu den Qualitätsmerkmalen bzw. Teilmerkmalen bewerten zu können [201], werden mit den Annotationstypen Qualitätsanforderungen gekennzeichnet, die während der Konzeption und Entwicklung zu berücksichtigen sind. Die Kennzeichnung durch Annotationstypen zielt nicht auf Vollständigkeit ab und bezieht sich immer auf bestimmte funktionale Anforderungen, da sie Elemente der Landkarten kennzeichnen. Mit den Annotationstypen werden also keine Qualitätsmerkmale gekennzeichnet, die sich auf das gesamte System beziehen. In der Herleitung und Beschreibung einzelner Annotationstypen (vgl. Kapitel 5.3.4 und Anhang C) wird Bezug zum ISO-Standard genommen, wo dies möglich ist.

Die Rahmenbedingungen der Kategorie Aufwandstreiber müssen bei der Konzeption und Implementierung funktionaler Anforderungen berücksichtigt, jedoch nicht aktiv umgesetzt werden. Neben gesetzlichen, organisatorischen und technischen Randbedingungen zählen zu dieser Kategorie auch Annotationstypen, die sich auf bereits bestehende Artefakte beziehen, beispielsweise die Unveränderlichkeit oder der Verbesserungsbedarf. Aus der Identifikation solcher Rahmenbedingungen ergeben sich weitere Anforderungen, die proaktiv umgesetzt werden müssen: Die Anbindung externer Ressourcen bedingt die Implementierung bestimmter Schnittstellen sowie die Einschätzung der Stabilität der Schnittstellen und der zu erwartenden Qualität gelieferter Daten. Die Notwendigkeit manueller Bearbeitung von Prozessschritten bedingt die Optimierung von Oberflächen im Hinblick auf die Verständlichkeit und die Zuverlässigkeit im Sinne der Robustheit gegenüber Fehleingaben.

<b>Funktionalität</b>	<b>Effizienz</b>	<b>Kompatibilität</b>	<b>Benutzbarkeit</b>
funktionale Vollständigkeit funktionale Richtigkeit funktionale Angemessenheit	Zeitverhalten Verhaltensverhalten Kapazität	Koexistenz Interoperabilität	Verständlichkeit Erlebarkeit Bedienbarkeit Fehlervermeidung Attraktivität Zugänglichkeit
<b>Zuverlässigkeit</b>	<b>Sicherheit</b>	<b>Wartbarkeit</b>	<b>Portierbarkeit</b>
Reife Verfügbarkeit Fehlertoleranz Wiederherstellbarkeit	Vertraulichkeit Integrität Nachweisbarkeit Verantwortlichkeit Authentizität	Modularität Wiederverwendbarkeit Analysierbarkeit Modifizierbarkeit Testbarkeit	Anpassbarkeit Installierbarkeit Austauschbarkeit

Abbildung 26: Softwarequalitätsmerkmale nach ISO 25010 (angelehnt an [227])

### 5.3.3.3 Risikotreiber

In frühen Phasen von Softwareprojekten ist es natürlich, dass einige Aspekte besser verstanden sind als andere. Gut verstandene Aspekte können detaillierter beschrieben werden als weniger gut verstandene Aspekte. Dies kann zu einer Verschiebung der wahrgenommenen Komplexität führen: Wenn gut verstandene Aspekte in Anforderungsdokumenten überwiegen, besteht die Möglichkeit, dass diese präferiert umgesetzt werden. Der scheinbar zu leistende Aufwand erscheint hierbei größer und ist durch die detaillierte Beschreibung einfacher umzusetzen. Die Gefahr liegt in der Natur des geringen Verständnisses: Es können unerwartete Aufwände entstehen, die sich erhöhen, wenn bereits Entscheidungen getroffen worden sind, deren Umsetzung nicht kompatibel zu den bislang unbekanntem Aspekten ist (z. B. Architekturentscheidungen). Auch inhärente Risiken der weniger gut verstandenen Artefakte stellen eine Gefahr dar: Risiken werden nicht rechtzeitig erkannt, bewertet und durch geeignete Maßnahmen behandelt.

Um unterscheiden zu können, ob Elemente der Landkarten des Interaction Rooms gut verstanden sind und keine unbekanntem Risiken bergen, ist eine Kennzeichnung gut verstandener oder weniger gut verstandener Elemente notwendig. Genauso ist es notwendig, die Anforderungen zu identifizieren, deren Konzeption und Umsetzung nicht offensichtliche fachliche oder technische Komplexität birgt.

### 5.3.4 Herleitung und Beschreibung ausgewählter Annotationstypen

In diesem Kapitel werden einzelne Annotationstypen vorgestellt. Dazu wird motiviert warum ein Annotationstyp relevant ist und daraufhin mit Hilfe einer Vorlage beschrieben (vgl. Tabelle 4).

Tabelle 4: Vorlage zur Beschreibung von Annotationstypen

<Symbol>	Annotationskategorie:
	Untertyp(en):
	Verwendung auf: <input type="checkbox"/> Prozesslandkarte <input type="checkbox"/> Objektlandkarte <input type="checkbox"/> Integrationslandkarte
Erläuterung	
Anschlussfragen	

Ein Annotationstyp wird durch ein Symbol repräsentiert und ist einer Kategorie zugeordnet. Annotationstypen können Untertypen besitzen, durch die sie konkretisiert werden. Die textuelle Erläuterung zeigt, was mit dem Annotationstypen gekennzeichnet werden kann und welche Konsequenz daraus entsteht. Die Anschlussfragen dienen dem Erfragen der Begründung des Befürworters der Annotation. Die Annotationstypen sind für die Verwendung auf bestimmten Landkarten vorgesehen.


In den folgenden Unterkapiteln werden ausgewählte Annotationen der Kategorien Werttreiber, Aufwandstreiber und Risikotreiber vorgestellt. In Kapitel 5.3.4.8 wird ein Überblick über alle Annotationstypen unter Angabe ihrer Kategorie und der Landkarten, auf denen sie verwendet werden können, gegeben. Die Herleitung und Beschreibung aller 20 Annotationstypen befindet sich in Anhang C.

#### 5.3.4.1 Annotationstyp User Value

Funktionalität zu entwickeln, die nicht die Bedürfnisse der Nutzer erfüllt, ist ein Projektrisiko [222], [225], [228], [229], denn nur wenn Nutzer einen Wertzuwachs durch Funktionalität und Qualität eines Informationssystems erkennen, akzeptieren sie dieses [230]. Der Wert für einen Nutzer muss nicht mit dem Wert für eine Organisation korrelieren, jedoch sollte ein bestmöglicher Kompromiss gefunden werden, um alle Stakeholder eines Systems zufriedenzustellen [230]. Der Wert für Nutzer wird in Basisfaktoren, Leistungsfaktoren und Begeisterungsfaktoren klassifiziert [231]. Basisfaktoren sind Anforderungen, die zu Inakzeptanz eines Systems führen, wenn diese fehlen. Die Umsetzung von Basisfaktoren führt jedoch nicht unmittelbar zu Akzeptanz [221], sondern wird als selbstverständlich vorausgesetzt. Leistungsfaktoren werden von Nutzern explizit gefordert, deren Umsetzung führt zu Zufriedenheit mit dem System und stiftet großen Nutzen [ebenda]. Begeisterungsfaktoren sind dem Nutzer nicht bekannt, stellen sich aber als angenehme und nützliche Überraschung heraus und führen zu Begeisterung [ebenda]. Diejenigen Prozess- und Systemteile, die überdurchschnittlich zur Zufriedenheit der späteren

Nutzer eines Systems beitragen und somit dessen Akzeptanz fördern, werden mit dem Annotationstyp User Value gekennzeichnet (vgl. Tabelle 5).


Tabelle 5: Annotationstyp User Value

	Annotationskategorie: Werttreiber
	Untertyp(en): Basisfaktor, Leistungsfaktor, Begeisterungsfaktor
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
Erläuterung	
Der Annotationstyp User Value wird verwendet, um Landkartenelemente zu kennzeichnen, die einen Mehrwert für Anwender liefern. Der Annotationstyp zeigt, dass der Entwurf und die Implementierung dieser Prozess- und Systemteile in besonderem Maße zur Zufriedenheit der Nutzer und deren Akzeptanz des Systems beitragen.	
Anschlussfragen	
<ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil hat besonderen Einfluss auf den User Value?</li> <li>• Was ist die Erwartungshaltung an diesen Teil? (Basis-/Leistungs-/Begeisterungsfaktor)</li> <li>• Was ist erforderlich, um einen positiven Einfluss zu erreichen bzw. negativen Einfluss zu vermeiden?</li> </ul>	

#### 5.3.4.2 Annotationstyp Hohe Last

Das Verbrauchsverhalten ist ein Teilmerkmal der Effizienz nach ISO 25010 und bezeichnet die „Fähigkeit des Softwareprodukts, eine angemessene Anzahl und angemessene Typen von Ressourcen zu verwenden, wenn die Software ihre Funktionen unter festgelegten Bedingungen ausführt“ [201]. Lastanforderungen können mit ausreichenden Ressourcen oder skalierbaren Infrastrukturen erfüllt werden. Sofern Lastanforderungen Schwankungen unterliegen, ermöglichen elastische Infrastrukturen lastgerechtes, automatisches Provisionieren und Deprovisionieren von Ressourcen [22]. Die Konsequenz aus der Identifikation funktionaler Anforderungen, die besondere dauerhafte oder temporäre Lastanforderungen stellen, muss eine entsprechende Softwarearchitektur und Struktur des Quellcodes sein. Sofern den Lastanforderungen durch die Auslagerung von Systemteilen in die Cloud genügt werden soll, muss die Softwarearchitektur durch geeigneten Komponentenschnitt auf unterschiedlichen Infrastrukturkomponenten lauffähig sein [232]. Mit dem Annotationstyp Hohe Last werden Prozess- und Systemteile gekennzeichnet, die besonderen Lastanforderungen genügen müssen (vgl. Tabelle 6).

Tabelle 6: Annotationstyp Hohe Last

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): dauerhaft, temporär
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
Erläuterung	



Der Annotationstyp Hohe Last wird verwendet, um Landkartenelemente zu kennzeichnen, die Lastanforderungen genügen müssen. Der Annotationstyp zeigt, dass bei der Konzeption und Implementierung entsprechende fachliche oder technische Maßnahmen vorgesehen werden müssen, um dauerhaften oder temporären Lastanforderungen gerecht werden zu können.


#### Anschlussfragen

- Welcher System-/Prozessteil ist voraussichtlich hoher Last ausgesetzt?
- Welcher Art ist die Last? Wann, wie oft und wie lange wird sie erwartet?
- Welche Auswirkungen hätte eine Überlastung?

### 5.3.4.3 Annotationstyp Flexibilität

Durch technologische Innovation, fachliche Weiterentwicklung und Markterfordernisse steigt der Bedarf, Informationssysteme flexibel zu gestalten [233]. Rasche und zuverlässige Wartung und Weiterentwicklung sind kostenintensiv [234], [235] und eine der größten Herausforderungen im Software Engineering [236]. Informationssysteme können nach dem Prinzip „Design for Chance“ entwickelt werden, um deren Wartung und Weiterentwicklung schon bei der Konzeption einzuplanen sowie die Struktur und den Schnitt der Anwendung auf zukünftige Änderbarkeit auszulegen [237]. Flexibilitätsanforderungen umzusetzen, verursacht Aufwand, weswegen die Anwendungsteile identifiziert werden müssen, deren Umsetzung nach dem Prinzip „Design for Change“ ein positives Kosten-Nutzen-Verhältnis besitzen. Neben Flexibilitätsanforderungen, die aus der Evolution von Software erwachsen, kann bereits zum Entwicklungszeitpunkt zukünftige Änderungsnotwendigkeit bekannt sein, jedoch ohne dass die konkrete Ausgestaltung der Änderung bekannt ist (z. B. durch die Anpassung von Gesetzen). Außerdem kann der zielgruppengerechte Funktionsumfang einer Software das Bilden von Varianten erfordern [238]. Die Konfiguration von Software bietet Nutzern einen gewissen Individualisierungsgrad und kann beispielsweise durch die Orchestrierung von Services (vgl. z. B. [239]) oder regel- bzw. fallbasierte Systeme [240] umgesetzt werden. Eine frühe Identifikation von flexiblen Prozess- und Systemteilen ist notwendig, um in der Softwarearchitektur dafür sowohl gemeinsam genutzte als auch variantenspezifische Komponenten vorsehen zu können. Für die Kennzeichnung flexibler Bestandteile existiert der Annotationstyp Flexibilität (vgl. Tabelle 7).

Tabelle 7: Annotationstyp Flexibilität

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Zukünftige Änderung, Anpassung zur Laufzeit, Variantenbildung
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
Erläuterung	

Der Annotationstyp Flexibilität wird verwendet, um Landkartenelemente zu kennzeichnen, bei denen bestimmte Flexibilitätsanforderungen vor der Konzeption, bei der Konzeption oder während der Laufzeit berücksichtigt werden müssen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung die Prinzipien Design for Change, Design for Customization oder Prinzipien zur Bildung von Varianten berücksichtigt werden müssen.


#### Anschlussfragen

- Welcher System-/Prozessteil soll flexibel ausgelegt sein?
- Welche Konfigurationen/Varianten/Systemanpassungen sind erforderlich?
- Welche Auswirkungen hätte Inflexibilität an dieser Stelle?

#### 5.3.4.4 Annotationstyp Manuelle Bearbeitung

Die erfolgreiche Ausführung bestimmter Aktivitäten in wissensintensiven Geschäftsprozessen ist abhängig von den kognitiven Fähigkeiten ihrer Bearbeiter, weil sie sich durch indeterministische Informationsquellen, Kollaboration unterschiedlicher Akteure, geringe Vorhersagbarkeit und geringere Struktur der Abläufe auszeichnen [241]. Die Ausführungspfade zur Zielerreichung dieser flexiblen Prozesse sind durch informelle Arbeitspraktiken geprägt. Individualisierungsanforderungen dieser Prozesse können die informationstechnische Unterstützung durch ein System teuer oder unmöglich machen, weil die Eigenschaften dieser Prozesse eine Komplexität erzeugen, die auch deren informationstechnische Unterstützung komplex werden lassen. Es ist also wichtig, Prozessschritte zu erkennen, die durch Menschen bearbeitet werden müssen, und es ist für diese zu prüfen, ob eine informationstechnische Unterstützung wirtschaftlich sinnvoll bzw. grundsätzlich möglich ist. Um Prozessteile zu kennzeichnen, deren Ausführung von den kognitiven Fähigkeiten ihres Bearbeiters abhängig ist, existiert der Annotationstyp Manuelle Bearbeitung (vgl. Tabelle 8)

Tabelle 8: Annotationstyp Manuelle Bearbeitung


	Annotationskategorie: Aufwandstreiber
	Untertyp(en): -
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input type="checkbox"/> Objektlandkarte <input type="checkbox"/> Integrationslandkarte
<p><b>Erläuterung</b></p> <p>Der Annotationstyp Manuelle Bearbeitung wird verwendet, um Landkartenelemente zu kennzeichnen, deren informationstechnische Unterstützbarkeit nicht sicher und die von den kognitiven Fähigkeiten des Bearbeiters abhängig sind. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung darauf zu achten ist, geeignete Mechanismen umzusetzen, die eine (teilweise) informationstechnische Unterstützung ermöglichen, oder geeignete organisatorische Maßnahmen zu ergreifen, um diesen Prozess ohne Informationstechnologie bearbeiten zu können.</p>	
<p><b>Anschlussfragen</b></p> <ul style="list-style-type: none"> <li>• Was erfordert menschliches Handlungs-/Entscheidungsvermögen?</li> <li>• Was sollte die Schnittstelle zwischen Mensch und System leisten, um die manuelle Bearbeitung möglichst gut zu unterstützen / integrieren?</li> <li>• Welche Auswirkungen hätten Fehler in der manuellen Bearbeitung?</li> <li>• Warum ist die manuelle Bearbeitung der Automatisierung vorzuziehen?</li> </ul>	

#### 5.3.4.5 Annotationstyp Randbedingung

Randbedingungen haben Einfluss auf die Umsetzung eines Informationssystems und können obligatorisch oder optional sein. Gesetzliche Vorschriften gehören zu den obligatorischen Randbedingungen, lassen jedoch Gestaltungsspielraum, denn maßgebend ist die Konformität zu Gesetzen und nicht, wie diese Konformität erreicht wird. Vorgaben des Betriebsrats oder Entscheidungen des Managements zählen, genau wie die Aufbau- und Ablauforganisation, zu Quellen organisatorischer Randbedingungen. Technische Randbedingungen wie die Vorgabe bestimmter Programmiersprachen, Technologie-Stacks, Datenbankmanagementsysteme oder die Empfehlung bestimmter Entwurfsmuster schränken die Konzeption und Implementierung eines Informationssystems ein. Jedoch sind gesetzliche, technische und organisatorische Randbedingungen nicht immer trennscharf, sondern können sich gegenseitig bedingen: Produktverantwortliche von Warenherstellern, die sowohl einen Wholesale- als auch Retail-Vertrieb bedienen, dürfen wegen des Kartellrechts keine Preisabsprachen treffen. Organisatorische und technische Vorschriften können zur Erfüllung der gesetzlichen Vorschrift beitragen, indem zum Beispiel die Produktverantwortlichen den Verkaufspreis vorgeben und die technisch unabhängige Implementierung der Bestellsysteme die Einhaltung der kartellrechtlichen Randbedingung sicherstellt. Randbedingungen sind also vielfältiger Natur, daher ist es wichtig, die kritischen Randbedingungen zu identifizieren und in deren Konzeption und Implementierung zu

investieren. Zur Kennzeichnung überdurchschnittlich relevanter Randbedingungen existiert der Annotationstyp Randbedingung (vgl. Tabelle 9).


Tabelle 9: Annotationstyp Randbedingung

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): gesetzlich, fachlich, organisatorisch, technisch
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Randbedingung wird verwendet, um Landkartenelemente zu kennzeichnen, dessen Entwurf und Implementierung durch gesetzliche, fachliche, organisatorische oder technische Randbedingungen eingeschränkt ist. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung die Konformität zu der Randbedingung sichergestellt sein muss.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil ist der Randbedingung unterworfen?</li> <li>• Welche Randbedingung ist zu beachten?</li> <li>• Kann diese Randbedingung ein Show-Stopper werden?</li> </ul>	

#### 5.3.4.6 Annotationstyp Ablösung

Eine der größten Herausforderungen in großen Organisationen ist nicht die Entwicklung neuer Informationssysteme, sondern die kontinuierliche Weiterentwicklung und Pflege von Legacy-Systemen. Dabei wachsen Anwendungen eher organisch als strukturiert, was dazu führt, dass der Aufwand für Wartung und Weiterentwicklung ab einem gewissen Zeitpunkt nicht mehr wirtschaftlich ist und die Anwendung am Ende ihres Lebenszyklus angelangt ist. Das Legacy-System wird abgelöst. Jedoch dauert ein Lebenszyklus oft länger, als die Verweildauer der Mitarbeiter in Organisationen ist, die eine Anwendung einst konzipiert und entwickelt haben, so dass fehlendes Wissen und fehlende Erfahrung zum Zeitpunkt der Migration nicht mehr zur Verfügung steht (vgl. zum vorherigen Abschnitt [151]). Um geeignete Maßnahmen für die Ablösung eines Legacy-Systems rechtzeitig ergreifen zu können, ist eine frühe Kennzeichnung abzulösender Systeme oder Systemkomponenten notwendig. Dazu existiert der Annotationstyp Ablösung (vgl. Tabelle 10).


Tabelle 10: Annotationstyp Ablösung

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Entfall, Ersatz
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Ablösung wird verwendet, um Landkartenelemente zu kennzeichnen, die in Zukunft abgelöst werden sollen oder müssen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung darauf zu achten ist, dass diese Prozess- oder Systemteile zukünftig entfallen oder ersetzt werden.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil soll abgelöst werden?</li> <li>• Wie soll dessen Aufgabe künftig umgesetzt werden (wenn überhaupt)?</li> <li>• Warum und auf wessen Initiative hin soll die Ablösung erfolgen?</li> </ul>	

#### 5.3.4.7 Annotationstyp Komplexität

Entwicklungsprojekte für Informationssysteme sind häufig komplexer, als vom Projektteam zunächst angenommen wird [81], denn die Komplexität ist nicht nur technischer Natur, sondern impliziert auch organisatorische Aspekte, auf die das Projektteam keinen Einfluss hat [80]. Darüber hinaus wächst der Leistungsdruck durch steigende Ressourcenknappheit [79] und raschen technologischen Wandel [242]. Es ist eine zentrale Aufgabe in Entwicklungsprojekten, für Informationssysteme Artefakte zu identifizieren, die Komplexitätstreiber darstellen, um sich der vorhandenen Komplexität früh bewusst zu werden und so die Möglichkeit zu besitzen, die entsprechenden Ressourcen mit adäquater Qualifikation und Erfahrung für die Lösung komplexer Probleme zu allokalieren [88]. Komplexität ist nicht immer unmittelbar erkennbar, sondern erfordert die Einschätzung durch erfahrene Mitglieder eines Projektteams. Um Komplexitätstreiber zu kennzeichnen, existiert der Annotationstyp Komplexität (vgl. Tabelle 11).





















Tabelle 11: Annotationstyp Komplexität

	Annotationskategorie: Risikotreiber
	Untertyp(en): -
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p><b>Erläuterung</b></p> <p>Der Annotationstyp Komplexität wird verwendet, um Landkartenelemente zu kennzeichnen, die durch ihre inhärente Komplexität zur Herausforderung im Projekt werden. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung algorithmische Komplexität, Komplexität durch das Zusammenspiel vieler Beteiligter und Komplexität durch technologische Herausforderungen behandelt werden müssen.</p>	
<p><b>Anschlussfragen</b></p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil ist besonders komplex?</li> <li>• Worin besteht die Komplexität?</li> <li>• Kann dies ein Show-Stopper werden?</li> </ul>	

#### 5.3.4.8 Übersicht über alle Annotationstypen

In diesem Abschnitt wird ein Überblick über alle Annotationstypen geben. Für jeden Annotationstypen werden sein Symbol, seine Kategorien und die Landkarten, auf denen er verwendet werden kann, abgebildet (vgl. Tabelle 12).

Tabelle 12: Überblick über die Annotationstypen

Annotationstyp	Symbol	Annotationskategorie			Landkarten		
		Werttreiber	Aufwands- treiber	Risikotreiber	Prozessland- karte	Objektland- karte	Integrations- landkarte
Business Value		X			X	X	X
User Value		X			X	X	X
Innovation		X			X	X	X
Sicherheit			X		X	X	X
Hohe Last			X		X	X	X
Zeitbeschränkung			X		X	X	X
Korrektheit			X		X	X	X
Zuverlässigkeit			X		X		X
Benutzbarkeit			X		X		
Flexibilität			X		X	X	X
Mobilität			X		X	X	X
Manuelle Bearbeitung			X		X		
Automatisierung			X		X		
Externe Ressource			X				X
Randbedingung			X		X	X	X
Verbesserungsbedarf			X		X	X	X
Ablösung			X		X	X	X
Unveränderlichkeit			X		X	X	X
Komplexität				X	X	X	X
Ungewissheit				X	X	X	X

### 5.3.5 Verwendung der Annotationen

Das Vorgehen zur Auswahl von Annotationstypen für das Annotieren einer Landkarte und deren Verwendung besteht aus sechs Schritten (vgl. Abbildung 27). Im Folgenden wird ein Überblick über den Annotationsprozess gegeben, um diesen in den darauffolgenden Unterkapiteln zu detaillieren.

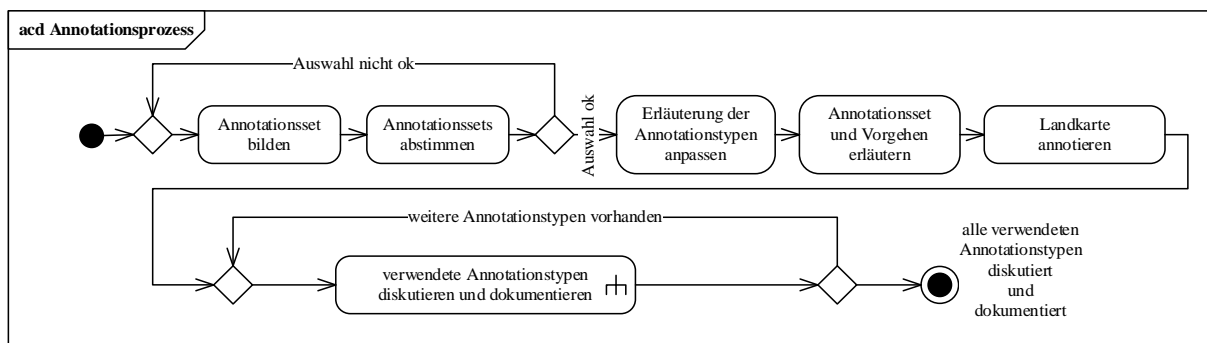


Abbildung 27: Annotationsprozess

Im ersten Schritt werden die zu verwendenden Annotationstypen für eine Landkarte ausgewählt und ein so genanntes Annotationsset gebildet. Die Grundlage für das Bilden eines Annotationssets sind die Informationen aus dem Methodenbaustein Vorbereitung eines Interaction-Room-Workshops und die Landkarten, auf denen ein Typ verwendet werden darf. Ein Annotationsset wird mit den Projektverantwortlichen abgestimmt. Entsprechend des Ergebnisses der Abstimmung wird das Annotationsset entweder angepasst oder die Erläuterung der Annotationstypen wird situationsspezifisch angepasst (mindestens durch projektspezifische Beispiele). Den Stakeholdern wird das Vorgehen zum Annotieren erläutert und die ausgewählten Annotationstypen werden vorgestellt. Daraufhin annotieren die Stakeholder eine Landkarte in dem dafür vorgesehenen Zeitraum. Das Annotieren geschieht individuell. Im Anschluss werden alle verwendeten Annotationstypen diskutiert und dokumentiert.

#### 5.3.5.1 Annotationssets bilden

Um die kognitive Last der Stakeholder zu reduzieren, werden nicht alle Annotationstypen gleichzeitig verwendet, sondern nur eine Teilmenge. Die Anzahl der Annotationstypen je Annotationsdurchgang ist auf sieben beschränkt, da Untersuchungen gezeigt haben, dass das menschliche Kurzzeitgedächtnis  $7 \pm 2$  Informationseinheiten behalten und verarbeiten kann, ohne dass eine kognitive Überlastung entsteht [243]. Um die Annotationstypen auszuwählen, müssen unterschiedliche Informationen berücksichtigt werden. Ausgehend von den benutzbaren Annotationen je Landkarte (vgl. Kapitel 5.3.4), werden die Informationen aus der



Vorbereitung für den Workshop (vgl. Kapitel 5.1.2) genutzt, um die zu verwendenden Annotationstypen eines Annotationssets festzulegen.

#### 5.3.5.2 Annotationssets abstimmen

Die Auswahl der Annotationstypen wird mit den Projektverantwortlichen abgesprochen. Dazu wird erläutert, was zur Entscheidung der ausgewählten Annotationstypen geführt hat. Stimmen die Projektverantwortlichen der Auswahl zu, wird mit dem folgenden Schritt fortgefahren, andernfalls wird die Auswahl in Abstimmung mit den Projektverantwortlichen angepasst, bevor fortgefahren wird.

#### 5.3.5.3 Beschreibung der Annotationstypen anpassen

Bestimmte Situationen können eine Anpassung der Erläuterung einzelner Annotationstypen erfordern, um den Stakeholdern den Zugang zu deren Verwendung zu erleichtern. Beispielsweise kann die Erläuterung des Annotationstypen Komplexität (vgl. Kapitel 5.3.4.7) auf Komplexität durch politische Einflüsse, Komplexität durch das Zusammenspiel vieler Stakeholder oder durch algorithmische Komplexität ausgerichtet werden. Zur Erläuterung der Annotationen wird die Vorlage aus Tabelle 13 verwendet.

Tabelle 13: Vorlage zur Erläuterung der Annotationstypen

---

Kennzeichnet Elemente, bei denen Y passieren kann/wichtig ist. [Optional: Sie zeigt, dass bei Entwurf und Implementierung besonders zu beachten ist, wie mit dieser Herausforderung umgegangen werden kann].

---

Die Vorlage besteht aus einem obligatorischen und einem optionalen Teil (Letzteres in Tabelle 13 in eckigen Klammern). Im obligatorischen Teil wird die Verwendung eines Annotationstyps erklärt und im optionalen Teil, welche Konsequenzen daraus folgen können. Um den Einsatzzweck der Annotationstypen zu verdeutlichen, werden situationsspezifische Beispiele je Annotationstyp gegeben.

#### 5.3.5.4 Annotationsset vorstellen und Vorgehen erläutern

Die Annotationstypen werden den Stakeholdern, inklusive deren Symbol, Erläuterung und Beispiele präsentiert. Die Information über die zu verwendenden Annotationstypen eines Annotationssets bleibt während der nachfolgenden Schritte für die Stakeholder sichtbar, damit sie diese Information stets abrufen können. Neben der Erläuterung der Annotationstypen wird der

Prozess zum Annotieren, Diskutieren und Dokumentieren erörtert (vgl. Kapitel 5.3.5.5 und 5.3.5.6). Es wird explizit auf das Ziel der Annotationen hingewiesen: Landkartenelemente kennzeichnen, die überdurchschnittlichen Wert, überdurchschnittlichen Aufwand oder überdurchschnittliches Risiko besitzen. Abschließend werden Fragen der Stakeholder geklärt, bevor mit dem nächsten Schritt fortgefahren wird.

#### 5.3.5.5 Modellskizze annotieren

Die Stakeholder bekommen ca. zehn Minuten Zeit, eine Landkarte mit den zur Verwendung vorgeschlagenen Annotationstypen des Annotationssets zu kennzeichnen. Das Annotieren erfolgt individuell, um Hierarchiekonflikten und weiteren negativen gruppenspezifischen Effekten (vgl. z. B. [244]) vorzubeugen. Jeder Stakeholder darf so viele Annotationstypen verwenden, wie aus seiner Sicht notwendig sind. Es existiert keine Einschränkung in der Anzahl gleicher oder verschiedener Annotationstypen je Landkartenelement. Eine Konsolidierung der Annotationstypen erfolgt während der anschließenden Diskussion und Dokumentation.

#### 5.3.5.6 Annotationen diskutieren und dokumentieren

Nach den zehn Minuten individueller Kennzeichnung einer Landkarte wird jeder Annotationstyp an einem Landkartenelement im Plenum einzeln diskutiert und die wichtigsten Aspekte dokumentiert. Die Dokumentation erfolgt mit Hilfe einer speziellen Vorlage (vgl. Anhang D, Abbildung 56). Auf der Vorlage wird der entsprechende Annotationstyp, sein Untertyp, der Befürworter, Nutzen und Schwierigkeit sowie der Grund dokumentiert. Sofern ein Konflikt auftritt und dokumentiert werden muss, geschieht dies unter Angabe der Konfliktparteien. Um die Vorlage auszufüllen, werden die folgenden Schritte durchgeführt (vgl. Abbildung 28). Zur Diskussion und Dokumentation werden beide IR-Coaches benötigt: Der IR-MC steht an der annotierten Landkarte und leitet die Diskussion, während der IR-DC dokumentiert.

Der IR-MC zeigt auf ein Annotationssymbol und erfragt, wer diesen Annotationstyp verwendet hat. Der Name des Stakeholders wird vom IR-DC als Befürworter der Annotation dokumentiert. Der IR-MC stellt die spezifischen Anschlussfragen des Annotationstyps und bittet den Befürworter dadurch zur Abgabe der Begründung für die Annotation. Der IR-DC ist für die Reflexion und Beurteilung fachlicher Diskussionen zuständig (vgl. Kapitel 5.1.1): Ist die Begründung des Befürworters offensichtlich oder trivial, fragt der IR-DC die Stakeholder, ob die Begründung dokumentiert werden muss. Sollte die Mehrheit der Stakeholder der Auffassung sein, dass die Annotation keine überdurchschnittliche Herausforderung kennzeichnet, entfernt

der IR-MC den Annotationstyp von der Landkarte und der IR-DC dokumentiert den Grund der Annotation nicht. Andernfalls wird der Grund der Annotation dokumentiert.

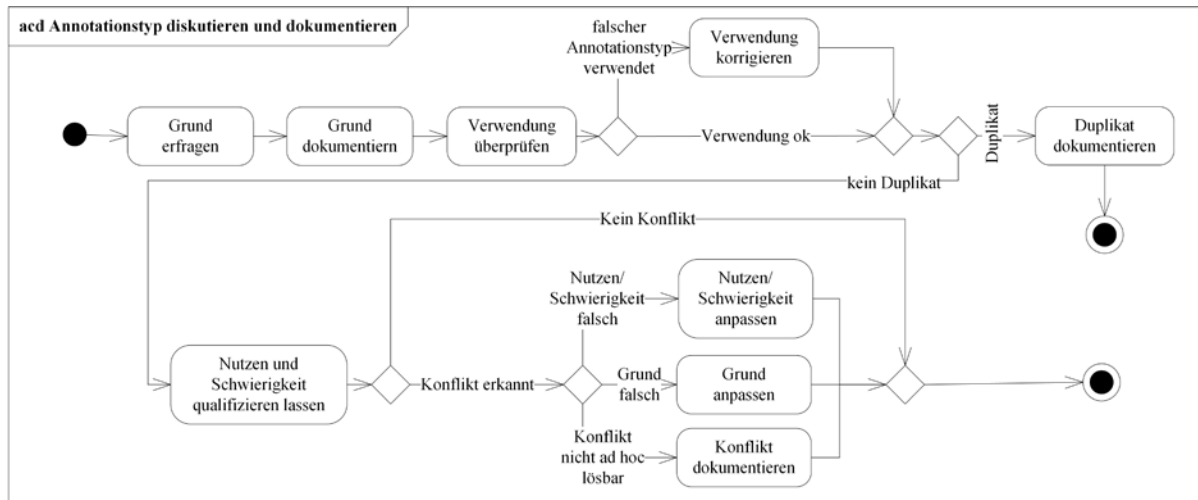


Abbildung 28: Annotationstyp diskutieren und dokumentieren

Der IR-MC prüft, ob der korrekte Annotationstyp verwendet wurde und schlägt dessen Austausch vor, sofern ein anderer Annotationstyp besser zur Begründung passt. Der IR-DC notiert den endgültig ausgewählten Annotationstyp. Sofern aus der Erläuterung des Stakeholders der Untertyp des Annotationstypen offensichtlich wurde, notiert der IR-DC diesen. Andernfalls fragt er den Stakeholder, welche der annotationsspezifischen Untertypen zutreffen, und notiert diese.

In der Praxis existieren Annotationsduplikate. Ein Duplikat besitzt denselben Annotationstyp und dieselbe Begründung. Es kann dabei am selben oder an unterschiedlichen Landkartenelementen heften. Sofern ein Duplikat identifiziert wird, wird dieselbe ID der ursprünglichen Annotation erneut vergeben und es wird mit der Diskussion der nächsten Annotation fortgefahren. Andernfalls vergibt der IR-DC eine fortlaufende eindeutige ID für die Annotation, dokumentiert diese, teilt sie dem IR-MC mit und Letzterer notiert die ID neben dem Annotationstyp auf der Landkarte.

Durch die Erläuterung der Begründung haben alle Stakeholder ein gemeinsames Verständnis über die Annotation. Die Stakeholder werden gebeten, eine Einschätzung des erwarteten Nutzens und der erwarteten Schwierigkeit abzugeben, welche durch Umsetzung der Annotation entstehen würde. Die Einschätzung wird mit Hilfe von T-Shirt-Größen vorgenommen, da diese Näherungstechnik insbesondere für den Einsatz in frühen Projektphasen geeignet ist [74]. Die Stakeholder qualifizieren den Nutzen und die Schwierigkeit in den Kategorien small, medium, large oder extra large. Um die qualifizierende T-Shirt-Größe festzulegen, soll ein Konsens

gefunden werden. Auf diese Weise können Konflikte zwischen Stakeholdern identifiziert werden. Der Konflikt entsteht entweder durch Uneinigkeit der Stakeholder über die Größe von Nutzen bzw. Schwierigkeit oder, wenn eine starke Divergenz zwischen der T-Shirt-Größe und der Begründung besteht. Die T-Shirt-Größe muss also zur geäußerten Begründung passen. Ist dies nicht der Fall, ist es die Aufgabe der IR-Coaches, diese Divergenz zu erkennen, anzusprechen und aufzulösen. Falls dies nicht unmittelbar möglich ist, wird ein Konflikt dokumentiert.

### 5.3.6 Praxisbeispiel

In diesem Kapitel wird das Praxisbeispiel aus Kapitel 5.2.5 aufgegriffen und die Landkarten um Annotationen ergänzt. Nachdem die Bedeutung und Begründung der Annotationen auf den Landkarten dargestellt wurde, wird in Kapitel 5.3.6.3 auf die Besonderheiten bei der Verwendung der Annotationstypen und auf Erkenntnis aus den Annotationen eingegangen.

#### 5.3.6.1 Annotierte Prozesslandkarten

In der Kontextanalyse wurde das Projekt als Neuentwicklungsprojekt klassifiziert, daher wurde die folgende Auswahl der Annotationstypen für die Prozesslandkarte getroffen (vgl. Abbildung 29), die Auswahl wird je Annotationstyp kurz begründet:

- Der Annotationstyp Business Value wurde ausgewählt, um Hinweise zu erhalten, welche Werttreiber zu der Entscheidung geführt haben, das System implementieren zu lassen, und um Argumente zu sammeln, die entstehende Entwicklungskosten rechtfertigen.
- Der Annotationstyp User Value wurde ausgewählt, damit die Nutzer des Systems (ADM, Mitarbeiter aus dem Innendienst) ausdrücken können, welche Funktionalität ihnen besonders wichtig ist.
- Der Annotationstyp Zeitbeschränkung wurde wegen der Anforderung, die Reaktionszeit bis zur Bearbeitung eines Leads zu verringern, ausgewählt.
- Der Annotationstyp Manuelle Bearbeitung wurde ausgewählt, weil die Reklamation, die Buchung von PLZ-Gebieten und der Lead-Kauf nicht vollautomatisch erfolgen können.
- Der Annotationstyp Flexibilität wurde wegen der Rabattaktionen, der nachträglichen Preis-anpassung und der unterschiedlichen Rollen ausgewählt.

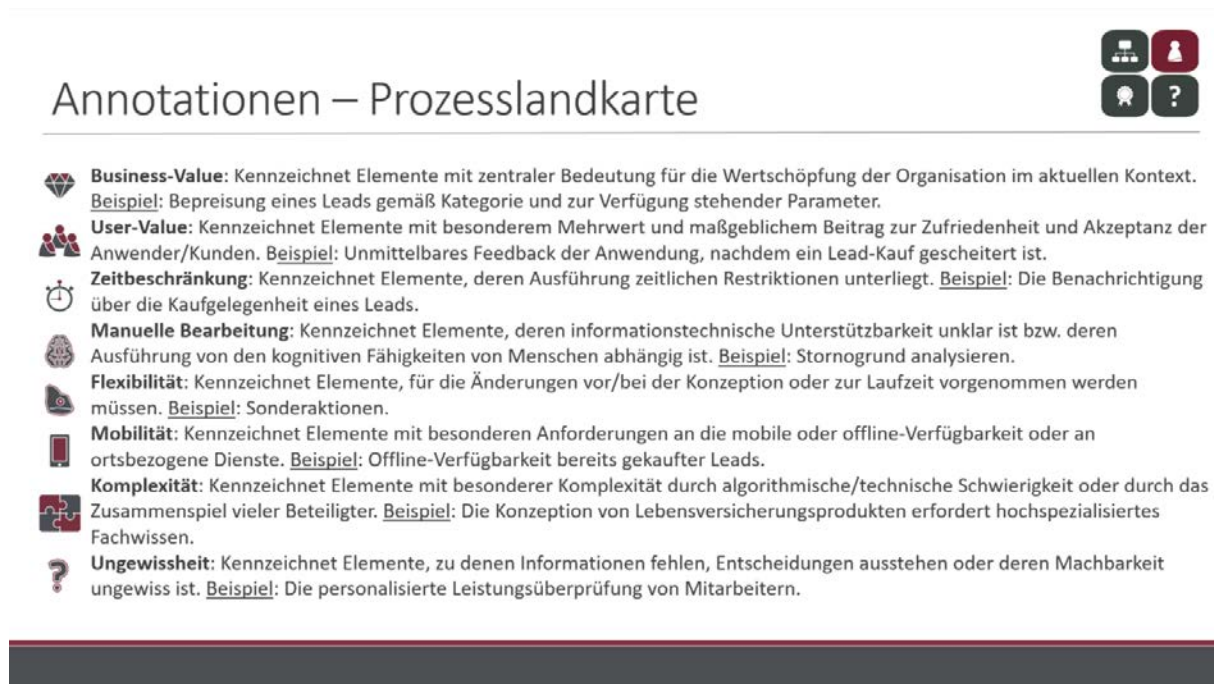


Abbildung 29: Zur Verwendung vorgeschlagene Annotationstypen für die Prozesslandkarten des Praxisbeispiels

- Der Annotationstyp Mobilität wurde ausgewählt, um zu kennzeichnen, welche Funktionen auf dem mobilen System für die ADMs zur Verfügung stehen sollen.
- Der Annotationstyp Komplexität wurde ausgewählt, um ein Gefühl für komplexe Funktionen zu erhalten.
- Der Annotationstyp Ungewissheit wurde ausgewählt, um herauszufinden, wo Ungewissheit in den Prozessen existiert, die für eine Ausschreibung eliminiert werden müssen.

Abbildung 30 zeigt die annotierte Prozesslandkarte zum Epic „Lead erfassen“. Auf den ersten Blick fällt auf, dass der Annotationstyp Verbesserungsbedarf (Annotation 25) verwendet wurde, obwohl dieser nicht zur Verwendung auf der Prozesslandkarte vorgeschlagen wurde. Es stellte sich heraus, dass das Lead-Erfassungsformular technisch an die neuen Anforderungen angepasst und die Komponente zum Abgleich des Leads mit dem CRM in den Prozess integriert werden muss. Die Stakeholder haben den Annotationstyp verwendet, als er für die Verwendung auf der Objekt- und Integrationslandkarte vorgeschlagen wurde (vgl. Kapitel 5.3.6.2). Genau genommen hätte die Annotation 25 in zwei Annotationen geteilt werden müssen, eine zur Anpassung des Formulars und eine Weitere für die Integration der Komponente. Letztere hätte auf der Integrationslandkarte platziert werden müssen. Das Teilen der Annotation während des Workshops hätte zur theoriekonformen Anwendung der Methode beigetragen, aber Overhead

verursacht. Daher wurde die Information dokumentiert und sichergestellt, dass diese Information zugeordnet werden kann.

