



# UNIVERSITÄT PADERBORN

*Die Universität der Informationsgesellschaft*

Fakultät für Elektrotechnik, Informatik und Mathematik  
Institut für Informatik  
Zukunftsmeile 1  
33102 Paderborn

# Testaufwandsschätzung in der analytischen Qualitätssicherung

Dissertation

als Teilleistung zur Erlangung des  
akademischen Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

vorgelegt von

M.Sc. Hendrik Schreiber

Stephanusstr. 29  
33098 Paderborn

1. Gutachter: Prof. Dr. Gregor Engels
2. Gutachter: Prof. Dr. Steffen Becker

Paderborn, 15.04.2015



# Danksagung

Diese Dissertation wäre ohne die fortwährende Unterstützung von meiner Familie, meinen Freunden und meinen Kollegen nicht möglich gewesen. Ich möchte mich daher an dieser Stelle bei allen bedanken, die mich durch diese Zeit begleitet haben.

Mein Dank geht in erster Linie an Svetlana Arifulina. Sie hat mir während der letzten Jahre unbeschreiblich viel Energie, Motivation und Halt gegeben. Ohne sie wäre das Ergebnis nicht das gleiche.

Weiterer Dank geht an meine Familie, die für mich ein ganz starker Rückhalt ist. Meine Eltern und meine Geschwister haben mir immer einen Rückzugsort geboten, um Kraft für die nächsten Schritte zu sammeln.

Die Unterstützung guter Freunde und hilfsbereiter Kollegen hat mir sehr geholfen. Dafür möchte ich ohne besondere Reihenfolge Alexander B., Josef W., Marlene R., Eugen W., Galina B., Dominik S., Florian R., Masud F., Baris G., Yavuz S. sowie allen meinen Kollegen aus dem s-lab, der FG-Engels und dem Unternehmen CRM IT von ganzem Herzen danken.

Die Arbeit wäre ohne die Zusammenarbeit mit dem Unternehmen CRM IT nicht möglich gewesen. Mein Dank gilt dabei insbesondere Dr. Thomas von der Maßen. Er hat über eine Kooperation mit dem s-lab der Universität Paderborn meine Arbeitsstelle im Unternehmen CRM IT gefördert und die Entstehung der Dissertation intensiv begleitet. Hierbei ist auch die gute Zusammenarbeit mit Dr. Stefan Sauer, dem Geschäftsführer des s-labs, hervorzuheben.

Schließlich gilt mein Dank meinem Doktorvater Prof. Dr. Gregor Engels, der mich mit vielen Diskussionen, Ideen und Ratschlägen durch meine Promotion begleitet hat. Danke für die vielen produktiven abendlichen und früh morgendlichen Treffen bei arvato in Gütersloh oder in der Zukunftsmeile.



# **Abstract**

A professional quality assurance is a key factor for companies to face the market challenges and to stay competitive. In the next years, the average budget for quality assurance in a software project is predicted to increase up to 28 % of the overall budget. In order to plan and, therefore, control the rising budgets, test effort estimations are crucial. Using the knowledge about the expected effort and the corresponding costs for quality assurance activities, the budget, resources and time can be reliably planned to achieve a high quality software product.

In a usual case, an expert performs a test effort estimation using her experience. However, this kind of approaches lacks a formal definition that leads to limited comprehensibility and verifiability of the estimated test effort for a reviewer or customer. In order to achieve these goals, a formal method is needed, which considers experiences from previous test projects. The main drawbacks of related approaches is a lack of a complete formalization and an empirical evaluation.

As a result of this PhD Thesis, the TAQ method is developed, which supports test effort estimation for software testing activities in quality assurance. An algorithmic model consisting of two models formally defines the TAQ method. The test complexity model determines, how the test complexity has to be estimated considering the characteristics of each test level. The test cost driver model formalizes the description and the rating of an identified set of influential test cost drivers. These models are combined in a mathematical formula, which can be calibrated on data from previous test projects.

The TAQ method is evaluated empirically. The evaluation includes the analysis of completed test projects in the company CRM-IT, an expert questionnaire for test cost drivers, and the calibration of the TAQ method to the CRM-IT context. The formally defined and empirically evaluated TAQ method guarantees the creation of comprehensible and verifiable test effort estimations.



# Zusammenfassung

Eine professionelle Qualitätssicherung ist für Unternehmen der Schlüsselefaktor, um den Herausforderungen des Marktes zu begegnen und die Wettbewerbsfähigkeit zu erhalten. Für die nächsten Jahre wird eine Steigerung des Budgets für die Qualitätssicherung auf bis zu 28 % des Gesamtbudgets vorhergesagt. Für die Planung und damit die Kontrolle der steigenden Budgets sind Testaufwandschätzungen entscheidend. Nur mit Kenntnis über die zu erwartenden Testaufwände und die daraus entstehenden Kosten sind Budget, Ressourcen und Zeit verlässlich mit dem Ziel eines hochqualitativen Softwareproduktes vereinbar.

Im Normalfall führt ein Experte eine Testaufwandsschätzung basierend auf seiner Erfahrung durch. Die in solchen Ansätzen fehlende formale Definition führt zu einer eingeschränkten Verständlichkeit und Nachvollziehbarkeit des geschätzten Testaufwands. Um dem zu begegnen, wird eine formale Methode benötigt, die Erfahrungen aus vorangegangenen Testprojekten berücksichtigt.

Als Ergebnis wird in dieser Dissertation die Methode TAQ entwickelt, die Testaufwandsschätzungen für Testaktivitäten unterstützt. Die Methode ist durch ein algorithmisches Modell definiert, das aus zwei Modellen zusammengesetzt ist. Das Modell für Testkomplexität legt fest, wie Testkomplexität unter Berücksichtigung der Charakteristiken der Teststufen geschätzt wird. Das Modell für Testeinflussfaktoren formalisiert die Beschreibung und die Bewertung von Testeinflussfaktoren. Beide Modelle werden in einer mathematischen Formel kombiniert, die mit Erfahrungen aus Testprojekten kalibriert werden kann.

Die Methode TAQ ist empirisch evaluiert worden. Die Evaluation umfasst eine Analyse von Testprojekten des Unternehmens CRM-IT, eine Expertenbefragung zu Testeinflussfaktoren und die Kalibrierung an den Kontext des Unternehmens CRM-IT. Die formal definierte und empirisch evaluierte Methode TAQ garantiert verständliche und nachvollziehbare Testaufwandsschätzungen.



# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	xiii
<b>Tabellenverzeichnis</b>	xix
<b>Formelverzeichnis</b>	xxiii
<b>Abkürzungsverzeichnis</b>	xxv
<b>1. Einleitung</b>	1
1.1. Problemstellungen . . . . .	7
1.2. Aufbau der Dissertation . . . . .	11
<b>2. Einführung in Grundlagen und Identifikation von Anforderungen</b>	13
2.1. Vorgehensmodelle für die Softwareentwicklung . . . . .	14
2.1.1. Das V-Modell . . . . .	15
2.1.2. Anforderungen an die Lösung . . . . .	16
2.2. Anforderungsspezifikation . . . . .	18
2.2.1. Rollen bei der Anforderungsspezifikation . . . . .	19
2.2.2. Arten der Anforderungsspezifikation . . . . .	20
2.2.3. Anforderungen an die Lösung . . . . .	23
2.3. Dynamische Analyse: Testen . . . . .	24
2.3.1. Testaktivitäten . . . . .	25
2.3.2. Teststufen . . . . .	28
2.3.3. Anforderungen an die Lösung . . . . .	28
2.4. Einführung in die Domäne Aufwandsschätzung . . . . .	29
2.4.1. Klassifizierung von Methoden zur Aufwandsschätzung . . . . .	30
2.4.2. Expertenschätzungen . . . . .	31
2.4.3. Algorithmische Modelle . . . . .	33
2.4.4. Unterstützende Methoden . . . . .	51
2.4.5. Anforderungen an die Lösung . . . . .	54
2.5. Zusammenfassung . . . . .	56
<b>3. Verwandte Arbeiten und Überblick über den Lösungsansatz</b>	59
3.1. Methoden zur Testaufwandsschätzung . . . . .	59
3.1.1. Berücksichtigung der Testaufwandsschätzung in Standards	60
3.1.2. Testpunktanalyse . . . . .	62
3.1.3. Test Case Points . . . . .	69

## Inhaltsverzeichnis

3.1.4. Expertenschätzung bei dem Unternehmen CRM IT . . . . .	72
3.1.5. Weitere Methoden zur Testaufwandsschätzung . . . . .	76
3.2. Überblick über den Lösungsansatz . . . . .	86
3.3. Zusammenfassung . . . . .	89
<b>4. Entwicklung des Modells für Testkomplexität</b>	<b>91</b>
4.1. Modell für Testkomplexität . . . . .	93
4.1.1. (System-)Integrationstest . . . . .	94
4.1.2. Systemtest . . . . .	100
4.1.3. Abnahmetest . . . . .	109
4.1.4. Fazit . . . . .	113
4.2. Evaluation des Modells für Testkomplexität . . . . .	113
4.2.1. Empirische Untersuchung über Daten aus abgeschlossenen Testprojekten . . . . .	114
4.2.2. Beurteilung des Modells . . . . .	122
4.3. Zusammenfassung . . . . .	125
<b>5. Entwicklung des Modells für Testeinflussfaktoren</b>	<b>127</b>
5.1. Identifikation von Testeinflussfaktoren . . . . .	130
5.1.1. Einflussfaktoren auf den Aufwand in Softwareprojekten . . . . .	131
5.1.2. Einflussfaktoren auf den Testaufwand . . . . .	138
5.1.3. Synthese zu Testeinflussfaktoren . . . . .	144
5.2. Modell für Testeinflussfaktoren . . . . .	145
5.2.1. Entwicklung eines systematischen Vorgehens für den Aufbau des Modells für Testeinflussfaktoren . . . . .	145
5.2.2. Bewertungsverfahren für Testeinflussfaktoren . . . . .	148
5.2.3. Vorgaben zur Beschreibung der Testeinflussfaktoren . . . . .	149
5.2.4. Qualität der Testbasis . . . . .	154
5.2.5. Qualitätsziele . . . . .	156
5.2.6. Wiederverwendbarkeit . . . . .	158
5.2.7. Qualifikation und Erfahrung . . . . .	160
5.2.8. Kommunikationsaufwand . . . . .	164
5.2.9. Zusammenarbeit . . . . .	166
5.2.10. Stabilität und Verfügbarkeit der Testumgebung . . . . .	168
5.2.11. Testinfrastruktur . . . . .	171
5.2.12. Gesamttestorganisation . . . . .	174
5.2.13. Dokumentationsbedarf . . . . .	176
5.2.14. Zeitplanung . . . . .	177
5.3. Evaluation des Modells für Testeinflussfaktoren . . . . .	178
5.3.1. Empirische Untersuchung über eine Expertenumfrage . . . . .	179
5.3.2. Empirische Untersuchung über Daten aus abgeschlossenen Projekten . . . . .	197
5.3.3. Beurteilung des Modells . . . . .	207

5.4. Hinweise zur organisationsspezifischen Kalibrierung . . . . .	209
5.4.1. Nutzung von Daten aus abgeschlossenen Testprojekten .	209
5.4.2. Hinzufügen und Entfernen von Testeinflussfaktoren .	210
5.5. Zusammenfassung . . . . .	211
<b>6. Die Methode TAQ</b>	<b>213</b>
6.1. Die Testaufwandsschätzung mit der Methode TAQ . . . . .	213
6.2. Werkzeugunterstützung für die Methode TAQ . . . . .	217
6.2.1. Konfiguration des Modells für Testeinflussfaktoren .	217
6.2.2. Durchführung einer Testaufwandsschätzung . . . . .	220
<b>7. Zusammenfassung, Bewertung und Ausblick</b>	<b>225</b>
7.1. Zusammenfassung . . . . .	225
7.2. Bewertung der Methode TAQ . . . . .	227
7.2.1. Bewertung in Bezug zu den Problemstellungen . . . . .	227
7.2.2. Bewertung in Bezug zu den Anforderungen . . . . .	228
7.2.3. Vorteile und Gültigkeitsbeschränkung . . . . .	233
7.3. Ausblick . . . . .	234
7.3.1. Erweiterungen und Verbesserungen der Methode TAQ .	235
7.3.2. Weitere Perspektiven für den Einsatz der Methode TAQ	237
<b>Literaturverzeichnis</b>	<b>241</b>
<b>Anhang</b>	
<b>A. Expertenumfrage</b>	<b>255</b>
A.1. Fragebogen . . . . .	255
A.2. Detailergebnisse . . . . .	258
A.2.1. Auswertung der Fragen zu der Berufserfahrung des Probanden . . . . .	258
A.2.2. Auswertung der Fragen zur Erfahrung des Probanden mit Testaufwandsschätzungen . . . . .	262
A.2.3. Auswertung der Fragen zu den Testeinflussfaktoren . .	266
<b>B. Projektdaten</b>	<b>289</b>
B.1. Testkomplexität und Testaufwände nach Testaktivität getrennt .	290
B.1.1. (System-)Integrationstest . . . . .	290
B.1.2. Systemtest . . . . .	291
B.1.3. Abnahmetest . . . . .	294
B.2. Bewertung der Testeinflussfaktoren . . . . .	294
B.3. Expertenschätzungen . . . . .	301
<b>C. Evaluation des Modells für Testeinflussfaktoren</b>	<b>303</b>
C.1. Finalen quantitative Auswirkungen der Testeinflussfaktoren .	303
C.2. Ausgabe der Statistikumgebung R . . . . .	304



# Abbildungsverzeichnis

1.1.	V-Modell nach [WEI09] . . . . .	3
1.2.	Fundamentaler Testprozess (FTP) nach dem ISTQB [Spi11] . . . . .	5
2.1.	V-Modell nach [SL10, S. 33] . . . . .	17
2.2.	Ausschnitt über verantwortliche Rollen für die Erstellung der Artefakte nach [WEI09] . . . . .	19
2.3.	Qualitätsmodell nach ISO 9126 [ISO01] bzw. dem Nachfolger ISO 25010 [ISO11b] . . . . .	25
2.4.	Klassifizierung von Methoden zur Aufwandsschätzung nach Boehm [Boe81, S. 329] . . . . .	31
2.5.	Methoden zur Durchführung einer Expertenschätzung . . . . .	32
2.6.	Methoden zur Aufwandsschätzung mit algorithmischen Modellen . . . . .	35
2.7.	Übersicht über die Ermittlung der UFP nach [AG83] . . . . .	37
3.1.	Schematische Darstellung der Methode Testpunktanalyse nach [vBKV08, S. 459] . . . . .	63
3.2.	Unterschiedliche Abstraktionsgrade in der Modellierung mit Ablaufdiagrammen . . . . .	70
3.3.	Vorlage zur Erstellung einer Testaufwandsschätzung bei der Abteilung QM . . . . .	75
4.1.	Übersicht über den verfeinerten Prozess zur Entwicklung des Modells für Testkomplexität . . . . .	92
4.2.	Darstellung der Architektur des Systemverbundes mit einem UML Komponentendiagramm . . . . .	95
4.3.	Darstellung der Kommunikation der Online Shops mit dem Back-End System in UML Sequenzdiagrammen . . . . .	96
4.4.	Signaturen der Operationen der Schnittstelle I-Back-End in einem UML Klassendiagramm . . . . .	97
4.5.	Die zu berücksichtigenden Eigenschaften der Testobjekte für den (System-)Integrationstest . . . . .	98
4.6.	Übersicht über die Use-Cases der Online Shops und des Back-End Systems als UML Use Case Diagramm . . . . .	101
4.7.	Schritte des Use Cases <i>Online Shop 1 - UC 1 Neuen Kunden mit Nummer anlegen</i> als UML Aktivitätendiagramm . . . . .	103
4.8.	Das <i>Nabe-Speiche</i> -Modell der Anforderungen nach [Coc03, S. 32]	106

## Abbildungsverzeichnis

4.9. Die zu berücksichtigenden Eigenschaften der Testobjekte für den Systemtest . . . . .	107
4.10. Darstellung eines Geschäftsprozesses als UML Aktivitätendiagramm . . . . .	111
4.11. Die zu berücksichtigenden Eigenschaften der Testobjekte für den Abnahmetest . . . . .	112
4.12. Übersicht über die Werkzeugunterstützung als Komponentendiagramm . . . . .	116
4.13. Übersicht über Schwankungen der Produktivität über mehrere Testprojekte. Dabei gehören die Testprojekte zu derselben Software in unterschiedlichen Versionen. . . . .	121
4.14. Gegenüberstellung der Histogramme für die Testkomplexität (Systemtest) vor und nach der Anwendung des Logarithmus . . . . .	123
4.15. Gegenüberstellung der Histogramme für die Testaktivität Testmanagement (Systemtest) vor und nach der Anwendung des Logarithmus . . . . .	123
4.16. Streudiagramm Testkomplexität und Aufwand für die Testaktivität Testmanagement (Systemtest) . . . . .	125
5.1. Übersicht über den Prozess für die Entwicklung des Modells für Testeinflussfaktoren . . . . .	129
5.2. Verbindung des Modells für Testeinflussfaktoren zu GQM . . . . .	147
5.3. Struktur des Modells für Testeinflussfaktoren . . . . .	150
5.4. Ablauf der Zusammenführung der Expertenumfrage mit den Daten aus den abgeschlossenen Testprojekten . . . . .	203
6.1. Übersicht über Use Cases der Methode TAQ . . . . .	214
6.2. Konfiguration des Modells für Testeinflussfaktoren . . . . .	219
6.3. Kalibrieren der Regressionskoeffizienten und Einstellen der Projektphasen für den Testeinflussfaktor Qualität der Testbasis . . . . .	219
6.4. Übersicht über den Ablauf zur Erstellung einer neuen Testaufwandsschätzung mit der TAQ Anwendung . . . . .	221
6.5. Bewertung der Testeinflussfaktoren . . . . .	222
6.6. Eingabe der Parameter für die Berechnung der Testkomplexität . . . . .	223
6.7. Ausgabe der Testaufwandsschätzung für die Teststufe Systemtest und die Testaktivität Testdesign . . . . .	223
A.1. Expertenumfrage: Startseite der Umfrage . . . . .	256
A.2. Expertenumfrage: Exemplarische Seite für Fragen zur Berufserfahrung des Probanden . . . . .	256
A.3. Expertenumfrage: Exemplarische Seite für einen Testeinflussfaktor. Gezeigt wird hier die Seite für den Testeinflussfaktor die Stabilität der Testumgebung. . . . .	257
A.4. Expertenumfrage: Wie gleichartig sind die Projekte im Unternehmen des Probanden? . . . . .	258

A.5. Expertenumfrage: Existiert in dem Unternehmen des Probanden eine selbstständige Qualitätssicherungsgruppe/-abteilung? . . . . .	258
A.6. Expertenumfrage: Welchen Größenklassen sind den durchgeführten Projekte des Probanden zugeordnet? . . . . .	259
A.7. Expertenumfrage: Wie lange dauern die durchgeführten Projekte des Probanden durchschnittlich? . . . . .	259
A.8. Expertenumfrage: Welche Teststufen sind in den Projekten des Probanden relevant? . . . . .	260
A.9. Expertenumfrage: Welcher Rolle ordnet der Proband seine Hauptaufgaben zu? Wie umfangreich ist die Berufserfahrung des Probanden in der Industrie? . . . . .	261
A.10. Expertenumfrage: Wie wird der Aufwand (Budget und Zeit) der Qualitätssicherung in den Projekten des Probanden geplant? . . . . .	262
A.11. Expertenumfrage: Wie häufig führt der Proband Aufwandsschätzungen für die Teststufe Abnahmetest durch? . . . . .	262
A.12. Expertenumfrage: Wie häufig führt der Proband Aufwandsschätzungen für die Teststufe Systemtest durch? . . . . .	263
A.13. Expertenumfrage: Wie häufig führt der Proband Aufwandsschätzungen für die Teststufe (System-)Integrationstest durch? . . . . .	263
A.14. Expertenumfrage: Wie häufig führt der Proband Aufwandsschätzungen für die Teststufe Komponententest durch? . . . . .	264
A.15. Expertenumfrage: Wie häufig führt der Proband Aufwandsschätzungen für die Teststufe Last-/Performancetest durch? . . . . .	264
A.16. Expertenumfrage: Wie häufig führt der Proband Aufwandsschätzungen für die Teststufe Usability Test durch? . . . . .	265
A.17. Expertenumfrage: Wie häufig führt der Proband Aufwandsschätzungen für andere Teststufen durch? . . . . .	265
A.18. Expertenumfrage: Ist für eine Testaufwandsschätzung eine Be trachtung des Kontextes z. B. über Testeinflussfaktoren erforderlich? . . . . .	266
A.19. Expertenumfrage: Boxplot für den Testeinflussfaktor Qualität der Testbasis . . . . .	267
A.20. Expertenumfrage: Hat die Qualität der Testbasis relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	268
A.21. Expertenumfrage: Wie häufig ist in Ihren Projekten die Qualität der Testbasis während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	269
A.22. Boxplot für den Testeinflussfaktor Qualitätsziele . . . . .	269
A.23. Expertenumfrage: Haben die Qualitätsziele relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	270

## Abbildungsverzeichnis

A.24.Expertenumfrage: Wie häufig sind in Ihren Projekten die Qualitätsziele während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	271
A.25.Boxplot für den Testeinflussfaktor Qualifikation und Erfahrung .	272
A.26.Expertenumfrage: Hat die Qualifikation und die Erfahrung der Projektmitglieder relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	273
A.27.Expertenumfrage: Wie häufig ist in Ihren Projekten die Qualifikation und die Erfahrung der Projektmitglieder während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	274
A.28.Boxplot für den Testeinflussfaktor Kommunikationsaufwand . .	275
A.29.Expertenumfrage: Hat der Kommunikationsaufwand relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	275
A.30.Expertenumfrage: Wie häufig ist in Ihren Projekten der Kommunikationsaufwand während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	276
A.31.Boxplot für den Testeinflussfaktor Zusammenarbeit . . . . .	276
A.32.Expertenumfrage: Hat die Zusammenarbeit der Projektmitglieder relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	277
A.33.Expertenumfrage: Wie häufig ist in Ihren Projekten die Zusammenarbeit der Projektmitglieder während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	278
A.34.Boxplot für den Testeinflussfaktor Stabilität der Testumgebung	279
A.35.Expertenumfrage: Hat die Stabilität der Testumgebung relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	280
A.36.Expertenumfrage: Wie häufig ist in Ihren Projekten die Stabilität der Testumgebung während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	281
A.37.Boxplot für den Testeinflussfaktor Testinfrastruktur . . . . .	281
A.38.Expertenumfrage: Hat die Testinfrastruktur relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	282
A.39.Expertenumfrage: Wie häufig ist in Ihren Projekten die Testinfrastruktur während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	283
A.40.Boxplot für den Testeinflussfaktor Gesamttestorganisation . . .	284
A.41.Expertenumfrage: Hat die Gesamttestorganisation relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	284

## *Abbildungsverzeichnis*

A.42.Expertenumfrage: Wie häufig ist in Ihren Projekten die Testinfrastruktur während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	285
A.43.Boxplot für den Testeinflussfaktor Dokumentationsbedarf . . . . .	286
A.44.Expertenumfrage: Wie häufig ist in Ihren Projekten der Grad des Dokumentationsbedarfs während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	286
A.45.Boxplot für den Testeinflussfaktor Zeitplanung . . . . .	287
A.46.Expertenumfrage: Hat die Zeitplanung einen relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung? . . . . .	287
A.47.Expertenumfrage: Wie häufig ist in Ihren Projekten die Zeitplanung während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? . . . . .	288



# Tabellenverzeichnis

2.1. Beschreibung der Rollen Testmanager, Testdesigner und Tester . . . . .	26
2.2. Gewichtung der Funktionen nach [Int10] zur Ermittlung der UFP . . . . .	39
2.3. GSC der Methode Function Point nach [AG83] . . . . .	41
2.4. SF von COCOMO II nach [Boe00] . . . . .	44
2.5. EM im <i>Post-Architecture</i> Modell von COCOMO II nach [Boe00]	45
2.6. Gewichtung der Use Cases nach [Kar93a] . . . . .	47
2.7. Gewichtung der Akteure nach [Kar93a] . . . . .	47
2.8. Einflussfaktoren zur Berechnung des TCF nach [Kar93a] . . . . .	49
2.9. Einflussfaktoren zur Berechnung des EFA nach [Kar93a] . . . . .	49
2.10. Übersicht über die Anforderungen (A) und die Zuordnung zu den Problemstellungen (P) . . . . .	57
3.1. Funktionsabhängige Faktoren zur Berechnung des Einflussfaktors $A_f$ nach [vBKV08] . . . . .	64
3.2. Faktoren zur Berechnung des Einflussfaktors $Q_d$ nach [vBKV08] . . . . .	65
3.3. Multiplikatoren zur Berechnung des Einflussfaktors $Q_d$ nach [vBKV08] . . . . .	65
3.4. Variablen zur Bewertung des Umgebungsfaktors nach [vBKV08] . . . . .	67
3.5. Gewichtung der Entscheidungspunkte nach [Nel12] . . . . .	71
3.6. Schlüsselwörter für die Literaturrecherche . . . . .	78
3.7. Übersicht über die gefundenen Literaturquellen . . . . .	81
3.8. Übersicht über die Parameter, deren Veränderung eine erneute Berechnung der Testkomplexität und lokalen Produktivität erfordert . . . . .	88
4.1. Annahmen für die Berechnung der Testkomplexität für einen (System-)Integrationstest in frühen Projektphasen oder bei nicht vollständiger Dokumentation . . . . .	100
4.2. Gewichtung der Akteure in der Methode TAQ . . . . .	104
4.3. Annahmen für die Berechnung der Testkomplexität für einen Systemtest in frühen Projektphasen oder bei nicht vollständiger Dokumentation . . . . .	108
4.4. Anzahl Datensätze in der relationalen TAQ Datenbank . . . . .	116
4.5. Globale Produktivität für das Unternehmen CRM IT in $\frac{TKP}{PT}$ . . . . .	119
5.1. Einflussfaktoren der Kategorie Produkt auf den Aufwand eines Softwareentwicklungsprojektes . . . . .	134

## Tabellenverzeichnis

5.2. Einflussfaktoren der Kategorie Personen auf den Aufwand eines Softwareentwicklungsprojektes . . . . .	135
5.3. Einflussfaktoren der Kategorie Infrastruktur auf den Aufwand eines Softwareentwicklungsprojektes . . . . .	136
5.4. Einflussfaktoren der Kategorie Organisation auf den Aufwand eines Softwareentwicklungsprojektes . . . . .	136
5.5. Einflussfaktoren der Kategorie Produkt auf den Aufwand eines Testprojektes . . . . .	140
5.6. Einflussfaktoren der Kategorie Personal auf den Aufwand eines Testprojektes . . . . .	141
5.7. Einflussfaktoren der Kategorie Infrastruktur auf den Aufwand eines Testprojektes . . . . .	142
5.8. Einflussfaktoren der Kategorie Organisation auf den Aufwand eines Testprojektes . . . . .	142
5.9. Synthese zu Testeinflussfaktoren . . . . .	144
5.10. Vorlage für die Beschreibung der Testeinflussfaktoren . . . . .	152
5.22. Analyse der Antworten für die Frage nach der Wahrscheinlichkeit der Auswirkungen der Testeinflussfaktoren auf die Testaktivitäten	188
5.23. Analyse der Antworten für die Frage nach der Möglichkeit einer guten Einschätzung der Testeinflussfaktoren in den Projektphasen	189
5.24. Analyse der Antworten für die Frage nach der Höhe der Auswirkungen eines Testeinflussfaktors auf den Testaufwand. . . . .	190
5.25. Initialen quantitativen Auswirkungen (IQA), die aus den Ergebnissen der Expertenumfrage abgeleitet worden ist . . . . .	204
5.26. Konstante $C$ . . . . .	206
5.27. Berechnete FQA je qualitativer Ausprägung der Testeinflussfaktoren für die Testaktivität Testdesign. Dabei wurden die IQA zu 30 % berücksichtigt. . . . .	207
5.28. Durchschnittliche Vorhersagegenauigkeit von 10.000 berechneten Modellen und die höchste erreichte Vorhersagegenauigkeit eines Modells je Testaktivität . . . . .	208
5.29. Vorhersagegenauigkeit für die Teststufe Systemtest des expertenbasierten Ansatzes des Unternehmens CRM IT . . . . .	208
6.1. Übersicht über die Anwendbarkeit der Testeinflussfaktoren in Projektphasen (Anforderungsphase (AP), Designphase (DP), Entwicklungsphase (EP)) und für Testaktivitäten (Testmanagement (TM), Testdesign (TD) und Testausführung (TA)) . . . . .	216
B.1. Überblick über die Testprojekte für die Stufe (System-)Integrationstest - Testkomplexität in TKP und Aufwände in PT . . . . .	290
B.2. Überblick über die Testprojekte für die Stufe Systemtest - Testkomplexität in TKP und Aufwände in PT . . . . .	293

B.3. Überblick über die Testprojekte für die Stufe Abnahmetest - Testkomplexität in TKP und Aufwände in PT . . . . .	294
B.4. Bewertung der Testeinflussfaktoren entsprechend der bijektiven Funktion $S$ in Formel 5.4 . . . . .	300
B.5. Expertenschätzungen in PT der Firma CRM IT für die Stufe Systemtest . . . . .	302
C.1. Berechnete FQA je qualitativer Ausprägung der Testeinflussfaktoren für die Testaktivität Testmanagement. Dabei wurden die IQA zu 60 % berücksichtigt. . . . .	303
C.2. Berechnete FQA je qualitativer Ausprägung der Testeinflussfaktoren für die Testaktivität Testausführung. Dabei wurden die IQA zu 90 % berücksichtigt. . . . .	304
C.3. Übersicht über die $R^2$ und die korrigierten $R^2$ für die Testaktivitäten Testmanagement, Testdesign und Testausführung . . . . .	304
C.4. Ausgabe der Statistikumgebung R für die Testaktivität Testmanagement . . . . .	305
C.5. Ausgabe der Statistikumgebung R für die Testaktivität Testdesign	306
C.6. Ausgabe der Statistikumgebung R für die Testaktivität Testausführung . . . . .	307



# Formelverzeichnis

2.1.	Algorithmisches Modell . . . . .	34
2.2.	Lineare Form einer Formel zur Aufwandsschätzung . . . . .	34
2.3.	Multiplikative Form einer Formel zur Aufwandsschätzung . . . . .	34
2.4.	UFP durch Aufsummieren der Gewichtungen . . . . .	39
2.5.	VAF Berechnung . . . . .	40
2.6.	AFP Berechnung . . . . .	40
2.7.	Berechnung des Aufwands in PM nach COCOMO II . . . . .	43
2.8.	Berechnung des Exponenten in COCOMO II . . . . .	43
2.9.	Menge der UUCP . . . . .	47
2.10.	Berechnung des TCF . . . . .	48
2.11.	Berechnung des EF . . . . .	48
2.12.	Berechnung der UCP . . . . .	48
3.8.	Übersicht über die Formel der Lösung . . . . .	87
4.1.	Testkomplexität (System-)Integrationstest . . . . .	99
4.2.	Testkomplexität Systemtest . . . . .	108
4.3.	Testkomplexität Abnahmetest . . . . .	112
4.4.	Berechnung des Mittelwertes einer Stichprobe . . . . .	124
4.5.	Berechnung der Varianz einer Stichprobe . . . . .	124
4.6.	Berechnung der Standardabweichung einer Stichprobe . . . . .	124
4.7.	Berechnung der Kovarianz einer Stichprobe . . . . .	124
4.8.	Berechnung des Korrelationskoeffizienten . . . . .	125
5.1.	Ordinalskala zur Bewertung Testeinflussfaktoren . . . . .	148
5.4.	Funktion $S$ zur Übersetzung der Skala aus Formel 5.1 . . . . .	153
5.5.	Berechnung des Medians einer Stichprobe . . . . .	186
5.6.	Berechnung der Prüfgröße $\chi^2$ . . . . .	187
5.7.	Einfache lineare Regression . . . . .	201
5.8.	Multiple lineare Regression . . . . .	201
5.9.	Ausgangspunkt für die Regressionsanalyse . . . . .	202
5.10.	Anwendung Logarithmus auf Gleichung zur Regressionsanalyse	202
5.11.	Eingabe für Regressionsanalyse . . . . .	202
5.12.	Formel mit Regressionskoeffizienten $b_i$ nach Regressionsanalyse	205
5.13.	Umformung der Formel nach Regressionsanalyse . . . . .	205
5.14.	Endgültige quantitative Auswirkung für einen Testeinflussfaktor	206
5.15.	Berechnung der Konstante mit FQA . . . . .	206

*Formelverzeichnis*

6.1.	Schätzung der Testkomplexität für den Systemtest . . . . .	215
6.2.	Testaufwandsschätzung für den Systemtest . . . . . . . . .	216

# Abkürzungsverzeichnis

AFP	Adjusted Function Points.
AG	Auftraggeber.
AN	Auftragnehmer.
BM	Bewertungsmodell.
BPMN	Business Process Model and Notation.
COCOMO	Constructive Cost Model.
CRM IT	Unternehmen CRM IT - BA der arvato business support GmbH.
DP	Entscheidungspunkt ( <i>Decision Point</i> ).
EF	Einflussfaktor.
EFA	Environmental Factor.
EFT	Einflussfaktoren auf den Testaufwand.
EI	External Inputs.
ELF	External Logical Files.
EM	Effort Multiplier.
EO	External Outputs.
EQ	External Inquiry.
FL	Pfad ( <i>Flow</i> ) in einem Ablaufdiagramm.
FP	Function Point Methode.
FQA	Finale quantitative Auswirkungen eines Testeinflussfaktors.
GSC	General System Characteristics.
IEC	International Electrotechnical Commission.
IFPUG	International Function Point User Group.
ILF	Internal Logical Files.
IP	Schnittstellenpunkt ( <i>Interface Point</i> ).
IQA	Initiale quantitative Auswirkungen eines Testeinflussfaktors.
ISBSG	International Software Benchmarking Standards Group.
ISO	International Organization for Standardization.

## *Abkürzungsverzeichnis*

ISTQB	International Software Testing Qualifications Board.
IT	Informationstechnik.
KF	Kompetenzfaktor.
LOC	Lines of Code.
MBT	Modellbasierte Testen.
NUCP	Nageswaran Use Case Points.
OMG	Object Management Group.
PMBOK	Project Management Body of Knowledge.
PT	Personentag, 8 Stunden Arbeitstag.
PTS	Primären Teststunden.
QA	Qualitative Ausprägung.
QM	Abteilung für Qualitätsmanagement bei der CRM IT.
RUP	Rational Unified Process.
SEER-SEM	Software Evaluation and Estimation of Resources - Software Estimating Model.
SF	Scale Factors.
SLIM	Putnam's Software Life-cycle Model.
SPEM	Software & Systems Process Engineering Meta-Model.
TAQ	Methode zur Testaufwandsschätzung in der analytischen Qualitätssicherung.
TCF	Technical Complexity Factor.
TCP	Test Case Point.
TF	Testeinflussfaktor.
TKP	Testkomplexitätspunkt.
TP	Testpunkt.
TPA	Testpunktanalyse.
TPI	Test Process Improvement.
UAW	Unadjusted Actor Weights.
UCP	Use Case Point.
UCPM	Use Case Point Methode.

*Abkürzungsverzeichnis*

UF	Umgebungs faktor.
UFP	Unadjusted Function Points.
UML	Unified Modeling Language.
UUCP	Unadjusted Use Case Points.
UUCW	Unadjusted Use Case Weights.
VAF	Value Adjustment Factors.



# 1. Einleitung

It is very difficult to make a vigorous, plausible, and job-risking defense of an estimate that is derived by no quantitative method, supported by little data and certified chiefly by the hunches of the managers.

---

(Fred Brooks [Bro95])

Immer komplexere IT-Systeme bilden den Kern für das operative Geschäft von Unternehmen. Dabei sind Geschäftsprozesse entweder gesamt oder zumindest in Teilen nur noch durch Software unterstützt ausführbar. Die notwendige Software verteilt sich hierbei über eine Vielzahl von Systemen mit unterschiedlichen Schnittstellen. Das korrekte Zusammenspiel der Systeme und das korrekte Ausführen von Geschäftsprozessen sicherzustellen ist bei der steigenden Komplexität eine große Herausforderung für Unternehmen. Nur mit einer professionellen Qualitätssicherung kann diesen Herausforderungen begegnet und damit die Wettbewerbsfähigkeit der Unternehmen erhalten werden.

Die Notwendigkeit einer professionellen Qualitätssicherung ist bereits erkannt worden. Dies spiegelt sich in Umfragen, wie z. B. *Softwaretest 2011: Softwaretest in der Praxis* [SVWH12], *World Quality Report 2013* [BT13, S. 18], *Acceptance Monitor 2014: Studienergebnisse* [BDC14] und *ASQF-Umfrage 2014* [ASQ14]. Die Zahl von separaten Unternehmensabteilungen für Qualitätssicherung, der Anteil von ausgebildeten Testern und der Wunsch nach Testprozessverbesserungen sind signifikant gestiegen [van11, SVWH12, BDC14][BT13, S. 18]. Auch die steigende Verbreitung von Standards im Bereich der Qualitätssicherung, wie der des ISTQB, ist ein Hinweis auf die heutige Bedeutung. Beispielsweise sind nach der offiziellen Internetseite des ISTQB mittlerweile 280.000 Zertifizierungen vergeben worden [IST].

## *1. Einleitung*

Ein weiteres Indiz für die Wichtigkeit von professionellen Qualitätssicherungsmaßnahmen ist der durchschnittliche Anteil von 23 % am Gesamtbudget für die Entwicklung des Softwareproduktes [BT13, S. 18]. Vereinzelt kann der Anteil am Gesamtentwicklungsbudget sogar auf mehr als 80 % ansteigen [Spi11, SVWH12, BDC14]. Über die nächsten zwei Jahre wird ein weiterer Anstieg des durchschnittlichen Budgets für die Qualitätssicherung auf bis zu 28 % prognostiziert [BT13, S. 7]. Dieser hohe steigende Anteil am Gesamtbudget ist ein Zeichen für den Bedarf an Aufwandsschätzungen im Bereich der Qualitätssicherung. Nur mit Kenntnis über die zu erwartenden Aufwände und die daraus entstehenden Kosten für die Maßnahmen zur Qualitätssicherung können Budget, Ressourcen und Zeit verlässlich mit dem Ziel eines qualitativ hochwertigen Softwareproduktes vereinbart werden.

Eine professionelle Qualitätssicherung umfasst alle Maßnahmen, die notwendig sind, ein Softwareprodukt auf die Erfüllung spezifizierter Anforderungen zu überprüfen [IEE91, S. 163]. Damit lässt sich die Qualität des Softwareproduktes in dem Grad der Erfüllung dieser Anforderungen ausdrücken. Die Maßnahmen für die Qualitätssicherung lassen sich in prozessorientierte, konstruktive und analytische trennen.

Zu prozessorientierten Maßnahmen gehört die Überprüfung der Prozesse, die für die Erstellung des Produktes genutzt werden [IEE91, S. 163]. Dazu könnten vereinfacht die Prozesse der Anforderungsaufnahme, die der Entwicklung oder die des Testens gehören. Konstruktive Maßnahmen dienen der Fehlervermeidung während der Entwicklung des Softwareproduktes. Aktivitäten können hierbei z. B. die Verwendung von Standards, Entwurfsmustern und passender Modellierungssprachen sein. Da die prozessorientierten und konstruktiven Maßnahmen zur Qualitätssicherung übergreifend für alle Softwareprodukte einer Organisation gelten, sind sie nur schwer für ein Softwareprodukt quantifizierbar. Daher werden solche Maßnahmen in dieser Dissertation nicht weiter betrachtet.

Analytische Maßnahmen arbeiten auf Artefakten, die bereits erstellt worden sind. Dazu gehört zunächst die statische Analyse. Zu der statischen Analyse zählen Überprüfungen von Dokumenten, wie z. B. Pflichtenheften oder auch des fertigen Quellcodes ohne ihn auszuführen [Spi11, S. 81ff]. Als Gegenpart zur statischen Analyse existiert die dynamische Analyse, die das Überprüfen der Software zur Laufzeit bedeutet [Spi11, S. 109ff]. Im Weiteren wird hierfür das

verbreitete Synonym (Software-) *Test* verwendet. Das Testen einer Software hat einen großen Anteil am Gesamtbudget. Die weiter oben genannten Prozentzahlen beziehen sich z. B. fast ausschließlich auf den Softwaretest. Daher wird im Rahmen dieser Dissertation der Fokus auf das Testen eines Softwareproduktes gesetzt.

Das Testen eines Softwareproduktes erfolgt in Testprojekten über mehrere sogenannte Teststufen [Spi11, S. 41ff]. In Abbildung 1.1 sind in Anlehnung an das etablierte Vorgehensmodell für die Softwareentwicklung V-Modell XT [WEI09] die Teststufen Komponententest bis Abnahmetest zu sehen. In dieser Dissertation werden die hervorgehobenen Teststufen (System-)Integrationstest, Systemtest und Abnahmetest betrachtet. Der Aufwand für die Durchführung der Teststufe Komponententest ist mit dem Aufwand für das Programmieren des Softwareproduktes zu schätzen, da die Tests typischerweise von den Entwicklern des Softwareproduktes selbst programmiert und durchgeführt werden müssen [SL10, S. 47]. Damit ist kein dediziertes Testpersonal an der Teststufe Komponententest beteiligt.

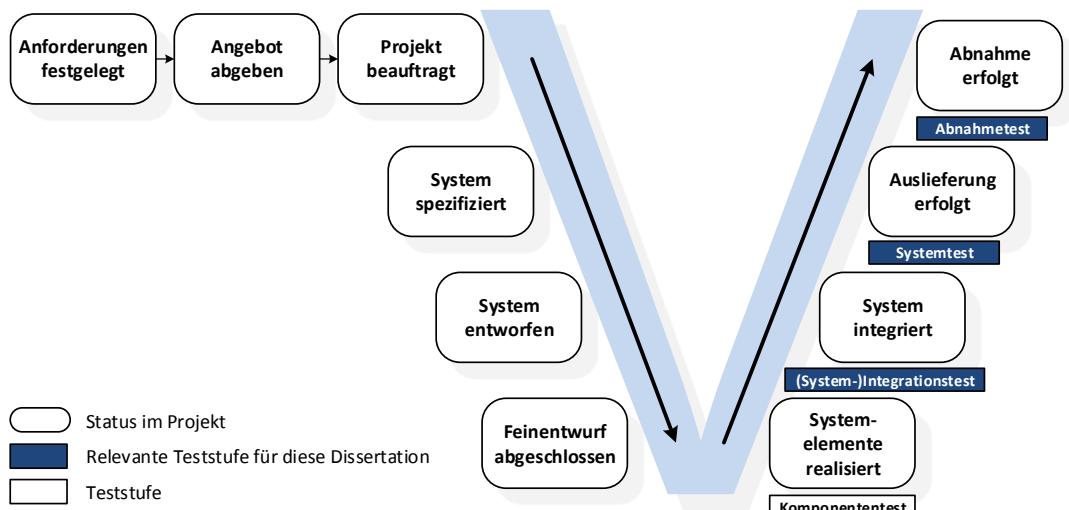


Abbildung 1.1.: V-Modell nach [WEI09]

Für jede Teststufe wird typischerweise eine Testaufwandsschätzung durchgeführt. Dabei wird bewusst der Begriff *Schätzung* verwendet (vgl. Definition 1), da auch der mathematisch ermittelte Testaufwand nur eine Annäherung an den tatsächlichen Testaufwand darstellt. Dabei liegt diese Annäherung nach

## 1. Einleitung

DeMarco mit gleicher Wahrscheinlichkeit unter oder über dem tatsächlichen Wert [DeM82, S. 14].

### Definition 1: Testaufwandsschätzung

Die Testaufwandsschätzung bedeutet eine Ermittlung oder Berechnung eines Zahlenwertes, z. B. in Personentagen, der eine Annäherung an den tatsächlichen Testaufwand darstellt.

Ein Testprojekt unterliegt häufig besonderem Zeitdruck, da es am Ende eines Softwareentwicklungsprojektes steht. Beispielsweise müssen bestimmte Lieftertermine eingehalten werden, die Entwicklung hat sich verzögert oder das Gesamtprojektbudget ist angespannt. Eine mögliche Folge des Drucks ist, dass das Testen nicht ausreichend berücksichtigt wird und Fehler im IT-System verbleiben. Diese Fehler können sich leicht zu größeren Problemen entwickeln, wenn das Unternehmen in dem Ausüben seiner Geschäftsprozesse beeinträchtigt wird.

Um mit dem Zeitdruck in einem Testprojekt umgehen zu können, ist ein hohes Maß an Planung und Kontrolle des Soll- und Istzustandes des Testprojektes notwendig. Abweichungen vom Sollzustand können damit entsprechend schnell erkannt und Gegenmaßnahmen für das Testprojekt ergriffen werden. Voraussetzung für die Kenntnis über den Soll- und Istzustand eines Testprojektes ist ein kontrollierter Prozess für dessen Durchführung. In so einem Prozess wird das Testen eines Softwareproduktes in einzelne Aktivitäten zerlegt und angeordnet. Darüber ist die Kontrolle über den Soll- und Istzustand möglich.

Ein Beispiel für einen kontrollierten Prozess ist der fundamentale Testprozess (FTP) des ISTQB [Spi11]. Dieser Prozess wird in Abbildung 1.2 gezeigt. Er besteht aus den Aktivitäten *Planung und Steuerung*, *Analyse und Design*, *Realisierung und Durchführung*, *Auswertung und Bericht* und *Abschluss*. Die hervorgehobene Aktivität *Planung und Steuerung* ist für die Testaufwandsschätzung von besonderer Bedeutung, da sie die Planung des Testprojektes und die fortlaufende Kontrolle und Steuerung der anderen Aktivitäten im Testprozess umfasst. Zur Planung des Testprojektes gehören das Festlegen der Teststrategie und eine Schätzung des Testaufwands in Personentagen, der benötigt wird um die Teststrategie umzusetzen. Auf dieser Basis können anschließend Aussagen zu den benötigten Ressourcen (z. B. Personal oder Infrastruktur), dem Zeitplan

und der voraussichtlichen Dauer bis zum Abschluss der Maßnahmen getroffen werden. Mit der Testaufwandsschätzung und der darauf basierenden Planung ist der Status des Testprojekts zur Durchführungszeit nachverfolgbar, da der Sollzustand mit dem Istzustand vergleichbar ist.

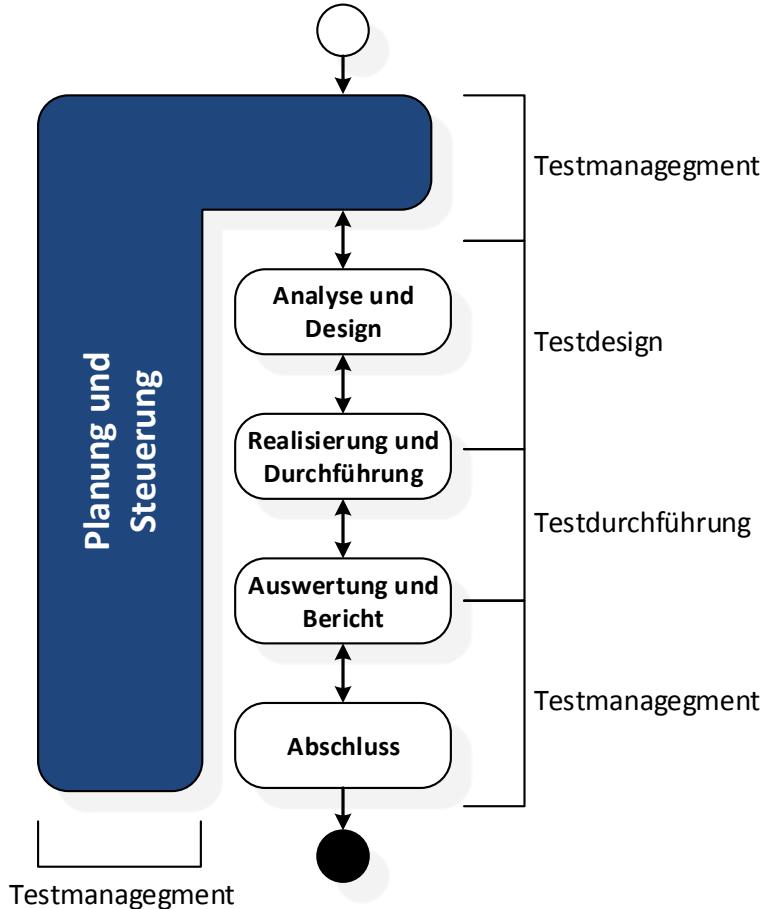


Abbildung 1.2.: Fundamentaler Testprozess (FTP) nach dem ISTQB [Spi11]

Eine entsprechend nachvollziehbare und detaillierte Testaufwandsschätzung ist der Ausgangspunkt für die Planung von (Personal-)Ressourcen, da die Aktivitäten des FTP von Personen in unterschiedlichen Rollen ausgeführt werden. Speziell die Auslastung von hochqualifiziertem Testpersonal muss lange im Vorfeld geplant werden, damit die betreffenden Personen auch bei Testprojektstart zur Verfügung stehen. Bedingung für die Planung von (Personal-)Ressourcen ist die Zuordnung der Aktivitäten des FTP zu Rollen, die diese Aktivitäten ausführen. Damit können die geschätzten Aufwände auf die Rollen und anschließend auf Personen verteilt werden. So kann sichergestellt werden, dass ausreichend Personal zur Verfügung steht.

## 1. Einleitung

Das ISTQB gibt im FTP und im Allgemeinen kein vollständiges Rollenmodell vor. Dennoch existieren in der Standardliteratur zum ISTQB von Spillner Vorschläge für fachliche (Team-)Rollen, die in dieser Dissertation übernommen werden: Testmanager, Testdesigner und Tester [Spi11, S. 271]. Die Rolle Testmanager ergibt sich direkt aus der Zertifizierung des ISTQB zum Testmanager [Int12c]. Dieser Rolle sind die Aktivitäten *Planung und Steuerung*, *Bericht* und *Abschluss* des FTP zugeordnet. Da die verbleibenden Aktivitäten des FTP nicht direkt über eine verfügbare Zertifizierung des ISTQB einer Rolle zugewiesen werden können, wird eine eigene Einteilung der Aktivitäten gewählt. Dabei werden die Aktivitäten zur Testvorbereitung (*Analyse und Design*, *Realisierung*) und die Aktivität zur Testausführung (*Durchführung*) getrennt betrachtet. Daraus ergeben sich die Rolle Testdesigner für die Aktivitäten der Testvorbereitung und die Rolle Tester für die Aktivität der Testausführung. Ein weiterer Grund für diese Rollenaufteilung sind Unterschiede in der Qualifikation des Personals, da Testdesigner in vielen Fällen höher als Tester qualifiziert sein müssen [Int11, Int12b, Int12a]. Mit dieser Zuordnung von Rollen zu Aktivitäten ist die Verteilung der Aufwände auf Rollen und anschließend auf Personen möglich.

Die Testaufwandsschätzung als zentrales Element bei der Planung des Testprojektes wird oft vernachlässigt. Hier fehlt auf der einen Seite das entsprechende methodische Vorgehen, was zu nicht nachvollziehbaren und ungenauen Testaufwandsschätzungen führt. Auf der anderen Seite wird der Testaufwand oft pauschal mit anderen Projektaktivitäten angegeben [SVWH12]. In beiden Fällen findet keine ausreichende Betrachtung des erforderlichen Testaufwands statt und der Testaufwand wird unter- oder überbewertet. Im Fall der Unterbewertung kann dies zu deutlichen Budget- und auch Zeitüberschreitungen führen. Teilweise sind 100 % Budgetüberschreitungen möglich [SVWH12]. Im Fall der Überbewertung tritt oft ein anderer Effekt ein, der mit Parkinsons Gesetz beschrieben wird:

“Work expands so as to fill the time available for its completion.“ [Par57]

Das Gesetz von Parkinson bedeutet, dass eine Aufgabe so ausgedehnt wird, bis die zur Verfügung Zeit verbraucht ist. Damit wird die Vorhersage des Aufwands

zu 100 % erfüllt, aber dem Kunden wird zu viel in Rechnung gestellt und die Mitarbeiter sind zu lange für das Projekt eingeplant.

Die Erstellung einer nachvollziehbaren und detaillierten Testaufwandsschätzung erfordert mehrjährige Erfahrung im Testbereich, da die Höhe des Testaufwands von vielen Einflussfaktoren abhängig ist. Über diese Einflussfaktoren existieren oft zum Zeitpunkt der Bestimmung des Testaufwands nur wenige Informationen. Auch nach Abschluss eines Testprojektes ist die Menge der Einflussfaktoren nicht vollständig bestimmt, da zu viele Kontextfaktoren auf den Testaufwand wirken. Damit ist der Testaufwand nicht direkt aus bekannten Parametern, Einflussgrößen und Konstanten ableitbar.

Für die Planung des Testprojektes ist ein methodischer und nachvollziehbarer Ansatz für die Testaufwandsschätzung unerlässlich. Ansonsten wird die Verteidigung einer Testaufwandsschätzung und auch einer darauf basierenden Planung sehr schwierig, wie Fred Brooks in dem Zitat am Anfang des Abschnitts geschrieben hat. Weiterhin sind Unter- und Überschätzungen mit den beschriebenen Folgen nicht die Ausnahme sondern die Regel. Bei der Erstellung von nachvollziehbaren Testaufwandsschätzungen entstehen viele Problemstellungen, von denen einige in dieser Arbeit behandelt werden. Diese Problemstellungen werden im Abschnitt 1.1 definiert.

## 1.1. Problemstellungen

Das Unternehmen CRM IT - BA der arvato business support GmbH (CRM IT) entwickelt Softwareprodukte zur Unterstützung von Geschäftsprozessen. Ein Beispiel sind Kundenbindungsprogramme, bei denen der darunterliegende Systemverbund Geschäftsprozesse von der Kasse in einem Laden bis zur Druckstraße für den Rechnungsdruck unterstützt. Für die Qualitätssicherung des Softwareproduktes oder auch des gesamten Systemverbundes ist die Abteilung Qualitätsmanagement (QM) als dauerhafte Einrichtung verantwortlich.

Bei der Organisation eines Testprojektes orientiert sich die Abteilung QM eng an dem FTP (vgl. Abbildung 1.2). Für den Test sind hier speziell die höheren Teststufen Systemintegrationstest, Systemtest und Abnahmetest relevant, da diese von der Abteilung selber durchgeführt werden. Ausgehend von den Er-

## *1. Einleitung*

fahrungen der Abteilung im Bereich von Testaufwandsschätzungen wurde eine allgemeine Problemstellung abstrahiert.

Der Testaufwand für die Teststufen Systemintegrationstest, Systemtest und Abnahmetest wird in der Abteilung QM von den verantwortlichen Testmanagern im Rahmen einer Expertenschätzung ermittelt. Die Expertenschätzung wird durch den hinsichtlich mehrerer Merkmale heterogenen Kontext der Testprojekte erschwert. Zu den Merkmalen gehören u. a. die am Systemverbund beteiligten Organisationen, die Verteilung der Entwicklung auf unterschiedliche Standorte und die Qualifikationen der Projektbeteiligten [vW10, S. 18]. Basierend auf der Expertenschätzung wird die Planung der Ressourcen und der Kosten vorgenommen. Die ermittelten Kosten sind der Startpunkt für die Verhandlungen mit dem Auftraggeber über die notwendigen Maßnahmen im Test.

Die Testaufwandsschätzung wird bei QM nach dem Vier-Augen-Prinzip durchgeführt, d.h. es sind immer zwei Experten an einer Testaufwandsschätzung beteiligt. Dabei erstellt zunächst einer der Experten die Testaufwandsschätzung und der zweite Experte überprüft diese im Anschluss. Bei der Prüfung wird als erstes die Vollständigkeit der Testaufwandsschätzung validiert. Dazu müssen die zur Verfügung stehenden Informationen zu dem neuen Testprojekt durchgegangen und geprüft werden, ob die Komplexität der zu testenden Funktionalität vollständig in der Testaufwandsschätzung berücksichtigt ist. Im Folgenden wird diese Komplexität als Testkomplexität bezeichnet.

Da es keine einheitliche Aufnahme oder Beschreibung der Testkomplexität im Rahmen der Expertenschätzung gibt, ergeben sich hier unterschiedliche Interpretationen. Dies resultiert in Schwierigkeiten bei der Prüfung auf die Vollständigkeit der Testaufwandsschätzung. Verstärkt werden diese dadurch, dass die Darstellung der Informationen für Testprojekte selbst schon heterogen ist. Dies bedeutet, dass z. B. verschiedene (Modellierungs-)Sprachen zur Spezifikation der funktionalen Anforderungen eines Softwareproduktes eingesetzt werden. Die Schwierigkeiten bei der Prüfung der Vollständigkeit sind nur durch ein erhöhtes Maß an Kommunikation zwischen den zwei Experten behebbar. Was bleibt, ist die eingeschränkte Vergleichbarkeit zwischen Testaufwandsschätzungen hinsichtlich der Testkomplexität.

Weiterhin lassen sich grundlegende Kennzahlen, wie die Produktivität [Boe81, S. 641ff][Sne05, S. 4] nicht bestimmen, da keine vergleichbaren Werte für die

Testkomplexität existieren. Dadurch ist zum Einen mögliches Verbesserungspotential zur Erhöhung der Produktivität nur schwer identifizierbar. Zum Anderen sind die Ergebnisse von Veränderungen zur Erhöhung der Produktivität nicht belegbar.

Aus den beschriebenen Schwierigkeiten bei der Prüfung auf die Vollständigkeit der Testaufwandsschätzung hinsichtlich der Testkomplexität wird die erste Problemstellung abgeleitet:

### **Problemstellung 1:**

Formalisierung der Erfassung von Testkomplexität, um durch vergleichbare Werte den Interpretationsspielraum zu verringern.

Mit der Formalisierung existieren eindeutige Vorgaben für die Erfassung der Testkomplexität. Dadurch wird der Wert für die Testkomplexität durch einen zweiten Experten mit wenig Aufwand überprüfbar. Dies liegt vor allem an dem verringerten Interpretationsspielraum. Basierend auf diesem vergleichbaren Wert sind nun auch Kennzahlen für die Produktivität berechenbar und aussagekräftig.

Nach der Prüfung auf Vollständigkeit muss der überprüfende Experte die Werte für den Testaufwand auf Plausibilität überprüfen. Das beinhaltet die Prüfung, ob die Werte realistisch im Vergleich zu der Testkomplexität sind. Dafür ist die Formalisierung der Erfassung der Testkomplexität nicht ausreichend, da die stark variierenden Kontexte von Testprojekten mitberücksichtigt werden müssen.

Wie der Kontext des Testprojektes bewertet worden ist und welche Einflussfaktoren auf den Testaufwand (Testeinflussfaktoren) berücksichtigt worden sind, wird in den Expertenschätzungen nicht explizit ausgewiesen. Trotzdem enthält die Expertenschätzung implizit die Erfahrung in der technologischen/fachlichen Domäne und dem Kontext des Testprojektes des ersten Experten. Speziell wenn der zweite Experte kein Domänenexperte ist, wird für ihn die Schätzung nur eingeschränkt nachvollziehbar und überprüfbar sein. Damit ist viel Kommunikation zwischen den Experten für die Plausibilitätsprüfung notwendig. Ähnlich wie bei der Erfassung der Testkomplexität können mögliche Verbesserungspotentiale für Testprojekte unausgeschöpft bleiben, da z. B. einem Projektleiter die Möglichkeit genommen wird auf negative Einflussfaktoren zu reagieren.

## *1. Einleitung*

Aus den dargestellten Herausforderungen bei der Prüfung auf die Plausibilität des Testaufwands wird die zweite Problemstellung formuliert:

### **Problemstellung 2:**

Identifikation und Formalisierung von Testeinflussfaktoren, um eine einheitliche Beschreibung und Nutzung von Testeinflussfaktoren zu gewährleisten.

Über die Identifikation der Testeinflussfaktoren können diese explizit in Testaufwandsschätzungen verwendet werden. Die Formalisierung garantiert dabei die einheitliche Nutzung. Damit werden die Testaufwände in einer Testaufwandsschätzung für einen zweiten Experten bei der Prüfung auf die Plausibilität nachvollziehbar. Dem Projektleiter steht die Grundlage zur Verfügung, um auf negative Testeinflussfaktoren einzugehen und Verbesserungspotentiale für Testprojekte auszuschöpfen.

Die Testaufwandsschätzung wird bei der Expertenschätzung unter Verbindung der Testkomplexität mit den Testeinflussfaktoren durchgeführt. Allerdings wird kein formales Vorgehen ausgewiesen, wie aus der Verbindung der Testaufwand bestimmt wird. Damit ist der ausgewiesene Testaufwand nur eingeschränkt nachzuvollziehen. Daher wird die folgende Problemstellung betrachtet:

### **Problemstellung 3:**

Formalisierung der Verbindung der Testkomplexität mit den Testeinflussfaktoren, um die Nachvollziehbarkeit der Testaufwandsschätzung zu erhöhen.

Die Formalisierung der Verbindung ermöglicht die Erstellung von nachvollziehbaren Testaufwandsschätzungen, da der Zusammenhang zwischen den bestimmten Werten für die Testkomplexität sowie den Testeinflussfaktoren explizit dargestellt ist.

Zusammenfassend muss eine Methode zur Testaufwandsschätzung entwickelt werden, die von Testmanagern einheitlich verwendet wird und zu vergleichbaren Testaufwandsschätzungen führt. Dabei muss zunächst die Erfassung der Testkomplexität formalisiert werden. Danach sind Testeinflussfaktoren zu identifizieren und zu formalisieren. Abschließend ist die Verbindung zwischen der Testkomplexität und den Testeinflussfaktoren zu formalisieren.

## 1.2. Aufbau der Dissertation

In der vorliegenden Dissertation werden über Kapitel 2 Anforderungen an eine Lösung der in Abschnitt 1.1 beschriebenen Problemstellungen identifiziert. Im Anschluss wird unter Berücksichtigung dieser Anforderungen in Kapitel 3 der Lösungsansatz formuliert, der in den darauf folgenden Kapiteln 4 bis 6 ausgearbeitet wird.

Zunächst wird in Kapitel 4 ein Modell entwickelt, mit dem die Erfassung der Testkomplexität formalisiert wird. Dazu wird schrittweise anhand von Beispielen ein Modell aufgebaut und anschließend an Testprojekten der Abteilung QM des Unternehmens CRM IT evaluiert.

Darauf folgend werden in Kapitel 5 Testeinflussfaktoren identifiziert und ein Modell zur Formalisierung der Beschreibung und Nutzung für Testeinflussfaktoren entwickelt. Auch hier erfolgt als Abschluss die Evaluation über Testprojekte der Abteilung QM.

Anschließend wird in Kapitel 6 eine Methode beschrieben, mit der die Verbindung zwischen der Testkomplexität und den Testeinflussfaktoren formalisiert wird. Dazu werden die Modelle aus den Kapiteln 4 und 5 zusammengeführt. Ergänzend wird die Entwicklung einer Werkzeugunterstützung beschrieben, die Experten bei der Erstellung von Testaufwandsschätzungen unterstützen.

Die Dissertation wird mit einer Zusammenfassung in Kapitel 7 sowie eine Bewertung der Lösung und ein Ausblick auf mögliche weitere Schritte geschlossen. Dabei werden die definierten Problemstellungen aus Abschnitt 1.1 und die weiteren Anforderungen aus den Kapiteln 2 und 3 aufgegriffen und mit der entwickelten Lösung abgeglichen.



## **2. Einführung in Grundlagen und Identifikation von Anforderungen**

In diesem Kapitel werden grundlegende Konzepte, Begriffe und Methoden eingeführt, die den Rahmen dieser Dissertation darstellen. Basierend auf der Einführung werden gleichzeitig Anforderungen an die Lösung, der in Abschnitt 1.1 definierten Problemstellungen, identifiziert.

Da das Vorgehen in einem Softwareprojekt den Zeitpunkt für die Testaufwandschätzung und des Testens bestimmt, ist das Vorgehen ein wesentlicher Teil des Kontextes für ein Testprojekt. Daher wird in Abschnitt 2.1 eine Detaillierung des in Kapitel 1 vorgestellten V-Modells vorgenommen. Basierend darauf werden Anforderungen an die Lösung der Problemstellungen identifiziert.

Aus den Artefakten eines Softwareprojektes ist für ein Testprojekt die zu testende Funktionalität abzuleiten. Diese ist der Ausgangspunkt für die Bestimmung der Testkomplexität. Die Artefakte werden bei der Detaillierung des V-Modells identifiziert und für ein besseres Verständnis in Abschnitt 2.2 zusammen mit gängigen Techniken zur Anforderungsspezifikation erläutert. Über die Beschreibung der Artefakte und der hiermit zusammenhängenden Techniken zur Anforderungsspezifikation, werden weitere Anforderungen an die Lösung abgeleitet.

Wann und wie die Testaktivitäten durchgeführt werden, ist mitbestimmend für den Testaufwand. Daher werden aufbauend auf der Detaillierung des V-Modells in Abschnitt 2.3 die relevanten Testaktivitäten und Teststufen ergänzend zu Kapitel 1 beschrieben und die Begriffe der Qualitätssicherung für diese Dissertation definiert. Basierend darauf werden Anforderungen an die Lösung definiert.

## *2. Einführung in Grundlagen und Identifikation von Anforderungen*

Für ein gemeinsames Verständnis der Konzepte, Begriffe und Methoden in der Domäne Aufwandsschätzung werden diese in Abschnitt 2.4 eingeführt. Dabei wird zunächst eine übergeordnete Klassifizierung von Methoden zur Aufwandschätzung vorgestellt. Anschließend werden für jede Klasse etablierte Methoden diskutiert. Darauf basierend werden Eigenschaften der Methoden identifiziert, die sich bewährt haben und in der Lösung der Problemstellungen berücksichtigt werden müssen.