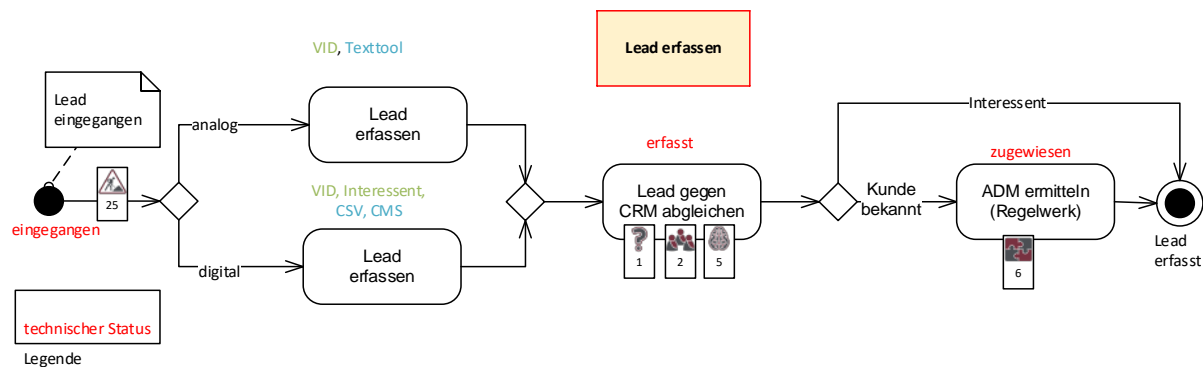


Abbildung 30: Annotierte Prozesslandkarte „Lead erfassen“

Der Prozessschritt, in dem das Lead gegen das CRM abgeglichen wird, wurde als ungewiss gekennzeichnet (Annotation 1), weil das Verhältnis zwischen Geschwindigkeit und Datenqualität des Abgleichs in der Komponente abgewägt werden muss. Durch eine mögliche Reduktion der Datenqualität kann eine schnellere Verteilung der Leads erreicht werden, wobei eine bessere Datenqualität eine längere Bearbeitungsdauer durch manuelles Eingreifen benötigt. Die Schwierigkeit der Entscheidungsfindung ist gering (T-Shirt-Größe: S).

Die Akzeptanz des Systems durch den Außendienst wird maßgeblich von der Datenqualität der Leads beeinflusst, weil der Außendienst zukünftig Geld für den Erwerb von Leads zahlt (Annotation 2). Die automatische Verarbeitung eingehender Leads kann zu einer geringen Bearbeitungszeit, aber auch zu Doubletten führen, die negative Auswirkungen auf den User Value haben. Die Lösung stellt ein detaillierterer Datenabgleich gemeldeter Doubletten dar, von dem sehr hoher Nutzen erwartet wird (T-Shirt-Größe: XL).

Es ist ein händisches Clearing beim Abgleich der Leads mit dem Bestand erforderlich, sobald die Lead-Daten in mindestens einem Parameter von Kundendaten abweichen. Dies ist sehr einfach umzusetzen und bietet einen hohen Nutzen (T-Shirts-Größen: Nutzen XL, Schwierigkeit S).

Die Implementierung des Regelwerks (Annotation 6) ist komplex: Die Herausforderung liegt in der Ermittlung fachlicher Kriterien zur Identifikation des korrekten Außendienstmitarbeiters, der auch ein Makler sein kann.

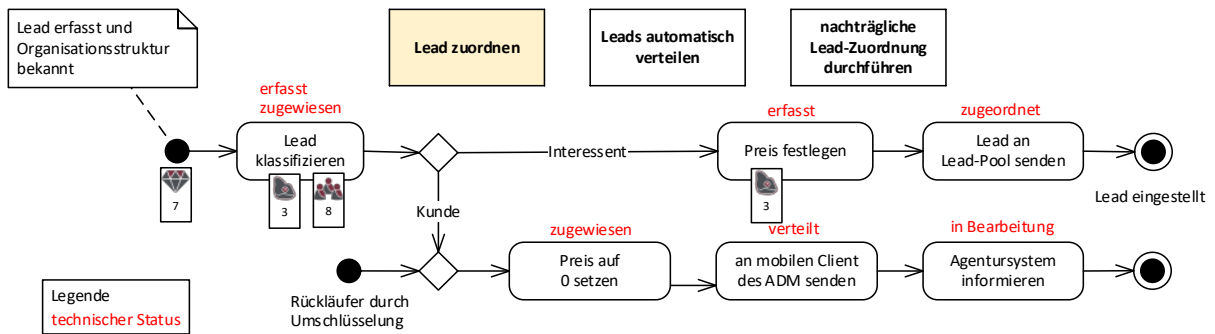


Abbildung 31: Annotierte Prozesslandkarte „Leads zuordnen“

Die Annotation 7 aus Abbildung 31 kennzeichnet den gesamten Prozess: Der Business Value entsteht durch die Verbesserung der Geschwindigkeit innerhalb der Klassifizierung und Verteilung der Leads. Daraus resultiert, dass ein Lead schneller verkauft werden kann, was zur zügigeren Bearbeitung des Leads führt, die Chancen auf einen Abschluss erhöht und somit einen Wert für das VU darstellt. Nach dem Nutzen wurde nicht gefragt, weil die Annotation ein Ziel des Projektes widerspiegelt.

Durch Annotation 3 wird sowohl für die Klassifikation als auch für die Bepreisung von Leads eine Flexibilitätsanforderung gestellt. Im CMS ist hinterlegt, welche Produkte welche Wertigkeit besitzen (Kategorie Gold, Silber oder Bronze). Die Klassifikation und Bepreisung muss jedoch über eine Rabattstufe individuell angepasst werden können. Der Nutzen der Flexibilität ist hoch, bei mittlerer Höhe der Schwierigkeit (T-Shirt-Größen: Nutzen L, Schwierigkeit M).

Aus der Klassifikation und Anpassung der Lead-Informationen entsteht ein Wert (Annotation 8), weil detaillierte Auswertungen erstellt werden können. Auswertungen sollen über Sparten und über den Erfolg von Rabattaktionen erstellt werden, was einen mittleren Nutzen stiftet (T-Shirt-Größe: M).

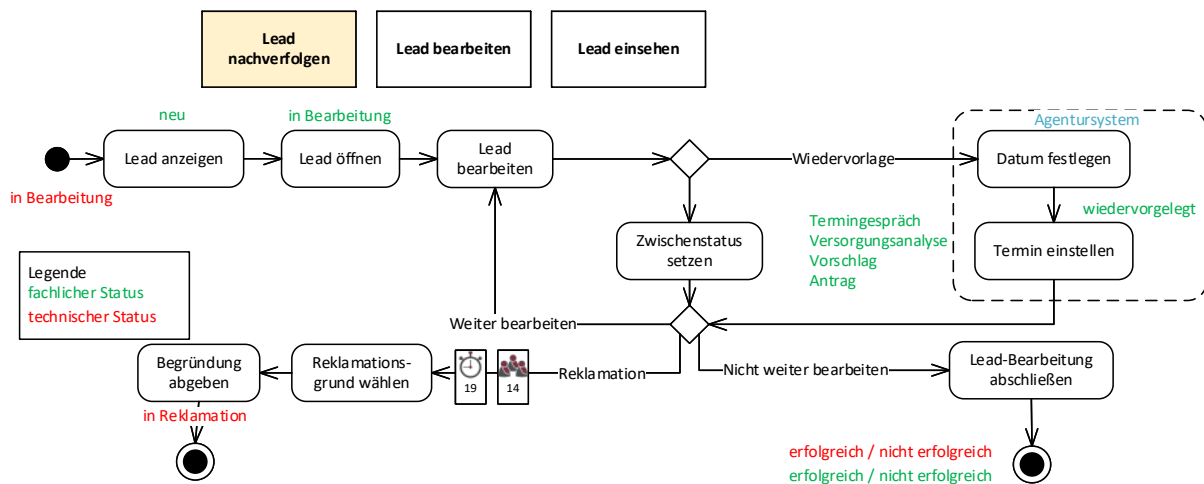


Abbildung 32: Annotierte Prozesslandkarte „Leads nachverfolgen“

Für die Prozesslandkarte aus Abbildung 32 wurde nur die Reklamation als überdurchschnittliche Herausforderung gekennzeichnet, denn der Reklamationsprozess muss transparent für einen ADM gestaltet sein. Ein kontinuierliches Feedback zum Bearbeitungsstatus der Reklamation soll dem ADM diese Transparenz bieten, die zu einer höheren Akzeptanz der Veränderung durch die Einführung des Kaufs von Leads führen soll. Fehlende Transparenz kann zu vielen Statusanfragen beim Vertriebsinnendienst führen, wodurch zusätzliche Arbeitsbelastung entsteht (T-Shirt-Größe: Nutzen L). Trotzdem soll die Möglichkeit zur Reklamation eines Leads nur innerhalb der ersten 14 Tage nach dem Kauf des Leads gewährt werden (Annotation 19).

### 5.3.6.2 Annotierte Integrations- und Objektlandkarte

Abbildung 33 zeigt die zur Verwendung vorgeschlagenen Annotationstypen für die Objekt- und Integrationslandkarte. Für beide Landkarten wurde dasselbe Annotationsset verwendet. Die Auswahl wird im Folgenden kurz begründet.

- Der Annotationstyp Zuverlässigkeit wurde aufgrund der Anforderung des Lead-Kaufs über eine mobile Anwendung ausgewählt. Beim Kauf konkurrieren ADMs, die dasselbe PLZ-Gebiet gebucht haben, weswegen Robustheit und Verfügbarkeit relevante Kriterien darstellten.
- Der Annotationstyp Zeitbeschränkung wurde ebenfalls wegen des Zusammenspiels von Webanwendung und mobiler Anwendung verwendet.
- Der Annotationstyp Korrektheit wurde wegen notwendiger Konsistenzbedingungen im Zusammenspiel mit Umsystemen ausgewählt und, um Anforderungen an die Aktualität und Präzision der Daten für das Controlling zu kennzeichnen.



- Der Annotationstyp Verbesserungsbedarf wurde ausgewählt, um zu kennzeichnen, welche Schnittstellen und Datenstrukturen angepasst oder erweitert werden müssen.
- Der Annotationstyp Externe Ressource wurde ausgewählt, um zu kennzeichnen, ob Umssysteme außerhalb der Anwendungslandschaft und damit der Beeinflussbarkeit des VUs liegen.
- Der Annotationstyp Ungewissheit wurde ausgewählt, um fehlende Informationen und Entscheidungen kenntlich zu machen.

## Annotationen – Objekt- und Integrationslandkarte









-  **Zuverlässigkeit:** Elemente mit besonderen Anforderungen an die Robustheit oder Verfügbarkeit. Beispiel: Die Schnittstelle der App muss sehr robust sein, besonders bei Verbindungsabbrüchen.
-  **Zeitbeschränkung:** Elemente, die besonderen zeitlichen Anforderungen genügen müssen. Beispiel: Die Bestätigung eines Leadkaufs muss 15 Sekunden nach dem Kauf beim Käufer angelangt sein.
-  **Korrektheit:** Elemente mit besonderen Anforderungen an die Aktualität, Präzision oder Konsistenz von Daten. Beispiel: Sofern ein Lead zum Kauf vorgemerkt ist, darf dieses Lead nicht mehr für den Kauf durch andere AMDs zur Verfügung stehen.
-  **Verbesserungsbedarf:** Elemente, die fachlich/technisch angepasst/erweitert/vereinfacht werden sollen/müssen. Beispiel: Das Gutscheinsystem muss vereinfacht werden, sodass keine Restbeträge aus den Gutscheinen übrig bleiben, die separat verwaltet werden müssen.
-  **Externe Ressource:** Elemente, die Daten oder Funktionalitäten aus einer externen Quelle beziehen. Beispiel: Schnittstellen zu Maklerorganisationen.
-  **Ungewissheit:** Kennzeichnet Elemente, zu denen Informationen fehlen, Entscheidungen ausstehen oder deren Machbarkeit ungewiss ist. Beispiel: Die personalisierte Leistungsüberprüfung von Mitarbeitern.

Abbildung 33: Zur Verwendung vorgeschlagene Annotationstypen für die Objekt- und die Integrationslandkarte des Praxisbeispiels

Auf der Integrationslandkarte (vgl. Abbildung 34) wurde der Annotationstyp Ungewissheit verwendet, der kennzeichnet, dass nicht entschieden ist, ob eine Synchronisation zwischen dem Agentursystem und dem zentralen Informationssystem implementiert wird (Annotation 29). Sollten sich die Entscheider gegen Implementierung der Synchronisation von Leads mit dem Agentursystem entscheiden, muss im zentralen Informationssystem eine Terminverwaltung implementiert werden. Zusätzlich muss eine Schnittstelle zur To-do-Liste des Agentursystems geschaffen werden, damit wiedervorgelegte Leads im Agentursystem erscheinen. Darüber hinaus ist unklar, ob die Aktionsnummern weiterhin zugeliefert werden (Annotation 30), oder ob das Tool zur Generierung der Aktionen und Aktionsnummern in das neue System integriert werden soll.

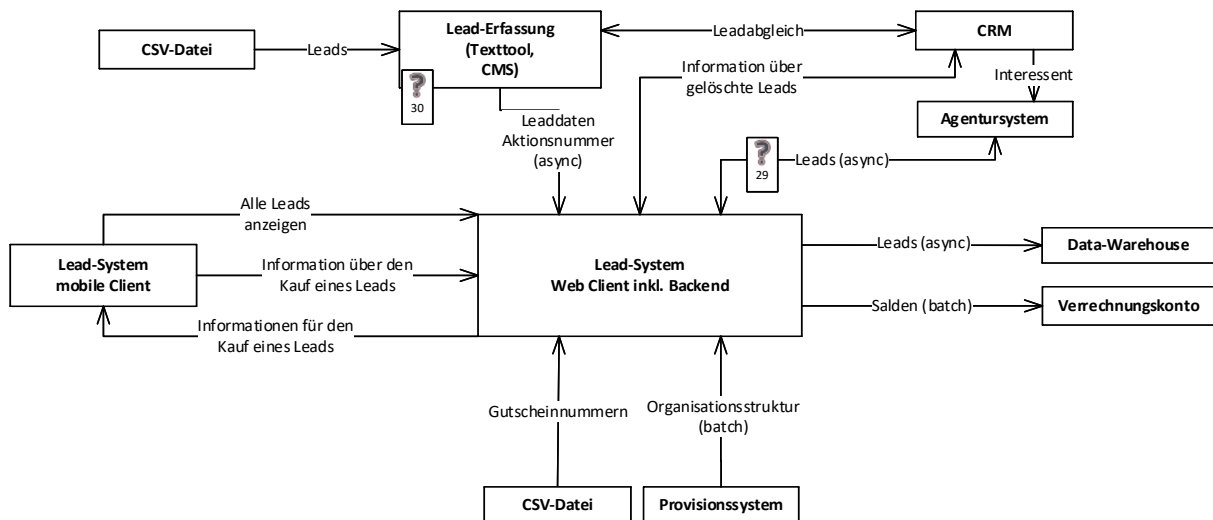


Abbildung 34: Annotierte Integrationslandkarte des Praxisbeispiels

Abbildung 35 zeigt die annotierte Objektlandkarte. Mehrere ADMs können zu einer Bürogemeinschaft zusammengeschlossen sein, in der die ADMs Lead-Konten gegenseitig einsehen können. Dieser Umstand führt zur Notwendigkeit, ADMs identifizieren zu können, die in einer Bürogemeinschaft ansässig sind. Das derzeitige Rollen- und Berechtigungskonzept soll vereinfacht werden (Annotation 18). Ein Lösungsansatz zur Vereinfachung ist, dass die notwendigen Rechte von ADMs in einer Bürogemeinschaft Ähnlichkeit mit den Berechtigungen einer mittleren Führungskraft besitzen. Allerdings ist die Rechtee Ausstattung der mittleren Führungskräfte ungewiss, denn diese können auch vermittelnd tätig sein, was zu einem Interessenskonflikt führen kann (Annotation 26).

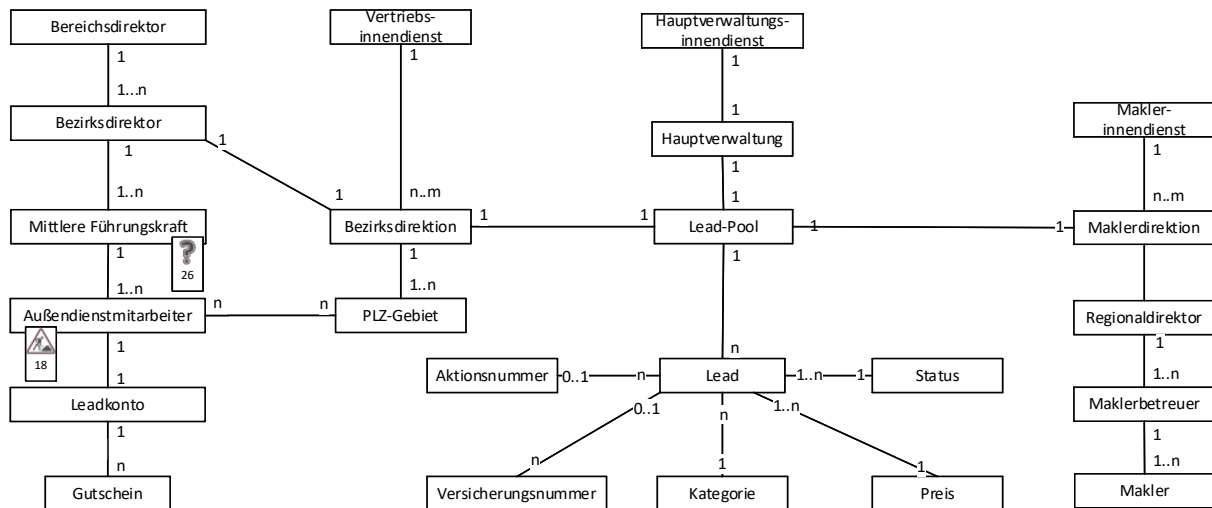


Abbildung 35: Annotierte Objektlandkarte des Praxisbeispiels

### 5.3.6.3 Besonderheiten und Resultate aus der Benutzung der Annotationen

Das Praxisbeispiel zeigt, dass die meisten Annotationen auf den Prozesslandkarten verwendet wurden. Dabei wurden jedoch nicht nur Aspekte, die sich unmittelbar auf die Prozesse beziehen, annotiert, sondern auch technische Aspekte, die auch auf der Integrationslandkarte hätten annotiert werden können (z. B. Annotation 25). Einerseits wurden nicht alle zur Verwendung vorgeschlagenen Annotationen verwendet, denn die Verwendung der Annotationen ist ein Angebot an die Stakeholder. Andererseits wurden ebenfalls Annotationstypen auf der Prozesslandkarte verwendet, deren Verwendung dort nicht vorgesehen war. Anstatt auf die stringente Einhaltung der Methode zu bestehen, haben die IR-Coaches pragmatisch gehandelt und die Stakeholder gewähren lassen.

Die Annotationstypen können einen Impuls setzen, über bestimmte Wert-, Aufwands- und Risikotreiber nachzudenken, jedoch verleihen die Stakeholder den Annotationstypen durch die Begründung erst ihre konkrete Bedeutung. Deswegen wurde die Frage nach Nutzen und Schwierigkeit situativ eingesetzt: Ergibt die Frage nach Nutzen und/oder Schwierigkeit keinen Sinn, wird diese weggelassen. Durch die Frage nach der Höhe von Nutzen und Schwierigkeit der Entscheidung, ob das Tool für die Aktionsverwaltung abgelöst werden soll, ist zum Beispiel der Mehrwert gering, denn die Qualifikation in jeweils vier Schwierigkeiten wird der Komplexität der Entscheidungsfindung nicht gerecht. Die IR-Coaches besitzen mit der Qualifikation von Nutzen und Schwierigkeit ein methodisches Werkzeug zur Identifikation von Konflikten in der Stakeholder-Gruppe: Sofern die Begründung des Befürworters fraglich erscheint, kann durch die Qualifikation die Meinung der Gruppe eingeholt werden, ohne dass der Coach seine neutrale Position verlassen und die Aussage des Befürworters bewerten muss.

Trotz des Umstands, dass das Projekt eine Neuentwicklung ist, wurden Abhängigkeiten zu bestehenden Systemen und Komponenten sowie deren Anpassung diskutiert. Die Nähe von Stakeholdern zum gelebten Ist kann eine Hürde für die Sollkonzeption bedeuten, insbesondere wenn das Soll einen Innovationscharakter besitzt. Mit den Annotationen werden Soll-Anforderungen formuliert, die ohne Annotationen vermutlich auch diskutiert worden wären. Deren Dokumentation und Transfer in das spätere Entwicklungsprojekt wären jedoch von der Aufnahme in ein Protokoll der Besprechung oder der Erinnerung Einzelner abhängig gewesen. Durch die Annotationen können diese Informationen nicht nur strukturiert erhoben werden, sondern durch die Lokalisierung an Landkartenelementen zusätzlich in den Kontext des relevanten Projektausschnitts gesetzt werden (z. B. wurde über die Annotation 18 eine Komplexitätsreduktion der Berechtigungen für ADMs aus Bürogemeinschaften gefordert).

#### 5.4 Erkenntnisdokumentation

In Kapitel 6 werden 23 Industrieprojekte quantitativ untersucht. Als Ergebnis einiger dieser Projekte wurden mehrere hundert Seiten Anforderungsdokumente verfasst und an die Projektpartner übergeben. In den Anforderungsdokumenten wurden fachliche Grobkonzepte, Handlungsalternativen und -empfehlungen beschrieben (vgl. Abbildung 36).

Zur Veranschaulichung der Anschlussfähigkeit der Ergebnisse aus Interaction-Room-Workshops in Anforderungsdokumente und Artefakte des Softwareprozesses (z. B. Use-Case-Diagramme und Zustandsautomaten) wird zunächst die verwendete Dokumentenstruktur vorgestellt. Anschließend wird die Überführung von Ergebnissen aus Interaction Rooms in Artefakte in der Dokumentenstruktur aufgezeigt.

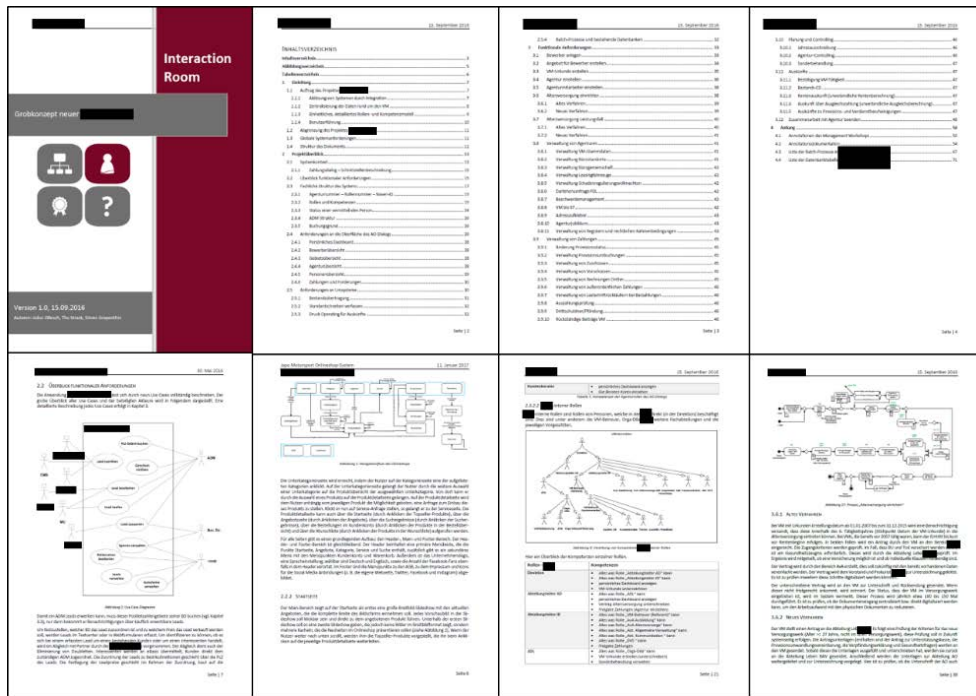


Abbildung 36: Beispiele für Ergebnisse des Interaction Rooms in Anforderungsdokumenten<sup>7</sup>

### 5.4.1 Struktur von Anforderungsdokumenten

Während viele unterschiedliche Strukturen zur Dokumentation von Anforderungen existieren (z. B. [245]–[247]), werden diese häufig an die notwendigen Rahmenbedingungen des Projektes angepasst [63], [248]. Unterauer [133] zeigt, wie von ihm vorgeschlagene Workshop-Ergebnisse in eine Dokumentenstruktur überführt werden können. Die vorgeschlagene Dokumentenstruktur enthält sieben Kapitel (vgl. Abbildung 37).

In der Einleitung werden der Zweck des Dokuments und seine Versionshistorie dargestellt. Das zweite Kapitel gibt einen Überblick über Ziele und Risiken, es wird die Vision erläutert und die Stakeholder werden genannt. Im dritten Kapitel wird ein Überblick der relevanten Geschäftsprozesse als Prozesslandkarte gegeben, um daraufhin einzelne Ist- und Soll-Prozesse durch Ablaufdiagramme zu beschreiben. Das Kapitel schließt mit einer Darstellung des Systemkontextes, durch ein Kontextdiagramm. Unterauer schlägt dazu ein Use-Case-Diagramm vor, in dem das zu entwickelnde System als Black-Box, also ohne Anwendungsfälle, dargestellt wird [133]. Das vierte Kapitel kann als Kernkapitel verstanden werden, denn es werden die Funktionen des Systems niedergeschrieben. Nach einem Überblick über alle Funktionen als Use-Case-Diagramm werden einzelne Anwendungsfälle beschrieben, es wird ein fachliches Datenmodell

<sup>7</sup> In der Abbildung wurden Bereiche geschwärzt, die Rückschlüsse auf die Industriepartner zulassen.

dokumentiert sowie Oberflächen und Berichte durch entsprechende Prototypen dokumentiert. Für die Dokumentation der Schnittstellen verwendet Unterauer eine vereinfachte Form eines UML-Sequenzflussdiagramms [133]. Die Kapitel fünf, sechs und sieben beinhalten Listen mit Qualitätsanforderungen, Rahmenbedingungen und einem Glossar.

<b>Dokumentstruktur</b>	<b>Ergebnisartefakt aus den Workshops</b>
<b>1. Einleitung</b>	
1.1 Zweck des Dokuments	
1.2 Versionshistorie	
<b>2. Überblick</b>	
2.1 Vision	Elevator Pitch
2.2 Stakeholder	Stakeholder-Liste, Kontextdiagramm
2.3 Ziele	Zieleliste
2.4 Risiken	Risiko-Backlog, Risikodetails
<b>3. Kontext</b>	
3.1 Prozessübersicht	Prozesslandkarte
3.2 Optional: Ist-Prozesse	Ablaufmodelle, Prozesskarten
3.3 Sollprozesse	Ablaufmodelle, Prozesskarten
3.4 Systemkontext	Kontextdiagramm
<b>4. Funktionen</b>	
4.1 Funktionsüberblick	Anwendungsfalldiagramm
4.2 Funktionen	Anwendungsfallkarten, Maskenentwürfe
4.3 Datenmodell	Fachliches Datenmodell, Mengengerüst
4.4 Masken	Papierprototypen, Storyboards
4.5 Berichte	Excel-Prototypen
4.6 Schnittstellen	Ablaufdiagramme, Beschreibungen
<b>5. Qualitätsanforderungen</b>	Liste der Anforderungen, Mengengerüst
<b>6. Rahmenbedingungen</b>	Liste der Rahmenbedingungen
<b>7. Glossar</b>	Glossardokument

Abbildung 37: Abbildung von Artefakten aus Workshops in eine Dokumentstruktur (angelehnt an [133])

Die vorgeschlagenen Inhalte und Ergebnisse der Workshops nach Unterauer sind teils deckungsgleich mit den Ergebnissen eines Interaction-Room-Workshops. Während mit dem Interaction Room keine Masken- und Berichtsprototypen erstellt werden, finden sich alle anderen Artefakte im Interaction Room wieder. Zur Dokumentation der Ergebnisse aus Interaction-Room-Workshops wird die vorgeschlagene Struktur von Brückmann und Grapenthin [116] verwendet:

## 1. Projektvision

In der Projektvision wird festgehalten, was das Ziel bzw. der Zweck des Projekts ist, welche Motivation für das Projekt existiert. Des Weiteren wird die Ebene der dokumentierten Anforderungen festgelegt.

## 2. Überblick

Die Überblicksebene dient zur technischen und fachlichen Einordnung der Anwendung. Das Kapitel beinhaltet eine Verortung des Systems in die fachliche Domäne. Hierbei wird kurz beschrieben, welche fachlichen Prozesse und Funktionen durch das System unterstützt werden. Es wird ein Überblick über die Hauptfunktionen des Systems durch eine Kurzbeschreibung der wichtigsten Systemfunktionen geben. Die Einbettung der Anwendung in die Systemlandschaft und eine Kurzbeschreibung der technischen Schnittstellen dient der Darstellung technischer Abhängigkeiten. Ergänzend zu der Dokumentation der Systemfunktionen und technischen Integration wird ein übergreifendes fachliches Objektmodell zur Veranschaulichung der fachlichen Zusammenhänge und Eigenschaften von Geschäftsobjekten beschrieben.

## 3. Detaillierte Anforderungen

Auf die Systemfunktionen, die im zweiten Kapitel kurz beschrieben wurden, wird in diesem Kapitel detailliert eingegangen. Bei der Beschreibung der jeweiligen Anforderungen sollte jede Systemfunktion in Teilfunktionen zerlegt und diese kurz erläutert werden. Zu jeder Teilfunktion werden alle Informationen zusammengestellt, die verfügbar sind. Es sollten alle relevanten funktionalen Anforderungen, Qualitätsanforderungen und Randbedingungen dokumentiert werden. Diese Anforderungen werden in Form von Modellen und beschreibenden Texten dokumentiert.

## 4. Anhang

Im Anhang werden alle Elemente dokumentiert, die nicht in den vorherigen Kapiteln dokumentiert wurden. Dazu kann ein Glossar zählen, Gesprächsnotizen, Interview-Mitschriften oder Fotos von Ergebnissen aus Workshops und Besprechungen.

## 5.4.2 Abbildung der Interaction-Room-Erkenntnisse auf eine Struktur für Anforderungsdokumente

Einige Erkenntnisse aus Interaction-Room-Workshops entstehen obligatorisch, andere optional. Die obligatorischen Erkenntnisse entstehen durch die Verwendung von Landkarten und Annotationen, wie sie in den Kapiteln 5.2 und 5.3 beschrieben wurden. Im Praxisbeispiel (vgl. Kapitel 5.2.5 und Kapitel 5.3.6) wurde bereits deutlich, dass zusätzliche Informationen auf den Landkarten dokumentiert und durch die Annotationen erhoben werden. Diese fließen ebenfalls in das Dokument ein und zählen zu den optionalen Erkenntnissen. Weil die Zusatzinformationen nicht immer verfügbar sind, wird in den folgenden Kapiteln angegeben, welche Artefakte optional und nur bei Verfügbarkeit in das Ergebnisdokument einfließen. Abbildung 38 zeigt in einem Überblick die zuvor erläuterte Dokumentstruktur und wo darin die entsprechenden Artefakte aus Interaction-Room-Workshops dokumentiert werden.

<b>Dokumentstruktur</b>	<b>Artefakt aus dem Interaction Room</b>
<b>1. Projektvision</b>	
1.1 Ausgangssituation	Vorabfragebogen, Notizen aus Vorgesprächen, analysierte Dokumente
1.2 Ziele und Vision	Projektziel, Workshop-Ziele
1.3 Stakeholder	Stakeholder-Analyse
1.4 Struktur des Dokuments	
<b>2. Überblick</b>	
2.1 Fachlicher Systemkontext*	Feature-Landkarte, Prozesslandkarten, Rollen, Systeme, Annotationen
2.2 Technischer Systemkontext	Integrationslandkarte, Annotationen
2.3 Fachliches Objektmodell	Objektlandkarte, Annotationen
2.4 Navigationsstruktur**	Prozesslandkarten, analysierte Dokumente
2.5 Zustandsdiagramm zentraler Geschäftsobjekte**	Prozesslandkarten
2.6 Rollen- und Berechtigungskonzept**	Prozesslandkarten, Objektlandkarte
<b>3. Detaillierte Anforderungen</b>	
3.1 Je Use-Case eine detaillierte Anforderung	Prozesslandkarten, Annotationen
<b>4. Anhang</b>	
	Feature-Landkarte, Annotationsdokumentation, Fotos, Glossar

\* Sofern keine Rollen und Systeme in den Prozesslandkarten dokumentiert wurden, muss diese Information nachträglich ergänzt werden.

\*\* Optionaler Bestandteil

Abbildung 38: Abbildung von Artefakten aus Interaction-Room-Workshops in eine Dokumentstruktur



#### 5.4.2.1 Projektvision

Im ersten Kapitel des Anforderungsdokumentes wird zunächst die Ausgangslage beschrieben. Diese Information existiert durch die Erkenntnisse aus den Vorabfragebögen, persönlichen Gesprächen und analysierten Dokumenten. Das Projektziel und die Ziele der Workshops werden verwendet, um den intendierten Zielzustand nach Umsetzung des Projektes zu beschreiben. Die Stakeholder des Projekts werden im Methodenbaustein Vorbereitung durch die Stakeholder-Analyse erhoben und kurz beschrieben. Das Kapitel schließt mit einer Darstellung der Struktur des Dokuments.

#### 5.4.2.2 Fachlicher Systemkontext

Aus den Epics und Features der Feature-Landkarte, Prozesslandkarten und der Information der Annotationen wird ein Use-Case-Diagramm erstellt, um den Überblick der funktionalen Anforderungen in Kapitel 2 zu dokumentieren. Um die Use Cases zu identifizieren, müssen die Informationen aus der Feature-Landkarte analysiert und konsolidiert bzw. dekomponiert werden. Es ist nicht davon auszugehen, dass eine eindeutige Beziehung zwischen Epics und Use Cases besteht. Außerdem kann die zusätzliche Information aus den Prozesslandkarten und Annotationen dazu führen, dass Features neu gruppiert werden und entsprechende Use Cases entstehen. Sofern auf den Prozesslandkarten Systeme und Rollen an Aktivitäten notiert wurden, können menschliche und technische Akteure an Use Cases identifiziert und das Use-Case-Diagramm vollständig dargestellt werden. Fehlt diese Information, muss diese nachträglich beschafft werden. Jeder Use Case wird kurz textuell beschrieben.

#### 5.4.2.3 Technischer Systemkontext

Das Use-Case-Diagramm ist eine mögliche Darstellung des Systemkontextes, die auch technische Akteure darstellt, jedoch von der Flussrichtung und der Art transportierter Informationen abstrahiert. Diese Information wird in Interaction-Room-Workshops mit Hilfe der Integrationslandkarte erhoben, welche das Modell der technischen Integration in Kapitel 2 des Anforderungsdokumentes darstellen und das zentrale Informationssystem im Kontext der Anwendungslandschaft einer Organisation zeigt. Wenn die Integrationslandkarte annotiert wurde, sind zusätzliche Anforderungen der technischen Integration formuliert und dokumentiert worden. Abzulösende Schnittstellen und Systeme, notwendige Integrationsaufwände externer Dienste und unveränderliche Umsysteme können solche Anforderungen darstellen. Diese Information wird im Text zur Beschreibung der technischen Information verarbeitet. Dabei kann direkter Bezug zu einzelnen Annotationen genommen werden oder die Information wird implizit verarbeitet.

#### 5.4.2.4 Fachliches Objektmodell

Mit der Objektlandkarte werden die relevanten Geschäftsobjekte in Bezug zueinander gesetzt. Dadurch wird die fachliche Struktur der Anwendung beschrieben. Während die Geschäftsobjekte der Objektlandkarte als einzelne Entitäten skizziert wurden, werden für die Erstellung des fachlichen Objektmodells Objekte zusammengefasst und durch entsprechende Attribute verfeinert. Durch die Aggregation von Objekten zu Attributen wird die Beschreibung der fachlichen Struktur technischer, was sie zur Grundlage eines objektorientierten Systementwurfs werden lässt. Verwendete Annotationen werden, wie bereits im technischen Systemkontext, in die textuelle Beschreibung des Modells integriert. Das fachliche Objektmodell ist Bestandteil von Kapitel 2 des Anforderungsdokumentes.

#### 5.4.2.5 Navigationsstruktur

Eine Navigationsstruktur – diese ist ebenfalls Bestandteil des zweiten Kapitels im Anforderungsdokument – kann durch Informationen aus Interaction-Room-Workshops erstellt werden, indem die funktionalen Anforderungen auf Oberflächen verteilt werden. Die Erfahrung hat gezeigt, dass Experten Anwendungen häufig in Masken beschreiben, weil ihnen diese Repräsentation eines Systems geläufig ist. Daher werden Anforderungsdokumente von Experten häufig entlang der Masken geschnitten. Liegt diese Information vor, kann, gemeinsam mit den übrigen Informationen aus den Workshops, eine Navigationsstruktur der Anwendung durch ein Zustandsdiagramm erstellt und dokumentiert werden. Funktionale Anforderungen werden auf Masken verteilt und Maskenübergänge mit Hilfe von Triggern, Guards und Aktivitäten beschrieben. Falls diese Information nicht vorliegt, kann eine Navigationsstruktur entworfen und unter entsprechender Kennzeichnung vorgeschlagen werden.

#### 5.4.2.6 Zustände zentraler Geschäftsobjekte

Zustandsautomaten sind ein optionaler Bestandteil des zweiten Kapitels im Anforderungsdokument und werden aus den Zuständen zentraler Geschäftsobjekte an den Aktivitäten der Prozesslandkarten abgeleitet. Über die Information der Prozesslandkarten über Aktivitäten, Entscheidungen und entsprechende Bedingungen existieren bereits Informationen, die zur Erzeugung eines Zustandsdiagramms mit Triggern, Guards und Aktivitäten (vgl. [249]) benötigt werden.

#### 5.4.2.7 Rollen- und Berechtigungskonzept

Sofern Rollen als Geschäftsobjekte angesehen und in der Objektlandkarte skizziert wurden, kann über die Rollen und deren Beziehungen zu Geschäftsobjekten sowie dem Wissen aus dem Use-Case-Diagramm und den Prozesslandkarten ein Grundstein für ein Rollen- und Berechtigungskonzept erarbeitet werden. Dieses ist ebenfalls in Kapitel 2 des Anforderungsdokumentes zu finden. Die Information aus Interaction-Room-Workshops genügt in der Regel nicht, um ein vollständiges und detailliertes Konzept zu erstellen, jedoch um eine initiale Version als Ausgangsbasis zur weiteren Verfeinerung zu erstellen.

#### 5.4.2.8 Detaillierte Soll-Anforderungen

Die detaillierten Soll-Anforderungen werden im dritten Kapitel des Dokuments dargestellt. Jeder Use Case aus dem Use-Case-Diagramm wird in einem eigenen Unterkapitel detailliert beschrieben. Führend bei der Beschreibung sind die Prozesslandkarten: Mit Hilfe der Prozesslandkarten, Annotationen, notierten Systeme, Rollen und Zustände von Geschäftsobjekten an den Aktivitäten der Prozesslandkarten werden die detaillierten funktionalen und nicht funktionalen Soll-Anforderungen dokumentiert. Dabei ist die Abbildung von Prozesslandkarten auf Use Cases nicht eindeutig, denn die Prozesslandkarten entstehen auf Grundlage der Feature-Landkarte. Jede Prozesslandkarte eines Use Cases wird modelliert und zusammen mit allen weiteren verfügbaren Informationen textuell beschrieben.

#### 5.4.2.9 Anhang

Im Anhang des Anforderungsdokumentes werden die Feature-Landkarte und die vollständige Dokumentation der Annotationen dokumentiert. Zusätzlich können Fotos von Landkarten aus Workshops eingefügt werden, was insbesondere dann sinnvoll ist, wenn die Landkarten im Überblickskapitel und im Kapitel zu den detaillierten Anforderungen verändert wurden. Zusätzlich kann aus der Objektlandkarte ein Glossar abgeleitet werden, in dem die Bedeutung der Geschäftsobjekte definiert wird. Das Glossar ist insbesondere für Leser, die keine Stakeholder in einem Workshop waren, nützlich, weil das Glossar die verwendete Terminologie in den Workshops wiedergibt.

## 6 Evaluation

Im Zeitraum von November 2011 bis Februar 2017 wurden 23 Industrieprojekte mit dem Interaction Room durchgeführt. Dabei wurde die in Kapitel 5 beschriebene Methode entwickelt. Das erfahrungsbasierte Vorgehen zur Entwicklung und Evolution der Methode wird hier anhand der gesammelten Daten aus den Industrieprojekten überprüft. Ziel der Überprüfung ist es, festzustellen, ob sich die gewonnene Erfahrung in den Daten widerspiegelt. Bevor im Folgenden drei Forschungsfragen formuliert werden (vgl. Kapitel 6.1.1), die Statistik zur Überprüfung der Forschungsfragen erläutert wird (vgl. Kapitel 6.1.2), die Ergebnisse der Überprüfung präsentiert (vgl. Kapitel 6.1.3) und diskutiert (vgl. Kapitel 6.1.4) werden, wird ein kurzer Überblick über die Projekte gegeben.

Die Projekte unterscheiden sich in ihrer Anwendungsdomäne und ihrem Typ. Von den 23 Projekten waren 26 % Analyseprojekte, in denen das Ziel die Untersuchung einer fachlichen Fragestellung war. Es wurde jedoch kein Informationssystem abgelöst, neuentwickelt, weiterentwickelt oder eingeführt. 17 % waren Neuentwicklungsprojekte und 22 % waren Weiterentwicklungsprojekte: Während in Weiterentwicklungsprojekten inkrementell Funktionalität angepasst, erweitert oder hinzugefügt wird, wurden Projekte als Neuentwicklung klassifiziert, deren Ziel die Entwicklung eines neuen Informationssystems war. Die übrigen 35 % waren Migrationsprojekte, in denen ein Legacy-System durch ein anderes System abgelöst wurde (vgl. Abbildung 39).

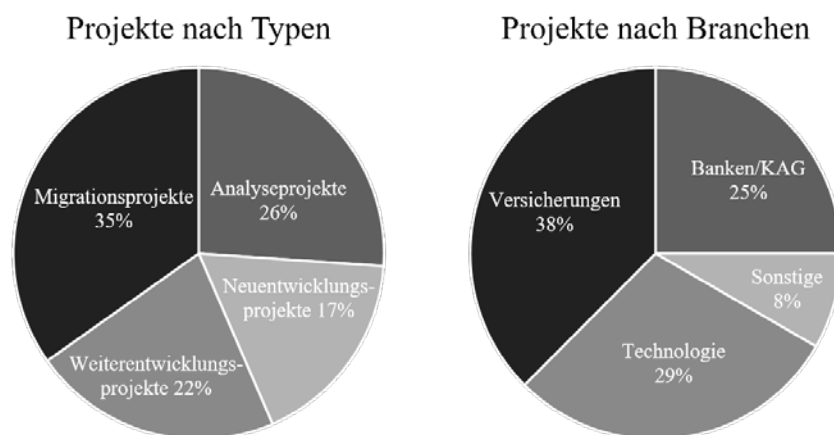


Abbildung 39: Durchgeführte Projekte nach Typen und Branchen

35 % der Projekte wurde mit Versicherungsunternehmen durchgeführt, ein Viertel der Projekte wurde mit Banken und Kapitalanlagegesellschaften (KAG) durchgeführt und 29 % der Projekte mit Technologieunternehmen. Die restlichen 8 % wurden mit Unternehmen anderer Branchen durchgeführt.

Die durchgeführten Projekte haben sich nicht nur in Branche und Typ unterschieden, sondern hatten jeweils auch unterschiedliche Ausgangslagen, und es wurden unterschiedliche Zielsetzungen durch die Projektpartner formuliert. Tabelle 14 zeigt eine Auswahl.