Geschlossen wird das Kapitel in Abschnitt 2.5 mit einer Zusammenfassung über die identifizierten Anforderungen.

### **2.1. Vorgehensmodelle für die Softwareentwicklung**

Aus dem Vorgehen in einem Softwareprojekt sind Anforderungen an die Lösung der Problemstellung aus Abschnitt 1.1 ableitbar, da das Vorgehen den Zeitpunkt der Testaufwandsschätzung und des Testens bestimmt. Um die Anforderungen definieren zu können, wird zunächst auf Vorgehensmodelle in der Softwareentwicklung eingegangen. Vertieft wird die Beschreibung von Vorgehensmodellen in Abschnitt 2.1.1 am V-Modell. Abschließend werden in Abschnitt 2.1.2 die aus den Vorgehensmodellen abgeleiteten Anforderungen diskutiert.

Das Vorgehen in einem Softwareprojekt wird über Vorgehensmodelle formalisiert. Allgemein kann ein Vorgehensmodell mit der nachfolgenden Definition beschrieben werden.

“The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use. Note: These activities may overlap or be performed iteratively. [...]“ [IEE91, S. 185]

Der erste Satz in dem Zitat beschreibt das übergeordnete Ziel Benutzerbedürfnisse in Software zu überführen. Die dafür notwendigen Aktivitäten werden in

## 2.1. Vorgehensmodelle für die Softwareentwicklung

dem zweiten Satz grob beschrieben und mit dem Hinweis versehen, dass die Aktivitäten iterativ oder überlappend ausgeführt werden können.

In bekannten Vorgehensmodellen aus der Literatur werden die Aktivitäten aufgegriffen und detailliert. Beispiele für Vorgehensmodelle der Softwareentwicklung sind das Wasserfallmodell [Boe81, S. 35ff], das Spiralmodell [Boe88], das V-Modell (XT) [WEI09, Boe79], der Rational Unified Process (RUP) [KK03] und Scrum [Coh10]. Zusammenfassungen über diese Vorgehensmodelle finden sich in [Dum01, S. 103ff], [SL10, S. 18ff] und [Spi11, S. 29ff]. In dem Standard ISO/IEC 12207 [IEE98a] werden allgemein Begriffe, Aktivitäten und deren Anordnung in Prozessen, die den gesamten Lebenszyklus einer Software abdecken, beschrieben.

### 2.1.1. Das V-Modell

Im Folgenden wird eine Variante des V-Modells vorgestellt, um in dieser Dissertation Status im Vorgehen der Softwareentwicklung für den Zeitpunkt der Testaufwandsschätzung benennen zu können. Das V-Modell wurde gewählt, da der Schwerpunkt in diesem Vorgehensmodell auf dem Verifizieren und Validieren des Softwareproduktes liegt. Dabei wird für jeden Status im Softwareprojekt definiert, welches Artefakt entsteht und verifiziert oder validiert wird. Diese Definition wird bezüglich der für diese Dissertation relevanten Artefakte in Abschnitt 2.2 mit Techniken zur Spezifikation detailliert.

In Abbildung 2.1 ist die Variante des V-Modells zu sehen. Die für diese Arbeit relevanten Artefakte und Teststufen sind blau hinterlegt. Abweichend von der gängigen Darstellung werden hier keine Aktivitäten gezeigt, sondern die Status im Projekt, die sich nach dem Abschluss der gleichlautenden Aktivität ergeben. Diese Status werden in die Phasen Anforderungsphase, Designphase und Entwicklungsphase eingeteilt, um den Zeitpunkt für die Durchführung der Testaufwandsschätzung unter Berücksichtigung der zur Verfügung stehenden Informationen beschreiben zu können. Dabei umfasst die Anforderungsphase alle Status, die zu dem Artefakt Anforderungsdefinition führen. Die Designphase enthält die Status bei denen eine Spezifikation als Ergebnis entsteht. Die Entwicklungsphase umfasst die restlichen Status.

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

Der Pfad auf der linken Seite in Abbildung 2.1 zeigt die Entwicklung des Softwareproduktes von der Beauftragung des Projektes bis zur Umsetzung (Konstruktion). Dabei sind die Artefakte, die bei dem Erreichen eines Projektstatus vorliegen, unter dem Status vermerkt. Auf der rechten Seite der Abbildung ist der Pfad von der Umsetzung zur Abnahme zu sehen (Validierung<sup>1</sup>). Die Teststufen, die unter dem Status auf dem Pfad auf der rechten Seite stehen, müssen für das Erreichen des Status erfolgreich abgeschlossen sein. Die Pfeile innerhalb des V-Modells kennzeichnen, welcher Teil der Konstruktion des Softwareproduktes durch welche Teststufe validiert wird.

Das V-Modell sieht neben der Validierung der Entwicklungsergebnisse Schritte zur Verifikation<sup>2</sup> vor [Boe79]. Diese sind in Abbildung 2.1 auf der linken Seite zu sehen. Hier wird nachgewiesen, dass das Ergebnis nach Abschluss einer Aktivität die Anforderungen aus den eingehenden Artefakten berücksichtigt. Ein Beispiel ist, dass die funktionale Spezifikation die Anforderungsdefinition vollständig und korrekt abdeckt.

Vor dem Einsatz der oben genannten Vorgehensmodelle müssen diese dem jeweiligen Unternehmenskontext angepasst und detailliert werden, da die meisten Definitionen von Vorgehensmodellen nur einen abstrakten Prozess und Handlungsanweisungen beinhalten. Für die Entwicklung eigener Vorgehensmodelle hat die *Object Management Group* (OMG) das *Software & Systems Process Engineering Meta-Model* (SPEM) veröffentlicht [Obj08]. Eine weitergehende Arbeit behandelt die systematische Entwicklung von passgenauen Vorgehensmodellen [Sau11].

### 2.1.2. Anforderungen an die Lösung

Kaufmännische Angebote für das Testen eines Softwareproduktes, die eine Testaufwandsschätzung voraussetzen, müssen oft sehr früh im Gesamtprojekt abgegeben werden. Zu diesem Zeitpunkt stehen meistens noch keine detaillierten

---

<sup>1</sup> „Validation - to establish the fitness or worth of a software product for its operational mission.“ [Boe79] Diese Aussage anders formuliert bedeutet, dass geprüft wird, ob das richtige System entwickelt wurde.

<sup>2</sup> „Verification - to establish the truth of the correspondence between a software product and its specification.“ [Boe79] Diese Aussage anders formuliert bedeutet, dass geprüft wird, ob das System richtig entwickelt wurde.

## 2.1. Vorgehensmodelle für die Softwareentwicklung

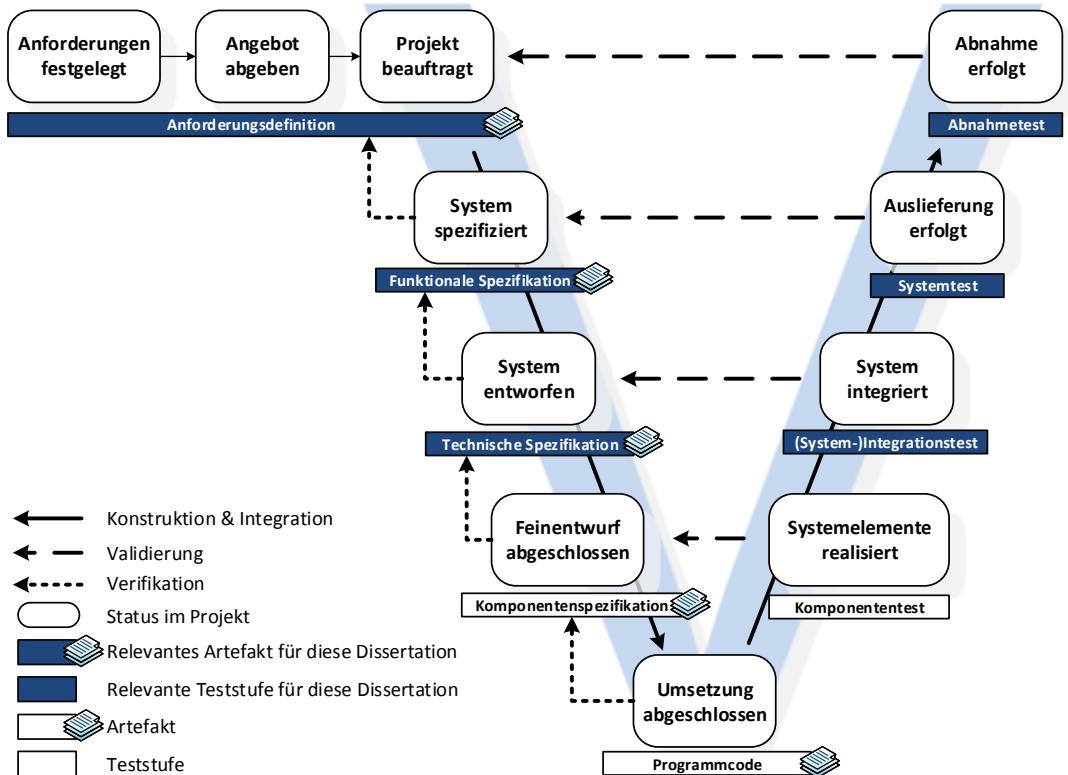


Abbildung 2.1.: V-Modell nach [SL10, S. 33]

Informationen zum Softwareprodukt oder den Rahmenbedingungen des Projektes zur Verfügung. Genauso existiert der umgekehrte Fall, wo erst sehr spät in einem Projekt ein Angebot für den Test angefordert wird. Hier existieren je nach Fortschritt des Projekts detaillierte Spezifikationen des Softwareproduktes und die Rahmenbedingungen des Projektes sind bekannt. Da somit ein Angebot zu jedem möglichen Zeitpunkt angefordert werden kann, muss auch die Testaufwandsschätzung zu jedem beliebigen Zeitpunkt in einem Softwareprojekt ausführbar sein und die zur Verfügung stehenden Informationen berücksichtigen. Damit ergibt sich folgende Anforderung:

### Anforderung 1:

Beliebiger Zeitpunkt für die Durchführung der Testaufwandsschätzung

Auf Basis einer früh im Softwareprojekt abgegebenen Testaufwandsschätzung ist eine Planung oft nur unzureichend möglich, da die Aufwandsschätzung aufgrund fehlender Informationen sehr ungenau ist. Beispielsweise kann die Anforderungsspezifikationen noch nicht fertig gestellt sein oder der Kontext des Testprojektes ist noch nicht vollständig bekannt. Daher muss die Testaufwandsschätzung zu

## *2. Einführung in Grundlagen und Identifikation von Anforderungen*

einem späteren Zeitpunkt in einem Projekt wiederholt werden, wenn mehr Informationen zur Verfügung stehen. Diese neue Testaufwandsschätzung stellt dann eine geeignete Grundlage für die Planung dar. Damit Aufwände zur Erstellung der Testaufwandsschätzung vermieden werden, muss die erste Schätzung aktualisierbar sein. Dies wird mit der nachfolgenden Anforderung beschrieben:

### **Anforderung 2:**

#### Aktualisierbare Testaufwandsschätzung

Zusammenfassend muss die Testaufwandsschätzung zu einem beliebigen Zeitpunkt in einem Softwareprojekt durchgeführt werden können, um jederzeit ein kaufmännisches Angebot für den Test erstellen zu können. Da speziell zu frühen Zeitpunkten wenig und ungenaue Informationen zur Verfügung stehen, müssen Testaufwandsschätzungen aktualisierbar sein. Damit werden Folgeaufwände für das wiederholte Überarbeiten der Testaufwandsschätzungen beim Vorliegen neuer Informationen vermieden.

## **2.2. Anforderungsspezifikation**

Im Folgenden wird zunächst in Abschnitt 2.2.1 auf die Rollen bei der Anforderungsspezifikation eingegangen, um diese bei der Betrachtung des Testprojektkontextes entsprechend berücksichtigen zu können. Die Rollen müssen bei einer Testaufwandsschätzung berücksichtigt werden, da die Qualifikationen des Personals mit diesen Rollen ein Indikator für die Qualität der Artefakte ist. Dabei werden die Rollen betrachtet, die für die Erstellung der in Abbildung 2.1 blau hinterlegten Artefakte verantwortlich sind, da diese eine Eingabe für die in dieser Dissertation relevanten Teststufen sind.

Nach der Betrachtung der Rollen werden in Abschnitt 2.2.2 Arten der Anforderungsspezifikation vorgestellt, um zu unterstützende Modellierungssprachen bei der Erfassung der Testkomplexität zu identifizieren. Dabei werden Arten der Anforderungsspezifikation wieder nur für die blau hinterlegten Artefakte in Abbildung 2.1 betrachtet. Die sich hieraus ergebenden Anforderungen werden in Abschnitt 2.2.3 zusammengefasst.

### 2.2.1. Rollen bei der Anforderungsspezifikation

Die Rollen für die Erstellung von Anforderungsspezifikationen sind in Abbildung 2.2 als Use Case Diagramm dargestellt. Für die Bezeichnungen der Rollen wurden die Bezeichnungen aus der Definition des Vorgehensmodells V-Modell XT übernommen [WEI09]. Die Benennung der Artefakte wurde dagegen gegenüber dem V-Modell XT geändert, da sie in der Literatur zum Thema Test abweichend bezeichnet werden und diese Dissertation auf den Begriffen der Testdomäne beruhen soll [SL10, S. 42]. Im V-Modell XT stellt die Anforderungsdefinition das Lastenheft, die funktionale Spezifikation das Pflichtenheft und die technische Spezifikation die Systemspezifikation dar.

Das erste Artefakt in Abbildung 2.2 ist nach dem V-Modell in Abbildung 2.1 die Anforderungsdefinition. Verantwortlich für die Erstellung der Anforderungsdefinition sind in der Regel Personen in der Rolle Anforderungsanalytiker des Auftraggeber (AG). Das zweite Artefakt, die funktionale Spezifikation, wird von einem Anforderungsanalytiker des Auftragnehmers (AN) unter Mitwirkung eines Systemarchitekten erstellt. Von dem Systemarchitekten wird anschließend die technische Spezifikation erstellt.

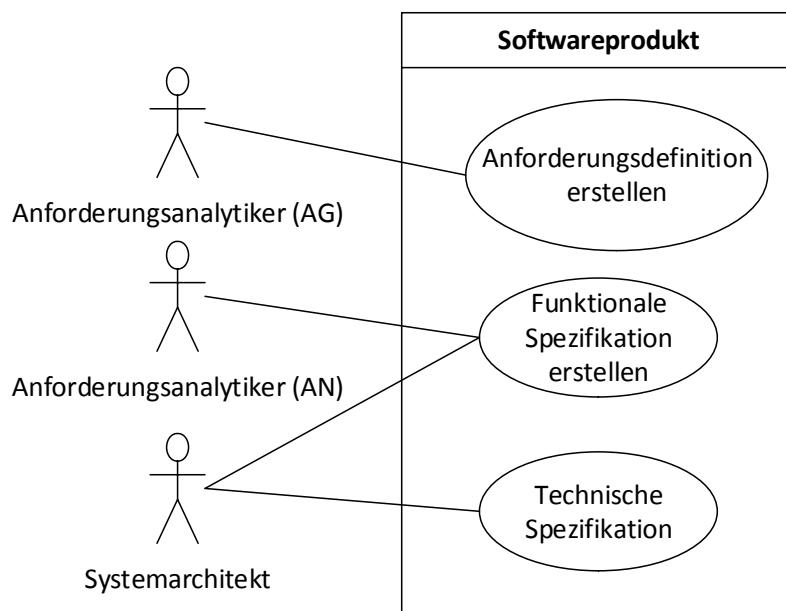


Abbildung 2.2.: Ausschnitt über verantwortliche Rollen für die Erstellung der Artefakte nach [WEI09]

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

Die Sicht in Abbildung 2.2 auf die beteiligten Rollen ist eine starke Vereinfachung, da noch viele weitere Rollen bei der Erstellung der Artefakte mitwirken. Dies können Vertreter des Einkaufs, der Rechtsabteilungen und der Qualitätssicherung sein. Jedoch werden die Fähigkeiten der Personen, die in einem Projekt die in Abbildung 2.2 abgebildeten Rollen einnehmen, als stark mitbestimmend für die Qualität des Softwareproduktes aus Sicht des Anwenders gesehen, da sie verantwortlich für die erstellten Artefakte sind. Ohne eine valide Anforderungsdefinition oder bei einer falschen Ableitung der technischen Spezifikation aus der fachlichen Spezifikation ist eine zufriedenstellende Qualität für den Anwender des Softwareproduktes nur mit hohem Aufwand u. a. in den in Abbildung 2.1 blau hinterlegten Teststufen erreichbar. Die Fehler die dort gefunden werden sind in der Regel am aufwands- und kostenintensivsten zu beheben [Boe81, S. 39][BB01][Res02, S. 5-4].

### 2.2.2. Arten der Anforderungsspezifikation

Die Arten der Anforderungsspezifikation sind bei der Formalisierung der Erfassung von Testkomplexität zu berücksichtigen (vgl. Problemstellung 1 in Abschnitt 1.1). Während der Erstellung einer Anforderungsspezifikation werden fachliche Anforderungen natürlichsprachlich formuliert oder mithilfe einer Modellierungssprache aufgenommen. Die dabei entstehenden Artefakte sind die Grundlage für die Ermittlung der zu testenden Funktionalität und damit für die Schätzung der Testkomplexität. Zur Identifikation von zu berücksichtigenden Arten der Anforderungsspezifikation werden im Folgenden zunächst natürlichsprachliche Arten und danach Arten unter Verwendung einer visuellen Modellierungssprache vorgestellt.

In dem Standard ISO/IEC/IEEE 29148 [IEE11b, S. 9] wird die Anforderungsspezifikation mit “Transformation of needs into requirements” beschrieben und Hinweise zur natürlichsprachlichen Formulierung der Anforderungen gegeben. Dabei werden der Formulierung durch eine vorgegebene Syntax und Begriffe enge Grenzen gesetzt (vgl. auch [PR11, S. 49]). Zur Aufnahme der Anforderungen in natürlichsprachlicher Form bietet sich eine Vorlage wie die *Volere Shell* an [RR06, S. 15]. Eine Bedingung an die so formulierten Anforderungen ist, dass

die Anforderungen nachprüfbar sind. Dabei werden im optimalen Fall für jede Anforderung die Kriterien für die Abnahme durch den Kunden dokumentiert.

Neben der natürlichsprachlichen Anforderungsdefinition kann die Definition der erwarteten Funktionalität auch mit einer visuellen Modellierungssprache wie Use Cases oder BPMN erfolgen. Die Spezifikation mit Use Cases ist zweigeteilt in *Unified Modeling Language* (UML<sup>3</sup>) Use Case Diagramme und die Use Case Definition. Use Case Diagramme bieten eine schnelle Übersicht über die Funktionen eines Softwareproduktes und deren Abhängigkeiten aus Benutzersicht [RQ07, S. 238] [Obj11a, S. 597]. Eine Funktion ist dabei ein Use Case. Zentral in den Diagrammen ist ein begrenzter Teil des Softwareproduktes mit den zugehörigen Use Cases und den Beziehungen der Use Cases zu Akteuren außerhalb des begrenzten Teils. Die Use Case Definition ist eine Ablaufbeschreibung in natürlichsprachlicher oder visueller Form für jeden Use Case, die die einzelnen Aktionen für die Erfüllung des Use Cases miteinander verknüpft. Für die Erstellung von Use Case Definitionen in natürlicher Sprache sollten Vorgaben zur Strukturierung befolgt werden, wie von Cockburn in [Coc03] formuliert. Eine Alternative in visueller Form ist der Einsatz von UML Aktivitäten Diagrammen [Obj11a, S. 303].

Use Cases können auf verschiedenen Abstraktionsebenen für die Beschreibung der Funktionalität eingesetzt werden [Coc03, S. 191]. Eine Möglichkeit ist, dass die Anforderungsdefinition aus einem Use Case Diagramm mit zusätzlichen Anmerkungen zu den einzelnen Use Cases besteht. Mit dieser Form der Anforderungsdefinition kann schon zu Beginn des Projektes transparent festgehalten werden, welche Akteure mit dem Softwareprodukt kommunizieren und wo die Grenzen des zu entwickelnden Produkts liegen. Die funktionale Spezifikation in Abbildung 2.1 kann anschließend mit Use Cases Definitionen zu den in der Anforderungsdefinition festgehaltenen Use Cases umgesetzt werden. Damit wird deutlich, wie der Benutzer mit dem Softwareprodukt interagiert und welches Verhalten des Produktes von außen erwartet wird.

---

<sup>3</sup>“The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of softwarebased systems as well as for modeling business and similar processes.“ [Obj11a, S. 1] Um die Werkzeuge realisieren zu können, wird eine abstrakte Syntax formal definiert, die Semantik ausführlich beschrieben, eine konkrete Syntax vorgeschlagen und ein Format für den Austausch von Modellen zwischen Werkzeugen unterschiedlicher Hersteller festgelegt. Mit diesem Konzept ist die UML heute die verbreitetsten Notationsform für die Analyse und den Entwurf von Softwaresystemen [RQ07, S. 11].

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

Die technische Spezifikation kann auch mit Use Case Definitionen zu den Use Cases aus der Anforderungsdefinition umgesetzt werden, wobei hier ein Wechsel der Benutzersicht zur Produktsicht erfolgt. Bei dem Wechsel der Perspektive können auch Use Cases aus der Anforderungsdefinition wegfallen oder weitere aus der Sicht des Produktes hinzukommen. Allgemein besteht eine  $n:m$ -Beziehung zwischen den Use Cases der funktionalen und technischen Spezifikation.

Die visuelle Modellierungssprache *Business Process Model and Notation* (BPMN) hat die Bereitstellung einer Notationsform für Geschäftsprozesse zum Ziel, die von allen Projektbeteiligten möglichst gleich verstanden wird [Obj11b, S.,1]. Speziell die Lücke zwischen der Anforderungsdefinition und der funktionalen Spezifikation auf der einen Seite und der technischen Spezifikation auf der anderen Seite soll verkleinert werden. Ein Geschäftsprozess wird als eine Abfolge von Aktivitäten aufgefasst. Diese Aktivitäten können *Tasks*, *Sub-Processes* oder Aufrufe von wiederverwendbaren Aktivitäten oder Prozessen sein [Obj11b, S. 151]. Die BPMN bietet für die Unterscheidung von *Tasks* verschiedene Vorgaben an, wie z.B. einen *User Task*<sup>4</sup> oder *Service Task*<sup>5</sup>.

BPMN ermöglicht die Darstellung von Geschäftsprozessen in unterschiedlichen Abstraktionsebenen, wobei die Ebenen direkt miteinander verbunden sind [Obj11b]. Dies wird erreicht, indem ein Geschäftsprozess, der in der Anforderungsdefinition aufgenommen worden ist, bei der Erstellung der fachlichen Spezifikation aufgegriffen und detailliert wird. Dies kann z. B. über die Ersetzung einer *Task* in dem Geschäftsprozess durch einen *Sub-Process* geschehen, der die Interaktion mit dem Softwareprodukt genauer beschreibt. Genauso kann die Detaillierung für die technische Spezifikation erfolgen, da die BPMN auch dafür die notwendigen Elemente zur Modellierung bereitstellt.

---

<sup>4</sup>“A **Service Task** is a **Task** that uses some sort of service, which could be a Web service or an automated application.”[Obj11b, S. 158]

<sup>5</sup>“A **User Task** is a typical “workflow” **Task** where a human performer performs the **Task** with the assistance of a software application and is scheduled through a task list manager of some sort.”[Obj11b, S. 158]

### **2.2.3. Anforderungen an die Lösung**

In Abschnitt 2.2.1 wurden die beteiligten Rollen bei der Erstellung von Anforderungsspezifikationen erläutert. Die Qualifikation der Personen, die diese Rollen ausführen, ist mitbestimmend für die Qualität der Anforderungsspezifikationen in einem Softwareprojekt. Da diese Artefakte die Grundlage für den Test sind, müssen Informationen über die Qualifikation der Personen mit den Rollen Anforderungsanalytiker (AG), Anforderungsanalytiker (AN) und Softwarearchitekt in einer Testaufwandsschätzung berücksichtigt werden. Daraus folgt die Anforderung:

#### **Anforderung 3:**

Berücksichtigung der Qualifikationen von Personen in den Rollen Anforderungsanalytiker des Auftraggebers sowie des Auftragnehmers und des Softwarearchitekten

Die Beschreibung der Arten von Anforderungsspezifikationen in Abschnitt 2.2.2 hat gezeigt, dass es eine Vielzahl von Möglichkeiten zur Anforderungsspezifikation gibt. Diese können in dieser Dissertation nicht alle unmittelbar bei der Lösung der Problemstellungen aus Abschnitt 1.1 berücksichtigt werden. Daher wird aufgrund der Verbreitung der Standardmodellierungssprachen der UML folgende Anforderung festgehalten:

#### **Anforderung 4:**

Unterstützung von Standardmodellierungssprachen der UML

Um neben den Standardmodellierungssprachen der UML weitere Modellierungssprachen nutzen zu können, muss eine Möglichkeit zur Anbindung dieser weiteren Sprachen bereitgestellt werden. Daraus ergibt sich die Anforderung:

#### **Anforderung 5:**

Möglichkeit zur Anbindung von weiteren Modellierungssprachen

Zusammenfassend sind bei der Formalisierung der Erfassung von Testkomplexität schwerpunktmaßig die Standardmodellierungssprachen der UML zu berücksichtigen (vgl. Problemstellung 1 in Abschnitt 1.1). Daneben soll die Möglichkeit zur Anbindung alternativer Modellierungssprachen geboten werden.

## 2.3. Dynamische Analyse: Testen

Für die Entwicklung der Lösung ist ein gemeinsames Verständnis über die Begriffe Qualität, Testen, die zu schätzenden Testaktivitäten und Teststufen notwendig. Daher wird in diesem Abschnitt zunächst allgemein auf den Begriff Qualität eingegangen und der Begriff Testen definiert. Danach werden die für diese Dissertation relevanten Testaktivitäten (vgl. Abschnitt 2.3.1) und Teststufen (vgl. Abschnitt 2.3.2) beschrieben. Auf Basis dieser Beschreibungen werden abschließend in Abschnitt 2.3.3 Anforderungen an die Lösung zusammengefasst.

In der Einleitung in Kapitel 1 wurde der Begriff Qualität eines Softwareproduktes mit dem Grad der Erfüllung der Anforderungen an das Softwareprodukt definiert. Dabei umfasst der Begriff Anforderung neben den funktionalen Anforderungen, die mit den im vorherigen Abschnitt beschriebenen Spezifikationstechniken aufgenommen werden, auch Qualitätsanforderungen.

Die Qualitätsanforderungen können den in der Abbildung 2.3 zu sehenden Qualitätsmerkmalen in einem Qualitätsmodell zugeordnet werden. Für jedes Merkmal sind in den Standards ISO 9126 [ISO01] bzw. dem Nachfolger ISO 25010 [ISO11b] Teilmerkmale zur weiteren Unterscheidung der Anforderungen aufgeführt. Dies ist an dem Beispiel des Merkmals *Funktionalität* in Abbildung 2.3 zu sehen.

Zur Überprüfung der Teilmerkmale werden in den Standards ISO 9126 und ISO 25010 Metriken definiert. Die Metriken können sich auf das Produkt von außen (externe Metriken) oder von innen beziehen (interne Metriken). Ein Beispiel für eine externe Metrik bzgl. des Teilmerkmals *Erfüllung der Benutzeranforderungen* ist die *Functional implementation completeness* Metrik. Sie gibt das Verhältnis von implementierten zu fehlenden Funktionen eines Softwareproduktes aus. Dieses Verhältnis muss bei der Formulierung einer Qualitätsanforderung für dieses Teilmerkmal festgehalten werden.

Die Qualitätsanforderungen bzgl. des Merkmals Funktionalität in Abbildung 2.3 sind speziell für den Bereich der analytischen Qualitätssicherung relevant, da die Erfüllung der Anforderungen aus Benutzersicht oft nur durch die dynamische Analyse (*Testen*) überprüft werden kann. Dabei ist das Testen kein konstruktives Verfahren, das Qualität in ein Softwareprodukt einbringt, sondern ein Pro-

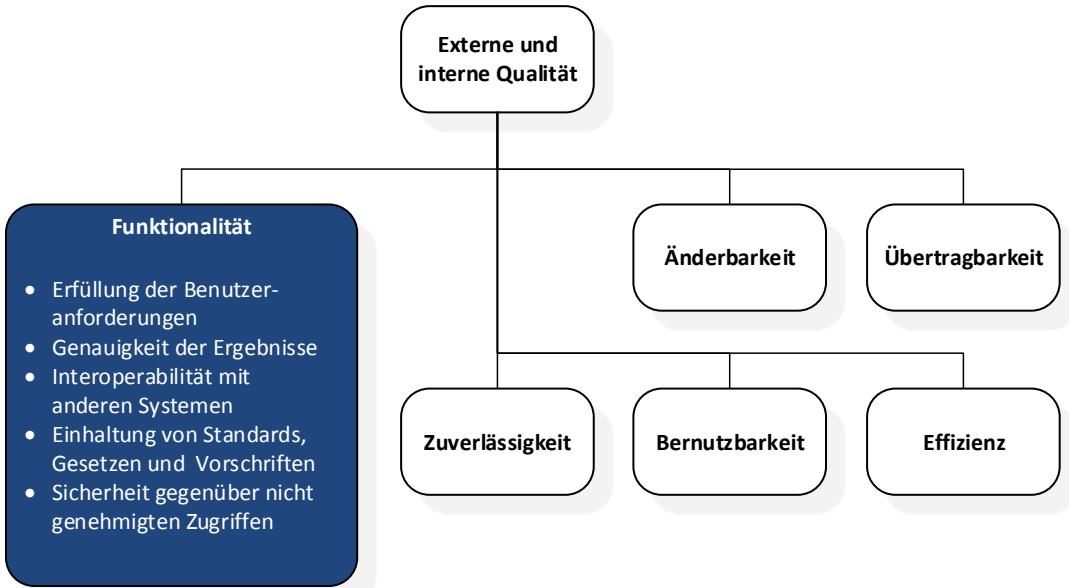


Abbildung 2.3.: Qualitätsmodell nach ISO 9126 [ISO01] bzw. dem Nachfolger ISO 25010 [ISO11b]

zess, um Qualität bzgl. definierter Qualitätsanforderungen nachzuweisen. Testen kann wie folgt definiert werden:

„Testen ist ein Prozess, der Einsicht in und Auskunft über die Qualität eines Objektes und die damit verbundenen Risiken liefert.“ [van11, S.9]

Beim Testen wird das Sollverhalten aus der Anforderungsspezifikation mit dem Istverhalten der Software verglichen. Wenn das Verhalten nicht übereinstimmt, ist eine Fehlerwirkung (*failure*) festgestellt worden. Die Ursache für das Auftreten der Fehlerwirkung kann ein Fehlzustand (*fault*) im Programmcode sein, der durch eine Fehlhandlung (*error*) eines Programmierers entstanden ist.

### 2.3.1. Testaktivitäten

Basierend auf der Testaufwandsschätzung wird die Personal- und Projektplanung vorgenommen, d.h. wie viele Personen werden in welcher Rolle in einem Testprojekt benötigt. Die Rollen sind Testmanager, Testdesigner und Tester mit den gleichlautenden Testaktivitäten Testmanagement, Testdesign und Testdurchführung. Diesen Rollen können, wie in Abbildung 1.2 gezeigt und in der

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

Einleitung in Kapitel 1 erläutert, Aktivitäten im FTP zugeordnet werden. Für eine Abgrenzung der Rollen wird in diesem Abschnitt jede Rolle bzw. die entsprechende Aktivität erläutert.

Rolle	Beschreibung
Testmanager	Planung der Ressourcen (Personal, Hardware und Software) Kontrolle des Projektstatus (Fortschritt und Testergebnisse) Reaktion bei Abweichungen vom Soll-Projektstatus Erstellung von Dokumentation (Testplan, Testkonzept,...)
Testdesigner	Analyse der Testbasis Identifikation von Testobjekten Erstellung von logischen und konkreten Testfällen
Tester	Ausführung der Testfälle Dokumentation von Fehlerwirkungen Protokollierung von Testergebnissen

Tabelle 2.1.: Beschreibung der Rollen Testmanager, Testdesigner und Tester

Eine Übersicht über die Testaktivitäten bietet die Tabelle 2.1. Die Aktivität Testmanagement wird von der Rolle Testmanager ausgeführt. Zu den Aufgaben des Testmanagers gehören die Planung und Steuerung des Testprojektes. Der erste Schritt in der Planung eines Testprojektes ist die Festlegung einer Teststrategie und die darauf basierende Testaufwandsschätzung. Das Hauptergebnis ist eine Zeitplanung und der Ressourcenbedarfsplan [IEE86, S. 5]. Dies wird in einem Dokument festgehalten, das nach dem IEEE Standard 829 [IEE08, S. 42] strukturiert ist. In dem Dokument werden neben der Zeitplanung und dem Ressourcenbedarf die Eingangs- und Ausgangskriterien für die Testphase, die Dokumentationsanforderungen, die zu testende Funktionalität, die einzusetzenden Methoden und die Werkzeuge festgehalten.