Tabelle 14: Ausgangslagen und Ziele durchgeführter Projekte (Auszug)

Ausgangslage	Ziel
Die Betreuung und Verwaltung von Vertriebsagenturen erfolgt über viele historisch gewachsene Host-Anwendungen. Die Anwendungen besitzen keine gemeinsame Datenbasis, kein detailliertes Rollen- und Berechtigungskonzept und erzeugen hohen manuellen Verwaltungsaufwand.	Erhebung der Anforderungen an die Ablösung der Host-Anwendungen, um Medienbrüche zu reduzieren, den Automatisierungsgrad zu erhöhen und eine geführte, jedoch flexible Bearbeitung der Prozesse durch die Mitarbeiter zu realisieren.
Es besteht eine hohe Abhängigkeit von einem Druckdienstleister, da Teile der Business-Logik für die Output-Steuerung beim Dienstleister implementiert sind.	Erarbeitung von Migrationsalternativen zur Reduktion der Abhängigkeit vom Dienstleister.
Entwicklung eines Produkts zur Leistungsabrechnung in der privaten Krankenversicherung.	Die minimalen Anforderungen an das zukünftige Produkt erheben und bewerten.
Eine Anwendung für Bausparen, Darlehensvergabe und -verwaltung ist veraltet und soll durch ein Standardprodukt abgelöst werden.	Beurteilung der Anpassungsnotwendigkeit des Standardproduktes für dessen Einführung und zur Migration der Bestandsdaten.
Ein Versicherungsunternehmen vertreibt bereits Produkte über das Internet, jedoch werden Policen nur über Postweg versendet.	Analyse der notwendigen Änderungen in der Anwendungslandschaft zur Realisierung des digitalen Versandes von Versicherungspolicen.
Die Beiträge von Versicherungen werden ausschließlich über ein Lastschriftverfahren eingezogen.	Anforderungen für die Einführung von PayPal und Kreditkarten als Zahlungsmittel erheben.
Mehrere Abteilungen sind unzufrieden mit der Funktionalität einer Anwendung zur Projektplanung- und -steuerung.	Anforderungen zur Steigerung der Benutzerzufriedenheit erheben und priorisieren, um daraus eine Handlungsempfehlung für die Weiterentwicklung der Anwendung abzuleiten.

In den Projekten wurden 48 Workshops an 71 Workshop-Tagen durchgeführt. An den 71 Workshop-Tagen wurden insgesamt 195 Landkarten erstellt. Die 195 Landkarten verteilen sich auf

31 Feature-Landkarten, 124 Prozesslandkarten, 20 Objektlandkarten und 20 Integrationslandkarten. Auf den 195 Landkarten wurden 1629 Annotationen verwendet, wovon 524 Annotationen auf der Feature-Landkarte, 789 Annotationen auf der Prozesslandkarte, 90 Annotationen auf der Objektlandkarte und 226 Annotationen auf der Integrationslandkarte verwendet wurden (vgl. Tabelle 15).

Tabelle 15: Anzahl der Landkarten und Annotationen je Landkarte

	Feature-Landkarte	Prozesslandkarte	Objektlandkarte	Integrationslandkarte	Summe
Anzahl Landkarten	31	124	20	20	195
Anzahl Annotationen	524	789	90	226	1629

## 6.1 Forschungsfragen

Um die gesammelten Erfahrungen aus den 23 Projekten besser verstehen und wahrgenommene Zusammenhänge überprüfen zu können, werden die Projektdaten im Hinblick auf drei Forschungsfragen untersucht.

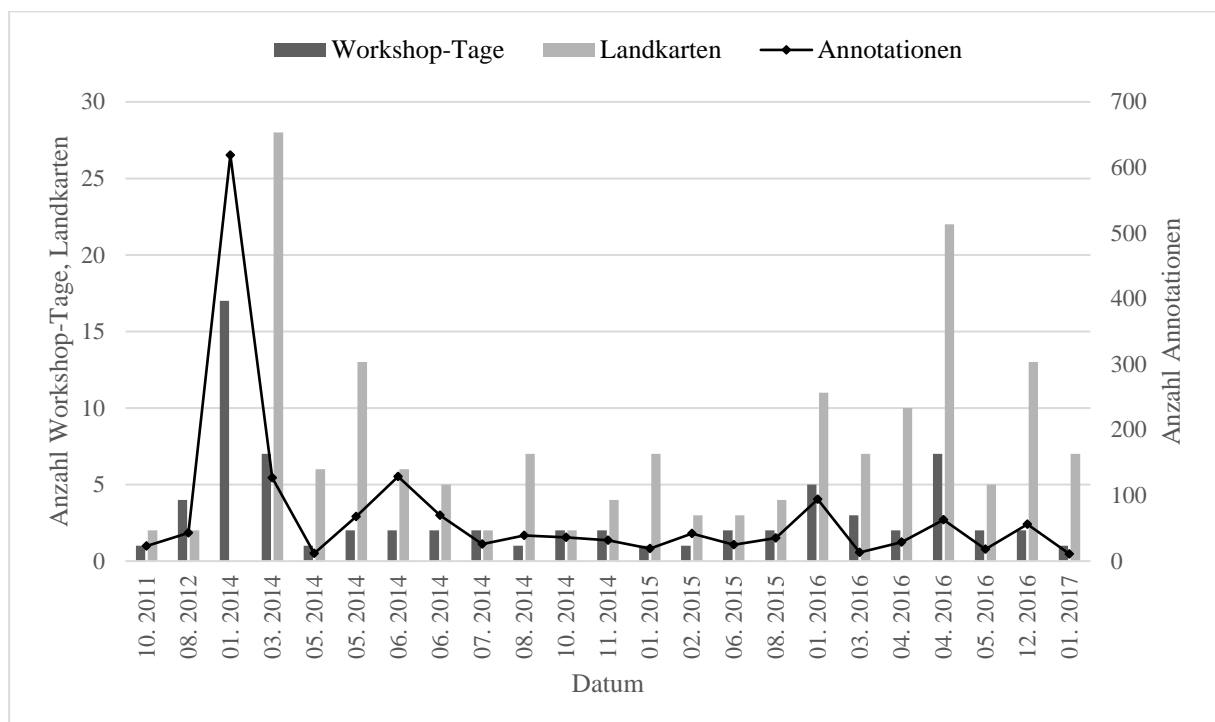


Abbildung 40: Workshop-Tage, Landkarten und Annotationen je Projekt

Abbildung 40 zeigt die Anzahl der Workshop-Tage, Landkarten und Annotationen je Projekt im chronologischen Verlauf. Die Größe eines Projektes wird durch die Anzahl der Workshop-

Tage, die Anzahl der verwendeten Landkarten und die Anzahl der verwendeten Annotationen definiert. Es ist zu erwarten, dass die unterschiedlichen Metriken der Projektgröße einen positiven Zusammenhang besitzen. Je mehr Zeit aufgewendet wird, desto mehr Landkarten können erstellt und desto mehr Annotationen können verwendet werden. Die Daten in Abbildung 40 zeigen jedoch, dass zu Beginn des Auswertungszeitraums (Oktober 2011), im Vergleich zur Dauer und Anzahl der Landkarten, mehr Annotationen verwendet wurden als gegen Ende des Auswertungszeitraums (Januar 2017). Gegen Ende des Auswertungszeitraums wurden weniger Annotationen, bei einer vergleichbaren Anzahl Landkarten und Workshop-Tagen, verwendet. Um festzustellen, ob ein Zusammenhang zwischen der Anzahl der Workshop-Tage, der Anzahl der Landkarten und der Anzahl der verwendeten Annotationen existiert, wurde die folgende Forschungsfrage formuliert:

**Forschungsfrage 1:** Besteht ein Zusammenhang zwischen den Metriken der Projektgröße?

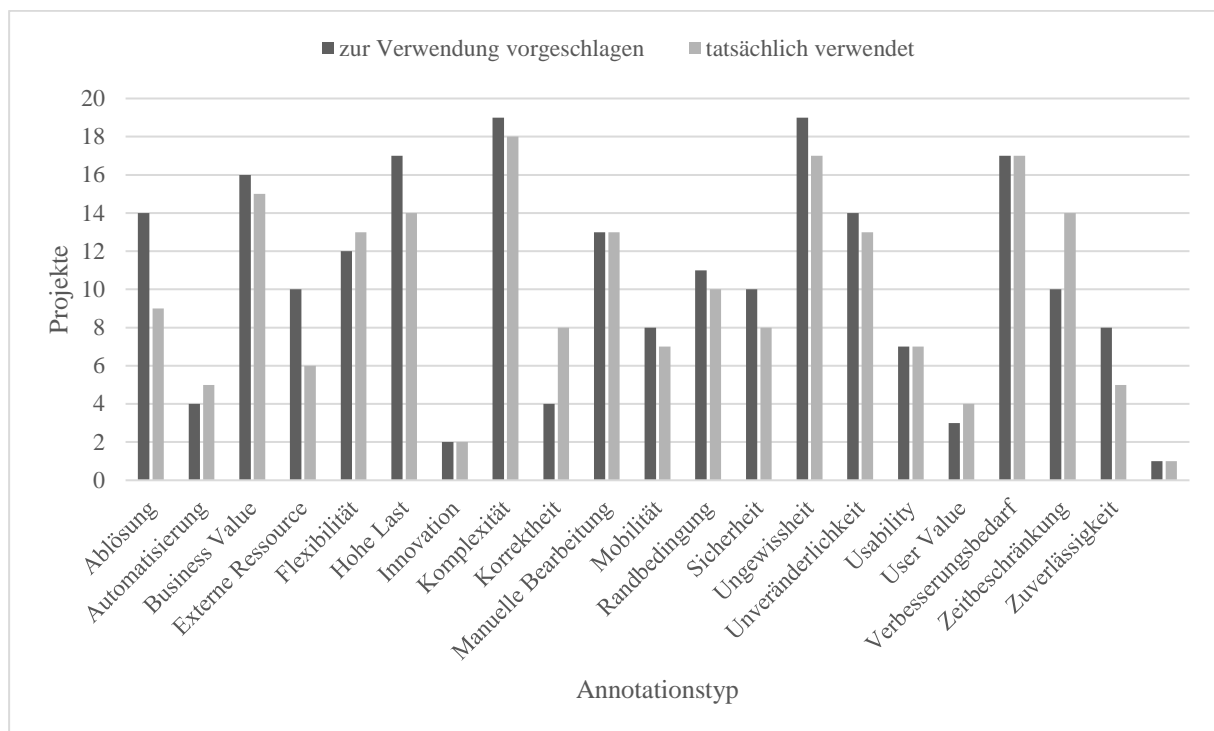


Abbildung 41: Zur Verwendung vorgeschlagene und verwendete Annotationstypen je Projekt

Der IR-MC und der IR-DC bereiten IR-Workshops vor, indem alle notwendigen Informationen zusammengetragen werden, um die Landkarten und Annotationstypen zur Erreichung des Workshop-Ziels auszuwählen (vgl. Kapitel 5.1). Abbildung 41 stellt die Anzahl der Projekte dar, in denen ein Annotationstyp zur Verwendung vorgeschlagen und tatsächlich verwendet wurde. Die Grafik zeigt, dass elf Annotationstypen öfter zur Verwendung vorgeschlagen als verwendet wurden, vier Annotationstypen wurden immer verwendet, wenn sie zur Verwendung

vorgeschlagen wurden und fünf Annotationstypen wurden häufiger verwendet, als sie zur Verwendung vorgeschlagen wurden. Wenn ein Annotationstyp häufiger verwendet wurde, als er zur Verwendung vorgeschlagen wurde, bedeutet dies, dass die Stakeholder eigenmächtig Annotationstypen verwendet haben. Stakeholder können Annotationstypen kennen, wenn sie zur Verwendung auf anderen Landkarten vorgeschlagen wurden, durch eine im physischen Raum aufgehängte Übersicht oder durch vorherige Erfahrung mit der Methode. Wenn ein Annotationstyp trotz Vorschlag nicht verwendet wurde, existiert entweder keine überdurchschnittliche Herausforderung dieses Typs, was gleichbedeutend mit einer inkorrekten Auswahl der Annotationstypen durch die IR-Coaches sein kann. Dies birgt die Gefahr, dass andere, unter Umständen relevante Annotationstypen, nicht zur Verwendung angeboten wurden, denn die Menge der Annotationstypen ist auf  $7 \pm 2$  je Annotationsset begrenzt. Um herauszufinden, ob ein Zusammenhang zwischen dem Vorschlag zur Verwendung von Annotationstypen und der tatsächlichen Verwendung existiert, wurde folgende Forschungsfrage formuliert:

**Forschungsfrage 2:** Besteht ein Zusammenhang zwischen zur Verwendung vorgeschlagenen und verwendeten Annotationstypen?

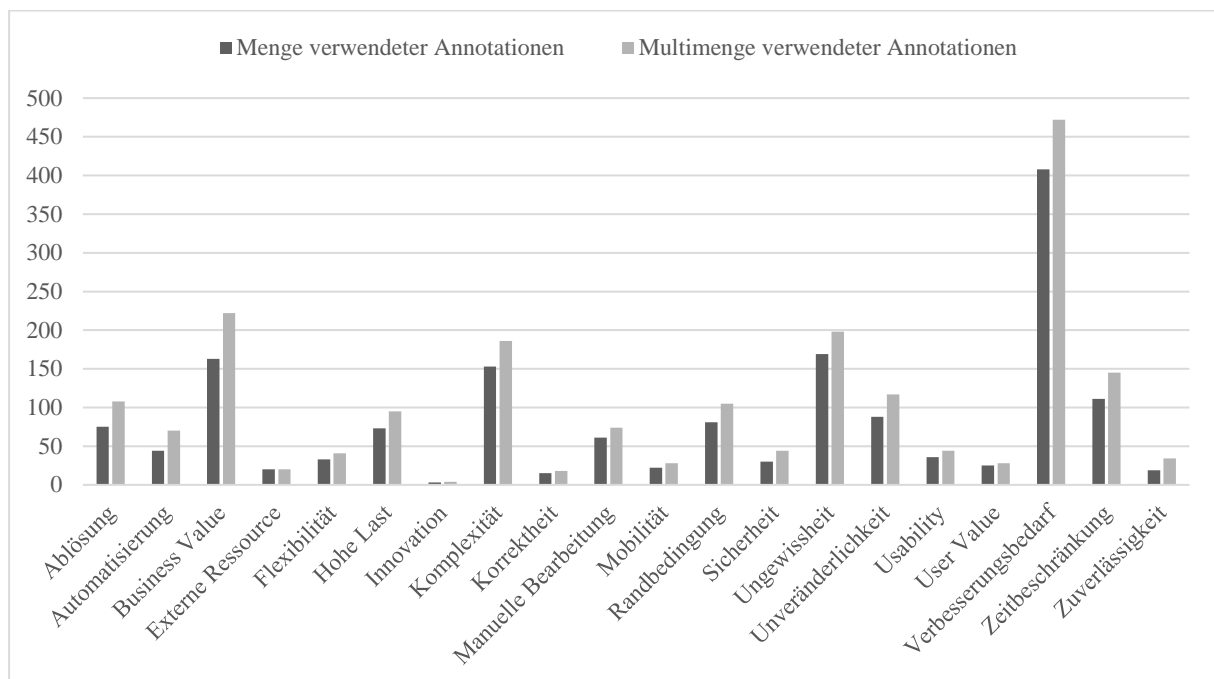


Abbildung 42: Menge und Multimenge verwendeter Annotationen

Um die Effizienz der Methode zu messen, müssten viele unterschiedliche Metriken kombiniert werden. Beispielsweise kann die Dauer der Erstellung einer Landkarte in Bezug zu den Landkartenelementen einen Hinweis auf die Effizienz der Moderation liefern. Je mehr Landkartenelemente je Zeiteinheit skizziert werden, desto größer ist der inhaltliche Fortschritt. Einer rein



quantitativen Betrachtung würde diese These vermutlich standhalten. Jedoch kann der qualitative Fortschritt nicht gemessen werden, denn der vermeintlich geringe Fortschritt durch die Skizzierung eines Elements kann zur Klärung eines Missverständnisses beigetragen haben, das unerkannt im späteren Projektverlauf zu einem Risiko geworden wäre. Eine weitere Metrik der Effizienz kann die Dauer der Diskussion und Dokumentation einzelner Annotationstypen sein. Je weniger Zeit benötigt wird, desto effizienter wurde die Methode durchgeführt. Auch hier kann die rein quantitative Betrachtung stimmen. Es kann jedoch auch bedeuten, dass nur triviale Aspekte geäußert wurden, deren Erklärung und Diskussion wenig Zeit in Anspruch nimmt, wodurch die Intention der Annotationen verfehlt worden wäre.

Das Stoppen der Rededauer oder die Aufzeichnung von Diskussionen über Landkartenelemente und Annotationen hätte einerseits die Zustimmung der Stakeholder erfordert und andererseits die Gefahr mit sich gebracht, dass Stakeholder sich beobachtet fühlen, nicht natürlich handeln und nicht offen kommunizieren. Selbst wenn dieses Einverständnis eingeholt worden wäre und es zu keiner Verzerrung gekommen wäre, hätte durch diese Daten die effiziente Durchführung der Methode nicht zweifelsfrei beurteilt werden können. Die einzelnen Parameter hätten in der Praxis nicht isoliert werden können, Umwelteinflüsse hätten nicht kontrolliert werden können und das indeterministische Verhalten von Menschen (in Gruppen) wäre neben der Identifikation und Erhebung aller relevanten Kennzahlen notwendig gewesen. Stattdessen wurden vorliegende Projektdaten analysiert, die ebenfalls Bestandteil Hinweise auf eine effiziente Methodendurchführung liefern können.

Annotationen werden projektweit eindeutig durch die Kombination von Annotationstyp, Grund und ID identifiziert. Annotationstypen mit gleicher ID zählen als Duplikat (vgl. Kapitel 5.3.5). Abbildung 42 stellt die Menge und die Multimenge der verwendeten Annotationen je Annotationstyp gegenüber. Unter der Verwendung von fast jedem Annotationstyp lassen sich Duplikate finden, die jedoch unterschiedliche Erkenntnis für das Projekt stiften. Duplikate können (1) auf derselben Landkarte, aber an unterschiedlichen Landkartenelementen, (2) auf unterschiedlichen Landkarten oder (3) an demselben Landkartenelement heften. Während (1) und (2) einen Erkenntnisgewinn liefern, weil durch sie zwar derselbe Aspekt, aber an anderer Stelle gekennzeichnet wird, liefern Duplikate des Typen (3) keine Erkenntnis, sondern bedeuten unnötigen Mehraufwand, weil jede Annotation einzeln diskutiert werden muss, um herauszufinden, dass es sich um ein Duplikat handelt. Auch die Annahme, dass Duplikate des Typs (3) einen besonders wichtigen Aspekt kennzeichnen, denn schließlich wurde dieser Aspekt von mehreren Stakeholdern genannt, ist nicht konform zur Intention der Annotationen. Je mehr

Stakeholder einen Aspekt an einem Landkartenelement erkennen, desto präsenter und besser verstanden ist dieser Aspekt. Zur Überprüfung, ob ein Zusammenhang zwischen der verwendeten Menge der Annotationen und Duplikaten besteht, wurde die folgende Forschungsfrage formuliert:

**Forschungsfrage 3:** Besteht ein Zusammenhang zwischen der Menge verwendeter Annotationen und der Anzahl von Duplikaten?

## 6.2 Statistik zur Überprüfung der Forschungsfragen

Um die Forschungsfragen zu beantworten, wurden die Daten der 23 Projekte im Hinblick auf Korrelationen zwischen mindestens zwei Variablen untersucht. In Abhängigkeit der Beschaffenheit und Verteilung der Daten werden unterschiedliche Testverfahren angewendet.

### 6.2.1 Korrelation nach Bravais-Pearson

Die folgenden Erläuterung basieren auf den Ausführungen in [250]. Sofern Daten mindestens intervallskaliert sind, bestimmt der Korrelationskoeffizient  $r$  nach Bravais-Pearson die Stärke des linearen Zusammenhangs zweier Variablen. Um den Korrelationskoeffizienten von zwei Variablen  $x$  und  $y$  zu berechnen, wird die Kovarianz  $\sigma_{xy}$  verwendet. Die Kovarianz misst die Streuung zwischen den Wertebelegungen der Variablen um den durchschnittlichen Wert  $\bar{x}$  und  $\bar{y}$ . Um die Kovarianz von  $n$  Wertebelegungen von  $x$  und  $y$  zu berechnen, wird Formel 1 verwendet.

Formel 1: Berechnung der Kovarianz

$$\sigma_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}) * (y_i - \bar{y})$$

Um eine Aussage über die Stärke des Zusammenhangs treffen zu können, wird die Kovarianz auf den Wertebereich zwischen -1 und 1 normiert, was dem Korrelationskoeffizienten  $r$  entspricht (vgl. Formel 2).

Formel 2: Korrelationskoeffizient  $r$  nach Bravais-Pearson

$$r = \frac{\sigma_{xy}}{\sigma_x * \sigma_y}$$

Der Korrelationskoeffizient ist also auf  $-1 \leq r \leq 1$  normiert, das Vorzeichen bestimmt die Richtung des Zusammenhangs, und der Betrag von  $r$  dessen Stärke. Nimmt  $r$  einen positiven Wert an, wird der Wert einer Variablen  $y$  größer, wenn der Wert der Variablen  $x$  größer wird. Dieser positive Zusammenhang wird auch als gleichläufig bezeichnet, sofern  $r < 0$  ist, wird von einem gegenläufigen oder negativen Zusammenhang gesprochen, bei dem  $y$  tendenziell kleiner wird, sofern  $x$  größer wird.

Je näher der Betrag von  $r$  an 1 liegt, desto geringer ist die Streuung der Variablen um die Regressionsgerade. Um die Stärke qualitativ einzuordnen, kann die Effektstärke nach Cohen [251] verwendet werden:

- $r = \pm 0,10$  entspricht einem schwachen Effekt
- $r = \pm 0,30$  entspricht einem mittleren Effekt
- $r = \pm 0,50$  entspricht einem starken Effekt

Um den Einfluss einer Variablen auf eine andere Variable zu bestimmen und damit deren Abhängigkeit, wird das Bestimmtheitsmaß verwendet, das im Folgenden an [250] angelehnt erläutert wird. Die Abhängigkeit zwischen zwei Variablen  $x$  und  $y$  kann bestimmt werden, indem überprüft wird, ob die Abweichung der Werte von  $y$  vom durchschnittlichen Wert  $\bar{y}$  durch die Werte von  $x$  verursacht werden (vgl. Abbildung 43).

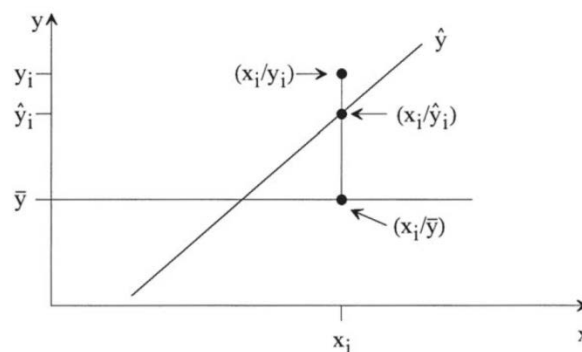


Abbildung 43: Grafische Darstellung zur Erläuterung der Herleitung des Bestimmtheitsmaßes [250]

Bei einer Variablenbelegung  $i$  weicht  $y_i$  um  $y_i - \bar{y}$  vom durchschnittlichen Wert  $\bar{y}$  ab. Um die Abweichung zu erklären, kann die Regression und  $x$  verwendet werden, denn für ein  $x_i$  wird ein  $\hat{y}_i$  erwartet. Daher kann ein Teil der Abweichung ( $\hat{y}_i - \bar{y}$ ) durch die Regression bestimmt werden und ein Teil ( $y_i - \hat{y}_i$ ) nicht. Um den Zusammenhang von zwei Variablen ermitteln zu können, muss die Abweichung aller Merkmalsausprägungen in die Berechnung einfließen.

Damit sich die bipolaren Abweichungen nicht gegenseitig aufheben, werden sie quadriert (vgl. Formel 3).

Formel 3: Berechnung des Bestimmtheitsmaßes  $B^2$

$$B^2 = \frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2}$$

Das Bestimmtheitsmaß liegt zwischen  $0 \leq B^2 \leq 1$ . Sofern ein Zusammenhang linear ist, ist das Bestimmtheitsmaß gleich dem quadrierten Korrelationskoeffizienten nach Bravais-Pearson (vgl. Formel 4).

Formel 4: Bestimmtheitsmaß eines linearen Zusammenhangs

$$B^2 = r^2$$

Durch die Größe von  $r^2$  kann eine Aussage über den Grad des Einflusses einer Variablen  $x$  auf eine abhängige Variable  $y$  bestimmt werden. Je größer  $r^2$  ist, desto größer ist der Einfluss. Das Bestimmtheitsmaß wird verwendet, um Vorhersagen treffen zu können, wie sich die abhängige Variable verhält, wenn die unabhängige Variable sich verändert. Ist beispielsweise ein stark positiver Zusammenhang zwischen  $x$  und  $y$  gegeben und  $r^2 = 1$ , bedeutet dies, dass keine Abweichung im beobachteten Zusammenhang existiert.

### 6.2.2 Kontingenzanalyse mit dem Pearson-Chi-Quadrat-Test

Sofern die Belegung von zwei Variablen  $x$  und  $y$ , deren Zusammenhang untersucht werden soll, nicht intervallskaliert ist, kann der Bravais-Pearson-Test nicht angewendet werden [252]. Stattdessen wird, sofern eine Variable nominalskaliert ist, der Zusammenhang mit Hilfe der Kontingenzanalyse überprüft [250]. Dazu werden Kreuztabellen verwendet, welche die Ausprägungen  $x_i$  und  $y_i$  gegenüberstellen und die Häufigkeit  $h$  des Auftretens jeder Kombination wird bestimmt.

Um die Stärke des Zusammenhangs zu berechnen, wird die relative Abweichung der tatsächlich auftretenden Häufigkeiten in Relation zu den theoretischen Häufigkeiten verwendet (vgl. Formel 5). Je größer das Ergebnis ist, desto stärker ist die Abhängigkeit.

Formel 5: Berechnung der relativen Abweichung tatsächlich auftretender und theoretischer Häufigkeiten

$$\frac{h_{ik} - \frac{h_i * h_k}{n}}{\frac{h_i * h_k}{n}}$$

Um den Kontingenzkoeffizienten zu bilden, werden alle Abweichungen der Kombinationen berücksichtigt. Damit sich gegenseitig aufhebende positive und negative Abweichungen im Quotienten nicht eliminieren, wird dieser quadriert, woraus sich Chi-Quadrat ergibt (vgl. Formel 6).

Formel 6: Berechnung von Chi-Quadrat als Vorbereitung zur Bestimmung des Kontingenzkoeffizienten

$$X^2 = \sum_{i=1}^v \sum_{k=1}^w \frac{\left(h_{ik} - \frac{h_i * h_k}{n}\right)^2}{\frac{h_i * h_k}{n}}$$

Aus dem Wert von Chi-Quadrat kann noch keine Aussage über die Stärke des Zusammenhangs abgeleitet werden, denn durch die Veränderung der Häufigkeiten der Ausprägungen verändert sich auch Chi-Quadrat (z. B. würde sich bei einer Verdopplung der Merkmalsausprägungen auch Chi-Quadrat verdoppeln, obwohl die Stärke des Zusammenhangs gleichgeblieben ist). Deswegen wird der Einfluss der Häufigkeiten durch den Kontingenzkoeffizienten  $K$  nach Pearson eliminiert (vgl. Formel 7).

Formel 7: Kontingenzkoeffizient nach Pearson

$$K = \sqrt{\frac{X^2}{X^2 + n}}$$

Bei vollständiger Unabhängigkeit ist  $K = 0$ , bei vollständiger Abhängigkeit ist  $K = K_{max}$ . Um einen normierten Wert zwischen 0 und 1 zu erhalten, wird der korrigierte Kontingenzkoeffizient  $K_{korr}$  verwendet (vgl. Formel 8).

Formel 8: Korrigierter Kontingenzkoeffizient nach Pearson

$$K_{korr} = \frac{K}{K_{max}} = \frac{\sqrt{\frac{X^2}{X^2 + n}}}{\sqrt{\frac{K^* - 1}{K^*}}} = \sqrt{\frac{X^2}{X^2 + n} * \frac{K^*}{K^* - 1}} \text{ mit } K^* = \min\{v, w\}$$

Je näher  $K_{korr}$  an 1 ist, desto stärker ist der Zusammenhang.

## 6.3 Ergebnisse

Um die Forschungsfragen zu beantworten zu können, wurden zunächst die Skalenniveaus der jeweiligen Datensätze untersucht. Alle Datensätze weisen metrisches Niveau auf und sind auf einer Verhältnisskala. Daraus resultiert, dass für die Beantwortung aller Forschungsfragen der Bravais-Pearson-Test verwendet werden kann.

### 6.3.1 Forschungsfrage 1: Besteht ein Zusammenhang zwischen den Metriken der Projektgröße?

Um die erste Forschungsfrage zu beantworten, wurde der Zusammenhang zwischen Workshop-Tagen und Landkarten, der Zusammenhang zwischen Workshop-Tagen und Annotationen sowie der Zusammenhang zwischen Landkarten und Annotationen untersucht. Dazu wurde jeweils der Korrelationskoeffizient  $r$ , das Bestimmtheitsmaß  $r^2$  und die statistische Signifikanz mittels des Bravais-Pearson-Tests bestimmt (vgl. Tabelle 16).

Die Korrelation zwischen Landkarten und Workshop-Tagen ist positiv ( $r = 0,766$ ) und, entsprechend der Effektstärke nach Cohen [251], mit einem stark signifikanten Effekt ( $p = 0$ ). Dabei zeigt  $r^2 = 0,586$ , dass in 58,6 % der Fälle die Anzahl der Landkarten mit der Anzahl der Workshop-Tage korreliert.

Die Korrelation zwischen Landkarten und Annotationen ist positiv ( $r = 0,624$ ) und, entsprechend der Effektstärke nach Cohen [251], mit einem stark signifikanten Effekt ( $p = 0,001$ ). Dabei zeigt  $r^2 = 0,389$ , dass in 38,9 % die Anzahl der Annotationen mit der Anzahl der Landkarten korreliert.

Die Korrelation zwischen Annotationen und Workshop-Tagen ist positiv ( $r = 0,914$ ) und, entsprechend der Effektstärke nach Cohen [251], mit einem stark signifikanten Effekt ( $p = 0$ ). Dabei zeigt  $r^2 = 0,835$ , dass in 83,5 % der Fälle die Anzahl der Annotationen mit der Anzahl der Workshop-Tage korreliert.

Tabelle 16: Korrelation zwischen Workshop-Tagen, Landkarten und Annotationen

		Landkarten	Annotationen	Tage
Landkarten	Pearson-Korrelation	1	,624	,766
	Bestimmtheitsmaß	1	,389	,586
	Sig. (2-seitig)		,001	,000
	N		23	23
Annotationen	Pearson-Korrelation	,624	1	,914
	Bestimmtheitsmaß	,389	1	,835
	Sig. (2-seitig)	,001		,000
	N	23		23
Tage	Pearson-Korrelation	,766	,914	1
	Bestimmtheitsmaß	,586	,835	
	Sig. (2-seitig)	,000	,000	
	N	23	23	

Der Zusammenhang zwischen den Metriken der Projektgröße existiert, besitzt jedoch unterschiedliche Varianz je Parameter. Das Projekt mit Beginn im Januar 2014 war mit Abstand das größte Projekt (17 Workshop-Tage und 619 verwendete Annotationen), wohingegen im nächst größten Projekt 129 Annotationen in sieben Workshop-Tagen verwendet wurden. Um den Einfluss des größten Projekts festzustellen, wurde der Bravais-Pearson-Test für die 22 Projekte exklusive des größten Projekts erneut durchgeführt (vgl. Tabelle 17).

Tabelle 17: Korrelation zwischen Workshop-Tagen, Landkarten und Annotationen exklusive des größten Projekts

		Landkarten	Annotationen	Tage
Landkarten	Pearson-Korrelation	1	,574	,758
	Bestimmtheitsmaß	1	,329	,574
	Sig. (2-seitig)		,005	,000
	N		22	22
Annotationen	Pearson-Korrelation	,574	1	,575
	Bestimmtheitsmaß	,329	1	,331
	Sig. (2-seitig)	,005		,005
	N	22		22
Tage	Pearson-Korrelation	,758	,575	1
	Bestimmtheitsmaß	,574	,331	1
	Sig. (2-seitig)	,000	,005	
	N	22	22	

Während sich die Stärke des Zusammenhangs und das Bestimmtheitsmaß den Zusammenhang von Landkarten und Annotationen sowie Landkarten und Workshop-Tage geringfügig

verändern, reduziert sich die Stärke des Zusammenhangs zwischen den Workshop-Tagen und den Annotationen deutlich, besitzt mit  $r = 0,575$  jedoch einen starken Effekt. Die Signifikanz aller Tests liegt im Konfidenzintervall, sodass der Zusammenhang zwischen den Metriken der Projektgröße existiert.

### 6.3.2 Forschungsfrage 2: Besteht ein Zusammenhang zwischen zur Verwendung vorgeschlagenen und verwendeten Annotationstypen?

Zur Beantwortung der zweiten Forschungsfrage wurde auf eine Korrelation zwischen der Anzahl der Projekte, in denen ein Annotationstyp zur Verwendung vorgeschlagen und in denen ein Annotationstyp verwendet wurde, geprüft. Ein positiver Zusammenhang würde bedeuten, dass den Vorschlägen der IR-Coaches gefolgt wird, ein negativer Zusammenhang würde bedeuten, dass entgegen der Vorschläge der IR-Coaches gehandelt wird. Da sich die Anzahl der Projekte auf einer Verhältnisskala befindet, wurde ein Bravais-Pearson-Test angewendet.

Zwischen der Anzahl der Projekte, in denen ein Annotationstyp vorgeschlagen wurde und der Anzahl der Projekte, in denen ein Annotationstyp verwendet wurde, besteht ein sehr starker positiver und statistisch signifikanter Zusammenhang ( $r = 0,902$ ;  $p = 0$ ). Das Bestimmtheitsmaß ( $r^2 = 0,813$ ) zeigt, dass mehr als 80 % der Verwendungen durch die Vorschläge erklärt werden (vgl. Tabelle 18).

Tabelle 18: Korrelation zwischen zur Verwendung vorgeschlagenen und verwendeten Annotationstypen

		zur Verwendung vorgeschlagene Annotationstypen	verwendete Annotationstypen
zur Verwendung vorgeschlagene Annotationstypen	Pearson-Korrelation	1	,902
	Bestimmtheitsmaß	1	,813
	Sig. (2-seitig)		,000
	N		23
verwendete Annotationstypen	Pearson-Korrelation	,902	1
	Bestimmtheitsmaß	,813	1
	Sig. (2-seitig)	,000	
	N	23	

Die vorherige Analyse berücksichtigt nicht, ob der Vorschlag zur Verwendung und die Verwendung einzelner Annotationstypen im selben Projekt stattgefunden haben. Es besteht also die Möglichkeit, dass eine Annotation in einem Projekt zur Verwendung vorgeschlagen wurde, jedoch in einem anderen Projekt, ohne vorherigen Vorschlag, genutzt wurde. Um diesen



Umstand zu berücksichtigen, wurde ein Pearson-Chi-Quadrat-Test auf Ebene der Annotations-typen durchgeführt. Dabei werden die einzelnen Merkmalsausprägungen (vorgesehen/genutzt, nicht vorgesehen/genutzt, vorgesehen/nicht genutzt, nicht vorgesehen/nicht genutzt) je Projekt in einer Kreuztabelle gegenübergestellt und untersucht, ob der Zusammenhang je Annotations-typ je Projekt statistisch signifikant ist.

Tabelle 19 zeigt die Ergebnisse des Pearson-Chi-Quadrat-Tests aller Annotationstypen über alle Projekte. Neben dem Signifikanzniveau sind auch die Anzahl der Projekte, in denen ein Annotationstyp zur Verwendung vorgeschlagen wurde, und die Anzahl der Projekte, in denen ein Annotationstyp verwendet wurde, abgebildet. Die Ergebnisse des Pearson-Chi-Quadrat-Tests bestätigen die Ergebnisse des vorherigen Bravais-Pearson-Tests für fast alle Annotations-typen. Ausnahmen sind die Annotationstypen Hohe Last ( $p = 0,106$ ) und Innovation ( $p = 0,752$ ). Der Zusammenhang des Vorschlags zur Verwendung und der Verwendung ist bei beiden Annotationstypen also zufällig, wobei die Menge der Vorschläge und Verwendungen des Annotationstypen Innovation in Bezug zur Aussagekraft des statistischen Tests gesetzt werden muss. Die Annotation Innovation wurde in zwei Projekten vorgeschlagen und in einem Projekt verwendet, wobei die drei Projekte disjunkt sind.

Tabelle 19: Korrelation zwischen Vorschlag und Verwendung je Annotationstyp

Annotationstyp	Anzahl Projekte – vorgeschlagen	Anzahl Projekte – verwendet	Sig. (2-seitig)
Ablösung	14	9	0,021
Automatisierung	4	5	0,004
Benutzbarkeit	7	7	0,000
Business Value	16	15	0,000
Externe Ressource	10	6	0,003
Flexibilität	12	13	0,002
Hohe Last	17	14	0,106
Innovation	2	1	0,752
Komplexität	19	18	0,030
Korrektheit	4	8	0,033
Manuelle Bearbeitung	13	13	0,002
Mobilität	8	7	0,000
Randbedingung	11	10	0,002
Sicherheit	10	8	0,013
Ungewissheit	19	17	0,005
Unveränderlichkeit	14	13	0,016
User Value	3	4	0,000
Verbesserungsbedarf	17	17	0,004
Zeitbeschränkung	10	14	0,021
Zuverlässigkeit	8	5	0,001

Aufgrund der Ergebnisse des Bravais-Pearson- und der Pearson-Chi-Quadrat-Tests lässt sich schlussfolgern, dass ein Zusammenhang zwischen zur Verwendung vorgeschlagenen und verwendeten Annotationen existiert.

### 6.3.3 Forschungsfrage 3: Besteht ein Zusammenhang zwischen der Menge verwendeter Annotationen und der Anzahl von Duplikaten?

Die dritte Forschungsfrage zielt darauf ab, anhand der Projektdaten zu überprüfen, ob ein Zusammenhang zwischen der Menge verwendeter Annotationen und der Anzahl der Duplikate existiert. Eine Annotation ist durch den Annotationstyp und ihre ID projektweit eindeutig. Sofern die Kombination aus Annotationstyp und ID mehrfach in einem Projekt verwendet wurde, handelt es sich um ein Duplikat. Die Menge aller Annotationen beinhaltet alle eindeutigen Annotationen der untersuchten Projekte, die Multimenge enthält zusätzlich die Duplikate.

Die Menge der Annotationen aller Projekte besteht aus 1629 Annotationen, wohingegen die Multimenge der Annotationen aus 2043 Annotationen besteht. Dies bedeutet, dass 25,41 % oder ungefähr jede fünfte Annotation ein Duplikat darstellt. Um zu überprüfen, ob ein Zusammenhang zwischen der Menge verwendeter Annotationen und der Multimenge verwendeter Annotationen eines Projektes existiert, wurde ein Bravais-Pearson-Test durchgeführt (vgl. Tabelle 20).

Tabelle 20: Korrelation zwischen der Menge verwendeter Annotationen und der Multimenge verwendeter Annotationen

		Menge der Annotationen	Multimenge der Annotationen
Menge der Annotationen	Pearson-Korrelation	1	,998
	Bestimmtheitsmaß	1	,996
	Sig. (2-seitig)		,000
	N	23	23
Multimenge der Annotationen	Pearson-Korrelation	,998	1
	Bestimmtheitsmaß	,996	1
	Sig. (2-seitig)	,000	
	N	23	23

Die Ergebnisse des Bravais-Pearson-Tests zeigen die statistische Signifikanz eines positiven Zusammenhangs zwischen der Menge verwendeter Annotationen und der Multimenge verwendeter Annotationen ( $r = 0,998$ ;  $p = 0$ ;  $r^2 = 0,996$ ). Dies bedeutet, dass jede fünfte Annotation ein Duplikat ist.

Weil in dem größten Projekt mit Abstand die meisten Annotationen verwendet wurden, wurde der Bravais-Pearson-Test für die übrigen 22 Projekte wiederholt. Dadurch wird ein überproportionaler Einfluss des größten Projektes, auf den sehr hohen Korrelationskoeffizienten, ausgeschlossen.

Tabelle 21: Korrelation zwischen der Menge verwendeter Annotationen und der Multimenge verwendeter Annotationen exklusive des größten Projekts

		Menge der Annotationen	Multimenge der Annotationen
Menge der Annotationen	Pearson-Korrelation	1	,973
	Bestimmtheitsmaß	1	,946
	Sig. (2-seitig)		,000
	N	22	22
Multimenge der Annotationen	Pearson-Korrelation	,973	1
	Bestimmtheitsmaß	,946	1
	Sig. (2-seitig)	,000	
	N	22	22

Die Ergebnisse des Bravais-Pearson-Tests für die übrigen 22 Projekte bestätigen statistische Signifikanz des vorherigen Tests ( $p = 0$ , vgl. Tabelle 21). Zwar sinkt der Korrelationskoeffizient auf  $r = 0,973$  und das Bestimmtheitsmaß auf  $r^2 = 0,946$ , jedoch ist der Unterschied so gering, dass die Interpretation dieselbe ist: jede fünfte Annotation ist ein Duplikat.

Ein Duplikat kann entweder auf derselben Landkarte und am selben Landkartenelement, auf derselben Landkarte jedoch auf einem anderen Landkartenelement oder auf unterschiedlichen Landkarten vorkommen. Der Erkenntnisgewinn durch Duplikate ist unterschiedlich. Eine Annotation am selben Landkartenelement stiftet keine Erkenntnis, denn diese entstand bereits in der Diskussion und Dokumentation der ursprünglichen Annotation. Die „Erkenntnis“, dass mehrere Stakeholder die gleiche Meinung besitzen, deutet eher darauf hin, dass der Aspekt ausreichend verstanden und nicht thematisiert werden muss. Annotationen, die zwar ein Duplikat sind, jedoch an anderen Landkartenelementen heften, liefern einen Erkenntnisgewinn, weil dieselbe Anforderung an einer anderen Stelle im Projekt zu berücksichtigen ist.

Von den insgesamt 414 Duplikaten wurden 189 Duplikate an dasselbe Landkartenelement geheftet, 194 Duplikate befanden sich auf derselben Landkarte, jedoch auf einem anderen Landkartenelement und 31 Duplikate befanden sich auf unterschiedlichen Landkarten. 45,65 % aller Duplikate brachten demnach keinen Erkenntnisgewinn. Bezogen auf die Menge und

Multimenge der Annotationen bedeutet dies, dass 9,25 % der verwendeten Annotationen keinen Erkenntnisgewinn lieferten (vgl. Tabelle 22).

Tabelle 22: Erkenntnislose Annotationen aller Projekte

Annotationen		Duplikate			erkenntnislose Duplikate	erkenntnislose Annotationen
Menge	Multimenge	selbes Landkartenelement	selbe Landkarte, unterschiedliches Element	unterschiedliche Landkarte		
1629	2043	189	194	31	45,65 %	9,25 %

Weil die Menge der Annotationen des größten Projekts (619 Annotationen) 4,8 mal so groß war, wie die Menge und Multimenge des nächstgrößten Projekts (129 Annotationen) und weil die Multimenge des größten Projekts (800 Annotationen) 5,23 mal so groß war, wie die Menge und Multimenge des nächstgrößten Projekts (153 Annotationen), wurden die vorherigen Berechnungen ohne das größte Projekt erneut durchgeführt.

Ohne das größte Projekt wurden 87 Duplikate an dasselbe Landkartenelement geheftet, 128 Duplikate wurden an ein anderes Landkartenelement derselben Landkarte geheftet und 18 Duplikate wurden auf andere Landkarten geheftet, sodass sich der Anteil der Duplikate, die keine Erkenntnis brachten, auf 37,34 % reduziert, was den Gesamtanteil der Duplikate ohne Erkenntnisgewinn über die Hälfte auf 7 % reduziert (vgl. Tabelle 23).

Tabelle 23: Erkenntnislose Annotationen exklusive des größten Projekts

Annotationen		Duplikate			erkenntnislose Duplikate	erkenntnislose Annotationen
Menge	Multimenge	selbes Landkartenelement	selbe Landkarte, unterschiedliches Element	unterschiedliche Landkarte		
1019	1243	87	128	18	37,34 %	7 %

2,25 % aller Duplikate ohne Erkenntnisgewinn lassen sich also auf das mit Abstand größte Projekt zurückführen.

Der Anteil erkenntnisloser Annotationen lässt keinen direkten Rückschluss auf die Höhe des Mehraufwandes zu. Zwar wird jede Annotation einzeln diskutiert, jedoch ist die Dauer bis zur Feststellung eines Duplikats unterschiedlich. Während manche Stakeholder direkt äußern, dass sie durch einen Annotationstypen dasselbe meinen, und sich der Aufwand dadurch auf ein Minimum reduziert, stellt sich für andere Annotationstypen erst im Verlauf einer Diskussion heraus, dass es sich um ein Duplikat handelt. IR-Coaches könnender doppelten Diskussion

entgegenwirken, indem sie für gleiche Annotationstypen am selben Element fragen, ob es sich um ein Duplikat handelt. Dieser Eingriff ist aus zwei Gesichtspunkten umsichtig zu verwenden: Auf der einen Seite können Stakeholder durch die Suggestivfrage beeinflusst werden, auf der anderen Seite gibt es Stakeholder, die ihre Meinung äußern wollen und bereits Gesagtes wiederholen. Letzteres könnte durch einen IR-Coach unterbunden werden, was jedoch negative Konsequenzen auf die Kooperationsbereitschaft des Stakeholders haben kann.

Es besteht also ein sehr starker Zusammenhang zwischen der Menge und der Multimenge der Annotationen: jede fünfte Annotation ist ein Duplikat. Bezogen auf Duplikate am selben Landkartenelement brachte ungefähr jede elfte Annotation keine Erkenntnis. Bereinigt um das größte Projekt reduziert sich der Wert, sodass knapp jede vierzehnte Annotation ohne Erkenntnis verwendet wurde.

## 6.4 Diskussion

Abschließend wird die Validität der Ergebnisse diskutiert und ein Fazit gezogen.

### 6.4.1 Konstruktvalidität

Für die Beantwortung der Forschungsfrage 1 nach dem Zusammenhang von Metriken der Projektgröße, wurden die Metriken direkt gemessen, weshalb die Validität des Konstrukts gegeben ist.

Für die Beantwortung der Forschungsfrage 2, „Gibt es einen Zusammenhang zwischen der zur Verwendung der vorgeschlagenen und den verwendeten Annotationstypen“, wird davon ausgegangen, dass dies eine Metrik für den Einfluss der IR-Coaches auf die Durchführung der Methode ist. Neben der Auswahl der Annotationen entscheiden die IR-Coaches auch über die zu verwendenden Landkarten, welche sich jedoch zur Überprüfung ihres Einflusses nicht eignen, denn die Landkarten werden kooperativ im Plenum mit den Stakeholdern erstellt. Dadurch können Stakeholder keine zusätzlichen Landkarten verwenden oder Landkarten nicht verwenden. Über die Verwendung einzelner Annotationstypen hingegen entscheiden die Stakeholder individuell, sodass hier der Effekt falscher und oder fehlender Vorschläge messbar ist.

Für die Beantwortung der dritten Forschungsfrage nach dem Zusammenhang zwischen Menge und Multimenge verwendeter Annotationen waren insbesondere die erkenntnislosen Annotationen von Interesse. Anhand doppelter Erläuterungen kann gemessen werden, wann

Erkenntnisgewinn durch Neues entsteht und wann nicht. Diese Metrik ist ein kleiner Teil der Beurteilung, ob die Methode effizient durchgeführt wurde. Andere Daten hätten nicht ohne Eingriff in die Durchführung der Methode erhoben werden können. Ein Eingriff ist beispielsweise das Führen eines Themenspeichers (vgl. Kapitel 5.2.2), in dem relevante, jedoch situativ unpassende Äußerungen notiert und dadurch unpassende Diskussionen unterbunden werden. Die Herausforderung dabei ist, die Kooperationsbereitschaft von Stakeholdern aufrechtzuerhalten, obwohl ihre Äußerungen nicht zielführend sind (vgl. [88]). Insbesondere die extensive Nutzung solcher Eingriffe kann die Ergebnisse des Interaction Rooms beeinflussen. Um diese Verzerrung zu vermeiden, wurden stattdessen Duplikate am selben Landkartenelement als Metrik für die Beurteilung der Höhe doppelter Diskussionen verwendet.

#### 6.4.2 Interne Validität

Die interne Validität muss im Hinblick auf unterschiedliche Gesichtspunkte für alle Forschungsfragen diskutiert werden:

Die Methode wird durch Interaction-Room-Coaches geleitet, wovon der Methodencoach (IR-MC) die Gruppe leitet und der Domänencoach (IR-DC) Diskussionen reflektiert, Impulse setzt und dokumentiert. Die Methodenkonfiguration übernehmen die IR-Coaches gemeinsam. In den untersuchten Projekten kamen insgesamt 18 unterschiedliche IR-Coaches zum Einsatz, wovon jedoch lediglich zwei die Rolle des IR-MC eingenommen haben. Dies kann einen Einfluss auf die Ergebnisse der ersten Forschungsfrage besitzen, denn der IR-MC erstellt die Landkarten mit den Stakeholdern und erläutert die Annotationen sowie deren Verwendung. Da sowohl die Verwendung als auch die Erläuterung der Annotationen durch eine Vorlage standardisiert sind, ist die Möglichkeit einer Verzerrung gering. Die Auswahl der Annotationen erfolgte durch beide Coaches und in Absprache mit den jeweiligen Projektverantwortlichen, sodass eine Verzerrung durch die zwei IR-MCs, sofern vorhanden, als gering angesehen werden kann. Gleiches gilt für die Duplikate, denn zwischen den Projekten existierte keine Schnittmenge der Stakeholder.

Die Erfahrung der Stakeholder mit dem Interaction Room unterscheidet sich je nach Projektdauer. Während keinem Stakeholder die Methode im Vorhinein bekannt war, besaßen Stakeholder, die in größeren Projekten an mehreren Workshops teilgenommen haben, in Folge-Workshops bereits Erfahrung mit der Methode, was zu einer Verzerrung bei der Verwendung der Annotationen geführt haben könnte. Die Verteilung der Workshops über mehrere

Workshop-Tage relativiert den Effekt, denn ab dem zweiten Tag besitzen alle Stakeholder eine Erfahrung mit der Methode. Darüber hinaus gab es keine Schnittmenge zwischen den Stakeholdern unterschiedlicher Projekte, sodass für die Untersuchung der Forschungsfragen nicht von einer Verzerrung ausgegangen wird.

#### 6.4.3 Externe Validität

Im Sinne der externen Validität, zur Verallgemeinerbarkeit der Ergebnisse, ist die mit 23 Projekten geringe Stichprobe zu berücksichtigen. Insbesondere die Ergebnisse der ersten Forschungsfragen weisen auf eine Varianz im Zusammenhang zwischen Metriken der Projektgröße hin, die vor diesem Hintergrund – trotz der statistischen Signifikanz und Stärke des Effekts – mit Vorsicht zu interpretieren sind. Der Zusammenhang zwischen den Variablen der Forschungsfragen 2 und 3 ist sehr hoch, sodass hier in Richtung einer Verallgemeinerung argumentiert wird.

Die Projekte wurden in unterschiedlichen Branchen durchgeführt, jedoch mehrere Projekte je Branche, sodass ein branchenspezifischer Zusammenhang zwischen den Projekten existieren könnte und die Projekte in einer Branche nicht als unabhängige Variablen aufgefasst werden könnten. Jedoch wurden in jeder Branche unterschiedliche Projekttypen durchgeführt – selbst wenn Projekttyp und Unternehmen dasselbe waren. So befinden sich beispielsweise zwei der vier größten Banken und zwei der zehn größten Versicherer Deutschlands unter den Projektpartnern. Deren Größe hat dazu beigetragen, dass es keine thematischen oder personellen Überschneidungen in den Projekten gab und die Projekte als unabhängige Variablen betrachtet werden.

#### 6.4.4 Fazit

Die Ergebnisse der statistischen Analyse zeigen, dass ein positiver Zusammenhang zwischen Metriken der Projektgröße existiert. Es besteht auch eine Korrelation zwischen zur Verwendung vorgeschlagenen und verwendeten Annotationen, was darauf hinweist, dass die IR-Coaches einen Einfluss auf die Durchführung der Methode besitzen. Der Zusammenhang konnte für 18 aus 20 Annotationstypen gezeigt werden. Die mangelnde Korrelation beim Annotationstyp Innovation lässt sich durch die geringe Anzahl ihrer Vorschläge und Verwendungen erklären. Die mangelnde statistische Signifikanz des Zusammenhangs zwischen Vorschlag und Verwendung des Annotationstypen Hohe Last lässt sich, ohne Vermutungen anzustellen, nicht

erklären. Es wurde festgestellt, dass bei der Durchführung der Methode jede vierzehnte Annotation ein Duplikat ohne Erkenntnis ist und die zu deren Identifikation notwendige Diskussion des Annotationstyps überflüssig war.



## 7 Fallstudie: Unterstützung der Aufwandsschätzung durch Interaction-Room-Annotationen

Unpräzise Aufwandsschätzung ist eine häufige Ursache für die unrealistische Planung von IT-Projekten (vgl. [24], Kapitel 1). Schätzungen gleichen Wünschen, anstatt realistische Aussagen über den erwarteten Aufwand zu treffen [12], [39]. Um herauszufinden, ob die Interaction-Room-Annotationen einen Einfluss auf die Aufwandsschätzung besitzen und zu einem gemeinsamen Verständnis unter den Schätzenden beitragen, wurde eine Studie im Rahmen einer Lehrveranstaltung im Fach Informatik an der Hochschule Rhein-Waal durchgeführt. Deren Ergebnisse, veröffentlicht von Grapenthin, Book, Richter und Gruhn [153], werden hier wiedergegeben. Im Folgenden werden kurz unterschiedliche Arten der Schätzung vorgestellt, um daraufhin Expertenschätzungen im Detail zu diskutieren und eine Abgrenzung zu verwandten Arbeiten vorzunehmen.

Grundsätzlich kann zwischen zwei Arten von Techniken zur Aufwandsschätzung unterschieden werden: Auf der einen Seite werden in modellbasierten Schätztechniken historische Daten zur Kalibrierung von Parametern verwendet, um Aufwände berechnen zu können. Beispiele für solche Techniken sind COCOMO II [253] und SLIM [254]. Auf der anderen Seite wird in Expertenschätzungen das Wissen und die Erfahrung von Stakeholdern für die Aufwandsschätzung verwendet [255]. Beispiele für solche Techniken sind Wideband Delphi [73] und Planning Poker [256]. Auch wenn Schätzungen durch Experten ausschließlich auf deren Expertise beruhen, existieren keine Hinweise darauf, dass modellbasierte Schätztechniken genauer sind [257]. Allerdings ist die Anzahl wissenschaftlicher Auseinandersetzungen mit Expertenschätzungen deutlich geringer als mit modellbasierten Schätztechniken [258], wohingegen Expertenschätzungen in der Industrie weiter verbreitet sind [259].

Expertenschätzungen sind insbesondere dann nützlich, wenn keine historischen Daten zur Kalibrierung von Modellen existieren [260] oder verfügbare Ressourcen für die Schätzung limitiert sind [261]. In der Literatur werden unterschiedliche Parameter zur Gestaltung von Expertenschätzungen genannt: In strukturiertem oder unstrukturiertem Vorgehen werden Einzelschätzungen oder Gruppenschätzungen durchgeführt; die Schätzenden sind entweder Entwickler oder keine Entwickler [257]. Entwickler, die geschätzte Anforderungen später umsetzen, schätzen genauer [262], da sie meist mehr mit dem Projekt vertraut sind und eine höhere Motivation besitzen als Schätzende, die Anforderungen im Anschluss nicht implementieren [263].

Die Genauigkeit der statistischen Kombination von Einzelschätzungen ist größer als die Genauigkeit von Einzelschätzungen [264]. Gruppenschätzungen nutzen einen strukturierten oder unstrukturierten Erfahrungsaustausch über Schätzartefakte [257], wobei unstrukturierte Schätzungen ungenauer sind, was durch die mangelnde Beteiligung Einzelner begründet wird [265].

Eine populäre Schätztechnik für iterative Softwareprozesse ist Planning Poker. Planning Poker ist eine strukturierte Gruppenschätztechnik, die auf die Delphi-Technik zurückzuführen ist [266]. Die Delphi-Technik verwendet die statistische Kombination von Einzelschätzungen, um eine Gruppenschätzung zu erhalten [267]. 1981 hat Barry Boehm mit Wideband Delphi [268] eine abgewandelte Form entwickelt, in der Experten Argumente austauschen, die zu ihren unterschiedlichen Einzelschätzungen führten. Planning Poker adaptiert beide Verfahren: Extrema in den Einzelschätzungen werden diskutiert und es wird versucht, über einen Konsens zur Gruppenschätzung zu gelangen.

Wegen seiner Verbreitung wurden viele Studien zu Planning Poker durchgeführt. Haugen [269] hat die Schätzgenauigkeit von Planning Poker gegenüber unstrukturierten Schätztechniken untersucht. Dabei hat er herausgefunden, dass Planning Poker zu genaueren Schätzungen führt, wenn die Schätzenden Erfahrung mit ähnlichen Aufgaben besitzen, jedoch wahrscheinlich weniger genau ist, wenn diese Erfahrung fehlt. Darüber hinaus kann Planning Poker zu extrem hohen bzw. niedrigen Schätzungen führen, die auf Gruppenpolarisierung [266] zurückgeführt werden kann.

Moløkken-Østvold, Haugen und Benestad [270] haben die Schätzgenauigkeit von Planning Poker mit der Schätzgenauigkeit von Einzelschätzungen einer Kontrollgruppe verglichen und herausgefunden, dass die Planning-Poker-Schätzungen geringfügig genauer waren. Die Autoren haben zusätzlich die Richtung der abweichenden Schätzungen untersucht und festgestellt, dass Schätzungen mit Planning Poker geringfügig weniger optimistisch waren als die statistische Kombination der Einzelschätzungen.

Mahnič und Hovelja [266] haben im Rahmen einer Lehrveranstaltung die Ergebnisse der statistischen Kombination von Einzelschätzungen durch Studierende und dem Ergebnis ihrer Planning-Poker-Gruppenschätzung verglichen. So haben sie überprüft, ob die Planning-Poker-Gruppenschätzungen weniger optimistisch und genauer sind als die statistische Kombination der Einzelschätzungen. Die Autoren haben herausgefunden, dass die statistische Kombination der Einzelschätzungen von Studierenden weniger optimistisch und genauer war als die Planning-Poker-Gruppenschätzungen. Dieses Ergebnis steht in Widerspruch zu den Ergebnissen

von Moløkken-Østvold et al. [270], auch wenn der Effekt in beiden Studien gering ausfiel. Des Weiteren schätzte eine Kontrollgruppe erfahrenerer Entwickler dieselben Artefakte, um feststellen zu können, ob ein Unterschied zwischen den Schätzungen der Studierenden und den Schätzungen erfahrenerer Entwickler existiert. Die statistische Kombination der Einzelschätzungen erfahrenerer Entwickler war pessimistischer und ungenauer als deren Planning-Poker-Gruppenschätzungen, jedoch war der Effekt gering und nicht statistisch signifikant. Der Vergleich der Einzelschätzungen von Studierenden mit den Einzelschätzungen der erfahreneren Entwickler zeigte, dass die Studierenden statistisch signifikant ungenauer schätzen.

Während Haugen [269] und Moløkken-Østvold et al. [270] Planning Poker mit anderen Schätztechniken verglichen haben, haben Mahnič und Hovelja [266] die Schätzgenauigkeit und die Richtung der Abweichung zwischen weniger erfahrenen Entwicklern und erfahreneren Entwicklern verglichen. Die Überprüfung des Einflusses der Interaction-Room-Annotationen ist an die Studie von Mahnič und Hovelja angelehnt, weil die Probanden auch weniger erfahrende Entwickler sind und die Studie im Rahmen einer Lehrveranstaltung durchgeführt wurde. Jedoch werden die Schätzungen nicht mit den Schätzungen von erfahrenen Entwicklern verglichen, sondern es wird untersucht, ob das explizite Diskutieren von Risiko- und Aufwandstreibern anhand von Interaction-Room-Annotationen einen Effekt auf die Schätzung besitzt.

## 7.1 Forschungsfragen

Die Interaction-Room-Annotationen (vgl. Kapitel 5.3) haben sich in zahlreichen Industrieprojekten als nützlich herausgestellt (vgl. z. B. [152] und Kapitel 6.1), weswegen angenommen wird, dass die explizite Kennzeichnung und anschließende Diskussion über Aufwands- und Risikotreiber zu einem gemeinsamen Verständnis unter Schätzenden beiträgt und zu genaueren Schätzungen führt.

Um die Annahme zu testen, wurden die drei folgenden Forschungsfragen anhand der Daten aus dieser Studie beantwortet. Die erste Forschungsfrage untersucht, ob die Verwendung von Annotationen das gemeinsame Verständnis fördert:

**Forschungsfrage 1:** Wird das gemeinsame Verständnis unter Stakeholdern über Faktoren, die für das Schätzen relevant sind, durch Annotationen unterstützt?

Angelehnt an die Arbeit von Mahnič und Hovelja [266] ist der Einfluss der Annotationen auf die Schätzgenauigkeit von Interesse, weswegen die zweite Forschungsfrage formuliert wurde:

**Forschungsfrage 2:** Existiert ein Unterschied in der Schätzgenauigkeit zwischen Schätzungen, die mit Annotationen und Planning Poker durchgeführt werden, im Gegensatz zu Schätzungen, die ohne Annotationen und mit Planning Poker durchgeführt werden?

Buehler, Messervey und Griffin [271] haben gezeigt, dass sowohl Gruppenschätzungen als auch Einzelschätzungen zu Optimismus neigen, dieser jedoch bei Gruppenschätzungen größer ist. Diese Ergebnisse stehen in Widerspruch zu den Ergebnissen von Moløkken-Østvold et al. [270], die zwar auch optimistische Schätzungen in beiden Verfahren identifiziert haben, der Optimismus fiel jedoch bei Gruppenschätzungen geringer aus. Um herauszufinden, welchen Einfluss die Annotationen auf eine Abweichung in der Schätzung besitzen, wurde die dritte Forschungsfrage formuliert:

**Forschungsfrage 3:** Existiert ein Unterschied in der Abweichung der Schätzgenauigkeit, wenn Schätzungen mit Annotationen und Planning Poker durchgeführt werden, im Gegensatz zu Schätzungen, die ohne Annotationen und mit Planning Poker durchgeführt werden?

## 7.2 Methodik zur Überprüfung der Forschungsfragen

Zur Beantwortung der Forschungsfragen wurde im Rahmen einer Lehrveranstaltung „Web Engineering“ acht parallel durchgeführte, jedoch inhaltlich verschiedene Projekte untersucht. Die Projekte wurden von Partnern aus der Industrie und der öffentlichen Verwaltung gestellt, sodass es sich nicht um synthetische Projekte handelte, sondern beispielsweise um die Implementierung eines Systems, mit dem Flüchtlinge Übersetzer finden können, einer web-basierten mobilen Anwendung für den Tourismus in Kamp-Lintfort oder eines Web-Clients für den augmentierten Interaction Room [272]. Die beteiligten 42 Studierenden befanden sich im fünften Semester ihres Bachelor-Studiums im Fach Informatik an der Hochschule Rhein-Waal.

Um die unterschiedliche Komplexität der Projekte zu berücksichtigen, wurden die Studierenden anhand ihrer Fähigkeiten, Erfahrung und Vorlieben in Teams unterteilt und den Projekten zugewiesen. Dazu wurde wie folgt vorgegangen:

1. Schritt: Fähigkeiten und Erfahrungen der Studierenden erfragen
  - a. Objektive Kriterien: Lebenslauf, Lehrveranstaltungen, Noten

- b. Subjektive Kriterien: Einschätzung der Erfahrungen mit Technologien, Systemen sowie Berücksichtigung von Wünschen
2. Schritt: Studierende analysieren
  - a. Subjektive Kriterien gewichten
  - b. Studierende bewerten und kategorisieren
3. Schritt: Studierende in Teams einteilen
  - a. Wünsche für Teammitglieder und Projekte analysieren
  - b. Studierende entsprechend Kategorie und Wünschen in Teams einteilen

Zu Beginn der Lehrveranstaltung wurde ein Fragebogen (vgl. Schritt 1) verteilt, in dem die Studierenden nach ihrem Werdegang bis zu Beginn des Studiums gefragt wurden, insbesondere die Erfahrung in der Programmierung abseits universitärer Lehrveranstaltungen, die Note des Abschlusses für den Hochschulzugang und eine eventuell bereits vorhandene Berufsausbildung wurden erfragt (Schritt 1a). Weil die Studierenden kurz vor dem Abschluss ihres Bachelors standen, wurde auch nach den bereits besuchten und erfolgreich abgeschlossenen Lehrveranstaltungen gefragt (Schritt 1a). Im Anschluss wurden die Studierenden um die Einschätzung ihrer Erfahrung mit relevanten Technologien (z. B. JavaScript, Ruby, Python), Konzepten (z. B. NoSQL, REST, Responsive Design), Systemen (z. B. Joomla!, Wordpress, Subversion SVN), Frameworks (z. B. Leaflet, AngularJS, Bootstrap und Vorgehensmodellen für Softwareentwicklungsprozesse (z. B. Scrum, Crystal, Feature-Driven-Development) gefragt (Schritt 1b). Darüber hinaus wurden die Studierenden nach Wünschen für Teammitglieder und Projekte gefragt (Schritt 1b). Um möglichst leistungshomogene Teams bilden und eine Zuordnung von Fähigkeiten zur Projektanforderungen vornehmen zu können, wurden die Fragen gewichtet (Schritt 2a). Die Summe der Produkte aus persönlicher Einschätzung und dem entsprechenden Gewicht wurde als Fähigkeitswert je Studierendem verwendet (Schritt 2b). Auf Basis des Fähigkeitswerts wurden leistungshomogene Teams erstellt (Schritt 3b). In die Zusammenstellung der Teams sind persönliche Präferenzen der Teammitglieder eingeflossen und Projektwünsche wurden bei deren Zuteilung berücksichtigt (Schritt 3a). Auf diese Weise wurden die 42 Studierenden in acht Teams mit drei bis sieben Teammitgliedern eingeteilt.

Die Entwicklung der Projekte wurde nach Scrum, mit einer vorgelagerten Phase zur Ermittlung der Kundenanforderungen, durchgeführt. Der Product Owner und der Scrum Master wurden durch das Team aus den eigenen Reihen bestimmt. Insgesamt wurden neun Entwicklungsiterationen (Sprints) mit einer Dauer von jeweils einer Woche durchgeführt.

### 7.2.1 Studienaufbau

Zu Beginn des Semesters ermittelten die Studierenden die Kundenanforderungen der Praxispartner. Diese wurden in einem Anforderungsdokument dokumentiert, das von den Praxispartnern abgenommen wurde. Auf Basis dieses expliziten gemeinsamen Verständnisses haben die Teams User Stories geschrieben, um eine initiale Schätzung vorzubereiten. Eine User Story ist eine strukturierte Anforderungsbeschreibung aus der Perspektive eines bestimmten Benutzers des zukünftigen Systems [273], die als eine Referenz auf die detaillierte Anforderung diene.

Entsprechend Cohn [106] wurden die User Stories mit Hilfe von Planning Poker durch die Teams geschätzt. Jedes Teammitglied erhielt ein Planning-Poker-Kartendeck, das üblicherweise die Zahlen 0, 1, 2, 3, 4, 5, 8, 13, 20, 40, und 100 beinhaltet. Die Zahlen stehen für Story Points: Ein fiktives Maß für das Bestimmen der Größe einer User Story [106], das beispielsweise in Stunden oder Arbeitstage umgerechnet werden kann. Weil die Teams nicht in Vollzeit an dem Projekt arbeiteten und um die Projektumfänge zu berücksichtigen, bekamen die Teams nur Karten mit den Story Points 0,5, 1, 2, 3 und 5 zur Verfügung gestellt, wobei ein Story Point einen idealen Arbeitstag (sechs Stunden) darstellte.

Entsprechend der Regeln von Planning Poker, haben die Teams nach dem folgenden Vorgehen geschätzt:

1. Der Product Owner verliest die User Story.
2. Das Team diskutiert Fragen zu der User Story.
3. Die Teammitglieder geben ihre Einzelschätzungen ab, indem sie eine Karte ihres Decks auswählen und verdeckt vor sich legen.
4. Alle ausgewählten Karten der Teammitglieder werden gleichzeitig umgedreht.
5. Die Teammitglieder, deren Schätzung am weitesten auseinander liegen, tauschen Argumente für die Rechtfertigung der eigenen Schätzung aus.
6. Sofern die Teammitglieder sich auf einen Schätzwert einigen können, ist die Schätzung der User Story beendet, andernfalls geht es weiter mit Schritt 3.

Für die Durchführung des zweiten Schritts wurden alle Teams aufgefordert, sowohl die fachlichen und technischen Anforderungen einer User Story als auch Aufwands- und Risikotreiber zu berücksichtigen. Dazu wurden einer Hälfte der Teams das Konzept der Interaction-Room-Annotationen erläutert und sieben Annotationstypen (Hohe Last, Flexibilität, Zuverlässigkeit,

Zeitkritikalität, Benutzbarkeit, Komplexität und Ungewissheit) als Klebezettel zur Verfügung gestellt. Die Teammitglieder wurden aufgefordert, User Stories zu annotieren und im Anschluss zu diskutieren. Der anderen Hälfte der Teams wurden die äquivalenten Aufwands- und Risikotreiber erläutert, jedoch nicht das Konzept der Annotationen. Sie wurden ebenfalls aufgefordert, die Aufwands- und Risikotreiber zu diskutieren.

Bei der Auswahl der Annotationstypen wurde bewusst auf Annotationstypen der Kategorie Werttreiber verzichtet, weil Aufwands- und Risikoannotationen Einfluss auf die Schätzung besitzen, wohingegen Werttreiber eher Einfluss auf die Priorität haben und von Kunden verwendet werden.

Jedes Team erhielt eine Tabelle, in der der Scrum Master die ID jeder geschätzten User Story, die Nummer der SchätZRunde, die Einzelschätzungen der Teammitglieder und die Gruppenschätzung protokollierte. Während der Projektlaufzeit wurden zusätzlich die tatsächlichen Aufwände jeder User Story in der Tabelle protokolliert. Tabelle 24 zeigt einen Ausschnitt einer solchen Tabelle, wobei der tatsächliche Aufwand in Story Points umgerechnet wurde.

Tabelle 24: Aufzeichnung der Schätzungen und tatsächlichen Aufwände (Auszug)

User Story	SchätZRunde	Einzelschätzung 1	Einzelschätzung 2	Einzelschätzung 3	Einzelschätzung 4	Gruppenschätzung	tatsächlicher Aufwand
1	1	3	7	5	5	-	
1	2	3	5	3	5	4	6
2	1	2	3	5	3	4	3,85
3	1	0,5	5	2	1	4	2,75

### 7.2.2 Berechnungen zur Beantwortung der Forschungsfragen

Um zu beantworten, ob die Annotationen zu einem gemeinsamen Verständnis beitragen, wurde die Streuung der Einzelschätzungen von Teams mit Annotationen mit der Streuung der Einzelschätzungen von Teams ohne Annotationen verglichen. Unter der Annahme, dass Schätzende mit einem gemeinsamen Verständnis gleich schätzen, wurde die Varianz der Einzelschätzungen als Metrik ausgewählt. Falls ein Team mehrere Runden für die Einigung auf eine Gruppenschätzung benötigte, wurde die Varianzanalyse auf Grundlage der Einzelschätzungen der ersten

SchätZRunde durchgeführt. In diesen Daten existiert keine Verzerrung durch das Wissen aus der Diskussion über die Extrema der Einzelschätzungen.

Für die Berechnung der Streuung wird der Variationskoeffizient  $v$ , als Ergebnis der Division der Standardabweichung  $s$  einer Stichprobe durch ihren Mittelwert  $\bar{x}$ , verwendet (vgl. Formel 9).

Formel 9: Berechnung des Variationskoeffizienten

$$v = \frac{s}{\bar{x}}$$

Je geringer die Streuung der Einzelschätzungen, desto einheitlicher haben die Mitglieder der Teams einzeln geschätzt.

Um die Forschungsfrage 2 zu beantworten, wurde die balancierte relative Abweichung (BRE) der Gruppenschätzungen und des tatsächlichen Aufwands verwendet. Entsprechend [264], wurde der BRE anstatt der relativen Abweichung (MRE) verwendet, da der BRE sowohl Über- als auch Unterschätzen berücksichtigt [274] (vgl. Formel 10).

Formel 10: Balancierte relative Abweichung

$$BRE = \frac{|tats\ddot{a}chlicher\ Aufwand - gesch\ddot{a}tzter\ Aufwand|}{\min(tats\ddot{a}chlicher\ Aufwand, gesch\ddot{a}tzter\ Aufwand)}$$

Um zu beantworten, ob eine optimistische oder pessimistische Abweichung in den Gruppenschätzungen existierte, wurde der BREbias verwendet, der nicht nur die Größe der Abweichung, sondern auch dessen Richtung berücksichtigt (vgl. Formel 11).

Formel 11: Richtung und Größe der balancierten relativen Abweichung

$$BREbias = \frac{(tats\ddot{a}chlicher\ Aufwand - gesch\ddot{a}tzter\ Aufwand)}{\min(tats\ddot{a}chlicher\ Aufwand, gesch\ddot{a}tzter\ Aufwand)}$$

### 7.3 Ergebnisse

Von den ursprünglich 203 geschätzten User Stories wurden nur komplett implementierte User Stories zur Beantwortung der Forschungsfragen herangezogen, sodass insgesamt 153 User Stories verwendet wurden. Von den 153 User Stories wurden 71 mit Planning Poker und



Annotationen geschätzt und 82 mit Planning Poker und ohne Annotationen. Tabelle 25 zeigt die deskriptive Statistik für die Berechnungen zur Beantwortung der Forschungsfragen. Die Verteilung der Daten von User Stories, die mit Planning Poker und Annotationen geschätzt wurden, und von User Stories, die mit Planning Poker und ohne Annotationen geschätzt wurden, wurde zunächst auf Normalität geprüft.

Um festzustellen, ob die Daten normalverteilt sind, wurde ein Shapiro-Wilk-Test [275] mit einem Konfidenzniveau von  $p > 0,05$  eingesetzt. Zusätzlich wurden die Histogramme, die Quantile-Quantile-Plots und Box-Plots analysiert, sowie die Schiefe und die Krümmung berücksichtigt [276].

Tabelle 25: Deskriptive Statistik für komplett implementierte User Stories

	Variationskoeffizient mit Annotationen	Variationskoeffizient ohne Annotationen	BRE mit Annotationen	BRE ohne Annotationen	BREbias mit Annotationen	BREbias ohne Annotationen
N	71	82	71	82	71	82
Mittelwert	,308	,328	2,980	4,921	-1,1682	-3,958
Median	,335	,372	1,000	1,792	-,250	-2,128
Standardabweichung	,238	,225	7,395	13,641	7,8936	,579
Schiefe	,513	-,026	6,472	7,466	,305	-7,257
Krümmung	,229	-,799	48,255	62,023	-42,044	-58,427

Die Schiefe und Krümmung der Variationskoeffizienten liegt leicht außerhalb des Intervalls von  $\pm 1,96$  (vgl. Tabelle 25), jedoch ergab eine visuelle Inspektion des Histogramms, dass sie nicht normalverteilt sind (vgl. Abbildung 44). Sowohl die Krümmung des BRE als auch des BREbias haben das Intervall von  $\pm 1,96$  deutlich unterschritten, sodass BRE und BREbias nicht als normalverteilt angesehen werden können.

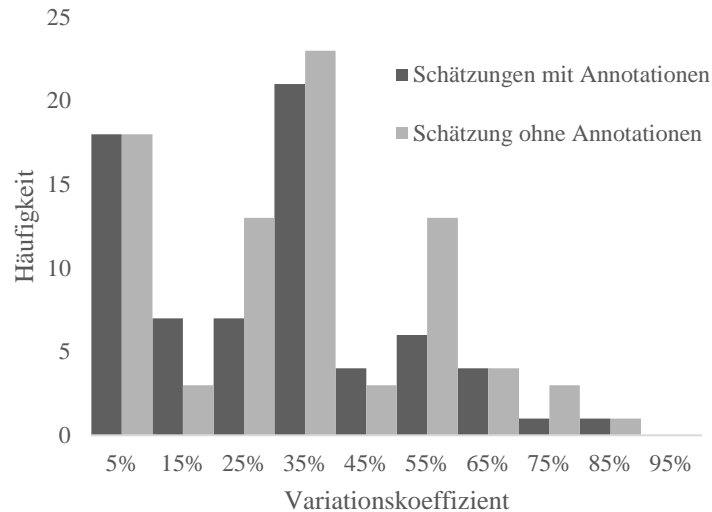


Abbildung 44: Histogramm des Variationskoeffizienten der Schätzungen

Weil die User Stories der Projekte unabhängige Variablen sind, wurde ein nichtparametrischer Mann-Whitney-Wilcoxon-Test [277] durchgeführt zur Beantwortung der drei Forschungsfragen. Die Stärke des Effekts wurde mit Hilfe von Cliff's Delta [278] bestimmt. Cliff's Delta eignet sich für die Bestimmung der Effektstärke nicht normalverteilter und mindestens ordinalskalierten Daten [266].

### 7.3.1 Forschungsfrage 1: Wird das gemeinsame Verständnis unter Stakeholdern über Faktoren, die für das Schätzen relevant sind, durch Annotationen unterstützt?

Auf den ersten Blick ist das gemeinsame Verständnis der Schätzenden, die Annotationen verwendet haben, größer als das gemeinsame Verständnis der Schätzenden ohne Annotationen. Sowohl die durchschnittliche Streuung (30,8 % ggü. 32,8 %) als auch der Median (33,5 % ggü. 37,2 %) sind geringer (vgl. Tabelle 25), sodass die Annotationen in dieser Stichprobe zum gemeinsamen Verständnis aller Teammitglieder beigetragen haben.

Der einseitige Mann-Whitney-Wilcoxon-Test (vgl. Tabelle 26) zeigt keine Signifikanz der Ergebnisse ( $p = 0,187$ ) und eine kleine Effektstärke ( $-0,072$ ). Der Beitrag der Annotationen zu gemeinsamem Verständnis liegt in diesem Fall zwar vor, ist aber nicht verallgemeinerbar.

Tabelle 26: Beitrag der Annotationen zum gemeinsamen Verständnis

Variationskoeffizient								
	N	durchschnittlicher Rang	Summe der Ränge	Mann-Whitney-U	Wilcoxon W	Z	Sig. (zweiseitig)	Cliff's Delta
mit Annotationen	71	73,585	5224,5	2668,5	5224,5	-0,892	0,187	-0,072
ohne Annotationen	82	79,957	6556,5					
Summe	153							

### 7.3.2 Forschungsfrage 2: Existiert ein Unterschied in der Schätzgenauigkeit zwischen Schätzungen, die mit Annotationen und Planning Poker durchgeführt werden, im Gegensatz zu Schätzungen, die ohne Annotationen und mit Planning Poker durchgeführt werden?

Die balancierte relative Abweichung zwischen Gruppenschätzungen und tatsächlichen Aufwänden besitzt eine große Spannweite, was dazu führt, dass die durchschnittlichen BREs stark durch Ausreißer beeinflusst sind (vgl. Tabelle 25). Die Mediane in Tabelle 25 besitzen deswegen größere Aussagekraft: Die Mitte der Schätzungen mit Annotationen war sehr genau (Median = 1,0; punktgenau geschätzt). Die Genauigkeit der Schätzungen mit Planning Poker und ohne Annotationen war im Median geringer: der relative Schätzfehler betrug fast 80 %.

Der zweiseitige Mann-Whitney-Wilcoxon-Test (vgl. Tabelle 27) zeigt eine Signifikanz der Ergebnisse ( $p = 0,033$ ) und eine kleine Effektstärke (-0,173). Die Annotationen haben also einen positiven Einfluss auf die Schätzgenauigkeit.

Tabelle 27: Einfluss der Annotationen auf die Schätzgenauigkeit

BRE								
	N	durchschnittlicher Rang	Summe der Ränge	Mann-Whitney-U	Wilcoxon W	Z	Sig. (zweiseitig)	Cliff's Delta
mit Annotationen	71	68,789	4884,0	2328,0	4884,0	-2,135	0,033	-0,173
ohne Annotationen	82	84,110	6897,0					

---

Summe	153
-------	-----

---

### 7.3.3 Forschungsfrage 3: Existiert ein Unterschied in der Abweichung der Schätzgenauigkeit, wenn Schätzungen mit Annotationen und Planning Poker durchgeführt werden, im Gegensatz zu Schätzungen, die ohne Annotationen und mit Planning Poker durchgeführt werden?

Während der BRE nur die Größe der Abweichung berücksichtigt, berücksichtigt der BREbias auch die Richtung der Abweichung, also eine Über- bzw. Unterschätzung.

Im Median haben sich die Schätzenden um ca. 25 % unterschätzt<sup>8</sup>, sofern sie Planning Poker und Annotationen verwendet haben (vgl. Tabelle 25). Die Schätzenden haben sich im Median um mehr als 200 % unterschätzt, sofern sie Planning Poker und keine Annotationen verwendet haben. Sowohl mit Annotationen als auch ohne Annotationen wurde in dieser Stichprobe pessimistisch geschätzt, auch wenn der Pessimismus unter Verwendung der Annotationen geringer ausfällt.

Der zweiseitige Mann-Whitney-Wilcoxon-Test (vgl. Tabelle 28) zeigt eine Signifikanz der Ergebnisse ( $p = 0,029$ ) und eine geringe Effektstärke (-0,176). Die Annotationen verringern also den Pessimismus bei den Schätzungen, ohne zu Optimismus zu tendieren.

Tabelle 28: Einfluss der Annotationen auf die Abweichung in Schätzungen

BREbias	N	durchschnittlicher Rang	Summe der Ränge	Mann-Whitney-U	Wilcoxon W	N	Sig. (zweiseitig)	Cliff's Delta
mit Annotationen	71	85,394	6063,0	2315,0	5718,0	-2,182	0,029	-0,176
ohne Annotationen	82	69,732	5718,0					
Summe	153							

---

<sup>8</sup> In Grapenthin et al. [153] wird von einer Überschätzung gesprochen, was nicht korrekt ist, denn eine optimistische Schätzung ist eine Überschätzung, eine pessimistische Schätzungen ist eine Unterschätzung. Ein negatives Vorzeichen des BREbias ist eine pessimistische bzw. eine Unterschätzung, weil weniger Aufwand benötigt wurde, als zunächst geschätzt wurde.

## 7.4 Diskussion

Abschließend wird die Validität der Ergebnisse diskutiert und ein Fazit gezogen.

### 7.4.1 Konstruktvalidität

Forschungsfrage 1 beantwortet die Frage, ob die Interaction-Room-Annotationen zu einem gemeinsamen Verständnis beitragen anhand der Streuung in den Einzelschätzungen mit und ohne Annotationen. Der Grad des gemeinsamen Verständnisses ist schwer durch einen Test zu bestimmen, was die Konstruktvalidität zu einem kritischen Erfolgsfaktor für den Studienaufbau macht. Der Variationskoeffizient der Einzelschätzungen von Teammitgliedern wurde verwendet, weil Stakeholder ihr gesamtes Wissen über ein Artefakt verwenden müssen, um es zu schätzen. Selbstverständlich benötigt ein Stakeholder mehr als nur Verständnis über Aufwands- und Risikotreiber eines Artefakts, um es zu schätzen, und die Schätzung ist von mehr Aspekten beeinflusst, als nur durch das Verständnis eines Stakeholders über diese Treiber. Trotzdem kann davon ausgegangen werden, dass Artefakte ähnlich geschätzt werden, wenn die Stakeholder ein ähnliches Verständnis des Artefakts besitzen. Deswegen ist die Varianz in Einzelschätzungen eine angemessene Metrik für das gemeinsame Verständnis, auch weil Annotationen auf die Kennzeichnung von Projektaufwänden abzielen und mit der Absicht, die Genauigkeit der Schätzung zu erhöhen, verwendet werden.

Die Forschungsfragen 2 und 3 können direkt gemessen werden, weswegen die Konstruktvalidität gegeben ist.

### 7.4.2 Interne Validität

Zur Berücksichtigung der internen Validität, müssen für alle drei Forschungsfragen mehrere Faktoren diskutiert werden.

Einer Schätzung liegt nicht nur die Komplexität einer Anforderung zu Grunde, sondern die Schätzung spiegelt immer auch die Erfahrung des Schätzenden wider. Um zu verhindern, dass die Ergebnisse durch große Unterschiede in den Fähigkeiten und Erfahrungen der Teammitglieder verzerrt werden, wurden leistungshomogene Teams gebildet (vgl. Kapitel 7.2.1). Daher wird kein großer Unterschied in den Fähigkeiten und der Erfahrung von Teammitgliedern der Teams angenommen, die die Ergebnisse beeinflussen könnten.

Als Resultat der leistungshomogenen Teams zeigte sich, dass einige Teams erfahrener als andere waren. Um dieses Ungleichgewicht zu normalisieren, wurden die Teams in den beiden Gruppen (mit/ohne Annotationen) anhand ihrer Erfahrung gleichverteilt.

Mehrfache Planning-Poker-SchätZRunden und viele Sprints können dazu führen, dass ein Routine-Effekt eintritt, aus dem ähnliche Schätzungen resultieren. Die Teams haben jeweils zwischen 15 und 20 User Stories geschätzt und implementiert. Daher ist dieser Effekt fast auszuschließen, außerdem tritt er eher in großen Industrieprojekten auf. Sofern der Effekt auftrat, war er für beide Gruppen gleich, daher wurden die Ergebnisse nicht zu Gunsten der einen oder anderen Gruppe beeinflusst.

Die Anforderungen eines Projekts wurden als unabhängige Variablen behandelt. Eine geringfügige Abhängigkeit könnte jedoch gegeben sein, denn die Anforderungen wurden durch ein Team geschätzt und umgesetzt und haben sich unter Umständen strukturell geglichen oder besaßen Abhängigkeiten zueinander. In der Industrie tritt derselbe Effekt auf und es wäre unnatürlich, wenn dieser kontrolliert worden wäre. Darüber hinaus wurden viele kleine, unabhängige Projekte durchgeführt, was dazu beigetragen hat, einen eventuellen derartigen Effekt gering zu halten. Zusätzlich wurde eine Abhängigkeitsanalyse durchgeführt, die für sieben von acht Teams keine statistisch signifikante Abhängigkeit in BRE und BREbias zeigte, weshalb weiterhin von unabhängigen Variablen ausgegangen wurde.

Die Qualität der Rohdaten besitzt ebenfalls Einfluss auf die interne Validität: Die Teams zeichneten die Daten selbst auf. Die Plausibilität der Daten wurde vor der Verwendung geprüft, Extremwerte wurden mit den Studierenden diskutiert und validiert. Zwei zusätzliche Teams haben an der Lehrveranstaltung teilgenommen, jedoch ihre tatsächlichen Aufwände nicht konsistent und plausibel aufgezeichnet, was dazu geführt hat, dass die Daten dieser Teams nicht in die Studie eingeflossen sind.

Ein Indikator für die Zuverlässigkeit der Ergebnisse der Forschungsfragen 2 und 3 ist die ähnliche Form der Histogramme in Abbildung 44: Der beobachtete Effekt scheint nicht durch ein verzerrtes Schätzverhalten der Teams hervorgerufen worden sein, stattdessen hat die Verwendung der Annotationen die Schätzungen in Richtung größerer Genauigkeit verschoben.