Während der Durchführung des Testprojektes kontrolliert der Testmanager fortlaufend den Status des Projektes, wie den Testfortschritt und die Testergebnisse [Spi11, S. 129ff]. Wenn bei der Kontrolle Abweichungen vom Soll-Status des Testprojektes deutlich werden, muss der Testmanager entsprechend reagieren. Ein Beispiel ist, dass eine Fehlerwirkung bei einer Funktion im Softwareprodukt festgestellt worden ist. Bis der Fehlzustand aufgedeckt worden und behoben ist, können mehrere Tage vergehen. Daher muss der Testmanager den Testern,

### 2.3. Dynamische Analyse: Testen

die für das Testen dieser Funktion vorgesehen gewesen sind, andere Aufgaben zuweisen und spätere erneute Tests für die Funktion einplanen.

Die Aktivität Testdesign wird von der Rolle Testdesigner ausgeführt. Zu den Aufgaben gehört die Analyse der Testbasis und die Identifikation von Testobjekten [IEE86, S. 7][IEE08, S. 50][SL10, S. 23]. Die Testbasis ist abhängig von der Teststufe und den verfügbaren Artefakten (vgl. Artefakte im V-Modell in Abschnitt 2.1.1). Für einen Systemtest ist beispielsweise die funktionale Spezifikation die Testbasis. Dabei kann die funktionale Spezifikation aus einer Vielzahl von Dokumenten bestehen, die von einem Testdesigner analysiert werden müssen. Auf Basis dieser Dokumente werden funktionale Einheiten abgegrenzt, die dann jeweils ein Testobjekt bilden.

Nach der Identifikation der Testobjekte erstellt ein Testdesigner in zwei Schritten die jedem Testobjekt zugehörigen Testfälle [SL10, S. 24]. Zunächst wird der logische Ablauf des Tests definiert und anschließend werden die konkreten Testdaten festgelegt. Ein Beispiel für eine logische Definition ist, dass eine Funktion mit einem Wert kleiner als Zwanzig aufgerufen wird und wahr zurückgibt. Ein möglicher konkreter Testfall ist der Aufruf der Funktion mit der Zahl Zehn und dem erwarteten Ergebnis wahr. In dem IEEE Standard 829 [IEE08, S. 52ff] werden Hinweise für die Dokumentation eines Testfalls gegeben. Die Ableitung der zu einem Testobjekt zugehörigen Testfälle erfolgt über verschiedene Testmethoden, wie z.B. der Äquivalenzklassenbildung [SL10, S. 114ff] und der Grenzwertanalyse [SL10, S. 125ff].

Die Aktivität Testdurchführung wird von der Rolle Tester ausgeführt. Zu den Aufgaben eines Testers gehört die Ausführung der vom Testdesigner erstellten konkreten Testfälle, die Protokollierung und die Bewertung der Testergebnisse. Die Reihenfolge der Testfälle und die Form der Protokollierung werden den Testern vom Testmanager vorgegeben. Wichtigste Ziele bei der Protokollierung sind die Nachvollziehbarkeit und die Reproduzierbarkeit der Ergebnisse [SL10, S. 28]. Nur damit kann bei dem Eintritt einer Fehlerwirkung die betroffene Funktion in dem Softwareprodukt durch einen Entwickler analysiert und der Fehlzustand beseitigt werden. In dem IEEE Standard 829 [IEE08, S. 58ff] wird eine Vorlage für die Protokollierung von Testergebnissen vorgestellt.

### 2.3.2. Teststufen

Die in Abschnitt 2.3.1 beschriebenen Testaktivitäten wiederholen sich in jeder Teststufe, die in Abbildung 2.1 zum V-Modell dargestellt sind. Dabei unterscheidet sich die Ausführung der Testaktivitäten leicht, da das Testziel und die Testobjekte der Teststufen unterschiedlich sind. Daher werden in diesem Abschnitt die für diese Dissertation relevanten Teststufen erläutert.

Im (System-)Integrationstest werden Schnittstellen von Teilsystemen geprüft. Das Ziel des Integrationstests ist, dass Fehlzustände in der Schnittstelle und im Zusammenspiel der Teilsysteme aufgedeckt werden [GTB10]. Dies können funktionale Fehlzustände bei der Verarbeitung der übergebenen oder der erzeugten Daten sein. Die Testbasis für diese Teststufe besteht aus der technischen Beschreibung der zu testenden Schnittstellen, die in der technischen Spezifikation enthalten ist [SL10, S. 55]. Für die Ausführung sind Testtreiber zur Erzeugung der Eingabedaten erforderlich. Dies kann z. B. eine Datei in einem bestimmten Format sein, die jeweils neu für die Testausführung zusammengestellt werden muss.

In der Teststufe Systemtest wird das integrierte Gesamtsystem geprüft. Das Ziel des Systemtests ist die Validierung durch Tester, ob das Gesamtsystem die Anforderungen aus Kundensicht erfüllt [SL10, S. 60]. Die Testbasis besteht daher primär aus der funktionalen Spezifikation. Weitere Bestandteile der Testbasis können fachliche Abnahmekriterien des Kunden oder auch die technische Spezifikationen des Systems sein.

Beim Abnahmetest wird genauso wie beim Systemtest das integrierte Gesamtsystem geprüft. Das Ziel ist, dass die Kunden bzw. die Anwender aus ihrer Sicht das System beurteilen [SL10, S. 63]. Die Testbasis enthält die vertraglich definierten Abnahmekriterien. Diese werden durch den Kunden überprüft, um zu beurteilen, ob das System abgenommen werden kann.

### 2.3.3. Anforderungen an die Lösung

Das Ergebnis der Testaufwandsschätzung soll den Testmanager bei der Personal- und Projektplanung unterstützen. Dazu muss laut dem Verfahren zum *Test*

*Process Improvement* (TPI<sup>6</sup>)[van11, S. 81] in der einfachsten Form der benötigte Umfang an (Personal-)Ressourcen pro (Test-)Aktivität prognostiziert werden.

Speziell für die Einplanung von hochqualifiziertem Testpersonal ist die Prognose des Umfangs an Ressourcen pro Testaktivität obligatorisch. Ohne eine frühzeitige Anfrage für die benötigten Ressourcen kann nicht sichergestellt werden, dass das Personal rechtzeitig zum Projektstart zur Verfügung steht. Daher wird folgende Anforderung abgeleitet:

**Anforderung 6:**

Aufschlüsselung des Testaufwands nach Testaktivitäten

Die Ausführung der Testaktivitäten unterscheidet sich in den für diese Dissertation relevanten Teststufen, da das Testziel und die Testobjekte der Teststufen unterschiedlich sind. Hieraus ergibt sich die Anforderung:

**Anforderung 7:**

Unterstützung der Teststufen (System-)Integrationstest, Systemtest und Abnahmetest

Zusammenfassend muss das Ergebnis der Testaufwandsschätzung aufgeschlüsselt nach den Testaktivitäten ausgegeben werden, um eine Personal- und Projektplanung zu unterstützen. Da die Testziele und Testobjekte der Teststufen unterschiedlich sind, wird explizit die Unterstützung der Teststufen (System-)Integrationstest, Systemtest und Abnahmetest gefordert.

## 2.4. Einführung in die Domäne Aufwandsschätzung

Die Domäne Aufwandsschätzung für Softwareprojekte hat eine lange Historie, die schon vor den 70er Jahren des 20. Jahrhunderts beginnt. Dies zeigen Zusammenfassungen über Methoden zur Aufwandsschätzung in Softwareprojekten [Boe81, BAC00, BF04, Sne05, McC06, ED07, Hum11]. Genauso wie für Testprojekte sind für Softwareprojekte realistische Aufwandsschätzungen verpflichtend. Nur auf der Grundlage der Aufwandsschätzung kann angemessen entschieden

---

<sup>6</sup>TPI ist ein von der Firma Sogeti entwickeltes Verfahren zur Bewertung der Testprozessreife in Unternehmen. Dabei wird anhand eines Fragenkataloges eine Einordnung der Reife des Testprozesses in bestimmten Kernbereichen vorgenommen. Darauf basierend werden konkrete Verbesserungen vorgeschlagen. Ein Kernbereich ist die Kostenschätzung und Planung.

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

werden, wie die Zeitplanung des Softwareprojektes auszusehen hat und wie viele Ressourcen benötigt werden.

Da Testprojekte ein Teil eines Softwareprojektes sind, können Erkenntnisse aus dieser Domäne auf Testaufwandsschätzungen übertragen und Anforderungen an die Lösung formuliert werden. Dazu wird in Abschnitt 2.4.1 zunächst eine Klassifizierung von Methoden zur Aufwandsschätzung vorgestellt. Anschließend werden exemplarisch einige Repräsentanten der Klassen in den Abschnitten 2.4.2 bis 2.4.4 erläutert. Basierend auf der Klassifizierung und der Erläuterungen der Repräsentanten werden in Abschnitt 2.4.5 Anforderungen an die Lösung definiert.

### 2.4.1. Klassifizierung von Methoden zur Aufwandsschätzung

In Abbildung 2.4 ist eine Übersicht über die Klassifizierung von Methoden zur Aufwandsschätzung dargestellt [Boe81, S. 329][Wol74, S. 618]. Unterschieden wird zwischen der Klasse der eigenständigen Methoden und den Klassen zur weitergehenden Unterscheidung der eigenständigen Methoden. Die Klassen zur weitergehenden Unterscheidung charakterisieren die eigenständigen Methoden näher, z.B. kann eine Expertenschätzung zuerst *Bottom-Up* erfolgen und anschließend der Aufwand mittels Analogie-Schätzung feinjustiert werden.

Was die in Abbildung 2.4 gezeigten Methoden beinhalten und wie sie genutzt werden, um den Aufwand eines Softwareprojektes abzuschätzen, wird in den nachfolgenden Abschnitten behandelt. Dabei wird zunächst in Abschnitt 2.4.2 allgemein auf Expertenschätzungen eingegangen. Darauf folgend werden verschiedene Methoden zur Durchführung einer Expertenschätzung diskutiert. Anschließend werden in Abschnitt 2.4.3 algorithmische Modelle zur Aufwandschätzung behandelt. Im Fokus stehen hier die bekannten Methoden COCOMO II, Function Point und Use Case Point. Danach werden in Abschnitt 2.4.4 die Klassen zur weitergehenden Unterscheidung erläutert und eine Zuordnung zu den eigenständigen Methoden vorgenommen. Zusammenfassend werden in Abschnitt 2.4.5 die Anforderungen an die Lösung definiert.

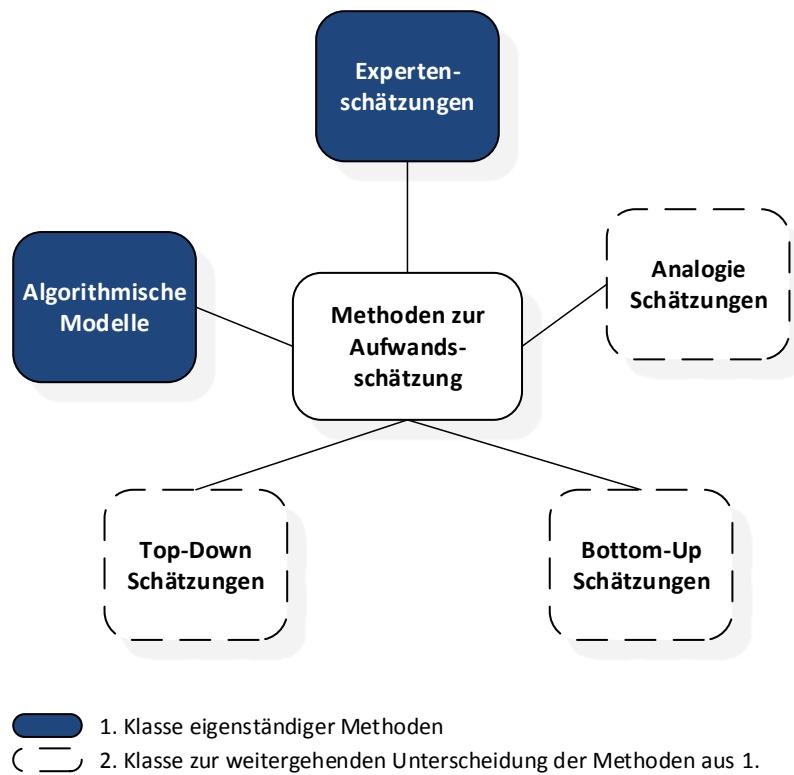


Abbildung 2.4.: Klassifizierung von Methoden zur Aufwandsschätzung nach Boehm [Boe81, S. 329]

## 2.4.2. Expertenschätzungen

Das Heranziehen von Experten zur Lösung von beliebigen Problemstellung ist sehr weit verbreitet und wird seit langem wissenschaftlich untersucht [HR58, S. 37ff]. Die Methoden, die dabei entstanden sind, lassen sich direkt auf das Problem der Aufwandsschätzung in Softwareprojekten übertragen. Für eine detaillierte Darstellung der Methoden zur Expertenschätzung in Softwareprojekten siehe [Boe81, S. 333ff], [Dum03, S. 172], [McC06, S. 185ff] und [Spi11, S. 88ff].

In Abbildung 2.5 ist eine Übersicht über gängige Methoden zur Expertenschätzung zu sehen. Die einfachste Methode ist die Schätzung durch einen einzelnen Experten. Damit sind die Kosten und der Aufwand am geringsten und das Ergebnis steht schnell zur Verfügung. Weiterhin kann der Experte aus seiner Erfahrung heraus auf Besonderheiten des Projektes eingehen. Dies können z. B. eine neue Technologie, personelle Restriktionen und zu erwartende Risiken sein.

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

Neben den Vorteilen, die eine Beurteilung durch einen Experten mitbringt, existieren auch eine Reihe von Nachteilen. Der größte Nachteil ist die Abhängigkeit von einer Person und deren Erfahrung sowie deren Objektivität. Ob ein Mensch optimistisch, pessimistisch oder voreingenommen einem Projekt gegenüber steht, spiegelt sich in hohem Maß in der Aufwandsschätzung wieder.

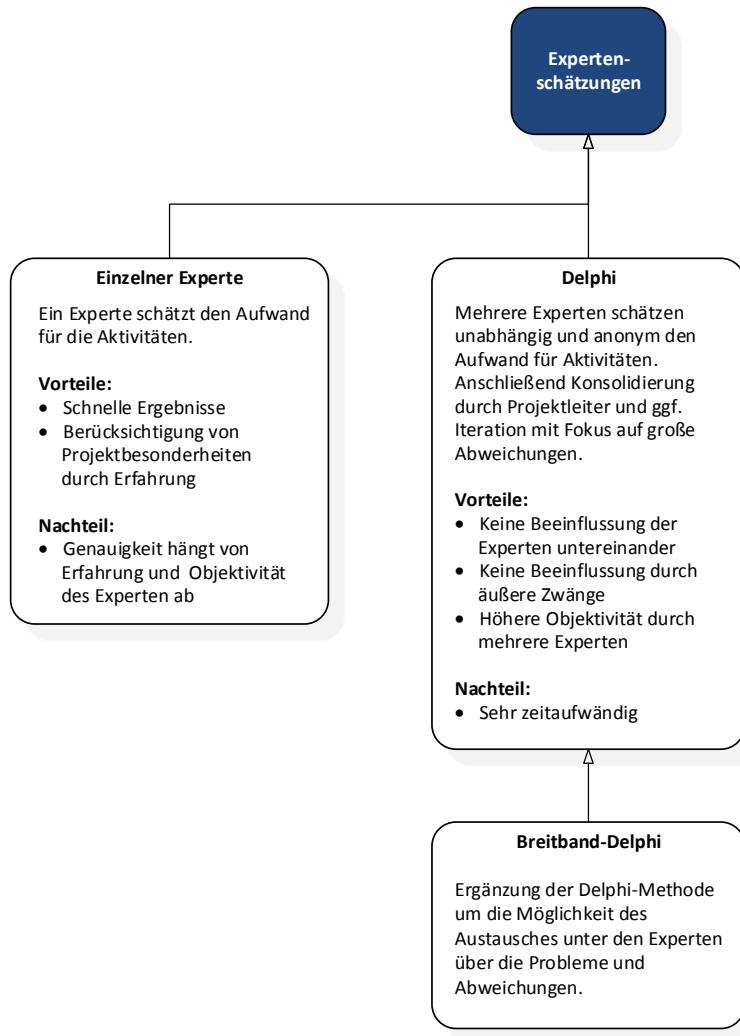


Abbildung 2.5.: Methoden zur Durchführung einer Expertenschätzung

Um dem Nachteil der Abhängigkeit von einer Person zu begegnen, können mehrere Experten zur Schätzung herangezogen werden. Dies bedeutet auf der einen Seite allerdings eine Steigerung der Kosten und des Aufwands für die Durchführung der Schätzung. Auf der anderen Seite wird eine höhere Objektivität erzielt.

Für das Zusammenarbeitsmodell von mehreren Experten und den Prozess der Ergebnisableitung aus den Expertenmeinungen gibt es zahlreiche Methoden (siehe u. a. [HR58, S. 47ff]). In Abbildung 2.5 sind exemplarisch die Delphi [Dal67] und die Breitband-Delphi [Boe81, S. 335] Methode dargestellt. Die Delphi Methode eliminiert den Einfluss der Experten aufeinander, in dem die Schätzungen anonym und ohne Abstimmung der Experten untereinander abgegeben werden. Ein möglicher äußerer Zwang oder auch Zwang durch die Gruppe selbst werden damit ausgeschlossen. Die Kommunikation findet nur zwischen dem Initiator/Koordinator der Schätzung und den Experten statt. Die Breitband-Delphi Methode *verbreitert* die Möglichkeiten zur Kommunikation, in dem Diskussionsrunden der Experten zum Austausch über Probleme und große Abweichungen in den Schätzungen vorgesehen werden. Die Schätzung selbst bleibt auch dabei anonym und unabhängig.

Varianten der Breitband-Delphi Methode finden sich auch im Bereich der agilen Softwareentwicklung. Eine mögliche Variante ist das *Planning Poker* (siehe [Gre02] und [Coh06, S. 49ff]). Dabei werden in einem Treffen des ganzen Entwicklungsteams inkl. der Tester die zu schätzenden Themen durchgegangen und hinsichtlich ihre Komplexität geschätzt. Jeder Teilnehmer in der Runde hat hierfür Karten mit Zahlen zur Hand (z. B. 0, 1, 2, 3, 5, 8, 13, 20, 40, und 100). Nach dem die Fragen zu einem Thema behandelt worden sind, wählt jeder für sich eine Karte. Danach werden die gewählten Karten gleichzeitig gezeigt und die Teilnehmer mit der größten Entfernung der gewählten Zahlen diskutieren über die Abweichung. Nach Abschluss der Komplexitätsschätzung können die ermittelten Werte in Aufwand umgerechnet werden. Der Vorteil dieser Methode ist das gemeinsame Verständnis der Schätzung und der Themen sowie das direkte Austauschen und Diskutieren über mögliche Risiken. Der größte Nachteil ist, dass die Werte immer relativ zum Team und nicht übertragbar auf andere Teams sind. Weiterhin ist dieses Verfahren nur bei kleinen Projektteams einsetzbar und aufwands- sowie kostenintensiv.

### 2.4.3. Algorithmische Modelle

Algorithmische Modelle nutzen einen mathematischen Algorithmus oder eine Kombination aus mehreren mathematischen Algorithmen, um den Aufwand ei-

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

nes Softwareprojektes zu bestimmen. Formal lassen sich die Modelle mit der Formel 2.1 beschreiben [Dum03]. Dabei stellt A die Eingabeparameter dar, die gemessen werden und B die geschätzten. Vereinfacht wird mit der Formel  $f$  aus der Menge der Eingabeparameter A und B der Ausgabeparameter Aufwand berechnet. Die Genauigkeit der Berechnung lässt sich an dem Verhältnis der Anzahl der gemessenen Parameter A ( $n$ ) zu der Anzahl der geschätzten Parameter B ( $m$ ) bewerten. Wenn  $m > n$  gilt, liegt eine schwache Schätzung vor. Im umgekehrten Fall eine solide Schätzung.

$$\text{Aufwand} = f(A, B) \quad (2.1)$$

Die Formel  $f$  selbst kann verschiedene Formen annehmen. Am bekanntesten sind die lineare und die multiplikative Form. Formel 2.2 repräsentiert die lineare Form. Dabei stellen die  $x_0, \dots, x_n$  die Eingabeparameter und die  $\beta_0, \dots, \beta_n$  die Gewichtung der Eingabeparameter dar. Die Gewichte sind aufgrund historischer Daten kalibriert worden, so dass bei gegebenem Aufwand und der Belegung der  $x_0, \dots, x_n$  die Abweichung zwischen dem berechneten und dem gegebenen Aufwand über die Gesamtheit der historischen Daten minimal ist. Für die Formel 2.3 gilt ähnliches. Der Vorteil der multiplikativen Methode gegenüber der linearen ist, dass hier auf nicht-lineare Effekte in der Softwareentwicklung eingegangen werden kann.

$$\text{Aufwand} = \beta_0 + \beta_1 \cdot x_1 + \dots + \beta_n \cdot x_n \quad (2.2)$$

$$\text{Aufwand} = \beta_0 \cdot \beta_1^{x_1} \cdot \dots \cdot \beta_n^{x_n} \quad (2.3)$$

Die Vor- und Nachteile von Methoden auf der Basis eines algorithmischen Modells sind konträr zu denen der Expertenschätzungen [Boe81, S. 332 u. 333]. Die Methoden werden im Gegensatz zu Experten nicht durch äußere oder Gruppenzwänge in ihrem Ergebnis beeinflusst. Weiterhin wird bei gleicher Eingabe immer das gleiche Ergebnis herauskommen und die Methode ist damit wiederholbar und nachvollziehbar. Der Aufwand zur Ausführung ist auf das Ermitteln

## 2.4. Einführung in die Domäne Aufwandsschätzung

der Eingabeparameter beschränkt und damit weniger aufwands- sowie kostenintensiv wie Expertenschätzungen.

Der größte Nachteil ist die fehlende Reaktionsfähigkeit auf Besonderheiten eines Projektes, da die Algorithmen meistens auf schon abgeschlossene Projekte angepasst sind. Ähnliche Schwierigkeiten gibt es bei der Verwendung neuer Technologien in zukünftigen Projekten. Darauf müssen die Algorithmen immer wieder neu angepasst werden. Im Gegensatz zu Experten kann ein Algorithmus die unzureichende Qualität der Eingabeparameter nicht ausgleichen. Wenn nur eine grobe Spezifikation für ein Projekt vorliegt und darauf eine Metrik zur Größen-/Umfangsbestimmung der Anforderungen angewendet wird, ist das Ergebnis genauso grob wie die Spezifikation. Ein Experte könnte hier aus einer Diskussion mit anderen Experten und seinen Erfahrungen eine genauere Größen-/Umfangsbestimmung ableiten.

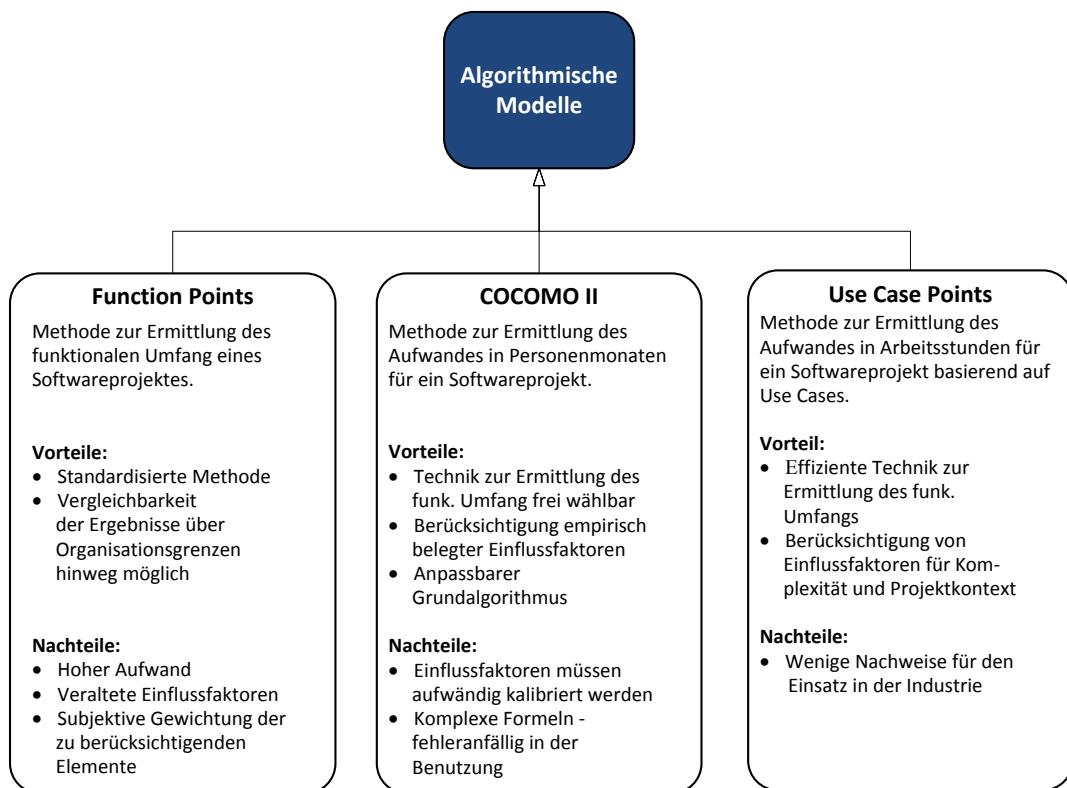


Abbildung 2.6.: Methoden zur Aufwandsschätzung mit algorithmischen Modellen

In Abbildung 2.6 ist eine Übersicht über drei Methoden zur Aufwandsschätzung für Softwareprojekte zu sehen. Die Methode Function Points wird berücksich-

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

tigt, weil sie eine hohe Verbreitung aufweist<sup>7</sup> und mittlerweile ein ISO-Standard ist [ISO09]. Die Methode COCOMO II wird aufgeführt, da sie Grundlage für viele weitere Methoden zur Aufwandsschätzung ist und speziell die Einflussfaktoren einer aufwändigen empirischen Untersuchung unterzogen worden sind [Boe00]. Aufgrund der Verbreitung von Use Cases in der Anforderungsspezifikation für Softwareprojekte wird die Methode Use Case Points betrachtet [Fro09].

Im Folgenden werden jeweils in einem Abschnitt die Methoden Function Points (vgl. Abschnitt 2.4.3.1), COCOMO II (vgl. Abschnitt 2.4.3.2) und Use Case Points (vgl. Abschnitt 2.4.3.3) vorgestellt. Dabei wird die grundlegende Idee der Methode und ihres Algorithmus beschrieben, um die Vor- und Nachteile der Methoden nachzuvollziehen. Basierend hierauf werden in Abschnitt 2.4.5 Anforderungen an die Lösung der Problemstellungen aus Abschnitt 1.1 definiert.

### 2.4.3.1. Function Point

Die Methode Function Point wurde von Allan Albrecht 1979 eingeführt [Alb79] und in den folgenden Jahren ausführlich beschrieben [AG83]. Seit 1984 führt die *International Function Point User Group* (IFPUG) unter dem Begriff Functional Size Measurement die Weiterentwicklung der Methode Function Point. Im Folgenden wird sich an den Begrifflichkeiten und dem Verfahren der IFPUG orientiert [Int10].

Die Idee der Methode Function Point ist, dass die Quantität der Funktionalität eines Softwareproduktes messbar ist. Das Maß der Quantität ist dabei unabhängig von der Technologie, mit der das Softwareprodukt umgesetzt werden soll. Weiterhin werden nicht-funktionale Anforderungen bei der Messung nicht mit berücksichtigt. Aus dem Ergebnis der Messung und der Kenntnis über den Umfang an Funktionalität sowie des Aufwands aus abgeschlossenen Projekten kann anschließend der Aufwand für die Umsetzung des aktuellen Softwareproduktes abgeleitet werden. Für weitergehende Informationen über Function Point eignen sich insbesondere [AR96], [BD08, S. 333ff] und [Sne05, S. 41ff].

Der Algorithmus der Methode Function Point besteht aus vier Schritten:

---

<sup>7</sup>In der Datenbank der *International Software Benchmarking Standards Group* (ISBSG) sind mehr als 6000 Projekte erfasst, deren funktionaler Umfang jeweils über die Methode Function Point ermittelt worden ist [IBS].

## 2.4. Einführung in die Domäne Aufwandsschätzung

1. Ermittlung von *Unadjusted Function Points* (UFP)
2. Bewertung der *General System Characteristics* (GSC)
3. Berechnung des *Value Adjustment Factors* (VAF) aus den GSC
4. Berechnung von *Adjusted Function Points* (AFP) aus den UFP und VAF

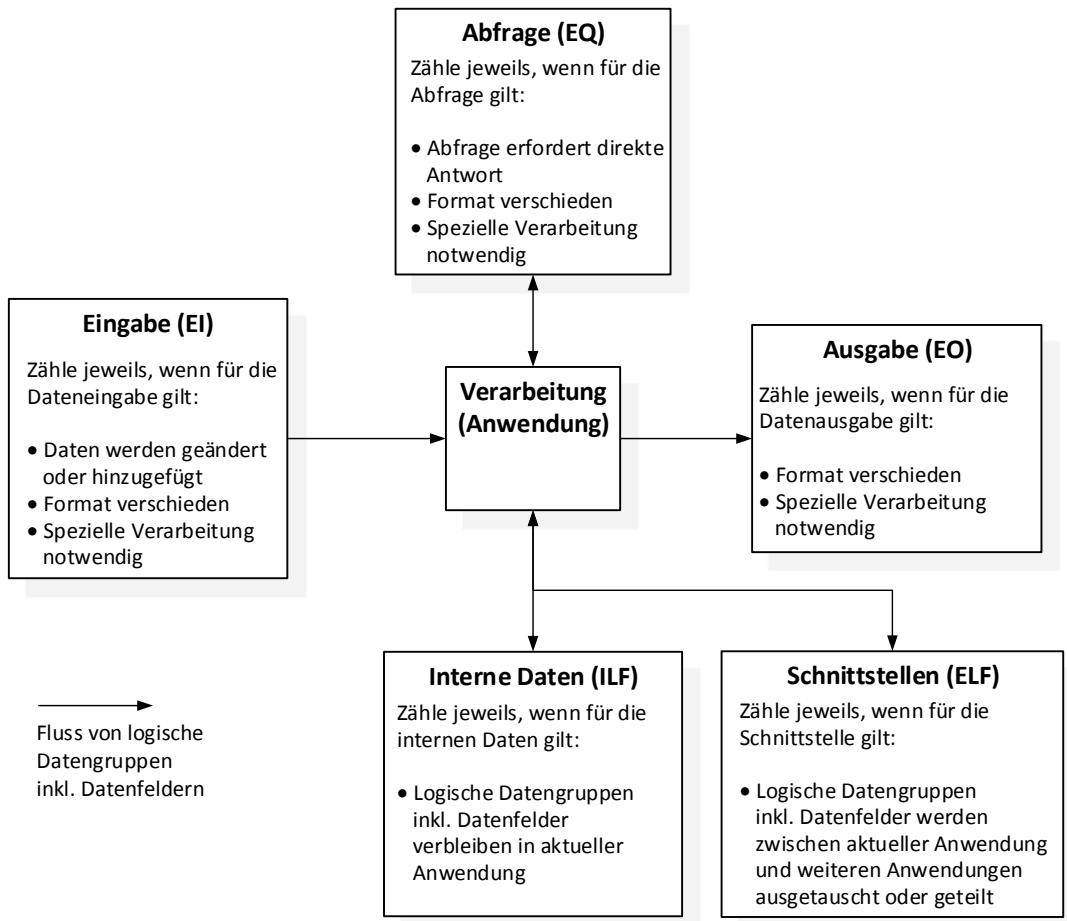


Abbildung 2.7.: Übersicht über die Ermittlung der UFP nach [AG83]

### Schritt 1 der Methode Function Point

Im ersten Schritt der Methode Function Point werden die UFP ermittelt. Dazu werden die Grundfunktionen der zu entwickelnden Anwendung identifiziert und fünf Kategorien zugeordnet (vgl. Abbildung 2.7). Die Identifikation und die Zuordnung erfolgt aus Endnutzerperspektive. Beispielsweise werden über mehrere Datenbanken abgespeicherte Daten als eine logische Datengruppe betrachtet. Im Folgenden werden die Kategorien erläutert:

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

- Die Eingaben (*External Inputs* (EI)) sind logische Datengruppen inkl. Datenfeldern, die an die Anwendung übergeben werden. Übertragen auf das objektorientierte Programmieren, wäre eine logische Datengruppe eine Klasse und die Datenfelder die zugehörigen Attribute. Berücksichtigt werden nur die Eingaben, die Daten ändern oder hinzufügen. Weiterhin wird jedes Format und jede Verarbeitung nur einmal berücksichtigt
- Die Ausgaben (*External Outputs* (EO)) sind logische Datengruppen inkl. Datenfeldern, die die Anwendung verlassen. Jedes Format und jede Verarbeitung wird dabei nur einmal berücksichtigt.
- Anfragen (*External Inquiry* (EQ)) sind Eingaben an das System, die eine direkte Ausgabe erfordern. Jedes Format und jede Verarbeitung wird dabei nur einmal berücksichtigt.
- Interne Daten (*Internal Logical Files* (ILF)) sind logische Datengruppen inkl. Datenfeldern, die in der Anwendung gespeichert oder verarbeitet werden.
- Schnittstellen (*External Logical Files* (ELF)) beschreiben Übergänge zu anderen Anwendungen über die logische Datengruppen inkl. Datenfelder ausgetauscht werden. Die logischen Datengruppen inkl. Datenfeldern werden dabei von der zu entwickelnden Anwendung nur referenziert aber nicht gespeichert.