### 7.4.3 Externe Validität

Die Probanden waren Studierende mit weniger Erfahrung als hauptberufliche Entwickler. Die Studierenden standen kurz vor ihrem Bachelor-Abschluss an einer technischen Hochschule und

besaßen daher ein ähnliches Erfahrungslevel wie Berufseinsteiger (Young Professionals). Die Studierenden haben zum Teil die Aufwände einzelner Anforderungen stark über- bzw. unterschätzt, daher wurde der Median statt des Durchschnitts für die Diskussion der Forschungsfragen 2 und 3 verwendet. Die Unsicherheit weniger erfahrener Schätzer könnte eine Erklärung für die fehlende Signifikanz der Forschungsfrage 1 sein. Gleichzeitig kann dieser Umstand bedeuten, dass der Effekt auf die Genauigkeit und die Richtung der Abweichung für weniger Erfahrene größer ist als für Erfahrene.

Weil alle Teams unterschiedliche Projekte durchgeführt haben, besteht kein Anlass zu der Annahme, dass die Ergebnisse durch einen bestimmten Projekttyp beeinflusst wurden.

#### 7.4.4 Fazit

Die Ergebnisse der Studie zeigen, dass die Verwendung von Interaction-Room-Annotationen einen positiven Effekt auf das Schätzen besitzt. Während anhand der Daten nicht gezeigt werden konnte, dass die Nutzung der Annotationen das gemeinsame Verständnis unter den Schätzenden erhöht, konnte gezeigt werden, dass sich die Schätzgenauigkeit verbessert. Unabhängig von der Nutzung der Annotationen tendierten die Teams zu einer pessimistischen Schätzung. Werden Interaction-Room-Annotationen verwendet, verringert sich der Pessimismus jedoch.

## 8 Fallstudie: Unterstützung der Sprint-Planung durch Interaction-Room-Landkarten

Agile Softwareentwicklung basiert auf den Werten des agilen Manifests [279] und erfordert wegen geringer Dokumentation viel implizites gemeinsames Verständnis bei den Teammitgliedern eines Projekts. Anforderungen werden in Form kurzer User Stories dokumentiert, das Entwicklungsteam plant auf dieser Grundlage eine Entwicklungsiteration, zerlegt Anforderungen in Aufgaben (sog. Tasks) und verwendet die Menge der Tasks zur Fortschrittsmessung in der Entwicklungsiteration [280]. Nach Schwaber und Beedle erkennt das Entwicklungsteam jedoch erst 40 % der Tasks während der Entwicklungsiteration [105], was eine präzise Fortschrittsmessung nahezu unmöglich macht.

Bei einem mittelständischen Softwaredienstleister, der bestandführende Systeme für gesetzliche Krankenversicherungen entwickelt, wurde in einem Projekt festgestellt, dass keine zuverlässige Prognose über den Erfolg von Entwicklungsiterationen getätigt werden konnte, weil 20 bis 30 % der Tasks erst während einer Entwicklungsiteration erkannt wurden. Die folgenden Fallstudien berichten von einer Unterstützung der Iterationsplanung durch den Interaction Room und geben die Ergebnisse der Fallstudie aus Grapenthin, Poggel, Book und Gruhn [113] sowie eine Überprüfung der ursprünglichen Fallstudie durch eine weitere (vgl. [281]) wieder.

### 8.1 Ausgangssituation

Scrum ist das am weitesten verbreitete Vorgehensmodell für agile Softwareentwicklung [282]. Scrum unterteilt Projekte in kurze Iterationen, jede Iteration dauert zwischen einer und vier Wochen. Eine Iteration besteht aus einer Iterationsplanung (sog. Sprint-Planning-Meeting), der Implementierung und dem Testen eines Produktinkrements (sog. Sprint), der Präsentation und Abnahme des Produktinkrements (sog. Sprint Review) und einer Reflexion der eigenen Arbeit vergangener Iteration (sog. Sprint Retrospektive) [280]. Scrum definiert drei Rollen eines sogenannten Scrum-Teams. Der Scrum Master verantwortet den Prozess und räumt Hindernisse aus dem Weg, der Product Owner (PO) ist produktverantwortlich und repräsentiert den Kunden. Das Entwicklungsteam (z. B. Entwickler, Architekten, Tester) setzt die Anforderungen des Product Owners um. Die Mitglieder des Entwicklungsteams arbeiten selbstorganisiert, was unter anderem bedeutet, dass sie die Menge der umzusetzenden Anforderungen je Sprint bestimmen



[ebenda]. In welcher Reihenfolge die Anforderungen (sog. Backlog Items oder kurz Items) umzusetzen sind, priorisiert der PO. Im Sprint-Planning-Meeting 1 stellt der PO die priorisierten Items und das Sprint-Ziel vor. Das Team verpflichtet sich, eine selbstdefinierte Menge Items umzusetzen (sog. Commitment). Wie viele Items umgesetzt werden können, wird auf Grundlage des geschätzten Aufwands der Items und der Kapazität des Teams (sog. Velocity) bestimmt. Die Velocity wird präziser, je mehr Sprints ein Team durchläuft, weil die Teammitglieder Erfahrung mit dem System und der eigenen Leistungsfähigkeit gewinnen [273]. Im Sprint-Planning-Meeting 2 (SPM2) werden die Items des Sprints in Tasks heruntergebrochen (sog. Task Breakdown). Die Menge aller Tasks wird verwendet, um eine Fortschrittmessung (sog. Burndown) während eines Sprints durchzuführen.

Das Ableiten von Tasks aus Items ist keine triviale Aufgabe, denn das Team muss detailliertes Wissen über fachliche und technische Aspekte des zu entwickelnden Systems kennen und einen Transfer leisten, wenn Items aus der Perspektive von Benutzern geschrieben sind. Sofern Tasks nicht während des Sprint-Planning-Meetings 2 identifiziert werden, sondern erst während des Sprints, ist anzunehmen, dass die zu Grunde liegende Anforderung nicht ausreichend verstanden wurde. Neben der daraus erwachsenden Herausforderung, eine angemessene Systemarchitektur zu entwerfen, kann der Fortschritt nicht zuverlässig gemessen und so sichergestellt werden, dass die geplanten Items innerhalb der vorgesehenen Dauer des Sprints (sog. Timebox) realisiert werden [106]. Mit einem Burndown Chart wird der Fortschritt während eines Sprints visualisiert, indem die Menge offener Aufgaben (y-Achse) gegenüber der Zeit (x-Achse) abgetragen wird [ebenda]

In einem mittelständischen Softwareunternehmen wurde ein bestandsführendes System für gesetzliche Krankenversicherungen entwickelt, das über 3000 Masken besitzt und die meisten Geschäftsvorfälle der gesetzlichen Krankenversicherung abdeckt. In dreimonatigen Teilprojekten organisiert, entwickelten mehrere Scrum Teams parallel Komponenten des Systems. In dieser Fallstudie wurde ein Scrum Team (sechs Entwickler, ein Product Owner, ein Scrum Master) über den Zeitraum von zwei Teilprojekten begleitet. Im ersten Teilprojekt (Projekt A) wurde eine Komponente zum Empfang, zur automatischen Verarbeitung, Speicherung und Verteilung von extern zugelieferten Daten implementiert. Das zweite Teilprojekt (Projekt B) besaß eine vergleichbare Komplexität: Es wurde eine Komponente für die Erstattung von Ausgaben für Ärzte implementiert, die keinem Patienten zuzuordnen sind. Das Projekt B wurde untersucht, um festzustellen, ob die Ergebnisse aus Projekt A spezifisch für das Projekt waren oder in Projekt B wiederholt beobachtet werden können.

## 8.2 Problemstellung

Zu Beginn des Projektes A führte das Scrum Team den Task Breakdown „nach Gefühl“ durch, anstatt mit Hilfe eines methodischen Vorgehens, was zu einer unvollständigen Identifikation der Tasks führte. Dabei wurde das SPM2 in separaten Gruppen durchgeführt: Die sechs Entwickler bildeten drei Zweiergruppen, jede Gruppe hat einen Teil der Items heruntergebrochen und die Tasks auf Karten notiert. Die Tasks wurden knapp dokumentiert (z. B. Datenbanktabelle für Sanitätsdienst erstellen), sodass nur Entwickler, die die Tasks geschrieben haben, wussten, was im Detail zur Erfüllung des Tasks zu tun ist (z. B. welche Attribute und Relationen zu anderen Tabellen die Datenbanktabelle besitzen muss). Die Tasks wurden an eine Pinnwand geheftet, jedoch den anderen Teammitgliedern nicht vorgestellt, sodass zwei Drittel der Teammitglieder keine Details über Tasks zur Umsetzung von Items besaßen. Sofern andere Entwickler Tasks erfüllen mussten, die sie nicht selbst heruntergebrochen hatten, entstand ein zusätzlicher Erklärungsaufwand. In einzelnen Fällen konnte sich gegen Ende eines Sprints niemand im Team erinnern, was ein Task bedeutet und wie er zur Erfüllung eines Items beiträgt.

Die Verteilung der Expertise über die Teammitglieder war gleichmäßig, allerdings besaß jedes Teammitglied eine Spezialisierung in der Persistenz-Schicht, der Logik-Schicht oder der Präsentationsschicht. Die Items waren vertikal geschnitten [273], jedes Item beinhaltete einen Teil der Persistenz-, Logik- und Präsentationsschicht, sodass durch die Zweiergruppen der Task Breakdown nicht unter Berücksichtigung der Spezialisierung aller Teammitglieder stattfand, was Einfluss darauf gehabt haben kann, dass die Menge identifizierter Aufgaben unvollständig blieb.

Abbildung 45 zeigt zwei Burndown Charts eines früheren Teilprojekts des Scrum Teams. Im linken Burndown Chart wurden 41 Tasks im SPM2 und weitere 16 Tasks während des Sprints identifiziert, sodass 28 % der Tasks während des Sprints identifiziert wurden. Im rechten Burndown Chart wurden 38 Tasks im SPM2 identifiziert und 12 weitere Tasks während des Sprints, was 24 % spät identifizierte Tasks bedeutet. In Summe wurden 26 % der Tasks erst während eines Sprints identifiziert.

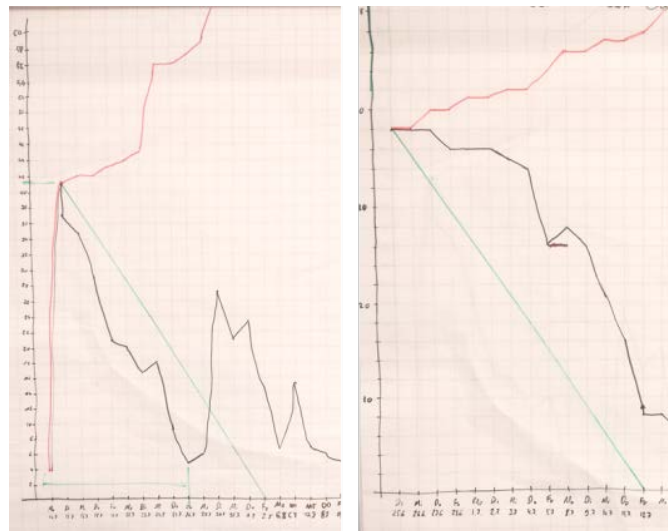


Abbildung 45: Burndown Charts von zwei Sprints vor der Einführung des Interaction Rooms

Diese Beobachtung führte zu dem Schluss, dass das Scrum Team Schwierigkeiten besaß, Tasks aus Items abzuleiten. Unter der Annahme, dass der Interaction Room zu einem besseren Verständnis der Items beiträgt, wurde angenommen, dass mehr Tasks während des SPM2 identifiziert werden, wenn Methodenbausteine des Interaction Rooms im SPM2 eingesetzt wird.

### 8.3 Lösungsansatz

Um das Scrum Team mit einem strukturierteren und gleichzeitig pragmatischen Ansatz beim Task Breakdown zu unterstützen, wurde vorgeschlagen, den Task Breakdown vom gesamten Scrum Team und mit Hilfe von Methodenbausteinen des Interaction Rooms durchzuführen, anstatt wie bisher ausschließlich vom Entwicklungsteam in Zweiergruppen.

Ein SPM2 dauerte zwischen einer und vier Stunden in Abhängigkeit der Länge des Sprints, was die Herausforderung bei der Einführung des Vorgehens barg, die leichtgewichtige Sprint-Planung nicht zu behindern oder zu verzögern. Der Scrum Master wurde mit der Moderation des Interaction Rooms betraut und der PO zur Klärung fachlicher Fragen als Teil des Entwicklungsteams verstanden. Jedes Item wurde nach dem in Abbildung 46 dargestellten Vorgehen detailliert.

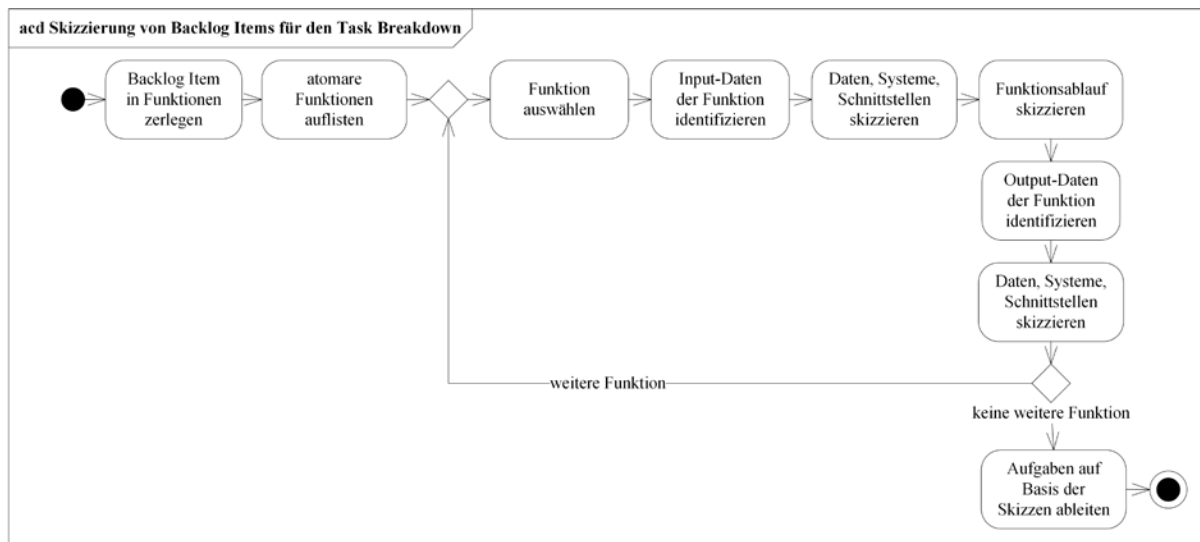


Abbildung 46: Vorgehen zur Skizzierung von Backlog Items für den Task Breakdown (angelehnt an [113])

Zunächst wurde ein Item in atomare Funktionen zerlegt, woraufhin für jede Funktion die benötigten Daten identifiziert wurden. Sofern die Daten durch eine Nutzerinteraktion erzeugt wurden, wurden die notwendigen Oberflächenelemente auf einer Interaktionslandkarte<sup>9</sup> skizziert. Sofern die Daten technisch bereitgestellt wurden, skizzierte das Scrum Team die liefernden Systeme oder Komponenten und Schnittstellen auf einer Integrationslandkarte. In beiden Fällen wurden die Datenstrukturen zur Speicherung auf einer Objektlandkarte skizziert, sofern eine Speicherung der Daten erforderlich war. Daraufhin wurde die Funktion als Teil einer Prozesslandkarte skizziert, um im Anschluss die Output-Daten der Funktion zu identifizieren und auf den gleichen Landkarten, wie zuvor beschrieben, skizziert. Dieses Vorgehen wiederholte sich, nicht bis eine vollständige Spezifikation erstellt wurde, sondern bis die Teammitglieder der Auffassung waren, dass sie genug Informationen über die benötigte Funktionalität und ihre Abhängigkeiten besitzen. Nachdem alle Funktionen detailliert wurden, hat das Scrum Team den Task Breakdown durchgeführt, indem Tasks auf Haftnotizen notiert und diese auf die jeweiligen Elemente der Landkarten geklebt wurden (vgl. Abbildung 47).

<sup>9</sup> Die Interaktionslandkarte ist, wegen mangelnder praktischer Relevanz, nicht Bestandteil dieser Arbeit, wird jedoch im Ausblick (vgl. Kapitel 9) kurz diskutiert.

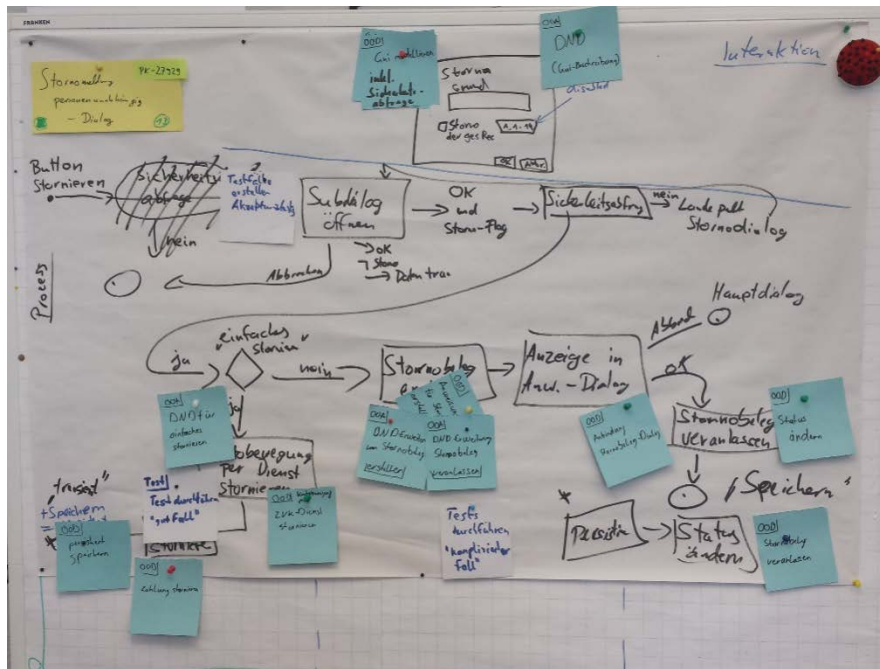


Abbildung 47: Task Breakdown mit Hilfe von Interaction-Room-Landkarten [113]

## 8.4 Ergebnisse

Um die Nützlichkeit des zuvor beschriebenen Vorgehens zu untersuchen, wurde das Projekt A quantitativ untersucht, das Scrum Team während der SMP2 beobachtet und am Ende des Projektes befragt (vgl. [113]). Weil das Projekt A vielversprechende Resultate zeigte, wurde die Fallstudie in Projekt B erneut durchgeführt und quantitativ untersucht sowie mit Hilfe eines dezidierten Beobachters qualitative Erkenntnisse gewonnen (vgl. [281]).

### 8.4.1 Quantitative Ergebnisse aus Projekt A

Nachdem der Interaction Room eingeführt wurde, wurde die Performanz des Teams über drei Sprints in Projekt A beobachtet, um die Wirksamkeit der Intervention zu messen. Als Indikator der Wirksamkeit wurden die Tasks dokumentiert, die während der Sprints identifiziert wurden. In den drei Sprints wurden 19 Items implementiert, für die insgesamt 155 Tasks erstellt und abgearbeitet wurden, wovon 145 Tasks während der SPM2 und zehn Tasks während der Sprints identifiziert wurden. Es wurden also 6,5 % der Tasks während des Sprints identifiziert. Abbildung 48 zeigt die Burndown Charts der drei Sprints. Die Anzahl identifizierter Tasks blieb für die meisten Sprints ungefähr gleich (dunkelste Linien) und erhöhte sich nur geringfügig im Vergleich zu den Sprints vor Projekt A (vgl. Abbildung 45).

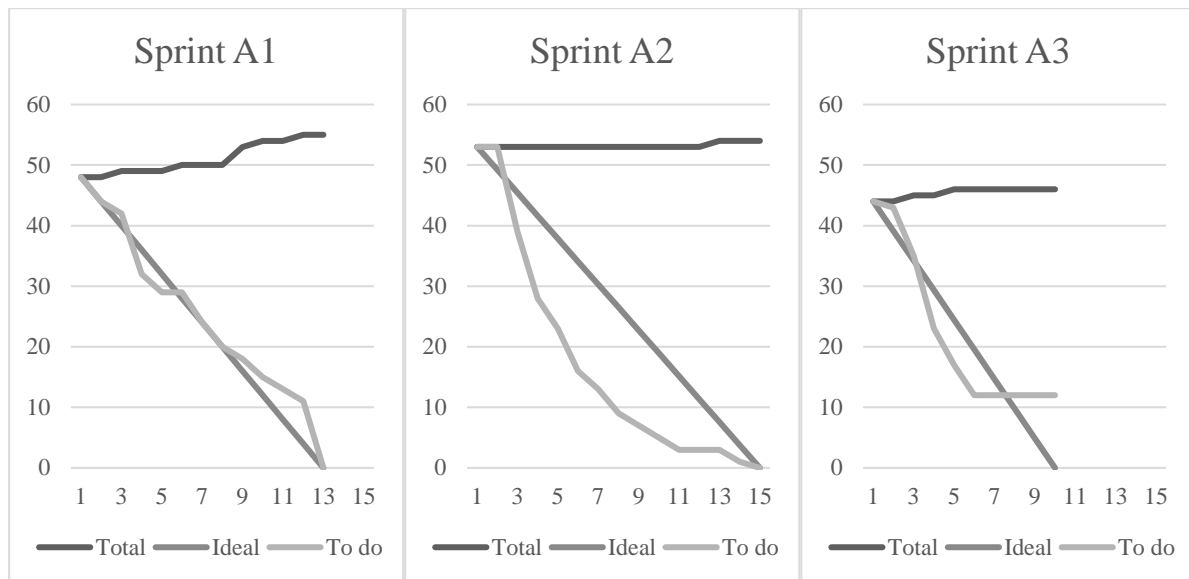


Abbildung 48: Burndown Charts der drei Sprints aus Projekt A nach Einführung des Interaction Rooms

Im Sprint A1 hat das Team sechs Items implementiert, wovon 48 Tasks im SPM2 und sieben Tasks (12,7 %) während des Sprints identifiziert wurden. In Sprint A2 wurden sieben Items implementiert, wovon 53 Tasks im SPM2 und ein Task (1,8 %) während des Sprints identifiziert wurden. Der Sprint A3 dauerte nur zwei anstatt vier Wochen, weil projektweit zwei Wochen pro Quartal für die Integration der entwickelten Funktionalität in das Produktivsystem angesetzt wurden. In Sprint A3 wurden sechs Items implementiert, 44 Tasks wurden im SPM2 und zwei zusätzliche Tasks (4,3 %) während des Sprints identifiziert (vgl. Tabelle 29).

Tabelle 29: Ergebnisse des Task Breakdowns nach Einführung des Interaction Rooms in Projekt A

<i>Sprint</i>	<i>Während des SPM2 identifizierte Tasks</i>	<i>Während des Sprints identifizierte Tasks</i>	<i>Anteil der spät identifizierten Tasks</i>
A1	48	7	12,7%
A2	53	1	1,8%
A3	44	2	4,3%

Der Vergleich von identifizierten Tasks während eines Sprints vor und nach Einführung des Interaction Rooms in Projekt A zeigt, dass deren Anteil von durchschnittlich 26 % vor der Einführung auf durchschnittlich 6,5 % nach der Einführung gesenkt werden konnte.

### 8.4.2 Quantitative Ergebnisse aus Projekt B

Abbildung 49 zeigt die Burndown Charts der drei Sprints aus Projekt B. Die dunkelste Linie stellt die Summe aller Tasks dar und steigt meist nur leicht, was zeigt, dass die Mehrheit der Tasks während der SPM2 identifiziert wurde und auf einen reibungslosen Ablauf der Task Breakdowns hinweist.

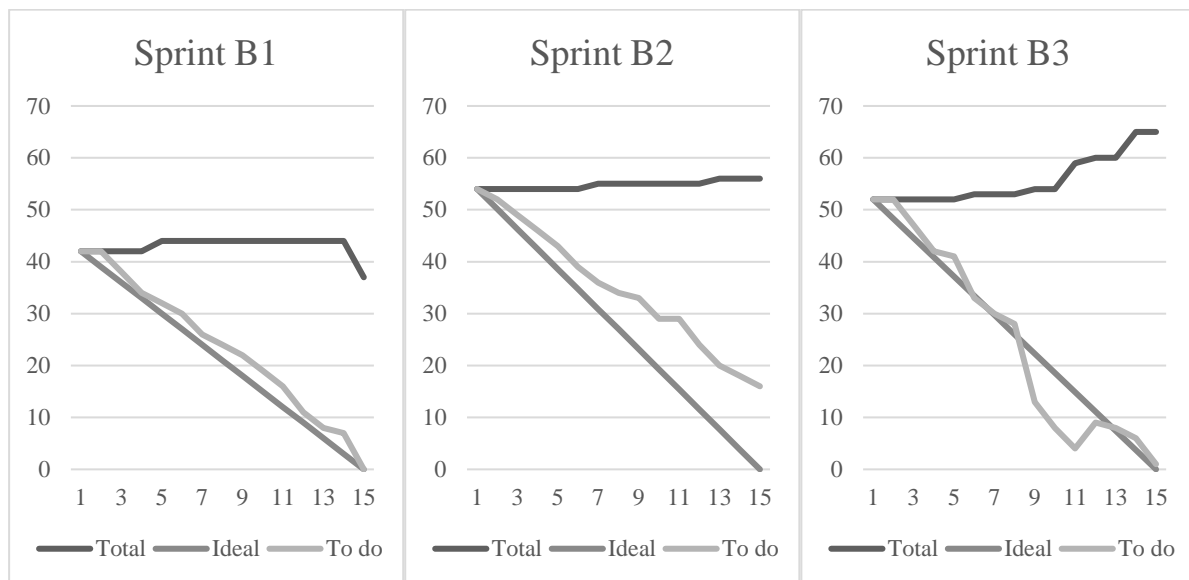


Abbildung 49: Burndown Charts der drei Sprints aus Projekt B nach Einführung des Interaction Rooms

Zwei unübliche Linienvläufe erfordern eine Erklärung: In Sprint B1 wurde ein Item, das in sieben Tasks heruntergebrochen wurde, am Ende des Sprints vom Kunden gestrichen, was zum plötzlichen Fall der dunkelsten Linie führte. In Sprint B3 realisierte das Team alle Tasks in zwei Drittel der veranschlagten Zeit, sodass das Team eigenverantwortlich zwei weitere Items implementierte. Die beiden Sprünge in der dunkelsten Linie zeigen die hinzugefügten Tasks für die beiden Items, die wiederum mit Hilfe eines kleinen Interaction Rooms identifiziert wurden. Da weder die Änderungen in Sprint B1 noch in Sprint B3 auf die späte Identifikation von Tasks zurückzuführen sind, sondern auf strategische Entscheidungen, wurden diese in Tabelle 30 nicht berücksichtigt.

In Sprint B1 wurden drei Items implementiert, von den insgesamt 44 Tasks wurden 42 im SPM2 und zwei (4,5 %) während des Sprints identifiziert. In Sprint B2 hat das Team sieben Items implementiert, die in 56 Tasks heruntergebrochen waren, wovon zwei Tasks (3,6 %) während des Sprints identifiziert wurden. In Sprint B3 wurden zehn Backlog Items in 54 Tasks heruntergebrochen, von denen zwei Tasks (3,7 %) während des Sprints identifiziert wurden.

Insgesamt hat das Team in Projekt B 154 Tasks komplettiert, wovon sechs Tasks während der Sprints identifiziert wurden.

Tabelle 30: Ergebnisse des Task Breakdowns nach Einführung des Interaction Rooms in Projekt B

<i>Sprint</i>	<i>Während des SPM2 identifizierte Tasks</i>	<i>Während des Sprints identifizierte Tasks</i>	<i>Anteil der spät identifizierten Tasks</i>
B1	42	2	4,5 %
B2	54	2	3,6 %
B3	52	2	3,7 %

In den Sprints des Projekts B blieb die Anzahl der spät identifizierten Tasks mit durchschnittlich 3,9 % gering. Dies bedeutet, dass die Ergebnisse aus Projekt A nicht projektspezifisch sind.

#### 8.4.3 Qualitative Ergebnisse aus Projekt A

Die Beobachtung des Scrum Teams während Projekt A zeigte zwei positive Nebeneffekte: Weil das Scrum Team fast alle Tasks während des SPM2 identifizierte, war eine zuverlässigere Vorhersage über den Ausgang des Sprints möglich und das Team konnte durch die präziseren Daten ihre Velocity besser messen. Der zweite positive Nebeneffekt lag in der kollaborativen Erstellung der Skizzen und dem anschließenden Task Breakdown, wodurch jedes Teammitglied in der Lage war, Tasks zu verstehen oder, ggf. mit geringem Zusatzaufwand, umzusetzen, denn das Wissen über die Items und Tasks war gleichverteilter als vorher.

Zusätzlich wurden die Teammitglieder nach ihrem Eindruck über den neuen Ansatz gefragt: Sie berichteten, dass sie sich früher manchmal unsicher fühlten, warum und wie ein Task umzusetzen ist; außerdem, dass sich durch die kollaborative Erstellung der Skizzen ein besseres gemeinsames Verständnis über die Items, heruntergebrochene Tasks und deren Zweck eingestellt hat. Dies war insbesondere für dieses Projekt nützlich, weil die Items separat entworfen wurden und nicht im Kontext des gesamten Systems (und damit anderen Items) standen. Items im Gesamtzusammenhang zu entwerfen, könnte einen zusätzlichen Nutzen im Sinne der vollständigen Identifikation von Tasks im Rahmen eines SPM2 stiften.



#### 8.4.4 Qualitative Ergebnisse aus Projekt B

In Projekt B wurden die Teammitglieder während der SPM2 durch eine dezidierte Person beobachtet, die zum Beispiel notierte, welche Aspekte im SPM2 diskutiert und welche auf den Landkarten des Interaction Rooms skizziert wurden. Außerdem sollte beobachtet werden, ob die Reduktion der Tasks, die während des Sprints identifiziert wurden, auf andere Aspekte aus einem SPM2 zurückzuführen ist. In der Analyse der Beobachtungen fielen zwei Aspekte auf, die eine zusätzliche Erklärung liefern, wie der Interaction Room das Verständnis von Items und entsprechenden Tasks unterstützt.

##### 8.4.4.1 Individueller Umgang mit Entwicklungs- und Test-Tasks

Es war zu beobachten, dass das Verständnis der identifizierten Tasks nicht ausschließlich auf die Methode zurückzuführen war, sondern auch auf die involvierten Rollen und deren Kommunikation: In Projekt A wurde das Team zum gemeinsamen Task Breakdown motiviert, in Projekt B stellte sich heraus, dass das nicht immer notwendig ist.

In Projekt B war ein Domänenexperte Teil des Scrum Teams und für die korrekte Umsetzung der Geschäftsprozesse verantwortlich, ohne Quellcode zu schreiben. Auf der einen Seite hat der Domänenexperte den PO bei der Erstellung von Items unterstützt, auf der anderen Seite fungierte er als fachlicher Ansprechpartner für die restlichen Teammitglieder. Entsprechend der Definition von Scrum könnte argumentiert werden, dass eine solche Rolle nicht Teil des Scrum Teams ist, weil der PO alleine für die Definition und Priorisierung der Items sowie für die Beantwortung von fachlichen Fragen verantwortlich ist. Allerdings ist es auch nicht verboten, wenn der PO einen Domänenexperten zur Unterstützung hinzuzieht.

Der Domänenexperte hatte eine weitere Rolle inne: Als Experte für die Qualitätssicherung (QS) war er verantwortlich für das Erstellen von Testfällen für Items und das Testen implementierter Items für die Abnahme im Sprint Review. Der QS-Experte arbeitete häufig alleine während des Task Breakdowns, jedoch in Abstimmung mit dem PO. Nach dem Task Breakdown erläuterte er den übrigen Teammitgliedern seine Tasks. Gelegentlich führte diese Erläuterung zu neuen Entwicklungstasks für einzelne Items. Im Gegensatz zum isolierten Task Breakdown vor Projekt A besaß die isolierte Identifikation von Test-Tasks keine negative Auswirkung auf Tasks, die während eines Sprints identifiziert wurden.

Dies scheint anzudeuten, dass ein QS-Experte Entwicklungs-Tasks nicht kennen muss, um Test-Tasks erstellen zu können. Für das Entwicklungsteam erscheint es wertvoll, Test-Tasks zu

kennen, um Entwicklungs-Tasks zu erstellen. In Bezug auf den Einsatz des Interaction Rooms bedeutet dies, dass der Task Breakdown von Test- und Entwicklungs-Tasks parallelisiert und dadurch beschleunigt werden kann.

Aufgrund der heterogenen Verteilung von Spezialwissen über Teammitglieder erscheint es praxisrelevant, dass Teammitglieder dezidiert für die Planung der Tests zuständig sind, was die zuvor beschriebenen Ergebnisse auf andere Teams übertragbar macht.

#### 8.4.4.2 Wahl des Abstraktionsniveaus während des Task Breakdowns

Das korrekte Abstraktionsniveau für die Detaillierung von Items durch die Landkarten des Interaction Rooms zu finden, ist keine triviale Aufgabe: wenn das Abstraktionsniveau zu gering ist, würde das Scrum Team mehr Aufwand leisten, als für den Task Breakdown nötig wäre. Wenn die Landkarten hingegen zu abstrakt bleiben, kann das Scrum Team einen falschen Eindruck von der Komplexität eines Items erhalten, was die Identifikation von Tasks während eines Sprints zur Folge haben kann, weil die Herausforderungen der Implementierung nicht ausreichend verstanden wurden.

Während der Beobachtung des Scrum Teams in Projekt B trat ein Effekt auf, der auf ein angemessenes Abstraktionsniveau im SPM2 hinweist: ein Item beinhaltete einen Prozessschritt „Bestelldaten zusammensetzen“, zu dem die Teammitglieder kurz besprachen, welche Daten zusammengesetzt und validiert werden müssen. Diese entschieden daraufhin, dass keine weitere Detaillierung des Prozessschrittes nötig ist, weil die Validierung „nur einige Überprüfungen“ beinhaltet. Deswegen wurde der Prozessschritt nicht auf der Prozesslandkarte skizziert und der Task „implementiere Zusammenstellung der Orderdaten“ abgeleitet und mit einem Personentag Aufwand geschätzt. Während des Sprints benötigte das Team zehn Personentage für die Implementierung des Tasks, im Nachhinein wurden für das Item vier Defekte festgestellt.

Dieses Beispiel illustriert, dass ein zu hohes Abstraktionsniveau zu Defekten und falscher Aufwandsschätzung führen kann, was zu der Empfehlung führte, dass jedes Item mindestens so detailliert werden sollte, dass es über den Status einer Black Box hinausgeht, damit versteckte Herausforderungen erkannt werden können.

## 8.5 Diskussion

Abschließend wird die Validität der Ergebnisse der Fallstudie zur Planung von Entwicklungsiterationen mit Hilfe von Interaction-Room-Landkarten diskutiert und ein Fazit gezogen.

### 8.5.1 Validität der Ergebnisse

Allein die Erfahrung aus den Projekten A und B reicht nicht aus, um mit Sicherheit zu wissen, dass die beobachteten Ergebnisse einzig auf die Einführung des Interaction Rooms zurückzuführen sind – zwei andere Aspekte können dazu beigetragen haben: Die zunehmende Erfahrung des Scrum Teams und die variierende Komplexität der Items in den Sprints.

Nichtsdestotrotz war die Granularität der Tasks in den Sprints eher homogen und die abrupte Reduktion um ca. 20 % identifizierter Tasks während eines Sprints müsste mit einem entsprechend sprunghaften Erfahrungszuwachs zusammenhängen. Die Tatsache, dass Projekt B ebenfalls eine geringe Anzahl identifizierter Tasks während eines Sprints aufwies, deutet an, dass ein Lerneffekt keine Rolle gespielt hat. Es ist außerdem auszuschließen, dass ein „Beobachter-Effekt“ (z. B. besondere Bemühungen der Teammitglieder, alle Tasks zu identifizieren, weil dies als Metrik für deren Performanz verwendet wird) Einfluss auf die Ergebnisse hatte, weil die Teammitglieder nicht wussten, dass die spät identifizierten Tasks nachverfolgt werden, nachdem der Interaction Room eingeführt wurde. Weil die Fallstudie in einem großen Industrieprojekt durchgeführt wurde, ließen sich organisatorische Schwierigkeiten beobachten, die für solche Projekte üblich sind (z. B. Änderungen des Managements, der Anforderungen, des Vorgehens oder der Abgabetermine). Die analysierten Daten wurden aus einer Phase des Projektes erhoben, in der es keine außergewöhnlichen Ereignisse im Gesamtprojekt gab, die die Zusammensetzung der Anforderungen und die Durchführung der Sprints beeinflusst hätte. Auch wenn die Analyse von zwei Projekten, die über einen längeren Zeitraum unkompliziert verlaufen wären und eine derartige Kontrolle über potentielle Störvariablen zugelassen hätten, bevorzugt worden wäre, sind diese Projekte außerhalb akademischer Konstellationen (z. B. Projekte im Rahmen von Lehrveranstaltungen) schwer zu finden. Dies limitiert die Generalisierbarkeit der Ergebnisse auf andere Projektkonstellationen. Trotzdem liefern die Ergebnisse die Erkenntnis, dass ein effektiver Task Breakdown in Industrieprojekten möglich ist, auch wenn die Umstände nicht idealtypisch sind.

### 8.5.2 Fazit

Die Beobachtung der Sprints in den Projekten A und B haben eine Reduktion der spät erkannten Tasks, von durchschnittlich 26 % auf ungefähr 5 % gezeigt – deutlich weniger als die von Schwaber und Beedle als unvermeidlich titulierten 40 % [105]. Zurückzuführen ist dieser Effekt vermutlich auf die umfassendere Analyse der Items durch die Landkarten des Interaction Rooms und die Diskussion der Items im gesamten Scrum Team während der Skizzierung funktionaler und struktureller Aspekte der Anforderungen.

Die Fallstudie war nicht auf den Entwurf eines formalen Ansatzes zur Verbesserung des Task Breakdowns ausgelegt. Das liegt daran, dass der Interaction Room für die Kommunikation und Kollaboration über relevantes Verhalten und relevante Strukturen eines Informationssystems, ferner für die Identifikation von Wert-, Aufwands- und Risikotreibern, der informellen Skizzierung dieser Aspekte und der pragmatischen Lösung offener Fragen ausgelegt ist. Zu welchem Grad die einzelnen Bestandteile Einfluss auf das beobachtete Ergebnis besaßen, würde ein formales Experiment erfordern, in dem die einzelnen Variablen isoliert und kontrolliert werden müssten.

Trotzdem scheint die gemeinsame Arbeit der Stakeholder mit dem Interaction Room einen vollständigeren Überblick über die zu leistenden Tasks als Folge aufzuweisen. Es kann argumentiert werden, dass der Ansatz, einen Task Breakdown in Zweiergruppen durchzuführen, falsch und durch den Effizienzgedanken motiviert war, Entwickler nur die Items herunterbrechen zu lassen, die später durch sie implementiert werden. Auch wenn Entwickler später Items implementiert haben, die nicht zur Umsetzung durch sie vorgesehen waren, hätten die Entwickler, die den Task Breakdown durchführten, Tasks ausführlicher beschreiben können. Den Task Breakdown ausschließlich durch das Entwicklungsteam durchführen zu lassen und so das Wissen der Domänenexperten nicht einzubeziehen, kann auch als Fehler gängiger Scrum-Praktiken angesehen werden. Durch den Interaction Room kann dieses Wissen in den Task-Breakdown-Prozess einbezogen und explizit auf komplexe oder ungewisse Aspekte fokussiert werden.

## 9 Fazit

In diesem Kapitel wird der wissenschaftliche Beitrag dieser Arbeit zusammengefasst, die Grenzen dieser Arbeit diskutiert und ein Ausblick auf weitere Forschungsfragen gegeben.

### 9.1 Beiträge

Der Interaction Room ist eine Methode für das Requirements Engineering, die in Analyse, Neuentwicklungs-, Weiterentwicklungs- und Migrationsprojekten angewendet werden kann. Gegenstand eines Projekts ist ein Informationssystem, das im Rahmen von moderierten Interaction-Room-Workshops durch ein interdisziplinäres Team analysiert und konzipiert wird. In Interaction-Room-Workshops werden Interaction-Room-Landkarten und Interaction-Room-Annotationen verwendet. Welche Landkarten und Annotationstypen verwendet werden, wird durch eine vorbereitende Kontextanalyse situationsspezifisch festgelegt (vgl. Kapitel 5.1). Um die funktionalen Anforderung an ein Informationssystem zu erheben und zu dokumentieren, werden die Interaction-Room-Landkarten verwendet (vgl. Kapitel 5.2). Mit der Feature-Landkarte und der Prozesslandkarte werden relevante Ausschnitte des Verhaltens eines Informationssystems skizziert, mit der Objektlandkarte und Integrationslandkarte werden relevante Ausschnitte der Struktur eines Informationssystems skizziert. Für jede Landkarte wurde eine Syntax, Semantik und Pragmatik definiert (vgl. Kapitel 5.2.1 bis 5.2.4), die interdisziplinäre Teams dabei unterstützen, ein gemeinsames Verständnis über die funktionalen Anforderungen eines Informationssystems zu erlangen. Um den unterschiedlichen Wert, nicht funktionale Anforderungen und Risiken explizit zu machen, werden die Interaction-Room-Annotationen verwendet (vgl. Kapitel 5.3). Mit Hilfe von Annotationstypen kennzeichnen Stakeholder Wert-, Aufwands- und Risikotreiber auf den Landkarten. Nachdem die Experten die Landkartenelemente individuell gekennzeichnet haben, wird jeder verwendete Annotationstyp durch die Beantwortung typspezifischer Anschlussfragen diskutiert und die Begründung eines Annotationstyps dokumentiert (vgl. Kapitel 5.3.1 bis 5.3.5). Zur Verwendung der Erkenntnisse aus Interaction-Room-Workshops in nachgelagerten Aktivitäten im Softwareprozess wurde eine Abbildung der Erkenntnisse auf eine Struktur für Anforderungsdokumente entwickelt (vgl. Kapitel 5.4).

Für die Evaluation des Interaction Rooms wurde dessen Anwendung in 23 Industrieprojekten untersucht und es wurde erwartungsgemäß festgestellt, dass zwischen Metriken der

Projektgröße, zwischen zur Verwendung vorgeschlagenen und tatsächlich verwendeten Annotationstypen und zwischen der Menge verwendeter Annotationen und Duplikaten ein positiver Zusammenhang besteht (vgl. Kapitel 6).

Die Interaction-Room-Annotationen zeigen einen positiven Einfluss auf die Aufwandsschätzung, indem Schätzungen unter Zuhilfenahme der Annotationen genauer werden und sich die Verzerrung durch Unterschätzen verringert (vgl. Kapitel 7). In der durchgeführten Fallstudie haben die Annotationen außerdem zum gemeinsamen Verständnis unter den Schätzenden beigetragen, was jedoch empirisch nicht belegt werden konnte.

Durch die Interaction-Room-Landkarten wurde die Planung von Entwicklungsiterationen unterstützt, indem zu leistende Arbeitspakete einer Entwicklungsiteration in zwei Industrieprojekten vollständiger identifiziert wurden als ohne den Einsatz der Landkarten (vgl. Kapitel 8). Aus der Einführung eines pragmatischen Vorgehens zur Konkretisierung von Entwicklungsaufgaben mit den Landkarten resultierte, dass die Entwicklungsteams ihren Fortschritt während einer Iteration zuverlässiger messen und eine präzisere Erfolgsprognose geben konnten. Dabei empfanden die Probanden der Fallstudie das Vorgehen mit den Interaction-Room-Landkarten als hilfreich und den zusätzlichen Aufwand als nützlich.