Nach der Zuordnung zu den Kategorien werden die Grundfunktionen hinsichtlich ihrer Komplexität bewertet und nach einer gegebenen Skala gewichtet. Die jeweils dreistufige Skala ist in Tabelle 2.2 zusammengefasst. Dabei werden die Kategorien in zwei Gruppen betrachtet. Die erste Gruppe besteht aus den ILF und EIF. Die Gewichtung eines ILF bzw. EIF ergibt sich aus der Anzahl der enthaltenden logischen Untergruppen und der Anzahl der Datenfelder. Eine Schnittstelle über die ein Kunde mit seinen Stammdaten und Informationen über das Auto des Kunden abgerufen werden könnte, hätte zwei logische Untergruppen (Kunde und Auto) und z. B. die Datenfelder Name, Vorname und Automarke.

Die zweite Gruppe besteht aus den EI, EO und EQ. Hier ergibt sich die Gewichtung aus der Anzahl der benötigten Datenquellen und der Anzahl der Datenfelder. Beispielsweise könnten die Stammdaten des Kunden aus dem internen

System (ILF) stammen und die Informationen zu dem Auto aus einem externen System (EIF). Damit liegen zwei Datenquellen vor. Datenfelder wären alle vom Benutzer identifizierbaren Datenfelder und Schaltflächen mit denen er interagieren muss, um die Datenfelder zu bedienen.

		Anzahl Datenfelder		
Interne Daten - ILF		1-19	20-50	>50
Anzahl logischer Untergruppen	< 2	7	7	10
	2-5	7	10	15
	> 5	10	15	15
Schnittstellen - EIF		1-19	20-50	>50
Anzahl logischer Untergruppen	< 2	5	5	7
	2-5	5	7	10
	> 5	7	10	10
Eingabe - EI		<5	5-15	>15
Anzahl benötigter ILF und ELF	< 2	3	3	4
	2	3	4	6
	> 2	4	6	6
Ausgabe - EO		<6	6-19	>19
Anzahl benötigter ILF und ELF	< 2	4	4	5
	2-3	4	5	7
	> 3	5	7	7
Anfragen - EQ		<6	6-19	>19
Anzahl benötigter ILF und ELF	< 2	3	3	4
	2-3	3	4	6
	> 3	4	6	6

Tabelle 2.2.: Gewichtung der Funktionen nach [Int10] zur Ermittlung der UFP

Im dritten Schritt werden die UFP durch das Aufsummieren der Gewichtungen aus Schritt 2 gemäß Formel 2.4 ermittelt.

$$UFP = \sum ILF + \sum ELF + \sum EI + \sum EO + \sum EQ \quad (2.4)$$

### Schritte 2-4 der Methode Function Point

Nachdem die UFP ermittelt worden sind, werden die GSC jeweils auf einer Skala von 0 (kein Einfluss) bis 5 (kritischer Einfluss) bewertet. Zur Übersicht

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

sind in Tabelle 2.3 die 14 GSC aus [AG83] aufgeführt. Für die Bewertung der GSC existieren keine formalen Metriken sondern nur natürlichsprachliche kurze Erklärungen, die einen großen Interpretationsspielraum lassen.

Nachdem die GSC bewertet worden sind, ist über die Formel 2.5 der VAF zu berechnen. Damit sind mit den UFP und der Formel 2.6 die AFP bestimbar. Da der Wertebereich des VAF von 0,65 bis 1,35 geht, können die UFP um 35 % nach oben oder unten verändert werden.

$$VAF = 0,01 \cdot \prod_{i=1}^{14} GSC_i + 0,65 \quad (2.5)$$

$$AFP = UFP \cdot VAF \quad (2.6)$$

Die Berechnung des Aufwands aus den AFP ist nicht direkter Teil der Methode Function Point. Eine Möglichkeit, AFP in Aufwand umzurechnen, ist die Analyse von abgeschlossenen Projekten, deren Größe auch mit der Methode Function Point bestimmt worden ist. Voraussetzung ist, dass der Aufwand für diese Projekte erfasst worden ist. Mit dem Aufwand und den AFP für die abgeschlossenen Projekte lässt sich passend für eine Organisation ein Produktivitätsfaktor ableiten. Mit diesem Produktivitätsfaktor, z.B. Aufwand pro 100 AFP, ist der Aufwand für die Umsetzung des Softwareprojektes schätzbar.

### Fazit

Der größte Vorteil der Function Point Methode ist die Verbreitung und die Standardisierung als ISO-Norm [ISO09]. Damit kann bei korrekter Anwendung der Methode auch ein Vergleich der Produktivität von Organisationen untereinander vorgenommen werden. Ein Einstiegsplatz ist die erwähnte Datenbank der IBSG.

Ein weiterer Vorteil ist, dass bei der Durchführung der Function Point Methode sehr genaue Kenntnisse über die gewünschte Funktionalität und die Grenzen der zu entwickelnden Software erworben werden. Dies hilft zum Einen der Projektleitung des Softwareproduktes bei der Planung des Softwareprojektes, da die

<b>Id</b>	<b>Bezeichnung</b>	<b>Kurzbeschreibung</b>
<i>GSC<sub>1</sub></i>	<i>Data Communications</i>	Daten werden durch die Anwendung verschickt
<i>GSC<sub>2</sub></i>	<i>Distributed Functions</i>	Verteilte Verarbeitung von Daten und Prozessen
<i>GSC<sub>3</sub></i>	<i>Performance</i>	Performance Anforderungen an Antwortzeit und Transaktionsrate beeinflussen das Design, die Entwicklung und die Installation der Anwendung
<i>GSC<sub>4</sub></i>	<i>Heavily Used Configuration</i>	Anpassbarkeit an Endgeräte
<i>GSC<sub>5</sub></i>	<i>Transaction Rate</i>	Hohe Transaktionsrate beeinflusst das Design, die Entwicklung, die Installation und den Betrieb der Anwendung
<i>GSC<sub>6</sub></i>	<i>Online Data Entry</i>	Online-Funktion zum Speichern von Dateneingaben
<i>GSC<sub>7</sub></i>	<i>End User Efficiency</i>	Performance Anforderungen an die Online-Funktion zum Speichern von Dateneingaben
<i>GSC<sub>8</sub></i>	<i>Online Update</i>	Online-Funktion zum Aktualisieren von gespeicherten Daten
<i>GSC<sub>9</sub></i>	<i>Complex Processing</i>	Komplexität von Prozessen in der Anwendung
<i>GSC<sub>10</sub></i>	<i>Reuseability</i>	Design für Wiederwendbarkeit
<i>GSC<sub>11</sub></i>	<i>Installation Ease</i>	Anforderungen bzgl. der Installation und der Datenmigration
<i>GSC<sub>12</sub></i>	<i>Operational Ease</i>	Anforderungen bzgl. des Betriebs und der Bedienung
<i>GSC<sub>13</sub></i>	<i>Multiple Sites</i>	Installation an mehreren Standorten
<i>GSC<sub>14</sub></i>	<i>Facilitate Change</i>	Änderungsfreundlichkeit der Anwendung

Tabelle 2.3.: GSC der Methode Function Point nach [AG83]

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

zu entwickelnde Funktionalität besser eingeschätzt werden kann. Zum Anderen müssen Vertreter der Softwareentwicklung mit dem Auftraggeber sehr detailliert über die gewünschte Funktionalität diskutieren. Damit wird die Chance auf ein gleiches Verständnis über die zu entwickelnde Funktionalität auf Seiten der Entwicklung und dem Auftraggeber erhöht.

Der Erfolg der Function Point Methode hat zu weiteren Varianten der Methode geführt. Beispielsweise gibt es die Methode COSMIC Function Point, die sich stärker auf den Bereich der eingebetteten Systeme mit Echtzeitanforderungen fokussiert. Genauso wie die Function Point Methode selbst ist auch die Methode COSMIC Function Point als ISO-Norm aufgenommen worden [ISO11a].

Die Nachteile der Function Point Methode sind vor allem die GSC, die nicht mehr zu modernen Technologien und Entwicklungsmethoden passen. Speziell die Berücksichtigung der heute selbstverständlichen Charakteristik *Online Update* oder *End User Efficiency* sind Hinweise auf eine notwendige Überarbeitung. Auch die fünf Kategorien, die am Anfang des Abschnitts erläutert worden sind, sind nicht mehr passend zu modernen Software-Architekturen und die Gewichtung der Kategorien wurde nie empirisch hinterfragt. Allgemein ist anzumerken, dass die GSC und die Kategorien inkl. Gewichtung für einen speziellen Kontext entwickelt worden sind, der keine Allgemeingültigkeit besitzt.

### 2.4.3.2. COCOMO II

Die Methode COCOMO II ist eine überarbeitete Version der Methode COCOMO (COnstructiveCOStMOdel) aus [Boe81]. Gründe für die Überarbeitung waren u. a. technologische Neuerungen und neue Prozesse im Software Engineering [Boe00, S. XXIX]. Die Idee der Methode ist, dass ein Zusammenhang zwischen der Größe und dem Aufwand zur Erstellung eines Softwareproduktes existiert. Damit kann bei Kenntnis der Größe des Softwareproduktes der Aufwand über einen mathematischen Algorithmus bestimmt werden. Bei der Berechnung des Aufwands werden weitere Einflussfaktoren aus dem Kontext des Projektes und den Eigenschaften des zu entwickelnden Produktes berücksichtigt.

Im Kern besteht die Methode COCOMO II aus einem Algorithmus, der mit der Formel in 2.7 beschrieben wird [Boe00, S. 29ff]. Zur Berechnung des Auf-

wands  $PM$  (Personenmonate) werden die Eingabeparameter  $Size$ ,  $EM_i$  und  $SF_j$  benötigt. Der Parameter  $Size$  stellt die Größe des zu entwickelnden Softwareproduktes dar. Die Größe kann mit einer beliebigen Technik ermittelt werden, sofern ein ganzzahliger Wert das Ergebnis ist. Vorgeschlagen sind u. a. die Methode Function Point oder die Metrik *Lines of Code* (LOC). Die Parameter  $EM_1, \dots, EM_i$  werden mit *effort multiplier* (EM) bezeichnet und beschreiben Einflussfaktoren, die multiplikativ auf den Aufwand wirken. Dagegen sind die Parameter  $SF_1, \dots, SF_5$  sogenannte *scale factors* (SF), die über die Variable  $E$  exponentiell auf die Größe wirken. Für die EM und die SF werden von Boehm Beschreibungen und Metriken mitgeliefert [Boe00].

$$PM = A \cdot Size^E \cdot \prod_{i=1}^n EM_i \quad (2.7)$$

mit  $A = 2,94$

$$\text{mit } E = B + 0,01 \cdot \sum_{j=1}^5 SF_j \quad (2.8)$$

mit  $B = 0,91$

In COCOMO II kann zwischen zwei Modellen gewählt werden, die sich in der Anzahl der zu verwendenden EM unterscheiden [Boe00, S. 12ff]. Das *Early Design* Modell ist für frühe Projektphasen gedacht und sieht 7 EM vor. Dagegen ist das *Post-Architecture* Modell für spätere Projektphasen vorgesehen und erfordert die Eingabe von 17 EM. Trotz der Unterschiede in den EM verwenden beide Modelle dieselben SF und denselben Algorithmus zur Berechnung des Aufwands.

### Bewertung der SF

Die fünf SF sind in Tabelle 2.4 zusammengefasst. Jeder SF muss für ein Projekt auf der qualitativen Skala von *Very Low* bis *Extra High* bewertet werden. Dazu werden von Boehm Vorschläge für Metriken gegeben [Boe00]. Diese Metriken sind in vielen Fällen sehr subjektiv und damit vom jeweiligen Schätzer abhängig. Beispielsweise wird für die qualitative Bewertung *Very Low* des SF TEAM der Hinweis *very difficult interactions* gegeben. Damit ist es der Interpretation des

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

Schäzers überlassen, wie für ein Projekt sehr schwierige Kommunikation zu verstehen ist.

Nachdem die SF qualitativ eingeordnet worden sind, können die vorgegebenen Gewichtungen aus der Tabelle 2.4 abgelesen werden. Diese werden über die Formel 2.8 addiert und über die Variable  $E$  in der Formel 2.7 berücksichtigt. Laut Boehm wirken sie damit skalierend auf den wichtigsten Einflussfaktor, die Größe des zu entwickelnden Softwareproduktes [Boe00, S. 30].

Kürzel	Kurzbeschreibung	<i>Scale Factors</i>		<i>Ratings</i>			
				<i>Very Low</i>	<i>Nominal</i>	<i>High</i>	<i>Very Extra High</i>
		<i>Low</i>	<i>High</i>				
PREC	Ähnliche Vorgängerprojekte	6,20	4,96	3,72	2,48	1,24	0,00
FLEX	Flexibilität in der Entwicklung	5,07	4,05	3,04	2,03	1,01	0,00
RESL	Risikomanagement	7,07	5,65	4,24	2,83	1,41	0,00
TEAM	Zusammenhalt Projektteam	5,48	4,38	3,29	2,19	1,10	0,00
PMAT	Prozess Reifegrad	7,80	6,24	4,68	3,12	1,56	0,00

Tabelle 2.4.: SF von COCOMO II nach [Boe00]

## Bewertung der EM

Da die EM des *Post-Architecture* Modell umfangreicher sind und die EM des *Early Design* Modell vielfach nur die EM des *Post-Architecture* Modells zusammenfassen [Boe00, S. 51], werden hier nur die EM des *Post-Architecture* Modell betrachtet. Diese sind in Tabelle 2.5 zusammengefasst. Genauso wie die SF müssen die EM auf der qualitativen Skala von *Very Low* bis *Extra High* bewertet werden. Auch hier werden Metriken und Hinweise zur Einstufung von Boehm vorgegeben. Wie bei den SF sind diese in vielen Fällen sehr subjektiv und können daher von den Anwendern unterschiedlich interpretiert werden. Nach dem die EM qualitativ eingeordnet worden sind, können die vorgegebenen Gewichtungen aus der Tabelle abgelesen und in die Formel 2.7 eingesetzt werden.

## Fazit

Die Methode COCOMO II ist ein sehr ausgereiftes Modell, das empirisch umfangreich validiert worden ist. Schon in [Boe00] wurden zahlreiche Erweiterungen des Grundalgorithmus beschrieben, um auch andere Bereiche der Softwareent-

		<i>Effort Multiplier</i>		<i>Ratings</i>					
		<b>Kürzel</b>	<b>Kurzbeschreibung</b>	<i>Very Low</i>		<i>Nominal</i>	<i>High</i>	<i>Very Extra High</i>	
<b>Produkt</b>	<b>Plattform</b>			<b>Low</b>	<b>High</b>	<b>High</b>	<b>High</b>	<b>High</b>	<b>High</b>
Personal	RELY	Zuverlässigkeit	0,82	0,92	1,00	1,10	1,26	n/a	
	DATA	Größe der Testdatenbank	n/a	0,90	1,00	1,14	1,28	n/a	
	CPLX	Komplexität	0,73	0,87	1,00	1,17	1,34	1,74	
	RUSE	Wiederverwendung	n/a	0,95	1,00	1,07	1,15	1,24	
	DOCU	Dokumentationsbedarf	0,81	0,91	1,00	1,11	1,23	n/a	
Projekt	TIME	Verbrauch Rechenzeit	n/a	n/a	1,00	1,11	1,29	1,63	
	STOR	Speicherverbrauch	n/a	n/a	1,00	1,05	1,17	1,46	
	PVOL	Stabilität Plattform	n/a	0,87	1,00	1,15	1,30	n/a	
Projekt	ACAP	Qualifikation Analysten	1,42	1,19	1,00	0,85	0,71	n/a	
	PCAP	Qualifikation Programmierer	1,34	1,15	1,00	0,88	0,76	n/a	
	PCON	Personalfluktuation	1,29	1,12	1,00	0,90	0,80	n/a	
	APEX	Anwendungserfahrung	1,22	1,10	1,00	0,88	0,81	n/a	
	PLEX	Plattform Erfahrung	1,19	1,09	1,00	0,91	0,85	n/a	
	LTEX	Sprach- und Werkzeugerfahrung	1,20	1,09	1,00	0,91	0,84	n/a	
Projekt	TOOL	Nutzung von Werkzeugunterstützung	1,17	1,09	1,00	0,90	0,78	n/a	
	SITE	Verteilte Entwicklung	1,22	1,09	1,00	0,93	0,86	0,80	
	SCED	Zeitplanung des Projektes	1,43	1,14	1,00	1,00	1,00	n/a	

 Tabelle 2.5.: EM im *Post-Architecture* Modell von COCOMO II nach [Boe00]

wicklung, wie eingebettete Systeme oder die Fehlervorhersage für das entwickelte Softwareprodukt, abzudecken.

Ein Nachteil der Methode COCOMO II ist die Komplexität des Algorithmus und die notwendige Kalibrierung an neue Unternehmenskontexte. Die Schätzungen mit der Methode COCOMO II sollten aufgrund der Komplexität und der notwendigen Kalibrierung nur von Experten in dem Gebiet der Aufwandsschätzung durchgeführt werden. Dann sind speziell im Bereich der Wartungs- und Weiterentwicklungsprojekte gute Ergebnisse zu erwarten, da die Eingabeparameter EM und SF weitestgehend bekannt sind [Sne05, S. 66]. Für Neuentwicklungen besteht allerdings ein erhöhtes Risiko, da viele Parameter nur geschätzt werden können.

### 2.4.3.3. Use Case Point

Die Methode Use Case Point wurde 1993 von Gustav Karner im Rahmen seiner Diplomarbeit erstmalig beschrieben [Kar93b]. Die Idee der Methode ist, Use Cases als Ausgangspunkt für die Messung der Größe des Softwareproduktes zu benutzen. Die gemessene Größe wird anschließend über produkt- und projektbezogene Einflussfaktoren nach unten oder oben korrigiert. Danach ist die Ableitung des Aufwands mittels einem Vergleich mit früheren Projekten möglich. Da die Methode auf Use Cases basiert, ist ein Use Case basiertes Vorgehen in der Softwareentwicklung Voraussetzung. Für weitergehende Informationen sind insbesondere [Smi99], [ADSJ01], [And02], [MAC05], [Sne05, S. 58ff] und [Fro09] geeignet.

Die Methode Use Case Point besteht im Kern aus drei Schritten:

1. Komplexitätsbewertung der Use Cases mit dem Ergebnis der *Unadjusted Use Case Weights* (UUCW) und der Akteure mit dem Ergebnis der *Unadjusted Actor Weights* (UAW)
2. Berechnung der *Unadjusted Use Case Points* (UUCP) den aus UUCW und den UAW
3. Ableitung der *Use Case Points* (UCP) aus den UUCP unter Berücksichtigung eines *Technical Complexity Factor* (TCF) und eines *Environmental Factor* (EFA)

#### Schritt 1: Komplexitätsbewertung der Use Cases

Im ersten Schritt wird zunächst jeder Use Case in eine von drei Komplexitätsklassen eingeordnet. In Tabelle 2.6 sind hierzu die drei Klassen mit ihrer Definition und resultierenden Gewichtung aufgeführt. Die Klassen *Simple*, *Average* und *Complex* unterscheiden sich hierbei nur an der Anzahl *transactions*. *transactions* selbst wurden von Karner nicht weiter erläutert und werden von mir analog zu Sneed und Frohnhoff als Anzahl Schritte in einem Use Case übersetzt [Sne05, S. 59][Fro09, S. 29].

Beim Bewerten der Use Cases kann parallel direkt die Einordnung der Akteure in drei Komplexitätsklassen erfolgen. Diese sind ähnlich zu denen der Use Cases

## 2.4. Einführung in die Domäne Aufwandsschätzung

in Tabelle 2.7 aufgeführt. Mit den Komplexitätsklassen ist hier die Schnittstelle zu bewerteten, über die die Software mit dem Akteur interagiert.

Komplexität Definition		Gewicht
<i>Simple</i>	Ein Use Case ist <i>simple</i> , wenn er 3 oder weniger <i>transactions</i> beinhaltet.	5
<i>Average</i>	Ein Use Case ist <i>average</i> , wenn er 3 bis 7 <i>transactions</i> beinhaltet.	10
<i>Complex</i>	Ein Use Case ist <i>complex</i> , wenn er mehr als 7 <i>transactions</i> beinhaltet.	15

Tabelle 2.6.: Gewichtung der Use Cases nach [Kar93a]

Komplexität Definition		Gewicht
<i>Simple</i>	Ein Akteur ist <i>simple</i> , wenn er ein anderes System repräsentiert, das über eine definierte <i>application programming interface</i> (API) aufgerufen wird.	1
<i>Average</i>	Ein Akteur ist <i>average</i> , wenn er:	2
	1. eine Interaktion mit einem anderen System über ein definiertes Protokoll repräsentiert. 2. eine menschliche Interaktion über ein Terminal (Kommandozeile ohne grafische Benutzeroberfläche) repräsentiert.	
<i>Complex</i>	Ein Akteur ist <i>complex</i> , wenn er über eine grafische Benutzeroberfläche mit dem System interagiert.	3

Tabelle 2.7.: Gewichtung der Akteure nach [Kar93a]

### Schritt 2: Berechnung der UUCP

Im zweiten Schritt wird die Menge der UUCP über die Formel 2.9 bestimmt. Die Variable  $UUCW_i$  stellt das Gewicht für Use Case  $i$  dar, das entsprechend der zugeordneten Komplexitätsklasse aus Tabelle 2.6 abgelesen werden kann. Analog stellt die Variable  $UAW_j$  das Gewicht für Akteur  $j$  dar.

$$UUCP = \sum_{i=1}^{\text{Anzahl Use Cases}} UUCW_i + \sum_{j=1}^{\text{Anzahl Akteure}} UAW_j \quad (2.9)$$

### Schritt 3: Ermittlung des TCF und EFA

Im dritten Schritt werden nacheinander der TCF und der EFA ermittelt. Für die Berechnung des TCF müssen die 13 verschiedene Einflussgrößen aus Tabelle 2.8 auf der Skala 0, 1, 2, 3, 4, 5 bewertet werden. Dabei ist 0 gleichbedeutend mit keinem Einfluss und 5 mit erheblichem Einfluss. Die zu bewertenden Einflussgrößen stellen nicht-funktionale Anforderungen an das Softwareprodukt dar, wie z. B. Änderbarkeit (*Changeability*), Portabilität (*Portability*) und Performance (*End user efficiency (online)*). Über die Formel 2.10 ist anschließend aus der Bewertung jeder Einflussgröße  $T_i$  und der Gewichtung  $W_{T_i}$  aus Tabelle 2.8 der TCF zu berechnen. Damit bewegt sich der TCF im Wertebereich [0, 6 – 1, 3].

$$TCF = 0,6 + 0,01 \cdot \sum_{i=1}^{13} T_i \cdot W_{T_i} \quad (2.10)$$

$$EFA = 1,4 - 0,03 \cdot \sum_{i=1}^8 E_i \cdot W_{E_i} \quad (2.11)$$

$$UCP = UUCP \cdot TCF \cdot EF \quad (2.12)$$

Der EFA wird analog zu dem TCF berechnet. Zunächst müssen die 8 Einflussgrößen aus Tabelle 2.9 auf der Skala 0, 1, 2, 3, 4, 5 bewertet werden. Dabei ist 0 wieder gleichbedeutend mit keinem Einfluss und 5 mit erheblichem Einfluss. Die zu bewertenden Einflussgrößen sind auf den Projektkontext bezogen, wie z. B. die Fähigkeiten der Mitarbeiter (*Analyst Capability*), die Motivation und die Erfahrung der Mitarbeiter im Entwicklungsprozess (*Familiar with Rational Unified Process*). Mit der Formel 2.11 werden die einzelnen Bewertungen der Einflussgrößen  $E_i$  mit der zugehörigen Gewichtung  $W_{E_i}$  zum EFA verrechnet. Der EFA bewegt sich, speziell mit Berücksichtigung von negativen Gewichten, in dem relativ großen Wertebereich [0, 430 – 1, 705].

Nachdem der TCF und der EFA berechnet worden sind, ist über die Formel 2.12 die Menge der UCP bestimmbar. Dabei werden die UUCP mit dem TCF und dem EFA multipliziert, d. h. die UUCP werden nach oben oder unten korrigiert. Durch die Wertebereiche des TCF und des EFA bewegt sich die Korrektur im Bereich einer Reduktion um 75 % bis zu einer Erhöhung um 220 %.

## 2.4. Einführung in die Domäne Aufwandsschätzung

Ähnlich wie bei der Methode Function Point können die berechneten UCP in Aufwand umgerechnet werden. Dies ist über einen Produktivitätsfaktor möglich, der aus der Analyse abgeschlossener Projekte ableitbar ist. Dazu muss für diese Projekte die Methode Use Case Point angewendet worden sein und der tatsächliche Aufwand bekannt sein. Der Produktivitätsfaktor, z. B. 8 Stunden Arbeit pro Use Case Point, ist dann mit den UCP zu multiplizieren, um den Aufwand zu berechnen.

$T_i$	Einflussgröße	Gewicht $W_{T_i}$
$T_1$	<i>Distributed systems</i>	2
$T_2$	<i>Application performance objectives, in either response or throughput</i>	1
$T_3$	<i>End user efficiency (online)</i>	1
$T_4$	<i>Complex internal processing</i>	1
$T_5$	<i>Reusability, the code must be able to be reused in other applications</i>	1
$T_6$	<i>Installation ease</i>	0,5
$T_7$	<i>Operational ease, usability</i>	0,5
$T_8$	<i>Portability</i>	2
$T_9$	<i>Changeability</i>	1
$T_{10}$	<i>Concurrency</i>	1
$T_{11}$	<i>Special security features</i>	1
$T_{12}$	<i>Provide direct access for third parties</i>	1
$T_{13}$	<i>Special user training facilities</i>	1

Tabelle 2.8.: Einflussfaktoren zur Berechnung des TCF nach [Kar93a]

$E_i$	Einflussgröße	Gewicht $W_{E_i}$
$E_1$	<i>Familiar with Rational Unified Process</i>	1,5
$E_2$	<i>Part time workers</i>	-1
$E_3$	<i>Analyst Capability</i>	0,5
$E_4$	<i>Application experience</i>	0,5
$E_5$	<i>Object oriented experience</i>	1
$E_6$	<i>Motivation</i>	1
$E_7$	<i>Difficult programming language</i>	-1
$E_8$	<i>Stable requirements</i>	2

Tabelle 2.9.: Einflussfaktoren zur Berechnung des EFA nach [Kar93a]

## Fazit

Es gibt eine Vielzahl von Erweiterungen der Use Case Point Methode. In dem Papier *Effort Estimation of Use Cases for Incremental Large-Scale Software Development* [MAC05] wird beispielsweise beschrieben, wie auf Änderungen in Use Cases eingegangen werden kann und somit die Weiterentwicklung von bestehenden Anwendungen geschätzt werden kann. Dazu werden u. a. Ideen aus COCOMO II aufgegriffen.

Eine zusätzliche Erweiterung zeigt eine mögliche Lösung für das Problem des unterschiedlichen Abstraktionsniveau von Use Case basierten Anforderungsspezifikationen (vgl. Abschnitt 2.2.2). Fachliche und technische Spezifikationen sind ähnlich mit Use Cases modellierbar. Damit können beide als Basis für eine Schätzung mit der Methode Use Case Point dienen. Da die Spezifikationen sich allerdings hinsichtlich des Abstraktionsniveaus unterscheiden, sind sie nicht vergleichbar.

Für das zugrundeliegende Problem des unterschiedlichen Abstraktionsniveau von Use Case basierten Anforderungsspezifikationen wurde schon 1999 in dem Papier *The Estimation of Effort Based on Use Cases* [Smi99] beschrieben, wie Schätzungen von einem in ein anderes Abstraktionsniveau umgerechnet werden können.

Neben den Erweiterungen gibt es auch vollständige Überarbeitungen hinsichtlich der Einflussfaktoren und der Zählmechanik für die Use Case Bewertung [Fro08, Fro09]. Die Verwendung der überarbeiteten Versionen ist zu empfehlen, da die ursprünglichen Einflussfaktoren nicht ausreichend beschrieben worden sind und keine Hinweise für empirische Belege der zugehörigen Gewichte vorhanden sind. Außerdem wurde mit der Überarbeitung der Einflussfaktoren auch die Zählmechanik stärker formalisiert, so dass die Unterschiede bei der Bewertung von Use Cases durch unterschiedliche Anwender geringer sind.

Der Hauptvorteil der Methode Use Case Point ist die einfache und effiziente Anwendbarkeit. Ein Projekt mit einem erwarteten Aufwand von 10 Bearbeiterjahren ist innerhalb eines Tages schätzbar [Fro09, S. 33]. Dabei ist die Methode, wenn man von den technischen nicht-funktionalen Einflussgrößen absieht, auch durch Nicht-Informatiker anwendbar.

Ein weiterer Vorteil ist, dass direkt auf den verfügbaren Informationen aufgesetzt werden kann. Die einzige Voraussetzung ist eine Use Case basierte Anforderungsaufnahme oder eine Abbildung der Technik zur Anforderungsspezifikation auf die Elemente der Use Case Point Methode [Fro09].

Die Nachteile der Use Case Point Methode sind vor allem die nicht vorhandene Standardisierung von Use Cases, die veralteten Einflussgrößen und die fehlende empirische Untersuchung der Gewichtung. In der überarbeiteten Methode *Use Case Point 3.0* von Frohnhoff wurde speziell an diesen Nachteilen gearbeitet und mögliche Lösungen aufgezeigt [Fro09]. Im Allgemeinen fehlt eine größere empirische Validierung, da bisher eher Validierungen im kleineren Rahmen existieren [ADSJ01, And02].

### 2.4.4. Unterstützende Methoden

Die in Abbildung 2.4 gezeigten drei Klassen *Top-Down*, *Bottom-Up* und Analogie-Schätzung charakterisieren eine Methode zur Aufwandsschätzung näher. Die Klassen *Top-Down* und *Bottom-Up* beziehen sich dabei gesamthaft auf das Vorgehen einer Methode. Die Analogie-Schätzung kann dagegen auch nur auf einen Schritt im Vorgehen der Methode bezogen sein. Im Folgenden werden die drei Klassen näher vorgestellt und den im vorangegangenen erläuterten Methoden zugeordnet.

#### ***Top-Down* Schätzungen**

Methoden zur Aufwandsschätzung, die nach dem *Top-Down* Prinzip vorgehen, leiten aus den globalen Eigenschaften eines Softwareproduktes und/oder eines Softwareprojektes den Aufwand ab [Boe81, S. 337-338][Spi11, S. 85]. Global sind hier Eigenschaften, wie z. B. die LOC, die Fähigkeiten der Mitarbeiter und die Stabilität der Anforderungen. Aus diesen Eigenschaften wird über die jeweilige Methode eine Beziehung zum Aufwand hergestellt. Oft ist hierbei in einem der letzten Schritte der Methode eine enge Verknüpfung zur Analogie-Schätzung zu finden, denn der Aufwand wird im Vergleich mit abgeschlossenen Projekten abgeschätzt. Der geschätzte Aufwand wird im Anschluss an die Anwendung der Methode auf einzelne Phasen und/oder Aktivitäten eines Projektes verteilt. Dies

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

kann z. B. über definierte Prozentsätze, die auch aus abgeschlossenen Projekten gewonnen werden konnten, vorgenommen werden.

Beispiele für Methoden, die ein *Top-Down* Vorgehen verfolgen, sind Function Point, COCOMO II und Use Case Point. Die drei Methoden berücksichtigen u. a. Produkt- und Projektmerkmale, um den Aufwand zu ermitteln. Die Vorteile dieser Methoden sind die effiziente Ermittlung des Aufwands und der Fokus auf das Gesamtprojekt. Die Nachteile sind die geringere Stabilität der Schätzung und die fehlende Berücksichtigung von Problemen einzelner Aktivitäten im Projekt. Die geringere Stabilität entsteht daraus, dass nur eine Schätzung durchgeführt wird und damit auch nur ein Schätzfehler existiert [McC06, S. 149ff].