## 9.2 Diskussion

In diesem Kapitel werden die Grenzen dieser Arbeit kurz diskutiert. Die kritische Betrachtung führt unmittelbar zu weiteren Forschungsfragen, die in Kapitel 9.3 diskutiert werden.

Das Ergebnis von Interaction-Room-Workshops sind Dokumente, in denen Information aus den Landkarten und Annotationen verarbeitet werden. Durch das Schreiben von Dokumenten besitzt der Interaction Room eine Anschlussfähigkeit an plangetriebene Vorgehensmodelle. Auf Grundlage eines solchen Dokuments ist es möglich, ein Product Backlog für eine agile Entwicklung zu befüllen (vgl. Kapitel 7.2.1), wodurch die Ergebnisse von Interaction-Room-Workshops mittelbar in agilen Vorgehensweisen verwendet werden können.

Der Interaction Room wurde explizit für die Analyse und Konzeption von Informationssystemen entworfen. Die Anwendbarkeit des Interaction Rooms für andere Systemklassen (z. B. eingebettete Systeme, cyber-physikalische Systeme, mobile Systeme) wurde in dieser Arbeit nicht betrachtet, woraus sich eine Forschungsfrage ergibt, die im folgenden Kapitel diskutiert wird.

Der Interaction Room wurde größtenteils mit Partnern aus der Finanzdienstleistungs- und Technologiebranche eingesetzt. Daher sind die Beiträge dieser Arbeit insbesondere für diese Branchen relevant. Eine geringe Anzahl der untersuchten Projekte wurde in anderen Branchen durchgeführt (vgl. Kapitel 6). Die Evaluation und die gesammelte Erfahrung stimmen zuversichtlich, dass eine Anwendbarkeit in anderen Branchen ohne Anpassungen des Interaction Rooms möglich ist.

Die Ergebnisse der Fallstudie zur Unterstützung der Planung von Entwicklungsiterationen durch Interaction-Room-Landkarten basieren auf zwei Projekten. Die Wiederholung der Fallstudie mit einer größeren Stichprobe und die empirische Untersuchung der Daten würde die Generalisierbarkeit der Ergebnisse stärken.

### 9.3 Ausblick

Im Folgenden werden weitere Forschungsfragen diskutiert. Dazu wird erst auf die Integration der Methoden in den Softwareprozess eingegangen, um daraufhin Forschungsfragen zu diskutieren, die sich aus den Grenzen der Arbeit ergeben und deren Beantwortung zur Erweiterung der Methode führen kann.

Der Interaction Room ist eine Methode für das Requirements Engineering, mit der sowohl fachliche als auch technische Anforderungen erhoben, dokumentiert und abgestimmt werden können. Eine weitere Forschungsfrage ist, wie die Erkenntnisse aus dem Interaction Room in den Entwurf einer Softwarearchitektur einfließen können. Insbesondere mit Hilfe der Annotationstypen können Anforderungen formuliert werden, aus denen Implikationen für eine Softwarearchitektur entstehen (vgl. Kapitel 5.3.4). Wie ein methodischer Übergang zur Konzeption einer Softwarearchitektur gestaltet werden muss, um gestellte Anforderungen nachweislich durch eine Softwarearchitektur zu erfüllen, kann Gegenstand zukünftiger Forschung sein.

Die unmittelbare Anschlussfähigkeit an agile Vorgehensweisen ist eine interessante Forschungsfrage, die sich aus den Grenzen dieser Arbeit ergibt. Epics und Features der Feature-Landkarte (vgl. Kapitel 5.2.1) können unmittelbar in ein Product Backlog übernommen werden. Allerdings sind Epics und Features der thematische Rahmen für Interaction-Room-Workshops und werden im Rahmen eines Workshops durch die übrigen Landkarten detailliert. Sofern Epics und Features unmittelbar übernommen werden würden, würde die Information der Prozess-, Objekt- und Integrationslandkarte sowie der Annotationen auf den Landkarten nicht

übernommen werden. Daher sollte in zukünftiger Forschungsarbeit untersucht werden, wie alle Erkenntnisse aus dem Interaction Room in einem Product Backlog dokumentiert und unmittelbar für die agile Entwicklung genutzt werden können.

Eine weitere sich ergebende Forschungsfrage ist, in welchen Phasen agiler Entwicklungsprojekte der Interaction Room eingesetzt werden kann. In der Fallstudie aus Kapitel 8 wurde der Interaction Room zur Unterstützung des Task Breakdowns im Sprint-Planning-Meeting 2 eingesetzt. Wie der Interaction Room Nutzen für das Backlog Grooming, die Releaseplanung und die Durchführung des Sprint-Planning-Meetings 1 stiftet, kann Gegenstand weiterer Forschung sein.

Die Anwendbarkeit des Interaction Rooms für andere Branchen und andere Systemklassen ist eine weitere Forschungsfrage. Für den Einsatz in anderen Branchen wurden bereits erste Erfahrungen gesammelt, die vermuten lassen, dass keine Anpassungen der Methode notwendig ist. Der Interaction Room wurde jedoch nicht für andere Systemklassen als Informationssysteme eingesetzt, sodass eine Überprüfung notwendig wäre, welche Landkarten und Annotationen übernommen werden können und ob Landkarten bzw. Annotationen angepasst oder neu entwickelt werden müssten.

Neben der Anpassung des Interaction Rooms für andere Branchen und Systemklassen kann die Methode dieser Arbeit durch neue und erweiterte Landkarten weiterentwickelt werden. Eine Interaktionslandkarte kann z. B. zur Veranschaulichung der Nutzerinteraktion auf unterschiedlichen Abstraktionsebenen entwickelt werden. Dazu kann ein Navigationsüberblick den strukturellen Aufbau der Menüs und den Ablauf zentraler Dialogflüsse veranschaulichen. Zentrale Masken könnten auf einem Storyboard durch Prototyping von Mock-ups detailliert werden. Darüber hinaus könnte eine Migrationslandkarte veranschaulichen, wie Datenstrukturen aus einem Legacy-System in ein neu entwickeltes System überführt werden. Dazu müssten Datenstrukturen geeignet abstrahiert werden, um darstellen zu können, welche Daten unmittelbar übernommen werden können, welche Daten im Rahmen der Migration wie transformiert werden müssen und welche Daten in Zukunft nicht mehr vom neuen System geführt werden sollen. Auch die existierenden Landkarten könnten weiterentwickelt werden, indem in der Syntax, Semantik und Pragmatik verankert wird, wie Zusatzinformationen (z. B. Rollen, Masken, System) erhoben werden, um Use-Case-Diagramme und initiale Rollen-Berechtigungskonzepte erstellen zu können (vgl. Kapitel 5.4).



Die Elemente der Landkarten stehen in Zusammenhang (z. B. werden Daten von Geschäftsobjekten in Aktivitäten manipuliert und über Schnittstellen gelesen oder geschrieben), sodass auch die Annotationen an den entsprechenden Landkartenelementen in Beziehung (z. B. harmonisierend, neutral oder konkurrierend) zueinander stehen. Die Art des Zusammenhangs lässt sich jedoch nicht pauschal auf der Ebene des Annotationstyps definieren, weil die Begründung eines Annotationstyps ausschlaggebend ist. Um die Art des Zusammenhangs feststellen zu können, ist ein paarweiser Vergleich der Annotationsbegründung notwendig, was sehr aufwendig ist. Gegenstand zukünftiger Forschung kann ein Verfahren sein, auf welcher Grundlage Annotationsgruppen gebildet werden können (z. B. semantische Nähe der Benennung unterschiedlicher Landkartenelemente, Zugehörigkeit von Landkartenelementen zu einer Anforderung), um daraufhin Eigenschaften dieser Gruppen zu definieren, die dazu führen, dass vorqualifiziert werden kann, für welche Gruppen die Analyse der Beziehungsarten zwischen ihren Annotationen den meisten Nutzen verspricht. Durch die Eigenschaften der Annotationsgruppen und die Beziehungen zwischen den Annotationen einer Gruppe könnte Optimierungspotential identifiziert werden, was z. B. zu einer Nutzenmaximierung oder Komplexitätsreduktion der Anforderungen führt. Es kann untersucht werden, wie die gewonnenen Informationen für die Aufwandschätzung und Priorisierung von Anforderungen verwendet werden können und ob dadurch ein weiterer Beitrag zur Wertorientierung und somit zur erfolgreichen Planung von IT-Projekten geliefert werden kann.

## Quellenverzeichnis

- [1] George Colony: *Your Opportunity in the Age of the Customer* | *Forrester Blogs*, 2013. [Online]. Verfügbar unter: [http://blogs.forrester.com/george\\_colony/13-12-10-our\\_opportunity\\_in\\_the\\_age\\_of\\_the\\_customer](http://blogs.forrester.com/george_colony/13-12-10-our_opportunity_in_the_age_of_the_customer). [Zugegriffen: 05-Aug-2016].
- [2] Martinus van Genuchten und Les Hatton: *Software: What's In It and What's It In?*, *IEEE Software*, Bd. 27, Nr. 1, S. 14–16, Januar 2010.
- [3] Jan Bosch: *Speed, Data, and Ecosystems: The Future of Software Engineering*, *IEEE Software*, Bd. 33, Nr. 1, S. 82–88, Januar 2016.
- [4] Christof Ebert und Capers Jones: *Embedded Software: Facts, Figures, and Future*, *Computer*, Bd. 42, Nr. 4, S. 42–52, April 2009.
- [5] Jürgen Mössinger: *Software in Automotive Systems*, *IEEE Software*, Bd. 27, Nr. 2, S. 92–94, März 2010.
- [6] Yuki Tsuchitoi und Hideki Sugiura: *10 MLOC in Your Office Copier*, *IEEE Software*, Bd. 28, Nr. 6, S. 93–95, November 2011.
- [7] Valeriy Vyatkin: *Software Engineering in Industrial Automation: State-of-the-Art Review*, *IEEE Transactions on Industrial Informatics*, Bd. 9, Nr. 3, S. 1234–1249, August 2013.
- [8] Willy C. Shih: *Does Hardware Even Matter Anymore?*, *Harvard Business Review*, 2015. [Online]. Verfügbar unter: <https://hbr.org/2015/06/does-hardware-even-matter-anymore>. [Zugegriffen: 27-Apr-2017].
- [9] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, und Michael Hoffmann: *Industry 4.0*, *Business & Information Systems Engineering*, Bd. 6, Nr. 4, S. 239–242, August 2014.
- [10] Bundesministerium für Bildung und Forschung: *Zukunftsbild „Industrie 4.0“*, Bonn, 2013.
- [11] Nicholas Negroponte: *Total digital. Die Welt zwischen 0 und 1 – Visionen über das Leben im digitalen 21. Jahrhundert*, München: Goldmann, 1997.
- [12] Robert Charette: *Why software fails*, *IEEE Spectrum*, Bd. 42, Nr. 9, S. 42–49, September 2005.
- [13] Marc Andreessen: *Why Software Is Eating The World*, *Wall Street Journal*, 2011. [Online]. Verfügbar unter: <http://www.wsj.com/articles/SB10001424053111903480904576512250915629460>. [Zugegriffen: 08-Mai-2016].
- [14] Brett King: *Breaking Banks*, New York: John Wiley & Sons, 2014.
- [15] Michael A. Cusumano: *The changing software business: Moving from products to*

- services*, Computer, Bd. 41, Nr. 1, S. 20–27, 2008.
- [16] Statista GmbH: *Statistiken zur Internetnutzung weltweit | Statista*, 2016. [Online]. Verfügbar unter: <http://de.statista.com/themen/42/internet/>. [Zugegriffen: 12-Aug-2016].
- [17] Internet Society: *Global Internet Report*, Washington, 2015.
- [18] Kevin Michael Ayres, Linda Mechling, und Frank J. Sansosti: *The Use of Mobile Technologies to Assist with Life Skills*, Psychology in the Schools, Bd. 50, Nr. 3, S. 259–271, März 2013.
- [19] Yonglin Ren, Richard Werner, Nelem Pazzi, und Azzedine Boukerche: *Monitoring patients via a secure and mobile healthcare system*, IEEE Wireless Communications, Bd. 17, Nr. 1, S. 59–65, Februar 2010.
- [20] Chi-Huang Hung, Ying-Wen Bai, und Ren-Yi Tsai: *Design of blood pressure measurement with a health management system for the aged*, IEEE Transactions on Consumer Electronics, Bd. 58, Nr. 2, S. 619–625, Mai 2012.
- [21] Larry Downes und Paul Nunes: *Big Bang Disruption: Strategy in the Age of Devastating Innovation*, New York: Penguin, 2014.
- [22] Nikolas Roman Herbst, Samuel Kounev, und Ralf Reussner: *Elasticity in Cloud Computing: What It Is, and What It Is Not*, in Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13), S. 23–27, 2013.
- [23] Bent Flyvbjerg und Alexander Budzier: *Why Your IT Project May Be Riskier Than You Think*, Harvard Business Review, Bd. 89, Nr. 9, S. 23–25, 2011.
- [24] June Verner, Jennifer Sampson, und Narciso Cerpa: *What factors lead to software project failure?*, in Proceedings of the 2nd International Conference on Research Challenges in Information Science (RICS 2008), S. 71–80, 2008.
- [25] Charalambos L. Iacovou und Albert S. Dexter: *Surviving IT project cancellations*, Communications of the ACM, Bd. 48, Nr. 4, S. 83–86, April 2005.
- [26] Michael Bloch, Sven Blumberg, und Jürgen Laartz: *Delivering large-scale IT projects on time, on budget, and on value*, 2012. [Online]. Verfügbar unter: [http://www.mckinsey.com/insights/business\\_technology/delivering\\_large-scale\\_it\\_projects\\_on\\_time\\_on\\_budget\\_and\\_on\\_value](http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value). [Zugegriffen: 22-Jan-2016].
- [27] Kurt R. Linberg: *Software developer perceptions about software project failure: a case study*, Journal of Systems and Software, Bd. 49, Nr. 2–3, S. 177–192, Dezember 1999.
- [28] Patrick Hamilton: *Wege aus der Softwarekrise*, Berlin, Heidelberg: Springer, 2008.
- [29] Johann Eveleens und Chris Verhoef: *The rise and fall of the Chaos report figures*, IEEE Software, Bd. 27, Nr. 1, S. 30–36, Januar 2010.
- [30] The Standish Group International: *CHAOS Manifesto 2013: Think Big, Act Small*,

- Boston, 2013.
- [31] Robert L. Glass: *IT Failure Rates – 70% or 10 – 15%?*, IEEE Software, Bd. 22, Nr. 3, S. 112, 110–111, Mai 2005.
- [32] Robert L. Glass: *The Standish report*, Communications of the ACM, Bd. 49, Nr. 8, S. 15–16, August 2006.
- [33] Kjetil Moløkken-Østvold und Magne Jørgensen: *A comparison of software project overruns – flexible versus sequential development models*, IEEE Transactions on Software Engineering, Bd. 31, Nr. 9, S. 754–766, September 2005.
- [34] Kjetil Moløkken-Østvold und Magne Jørgensen: *A review of software surveys on software effort estimation*, in Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE 2003), S. 223–230, 2003.
- [35] Khaled El Emam und Günes A. Koru: *A Replicated Survey of IT Software Project Failures*, IEEE Software, Bd. 25, Nr. 5, S. 84–90, September 2008.
- [36] Narciso Cerpa und June M. Verner: *Why did your project fail?*, Communications of the ACM, Bd. 52, Nr. 12, S. 130–134, Dezember 2009.
- [37] Andrew Taylor: *IT projects: sink or swim*, The Computer Bulletin, Bd. 42, Nr. 1, S. 24–26, Januar 2000.
- [38] Robert L. Glass: *Frequently forgotten fundamental facts about software engineering*, IEEE Software, Bd. 18, Nr. 3, S. 112, 110–111, Mai 2001.
- [39] Robert L. Glass: *Software runaways – some surprising findings*, ACM SIGMIS Database, Bd. 28, Nr. 3, S. 16–19, Juni 1997.
- [40] Hairul Nizam Nasir Mohd und Sahibuddin Shamsul: *Critical success factors for software projects: A comparative study*, Scientific Research and Essays, Bd. 6, Nr. 10, S. 2174–2186, Mai 2011.
- [41] Kate Davis: *Different stakeholder groups and their perceptions of project success*, International Journal of Project Management, Bd. 32, Nr. 2, S. 189–201, Februar 2014.
- [42] Karel de Bakker, Albert Boonstra, und Hans Wortmann: *Does risk management contribute to IT project success? A meta-analysis of empirical evidence*, International Journal of Project Management, Bd. 28, Nr. 5, S. 493–503, Juli 2010.
- [43] David Baccarini und Adam Collins: *The Concept of Project Success – What 150 Australian project managers think*, in Proceedings of the 2004 Australian Institute of Project Management Conference (AIPM 2004), Bd. 68, S. 48–57, 2004.
- [44] Aaron J. Shenhar, Dov Dvir, Ofer Levy, und Alan C. Maltz: *Project Success: A Multi-dimensional Strategic Concept*, Long Range Planning, Bd. 34, Nr. 6, S. 699–725, Dezember 2001.
- [45] Darren Dalcher: *Rethinking Success in Software Projects: Looking Beyond the Failure*

- Factors*, in *Software Project Management in a Changing World*, Günther Ruhe und Claes Wohlin (Hrsg.), Berlin, Heidelberg: Springer, S. 27–49, 2014.
- [46] Anton de Wit: *Measurement of project success*, *International Journal of Project Management*, Bd. 6, Nr. 3, S. 164–170, August 1988.
- [47] John Wateridge: *How can IS/IT projects be measured for success?*, *International Journal of Project Management*, Bd. 16, Nr. 1, S. 59–63, Februar 1998.
- [48] Terry Cooke-Davies: *The “real” success factors on projects*, *International Journal of Project Management*, Bd. 20, Nr. 3, S. 185–190, April 2002.
- [49] Kam Jugdev und Ralf Moller: *A retrospective look at our evolving understanding of project success*, *Project Management Journal*, Bd. 36, Nr. 4, S. 19–31, Dezember 2005.
- [50] Paul L. Bannerman und Alan Thorogood: *Celebrating IT Projects Success: A Multi-domain Analysis*, in *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS 2012)*, S. 4874–4883, 2012.
- [51] Paula Savolainen, Jarmo J. Ahonen, und Ita Richardson: *Software development project success and failure from the supplier’s perspective: A systematic literature review*, *International Journal of Project Management*, Bd. 30, Nr. 4, S. 458–469, Mai 2012.
- [52] Lavagnon A. Ika: *Project success as a topic in project management journals*, *Project Management Journal*, Bd. 40, Nr. 4, S. 6–19, Dezember 2009.
- [53] Melinda L. Korzaan: *The Influence Of Commitment To Project Objectives In Information Technology (IT) Projects*, *Review of Business Information Systems*, Bd. 13, Nr. 4, S. 89–97, Mai 2011.
- [54] Jim Whitehead: *Collaboration in Software Engineering: A Roadmap*, in *Proceedings of the 2007 Future of Software Engineering (FOSE 2007)*, S. 214–225, 2007.
- [55] Object Management Group (OMG): *UML 2.4.1 Superstructure Specification*, Needham, 2004.
- [56] Peter Fettke: *Business Process Modeling Notation*, *Wirtschaftsinformatik*, Bd. 50, Nr. 6, S. 504–507, Dezember 2008.
- [57] Jan Mendling, Hajo A. Reijers, und Jan Recker: *Activity labeling in process modeling: Empirical insights and recommendations*, *Information Systems*, Bd. 35, Nr. 4, S. 467–482, Juni 2010.
- [58] Eric Yu: *Towards modelling and reasoning support for early-phase requirements engineering*, in *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (ISRE 1997)*, S. 226–235, 1997.
- [59] Khaled Emam, Walcelio Melo, und Jean-Normand Drouin: *SPICE: The theory and practice of software process improvement and capability determination*, Los Alamitos: IEEE Computer Society Press, 1997.

- [60] Jeff Sutherland, Carsten Ruseng Jakobsen, und Kent Johnson: *Scrum and CMMI Level 5: The Magic Potion for Code Warriors*, in Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), S. 466–474, 2008.
- [61] Hans Robert Hansen, Jan Mendling, und Gustaf Neumann: *Wirtschaftsinformatik*, Berlin, München, Boston: de Gruyter, 2015.
- [62] Peter Wegner: *Why interaction is more powerful than algorithms*, Communications of the ACM, Bd. 40, Nr. 5, S. 80–91, Mai 1997.
- [63] Klaus Pohl und Chris Rupp: *Basiswissen Requirements Engineering – Aus- und Weiterbildung zum „Certified Professional for Requirements Engineering“: Foundation Level nach IREB-Standard*, 3. Aufl., Heidelberg: dpunkt, 2011.
- [64] Linda Wallace und Mark Keil: *Software project risks and their effect on outcomes*, Communications of the ACM, Bd. 47, Nr. 4, S. 68–73, April 2004.
- [65] Bill Curtis, Herb Krasner, und Neil Iscoe: *A field study of the software design process for large systems*, Communications of the ACM, Bd. 31, Nr. 11, S. 1268–1287, November 1988.
- [66] Michael L. Harris, Rosann Webb Collins, und Alan R. Hevner: *Control of Flexible Software Development Under Uncertainty*, Information Systems Research, Bd. 20, Nr. 3, S. 400–419, September 2009.
- [67] Matthias Book, Simon Grapenthin, und Volker Gruhn: *Seeing the Forest and the Trees: Focusing Team Interaction on Value and Effort Drivers*, in Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering (FSE 2012), S. 1–4, 2012.
- [68] Hadar Ziv, Debra Richardson, und René Klösch: *The uncertainty principle in software engineering*, Irvine, 1997.
- [69] Robert E. Kraut und Lynn A. Streeter: *Coordination in software development*, Communications of the ACM, Bd. 38, Nr. 3, S. 69–81, März 1995.
- [70] Alberto Sillitti, Martina Ceschi, Barbara Russo, und Giancarlo Succi: *Managing Uncertainty in Requirements: A Survey in Documentation-Driven and Agile Companies*, in Proceedings of the 11th IEEE International Software Metrics Symposium (METRICS 2005), S. 17–27, 2005.
- [71] Sharon McGee und Des Greer: *Software requirements change taxonomy: Evaluation by case study*, in Proceedings of the IEEE 19th International Requirements Engineering Conference (RE 2011), S. 25–34, 2011.
- [72] Bertrand Meyer: *Agile!*, Cham: Springer International Publishing, 2014.
- [73] Barry W. Boehm: *Software Engineering Economics*, IEEE Transactions on Software Engineering, Bd. SE-10, Nr. 1, S. 4–21, Januar 1984.

- [74] Steve McConnell: *Software Estimation: Demystifying the Black Art*, Redmond: Microsoft Press, 2006.
- [75] Mike Cohn: *Agile Design: Intentional Yet Emergent*, 2009. [Online]. Verfügbar unter: <https://www.mountangoatsoftware.com/blog/agile-design-intentional-yet-emergent>. [Zugegriffen: 27-Sep-2016].
- [76] David Greenwood und Ian Sommerville: *Expectations and Reality: Why an Enterprise Software System Did Not Work as Planned*, in *Information Systems Development: Reflections, Challenges and New Directions*, Rob Pooley, Jennifer Coady, Christoph Schneider, Henry Linger, Chris Barry, und Michael Lang (Hrsg.), New York: Springer, S. 51–62, 2013.
- [77] Volker Gruhn und Clemens Schäfer: *No-Frills Software Engineering for Business Information Systems Experience Report*, in *Proceedings of the 8th Conference on New Trends in Software Methodologies (SoMeT 2009)*, S. 93–105, 2009.
- [78] Robert E. Jones und Richard F. Deckro: *The social psychology of project management conflict*, *European Journal of Operational Research*, Bd. 64, Nr. 2, S. 216–228, Januar 1993.
- [79] Terry Williams: *The need for new paradigms for complex projects*, *International Journal of Project Management*, Bd. 17, Nr. 5, S. 269–273, Oktober 1999.
- [80] Weidong Xia und Gwanhoo Lee: *Grasping the complexity of IS development projects*, *Communications of the ACM*, Bd. 47, Nr. 5, S. 68–74, Mai 2004.
- [81] John Murray: *Reducing IT Project Complexity*, in *New directions in project management*, Paul C. Tinnirello (Hrsg.), Auerbach Publications, S. 435–446, 2001.
- [82] Karlheinz Kautz, Hans-Kristian Jørgensen, Casper Pedersen, Michael Sinnet, und Sameen Rab: *The Role of Improvisation and Politics in ISD Practice*, in *Information Systems Development: Challenges in Practice, Theory, and Education*, Wita Wojtkowski, Gregory Wojtkowski, Michael Lang, Kieran Conboy, und Chris Barry (Hrsg.), Boston: Springer, S. 431–446, 2009.
- [83] Jill Owen und Henry Linger: *Resolving Emergent Issues in Complex Projects: A Knowledge Management Approach*, in *Information Systems Development: Challenges in Practice, Theory, and Education*, Wita Wojtkowski, Gregory Wojtkowski, Michael Lang, Kieran Conboy, und Chris Barry (Hrsg.), Boston: Springer, S. 509–520, 2009.
- [84] John Leslie King und Carl P. Simon: *Complications with Complexity in Requirements*, *ACM Transactions on Management Information Systems*, Bd. 5, Nr. 3, S. 1–12, Januar 2015.
- [85] David Baccarini: *The concept of project complexity – a review*, *International Journal of Project Management*, Bd. 14, Nr. 4, S. 201–204, August 1996.

- [86] Kaitlynn M. Whitney und Charles B. Daniels: *The Root Cause of Failure in Complex IT Projects: Complexity Itself*, Procedia Computer Science, Bd. 20, Nr. 1, S. 325–330, 2013.
- [87] Mark Bergman, John Leslie King, und Kalle Lyytinen: *Large-Scale Requirements Analysis Revisited: The need for Understanding the Political Ecology of Requirements Engineering*, Requirements Engineering, Bd. 7, Nr. 3, S. 152–171, September 2002.
- [88] Simon Grapenthin, Matthias Book, Volker Gruhn, Christian Schneider, und Kai Völker: *Reducing Complexity Using an Interaction Room: An Experience Report*, in Proceedings of the 31th ACM International Conference on Design of Communication (SIGDOC 2013), S. 71–76, 2013.
- [89] Barry Boehm und Li Guo Huang: *Value-Based Software Engineering: A Case Study*, Computer, Bd. 36, Nr. 3, S. 33–41, März 2003.
- [90] Joachim Karlsson: *Software requirements prioritizing*, in Proceedings of the 2nd International Conference on Requirements Engineering (RE 1996), S. 110–116, 1996.
- [91] Jim Azar, Randy Smith, und David Cordes: *Value-Oriented Requirements Prioritization in a Small Development Organization*, IEEE Software, Bd. 24, Nr. 1, S. 32–37, Januar 2007.
- [92] Laura Lehtola, Marjo Kauppinen, und Sari Kujala: *Requirements Prioritization Challenges in Practice*, in Product Focused Software Process Improvement: 5th International Conference, PROFES 2004, Kansai Science City, Japan, April 5-8, 2004. Proceedings, Frank Bomarius und Hajimu Iida (Hrsg.), Berlin, Heidelberg: Springer, S. 497–508, 2004.
- [93] Aybüke Aurum und Claes Wohlin: *Engineering and Managing Software Requirements*, Berlin, Heidelberg: Springer-Verlag, 2005.
- [94] Aybuke Aurum und Claes Wohlin: *A value-based approach in requirements engineering: Explaining some of the fundamental concepts*, in Proceedings of the 13th international working conference on Requirements engineering: foundation for software quality (REFSQ 2007), S. 109–115, 2007.
- [95] John Favaro: *Managing requirements for business value*, IEEE Software, Bd. 19, Nr. 2, S. 15–17, 2002.
- [96] Robert Winter und Karl Landert: *IT/Business Alignment als Managementherausforderung*, Wirtschaftsinformatik, Bd. 48, Nr. 5, S. 309, Oktober 2006.
- [97] Karlheinz Kautz, Sabine Madsen, und Jacob Nørbjerg: *Persistent problems and practices in information systems development*, Information Systems Journal, Bd. 17, Nr. 3, S. 217–239, Juli 2007.
- [98] Barry Boehm und Richard Turner: *Balancing Agility and Discipline: A Guide for the Perplexed*, Boston: Addison Wesley, 2003.



- [99] Markus Hummel und Christoph Rosenkranz: *Is communication the key to success? Investigating the impact of agile practices on information systems development projects*, in Proceedings of the 7th International Research Workshop on Information Technology Project Management (IRWITPM 2012), S. 138–146, 2012.
- [100] James D. Herbsleb und Audris Mockus: *An empirical study of speed and communication in globally distributed software development*, IEEE Transactions on Software Engineering, Bd. 29, Nr. 6, S. 481–494, Juni 2003.
- [101] Annie I. Anton: *Successful software projects need requirements planning*, IEEE Software, Bd. 20, Nr. 3, S. 44–47, Mai 2003.
- [102] Don Wells: *Don't solve a problem before you get to it*, IEEE Software, Bd. 20, Nr. 3, S. 45–47, Mai 2003.
- [103] Kent Beck und Cynthia Andres: *Extreme Programming Explained*, 2. Aufl., Boston: Addison Wesley, 2005.
- [104] Martin Glinz und Samuel A. Fricker: *On shared understanding in software engineering: an essay*, Computer Science – Research and Development, Bd. 30, Nr. 3–4, S. 363–376, August 2015.
- [105] Ken Schwaber und Mike Beedle: *Agile Project Management with Scrum*, Redmond: Microsoft Press, 2004.
- [106] Mike Cohn: *Agile Estimating and Planning*, 12. Aufl., Upper Saddle River: Prentice Hall, 2013.
- [107] Boris Gloger: *Scrum: Produkte zuverlässig und schnell entwickeln*, München: Carl Hanser, 2009.
- [108] Balasubramaniam Ramesh, Lan Cao, und Richard Baskerville: *Agile requirements engineering practices and challenges: an empirical study*, IEEE Software, Bd. 25, Nr. 1, S. 449–480, Januar 2010.
- [109] Elizabeth Bjarnason, Krzysztof Wnuk, und Björn Regnell: *A case study on benefits and side-effects of agile practices in large-scale requirements engineering*, in Proceedings of the 1st Workshop on Agile Requirements Engineering (AREW 2011), S. 1–5, 2011.
- [110] Margaret Tan: *Establishing Mutual Understanding in Systems Design: An Empirical Study*, Journal of Management Information Systems, Bd. 10, Nr. 4, S. 159–182, März 1994.
- [111] Axel Hoffmann, Eva Alice Christiane Bittner, und Jan Marco Leimeister: *The Emergence of Mutual and Shared Understanding in the System Development Process*, in Requirements Engineering: Foundation for Software Quality: 19th International Working Conference, REFSQ 2013, Essen, Germany, April 8-11, 2013. Proceedings, Joerg Dorr und Andreas L. Opdahl (Hrsg.), Berlin, Heidelberg: Springer, S. 174–189, 2013.

- [112] Eva Alice Christiane Bittner und Jan Marco Leimeister: *Why shared understanding matters – Engineering a collaboration process for shared understanding to improve collaboration effectiveness in heterogeneous teams*, in Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS 2013), S. 106–114, 2013.
- [113] Simon Grapenthin, Steven Poggel, Matthias Book, und Volker Gruhn: *Facilitating Task Breakdown in Sprint Planning Meeting 2 with an Interaction Room: An Experience Report*, in Proceedings of the 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2014), S. 1–8, 2014.
- [114] Matthias Book, Simon Grapenthin, und Volker Gruhn: *Highlighting Value and Effort Drivers Early in Business and System Models*, in Intelligent Software Methodologies, Tools and Techniques: 13th International Conference, SoMeT 2014, Langkawi, Malaysia, September 22–24, 2014, Hamido Fujita und Ali Selamat (Hrsg.), Cham: Springer International Publishing, S. 211–222, 2014.
- [115] Michael Rosemann: *Potential pitfalls of process modeling: part B*, Business Process Management Journal, Bd. 12, Nr. 3, S. 377–384, Mai 2006.
- [116] Tobias Brückmann und Simon Grapenthin: *TextBook: Requirements Engineering*, CampusLab GmbH, Essen, 2014.
- [117] Volker Gruhn und Clemens Schäfer: *Systemic Approaches for Software Engineering*, in Proceedings of the 9th Conference on New Trends in Software Methodologies, Tools and Techniques (SoMeT 2010), S. 85–95, 2010.
- [118] Peter Darch, Annamaria Carusi, und Marina Jirotko: *Shared understanding of end-users’ requirements in e-Science projects*, in Proceedings of the IEEE 5th International Conference on E-Science Workshops (ESCIW 2009), S. 125–128, 2009.
- [119] Michiel Renger, Gwendolyn L. Kolfschoten, und Gert-Jan de Vreede: *Challenges in Collaborative Modeling: A Literature Review*, in Advances in Enterprise Engineering I: Lecture Notes in Business Information Processing, Nr.10, Jan L. G. Dietz, Antonia Albani, und Joseph Barjis (Hrsg.), Berlin, Heidelberg: Springer, S. 61–77, 2008.
- [120] Andrew Gemino und Yair Wand: *Complexity and clarity in conceptual modeling: Comparison of mandatory and optional properties*, Data & Knowledge Engineering, Bd. 55, Nr. 3, S. 301–326, Dezember 2005.
- [121] Thomas D. LaToza, Gina Venolia, und Robert DeLine: *Maintaining mental models*, in Proceeding of the 28th International Conference on Software Engineering (ICSE 2006), S. 492–501, 2006.
- [122] Mauro Cherubini, Gina Venolia, Rob DeLine, und Andrew J. Ko: *Let’s go to the whiteboard*, in Proceedings of the SIGCHI conference on Human factors in computing systems (CHI ’07), S. 557–566, 2007.
- [123] Uri Dekel und James D. Herbsleb: *Notation and representation in collaborative object-oriented design*, ACM SIGPLAN Notices, Bd. 42, Nr. 10, S. 261, Oktober 2007.

- [124] Jagoda Walny, Jonathan Haber, Marian Dork, Jonathan Sillito, und Sheelagh Cpendale: *Follow that sketch: Lifecycles of diagrams and sketches in software development*, in Proceedings of the 6th International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT 2011), S. 1–8, 2011.
- [125] Marco Mevius, Erich Ortner, und Peter Wiedmann: *Gebrauchssprachliche Modellierung als Grundlage für agiles Geschäftsprozessmanagement*, in GI-Edition – Lecture Notes in Informatics, Hans-Georg Fill, Dimitris Karagiannis, und Ulrich Reimer (Hrsg.), Bonn: Köllen, S. 169–184, 2014.
- [126] Sebastian Baltes und Stephan Diehl: *Sketches and diagrams in practice*, in Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2014), S. 530–541, 2014.
- [127] Jac A. M. Vennix: *Group Model Building: Facilitating Team Learning Using System Dynamics*, Chichester: John Wiley & Sons, 1996.
- [128] Mariëlle Den Hengst und Gert-Jan De Vreede: *Collaborative Business Engineering: A Decade of Lessons from the Field*, Journal of Management Information Systems, Bd. 20, Nr. 4, S. 85–114, 2004.
- [129] Odd I. Lindland, Guttorm Sindre, und Arne Solvberg: *Understanding quality in conceptual modeling*, IEEE Software, Bd. 11, Nr. 2, S. 42–49, März 1994.
- [130] Harry Howe Ransom: *Victims of Groupthink: A Psychological Study of Foreign-Policy Decisions and Fiascoes*, The Journal of Politics, Bd. 36, Nr. 1, S. 218–220, Februar 1974.
- [131] Paul Watzlawick, Janet H. Beavin, und Don D. Jackson: *Menschliche Kommunikation: Formen, Störungen, Paradoxien*, Bern: Hans Huber, 2000.
- [132] Douglas Dean, Richard Orwig, und Douglas Vogel: *Facilitation Methods for Collaborative Modeling Tools*, Group Decision and Negotiation, Bd. 9, Nr. 2, S. 109–128, 2000.
- [133] Markus Unterauer: *Workshops im Requirements Engineering: Methoden, Checklisten und Best Practices für die Ermittlung von Anforderungen*, Heidelberg: dpunkt, 2015.
- [134] Daniel M. Berry: *Formal methods: the very idea – Some thoughts about why they work when they work*, Science of Computer Programming, Bd. 42, Nr. 1, S. 11–27, Januar 2002.
- [135] Alan R. Dennis, Barbara H. Wixom, und Robert J. Vandenberg: *Understanding Fit and Appropriation Effects in Group Support Systems via Meta-Analysis*, MIS Quarterly, Bd. 25, Nr. 2, S. 167, Juni 2001.
- [136] David E. Avison, Francis Lau, Michael D. Myers, und Peter Axel Nielsen: *Action research*, Communications of the ACM, Bd. 42, Nr. 1, S. 94–97, Januar 1999.

- [137] Richard L. Baskerville und A. Trevor Wood-Harper: *A Critical Perspective on Action Research as a Method for Information Systems Research*, in *Enacting Research Methods in Information Systems*, Leslie P. Willcocks, Chris Sauer, und Mary C. Lacity (Hrsg.), Cham: Springer International Publishing, S. 169–190, 2016.
- [138] Peter Checkland und Sue Holwell: *Action Research: Its Nature and Validity*, *Systemic Practice and Action Research*, Bd. 11, Nr. 1, S. 9–21, 1998.
- [139] Jan A. Bergstra, Hans. B. M. Jonkers, und Henk J. Obbrink: *A software development model for method engineering*, in *ESPRIT' 84: Status report of ongoing work*, J. Roukens und J. Renuart (Hrsg.), Amsterdam: Elsevier Science Publishers, 1985.
- [140] Frank Harmsen, Sjaak Brinkkemper, und Han J. L. Oei: *Situational Method Engineering for Informational System Project Approaches*, in *Proceedings of the IFIP WG8.1 Working Conference on Methods and Associated Tools for the Information Systems Life Cycle*, S. 169–194, 1994.
- [141] Sjaak Brinkkemper: *Method engineering: engineering of information systems development methods and tools*, *Information and Software Technology*, Bd. 38, Nr. 4, S. 275–280, Januar 1996.
- [142] Kuldeep Kumar und Richard J. Welke: *Challenges and Strategies for Research in Systems Development*, in *Challenges and strategies for research in systems development*, William W. Cotterman und James A. Senn (Hrsg.), New York: John Wiley & Sons, S. 257–269, 1992.
- [143] Kees van Slooten und Sjaak Brinkkemper: *A Method Engineering Approach to Information Systems Development*, in *Proceedings of the IFIP WG8.1 Working Conference on Information System Development Process*, S. 167–186, 1993.
- [144] ISO/IEC TR 24774:2010: *Systems and software engineering – Life cycle management – Guidelines for process description*, Genf, 2010.
- [145] Brian Henderson-Sellers, Jolita Ralyté, Pär J. Ågerfalk, und Matti Rossi: *Situational Method Engineering*, Berlin, Heidelberg: Springer, 2014.
- [146] Nancy L. Russo, Judy L. Wynekoop, und Diane B. Walz: *The use and adaptation of system development methodologies*, in *Managing Information & Communications in a Changing Global Environment*, M. Khosrowpour (Hrsg.), Atlanta: Idea Group Publishing, 1995.
- [147] Brian Henderson-Sellers: *Method Engineering: Theory and Practice*, in *Proceedings of the 5th International Information Systems Technology and its Applications Conference (ISTA 2006)*, S. 13–23, 2006.
- [148] Richard J. Mayer, John W. Crump, Ronald Fernandes, Arthur Keen, und Michael K. Painter: *Information Integration for Concurrent Engineering (Iice) Compendium of Methods Report*, Wright-Patterson Air Force Base, 1995.

- [149] Jan Jürjens: *UMLsec: Extending UML for Secure Systems Development*, in Proceedings of the 5th International Conference on The Unified Modeling Language, S. 412–425, 2002.
- [150] Lars Mathiassen: *Collaborative practice research*, Information Technology & People, Bd. 15, Nr. 4, S. 321–345, Dezember 2002.
- [151] Matthias Book, Simon Grapenthin, und Volker Gruhn: *Risk-aware Migration of Legacy Data Structures*, in Proceedings of the 39th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2013), S. 53–56, 2013.
- [152] Matthias Book, Simon Grapenthin, und Volker Gruhn: *Value-Based Migration of Legacy Data Structures*, in Lecture Notes in Business Information Processing, Bd. 166, Dietmar Winkler, Stefan Biffel, und Johannes Bergsmann (Hrsg.), Cham: Springer International Publishing, S. 115–134, 2014.
- [153] Simon Grapenthin, Matthias Book, Thomas Richter, und Volker Gruhn: *Supporting Feature Estimation with Risk and Effort Annotations*, in Proceedings of the 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2016), S. 17–24, 2016.
- [154] Gerhard Schwarz: *Konfliktmanagement*, Wiesbaden: Gabler, 2014.
- [155] Douglas Schuler und Aki Namioka: *Participatory design: Principles and Practices*, Hillsdale: L. Erlbaum Associates, 1993.
- [156] Pelle Ehn und Morten Kyng: *Design at Work*, in Design at Work: Cooperative Design of Computer Systems, Joan Greenbaum und Morten Kyng (Hrsg.), Hillsdale: L. Erlbaum Associates, S. 306, 1992.
- [157] Åke Sandberg: *Computers dividing man and work: recent Scandinavian research on planning and computers from a Trade Union perspective*, Stockholm: Arbetslivscentrum, 1979.
- [158] Morten Kyng und Lars Mathiassen: *Systems Development and Trade Union Activities*, DAIMI Report Series, Bd. 8, Nr. 99, S. 6–12, Januar 1980.
- [159] Michael J. Muller und Sarah Kuhn: *Participatory design*, Communications of the ACM, Bd. 36, Nr. 6, S. 24–28, Juni 1993.
- [160] Finn Kensing, Jesper Simonsen, und Keld Bødker: *MUST: A Method for Participatory Design*, Human Computer Interaction, Bd. 13, Nr. 2, S. 167–198, Juni 1998.
- [161] N. Hallberg, Sofie Pilemalm, und Toomas Timpka: *Participatory design of inter-organizational systems: A method approach*, in Proceedings of the 5th Biennial Participatory Design Conference, S. 129–136, 1998.
- [162] Judith Gregory: *Scandinavian Approaches to Participatory Design*, International Journal of Engaging Education, Bd. 19, Nr. 1, S. 62–74, 2003.

- [163] Philip Kraft und Jørgen P. Bansler: *The collective resource approach: the Scandinavian experience*, Scandinavian Journal of Information Systems, Bd. 6, Nr. 1, S. 71–84, 1994.
- [164] Kaj Grønbaek, Jonathan Grudin, Susanne Bødker, und Liam J. Bannon: *Improving Conditions for Cooperative System Design: shifting from a product to a process focus*, DAIMI Report Series, Bd. 19, Nr. 331, September 1990.
- [165] Susanne Bødker, Ka Grønbaek, und Morten Kyng: *Cooperative Design: Techniques and Experiences From the Scandinavian Scene*, in Participatory design: Principles and practices, D. Schuler und A. Namioka (Hrsg.), Hillsdale: L. Erlbaum Associates, S. 157–175, 1993.
- [166] Kaj Grønbaek, Morten Kyng, und Preben Mogensen: *Toward a cooperative experimental system development approach*, in Computers and design in context, Morten Kyng und Lars Mathiassen (Hrsg.), Cambridge: MIT Press, S. 201–238, 1997.
- [167] Pelle Ehn: *Work-Oriented design of computer artifacts*, Hillsdale: L. Erlbaum Associates, 1990.
- [168] Jeanette Blomberg, Lucy Suchman, und Randall H. Trigg: *Reflections on a Work-Oriented Design Project*, Human Computer Interaction, Bd. 11, Nr. 3, S. 237–265, September 1996.
- [169] Karen Holtzblatt und Sandra Jones: *Contextual inquiry: A participatory technique for system design*, in Participatory Design: Principles and Practices, D. Schuler und A. Namioka (Hrsg.), Hillsdale: L. Erlbaum Associates, S. 177–210, 1993.
- [170] O.V.: *What ist PD?* [Online]. Verfügbar unter: <https://cs.stanford.edu/people/eroberts/cs181/projects/participatory-design/history.html>. [Zugegriffen: 28-Mai-2016].
- [171] Erran Carmel, Randall D. Whitaker, und Joey F. George: *PD and joint application design: a transatlantic comparison*, Communications of the ACM, Bd. 36, Nr. 6, S. 40–48, Juni 1993.
- [172] Elizabeth J. Davidson: *Joint application design (JAD) in practice*, Journal of Systems and Software, Bd. 45, Nr. 3, S. 215–223, 1999.
- [173] Jane Wood und Denise Silver: *Joint application design*, 2. Aufl., New York: John Wiley & Sons, 1995.
- [174] Bill Jennerich: *Joint Application Design – Business Requirements Analysis for Successful Re-Engineering*, 1990. [Online]. Verfügbar unter: <http://www.bee.net/bluebird/jad-doc.htm>. [Zugegriffen: 10-Feb-2017].
- [175] Peter Checkland: *Systems Thinking, Systems Practice*, New York: John Wiley & Sons, 1981.
- [176] Peter Checkland und Jim Scholes: *Soft Systems Methodology in Action*, New York:

- John Wiley & Sons, 1999.
- [177] Peter Checkland und John Poulter: *Soft Systems Methodology*, in *Systems Approaches to Managing Change: A Practical Guide*, London: Springer, S. 191–242, 2010.
- [178] Henry Mintzberg, Duru Raisinghani, und Andre Theoret: *The Structure of „Unstructured“ Decision Processes*, *Administrative Science Quarterly*, Bd. 21, Nr. 2, S. 246, Juni 1976.
- [179] Olle L. Bjerke: *Soft Systems Methodology in action: A case study at a purchasing department*, IT University of Göteborg, 2008.
- [180] Peter Checkland: *Soft Systems Methodology: A Thirty Year Retrospective*, *Systems Research and Behavioural Science*, Bd. 17, Nr. Supplement S1, S. 11–58, 2000.
- [181] Finn Kensing, Jesper Simonsen, und Keld Bødker: *Participatory IT Design – Designing for Business and Workplace Realities*, Cambridge: MIT Press, 2004.
- [182] Roger Martin und Jim Euchner: *Conversations: Design Thinking: An Interview with Roger Martin*, *Research-Technology Management*, Bd. 55, Nr. 3, S. 10–14, Mai 2012.
- [183] Stefanie Di Russo: *A Brief History of Design Thinking: How Design Thinking Came to Be*, 2012. [Online]. Verfügbar unter: <https://ithinkidesign.wordpress.com/2012/06/08/a-brief-history-of-design-thinking-how-design-thinking-came-to-be/>. [Zugegriffen: 22-Juli-2016].
- [184] O.V.: *Was ist Design Thinking*, 2016. [Online]. Verfügbar unter: <http://hpi-academy.de/design-thinking/was-ist-design-thinking.html>. [Zugegriffen: 22-Juli-2016].
- [185] Tim Brown: *Design thinking.*, *Harvard Business Review*, Nr. 6, S. 85–92, 2008.
- [186] Falk Uebernickel, Walter Brenner, Britta Pukall, Therese Naef, und Bernhard Schindholzer: *Design Thinking: Das Handbuch*, Frankfurt am Main: Frankfurter Allgemeine Buch, 2015.
- [187] Emmanuel Sauvonnet und Markus Blatt: *Wo ist das Problem? Design Thinking als neues Management-Paradigma*, 2. Aufl., Norderstedt: Books on Demand, 2015.
- [188] Hester Hilbrecht und Oliver Kempkens: *Design Thinking im Unternehmen – Herausforderung mit Mehrwert*, in *Digitalisierung und Innovation*, Frank Keuper, Hamidian Kiumars, Eric Verwaayen, Torsten Kalinowski, und Christian Kraijo (Hrsg.), Wiesbaden: Springer Fachmedien, S. 347–364, 2013.
- [189] Christoph Meinel und Julia von Thienen: *Design Thinking*, *Informatik-Spektrum*, Bd. 39, Nr. 4, S. 310–314, August 2016.
- [190] Chris Rupp und Die SOPHISTen: *Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil*, München: Carl Hanser, 2014.
- [191] Andreas Gebauer und Frederik Thormaehlen: *Einsatzerfahrungen mit Personas in der*

- Softwareentwicklung*, HMD Praxis der Wirtschaftsinformatik, Bd. 231, S. 71–78, 2003.
- [192] Jochen Gürtler und Johannes Meyer: *30 Minuten Design Thinking*, 5. Aufl., Offenbach: Gabal, 2013.
- [193] Capgemini Consulting Deutschland: *Accelerated Solutions Environment: Der innovative Weg zur Lösung komplexer Fragen*, 2013. [Online]. Verfügbar unter: [https://www.de.capgemini-consulting.com/resource-file-access/resource/pdf/ase-broschuere-de-web\\_0.pdf](https://www.de.capgemini-consulting.com/resource-file-access/resource/pdf/ase-broschuere-de-web_0.pdf). [Zugegriffen: 25-Feb-2017].
- [194] Isabelle Mirbel und Jolita Ralyté: *Situational method engineering: combining assembly-based and roadmap-driven approaches*, Requirements Engineering, Bd. 11, Nr. 1, S. 58–78, März 2006.
- [195] Josef W. Seifert: *Visualisieren Präsentieren Moderieren*, 37. Aufl., Offenbach: Gabal, 2011.
- [196] Etiënne A. J. A. Rouwette, Jac A. M. Vennix, und Cécile M. Thijssen: *Group Model Building: A Decision Room Approach*, Simulation & Gaming, Bd. 31, Nr. 3, S. 359–379, September 2000.
- [197] Jeff Kramer: *Is abstraction the key to computing?*, Communications of the ACM, Bd. 50, Nr. 4, S. 36–42, April 2007.
- [198] Gert-Jan de Vreede, Robert M. Davison, und Robert O. Briggs: *How a silver bullet may lose its shine*, Communications of the ACM, Bd. 46, Nr. 8, S. 96–101, 2003.
- [199] Helen Sharp, Anthony Finkelstein, und Galal Galal: *Stakeholder identification in the requirements engineering process*, in Proceedings of the 10th International Workshop on Database and Expert Systems Applications (DEXA 1999), S. 387–391, 1999.
- [200] Richard Woolridge, Denise McManus, und Joanne Hale: *Stakeholder Risk Assessment: An Outcome-Based Approach*, IEEE Software, Bd. 24, Nr. 2, S. 36–45, März 2007.
- [201] Helmut Balzert: *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*, 3. Aufl., Heidelberg: Spektrum, 2009.
- [202] Inge Hanschke: *Strategisches Management der IT-Landschaft*, 3. Aufl., München: Carl Hanser, 2013.
- [203] Ralf Wirdemann: *Scrum mit User Stories*, 2. Aufl., München: Carl Hanser Verlag, 2011.
- [204] Jan Mendling, Hajo A. Reijers, und Wil van der Aalst: *Seven process modeling guidelines (7PMG)*, Information and Software Technology, Bd. 52, Nr. 2, S. 127–136, Februar 2010.
- [205] Jakob Freund, Bernd Rücker, und Thomas Henninger: *Praxishandbuch BPMN*, München: Carl Hanser, 2010.
- [206] Jan Mendling, Hajo A. Reijers, und Jorge Cardoso: *What Makes Process Models*



- Understandable?*, in Business Process Management, Berlin, Heidelberg: Springer, S. 48–63, 2007.
- [207] Ann Langley: *Between 'Paralysis by analysis' and 'Extinction by instinct'*, Long Range Planning, Bd. 28, Nr. 4, S. 127, August 1995.
- [208] Daniel Kahneman: *Thinking, Fast and Slow*, London: Penguin, 2011.
- [209] Paolo Bocciarelli und Andrea D'Ambrogio: *A BPMN extension for modeling non functional properties of business processes*, in Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium (TMS-DEVS 2011), Bd. 57, Nr. 5, S. 160–168, 2011.
- [210] Kawther Saeedi, Liping Zhao, und Pedro R. Falcone Sampaio: *Extending BPMN for supporting customer-facing service quality requirements*, in Proceedings of the IEEE 8th International Conference on Web Services (ICWS 2010), S. 616–623, 2010.
- [211] Denis Gagne und André Trudel: *Time-BPMN*, in Proceedings of the IEEE Conference on Commerce and Enterprise Computing (CEC 2009), S. 361–367, 2009.
- [212] Alfonso Rodriguez, Eduardo Fernandez-Medina, und Mario Piattini: *Security requirement with a UML 2.0 profile*, in Proceedings of the 1st International Conference on Availability, Reliability and Security (ARES 2006), S. 670–677, 2006.
- [213] Alfonso Rodriguez, Eduardo Fernandez-Medina, und Mario Piattini: *A BPMN Extension for the Modeling of Security Requirements in Business Processes*, IEICE Transactions on Information and Systems, Bd. E90–D, Nr. 4, S. 745–752, März 2007.
- [214] Birgit Korherr und Beate List: *Extending the EPC and the BPMN with Business Process Goals and Performance Measures*, in Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS 2007), S. 287–294, 2007.
- [215] Meiko Jensen und Sven Feja: *A Security Modeling Approach for Web-Service-Based Business Processes*, in Proceedings of the 16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2009), S. 340–347, 2009.
- [216] Joe Zou und Christopher J. Pavlovski: *Modeling Architectural Non Functional Requirements: From Use Case to Control Case*, in Proceeding of the IEEE 3rd International Conference on e-Business Engineering (ICEBE 2006), S. 315–322, 2006.
- [217] Joe Zou und Christopher J. Pavlovski: *Control case approach to record and model non-functional requirements*, Information Systems and e-Business Management, Bd. 6, Nr. 1, S. 49–67, Januar 2008.
- [218] Christopher J. Pavlovski und Joe Zou: *Non-functional requirements in business process modeling*, in Proceedings of the 5th Asia-Pacific conference on Conceptual Modelling (APCCM 2008), S. 103–112, 2008.
- [219] The Object Management Group: *UML Profile for Modeling Quality of Service and*

- Fault Tolerance Characteristics and Mechanisms, v1.1*, Needham, 2008.
- [220] Ida Hogganvik und Ketil Stølen: *A Graphical Approach to Risk Identification, Motivated by Empirical Investigations*, in Proceedings of 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS 06), Oscar Nierstrasz, Jon Whittle, David Harel, und Gianna Reggio (Hrsg.), Berlin, Heidelberg: Springer, S. 574–588, 2006.
- [221] Johannes Bergsmann: *Requirements Engineering für die agile Softwareentwicklung: Methoden, Techniken und Strategien*, Heidelberg: dpunkt, 2014.
- [222] Brian Lawrence, Karl Wieggers, und Christof Ebert: *The top risk of requirements engineering*, IEEE Software, Bd. 18, Nr. 6, S. 62–63, 2001.
- [223] David J. Grimshaw und Godfrey W. Draper: *Non-functional requirements analysis: deficiencies in structured methods*, Information and Software Technology, Bd. 43, Nr. 11, S. 629–634, 2001.
- [224] Martin Glinz: *On Non-functional Requirements*, in Proceedings of the IEEE 15th International Requirements Engineering Conference (RE 2007), S. 21–26, 2007.
- [225] Christof Ebert: *Systematisches Requirements Engineering*, 5. Aufl., Heidelberg: dpunkt, 2014.
- [226] ISO/IEC 9126-1:2001: *Software engineering – Product quality – Part 1: Quality model*, Genf, 2001.
- [227] ISO/IEC 25010:2011: *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*, Genf, 2011.
- [228] Karl Wieggers: *First Things First: Prioritizing Requirements*, Software Development, Bd. 7, Nr. 9, S. 48–53, 1999.
- [229] Tharwon Arnuphaptrairong: *Top Ten Lists of Software Project Risks: Evidence from the Literature Survey*, Lecture Notes in Engineering and Computer Science, Bd. 2188, Nr. 1, S. 732–737, 2011.
- [230] Barry Boehm und Rony Ross: *Theory-W software project management principles and examples*, IEEE Transactions on Software Engineering, Bd. 15, Nr. 7, S. 902–916, Juli 1989.
- [231] Noriaki Kano, Nobuhiko Seraku, Fumio Takahashi, und Shin-ichi Tsuji: *Attractive Quality and Must-Be Quality*, Japanese Society for Quality Control, Bd. 14, Nr. 2, S. 147–156, 1984.
- [232] Sebastian Lehrig, Hendrik Eikerling, und Steffen Becker: *Scalability, Elasticity, and Efficiency in Cloud Computing*, in Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA 2015), S. 83–92, 2015.
- [233] Gwanhoo Lee und Weidong Xia: *The ability of information systems development*

- project teams to respond to business and technology changes: a study of flexibility measures*, European Journal of Information Systems, Bd. 14, Nr. 1, S. 75–92, März 2005.
- [234] Len Erlikh: *Leveraging legacy system dollars for e-business*, IT Professional, Bd. 2, Nr. 3, S. 17–23, 2000.
- [235] Alison Eastwood: *Firm fires shots at legacy systems*, Computing Canada, Bd. 19, Nr. 2, S. 17, 1993.
- [236] Tom Mens: *Introduction and Roadmap: History and Challenges of Software Evolution*, in Software Evolution, Berlin, Heidelberg: Springer, S. 1–11, 2008.
- [237] Sylvia Stuurman: *Design for change*, Dissertation in Computer Science an der Open Universiteit, Heerlen, 2015.
- [238] Klaus Pohl, Günter Böckle, und Frank van der Linden: *Software Product Line Engineering: Foundations, Principles and Techniques*, New York: Springer, 2005.
- [239] Jian Cao, Jie Wang, Kincho Law, Shensheng Zhang, und Minglu Li: *An interactive service customization model*, Information and Software Technology, Bd. 48, Nr. 4, S. 280–296, April 2006.
- [240] Christoph Beierle und Gabriele Kern-Isberner: *Methoden wissensbasierter Systeme*, Wiesbaden: Vieweg+Teubner, 2008.
- [241] Malte Unger, Henrik Leopold, und Jan Mendling: *How Much Flexibility is Good for Knowledge Intensive Business Processes: A Study of the Effects of Informal Work Practices*, in Proceedings of the 48th Hawaii International Conference on System Sciences (HICSS 2015), S. 4990–4999, 2015.
- [242] John Benamati und Albert L. Lederer: *Coping with rapid changes in IT*, Communications of the ACM, Bd. 44, Nr. 8, S. 83–88, August 2001.
- [243] George A. Miller: *The magical number seven, plus or minus two: some limits on our capacity for processing information.*, Psychological review, Bd. 101, Nr. 2, S. 343–52, April 1994.
- [244] Oliver König und Karl Schattenhofer: *Einführung in die Gruppendynamik*, 7. Aufl., Heidelberg: Carl Auer, 2014.
- [245] IEEE: *Std. 830-1998: Recommended Practice for Software Requirements Specifications*, 1998.
- [246] IEEE: *Std. 1233-1998: Guide for Developing System Requirements Specifications*, 1998.
- [247] ISO 29148:2011: *Systems and software engineering – Life cycle processes – Requirements engineering*, Genf, 2011.
- [248] Klaus Pohl: *Requirements Engineering – Grundlagen, Prinzipien, Techniken*, 2. Aufl.,

- Heidelberg: dpunkt, 2008.
- [249] Chriss Rupp, Stefan Queins, und Die Sophisten: *UML 2 glasklar*, 4. Aufl., München: Carl Hanser, 2012.
- [250] Günther Bourier: *Beschreibende Statistik: Praxisorientierte Einführung – Mit Aufgaben und Lösungen*, 12. Aufl., Wiesbaden: Springer, 2014.
- [251] Jacob Cohen: *A power primer*, Psychological Bulletin, Bd. 112, Nr. 1, S. 155–159, 1992.
- [252] Joachim Hartung: *Statistik: Lehr- und Handbuch der angewandten Statistik*, 15. Aufl., München: Oldenbourg, 2009.
- [253] Barry Boehm: *Software cost estimation with Cocomo II*, Upper Saddle River: Prentice Hall, 2000.
- [254] Lawrence H. Putnam und Ware Myers: *Measures for Excellence*, Upper Saddle River: Prentice Hall, 1991.
- [255] Magne Jørgensen, Barry Boehm, und Stan Rifkin: *Software Development Effort Estimation: Formal Models or Expert Judgment?*, IEEE Software, Bd. 26, Nr. 2, S. 14–19, März 2009.
- [256] James Grenning: *Planning poker or how to avoid analysis paralysis while release planning*, Hawthorn Woods, 2002.
- [257] Kjetil Moløkken-Østvold und Magne Jørgensen: *Group Processes in Software Effort Estimation*, Empirical Software Engineering, Bd. 9, Nr. 4, S. 315–334, Dezember 2004.
- [258] Magne Jørgensen und Martin Shepperd: *A Systematic Review of Software Development Cost Estimation Studies*, IEEE Transactions on Software Engineering, Bd. 33, Nr. 1, S. 33–53, 2007.
- [259] M. Jørgensen: *A review of studies on expert estimation of software development effort*, Bd. 70, Nr. 1–2, S. 37–60, 2004.
- [260] Martin Host und Claes Wohlin: *An experimental study of individual subjective effort estimations and combinations of the estimates*, in Proceedings of the 20th International Conference on Software Engineering (ICSE 1998), S. 332–339, 1998.
- [261] John Moses und John Clifford: *Learning how to improve effort estimation in small software development companies*, in Proceedings 24th Annual International Computer Software and Applications Conference (COMPSAC 2000), S. 522–527, 2000.
- [262] Magne Jørgensen: *An attempt to model software development effort estimation accuracy and bias*, in Proceedings of the 7th International Conference on Empirical Assessment in Software Engineering (EASE 2003). S. 117–128, 2003.
- [263] Robert T. Hughes: *Expert judgement as an estimating method*, Information and

- Software Technology, Bd. 38, Nr. 2, S. 67–75, 1996.
- [264] Kjetil Moløkken-Østvold und Nils C. Haugen: *Combining Estimates with Planning Poker – An Empirical Study*, in Proceeding of the Australian Software Engineering Conference (ASWEC 2007), S. 349–358, 2007.
- [265] Kjetil Moløkken-Østvold und Magne Jørgensen: *Expert estimation of web-development projects: Are software professionals in technical roles more optimistic than those in non-technical roles?*, Empirical Software Engineering, Bd. 10, Nr. 1, S. 7–29, 2005.
- [266] Viljan Mahnič und Tomaž Hovelja: *On using planning poker for estimating user stories*, Journal of Systems and Software, Bd. 85, Nr. 9, S. 2086–2095, September 2012.
- [267] Harold A. Linstone und Murray Turoff: *The Delphi Method: Techniques and Applications*, Technometrics, Bd. 18, Nr. 3, S. 363, August 1976.
- [268] Barry Boehm: *Software engineering economics*, Upper Saddle River: Prentice Hall, 1981.
- [269] N. C. Haugen: *An Empirical Study of Using Planning Poker for User Story Estimation*, in Proceedings of the conference on AGILE 2006 (AGILE 2006), S. 23–34, 2006.
- [270] Kjetil Moløkken-Østvold, Nils C. Haugen, und Hans C. Benestad: *Using planning poker for combining expert estimates in software projects*, Journal of Systems and Software, Bd. 81, Nr. 12, S. 2106–2117, Dezember 2008.
- [271] Roger Buehler, Deanna Messervey, und Dale Griffin: *Collaborative planning and prediction: Does group discussion affect optimistic biases in time estimation?*, Organizational Behavior and Human Decision Processes, Bd. 97, Nr. 1, S. 47–63, Mai 2005.
- [272] Markus Kleffmann, Matthias Book, und Volker Gruhn: *Supporting collaboration of heterogeneous teams in an augmented team room*, in Proceedings of the 6th International Workshop on Social Software Engineering (SSE 2014), S. 9–16, 2014.
- [273] Mike Cohn: *User Stories Applied: For Agile Software Development*, 18. Aufl., Boston: Addison Wesley, 2013.
- [274] Y. Miyazaki, A. Takanou, H. Nozaki, N. Nakagawa, und K. Okada: *Method to estimate parameter values in software prediction models*, Information and Software Technology, Bd. 33, Nr. 3, S. 239–243, April 1991.
- [275] Sam S. Shapiro und Martin B. Wilk: *An analysis of variance test for normality (complete samples)*, Biometrika, Bd. 52, Nr. 3–4, S. 591–611, Dezember 1965.
- [276] Duncan Cramer: *Fundamental statistics for social research: step-by-step calculations and computer techniques using SPSS for Windows*, New York: Routledge, 1998.
- [277] Henry Mann und Donald Whitney: *On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other*, The Annals of Mathematical Statistics, Bd. 18, Nr. 1, S. 50–60, März 1947.

- [278] Norman Cliff: *Ordinal methods for behavioral data analysis*, New York: Psychology Press, 1996.
- [279] Kent Beck, Mike Beedle, Arie van Bennekum, Alistar Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, und Dave Thomas: *Manifesto for Agile Software Development, The Agile Alliance*, 2001. [Online]. Verfügbar unter: <http://agilemanifesto.org/>. [Zugegriffen: 17-Mai-2017].
- [280] Kenneth Rubin: *Essential Scrum*, Boston: Addison Wesley, 2013.
- [281] Simon Grapenthin, Steven Poggel, Matthias Book, und Volker Gruhn: *Improving task breakdown comprehensiveness in agile projects with an Interaction Room*, Information and Software Technology, Bd. 67, Nr. 3, S. 254–264, November 2015.
- [282] Carey Schwaber, Gene Leganza, und David D’Silva: *Forrester Report: The truth about agile processes – Frank answers to frequently asked questions*, Cambridge, 2007.
- [283] Robert Koch: *The 80/20 Principle: The Secret of Achieving More with Less*, 3. Aufl., Vannes Cedex: Nicholas Brealey, 2017.
- [284] The Standish Group: *CHAOS report*, Boston, 1995.
- [285] Joachim Karlsson und Kevin Ryan: *A cost-value approach for prioritizing requirements*, IEEE Software, Bd. 14, Nr. 5, S. 67–74, 1997.
- [286] Michiel van Genuchten und Les Hatton: *Quantifying Software’s Impact*, Computer, Bd. 46, Nr. 10, S. 66–72, Oktober 2013.
- [287] Dieter Kempf: *Digitale Wirtschaftsspionage, Sabotage und Datendiebstahl*, Berlin, 2015.
- [288] Helmut Balzert: *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb*, 3. Aufl., Heidelberg: Spektrum, 2011.
- [289] Michael Stonebraker, Uğur Çetintemel, und Stan Zdonik: *The 8 requirements of real-time stream processing*, ACM SIGMOD Record, Bd. 34, Nr. 4, S. 42–47, Dezember 2005.
- [290] John D. Gould und Clayton Lewis: *Designing for usability: key principles and what designers think*, Communications of the ACM, Bd. 28, Nr. 3, S. 300–311, 1985.
- [291] Marc Hesenius, Tobias Griebe, und Volker Gruhn: *Towards a behavior-oriented specification and testing language for multimodal applications*, in Proceedings of the 2014 ACM SIGCHI symposium on Engineering interactive computing systems (EICS 2014), S. 117–122, 2014.
- [292] Philipp Gonzales-Scheller: *Trendthema Gamification: Was steckt hinter diesem Begriff?*, in Recrutainment, Wiesbaden: Springer Fachmedien, S. 33–51, 2013.

- 
- [293] Maximilian Hille und Julia Michel: *Mobile Web Experience in Deutschland*, Kassel, 2015.
- [294] Wil van der Aalst, Arthur ter Hofstede, und Matthias Weske: *Business Process Management: A Survey*, in *Business Process Management. Lecture Notes in Computer Science*, Nr. 2678, W. van der Aalst und M. Weske (Hrsg.), Berlin, Heidelberg: Springer, S. 1–12, 2003.
- [295] Wilhelm Hasselbring: *Information system integration*, *Communications of the ACM*, Bd. 43, Nr. 6, S. 32–38, Juni 2000.
- [296] Narayan Ramasubbu und Chris F. Kemerer: *Managing Technical Debt in Enterprise Software Packages*, *IEEE Transactions on Software Engineering*, Bd. 40, Nr. 8, S. 758–772, August 2014.
- [297] Eberhard Wolff und Sven Johann: *Technical Debt*, *IEEE Software*, Bd. 32, Nr. 4, S. 94–98, Juli 2015.
- [298] Philippe Kruchten, Robert L. Nord, und Ipek Ozkaya: *Technical Debt: From Metaphor to Theory and Practice*, *IEEE Software*, Bd. 29, Nr. 6, S. 18–21, 2012.

## Anhang

Im Anhang befinden sich Handreichungen für Interaction-Room-Coaches, die bei der Durchführung der Methode unterstützen können. Alle Elemente des Anhangs wurden bereits in den vorherigen Kapiteln eingeführt, kurz erläutert und referenziert.

Zunächst wird ein Überblick über alle Methodenbausteine, verfeinernde Prozess- und Produktelemente des Interaction Rooms gegeben (vgl. Anhang A). Daraufhin wird der so genannte Vorabfragebogen zur Vorbereitung von Interaction-Room-Workshops vorgestellt. Dieser stellt gleichzeitig ein zusammengesetztes Produktelement dar (vgl. Anhang B). In Anhang C werden alle Annotationstypen hergeleitet und vorgestellt, wie dies bereits für ausgewählte Annotationstypen in Kapitel 5.3.4 geschehen ist. Dabei wurden die Herleitung und Beschreibung der aus Kapitel 5.3.4 bekannten Annotationstypen der Vollständigkeit halber wiederholt. Im letzten Teil des Anhangs wird eine Vorlage zur Dokumentation von Annotationen abgebildet (vgl. Anhang D).

### A Überblick aller Methodenbausteine, Prozess- und Produktelemente

In diesem Kapitel wird ein Überblick über alle Methodenbausteine, Prozess- und Produktelemente gegeben, indem diese als Übersicht dargestellt und die jeweilige Abbildung im Anschluss kurz erläutert wird.

In einem Interaction-Room-Projekt wird der Methodenbaustein Interaction-Room-Workshop immer mindestens einmal verwendet. Ein Interaction-Room-Workshop besteht immer aus den Methodenbausteinen Vorbereitung, Landkarten, Annotationen und Erkenntnisdokumentation. Der Methodenbaustein Vorbereitung besteht aus den Methodenbausteinen Kontextanalyse und Methodenkonfiguration. In der Kontextanalyse wird zunächst der Kontext des Projekts mit Hilfe des ausgefüllten Vorabfragebogens (vgl. Anhang B) analysiert, und relevante Themen, das Projektziel, das Workshop-Ziel, der Projekttyp und die vorgeschlagenen Stakeholder werden gewonnen. Um diese Informationen zu konkretisieren, können Einzelgespräche geführt und Dokumente analysiert werden. Auf Basis dieser Information werden Landkarten ausgewählt und Annotationssets gebildet, es kann eine Anpassung der Annotationsbeschreibung erfolgen, die Epics für den thematischen Rahmen des Projekts werden aus den Themen abgeleitet und



dokumentiert und die konfigurierte Methodeninstanz wird mit dem Projektverantwortlichen abgestimmt (vgl. Abbildung 50).

Methodenbaustein			Prozesselement	Produktelement		Akteur			
Zusammengesetzt	Bestandteile	atomar		Input	Output	IR-MC	IR-DC	SH	PV
Interaction-Room-Workshop	Vorbereitung Landkarten Annotationen Erkenntnisdokumentation	-	-	-	-	-	-	-	-
Vorbereitung	Kontextanalyse Methodenkonfiguration	-	-	-	-	-	-	-	-
-	-	Kontextanalyse	Projektkontext identifizieren	ausgefüllter Vorabfragebogen	Themen, Projektziel, Workshop-Ziel, Projekttyp, Stakeholder-Auswahl	x	x	-	x
-	-	Kontextanalyse	Einzelgespräche führen	Themen, Projektziel, Workshop-Ziel, Projekttyp, Stakeholder-Auswahl	Themen, Projektziel, Workshop-Ziel, Projekttyp, Stakeholder-Auswahl	x	x	x	x
-	-	Kontextanalyse	Dokumente analysieren	Dokumente	Themen, Workshop-Ziel, Stakeholder-Auswahl	x	x	-	-
-	-	Methodenkonfiguration	Landkarten auswählen	Landkarten, Themen, Projektziel, Workshop-Ziel, Projekttyp	ausgewählte Landkarten	x	x	-	-
-	-	Methodenkonfiguration	Annotationssets bilden	Annotationstypen, Themen, Projektziel, Workshop-Ziel, Projekttyp	Annotationssets	x	x	-	-
-	-	Methodenkonfiguration	Annotationsbeschreibung anpassen	Annotationstypen, Themen, Projektziel, Workshop-Ziel, Projekttyp	angepasste Annotationsbeschreibung	x	x	-	-
-	-	Methodenkonfiguration	Epics identifizieren	Themen	Epics	x	x	-	-
-	-	Methodenkonfiguration	Methodeninstanz abstimmen	Epics, ausgewählte Landkarten, Annotationssets	abgestimmte Methodeninstanz	x	x	-	x

Abbildung 50: Methodenbausteine, Prozess- und Produktelemente des Interaction Rooms (1/3)

Methodenbaustein			Prozesselement	Produktelement		Akteur			
Zusammengesetzt	Bestandteile	atomar		Input	Output	IR-MC	IR-DC	SH	PV
Landkarten	Feature-Landkarte Prozesslandkarte Objektlandkarte Integrationslandkarte	-	-	-	-	-	-	-	-
-	-	Feature-Landkarte	Pragmatik der Feature-Landkarte	Epics	Epics, Features	x	x	x	x
-	-	Prozesslandkarte	Pragmatik der Prozesslandkarte	Feature, Syntax und Semantik der Prozesslandkarte	Prozesslandkarte, Kandidaten für Geschäftsobjekte, Systeme	x	x	x	x
-	-	Integrationslandkarte	Pragmatik der Integrationslandkarte	Systeme aus Prozesslandkarte, Syntax und Semantik der Integrationslandkarte	Integrationslandkarte, Geschäftsobjekte	x	x	x	x
-	-	Objektlandkarte	Pragmatik der Objektlandkarte	Kandidaten für Geschäftsobjekte aus Prozesslandkarten, Geschäftsobjekte aus Integrationslandkarte, Syntax und Semantik der Objektlandkarte	Objektlandkarte	x	x	x	x
Annotationen	Annotationsverwendung vorbereiten Annotationen verwenden	-	-	-	-	-	-	-	-
-	-	Annotationsverwendung vorbereiten	Annotationssets bilden	Annotationstypen, Projektziel, Workshop-Ziel, Projekttyp	Annotationsset	x	x	-	-
-	-	Annotationsverwendung vorbereiten	Annotationsbeschreibung anpassen	Annotationstypen, Projektziel, Workshop-Ziel, Projekttyp	Annotationsset	x	x	-	-
-	-	Annotationen verwenden	Landkarte annotieren	Annotationsset, Landkarte	annotierte Landkarte	x	x	x	x
-	-	Annotationen verwenden	Annotation diskutieren und dokumentieren	Annotation diskutierte Landkarte, Vorlage zur Dokumentation der Annotationen	Annotationsdokumentation	x	x	x	x

Abbildung 51: Methodenbausteine, Prozess- und Produktelemente des Interaction Rooms (2/3)

Der Methodenbaustein Landkarten setzt sich aus den vier Landkarten (vgl. Kapitel 5.2) zusammen (vgl. Abbildung 51). Jede Landkarte besitzt eine Syntax und Semantik, die mit Hilfe der beschriebenen Pragmatik angewendet wird. Der Methodenbaustein Annotationen besteht aus den Methodenbausteinen Annotationsverwendung vorbereiten und Annotationen verwenden. In der Vorbereitung werden die Annotationssets gebildet und die Annotationsbeschreibungen an den Projektkontext angepasst. Die Verwendung der Annotationen gliedert sich in das Annotieren einer Landkarte und die anschließende Diskussion und Dokumentation der Annotationstypen (vgl. Kapitel 5.3.5). Das Ergebnis ist die Dokumentation der Annotationen unter Zuhilfenahme der dafür vorgesehenen Vorlage (vgl. Anhang D).

Methodenbaustein		Prozesselement		Produktelement		Akteur			
Zusammengesetzt	Bestandteile	atomar		Input	Output	IR-MC	IR-DC	SH	PV
Erkenntnis-dokumentation	Landkartendigitalisierung Landkartenbeschreibung Dokumenterstellung Dokumentabstimmung	-	-	-	-	-	-	-	-
-	-	Landkartendigitalisierung	Landkarten in Software-Tool digitalisieren	Fotos der Landkarten	digitalisierte Landkarten	x	x	-	-
-	-	Landkartenbeschreibung	Landkarten in natürlicher Sprache beschreiben	Annotationsdokumentation, digitalisierte Landkarten	beschriebene Landkarten	x	x	-	-
-	-	Dokumenterstellung	Ergebnisdokument erstellen	Themen, Projektziel, Workshop-Ziel, beschriebene Landkarten, Annotations-dokumentation	Ergebnisdokument	x	x	-	-
-	-	Dokumentabstimmung	Ergebnisdokument mit Projektverantwortlichen abstimmen	Ergebnisdokument	abgestimmtes Ergebnisdokument	x	x	x	x

Abbildung 52: Methodenbausteine, Prozess- und Produktelemente des Interaction Rooms (3/3)

Der Methodenbaustein Erkenntnisdokumentation setzt sich aus der Landkartendigitalisierung, der Landkartenbeschreibung, der Dokumenterstellung und der Dokumentabstimmung zusammen (vgl. Abbildung 52). Zunächst werden die fotografierten Landkarten mit Hilfe eines Werkzeugs digitalisiert. Dabei müssen auch die Annotationen abgebildet werden. Im Anschluss werden die Landkarten beschrieben, dazu wird durch die Dokumentation der Annotationen verwendet. Sofern die Landkarten digitalisiert und beschrieben wurden, wird das Ergebnisdokument erstellt. Dabei werden nicht nur die Landkarten und deren Beschreibung verwendet, sondern auch die Ziele des Projekts und des Workshops. Sofern die notwendige Information vorliegt, werden auch Modelle erstellt, die nicht auf den Landkarten abgebildet wurden (vgl. Kapitel 5.4). Zum Schluss wird das Dokument mit den Projektverantwortlichen abgestimmt.

## B Fragebogen zur Vorbereitung eines Interaction-Room-Workshops

Der Fragebogen (vgl. Abbildung 53, Abbildung 54, Abbildung 55) wird eingesetzt, um Interaction-Room-Workshops vorzubereiten.

### Daten zum Projekt

Unternehmen: \_\_\_\_\_

Projektname: \_\_\_\_\_

Bitte beschreiben Sie das Projektziel:

Bitte wählen Sie den Projekttyp aus:

- |   |  |
|---|--|
| <input type="checkbox"/> Neuentwicklung | <input type="checkbox"/> Weiterentwicklung |
| <input type="checkbox"/> Migration      | <input type="checkbox"/> Analyse           |

Wie viele Stakeholder sind am Projekt beteiligt?

- |                                  |                                  |                                   |                                |                               |
|----------------------------------|----------------------------------|-----------------------------------|--------------------------------|-------------------------------|
| <input type="checkbox"/> 10 – 20 | <input type="checkbox"/> 20 – 50 | <input type="checkbox"/> 50 – 100 | <input type="checkbox"/> > 100 | <input type="checkbox"/> < 10 |
|----------------------------------|----------------------------------|-----------------------------------|--------------------------------|-------------------------------|

Wie würden sie die Hierarchie unter den Stakeholdern beschreiben?

- |                                |                                     |                                  |                                     |                                |
|--------------------------------|-------------------------------------|----------------------------------|-------------------------------------|--------------------------------|
| <input type="checkbox"/> flach | <input type="checkbox"/> eher flach | <input type="checkbox"/> neutral | <input type="checkbox"/> eher steil | <input type="checkbox"/> steil |
|--------------------------------|-------------------------------------|----------------------------------|-------------------------------------|--------------------------------|

Wie schätzen Sie die Domänenkenntnisse des Entwicklungsteams ein?

- |                                      |                                 |                                  |                               |                                    |
|--------------------------------------|---------------------------------|----------------------------------|-------------------------------|------------------------------------|
| <input type="checkbox"/> sehr gering | <input type="checkbox"/> gering | <input type="checkbox"/> neutral | <input type="checkbox"/> hoch | <input type="checkbox"/> sehr hoch |
|--------------------------------------|---------------------------------|----------------------------------|-------------------------------|------------------------------------|

Wie lange ist die geplante Dauer des Projekts?

- |                                   |                                     |   |                                      |                                    |
|-----------------------------------|-------------------------------------|---|--------------------------------------|------------------------------------|
| <input type="checkbox"/> < 3 Mon. | <input type="checkbox"/> 3 – 9 Mon. | <input type="checkbox"/> 9 Mon. – 2 Jahre | <input type="checkbox"/> 2 – 5 Jahre | <input type="checkbox"/> > 5 Jahre |
|-----------------------------------|-------------------------------------|---|--------------------------------------|------------------------------------|

Wie hoch ist das geplante Projektbudget?

- |                                  |  |   |  |                                    |
|----------------------------------|--|---|--|------------------------------------|
| <input type="checkbox"/> < 250 k | <input type="checkbox"/> 250k – 1000 k | <input type="checkbox"/> 1000 k – 10000 k | <input type="checkbox"/> 10000 k – 50000 k | <input type="checkbox"/> > 50000 k |
|----------------------------------|--|---|--|------------------------------------|

Wie gehen planen Sie in dem Projekt vorzugehen?

- |                                      |  |                                   |                                    |                               |
|--------------------------------------|--|-----------------------------------|------------------------------------|-------------------------------|
| <input type="checkbox"/> sequentiell | <input type="checkbox"/> eher iterativ | <input type="checkbox"/> iterativ | <input type="checkbox"/> eher agil | <input type="checkbox"/> agil |
|--------------------------------------|--|-----------------------------------|------------------------------------|-------------------------------|

Abbildung 53: Fragebogen zur Vorbereitung von Interaction-Room-Workshops (1/3)

## Workshop-Ziele

Bitte wählen Sie den Scope des Workshops:

Ist-Analyse

Soll-Definition

Bitte nennen Sie relevante Zielsetzungen:

Erhebung abstrakter Anforderungen

Analyse und Konzeption fachlicher Strukturen

Analyse und Konzeption von Abläufen

Einordnung des Systems in seinen Kontext

Bitte beschreiben Sie das Workshop-Ziel:

## Workshop-Inhalt

Bitte nennen Sie Themen, die im Workshop diskutiert werden sollen:

Bitte nennen Sie kritische Aspekte, die voraussichtlich im Workshop angesprochen werden:

Abbildung 54: Fragebogen zur Vorbereitung von Interaction-Room-Workshops (2/3)

## Stakeholder-Analyse

Namen des Stakeholders

Position des Stakeholders

Abteilung des Stakeholders

Bezug zum Projekt?

fachlich     organisatorisch     technisch     sonstiger Bezug: \_\_\_\_\_

Interesse am Projekt?

sehr gering     gering     hoch     sehr hoch

Einfluss auf das Projekt?

sehr gering     gering     hoch     sehr hoch

Begeisterungsfähigkeit für neue Methoden?

sehr gering     gering     hoch     sehr hoch

Weiteren Stakeholder hinzufügen

Welche Maßnahmen wurden den Stakeholdern empfohlen, um sich vorzubereiten?


Abbildung 55: Fragebogen zur Vorbereitung von Interaction-Room-Workshops (3/3)

## C Annotationstypen

### C.1 Annotationstyp Business Value

Software Engineering findet häufig in wertneutraler Umgebung statt, in der jede Anforderung, jeder Anwendungsfall, jeder Testfall, jeder Fehler und jedes Objekt als gleich wichtig erachtet wird [89]. Diese mangelnde Wertorientierung ist ein Problem, denn Anforderungen an Kerngeschäftsprozesse, wie beispielweise die Produktentwicklung, tragen anders zur Wertschöpfung bei als Unterstützungsprozesse, wie beispielsweise die Reisekostenabrechnung. Das Testen geschäftskritischer Funktionalität liefert einen anderen Beitrag zur Wertschöpfung als das Testen weniger kritischer Funktionalität. Der Anwendungsfall „Neuverträge abschließen“ liefert mehr Wert als der Anwendungsfall „Vertrag stornieren“, da eine Organisation mehr durch Neugeschäft verdient als durch eine Stornierung. Die wertneutrale Behandlung von Artefakten eines Softwareprozesses wird der Realität also nicht gerecht. Anstatt die Ressourcen auf das Erzielen maximaler Wertschöpfung auszurichten, werden Anforderungen durch technische Maßnahmen erfüllt, deren Kosten-Nutzen-Verhältnis nicht optimal ist [283], [284]. Außerdem existieren meist mehr Anforderungen, als im Rahmen der verfügbaren Zeit und mit verfügbaren Mitteln umsetzbar sind [285]. Daraus erwächst die Notwendigkeit, den Wertbeitrag von Anforderungen zu beurteilen. Mit dem Annotationstyp Business Value werden Landkartenelemente gekennzeichnet, die überdurchschnittlich zur Wertschöpfung einer Organisation beitragen (vgl. Tabelle 31).


Tabelle 31: Annotationstyp Business Value

	Annotationskategorie: Werttreiber
	Untertyp(en): -
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Business Value kennzeichnet Elemente, die zentrale Bedeutung für das Kerngeschäft besitzen. Der Annotationstyp zeigt, dass der Entwurf und die Implementierung in besonderem Maße zur Zielerreichung von IT- und Unternehmenszielen beitragen.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil hat besonderen Einfluss auf den Business Value?</li> <li>• Auf welche Unternehmensziele hat die Realisierung dieses Teils positiven bzw. negativen Einfluss?</li> <li>• Was ist erforderlich, um den positiven Einfluss zu erreichen bzw. den negativen Einfluss zu vermeiden?</li> </ul>	

## C.2 Annotationstyp User Value

Funktionalität zu entwickeln, die nicht die Bedürfnisse der Nutzer erfüllt, ist ein Projektrisiko [222], [225], [228], [229], denn nur wenn Nutzer einen Wertzuwachs durch Funktionalität und Qualität eines Informationssystems erkennen, akzeptieren sie dieses [230]. Der Wert für einen Nutzer muss nicht mit dem Wert für eine Organisation korrelieren, jedoch sollte ein bestmöglicher Kompromiss gefunden werden, um alle Stakeholder eines Systems zufriedenzustellen [230]. Der Wert für Nutzer wird in Basisfaktoren, Leistungsfaktoren und Begeisterungsfaktoren klassifiziert [231]. Basisfaktoren sind Anforderungen, die zu Inakzeptanz eines Systems führen, wenn diese fehlen. Die Umsetzung von Basisfaktoren führt jedoch nicht unmittelbar zu Akzeptanz [221], sondern wird als selbstverständlich vorausgesetzt. Leistungsfaktoren werden von Nutzern explizit gefordert, deren Umsetzung führt zu Zufriedenheit mit dem System und stiftet großen Nutzen [ebenda]. Begeisterungsfaktoren sind dem Nutzer nicht bekannt, stellen sich aber als angenehme und nützliche Überraschung heraus und führen zu Begeisterung [ebenda]. Diejenigen Prozess- und Systemteile, die überdurchschnittlich zur Zufriedenheit der späteren Nutzer eines Systems beitragen und somit dessen Akzeptanz fördern, werden mit dem Annotationstyp User Value gekennzeichnet (vgl. Tabelle 32).