### ***Bottom-Up* Schätzungen**

Methoden zur Aufwandsschätzung, die nach dem *Bottom-Up* Prinzip vorgehen, schätzen typischerweise den Aufwand für Einzelaktivitäten in einem Softwareprojekt separat [Boe81, S. 338][Spi11, S. 85]. Die Summe der Aufwände für die Einzelaktivitäten stellt dann den Gesamtaufwand für das Projekt dar. Varianten des *Bottom-Up* Prinzips können das Schätzen von Einzelkomponenten oder Phasen eines Softwareprojektes beinhalten.

Beispiele für Methoden, die ein *Bottom-Up* Vorgehen verfolgen, sind die Delphi-Methode oder das *Planning Poker*. Theoretisch könnte die Delphi Methode auch *Top-Down* angewendet werden, aber in der Regel werden durch Experten die einzelnen Aktivitäten eines Projektes separat betrachtet und bewertet. Beim *Planning Poker* wird vereinfacht für einzelne Komponenten die Größe ermittelt und danach in Aufwand umgerechnet. Der Summe der Einzelaufwände repräsentiert dann den Gesamtaufwand.

Die Vorteile der Methoden mit *Bottom-Up* Vorgehen sind die genaue Betrachtung der zu schätzenden Aktivitäten, Komponenten oder Phasen und die größere Stabilität der Schätzung. Die größere Stabilität resultiert daraus, dass die Einzelfehler der Schätzungen für Aktivitäten, Komponenten oder Phasen sich untereinander ausgleichen [McC06, S. 149ff]. Das Projekte, die mit einer Methode nach *Bottom-Up* Vorgehen, geschätzt worden sind, weniger Zeit- und Budgetüberschreitungen zur Folge haben, wurde in [LP92, S. 57] gezeigt. Ein Nachteil ist, dass die Aktivitäten des Gesamtprojektes in der Schätzung vernachlässigt werden.

sigt werden. Dies sind vor allem übergreifende Tätigkeiten wie das Gesamtprojektmanagement oder die Qualitätssicherung. Weiterhin ist ein Vorgehen nach *Bottom-Up* Prinzip aufwendig, da wesentlich detaillierter bewertet wird und an der Schätzung meistens mehrere Experten beteiligt sind.

### Analogie-Schätzungen

Methoden zur Aufwandsschätzung, die nach dem Analogie Prinzip vorgehen, vergleichen das zu schätzende Projekt mit einem oder mehreren ähnlichen abgeschlossenen Projekten [McC06, S. 163][Boe81, S . 336]. Der Vergleich kann z. B. über Produkt- und Projekteigenschaften stattfinden. Das Ergebnis des Vergleichs ist anschließend Eingabe für die Berechnung des Aufwands. Dabei wird der Aufwand des abgeschlossenen ähnlichen Projektes oder der durchschnittliche Aufwand von mehreren ähnlichen abgeschlossenen Projekten als Ausgangspunkt genutzt.

Fast jede Methode zur Aufwandsschätzung beinhaltet einen oder mehrere Schritte, die nach dem Analogie Prinzip vorgehen. Letztendlich wird in jeder Methode, die auf Daten aus abgeschlossenen Projekten zurückgreift, ein Vergleich durchgeführt. Der Vorteil ist, dass über die Betrachtung der abgeschlossenen Projekte, die aktuelle Schätzung auf einer reellen Grundlage beruht. Weiterhin werden viele Risiken durch den Vergleich mit abgeschlossenen Projekten schon bei der Aufwandsschätzung aufgedeckt. Ein Risiko ist, dass das abgeschlossene Projekt bzw. die abgeschlossenen Projekte nicht repräsentativ für das neue Projekt sein müssen [Boe81, S. 336].

Mit der Kenntnis der Vor- und Nachteile der Klassen können Methoden zur Aufwandsschätzung abhängig von der jeweiligen Situation ausgewählt und im Projektverlauf kombiniert werden. Wenn nur wenige Informationen verfügbar sind, bietet es sich z. B. an, mit einer Methode nach dem *Top-Down* Verfahren zu starten. Im späteren Projektverlauf, wenn mehr Details zu den einzelnen Aktivitäten verfügbar sind, ist die initiale Schätzung mit einer Methode nach dem *Bottom-Up* Prinzip verfeinerbar. Die Kombination der Verfahren mit dem Ziehen von Analogien zu abgeschlossenen Projekten bietet sich immer an. Allerdings sollten hierfür die Rahmenbedingungen der Projekte vergleichbar sein und damit möglichst von demselben Unternehmen durchgeführt werden.

### 2.4.5. Anforderungen an die Lösung

Die Einführung in die Domäne Aufwandsschätzung wurde ausgehend von einer Klassifizierung von Methoden zur Aufwandsschätzung in Abschnitt 2.4.1 vorgenommen. Dabei wurden für jede in Abschnitt 2.4.1 identifizierte Klasse bekannte Repräsentanten in den Abschnitten 2.4.2 bis 2.4.4 vorgestellt. Basierend auf der Klassifizierung und den Beschreibungen der Repräsentanten werden in diesem Abschnitt Anforderungen an die Lösung der Problemstellungen aus Abschnitt 1.1 formuliert.

Für den Testmanager selbst und auch für den Kunden, der die Testphase finanziert, ist die Nachvollziehbarkeit der Testaufwandsschätzung von hoher Bedeutung. Die Nachvollziehbarkeit ist durch die Transparenz der Aufwandsberechnung erreichbar. Diese ist in hohem Maß bei Methoden gegeben, die auf einem algorithmischen Modell beruhen (vgl. Abschnitt 2.4.3).

Neben der Nachvollziehbarkeit sind die schnelle Verfügbarkeit einer Testaufwandsschätzung und ein geringer Durchführungsaufwand für die Erstellung der Testaufwandsschätzung entscheidend. Gleichzeitig muss eine angemessene Genauigkeit in der Vorhersage des Testaufwands mit der Testaufwandsschätzung erreicht werden. Dies ist bei Expertenschätzungen mit einem Einsatz von mehreren Experten oft nicht möglich, da der Durchführungsaufwand zu hoch ist (vgl. Abschnitt 2.4.2). Dagegen führt der Einsatz von Methoden, die auf algorithmischen Modell beruhen, zu einem Ergebnis mit angemessener Genauigkeit bei geringem Durchführungsaufwand (vgl. die Methode Use Case Point in Abschnitt 2.4.3). Speziell Methoden mit einer Kombination aus algorithmischen Modell und einem *Top-Down* Vorgehen sind mit hoher Effizienz durchführbar. Daraus ergibt sich die Anforderung an die Lösung der Problemstellungen dieser Dissertation:

#### Anforderung 8:

Algorithmisches Modell mit *Top-Down* Vorgehen

In Abschnitt 2.4.3 wurden verschiedene Formen eines algorithmischen Modells vorgestellt. Die in Formel 2.3 vorgestellte multiplikative Form bietet den Vorteil, dass ähnlich wie bei COCOMO II oder Function Point Einflüsse auf den Aufwand als Prozentwerte ausgedrückt werden können. Damit ist die Höhe des Einflusses besser für den Benutzer interpretierbar als ein Summand in der li-

nearen Form. Ähnlich wurde in [Fro09, S. 133] argumentiert. Hieraus folgt die Anforderung:

**Anforderung 9:**

Multiplikative Form der Formel zur Berechnung des Testaufwands

Nach dem TPI Verfahren [van11, S. 84] ist die höchste Stufe für die Kostenschätzung und Planung im Testbereich erreicht, wenn die Kostenschätzung auf den Erfahrungen des Unternehmens beruht. Genauso werden die Methoden COCOMO II, Function Point und Use Case Point genauer, je mehr Daten aus abgeschlossenen Projekten zur Verfügung stehen (vgl. Abschnitt 2.4.3). Aus diesem Grund wird die Anforderung abgeleitet:

**Anforderung 10:**

Berücksichtigung von Projektdaten aus abgeschlossenen Projekten

Die bestehenden Methoden für Aufwandsschätzungen, wie COCOMO und Function Points, enthalten ausführliche Vorgaben zur Bewertung der jeweiligen Einflussfaktoren (vgl. Abschnitt 2.4.3). Dies hat den Vorteil, dass der Interpretationsspielraum für den Anwender klein ist und die Schätzungen unterschiedlicher Personen vergleichbar sind. Um dies auch für die zu entwickelnde Methode zur Testaufwandsschätzung zu gewährleisten, entsteht die nachfolgende Anforderung:

**Anforderung 11:**

Formale Vorgaben zur Bewertung von Testeinflussfaktoren

Methoden für Aufwandsschätzungen verlieren mit der Zeit an Aktualität und damit an Genauigkeit in der Vorhersage von Aufwänden. Ein Beispiel ist die Überarbeitung von COCOMO. COCOMO hatte durch sich verändernde Bedingungen, wie z. B. neue technologische Entwicklungen, an Aktualität und damit auch an Aussagekraft verloren. Gelöst wurde dies durch eine vollständige Überarbeitung der Methode COCOMO zu der Methode COCOMO II [Boe00] (vgl. Abschnitt 2.4.3). Ähnlich zu betrachten ist die Überarbeitung der Methode Use Case Point [Fro09, Fro08] (vgl. Abschnitt 2.4.3). Auf den Bereich der Testaufwandsschätzungen trifft diese Problematik genauso zu. Vor allem die Testeinflussfaktoren sind in zweierlei Hinsicht betroffen. Erstens können die Gewichte bestehender Testeinflussfaktoren durch sich verändernde Bedingungen, wie z. B. neue technologische Entwicklungen oder neue Zusammenarbeitsmodelle, verschoben werden. Die Verschiebung kann von einem nun nicht mehr vorhan-

## 2. Einführung in Grundlagen und Identifikation von Anforderungen

denen bis zu einem deutlich stärkeren Einfluss erfolgen. Zweitens können neue Testeinflussfaktoren hinzukommen, die in das Modell integriert werden müssen. Daher muss die Lösung diese beiden Anforderungen berücksichtigen:

### Anforderung 12:

Stabil gegenüber neuen/wechselnden Technologien zur Realisierung des Softwareproduktes

### Anforderung 13:

Aktualisierbare Testeinflussfaktoren

Mit den in diesem Abschnitt vorgestellten Anforderungen ist die Erhebung von Anforderungen an die Lösung der in Abschnitt 1.1 definierten Problemstellung abgeschlossen. Im Abschnitt 2.5 wird das Kapitel mit einer Zusammenfassung aller identifizierten Anforderungen geschlossen.

## 2.5. Zusammenfassung

In diesem Kapitel wurden, neben der Einführung der grundlegenden Begriffe, Anforderungen an die Lösung der in Abschnitt 1.1 definierten Problemstellung identifiziert. Im Folgenden werden diese Anforderungen zusammengefasst.

In Tabelle 2.10 sind die in diesem Kapitel identifizierten Anforderungen den entsprechenden Problemstellungen zugeordnet worden. Da es Anforderungen gibt, die alle drei Problemstellungen betreffen, werden diese mit *Alle* in der Tabelle ausgewiesen. Für die Lösung von Problemstellung 1 müssen die Standardmodellierungssprachen der UML unterstützt werden, da diese sehr verbreitet sind (A 4). Um jedoch zukünftig weitere Arten der Anforderungsspezifikation nutzen zu können, soll die Möglichkeit zur Anbindung alternativer Modellierungssprachen geboten werden (A 5).

Zur Lösung der Problemstellung 2 müssen bei der Identifikation von Testeinflussfaktoren u. a. die Qualifikationen der Rollen Anforderungsanalytiker des Auftraggebers sowie des Auftragnehmers und des Softwarearchitekten berücksichtigt werden (A 3). Für die Formalisierung der Testeinflussfaktoren müssen formale Vorgaben für die Bewertung der Testeinflussfaktoren vorgesehen werden (A 11). Damit ist der Interpretationsspielraum für den Anwender klein und

P	A	Beschreibung
P 1	A 4	Unterstützung von Standardmodellierungssprachen der UML
P 1	A 5	Möglichkeit zur Anbindung von weiteren Modellierungssprachen
P 2	A 3	Berücksichtigung der Qualifikationen von Personen in den Rollen Anforderungsanalytiker des Auftraggebers sowie des Auftragnehmers und des Softwarearchitekten
P 2	A 11	Formale Vorgaben zur Bewertung von Testeinflussfaktoren
P 2	A 12	Stabil gegenüber neuen/wechselnden Technologien zur Realisierung des Softwareproduktes
P 2	A 13	Aktualisierbare Testeinflussfaktoren
P 3	A 8	Algorithmisches Modell mit <i>Top-Down</i> Vorgehen
P 3	A 9	Multiplikative Form der Formel zur Berechnung des Testaufwands
Alle	A 1	Beliebiger Zeitpunkt für die Durchführung der Testaufwandschätzung
Alle	A 2	Aktualisierbare Testaufwandsschätzung
Alle	A 6	Aufschlüsselung des Testaufwands nach Testaktivitäten
Alle	A 7	Unterstützung der Teststufen (System-)Integrationstest, Systemtest und Abnahmetest
Alle	A 10	Berücksichtigung von Projektdaten aus abgeschlossenen Projekten

Tabelle 2.10.: Übersicht über die Anforderungen (A) und die Zuordnung zu den Problemstellungen (P)

die Schätzungen unterschiedlicher Personen sind vergleichbar. Um die Notwendigkeit der Überarbeitung von Testeinflussfaktoren weitestgehend zu vermeiden, müssen diese stabil gegenüber neuen/wechselnden Technologien zur Realisierung des Softwareproduktes sein (A 12). Da trotzdem Überarbeitungen erforderlich werden können, ist die Formalisierung der Testeinflussfaktoren aktualisierbar zu gestalten (A 13).

Die Lösung der Problemstellung 3 muss auf einem algorithmischen Modell beruhen, das dem *Top-Down* Vorgehen folgt (A 8). Damit ist eine hohe Effizienz bei der Durchführung einer Testaufwandsschätzung bei angemessener Genauigkeit gewährleistet. Um dabei die Interpretierbarkeit der Formalisierung der Verbindung von Testkomplexität und Testeinflussfaktoren zu unterstützen, wird eine multiplikative Form der Formel zur Berechnung des Testaufwands gefordert (A 9).

## *2. Einführung in Grundlagen und Identifikation von Anforderungen*

Die Lösung der Problemstellungen muss den beliebigen Zeitpunkt zur Durchführung der Testaufwandsschätzung beinhalten (A 1), da eine Testaufwandschätzungen jederzeit in einem Softwareprojekt erforderlich sein kann. Dabei sollte der Informationsstand nach dem aktuellen Zeitpunkt im Softwareprojekt unterschieden werden. Um dies zu ermöglichen, wurde ein Softwareprojekt bei der Vorstellung des V-Modell in die Phasen Anforderungsphase, Designphase und Entwicklungsphase unterteilt (vgl. Abschnitt 2.1.1). Diese Phasen müssen bei der Umsetzung der Anforderung A 1 berücksichtigt werden.

Neben dem beliebigen Zeitpunkt muss die Lösung der Problemstellungen die Testaktivitäten Testmanagement, Testdesign und Testausführung (A 6) für die Teststufen (System-)Integrationstest, Systemtest und Abnahmetest (A 7) unterstützen. Damit werden die für diese Dissertation relevanten Testaktivitäten des fundamentalen Testprozesses (FTP) des ISTQB und die Teststufen berücksichtigt.

Für die Lösungen der drei Problemstellungen muss auf die Erfahrungen aus abgeschlossenen Projekten zurückgegriffen werden (A 10). Darüber ist gewährleistet, dass die Testaufwandsschätzungen auf den Erfahrungen des Unternehmens beruhen.

Die in diesem Kapitel identifizierten Anforderungen sind der Ausgangspunkt für die Beschreibung des Lösungsansatzes im nächsten Kapitel. Dabei werden die Anforderungen zunächst für die Untersuchung verwandter Arbeiten genutzt. Anschließend wird ein Überblick über den Lösungsansatz für die Problemstellungen aus Abschnitt 1.1 unter Berücksichtigung der Anforderungen gegeben. Nach der Detaillierung des Lösungsansatzes in den Kapiteln 4 bis 6 werden in Kapitel 7 die Anforderungen erneut aufgegriffen und bewertet, ob diese durch die entwickelte Lösung erfüllt sind.

# **3. Verwandte Arbeiten und Überblick über den Lösungsansatz**

In diesem Kapitel werden verwandte Methoden zur Testaufwandsschätzung untersucht und ein Überblick über den Lösungsansatz für die Problemstellungen aus Abschnitt 1.1 gegeben. Dazu werden zunächst in Abschnitt 3.1 bekannte Methoden zur Testaufwandsschätzung identifiziert und evaluiert. Dabei wird jeweils die Erfüllung der in Abschnitt 1.1 definierten Problemstellungen sowie der in Kapitel 2 aufgestellten Anforderungen betrachtet.

Es wird sich zeigen, dass keine der Methoden die Problemstellung bzw. die Anforderungen ausreichend erfüllt. Daher wird im Anschluss in Abschnitt 3.2 ein Überblick über den Lösungsansatz dieser Dissertation für die Problemstellungen aus Abschnitt 1.1 erläutert.

## **3.1. Methoden zur Testaufwandsschätzung**

Um verwandte Methoden zur Testaufwandsschätzung zu identifizieren, werden in Abschnitt 3.1.1 die etablierten Standards im Testbereich des ISTQB und der IEEE in Bezug auf das Thema Testaufwandsschätzung untersucht. Anschließend wird in Abschnitt 3.1.2 die von dem ISTQB vorgeschlagene Methode Testpunktanalyse vorgestellt [Spi11, SL10, Int11, Int12c]. Darauf folgend wird in Abschnitt 3.1.3 die Methode Test Case Points erläutert, da sie von demselben Unternehmen wie die Methode Testpunktanalyse entwickelt worden ist.

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

In den Standards des ISTQB und der IEEE wird neben der Methode Testpunktanalyse auf Methoden zur Expertenschätzung verwiesen [Spi11, SL10, IEE11a]. Daher wird in Abschnitt 3.1.4 am Beispiel der Expertenschätzmethode des Unternehmens CRM IT eine mögliche Umsetzung der Breitband-Delphi Methode diskutiert. Dies ist eine Detaillierung des Vorgehens, das bei der Definition der Problemstellung in Abschnitt 1.1 beschrieben worden ist.

Da über die Standards des ISTQB und der IEEE nur zwei Methoden identifizierbar gewesen sind, wird in Abschnitt 3.1.5 eine systematische Literaturrecherche nach weiteren Methoden zur Testaufwandsschätzung vorgestellt. Abschließend wird in Abschnitt 3.1.5.4 zusammengefasst, warum die Entwicklung einer neuen Methode zur Testaufwandsschätzung notwendig ist.

#### **3.1.1. Berücksichtigung der Testaufwandsschätzung in Standards**

Im Bereich des Testens existieren de facto Standards, wie der des ISTQB oder die der IEEE. In diesem Abschnitt wird dargestellt, wie das Thema Testaufwandsschätzung in diesen Standards berücksichtigt wird und welche Methoden vorgeschlagen werden. Dazu wird zunächst auf den ISTQB Standard und anschließend auf die Vorschläge der IEEE eingegangen.

Im Glossar des ISTQB wird für eine Aufwandsschätzung im Testbereich der Begriff Testschätzung verwendet [GTB10]. Der Begriff ist in dem Glossar, wie nachfolgend definiert:

„Ermittelte Näherung eines Ergebnisses zu einem Aspekt des Testens (z. B. Aufwand, Endzeitpunkt, erforderliche Kosten, Anzahl der Testfälle usw.), das auch dann brauchbar ist, wenn die Eingabedaten unvollständig, unsicher oder gestört sind.“ [GTB10]

Vorschläge für Methoden zur Testaufwandsschätzung sind die Testpunktanalyse (vgl. Abschnitt 3.1.2) und die Breitband-Delphi Methode (vgl. Abschnitt 2.4.2). In Veröffentlichungen zur Vorbereitung auf die Zertifizierung durch das ISTQB werden die Vorschläge aufgegriffen und erläutert [Spi11, SL10, Int11, Int12c]. Dabei wird auch grundlegend darauf eingegangen, wie bei der Schätzung der Testaufwände *bottom-up* vorgegangen werden kann. Darüber hinausgehend gibt

### 3.1. Methoden zur Testaufwandsschätzung

es Empfehlungen, welche Einflussfaktoren in einer Schätzung zu berücksichtigen sind und welche durch den Testmanager beeinflussbar sind [SL10, S. 186-187][Int12c, S. 37]. Eine eigene Methode oder ein Ansatz zur Testaufwandsschätzung wird nicht beschrieben, sondern der Testmanager muss aus den Erläuterungen ein eigenes Vorgehen ableiten.

Von der IEEE gibt es verschiedene Standards, die den Bereich des Testens betreffen. In dem *IEEE Standard for Software Unit Testing* wird unter dem Punkt *Determine Resource Requirements* kurz darstellt, was zu einer Aufwandsschätzung gehören kann [IEE86, S. 5]. Dabei wird keine Methode zur Testaufwandschätzung vorgestellt und auch auf keine verwiesen.

Der *IEEE Standard 829 for Software and System Test Documentation* beschreibt Anforderungen an die Testdokumentation und auch einen Testprozess inklusive Aktivitäten [IEE08]. Hier wird an mehreren Stellen darauf eingegangen, dass eine Schätzung des Aufwands erfolgen soll. Speziell die Aktivität für die Akquisition enthält Hinweise auf die Schätzung von Testaktivitäten [IEE08, S. 76]. Auch hier wird nur ein allgemeines Vorgehen vorgestellt und kein Hinweis auf eine Methode gegeben. Die *Industry Implementation of International Standard ISO/IEC 12207* definiert, dass ein Plan u. a. eine Schätzung, eine Ressourcenbelegung, die Risiken und eine Zeitplanung enthalten sollte [IEE98b, S. 12]. Dabei wird allerdings kein Format oder Vorgehen vorgegeben.

In *a Guide to the Project Management Body of Knowledge (PMBOK Guide)*<sup>1</sup> werden Standardbegriffe und Richtlinien für Projektmanagement definiert [IEE11a]. Das PMBOK hat keinen direkten Bezug zum Testen, da die Aktivität Testmanagement aber mit Projektmanagement vergleichbar ist, sind viele Abläufe ähnlich. Im PMBOK wird in Kapitel 7 auf das Management von Projektkosten eingegangen. Dabei wird neben der Kontrolle und Feststellung des Budgets auch auf die Kostenschätzung eingegangen. Für die Durchführung der Kostenschätzung werden verschiedene Methoden kurz vorgestellt. Dazu zählen u. a. Expertenschätzungen, Analogie Schätzungen und *bottom-up* Schätzungen. Der Fokus des PMBOK liegt allerdings nicht darin, die einzelnen Methoden detailliert zu erläutern, sondern die Prozesse für das Management der Projekt-

---

<sup>1</sup>Der *PMBOK Guide* stammt von dem *Project Management Institut* und wurde von der IEEE und dem *American National Standards Institute* als Standard aufgenommen.

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

kosten mit Ein- und Ausgabeartefakten und der Eingliederung in den Gesamtprozess des Projektmanagements darzustellen.

Zusammenfassend hat sich bei der Recherche in den vorhandenen Standards im Bereich der Qualitätssicherung gezeigt, dass nur sehr oberflächlich auf das Thema Testaufwandsschätzung eingegangen wird. Die einzigen Methoden, die empfohlen werden, sind die Testpunktanalyse und die Expertenschätzungen, wie die Breitband-Delphi Methode.

#### 3.1.2. Testpunktanalyse

Die Methode Testpunktanalyse wurde zuerst von van Aalst in der Veröffentlichung *Testpointanalysis: a method for test estimation* beschrieben [vD99]. Mittlerweile ist die Methode in den Standard TMap der Firma Sogeti Nederland B.V. übernommen worden und wird in der zugehörigen Literatur ausführlich erläutert [vBKV08, S. 456ff]. Die Idee der Methode ist, dass ein Zusammenhang zwischen dem funktionalen Umfang eines Softwareproduktes und dem Testaufwand existiert. Voraussetzung für die Anwendung ist ein vorliegendes Ergebnis der Methode Function Point (vgl. Abschnitt 2.4.3.1). Aus diesem Ergebnis wird zusammen mit Einflussgrößen aus den Bereichen Produktivität und Teststrategie über die Methode Testpunktanalyse der Testaufwand berechnet. Für weitere Informationen eignet sich insbesondere die Umsetzungen der Methode in Excel [Sogb, Soga].

In Abbildung 3.1 ist eine Übersicht über den Ablauf der Methode Testpunktanalyse zu sehen. Im Kern besteht die Methode aus drei Schritten:

1. Bestimmung der dynamischen und statischen Testpunkte und die darauf basierende Berechnung der gesamten Testpunkte (TP)
2. Bestimmung des Umgebungs- und Kompetenzfaktors (UF, KF) und die Berechnung der primären Teststunden (PTS) aus TP, UF und KF
3. Bestimmung von zusätzlich notwendigen Stunden und die Summierung dieser Stunden mit den PTS

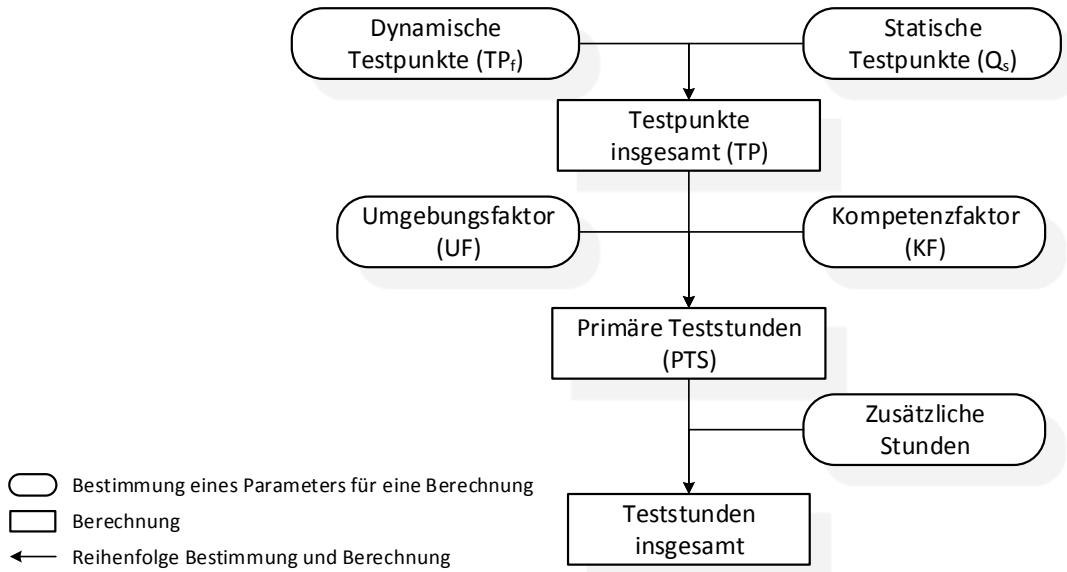


Abbildung 3.1.: Schematische Darstellung der Methode Testpunktanalyse nach [vBKv08, S. 459]

### Schritt 1: Bestimmung der dynamischen und statischen Testpunkte

Im ersten Schritt werden zunächst die dynamischen Testpunkte bestimmt. Dafür wird auf das Ergebnis der Methode Function Point und die funktionale Spezifikation zurückgegriffen. Für jede Funktion  $f$ , die in der funktionalen Spezifikation beschrieben ist, werden die zugehörigen Funktionspunkte ( $FP_f$ ) ermittelt. In der Methode Function Point wurden diese Funktionen in Grundfunktionen zerlegt und der Umfang der Grundfunktionen in AFP gemessen.

Die  $FP_f$  werden für jede Funktion  $f$  in zwei Teilen bestimmt. Den ersten Teil bilden die Summe der Grundfunktionen der Funktion  $f$  in AFP. Der zweite Teil ist die Summe der von den Grundfunktionen verwendeten internen Daten und Schnittstellen in AFP. Ein Beispiel wäre die Funktion Datenerfassung, die über die Grundfunktionen Erstellen, Aktualisieren und Löschen in der Methode Function Point berücksichtigt worden ist. Außerdem haben diese Grundfunktionen auf eine Datei zur Datenhaltung zurückgegriffen. Damit sind die  $FP_f$  für die Funktion Datenerfassung die AFP für das Erstellen, das Aktualisieren, das Löschen und die Datenhaltung.

Nach der Bestimmung der  $FP_f$  wird für jede Funktion  $f$  ein individueller Einflussfaktor  $A_f$  berechnet. Dieser Einflussfaktor dient der Abbildung von Ein-

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

Kürzel Beschreibung		Skala	Gewicht
Br	Benutzerrelevanz	niedrig	3
		neutral	6
		hoch	12
Ni	Nutzungsintensität	niedrig	2
		neutral	4
		hoch	8
As	Auswirkung	geringe	2
		mittlere	4
		hohe	8
C	Komplexität	geringe	3
		mittlere	6
		hohe	12
E	Einheitlichkeit	Ja	0,6
		Nein	1

Tabelle 3.1.: Funktionsabhängige Faktoren zur Berechnung des Einflussfaktors  $A_f$  nach [vBKV08]

flüssen, die sich direkt auf einzelne Funktionen beziehen. Dies kann z. B. die Nutzungsintensität sein. In Tabelle 3.1 sind die Faktoren für die Berechnung von  $A_f$  aufgeführt. Dazu existieren in der Literatur weitere Metriken und Hilfen zur Interpretation des Ergebnisses der Metriken für die Einordnung auf der gegebenen Skala [vBKV08]. Teilweise sind die Hilfen natürlichsprachlich formuliert und unterliegen damit einer subjektiven Interpretation. Ein Beispiel dafür ist die Stufe *neutral* für die *Benutzungsintensität*: „Die Funktion wird im Benutzerunternehmen viele Male am Tag ausgeführt“ [vBKV08, S. 464]. Nach der Bestimmung der einzelnen Faktoren wird der Einflussfaktor  $A_f$  über die Formel 3.1 berechnet.

$$A_f = \frac{((Br + Ni + As + C)) \cdot E}{20} \quad (3.1)$$

Neben dem individuellen Einflussfaktor  $A_f$  wird für die Bestimmung der dynamischen Testpunkte der Einflussfaktor  $Q_d$  berechnet. Dieser Einflussfaktor wird in der Regel einmal für das Gesamtsystem berechnet. Es besteht aber die Möglichkeit für jede Funktion oder Gruppen von Funktionen einen eigenen Einflussfaktor  $Q_d$  zu berechnen. In der Tabelle 3.2 sind die Faktoren für die Berechnung mit dem zugehörigem Gewicht aufgeführt. Die Faktoren repräsentieren zu

### 3.1. Methoden zur Testaufwandsschätzung

messende Qualitätsmerkmale. Wenn diese Qualitätsmerkmale explizit während der Testausführung gemessen werden sollen, werden sie je nach der Höhe der Qualitätsanforderungen nach unten oder oben skaliert. Die dafür notwendigen Multiplikatoren sind in Tabelle 3.3 zu finden. Falls die Qualitätsmerkmale implizit mitgemessen werden sollen, wird das Merkmal mit 0,02 in  $Q_d$  berücksichtigt. Mit der Formel 3.2 wird der Einflussfaktor  $Q_d$  schließlich berechnet.

$F_i$	Faktor	Gewicht
$F_1$	Funktionalität	0,75
$F_2$	Sicherheit	0,05
$F_3$	Effektivität	0,10
$F_4$	Leistungsfähigkeit	0,05
$F_5$	Übertragbarkeit	0,05

Tabelle 3.2.: Faktoren zur Berechnung des Einflussfaktors  $Q_d$  nach [vBKV08]

Multiplikator $M_i$	Beschreibung
0	nicht wichtig: Qualitätsmerkmal wird nicht gezählt
3	niedrige Qualitätsanforderungen: Im Test ist darauf zu achten
4	normale Qualitätsanforderungen: Trifft gewöhnlich zu, wenn sich das Informationssystem auf einen Unterstützungsprozess bezieht
5	hohe Qualitätsanforderungen: Trifft gewöhnlich zu, wenn sich das Informationssystem auf einen primären Prozess bezieht
6	extrem hohe Qualitätsanforderungen

Tabelle 3.3.: Multiplikatoren zur Berechnung des Einflussfaktors  $Q_d$  nach [vBKV08]

$$Q_d = \sum_{i=1}^n \left( \frac{F_i \cdot M_i}{4} \right) + m \cdot 0,02 \quad (3.2)$$

$n$  := Anzahl explizit zu messender Qualitätsmerkmale

$m$  := Anzahl implizit zu messender Qualitätsmerkmale

Mit der Kenntnis über  $FP_f$ ,  $A_f$  und  $Q_d$  kann die Anzahl der dynamischen Testpunkte für jede Funktion über die Formel 3.3 berechnet werden.

$$TP_f = FP_f \cdot A_f \cdot Q_d \quad (3.3)$$

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

Die statischen Testpunkte  $Q_s$  ergeben sich aus der Anzahl der statisch zu testenden Qualitätsmerkmale. Die Qualitätsmerkmale sind dabei frei wählbar [vBKV08]. Jedes zu testende statische Qualitätsmerkmal fließt mit dem Wert 16 in  $Q_s$  ein, so dass  $Q_s$  immer ein Vielfaches von 16 ist. Eine Begründung für die Zahl 16 fehlt.