Tabelle 32: Annotationstyp User Value

	Annotationskategorie: Werttreiber
	Untertyp(en): Basisfaktor, Leistungsfaktor, Begeisterungsfaktor
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp User Value wird verwendet, um Landkartenelemente zu kennzeichnen, die einen Mehrwert für Anwender liefern. Der Annotationstyp zeigt, dass der Entwurf und die Implementierung dieser Prozess- und Systemteile in besonderem Maße zur Zufriedenheit der Nutzer und deren Akzeptanz des Systems beitragen.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil hat besonderen Einfluss auf den User Value?</li> <li>• Was ist die Erwartungshaltung an diesen Teil? (Basis-/Leistungs-/Begeisterungsfaktor)</li> <li>• Was ist erforderlich, um einen positiven Einfluss zu erreichen bzw. negativen Einfluss zu vermeiden?</li> </ul>	

### C.3 Annotationstyp Innovation

Software ist ein wesentlicher Innovationstreiber und gewinnt in unserer Gesellschaft zunehmend an Bedeutung [286]. Steigender Wettbewerbsdruck etablierter Unternehmen führt zu immer kürzer werdenden Innovationszyklen. Durch das Verschwimmen von Branchengrenzen entsteht, neben steigendem Wettbewerbsdruck, auch die Chance, neue Märkte zu erschließen. Allerdings ist Innovation immer mit Risiko verbunden: Je nach Innovationshöhe ist die Machbarkeit unklar, Rahmenbedingungen wie Gesetze können eine Innovation be- bzw. verhindern und die Akzeptanz von Kunden ist ungewiss [14]. Für jede Innovation ist also zu beurteilen, ob die Chancen die Risiken rechtfertigen. Um eine Innovation umzusetzen, müssen Fähigkeiten vorhanden, Ressourcen sichergestellt und die Monetarisierung der Innovation entworfen und tragfähig sein. Mit dem Annotationstyp Innovation können Innovationspotentiale gekennzeichnet werden, um im Anschluss die Innovation zu beurteilen und werthaltige von weniger werthaltiger Innovation zu trennen (vgl. Tabelle 33).

Tabelle 33: Annotationstyp Innovation


	Annotationskategorie: Werttreiber
	Untertyp(en): inkrementell (fachlich/technisch), disruptiv (fachlich/technisch)
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Innovation wird eingesetzt, um Landkartenelemente zu kennzeichnen, die ein Innovationspotential besitzen. Der Annotationstyp zeigt, dass beim Entwurf und der Implementierung besonders zu beachten ist, ob die Innovation machbar und wie sie monetarisierbar ist, um im Anschluss evaluieren zu können, ob das Risiko und der Aufwand für die Umsetzung der Innovation in Kauf genommen werden soll. Eine Innovation kann entweder fachlich und/oder technisch inkrementell oder fachlich und/oder technisch disruptiv sein.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil stellt eine besondere Innovation dar?</li> <li>• Was macht die Innovation in diesem Teil aus?</li> <li>• Was ist erforderlich, um einen positiven Einfluss zu erreichen bzw. negativen Einfluss zu vermeiden?</li> </ul>	



## C.4 Annotationstyp Sicherheit

Der Branchenverband der deutschen Informations- und Telekommunikationsindustrie, Bitkom e. V., hat 2015 eine Studie zu digitaler Wirtschaftsspionage, Sabotage und Datendiebstahl veröffentlicht [287]. Auf die Frage, wie häufig Unternehmen Angriffen auf die IT ausgesetzt sind, antworteten 45 % der befragten Unternehmen, dass dies mindestens monatlich der Fall sei. Je höher die Anzahl von Mitarbeitern ist, desto häufiger werden Unternehmen angegriffen. Das hängt auch damit zusammen, dass sich Mitarbeiter unter den Angreifern befinden. Die Höhe des Schadens durch Cyberkriminalität beläuft sich jährlich auf geschätzte 102,4 Milliarden Euro, daher ist es nicht verwunderlich, dass jedes Unternehmen in die IT-Sicherheit investiert. Sicherheit ist ein Qualitätsmerkmal nach ISO 25010 und bezeichnet die Fähigkeit, Daten vor unautorisiertem Zugriff zu schützen, deren Echtheit und Nachweisbarkeit zu gewährleisten sowie die Nachverfolgbarkeit von Handlungen einzelner Anwender eindeutig zuordnen zu können [221]. Während alle Prozess- und Systemteile das organisationsübliche Sicherheitsmaß erfüllen müssen, wird der Annotationstyp Sicherheit verwendet, um Prozess- und Systemteile zu kennzeichnen, deren Sicherheitsanforderungen über das organisationsübliche Maß hinausgehen (vgl. Tabelle 34).


Tabelle 34: Annotationstyp Sicherheit

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Datenschutz, Datensicherheit
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Sicherheit wird verwendet, um Landkartenelemente zu kennzeichnen, die in besonderem Maße schützenswert sind. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung Maßnahmen zur Sicherstellung des Datenschutzes und der Datensicherheit zu realisieren sind, die über das übliche Maß hinausgehen.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Was soll gegen Zugriff/Verlust geschützt werden?</li> <li>• Welcher Art ist die Bedrohung? Wie stark soll der Schutz sein?</li> <li>• Welche Auswirkungen hätte ein fehlender Schutz?</li> </ul>	

## C.5 Annotationstyp Hohe Last

Das Verbrauchsverhalten ist ein Teilmerkmal der Effizienz nach ISO 25010 und bezeichnet die „Fähigkeit des Softwareprodukts, eine angemessene Anzahl und angemessene Typen von Ressourcen zu verwenden, wenn die Software ihre Funktionen unter festgelegten Bedingungen ausführt“ [201]. Lastanforderungen können mit ausreichenden Ressourcen oder skalierbaren Infrastrukturen erfüllt werden. Sofern Lastanforderungen Schwankungen unterliegen, ermöglichen elastische Infrastrukturen lastgerechtes, automatisches Provisionieren und Deprovisionieren von Ressourcen [22]. Die Konsequenz aus der Identifikation funktionaler Anforderungen, die besondere dauerhafte oder temporäre Lastanforderungen stellen, muss eine entsprechende Softwarearchitektur und Struktur des Quellcodes sein. Sofern den Lastanforderungen durch die Auslagerung von Systemteilen in die Cloud genügt werden soll, muss die Softwarearchitektur durch geeigneten Komponentenschnitt auf unterschiedlichen Infrastrukturkomponenten lauffähig sein [232]. Mit dem Annotationstyp Hohe Last werden Prozess- und Systemteile gekennzeichnet, die besonderen Lastanforderungen genügen müssen (vgl. Tabelle 35).


Tabelle 35: Annotationstyp Hohe Last

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): dauerhaft, temporär
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p><b>Erläuterung</b></p> <p>Der Annotationstyp Hohe Last wird verwendet, um Landkartenelemente zu kennzeichnen, die Lastanforderungen genügen müssen. Der Annotationstyp zeigt, dass bei der Konzeption und Implementierung entsprechende fachliche oder technische Maßnahmen vorgesehen werden müssen, um dauerhaften oder temporären Lastanforderungen gerecht werden zu können.</p>	
<p><b>Anschlussfragen</b></p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil ist voraussichtlich hoher Last ausgesetzt?</li> <li>• Welcher Art ist die Last? Wann, wie oft und wie lange wird sie erwartet?</li> <li>• Welche Auswirkungen hätte eine Überlastung?</li> </ul>	

## C.6 Annotationstyp Zeitbeschränkung

Das Zeitverhalten ist ein Teilmerkmal der Effizienz nach ISO 25010 und bezeichnet die „Fähigkeit des Softwareprodukts, angemessene Antwort- und Verarbeitungszeiten sowie Durchsatz bei der Funktionsausführung unter festgelegten Bedingungen sicherzustellen“ [288]. Während alle Funktionen einer Software stets angemessenes Zeitverhalten aufweisen sollten, stellen bestimmte Funktionen besondere Anforderungen an die Einhaltung bestimmter Fristen oder Zeitfenster, die entweder technisch oder fachlich motiviert sind. Das Antwortzeitverhalten bestimmter Funktionen, die fristgerechte Bearbeitung fachlicher Prozesse und die verteilte Berechnung komplexer Algorithmen erfordert die fristgerechte Verfügbarkeit von Daten. Sofern eine geringe Latenzzeit erforderlich ist, bedeutet dies besondere Anforderungen an die Verarbeitung von Datenströmen (z. B. Verarbeitung nach Speicherung oder regelbasierte Verarbeitung) und stellt bestimmte Anforderungen an die betroffenen Systemteile (z. B. Robustheit ggü. inkorrekten Datenströmen) [289]. Das Einhalten von Fristen und Zeitfenstern erfordert eine entsprechende Konzeption und Implementierung einer Softwarearchitektur, sodass diese Anforderungen zu Beginn eines Projektes identifiziert werden müssen. Für die Kennzeichnung solcher Zeitanforderungen existiert der Annotationstyp Zeitbeschränkung (vgl. Tabelle 36).


Tabelle 36: Annotationstyp Zeitbeschränkung

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Frist, Zeitfenster
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Zeitbeschränkung wird verwendet, um Landkartenelemente zu kennzeichnen, die überdurchschnittlichen Zeitanforderungen genügen müssen. Der Annotationstyp zeigt, dass beim Entwurf und bei der Implementierung fachliche, organisatorische und technische Maßnahmen vorgesehen werden müssen, um die Wahrung dieser Fristen und Zeitfenster gewährleisten zu können.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welcher Zeitrahmen ist einzuhalten?</li> <li>• Was soll binnen des Zeitrahmens geschehen?</li> <li>• Welche Auswirkungen hätte eine Überschreitung des Zeitrahmens?</li> </ul>	

## C.7 Annotationstyp Korrektheit

Die Korrektheit ist nach ISO 25010 in die funktionale Richtigkeit und die funktionale Angemessenheit unterteilt [227]. Die Korrektheit ist Grundvoraussetzung für alle Bestandteile eines Informationssystems, denn alle Funktionen, Module und Komponenten müssen entsprechend ihrer Spezifikation funktionieren. Die funktionale Angemessenheit muss entsprechend des Zwecks und der Anforderungen an Prozess- und Systemteile bewertet werden. Einige Prozess- und Systemteile erfordern eine hohe Aktualität der Daten, um Berechnungen auf aktuellen Daten vorzunehmen. Andere erfordern besondere Maßnahmen, um die Konsistenz des Datenbestands, bspw. durch geeignete Replikations- und Synchronisationsmechanismen, sicherstellen zu können. Wieder andere Prozess- und Systemteile benötigen hohe Präzision, weil kleine Abweichungen (z. B. durch Rundungsfehler) zu falschen Ergebnissen führen können (z. B. bei dynamischer Anpassung von Beiträgen kapitalaufbauender Produkte). Werden auf inkorrekten Informationen Entscheidungen getroffen, kann dies langfristig den Erfolg eines Unternehmens gefährden. Die Umsetzung von Korrektheitsanforderungen bedeutet Mehraufwand in der Konzeption und deren Implementierung, daher muss zwischen Anforderungen, die hohe Präzision, Aktualität und Konsistenz erfordern, und Anforderungen, die übliche Präzision, Aktualität und Konsistenz erfordern, unterschieden werden. Zur Kennzeichnung funktionaler Anforderungen mit überdurchschnittlichen Korrektheitsanforderungen wird der Annotationstyp Korrektheit verwendet (vgl. Tabelle 37).


Tabelle 37: Annotationstyp Korrektheit

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Präzision, Aktualität, Konsistenz
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Korrektheit wird verwendet, um Landkartenelemente zu kennzeichnen, die besondere Anforderungen an die Aktualität, Präzision oder Konsistenz von Daten stellen. Der Annotationstyp zeigt, dass bei der Konzeption und Implementierung geeignete Mechanismen vorgesehen werden müssen, um Korrektheitsanforderungen erfüllen zu können, die über das übliche Maß an Korrektheit hinausgehen.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Welche Prozesse/Daten sollen möglichst präzise/aktuell/konsistent sein?</li> <li>• Welcher Grad der Präzision/Aktualität/Konsistenz ist erforderlich?</li> <li>• Welcher Nutzen wird von der Präzision/Aktualität/Konsistenz erwartet?</li> </ul>	

## C.8 Annotationstyp Zuverlässigkeit

Die Zuverlässigkeit ist ein Qualitätsmerkmal nach ISO 25010, welches auf den ersten Blick selbstverständlich ist und der alle Systemteile genügen müssen. Jedoch stellen bestimmte System- oder Prozessteile höhere Anforderungen an die Zuverlässigkeit als andere. Kernkomponenten, die maßgeblich zur Funktionsfähigkeit eines Systems und dadurch zur Aufrechterhaltung der Geschäftstätigkeit einer Organisation beitragen, müssen besondere Anforderungen an deren Robustheit und Verfügbarkeit erfüllen. Die Güte der Robustheit und Verfügbarkeit muss evaluiert und vertraglich vereinbart werden (z. B. als Service Level). Um die vereinbarte Güte sicherzustellen, müssen auf der einen Seite entsprechende organisatorische Maßnahmen definiert und operationalisiert werden (z. B. Incident-Management-Prozesse) und auf der anderen Seite müssen technische Aspekte in der Softwarearchitektur vorgesehen und implementiert werden (z. B. die Portierbarkeit bestimmter Komponenten und Module). Überdurchschnittlich zuverlässige Prozess- und Systemteile zu implementieren, verursacht zusätzlichen Aufwand, was eine sorgfältige Identifikation dieser Prozess- und Systemteile notwendig macht. Zur Kennzeichnung dieser potentiellen Prozess- und Systemteile existiert der Annotationstyp Zuverlässigkeit (vgl. Tabelle 38).


Tabelle 38: Annotationstyp Zuverlässigkeit

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Robustheit, Verfügbarkeit
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Zuverlässigkeit wird verwendet, um Landkartenelemente zu kennzeichnen, die überdurchschnittlich hohe Anforderungen an die Robustheit oder Verfügbarkeit stellen. Der Annotationstyp zeigt, dass bei Entwurf und der Implementierung fachliche, organisatorische oder technische Maßnahmen realisiert werden müssen.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welche Prozesse/Daten sollen möglichst zuverlässig sein?</li> <li>• Welcher Grad der Verfügbarkeit ist erforderlich?</li> <li>• Welcher Nutzen wird von der hohen Zuverlässigkeit erwartet?</li> </ul>	

## C.9 Annotationstyp Benutzbarkeit

Bereits 1983 haben Gould und Lewis die Notwendigkeit der Berücksichtigung individueller Nutzer identifiziert und daraus abgeleitet, dass der Entwurf interaktiver Systeme und die Überprüfung der Güte von Entwürfen in engem Austausch mit echten Nutzern und vor Beginn des Systemdesigns stattfinden muss [290]. Heute ist die Benutzbarkeit als ein Qualitätsmerkmal im ISO-Standard 25010 verankert und vereint dort u. a. die Verständlichkeit, Bedienbarkeit, Attraktivität und Erlernbarkeit. Das rasche und intuitive Verstehen von Inhalten wird schwieriger, je mehr Informationen dargestellt werden, und wichtiger, je häufiger eine Oberfläche benutzt wird. Die Bedienbarkeit muss für bestimmte Prozess- und Systemteil besonders intuitiv sein, insbesondere wenn multimodale Interaktionsmodi realisiert werden [291]. Die Attraktivität von Prozess- und Systemteilen kann durch die motivationspsychologischen Grundbedürfnisse (Selbstbestimmung, Perfektionierung, sozialer Bezug) [292] als strategisches Mittel eingesetzt werden, um Nutzer zur Verwendung bestimmter Funktionen zu motivieren. Prozess- und Systemteile, die oft von unerfahrenen Nutzern bedient werden, müssen sich durch gute Erlernbarkeit auszeichnen. Aus der Umsetzung überdurchschnittlicher Benutzbarkeit resultiert konzeptioneller Mehraufwand, der nicht für alle Prozess- und Systemteile wirtschaftlich ist. Deshalb werden mit dem Annotationstyp Benutzbarkeit die Prozess- und Systemteile gekennzeichnet, die überdurchschnittlichen Anforderungen der Verständlichkeit, Bedienbarkeit, Attraktivität und Erlernbarkeit genügen müssen (vgl. Tabelle 39).


Tabelle 39: Annotationstyp Benutzbarkeit

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Verständlichkeit, Bedienbarkeit, Attraktivität, Erlernbarkeit
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input type="checkbox"/> Objektlandkarte <input type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Benutzbarkeit wird verwendet, um Landkartenelemente zu kennzeichnen, deren Verständlichkeit, Bedienbarkeit, Attraktivität oder Erlernbarkeit überdurchschnittlich sein soll. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung darauf zu achten ist, dass diese Elemente überdurchschnittlich effektiv die Lösung einer Aufgabe unterstützen müssen.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Was soll möglichst verständlich dargestellt, bedienbar, attraktiv oder erlernbar sein?</li> <li>• Was macht die Benutzbarkeit hier zu einer besonderen Herausforderung?</li> <li>• Welcher Nutzen wird von der Benutzbarkeit erwartet?</li> </ul>	

## C.10 Annotationstyp Flexibilität

Durch technologische Innovation, fachliche Weiterentwicklung und Markterfordernisse steigt der Bedarf, Informationssysteme flexibel zu gestalten [233]. Rasche und zuverlässige Wartung und Weiterentwicklung sind kostenintensiv [234], [235] und eine der größten Herausforderungen im Software Engineering [236]. Informationssysteme können nach dem Prinzip „Design for Chance“ entwickelt werden, um deren Wartung und Weiterentwicklung schon bei der Konzeption einzuplanen sowie die Struktur und den Schnitt der Anwendung auf zukünftige Änderbarkeit auszulegen [237]. Flexibilitätsanforderungen umzusetzen, verursacht Aufwand, weswegen die Anwendungsteile identifiziert werden müssen, deren Umsetzung nach dem Prinzip „Design for Change“ ein positives Kosten-Nutzen-Verhältnis besitzen. Neben Flexibilitätsanforderungen, die aus der Evolution von Software erwachsen, kann bereits zum Entwicklungszeitpunkt zukünftige Änderungsnotwendigkeit bekannt sein, jedoch ohne dass die konkrete Ausgestaltung der Änderung bekannt ist (z. B. durch die Anpassung von Gesetzen). Außerdem kann der zielgruppengerechte Funktionsumfang einer Software das Bilden von Varianten erfordern [238]. Die Konfiguration von Software bietet Nutzern einen gewissen Individualisierungsgrad und kann beispielsweise durch die Orchestrierung von Services (vgl. z. B. [239]) oder regel- bzw. fallbasierte Systeme [240] umgesetzt werden. Eine frühe Identifikation von flexiblen Prozess- und Systemteilen ist notwendig, um in der Softwarearchitektur dafür sowohl gemeinsam genutzte als auch variantenspezifische Komponenten vorsehen zu können. Für die Kennzeichnung flexibler Bestandteile existiert der Annotationstyp Flexibilität (vgl. Tabelle 40)


Tabelle 40: Annotationstyp Flexibilität

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Zukünftige Änderung, Anpassung zur Laufzeit, Variantenbildung
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Flexibilität wird verwendet, um Landkartenelemente zu kennzeichnen, bei denen bestimmte Flexibilitätsanforderungen vor der Konzeption, bei der Konzeption oder während der Laufzeit berücksichtigt werden müssen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung die Prinzipien Design for Change, Design for Customization oder Prinzipien zur Bildung von Varianten berücksichtigt werden müssen.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil soll flexibel ausgelegt sein?</li> <li>• Welche Konfigurationen/Varianten/Systemanpassungen sind erforderlich?</li> <li>• Welche Auswirkungen hätte Inflexibilität an dieser Stelle?</li> </ul>	

## C.11 Annotationstyp Mobilität

Die mobile Aufgabenbearbeitung ist bereits private Erlebniswirklichkeit: Zugtickets werden auf dem Weg zum Bahnhof erworben, Termine im Stau vereinbart und der nächste Urlaub während des Abendessens gebucht. Unternehmen sehen die mobile Arbeit von Mitarbeitern jedoch noch als Herausforderung an [293]. Die allgegenwärtige mobile Verfügbarkeit des Internets ermöglicht die ortsungebundene Bearbeitung betrieblicher Aufgaben, stellt neue Anforderungen an die informationstechnologische Unterstützung von Prozessen sowie die Gestaltung von Daten und Schnittstellen – der Umgang mit Verbindungsabbrüchen, geringere Rechenleistung und Speicherplatz mobiler Geräte, veränderte Sicherheitsanforderungen und kontextsensitive Funktionen sind nur einige Beispiele. Um die Prozess- und Systemteile zu kennzeichnen, die mobil oder offline-verfügbar bzw. ortsabhängiges Verhalten aufweisen sollen, existiert der Annotationstyp Mobilität (vgl. Tabelle 41).

Tabelle 41: Annotationstyp Mobilität


	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Mobile Verfügbarkeit, Offline-Verfügbarkeit, Ortsabhängigkeit
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Mobilität wird verwendet, um Landkartenelemente zu kennzeichnen, die ortsungebunden verwendbar sein sollen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung auf den Umgang mit Verbindungsabbrüchen geachtet werden, ortsbezogene Daten und Dienste berücksichtigt oder fachliche und technische Anforderungen an Daten und Prozesse erfüllt werden müssen, um die mobile Verfügbarkeit gewährleisten zu können.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil soll mobil/offline/ortsabhängig verfügbar sein?</li> <li>• Unter welchen Umständen ist mit Funktionseinschränkungen zu rechnen?</li> <li>• Welcher Nutzen wird von der Mobilität/Offline-Verfügbarkeit/Ortsabhängigkeit erwartet?</li> </ul>	



## C.12 Annotationstyp Manuelle Bearbeitung

Die erfolgreiche Ausführung bestimmter Aktivitäten in wissensintensiven Geschäftsprozessen ist abhängig von den kognitiven Fähigkeiten ihrer Bearbeiter, weil sie sich durch indeterministische Informationsquellen, Kollaboration unterschiedlicher Akteure, geringe Vorhersagbarkeit und geringere Struktur der Abläufe auszeichnen [241]. Die Ausführungspfade zur Zielerreichung dieser flexiblen Prozesse sind durch informelle Arbeitspraktiken geprägt. Individualisierungsanforderungen dieser Prozesse können die informationstechnische Unterstützung durch ein System teuer oder unmöglich machen, weil die Eigenschaften dieser Prozesse eine Komplexität erzeugen, die auch deren informationstechnische Unterstützung komplex werden lassen. Es ist also wichtig, Prozessschritte zu erkennen, die durch Menschen bearbeitet werden müssen, und es ist für diese zu prüfen, ob eine informationstechnische Unterstützung wirtschaftlich sinnvoll bzw. grundsätzlich möglich ist. Um Prozessteile zu kennzeichnen, deren Ausführung von den kognitiven Fähigkeiten ihres Bearbeiters abhängig ist, existiert der Annotationstyp Manuelle Bearbeitung (vgl. Tabelle 42)


Tabelle 42: Annotationstyp Manuelle Bearbeitung

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): -
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input type="checkbox"/> Objektlandkarte <input type="checkbox"/> Integrationslandkarte
<p><b>Erläuterung</b></p> <p>Der Annotationstyp Manuelle Bearbeitung wird verwendet, um Landkartenelemente zu kennzeichnen, deren informationstechnische Unterstützbarkeit nicht sicher und die von den kognitiven Fähigkeiten des Bearbeiters abhängig sind. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung darauf zu achten ist, geeignete Mechanismen umzusetzen, die eine (teilweise) informationstechnische Unterstützung ermöglichen, oder geeignete organisatorische Maßnahmen zu ergreifen, um diesen Prozess ohne Informationstechnologie bearbeiten zu können.</p>	
<p><b>Anschlussfragen</b></p> <ul style="list-style-type: none"> <li>• Was erfordert menschliches Handlungs-/Entscheidungsvermögen?</li> <li>• Was sollte die Schnittstelle zwischen Mensch und System leisten, um die manuelle Bearbeitung möglichst gut zu unterstützen / integrieren?</li> <li>• Welche Auswirkungen hätten Fehler in der manuellen Bearbeitung?</li> <li>• Warum ist die manuelle Bearbeitung der Automatisierung vorzuziehen?</li> </ul>	

### C.13 Annotationstyp Automatisierung

Die Automatisierung von Prozessen verspricht geringere Kosten und Bearbeitungszeiten, stellt jedoch unterschiedliche Herausforderungen bei deren Implementierung. Prozesse müssen strukturiert sein und für die automatische Bearbeitung optimiert werden, die Fallzahl muss den Aufwand für die Automatisierung rechtfertigen und es muss der angemessene Automatisierungsgrad festgelegt werden [294]. Sofern Prozesse vollautomatisiert ablaufen, kann ein Regelwerk Entscheidungen und alternative Ausführungspfade dieser Prozesse steuern. Der Fehlerfall kann das Eingreifen von Menschen notwendig machen, sodass Ereignisse identifiziert und die Art des Eingriffs spezifiziert werden müssen. Teilautomatisierte Prozesse werden zur Unterstützung des Menschen bei der digitalen Vorgangsbearbeitung als Workflows umgesetzt und schreiben eine bestimmte Ausführungsreihenfolge vor. Eine Identifikation der Prozesse, die einen bestimmten Automatisierungsgrad besitzen sollen, ist also notwendig, weil sich daraus Anforderungen an die Gestaltung der Prozesse und die Art der informationstechnischen Unterstützung ergeben. Um Prozessteile zu kennzeichnen, die zu einem bestimmten Grad automatisiert werden sollen, existiert der Annotationstyp Automatisierung (vgl. Tabelle 43).


Tabelle 43: Annotationstyp Automatisierung

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): teilweise, vollständig
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input type="checkbox"/> Objektlandkarte <input type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Automatisierung wird verwendet, um Landkartenelemente zu kennzeichnen, deren Ausführung (teil-)automatisiert erfolgen soll. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung die notwendigen fachlichen, technischen und organisatorischen Rahmenbedingungen für die Automatisierung berücksichtigt und umgesetzt werden müssen.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Was soll automatisiert werden?</li> <li>• Zu welchem Grad soll die Automatisierung erfolgen?</li> <li>• Welcher Nutzen wird von der Automatisierung erwartet?</li> </ul>	

## C.14 Annotationstyp Externe Ressource

Geschäftsprozesse überschreiten die Grenzen von Fachbereichen und Unternehmen, jedoch sind unterstützende Informationssysteme oft entlang der Aufbauorganisation geschnitten [295]. Daraus erwächst die Anforderung, dass die unterschiedlichen Informationssysteme integriert werden müssen, um eine Wertschöpfungskette zu unterstützen. Um die Entscheidungsautonomie bei den Systemverantwortlichen zu lassen, müssen stabile Schnittstellen konzipiert und implementiert werden. Deren Konzeption bedarf der Abstimmung Beteiligter und deren Einigung. Dabei kann die Abstimmung durch Heterogenität auf unterschiedlichen Ebenen erschwert werden. Unterschiedliche Datenbankmanagementsysteme oder Programmiersprachen erzeugen technische Heterogenität, unterschiedliche Programmierparadigmen, Datenmodelle und Interpretationen der Realwelt erzeugen konzeptuelle Heterogenität und durch redundante Datenhaltung sowie unterschiedliche Semantik für Gleiches entsteht inhaltliche Heterogenität [295]. Im Rahmen einer Abstimmung zur Integration ist zu prüfen, inwieweit Einfluss auf den Integrationspartner genommen werden kann, wie stabil die Schnittstelle ist, mit welcher Datenqualität zu rechnen ist, ob das Auflösen von Entscheidungsautonomie notwendig, möglich bzw. förderlich ist und welche Hürden durch technische, konzeptuelle und inhaltliche Heterogenität entstehen. Weil diese Abstimmungsprozesse mitunter langwierig sein können und eigene Entscheidungen Abhängigkeiten zum Abstimmungsergebnis besitzen, ist es notwendig, möglichst früh externe Ressourcen zu identifizieren, für die ein solcher Abstimmungsprozess notwendig ist. Zur Kennzeichnung extern anzubindender Ressourcen existiert der Annotationstyp Externe Ressource (vgl. Tabelle 44).


Tabelle 44: Annotationstyp Externe Ressource

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): -
	Verwendung auf: <input type="checkbox"/> Prozesslandkarte <input type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p><b>Erläuterung</b></p> <p>Der Annotationstyp Externe Ressource wird verwendet, um Landkartenelemente zu kennzeichnen, die Daten oder Funktionalität aus einer fremden Quelle beziehen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung eine Quelle im Hinblick auf die Kompatibilität, die Dauer der Verfügbarkeit, die Vertrauenswürdigkeit, die Stabilität, den Grad der Einflussmöglichkeit und die Datenqualität geprüft werden muss.</p>	
<p><b>Anschlussfragen</b></p> <ul style="list-style-type: none"> <li>• Zu welcher externen Ressource besteht eine Schnittstelle?</li> <li>• Welche Herausforderungen entstehen durch Heterogenität zur externen Ressource?</li> <li>• Welchen Nutzen/Zweck hat die Anbindung für uns und für die Ressource?</li> <li>• Welche Einflussmöglichkeit besteht auf den Ressourcen-Inhaber?</li> </ul>	

### C.15 Annotationstyp Randbedingung

Randbedingungen haben Einfluss auf die Umsetzung eines Informationssystems und können obligatorisch oder optional sein. Gesetzliche Vorschriften gehören zu den obligatorischen Randbedingungen, lassen jedoch Gestaltungsspielraum, denn maßgebend ist die Konformität zu Gesetzen und nicht, wie diese Konformität erreicht wird. Vorgaben des Betriebsrats oder Entscheidungen des Managements zählen, genau wie die Aufbau- und Ablauforganisation, zu Quellen organisatorischer Randbedingungen. Technische Randbedingungen wie die Vorgabe bestimmter Programmiersprachen, Technologie-Stacks, Datenbankmanagementsysteme oder die Empfehlung bestimmter Entwurfsmuster schränken die Konzeption und Implementierung eines Informationssystems ein. Jedoch sind gesetzliche, technische und organisatorische Randbedingungen nicht immer trennscharf, sondern können sich gegenseitig bedingen: Produktverantwortliche von Warenherstellern, die sowohl einen Wholesale- als auch Retail-Vertrieb bedienen, dürfen wegen des Kartellrechts keine Preisabsprachen treffen. Organisatorische und technische Vorschriften können zur Erfüllung der gesetzlichen Vorschrift beitragen, indem zum Beispiel die Produktverantwortlichen den Verkaufspreis vorgeben und die technisch unabhängige Implementierung der Bestellsysteme die Einhaltung der kartellrechtlichen Randbedingung sicherstellt. Randbedingungen sind also vielfältiger Natur, daher ist es wichtig, die kritischen Randbedingungen zu identifizieren und in deren Konzeption und Implementierung zu investieren. Zur Kennzeichnung überdurchschnittlich relevanter Randbedingungen existiert der Annotationstyp Randbedingung (vgl. Tabelle 45).


Tabelle 45: Annotationstyp Randbedingung

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): gesetzlich, fachlich, organisatorisch, technisch
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p><b>Erläuterung</b></p> <p>Der Annotationstyp Randbedingung wird verwendet, um Landkartenelemente zu kennzeichnen, dessen Entwurf und Implementierung durch gesetzliche, fachliche, organisatorische oder technische Randbedingungen eingeschränkt ist. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung die Konformität zu der Randbedingung sichergestellt sein muss.</p>	
<p><b>Anschlussfragen</b></p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil ist der Randbedingung unterworfen?</li> <li>• Welche Randbedingung ist zu beachten?</li> <li>• Kann diese Randbedingung ein Show-Stopper werden?</li> </ul>	

## C.16 Annotationstyp Verbesserungsbedarf

Durch die Weiterentwicklung fachlicher Anforderungen, die Optimierung von Prozessen und durch technischen Fortschritt müssen Informationssysteme gewartet und weiterentwickelt werden. Verbesserungsbedarf kann auch in laufenden Projekten entstehen, indem technische Schuld aufgenommen wurde. Technische Schuld führt zu geringer interner Softwarequalität, welche sich positiv auf die Time-to-Market auswirkt und dadurch strategisch eingesetzt werden kann [296]. Durch das unsaubere Entwickeln von Features kann schnelles Feedback zu der entwickelten Funktionalität eingeholt werden, um dieses in eine finale Version einfließen zu lassen. Zu zahlende Zinsen, in Form von Mehraufwand für das Beseitigen der technischen Schuld, werden durch die Entwicklung der verbesserten Feature-Version auf Basis des Nutzer-Feedbacks relativiert [297], [298]. Die Verbesserung von Prozess- und Systemteilen ist also ein relevanter Aspekt, der während der Konzeption und Entwicklung von Informationssystemen berücksichtigt werden muss. Für dessen Kennzeichnung existiert der Annotationstyp Verbesserungsbedarf (vgl. Tabelle 46).


Tabelle 46: Annotationstyp Verbesserungsbedarf

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Anpassung, Erweiterung, Vereinfachung
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Verbesserungsbedarf wird verwendet, um Landkartenelemente zu identifizieren, die fachlich oder technisch angepasst, erweitert oder vereinfacht werden sollen oder müssen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung die Konzepte auf Schwachstellen geprüft werden müssen, um Verbesserungspotential zu erkennen und umzusetzen.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil soll angepasst/erweitert/vereinfacht werden?</li> <li>• Welche Veränderung ist vorgesehen?</li> <li>• Warum und auf wessen Initiative hin soll die Veränderung erfolgen?</li> </ul>	

### C.17 Annotationstyp Ablösung

Eine der größten Herausforderungen in großen Organisationen ist nicht die Entwicklung neuer Informationssysteme, sondern die kontinuierliche Weiterentwicklung und Pflege von Legacy-Systemen. Dabei wachsen Anwendungen eher organisch als strukturiert, was dazu führt, dass der Aufwand für Wartung und Weiterentwicklung ab einem gewissen Zeitpunkt nicht mehr wirtschaftlich ist und die Anwendung am Ende ihres Lebenszyklus angelangt ist. Das Legacy-System wird abgelöst. Jedoch dauert ein Lebenszyklus oft länger, als die Verweildauer der Mitarbeiter in Organisationen ist, die eine Anwendung einst konzipiert und entwickelt haben, so dass fehlendes Wissen und fehlende Erfahrung zum Zeitpunkt der Migration nicht mehr zur Verfügung steht (vgl. zum vorherigen Abschnitt [151]). Um geeignete Maßnahmen für die Ablösung eines Legacy-Systems rechtzeitig ergreifen zu können, ist eine frühe Kennzeichnung abzulösender Systeme oder Systemkomponenten notwendig. Dazu existiert der Annotationstyp Ablösung (vgl. Tabelle 47).


Tabelle 47: Annotationstyp Ablösung

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): Entfall, Ersatz
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p><b>Erläuterung</b></p> <p>Der Annotationstyp Ablösung wird verwendet, um Landkartenelemente zu kennzeichnen, die in Zukunft abgelöst werden sollen oder müssen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung darauf zu achten ist, dass diese Prozess- oder Systemteile zukünftig entfallen oder ersetzt werden.</p>	
<p><b>Anschlussfragen</b></p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil soll abgelöst werden?</li> <li>• Wie soll dessen Aufgabe künftig umgesetzt werden (wenn überhaupt)?</li> <li>• Warum und auf wessen Initiative hin soll die Ablösung erfolgen?</li> </ul>	

## C.18 Annotationstyp Unveränderlichkeit

Genau wie bestimmte Prozess- oder Systemteile als abzulösend identifiziert werden können, können andere Prozess- oder Systemteile als unveränderlich identifiziert werden. Dabei sind die Gründe der Unveränderlichkeit unterschiedlich: Manche Artefakte des Systemkontextes können durch das Entwicklungsprojekt nicht beeinflusst werden, da die Hoheit über die Änderung nicht bei dem Projektteam liegt. Schwerwiegender ist, wenn fachliches oder technisches Wissen, welches für eine Veränderung notwendig wäre, nicht mehr in der Organisation vorhanden ist. Unveränderlichkeit von Artefakten kann auch gewollt sein, beispielsweise wenn eine Änderung unwirtschaftlich ist oder Prozess- und Systemteile über lange Zeit perfektioniert wurden. Die Unveränderlichkeit von Artefakten hat Implikationen auf die Konzeption von Informationssystemen, denn es muss im Rahmen der Entwicklung neuer Artefakte auf die Kompatibilität zu den unveränderlichen Artefakten geachtet werden. Um unveränderliche Prozess- und Systemteile zu kennzeichnen, existiert der Annotationstyp Unveränderlichkeit (vgl. Tabelle 48).


Tabelle 48: Annotationstyp Unveränderlichkeit

	Annotationskategorie: Aufwandstreiber
	Untertyp(en): -
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Unveränderlichkeit wird verwendet, um Landkartenelemente zu kennzeichnen, die im Rahmen der Konzeption und Entwicklung eines Informationssystems entweder nicht verändert werden können oder sollen. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung darauf geachtet werden muss, dass Komptabilität zu nicht veränderbaren Prozess- oder Systemteilen hergestellt wird.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil soll unverändert bleiben?</li> <li>• Wie soll dieser Teil in die veränderte System-/Prozesslandschaft integriert werden?</li> <li>• Warum und auf wessen Initiative hin soll der Teil unverändert bleiben?</li> </ul>	

## C.19 Annotationstyp Komplexität

Entwicklungsprojekte für Informationssysteme sind häufig komplexer, als vom Projektteam zunächst angenommen wird [81], denn die Komplexität ist nicht nur technischer Natur, sondern impliziert auch organisatorische Aspekte, auf die das Projektteam keinen Einfluss hat [80]. Darüber hinaus wächst der Leistungsdruck durch steigende Ressourcenknappheit [79] und raschen technologischen Wandel [242]. Es ist eine zentrale Aufgabe in Entwicklungsprojekten, für Informationssysteme Artefakte zu identifizieren, die Komplexitätstreiber darstellen, um sich der vorhandenen Komplexität früh bewusst zu werden und so die Möglichkeit zu besitzen, die entsprechenden Ressourcen mit adäquater Qualifikation und Erfahrung für die Lösung komplexer Probleme zu allokatieren [88]. Komplexität ist nicht immer unmittelbar erkennbar, sondern erfordert die Einschätzung durch erfahrene Mitglieder eines Projektteams. Um Komplexitätstreiber zu kennzeichnen, existiert der Annotationstyp Komplexität (vgl. Tabelle 49)

Tabelle 49: Annotationstyp Komplexität


	Annotationskategorie: Risikotreiber
	Untertyp(en): -
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<p>Erläuterung</p> <p>Der Annotationstyp Komplexität wird verwendet, um Landkartenelemente zu kennzeichnen, die durch ihre inhärente Komplexität zur Herausforderung im Projekt werden. Der Annotationstyp zeigt, dass bei Entwurf und Implementierung algorithmische Komplexität, Komplexität durch das Zusammenspiel vieler Beteiligter und Komplexität durch technologische Herausforderungen behandelt werden müssen.</p>	
<p>Anschlussfragen</p> <ul style="list-style-type: none"> <li>• Welcher System-/Prozessteil ist besonders komplex?</li> <li>• Worin besteht die Komplexität?</li> <li>• Kann dies ein Show-Stopper werden?</li> </ul>	



## C.20 Annotationstyp Ungewissheit

Zu Beginn eines IT-Projektes ist ein bestimmtes Maß an Ungewissheit unausweichlich [74]. Um Ungewissheit reduzieren zu können, muss das Bewusstsein über diese geschaffen werden. Dies geschieht nicht immer mit der notwendigen Priorität [68]. Das Übersehen von Ungewissheit kann dazu führen, dass Entscheidungen getroffen werden, die in der Form nicht getroffen worden wären, wenn ungewisse Prozess- oder Systemteile ausreichend verstanden worden wären. Das Auflösen der Ungewissheit sollte also stattfinden, bevor Prozess- oder Systemteile entworfen und implementiert werden. Ungewissheit ist fachlich, organisatorisch oder technisch bedingt. Fachliche Ungewissheit kann durch mangelnde Verfügbarkeit von Informationen entstehen oder organisatorischer Natur sein, weil eine Entscheidung aussteht. Technische Ungewissheit kann durch technologischen Fortschritt entstehen, aus dem die Chance einer innovativen Lösung entsteht, deren Machbarkeit ungewiss ist. Die Kennzeichnung von Ungewissheit geschieht mit dem gleichnamigen Annotationstypen, wohingegen das Auflösen nicht im Interaction Room geschieht, sofern Wissensträger nicht verfügbar sind (vgl. Tabelle 50).
















Tabelle 50: Annotationstyp Ungewissheit

	Annotationskategorie: Risikotreiber
	Untertyp(en): Informationsdefizit, Entscheidung ausstehend, Machbarkeit unklar
	Verwendung auf: <input checked="" type="checkbox"/> Prozesslandkarte <input checked="" type="checkbox"/> Objektlandkarte <input checked="" type="checkbox"/> Integrationslandkarte
<b>Erläuterung</b> Der Annotationstyp Ungewissheit wird verwendet, um Landkartenelemente zu kennzeichnen, die fachlich oder technisch ungewiss sind. Die Annotation zeigt, dass für Entwurf und Implementierung ein hohes Risiko besteht, welches priorisiert behandelt werden muss.	
<b>Anschlussfragen</b> <ul style="list-style-type: none"> <li>• Über welche Frage besteht Ungewissheit?</li> <li>• Was ist zu tun, um die Ungewissheit zu beheben?</li> </ul>	

## D Vorlage zur Dokumentation von Annotationen

Die in Abbildung 56 dargestellte Vorlage wird zur Dokumentation von Annotationstypen verwendet.

**Stakeholder** pro: \_\_\_\_\_ contra: \_\_\_\_\_ **ID:** \_\_\_\_\_

	<input type="checkbox"/> Basisfaktor <input type="checkbox"/> Leistungsfaktor <input type="checkbox"/> Begeisterungsfaktor		<input type="checkbox"/> fachl. disruptiv <input type="checkbox"/> techn. disruptiv <input type="checkbox"/> fachl. inkrementell <input type="checkbox"/> techn. inkrementell
	<input type="checkbox"/> Datenschutz <input type="checkbox"/> Datensicherheit		<input type="checkbox"/> dauerhaft <input type="checkbox"/> temporär
	<input type="checkbox"/> Präzision <input type="checkbox"/> Aktualität <input type="checkbox"/> Konsistenz		<input type="checkbox"/> Robustheit <input type="checkbox"/> Verfügbarkeit
	<input type="checkbox"/> zukünft. Änderung <input type="checkbox"/> Anpass. z. Laufzeit <input type="checkbox"/> Variantenbildung		<input type="checkbox"/> mobile Verfügbar. <input type="checkbox"/> offline-Verfügbar. <input type="checkbox"/> Ortsabhängigkeit
	<input type="checkbox"/> teilweise <input type="checkbox"/> vollständig		
	<input type="checkbox"/> Anpassung <input type="checkbox"/> Erweiterung <input type="checkbox"/> Vereinfachung		<input type="checkbox"/> Entfall <input type="checkbox"/> Ersatz
	<input type="checkbox"/> Informationsdefizit <input type="checkbox"/> Entsch. ausstehend <input type="checkbox"/> Machbarkeit unklar		<b>Nutzen:</b> S M L XL <b>Schwierigkeit:</b> S M L XL

**WAS:** Welche Anforderung beinhaltet die Annotation?  
**WO:** Worauf genau bezieht sich die Annotation?  
**WARUM:** Welcher Nutzen wird angestrebt bzw. welcher Schaden vermieden?

Abbildung 56: Vorlage zur Dokumentation von Annotationstypen