Die Gesamtzahl der Testpunkte wird mit Formel 3.4 berechnet. Dabei stellt die Variable  $TP_{f_j}$  die dynamischen Testpunkte für die Funktion  $j$  und die Variable AFP das Ergebnis der Methode Function Point dar. Laut der Literatur zur Testpunktanalyse wurde in der Formel 500 im Nenner verwendet, da 500 der Mindestwert für die Systemgröße ist. Eine Begründung für die Mindestsystemgröße 500 fehlt [vBKV08].

$$TP = \sum_{j=1}^{\# \text{AnzahlFunktionen}} TP_{f_j} + \frac{AFP \cdot Q_s}{500} \quad (3.4)$$

#### Schritt 2: Bestimmung des Umgebungs- und Kompetenzfaktors

Für den Umgebungsfaktor ( $UF$ ) müssen zunächst die Variablen in Tabelle 3.4 für das aktuelle Projekt bewertet werden [vBKV08]. Die Berechnung von  $UF$  erfolgt anschließend nach der Formel 3.5 durch Summieren der gewählten Gewichtungen  $W_i$  je Variable.

$$UF = \frac{\sum_{i=1}^6 W_i}{21} \quad (3.5)$$

Der Kompetenzfaktor ( $KF$ ) könnte auch mit Produktivitätsfaktor beschrieben werden, da er definiert, wie viele Stunden Aufwand pro Testpunkt zu veranschlagen sind. Als Richtwert wird von 1-2 Stunden pro Testpunkt für einen Tester und von 2-4 Stunden pro Testpunkt für einen Benutzer ausgegangen [vBKV08, S. 470]. Dabei wird nicht näher festgelegt, wie ein Tester oder ein Benutzer qualifiziert sein muss. Ein genauer Kompetenzfaktor ist für jedes Unternehmen individuell aufgrund von Daten aus abgeschlossenen Projekten festzulegen.

Mit Formel 3.6 werden die Gesamtzahl der Testpunkte mit den Faktoren KF und UF multipliziert und damit die primären Teststunden errechnet.

### 3.1. Methoden zur Testaufwandsschätzung

$$PTS = TP \cdot KF \cdot UF \quad (3.6)$$

Variable	Gewicht $W_i$	Beschreibung
Testwerkzeuge	1	Unterstützung für Testdesign und Einsatz eines <i>Capture and Replay</i> Werkzeuges
	2	Unterstützung für Testdesign oder Einsatz eines <i>Capture and Replay</i> Werkzeuges
	4	Keine Testwerkzeuge
Vorheriger Test	2	Testfälle und Ergebnisse von vorherigem Test verfügbar
	4	Es ist ein Testplan vorhanden
	8	Kein vorheriger Test
Testbasis	3	Systemdokumentation basiert auf Standards und Vorlagen, regelmäßige Inspektionen
	6	Systemdokumentation basiert auf Standards und Vorlagen
	12	Systemdokumentation basiert nicht auf Standards und Vorlagen
Entwicklungs- umgebung	2	Unterstützung durch zahlreiche Einrichtungen zur Verhinderung von Fehlerzuständen
	4	Unterstützung durch eine begrenzte Zahl an Einrichtungen zur Verhinderung von Fehlerzuständen
	8	Keine Einrichtungen zur Verhinderung von Fehlerzuständen
Testumgebung	1	Mehrmalig zur Testausführung genutzt
	2	Neue Umgebung, aber Erfahrung in der Einrichtung
	4	Neue experimentelle Umgebung
Testware	1	Testdaten und Testfälle sind spezifiziert und verfügbar
	2	Testdaten sind verfügbar
	4	Keine Testware

Tabelle 3.4.: Variablen zur Bewertung des Umgebungsfaktors nach [vBKV08]

### Schritt 3: Bestimmung von zusätzlich notwendigen Stunden

Die PTS umfassen nicht alle Aktivitäten und Einflussfaktoren, die zu einem Testprojekt gehören [vBKV08]. Daher sollte aus der Erfahrung des Schätzers ein Aufschlag zwischen 5 und 20 Prozent zu den PTS berechnet werden. Für die Ermittlung der Höhe des Aufschlages werden in der Literatur u. a. Hinweise auf

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

Teilaktivitäten der Planung und Steuerung, die Einflussgröße Teamgröße und das Vorhandensein einer permanenten Testorganisation gegeben [vBKV08].

Durch Addition der zusätzlichen Stunden und der primären Stunden ergeben sich die Teststunden insgesamt. Diese Teststunden gelten für alle Aktivitäten des Testprojektes und sind deshalb nicht direkt in der Planung von Aktivitäten verwendbar. Von van Aalst wird vorgeschlagen, dass die Verteilung der Teststunden über Aktivitäten und Phasen mit festen Prozentsätzen erfolgt [vBKV08]. Die Höhe dieser Prozentsätze ist dabei von Experten unternehmensspezifisch anzupassen.

### Fazit

Die Vorteile der Methode Testpunktanalyse sind, dass der Fokus auf dem Testaufwand liegt und spezielle Einflussfaktoren für Testaktivitäten berücksichtigt werden. Dies ist speziell bei der Überführung des Ergebnisses der Methode Function Point in ein Maß für den Umfang eines Testprojektes zu sehen. Hier ist individuell für jede Funktion festlegbar, wie wichtig die Funktion für den Endkunden ist oder welche Auswirkungen eine Änderung dieser Funktion hat.

Die Nachteile der Testpunktanalyse sind die Lücken in der Definition der Methode, die fehlenden empirischen Belege und die große Komplexität in der Anwendung. Für viele Einflussfaktoren sind die Metriken und Hilfen zur Interpretation des Ergebnisses der Metrik nicht formalisiert und unzureichend beschrieben. Ein Beispiel hierfür ist der Satz „Die relative Bedeutung der Funktion im Vergleich zu den anderen Funktionen ist hoch.“ [vBKV08, S. 463]. Aus diesem Satz wird nicht deutlich, wie die *Bedeutung* einer Funktion zu bewerten ist und wie *hoch* definiert ist. Damit sinkt die Nachvollziehbarkeit und Vergleichbarkeit von Testaufwandsschätzungen mit der Testpunktanalyse, da die Metriken von unterschiedlichen Anwendern abweichend interpretiert und angewendet werden.

In der Literatur zur Testpunktanalyse fehlen empirische Belege für die Wahl der Einflussfaktoren und deren Gewichtung. Im Allgemeinen gibt es auch keine Hinweise von Unternehmen, die die Methode Testpunktanalyse erfolgreich einsetzen. Dies ist besonders aufgrund der langen Historie der Methode seit mindestens 1999 auffällig.

Die Überführung des Ergebnisses der Function Point Methode ist sehr komplex, da bei der Überführung alle Details der funktionalen Spezifikation noch einmal aufgegriffen und ggf. neu bewertet werden müssen. Erschwerend fehlt für die Überführung des Ergebnisses der Function Point Methode eine genaue Definition, wie welche Elemente zu überführen sind. Hier bleibt ein großer Interpretationsspielraum für den Anwender der Testpunktanalyse bestehen. Damit sind die Schätzungen von unterschiedlichen Anwendern kaum vergleichbar.

Durch die Methode Testpunktanalyse werden die Problemstellungen 1 und 2 aus Abschnitt 1.1 nicht gelöst, da die Erfassung der Testkomplexität und die Beschreibung sowie Nutzung von Testeinflussfaktoren nicht ausreichend formalisiert sind. Auf eine Bewertung der den Problemstellungen zugehörigen Anforderungen wird wegen der direkten Nichterfüllung der Problemstellung abgesehen. Insgesamt ist die Methode Testpunktanalyse keine Lösung für die Problemstellungen dieser Dissertation.

#### 3.1.3. Test Case Points

Die Methode Test Case Points wird in einem internen Papier der Firma Sogeti und verschiedenen Fachvorträgen beschrieben [Nel12, Van09, Baa09]. Die Idee der Methode ist, dass ein Zusammenhang zwischen der Komplexität der zu testenden Software und der Größe des Testprojektes besteht. Über diesen Zusammenhang ist die Größe des Testprojektes schätzbar, wenn die Komplexität der Software bekannt ist.

Die Komplexität der zu testenden Software wird in der Methode Test Case Points als die Anzahl Pfade und der Entscheidungspunkte in Ablaufdiagrammen (engl. *Flow Charts*<sup>2</sup>), die die internen Abläufe in der Software abbilden, aufgefasst. Mit dieser Definition der Komplexität besteht eine Verbindung zu der Anzahl der Testfälle, die implizit wieder Teil der Größe eines Testprojektes sind [Nel12]. Basierend auf der berechneten Größe des Testprojektes ist beispielsweise mit einer Analogie-Schätzung oder einer direkten Umrechnung über einen Produktivitätsfaktor der Testaufwand schätzbar.

Die Anwendung der Methode Test Case Points erfolgt in vier Schritten:

---

<sup>2</sup>*Flow Charts* sind eine standardisierte Form Programmabläufe zu visualisieren [ISO85].

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

1. Bestimmung der Anzahl Pfade anhand von Ablaufdiagrammen (FL)
2. Bestimmung und Gewichtung der Anzahl Entscheidungspunkte in den Ablaufdiagrammen (DP)
3. Bestimmung der Anzahl Schnittstellenpunkte (IP)
4. Berechnung der Größe des Testprojekts in der Einheit Test Case Points (TCP)

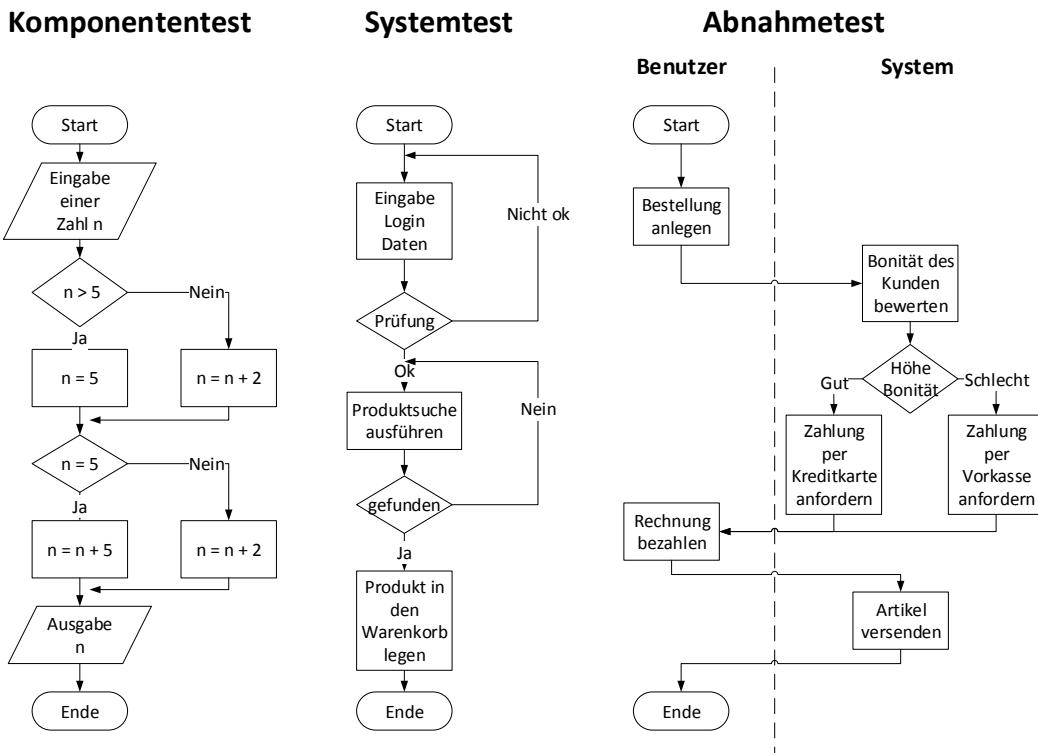


Abbildung 3.2.: Unterschiedliche Abstraktionsgrade in der Modellierung mit Ablaufdiagrammen

Der erste Schritt zur Bestimmung der Anzahl der Pfade (FL) in den Ablaufdiagrammen ist mittels einer beliebigen Technik durchführbar. Die Voraussetzung ist, dass die Abläufe der zu testenden Software mit einem entsprechenden Abstraktionsgrad modelliert sind. Dieser muss der Teststufe entsprechen, für die die TCP ermittelt werden sollen.

In Abbildung 3.2 sind drei unterschiedliche Ablaufdiagramme zu sehen, die sich jeweils auf einem anderen Abstraktionsgrad befinden. Links das Ablaufdiagramm beschreibt den Ablauf einer Programmfunktion, das mittlere einen

### 3.1. Methoden zur Testaufwandsschätzung

Teil eines Geschäftsprozesses und das rechte Diagramm einen gesamten Geschäftsprozess. Diese Ablaufdiagramme können entsprechend ihres Abstraktionsniveaus den Teststufen Komponenten-, System-, und Abnahmetest zugeordnet werden.

Im zweiten Schritt der Methode Test Case Points werden die Entscheidungspunkte in den Ablaufdiagrammen gezählt und jeweils anhand der Anzahl Bedingungen gewichtet. Dabei werden die drei Kategorien einfach, mittel und schwer unterschieden. Die den Kategorien zugehörigen Gewichtungen sind in Tabelle 3.5 aufgelistet. Die Summe der Gewichte der einzelnen Entscheidungspunkte stellt die Anzahl DP dar.

Im dritten Schritt wird die Anzahl der Schnittstellen zu anderen Systemen gezählt (IP), die nicht Teil der aktuellen Betrachtung sind. Dies umfasst jede Schnittstelle, über die Daten ausgetauscht wird. Teilweise werden die Schnittstellenpunkte auch als *enrichment point* bezeichnet [Van09]. Im Gegensatz zu den Entscheidungspunkten werden Schnittstellenpunkte nicht gewichtet.

Kategorie	Anzahl Bedingungen	Gewichtung
Einfach	1	1
Mittel	$\leq 3$	3
Schwer	$> 3$	5

Tabelle 3.5.: Gewichtung der Entscheidungspunkte nach [Nel12]

Im vierten Schritt werden die FL, DP und IP nach Formel 3.7 addiert. Das Ergebnis ist eine natürliche Zahl in der Einheit TCP. Diese Zahl ist noch nicht proportional zum Testaufwand, da die Teststrategie für jedes Testobjekt anders ist [Nel12]. Um die Proportionalität zu gewährleisten, wird vorgeschlagen, die Teststrategie über die Einführung von Risikoklassen zu berücksichtigen. Dazu werden den Ablaufdiagrammen Risikoklassen zugeordnet [Nel12]. Damit können die Anzahl der TCP für ein Ablaufdiagramm um bis zu 40 % erhöht oder um bis zu 15 % verringert werden.

Die TCP, die für eine bestimmte Teststufe berechnet worden sind, können in TCP für eine andere Teststufe umgerechnet werden. Dabei wird jeweils um den Faktor 4 multipliziert oder entsprechend dividiert, wenn die TCP für die nächst höhere oder niedrigere Teststufe auszurechnen sind.

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

$$TCP = FL + DP + IP \quad (3.7)$$

Nach der Ausführung der Schritte eins bis vier ist der ermittelte TCP Wert über eine Analogie Schätzung oder das Multiplizieren mit einem Produktivitätsfaktor zu Testaufwand umzurechnen. Die Voraussetzung ist, dass entsprechende Daten von abgeschlossenen Projekten vorliegen.

#### Fazit

Der Vorteil der Methode TCP ist die einfache Anwendbarkeit, da wenige Elemente gezählt und bewertet werden müssen. Damit ist gleichzeitig der Aufwand für die Durchführung der Methode beim Vorliegen eines Ablaufdiagramms sehr gering und es können sehr schnell über Vergleiche mit abgeschlossenen Projekten Aussagen zum erwarteten Testaufwand getroffen werden.

Die Nachteile der Methode TCP sind die fehlenden empirischen Nachweise über die Gewichtungen der Entscheidungspunkte, der Risikogruppen und des Umrechnungsfaktors für die Teststufen. Auch fehlen Vorschläge wie die Methode TCP auf verbreitete Spezifikationstechniken wie BPMN und Use Cases anzuwenden ist. Allgemein sind keine Hinweise vorhanden, dass Unternehmen diese Methode erfolgreich in der Praxis einsetzen.

Mit der Methode wird nur die Problemstellung 1 aus Abschnitt 1.1 adressiert, da keine Testeinflussfaktoren berücksichtigt werden und auch kein direkter Vorschlag zur Berechnung des Testaufwandes aus Testeinflussfaktoren und Testkomplexität gegeben wird. Auf eine Bewertung der den Problemstellungen zugehörigen Anforderungen wird wegen der direkten Nichterfüllung der Problemstellung 2 und 3 abgesehen. Insgesamt ist die Methode TCP keine Lösung für die Problemstellungen dieser Dissertation.

#### 3.1.4. Expertenschätzung bei dem Unternehmen CRM IT

Eine Testaufwandsschätzung findet bei QM entweder im Rahmen eines kaufmännischen Angebotes für ein IT-Projekt oder auf Anfrage als ein separates Angebot. In jedem Fall wird zwischen einer Kostenindikation und einer Kosten-

### 3.1. Methoden zur Testaufwandsschätzung

schätzung unterschieden. Der Unterschied zwischen Indikation und Kostenschätzung liegt im Detaillierungsgrad der Schätzung, wobei eine Kostenindikation ungenauer als eine Kostenschätzung ist. Die Gründe für die höhere Ungenauigkeit können in dem Zeitpunkt der Schätzung oder auch im verfügbaren Budget für die Durchführung der Schätzung begründet sein.

Bei der Abteilung für Qualitätssicherung QM in dem Unternehmen CRM IT werden Expertenschätzungen vorgenommen, um den Testaufwand in Personen-tagen (PT) zu ermitteln. Dabei entspricht ein Personentag 8 Arbeitsstunden. In den meisten Fällen findet dies in Form einer Einzelschätzung, die *bottom-up* vorgenommen wird, statt. Das Ergebnis der Einzelschätzung wird im Anschluss durch einen zweiten Experten validiert. Andere Methoden der Expertenschätzung, wie die Delphi oder die Breitband-Delphi (vgl. Abschnitt 2.4.2) Methode, werden nicht angewendet. Für die Durchführung der Schätzung gibt es kein standardisiertes Vorgehen, aber es besteht im Kern unabhängig von der zu schätzenden Teststufe immer aus den folgenden Schritten:

1. Bewertung des funktionalen Umfangs und Abgrenzung der Testobjekte
2. Bewertung des Aufwands für jedes Testobjekt aufgeschlüsselt nach den Testaktivitäten Testmanagement, Testdesign, Testentwicklung<sup>3</sup> und Testausführung
3. Bewertung von festen Aufwänden
4. Bewertung des Risikos

Im ersten Schritt stellt der Auftraggeber zunächst Informationen über das IT-Projekt bereit. Zu den Informationen gehört mindestens eine fachliche Spezifikation der zu testenden Software. Diese Spezifikation ist oft mehrere hundert Seiten stark und zum vollständigen Lesen ist für einen Testmanager oft keine Zeit vorhanden. Daher kann der Testmanager von dieser unübersichtlichen Basis nicht direkt den Aufwand für Testaktivitäten ableiten, sondern muss diese Basis erst in kleinere Einheiten (Testobjekte) zerlegen. Die Ableitung aus der fachlichen Spezifikation findet oft mit Unterstützung weiterer fachlicher An-

---

<sup>3</sup>Der Aufbau und die Pflege einer Testautomatisierung werden mit dem Begriff Testentwicklung umfasst. Praktisch ist die Testentwicklung eine Unteraktivität von der Aktivität Testdesign. Da sie in der Vorlage für die Testaufwandsschätzung bei QM separat ausgewiesen wird, wird sie hier mit aufgeführt.

### *3. Verwandte Arbeiten und Überblick über den Lösungsansatz*

sprechpartner statt. Wenn schon Systemspezifikationen zur Verfügung stehen oder Systeme von Dritten integriert werden sollen, müssen der IT-Manager und die externen Zulieferfirmen mit einbezogen werden. Am Ende des ersten Schrittes hat der Testmanager eine Liste mit Testobjekten erstellt, die er bezüglich des Aufwands der Testaktivitäten einschätzen kann. Mögliche Testobjekte sind hier Dateischnittstellen, Webservicemethoden und Use Cases.

Im zweiten Schritt bewertet der Testmanager mittels Analogie-Schätzung oder aufgrund von Erfahrungswerten die einzelnen Testobjekte hinsichtlich ihres Aufwands. Der Aufwand wird hierbei immer nach den Testaktivitäten Testmanagement, Testdesign, Testentwicklung und Testausführung aufgeschlüsselt. In der Regel kommen hier auch Pauschalen zum Einsatz, wie dass der Aufwand für Testmanagement 15 % von dem Aufwand für die anderen Aktivitäten beträgt.

Im dritten Schritt werden feste Kosten eingetragen, die weitgehend unabhängig vom Umfang des Testprojektes sind. Dazu gehören z. B. das Erstellen der Testplanung, das Schreiben des Abschlussberichtes oder das Einrichten des Fehlermanagementsystems. In der Regel wird an dieser Stelle auch geprüft, wie viele Tage für die Testausführung einzuplanen sind und jeder Tag mit einer Pauschale in die festen Kosten eingerechnet.

Im letzten Schritt wird das Risiko pauschal für das Projekt festgelegt. Dies geschieht aus der Erfahrung des Testmanagers mit ähnlichen Projekten und liegt meistens im Bereich von 10 % bis 20 %. Das Risiko gilt in der Regel für alle Testaktivitäten gleichermaßen.

Unterstützt werden die Schritte für die Durchführung der Testaufwandsschätzung bei QM durch eine weitgehend standardisierte Excel-Vorlage. Die erste Reiterseite der Vorlage ist in Abbildung 3.3 zu sehen. In der oberen Tabelle werden getrennt nach Teststufen (Zeilen) die Aufwände für die Testaktivitäten (Spalten) ausgewiesen. Die farbliche Hinterlegung sagt aus, welche Aufwände durch QM selbst (Grün), welche optional durch QM (Gelb) und welche durch den Kunden (Rot) zu leisten sind. In der zweiten Tabelle werden passend zu der farblichen Hinterlegung in der oberen Tabelle die Summen der Aufwände ausgewiesen. Auf der rechten Seite der Tabelle wird der Aufwand mit dem weiter unten stehenden Risiko von 20 % ausgewiesen.

### 3.1. Methoden zur Testaufwandsschätzung

Unter den Tabellen befinden sich Hinweise zu Begriffen und eine Liste von Annahmen, die der Schätzung zu Grunde liegen. Beispiele sind hier das explizite Ausklammern der Teststufe Modultest und der Testunterstützungsaufwände durch die IT Abteilungen. Diese Liste enthält in ihrer endgültigen Form in der Regel wesentlich mehr Annahmen.

	Testmanagement	Testdesign	Testentwicklung	Testdurchführung
<b>Gesamt</b>				
Integrationstest (Schnittstellen)	4,7 PT	6,0 PT	0,0 PT	2,0 PT
Systemtest	4,9 PT	3,0 PT	0,0 PT	3,0 PT
Abnahmetest	4,2 PT	2,0 PT	0,0 PT	6,0 PT
Effizienztest	3,6 PT	3,0 PT	0,0 PT	1,0 PT
Sicherheitstest	3,2 PT	0,0 PT	0,0 PT	1,0 PT
<i>Summe</i>	<b>20,6 PT</b>	<b>14,0 PT</b>	<b>0,0 PT</b>	<b>13,0 PT</b>
	Kostenindikation	Risiko	Gesamt inkl. Risiko	
Summe Aufwände, die von TS-QAM geleistet werden sollten	32,6 PT	6,5 PT		39,1
Summe Aufwände, die von TS-QAM geleistet werden können	5,0 PT	1,0 PT		6,0
Summe Aufwände, die vom Kunden geleistet werden müssen	10,0 PT	2,0 PT		12,0
<i>Summe exkl. Aufwand Fachbereich</i>	<b>37,6 PT</b>	<b>7,5 PT</b>	<b>45,1 PT</b>	
<i>Summe Gesamt</i>	<b>47,6 PT</b>	<b>9,5 PT</b>	<b>57,1 PT</b>	
<b>Hinweise:</b>				
Integrationstest ist ein Test der Schnittstellen.				
Funktionaler Test ist ein Test der einzelnen Geschäftsprozesse.				
Abnahmetest ist ein integrativer Test über mehrere Geschäftsprozesse.				
Risiko		20,00%		
<b>Annahmen:</b>				
<b>Es wird eine Business Modellierung zur Verfügung stehen</b>				
Die Teststufe Modultest ist nicht Bestandteil dieser Schätzung und obliegt der Entwicklung				
Die Testunterstützungsaufwände der IT Abteilungen sind nicht berücksichtigt				
Die Anzahl der Testszenarien für den Abnahmetest ist abhängig von Geschäftsprozessketten und gewünschter Abdeckung				

Abbildung 3.3.: Vorlage zur Erstellung einer Testaufwandsschätzung bei der Abteilung QM

Die einzelnen Teststufen werden in der Vorlage auf separaten Reiterseiten behandelt. Verallgemeinert besteht jede Reiterseite aus einer Liste der identifizierten Testobjekte. In der Liste wird für jedes Testobjekt der Aufwand aufgeschlüsselt nach Testaktivitäten aufgeführt. Zur Unterstützung des Ausfüllen der Vorlage durch einen Testmanager werden für die Teststufen und Testaktivitäten Pauschalen vorgegeben. Diese Pauschalen stellen dabei nur einen Ausgangswert dar und müssen aus der Erfahrung des Testmanagers für den jeweiligen Projektkontext angepasst werden.

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

#### Fazit

Der Vorteil der Expertenschätzung bei dem Unternehmen CRM IT ist, dass sie durch das *bottom-up* Vorgehen gut skalierbar mit der gewünschten Schätzgenauigkeit ist. Wenn wenig Zeit oder keine detaillierte fachliche Spezifikation zur Verfügung steht, können die Testobjekte auf einem höheren Abstraktionsniveau abgegrenzt werden. Ein Beispiel ist die Betrachtung einer Webserviceschnittstelle als Testobjekt anstatt einer Webservicemethode. Durch das *bottom-up* Vorgehen gleichen sich Unter- und Überschätzungen für einzelne Testobjekte im Mittel sehr gut aus und es wird eine hohe Schätzgenauigkeit erreicht. Weiterhin wird durch die Nutzung der Vorlage und der Berücksichtigung von festen Kosten dem Fehler der Vernachlässigung von übergreifenden Aufwänden bei *bottom-up* Vorgehen entgengewirkt.

Die Nachteile der Methode wurden bei der Definition der Problemstellung in Abschnitt 1.1 beschrieben. Vor allem fehlen eine Formalisierung der Erfassung der Testkomplexität, der Testeinflussfaktoren und der Berechnung des Testaufwandes. Für den Auftraggeber aber auch für einen zweiten Testmanager ist die Nachvollziehbarkeit einer über diese Methode erstellten Testaufwandsschätzung nur eingeschränkt möglich.

#### 3.1.5. Weitere Methoden zur Testaufwandsschätzung

Die weiteren Methoden zur Testaufwandsschätzung wurden im Rahmen einer systematischen Literaturrecherche identifiziert, da über die Standards des IST-QB und der IEEE nur zwei Methoden zur Testaufwandsschätzung identifizierbar gewesen sind (vgl. Abschnitt 3.1.1). Die Durchführung der Literaturrecherche erfolgt nach der Beschreibung von Kitchenham in dem Papier *Procedures for Performing Systematic Reviews* [Kit04, KC07]. Der Vorteil der systematischen Literaturrecherche nach Kitchenham sind u. a. die Nachvollziehbarkeit und die Reproduzierbarkeit der Ergebnisse.

Im Folgenden wird die Literaturrecherche zusammengefasst. Dabei folgt die Struktur der Zusammenfassung der Empfehlung von Kitchenham für die Einbindung einer Literaturrecherche in eine Dissertation [Kit04, S. 22f]. Zunächst wird in Abschnitt 3.1.5.1 erläutert welche Forschungsfragen der Ausgangs-

### 3.1. Methoden zur Testaufwandsschätzung

punkt für die Literaturrecherche waren und welche Methoden für die Literaturrecherche angewendet wurden. Anschließend werden in Abschnitt 3.1.5.2 die Ergebnisse der Literaturrecherche dargestellt. Diese werden im Anschluss in Abschnitt 3.1.5.3 aufgegriffen und hinsichtlich der Forschungsfragen evaluiert. Abschließend wird in Abschnitt 3.1.5.4 ein Fazit unter Bezug auf die in Abschnitt 1.1 formulierte Problemstellung und die in Kapitel 2 definierten Anforderungen gezogen.

#### 3.1.5.1. Forschungsfragen und Strategie für die Literaturrecherche

Die Forschungsfragen für die Literaturrecherche sind eine Verfeinerung der drei Problemstellungen aus Abschnitt 1.1:

1. Wie ist die Erfassung der Testkomplexität in bestehenden Methoden zur Testaufwandsschätzung formalisiert?
2. Wie sind die Beschreibung und die Nutzung von Testeinflussfaktoren in bestehenden Methoden zur Testaufwandsschätzung formalisiert?
3. Wie ist die Berechnung des Testaufwands aus Testkomplexität und Testeinflussfaktoren in bestehenden Methoden zur Testaufwandsschätzung formalisiert?

Zusammenfassend sind für die Literaturrecherche Methoden zur Testaufwandschätzung wichtig, deren Anwendung in großen Teilen formalisiert ist.

Die Strategie der Literaturrecherche besteht aus einer manuellen Suche mit Schlüsselwörtern in digitalen Bibliotheken und dem Internet. Die genutzten digitalen Bibliotheken waren: *Google Scholar* [Goob], *Springer* [Spr], *ACM Digital Library* [ACM] und *IEEE Xplore* [IEE]. Für die Suche im Internet ist die Suchmaschine *Google* [Gooa] verwendet worden.

Die Definition der Schlüsselwörter ist auf Deutsch und Englisch in zwei Kategorien erfolgt: *Bezeichnung der Methode* und *Eigenschaft der Methode*. Eine Übersicht der definierten Schlüsselwörter ist in Tabelle 3.6 zu sehen. Die Schlüsselwörter der Kategorie *Bezeichnung der Methode* werden ausgehend von *Methode zur Testaufwandsschätzung* gewählt. Beispiele hierfür sind *Methode zur Schätzung des Testaufwands* oder *test effort estimation method*.

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

Die Schlüsselwörter der Kategorie *Eigenschaft der Methode* sind ausgehend von den oben genannten Forschungsfragen definiert worden. Ergänzend wurde die zugehörigen in Kapitel 2 definierten Anforderungen genutzt. Die Schlüsselwörter *Formalisierung der Testkomplexität* oder *Algorithmisches Modell* sind entsprechende Beispiele.

Kategorie	Schlüsselwort
	Bezeichnung der Methode Testaufwandsschätzung, Schätzung des Testaufwands, Testschätzung, <i>test effort estimation</i> , Kostenschätzung des Tests, <i>test cost estimation</i> , Berechnung des Testaufwands, <i>test effort calculation</i> , <i>testing effort</i>
Eigenschaft der Methode	UML, Qualifikation, Fähigkeit, Erfahrung, <i>qualification</i> , <i>experience</i> , Formal, multiplikative Form, <i>multiplicative formula</i> , Aktualisierbare Testeinflussfaktoren, <i>updatable cost drivers</i> , <i>Top-Down</i> , Aktualisierbar, <i>updatable</i> , Aufschlüsselung nach Testaktivitäten, <i>break down of test activities</i> , Systemtest, System Test, Systemintegrationstest, Integrationstest, Integration Test, Abgeschlossene Projekte, <i>completed projects</i> , Historische Projekte, <i>previous projects</i>

Tabelle 3.6.: Schlüsselwörter für die Literaturrecherche

Für die während der Suche über die Schlüsselwörter gefundenen Literaturquellen wird anhand von Ein- und Ausschlusskriterien entschieden, ob sie in das Ergebnis der Literaturrecherche aufgenommen werden. Dabei definieren die Einschlusskriterien, wann eine Literaturquelle weiter berücksichtigt werden soll. Analog bestimmen die Ausschlusskriterien die Gründe, warum eine Literaturquelle nicht in das Ergebnis der Literaturrecherche aufgenommen werden sollte. Die Einschlusskriterien für diese Literaturrecherche sind:

- Es werden nur Literaturquellen aus dem Bereich Informatik berücksichtigt, da diese Dissertation auch im Bereich Informatik geschrieben und veröffentlicht werden soll.
- Es werden nur Literaturquellen in deutscher und englischer Sprache berücksichtigt.
- Es werden nur frei zugängliche und für die Universität Paderborn kostenlose Literaturquellen verwendet. Da die Universität zu einer großen Anzahl

an digitalen Bibliotheken freien Zugang bietet, werden keine kostenpflichtigen Quellen verwendet. Speziell die oben genannten Bibliotheken sind für die Universität Paderborn kostenlos nutzbar.

- Es werden nur Literaturquellen berücksichtigt, die für die Domäne Qualitätssicherung veröffentlicht worden sind. Da der Rahmen der in Abschnitt 1.1 definierten Problemstellung der Softwaretest ist, ist von Literaturquellen in der Domäne Qualitätssicherung der höchste Mehrwert zu erwarten.
- Es wurden nur Literaturquellen berücksichtigt, die eine Methode zur Testaufwandsschätzung beschreiben. Da in dieser Dissertation eine Methode zur Testaufwandsschätzung entwickelt werden soll, werden Arbeiten zur Fehlervorhersage, der Untersuchung von Softwarezuverlässigkeit, dem Optimieren von Teststrategien, der Reduzierung von Testaufwand und dem Priorisieren von Tests nicht betrachtet.

Ausschlusskriterien für die Literaturrecherche sind:

- Literaturquellen zum Thema Aufwandsschätzung in Softwareprojekten werden nicht betrachtet, da in Kapitel 2 bereits stellvertretend die Methoden *Function Point* und *Use Case Point* vorgestellt worden sind.
- Literaturquellen zum Thema Testaufwandsschätzung über genetische oder evolutionäre Algorithmen werden nicht betrachtet. Die Lösung der Problemstellung 3 aus Abschnitt 1.1 soll über eine Formel in multiplikativer Form realisiert werden, da hiermit eine hohe Nachvollziehbarkeit der Testaufwandsschätzung gewährleistet ist (vgl. Anforderung 9 in Abschnitt 2.4.5).

Im nachfolgenden Abschnitt 3.1.5.2 werden die Ergebnisse der Literaturrecherche auf Basis der in diesem Abschnitt definierten Strategie zusammengefasst.

#### 3.1.5.2. Ergebnisse der Literaturrecherche

In diesem Abschnitt wird das Ergebnis der Literaturrecherche zusammengefasst. Die gefundenen Literaturquellen sind in Tabelle 3.7 sortiert nach dem Nachna-

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

men des ersten Autors aufgeführt. In Abschnitt 3.1.5.3 wird dieses Ergebnis aufgegriffen und die Literaturquellen evaluiert.

<b>Erster Autor</b>	<b>Titel der Literaturquelle</b>	<b>Verweis</b>
Almeida	An Alternative Approach to Test Effort Estimation Based on Use Cases	[AAM09]
Aranha	An Estimation Model for Test Execution Effort	[AB07a]
Aranha	Test Effort Estimation Models Based on Test Specifications	[AB07b]
Cangussu	A formal model of the software test process	[CDM02]
Collofello	Modeling software testing processes	[CYT <sup>+</sup> 96]
Dowie	Teststeuerung und -schätzung im Umfeld dynamischer Anforderungen	[DH06]
Dowie	Testaufwandsschätzung in der Softwareentwicklung – Modell der Einflussfaktoren und Methode zur organisationsspezifischen Aufwandsschätzung	[Dow09]
Hauptmann	An Expert-Based Cost Estimation Model for System Test Execution	[HJE <sup>+</sup> 14]
Hosseini	A Model for Determining the Cost of Test Case Execution (or Test Execution Effort Estimation Model)	[Hos14]
Kumari	Test effort estimation in regression testing	[KS13]
Kushwaha	Software Test Effort Estimation	[KM08]
Mizuno	On estimating testing effort needed to assure field quality in software development	[MSTK02]
Nageswaran	Test Effort Estimation Using Use Case Points	[Nag01]
Nguyen	qEstimation: a process for estimating size and effort of software testing	[NPL13]

### 3.1. Methoden zur Testaufwandsschätzung

Sharma	A metric suite for early estimation of software testing effort using requirement engineering document and its validation	[SK11]
Sharma	Applying requirement based complexity for the estimation of software development and testing effort	[SK12]
Silva	A Simple Approach for Estimation of Execution Effort of Functional Test Cases	[edJ09]
Sneed	Produkt- und Prozessmetriken für den Software-test	[SJ06]
Stikkel	Dynamic model for the system testing process	[Sti06]
Zhu	An Experience-Based Approach for Test Execution Effort Estimation	[ZZH <sup>+</sup> 08]
Zhu	Estimate Test Execution Effort at an Early Stage: An Empirical Study	[ZZW <sup>+</sup> 08]

Tabelle 3.7.: Übersicht über die gefundenen Literaturquellen

#### 3.1.5.3. Evaluation der Ergebnisse

In diesem Abschnitt werden die Ergebnisse der Literaturrecherche anhand der Forschungsfragen aus Abschnitt 3.1.5.1 evaluiert. Dabei werden die interessantesten Literaturquellen (vgl. Tabelle 3.7) vorgestellt.

Aranha et al. definieren ein Modell für die Schätzung des Testausführungsauwands auf Basis der Testfälle [AB07a, AB07b]. Mit dem Modell werden jedem auszuführenden Testfall Ausführungspunkte zugewiesen. Dabei ergibt sich die Höhe der Ausführungspunkte aus den durch den Testfall abgedeckten nicht-funktionalen und funktionalen Eigenschaften des Softwareproduktes. Damit dies durch eine Analyse der Testfälle automatisch möglich ist, müssen die Schritte der Testfälle in natürlicher Sprache mit bestimmten Schlüsselwörtern geschrieben sein. Auf Basis der Ausführungspunkte, die der Testkomplexität entspricht,

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

ist anschließend mit einem aus abgeschlossenen Projekten gewonnenen Umrechnungsfaktor des Testausführungsaufwands berechenbar. Für die Berechnung werden keine Testeinflussfaktoren berücksichtigt.

Cangussu et al. beschreiben ein formales Modell für den Softwaretest, wozu auch der Aspekt der Testaufwandsschätzung gehört [CDM02]. Mit dem Modell besteht die Möglichkeit zu jedem Zeitpunkt im Testprojekt den Testaufwand basierend auf der Komplexität des Softwareproduktes, der Größe des Testteams und der noch zu erwartenden Fehleranzahl zu schätzen. Dabei werden keine formalen Vorgaben gegeben, wie die (Test-)Komplexität erfasst werden soll. Genauso gibt es keine formalen Vorgaben zur Nutzung und Beschreibung von Testeinflussfaktoren.

Dowie definiert in ihrer Dissertation ein allgemeines Modell des Testaufwands und eine Methode wie dieses Modell auf eine bestimmte Organisation angepasst werden kann [Dow09]. Dazu führt sie eine Recherche nach Testeinflussfaktoren durch und untersucht diese im Anschluss in zwei Organisationen. Dabei werden keine formalen Vorgaben zur Beschreibung oder Nutzung der Testeinflussfaktoren beschrieben. Die daraus entstandene Menge an Testeinflussfaktoren bildet das allgemeine Modell des Testaufwands. Für die Anpassung an eine bestimmte Organisation wird diese Menge reduziert, so dass nur die für die Organisation relevanten Testeinflussfaktoren zur Berechnung des Testaufwands verwendet werden. Auf Basis der reduzierten Menge ist dann eine Formel zur Berechnung des Testaufwandes definierbar. Da in der Dissertation von Dowie die Testkomplexität nicht betrachtet wird, ist diese auch nicht in der Formel berücksichtigt.

Nageswaran beschreibt eine Idee für die Anpassung der Methode Use Case Point (vgl. Abschnitt 2.4.3.3), um Testaufwand schätzen zu können [Nag01]. Die Hauptanpassung ist die Einführung von Einflussfaktoren mit einem direkten Bezug zum Softwaretest, wie *Test Environment* oder *Test-ware reuse*. Die Testkomplexität wird ähnlich wie die Komplexität in der Methode Use Case Point erfasst. Dazu wird zunächst die Komplexität der Aktoren und Use Cases anhand einer groben Skala ermittelt und über direkte technische Einflussfaktoren skaliert. Anschließend wird der Kontext des Testprojektes anhand von Einflussfaktoren bewertet. Dabei sind die Testeinflussfaktoren und deren Nutzung nicht

### 3.1. Methoden zur Testaufwandsschätzung

formal beschrieben. Abschließend ist aus der skalierten Komplexität und den bewerteten Einflussfaktoren der Testaufwand über eine Formel berechenbar.

Almeida et al. erweitern die Methode von Nageswaran um die Unterscheidung von Erfolgs- und Fehlerszenarien in Use Cases, da Erfolgsszenarien aufwendiger zu testen seien [AAM09]. Dazu werden zunächst die Erfolgs- und Fehlerszenarien der zu testenden Use Cases getrennt erfasst. Anschließend werden sie mit unterschiedlichen Gewichten multipliziert und aufsummiert. Die Berechnung des Testaufwands wird wie bei der Methode von Nageswaran durchgeführt.

Hauptmann et al. beschreiben ein Modell mit dem die Entscheidung zwischen manueller und automatischer Testausführung unterstützt wird [HJE<sup>+</sup>14]. Dafür wird der Testaufwand einer manuellen Ausführung gegenüber einer automatisierten Ausführung betrachtet. Das Modell gibt dabei nur einen Rahmen vor, der zur Analyse von bestehenden Testfällen genutzt werden kann. Daher existieren keine formalen Vorgaben zur Erfassung von der Testkomplexität oder der Beschreibung sowie der Nutzung von Testeinflussfaktoren.

Kumari et al. erläutern eine Methode zur Testaufwandsschätzung für die Testausführung in einem Regressionstest [KS13]. Dabei ist ein Regressionstest ein Test nach einer Änderung an einem bereits produktiv eingesetzten Softwareprodukt. Mit der Methode wird zunächst die Ausführungskomplexität der zu testenden Testfälle bestimmt. Anschließend werden alle Änderungen des Softwareprodukt nach der Art der Änderung gewichtet: Codeänderung, Designänderung oder Anforderungsänderung. Danach ist über eine Formel der Testausführungs aufwand für den Regressionstest unter Berücksichtigung der Ausführungskomplexität, der Summe der Änderungen und der Erfahrung der Tester schätzbar. In dem Modell werden keine formalen Vorgaben für die Erfassung von Testkomplexität beschrieben, da nur der Testausführungs aufwand betrachtet wird. Neben der Erfahrung der Tester werden keine weiteren Testeinflussfaktoren genannt und keine formalen Vorgaben zur Beschreibung und Nutzung von Testeinfluss faktoren gegeben.

Nguyen et al. definieren eine Methode, um über die Komplexität von Testfällen eine Testaufwandsschätzung durchzuführen [NPL13]. Dabei ergibt sich die Komplexität eines Testfalls aus der Anzahl enthaltener Prüfpunkte für den Tester, aus der Bewertung der Komplexität der Vorbedingungen des Testfalls, der Komplexität der verwendeten Testdaten und der Testart (z.B. funktionaler oder

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

nicht-funktionaler Test). Hierfür werden detaillierte Vorgaben zur Bewertung gegeben. Anschließend ist mit der Summe der Komplexität über alle Testfälle und einen Produktivitätsindex der Testaufwand berechenbar. Da in der Methode keine Testeinflussfaktoren genutzt werden, sind keine formalen Vorgaben für die Beschreibung und Nutzung von Testeinflussfaktoren vorhanden.

Sharma et al. beschreiben ein Modell für die Bestimmung der Komplexität des Softwareproduktes, die an der Methode Function Point (vgl. Abschnitt 2.4.3.1) angelehnt ist [SK11, SK12]. Basierend auf der Komplexität definieren sie eine Reihe von Metriken für den Softwaretest, die auch zur Testaufwandsschätzung genutzt werden können. Dazu wird zunächst die Komplexität des Softwareproduktes mit einem *technical und environmental factor* zur Testkomplexität umgerechnet. Der *technical und environmental factor* ist dabei eine Summe aus 9 Testeinflussfaktoren, deren Nutzung und Beschreibung nicht formalisiert sind. Anschließend wird über eine Formel die Anzahl der Testfälle abgeleitet und die Produktivität der Tester über eine Matrix bestimmt. Dann ist im letzten Schritt der Testaufwand aus der Anzahl der Testfälle und der Erfahrung der Tester berechenbar.

Sneed et al. definieren eine Reihe allgemeiner Metriken, die im Bereich der analytischen Qualitätssicherung anwendbar sind [SJ06]. Dazu gehören u. a. Metriken für die Beurteilung der Testbarkeit, der Testeffektivität und der Kalkulation des Testaufwands. Für die Berechnung des Testaufwands wird eine angepasste Formel von COCOMO II (vgl. Abschnitt 2.4.3) genutzt. Dabei wird aus der Testkomplexität und den Testeinflussfaktoren der Testaufwand berechnet. Die Testkomplexität wird mit der Anzahl der Testfälle gleichgesetzt. Für die Testeinflussfaktoren gibt es u. a. den Verweis auf die Testwerkzeuge und die Testumgebung. Weitere formale Vorgaben für die Beschreibung und Nutzung von Testeinflussfaktoren fehlen.

Zhu et al. entwickeln eine Methode zur Testaufwandsschätzung, die den Testaufwand über die Anzahl der Testfälle und deren Ausführungskomplexität schätzt [ZZW<sup>+</sup>08, ZZH<sup>+</sup>08]. Dabei wird zunächst über die zu testenden Use Cases die Anzahl der Testfälle bestimmt. Anschließend werden die Use Cases hinsichtlich der zu erwartenden Testausführungskomplexität analysiert. Hierfür sind 15 Testeinflussfaktoren zu bewerten, deren Beschreibung und Nutzung nicht formalisiert sind. Basierend auf der Anzahl der Testfälle und deren Ausführungs-

komplexität ist dann der Testaufwand über einen Vergleich mit abgeschlossenen Projekten schätzbar.

#### 3.1.5.4. Fazit

In den in Abschnitt 3.1.5.2 evaluierten Literaturquellen wird in fast allen Fällen eine Form der Formalisierung für die Erfassung von Testkomplexität angegeben (vgl. Problemstellung 1). Dabei wird die Testkomplexität oft auf Basis der Testfälle ermittelt [AB07a, AB07b, HJE<sup>+</sup>14, KS13, NPL13, SK11, SK12]. Damit sind diese Methoden erst nach der Testfallerstellung anwendbar und nicht zu jedem Zeitpunkt in einem Softwareprojekt (vgl. Anforderung 1).

Die Methoden mit einer formalisierten Erfassung von Testkomplexität aus Anforderungsspezifikationen setzen Use Cases voraus [AAM09, Nag01, ZZH<sup>+</sup>08, ZZW<sup>+</sup>08]. Daher ist der Testaufwand für Teststufen, deren Testbasis nicht aus Use Cases besteht, nicht schätzbar. Dies betrifft den (System-)Integrationstest und den Abnahmetest (vgl. Abschnitt 2.3.2), die durch eine Lösung der Problemstellung 1 unterstützt werden müssen (vgl. Anforderung 7).

Die Formalisierung der Beschreibung und der Nutzung von Testeinflussfaktoren fehlt bei den evaluierten Literaturquellen (vgl. Problemstellung 2, Anforderung 11). Obwohl die meisten Methoden Testeinflussfaktoren bei der Berechnung des Testaufwands berücksichtigen, werden keine formalen Vorgaben zu deren Nutzung gegeben. Damit ist der Interpretationsspielraum für den Anwender sehr groß und die Schätzungen unterschiedlicher Personen sind nicht vergleichbar.

Die Formalisierung der Berechnung des Testaufwands aus der Testkomplexität und den Testeinflussfaktoren ist für alle in Abschnitt 3.1.5.2 vorgestellten Methoden gegeben, die Testeinflussfaktoren berücksichtigen (vgl. Problemstellung 3). Dazu wurde immer eine mathematische Formel gewählt. Dabei wurde in keinem Fall für die Formel eine multiplikative Form gewählt, die den Anwender beim Nachvollziehen der Berechnung unterstützt (vgl. Anforderung 9).

Die weiteren Anforderungen 2-6, 8, 12 und 13 werden von den in Abschnitt 3.1.5.2 evaluierten Literaturquellen nicht behandelt. Davon ist die fehlende Aufschlüsselung der Testaufwands nach Testaktivitäten (vgl. Anforderung 6)

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

am wichtigsten, da nur hierüber die rechtzeitige Verfügbarkeit von entsprechend qualifizierten Testpersonal geplant werden kann.

Zusammenfassend bietet keine bestehende Methode eine Lösung für die drei Problemstellungen aus Abschnitt 1.1. Daher wird in Abschnitt 3.2 ein Überblick über einen neuen Lösungsansatz gegeben, der in dieser Dissertation ausgearbeitet wird.

## 3.2. Überblick über den Lösungsansatz

Die Betrachtung der bekannten Methoden zur Testaufwandsschätzung in Abschnitt 3.1 hat gezeigt, dass keine der bestehenden Methoden die Problemstellungen aus Abschnitt 1.1 lösen kann. Daher wird in diesem Abschnitt ein Überblick über einen neuen Lösungsansatz für eine Methode zur Testaufwandsschätzung gegeben, der in den nachfolgenden Kapiteln 4 und 5 ausführlich beschrieben wird.

Die neue Methode zur Testaufwandsschätzung in der analytischen Qualitäts sicherung (TAQ) dient der Schätzung des Testaufwands für IT-Systeme, die für die Ausführung von Geschäftsprozessen genutzt werden. Die Methode ist in den drei Projektphasen Anforderungsphase, Design- und Entwicklungsphase für die Teststufen (System-)Integrationstest, Systemtest und Abnahmetest anwendbar. Das Ergebnis der Methodenanwendung ist der Testaufwand in Personentagen (PT), der acht Zeitstunden entspricht, für eine Teststufe und eine Testaktivität. Dabei werden die Teststufen (System-)Integrationstest, Systemtest und Abnahmetest sowie die Testaktivitäten Testmanagement, Testdesign und Testausführung unterstützt.

Die Methode TAQ basiert auf einem algorithmischen Modell, das dem *Top-Down* Vorgehen folgt. Im Kern steht die multiplikative Formel 3.8, die aus vier Faktoren zusammengesetzt ist. An erster Stelle steht der Faktor  $C_{S,A}$ , der eine Konstante darstellt, die dediziert für jede Teststufe ( $S$ ) und der Testaktivität ( $A$ ) zu bestimmen ist. Über diese Konstante sind später einfache Kalibrierungen an einen neuen Unternehmenskontext möglich.

### 3.2. Überblick über den Lösungsansatz

Der zweite Faktor  $M_{\text{Testkomplexität}}$  steht für die Testkomplexität und wird über das Modell für Testkomplexität berechnet. Der dritte Faktor  $\frac{1}{P_{S,A}}$  ist eine Konstante und steht für die Produktivität einer Organisation in Abhängigkeit von der Teststufe ( $S$ ) und Testaktivität ( $A$ ). Mit dem vierten Faktor  $M_{\text{Testeinflussfaktoren}}$ , der über das Modell für Testeinflussfaktoren ermittelt wird, werden projektspezifische Einflüsse auf die Produktivität berücksichtigt. Das Ergebnis der Formel 3.8 ist der Testaufwand in Personentagen ( $PT$ ) für eine Teststufe ( $S$ ) und eine bestimmte Testaktivität ( $A$ ).

Im Folgenden wird zunächst das Modell für Testkomplexität und die Produktivitätskonstante beschrieben. Anschließend erfolgt die Erläuterung des Modells für Testeinflussfaktoren.

$$PT_{S,A} = C_{S,A} \cdot M_{\text{Testkomplexität}} \cdot \frac{1}{P_{S,A}} \cdot M_{\text{Testeinflussfaktoren}} \quad (3.8)$$

$$M_{\text{Testkomplexität}} = K_S \quad (3.9)$$

$$M_{\text{Testeinflussfaktoren}} = \prod_{i=1}^{k_{A,V}} TF_{(S,A)_i} \quad (3.10)$$

$S \in \{\text{(System-)Integrationstest, Systemtest, Abnahmetest}\}$

$A \in \{\text{Testmanagement, Testdesign, Testausführung}\}$

$V \in \{\text{Anforderungsphase, Designphase, Entwicklungsphase}\}$

Mit dem Modell für Testkomplexität wird die Komplexität des Testobjektes anhand der Dokumente in der Testbasis gemessen. Dabei umfasst die Testbasis nach dem ISTQB Glossar „Alle Dokumente, aus denen die Anforderungen ersichtlich werden, die an eine Komponente oder ein System gestellt werden, bzw. die Dokumentation, auf der die Herleitung oder Auswahl der Testfälle beruht.[...]" [GTB10, S. 51]. Da das Testobjekt in den Teststufen unterschiedlich ist, muss jede Teststufe separat betrachtet werden. Dieser Zusammenhang wird in Formel 3.9 beschrieben. Die Variable  $K_S$  steht für die Testkomplexität der Teststufe  $S$  und entspricht damit dem Faktor  $M_{\text{Testkomplexität}}$  in Formel 3.8. Zusammenfassend wird Testkomplexität für die Methode TAQ wie folgt definiert:

### 3. Verwandte Arbeiten und Überblick über den Lösungsansatz

#### Definition 2: Testkomplexität

Die Testkomplexität steht für die Komplexität des Testobjektes. Die Bewertung der Komplexität des Testobjektes wird anhand der Dokumente in der Testbasis vorgenommen. Sie wird als natürliche Zahl in der Einheit Testkomplexitätspunkt (TKP) angegeben und steht in linearem Zusammenhang zu den Aufwänden der Testaktivitäten, die für den Test der bewerteten Testobjekte notwendig sind.

	Teilmodell Testkomplexität	Teilmodell Testeinflussfaktoren	Manueller Aufwand
Teststufe $S$	X	X	X
Testaktivität $A$		X	
Projektphase $V$		X	X

Tabelle 3.8.: Übersicht über die Parameter, deren Veränderung eine erneute Berechnung der Testkomplexität und lokalen Produktivität erfordert

Für die Berechnung des Testaufwandes aus Testkomplexitätspunkten ist ein Umrechnungsfaktor notwendig, der einen Bezug zwischen Testkomplexität in Testkomplexitätspunkten und Zeiteinheiten herstellt. Dieser Umrechnungsfaktor wird in der Methode TAQ als durchschnittliche Produktivität einer Organisation (globale Produktivität) bezeichnet und ist in Formel 3.8 der Faktor  $\frac{1}{P_{S,A}}$ . Für die quantitative Bestimmung des Faktors müssen Daten aus abgeschlossenen Testprojekten über die Testkomplexität und die zugehörigen Aufwände getrennt nach Teststufe und Testaktivität gesammelt werden. Das Verhältnis von Testkomplexität und Aufwand in einem Testprojekt ist die lokale Produktivität. Die globale Produktivität entspricht dann dem Durchschnitt über die lokale Produktivität aller erfassten Testprojekte. Zusammenfassend ist die Definition der Produktivität für die Methode TAQ:

#### Definition 3: Produktivität

Produktivität drückt aus, wie viele Arbeitseinheiten in PT pro Fertigstellung einer Produktivitätseinheit in TKP notwendig sind. Damit ist die Produktivität der Quotient von TKP und Arbeitseinheiten in PT. Daraus folgt, dass die Einheit der Produktivität  $\frac{TKP}{PT}$  ist.

Abweichungen von der globalen Produktivität durch besondere Einflussfaktoren des zu entwickelnden Produktes, des Testprojektes, den Projektbeteiligten und der Testorganisation werden mit dem Modell für Testeinflussfaktoren ermittelt. Die Höhe der Abweichung wird mit Formel 3.10 berechnet. Die Variable  $TF_{(S,A)_i}$  steht für den prozentualen Einfluss des Testeinflussfaktors  $i$  aus  $k_{A,V}$  Einflussfaktoren in Bezug auf die Teststufe  $S$ , die Testaktivität  $A$  und die Projektphase  $V$ . Multipliziert ergeben die  $TF_{(S,A)_i}$  den Faktor  $M_{\text{Testeinflussfaktoren}}$ . Die Multiplikation des Faktors  $M_{\text{Testeinflussfaktoren}}$  mit der globalen Produktivität ergibt die lokale Produktivität.

#### **Definition 4: Testeinflussfaktor**

Ein Testeinflussfaktor (TF) ermöglicht projektspezifische Abweichungen von der globalen Produktivität. Der Wert des Testeinflussfaktors gibt die prozentuale Veränderung der globalen Produktivität getrennt für jede Teststufe und Testaktivität an. Das Ergebnis des Produktes aus globaler Produktivität und der Testeinflussfaktoren ist in der Einheit  $\frac{PT}{TKP}$  anzugeben. Dieses Ergebnis wird auch als lokale Produktivität für ein Testprojekt bezeichnet.

Die Methode TAQ muss für jede Änderung der Parameter Teststufe, Testaktivität und Projektphase erneut ausgeführt werden. Dies kann teilweise automatisiert erfolgen, so dass kein erneuter Aufwand für die Durchführung der Methode TAQ notwendig ist. In Tabelle 3.8 sind die Abhängigkeiten dargestellt.

## **3.3. Zusammenfassung**

In diesem Kapitel wurden zunächst in Abschnitt 3.1 existierende Methoden zur Testaufwandsschätzung betrachtet. Dazu wurden in Abschnitt 3.1.1 zunächst Methoden zur Testaufwandsschätzung über die bekannten Standards im Bereich der Qualitätssicherung identifiziert. Im Anschluss wurden die Methoden näher vorgestellt (vgl. Abschnitte 3.1.2 bis 3.1.4). Für die Methoden wurde jeweils festgestellt, dass sie die Problemstellungen aus Abschnitt 1.1 nicht lösen.

### *3. Verwandte Arbeiten und Überblick über den Lösungsansatz*

Da in den gängigen Standards das Thema Testaufwandsschätzung nur rudimentär betrachtet wird, wurde in einem zweiten Schritt eine Literaturrecherche nach weiteren Methoden zur Testaufwandsschätzung durchgeführt (vgl. Abschnitt 3.1.5). Dabei wurden zunächst Literaturquellen in elektronischen Bibliotheken und dem Internet identifiziert. Diese wurden anschließend evaluiert und in einem Fazit beurteilt, ob sie die Problemstellungen aus Abschnitt 1.1 lösen können. Auch dabei wurde erkannt, dass keine der Methoden hierfür geeignet ist.

Da keine der untersuchten Methoden eine Lösung der Problemstellungen aus Abschnitt 1.1 und den in Kapitel 2 definierten Anforderungen geboten hat, wurde in Abschnitt 3.2 ein Überblick über einen neuen Lösungsansatz gegeben. Dieser wird in den nachfolgenden Kapiteln ausgearbeitet. Dazu wird in Kapitel 4 das Modell für Testkomplexität beschrieben. Darauf aufbauend wird in Kapitel 5 das Modell für Testeinflussfaktoren vorgestellt.

## 4. Entwicklung des Modells für Testkomplexität

Software cost models, like other computer models, are “garbage in-garbage out” devices: put a too-small sizing estimate in, and you will get a too-small cost estimate out.

---

(Barry W. Boehm [Boe81])

In diesem Kapitel wird das Modell für Testkomplexität beschrieben, das zur Schätzung der Komplexität von Testobjekten genutzt wird (vgl. Definition 2 zu Testkomplexität). Das Ergebnis der Modellanwendung ist eine natürliche Zahl, die in linearem Zusammenhang zu den Aufwänden der Testaktivitäten steht, die für den Test der bewerteten Testobjekte notwendig sind.

Die Entwicklung des Modells für Testkomplexität ist nach dem Prozess in Abbildung 4.1 erfolgt. Dabei wird zunächst die Grundlage für die Definition des Modells über eine Untersuchung der Teststufen (System-)Integrationstest, Systemtest und Abnahmetest geschaffen (vgl. Abschnitt 4.1). Hierfür ist die Testbasis jeweils der Ausgangspunkt, da diese alle Dokumente zur Herleitung und Auswahl der Testfälle für die Testobjekte enthält [GTB10, S. 51].

Die Dokumente der Testbasis können mit unterschiedlichen Spezifikationstechniken erstellt worden sein. Damit existiert kein gemeinsames Verständnis über die Bezeichnungen, die für die Beschreibung des Modells für Testkomplexität notwendig sind. Daher werden für jede Teststufe die Testziele und Testobjekte erläutert und gezeigt, wie die Spezifikation der Testobjekte in der Testbasis mit den Standardmodellierungssprachen der UML aussehen kann.

#### 4. Entwicklung des Modells für Testkomplexität

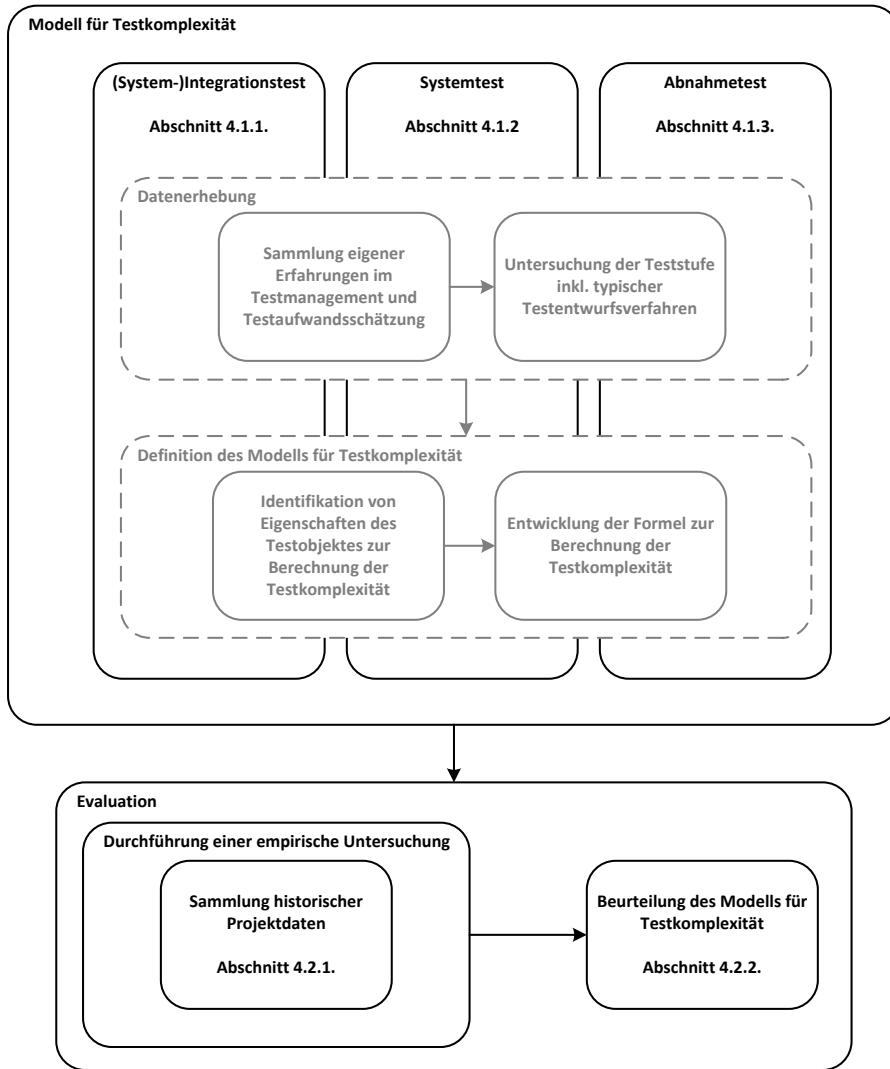


Abbildung 4.1.: Übersicht über den verfeinerten Prozess zur Entwicklung des Modells für Testkomplexität

Basierend auf den Begriffen der UML und dem Wissen über die Testziele bzw. die in der Teststufe zu betrachtenden Testobjekte werden die zu messenden Eigenschaften der Testbasis definiert. Um daraus eine Berechnungsvorschrift für Testkomplexität zu ermitteln, werden zunächst ergänzend je Teststufe typische Testentwurfsverfahren betrachtet. Aus der Kombination aus den zu messenden Eigenschaften und den Testentwurfsverfahren wird anschließend eine Formel zur Berechnung der Testkomplexität abgeleitet.

Über die Testkomplexität ist eine Bestimmung der (lokalen) Produktivität für abgeschlossene Testprojekte möglich. Dazu muss je Testprojekt der geleiste-

te Aufwand aufgeschlüsselt nach Testaktivitäten in Personentagen (PT) vorliegen und die Testkomplexität in Testkomplexitätspunkten (TKP) geschätzt worden sein. Dann ist die Produktivität aus der Division der Testkomplexität und den Personentagen in der Einheit  $\frac{TKP}{PT}$  berechenbar (vgl. Definition 3). Mit der Kenntnis über die lokale Produktivität in mehreren abgeschlossenen Projekten, kann eine globale Produktivität für eine Organisation (vgl. Konstante für die globale Produktivität in Formel 3.8) berechnet werden.

Abschließend wird in der Evaluation des Modells für Testkomplexität der lineare Zusammenhang zu den Aufwänden der Testaktivitäten gezeigt (vgl. Abschnitt 4.2). Ergänzend wird die globale Produktivität für die Organisation CRM IT berechnet und die Abweichungen von der globalen Produktivität stichprobenartig in einigen Testprojekten diskutiert. Eine weitergehende Betrachtung der Abweichungen erfolgt in Kapitel 5 mit der Entwicklung des Modells für Testeinflussfaktoren.

## 4.1. Modell für Testkomplexität

In Kapitel 2 wurden in Abschnitt 2.3.2 die Teststufen (System-)Integrationstest, Systemtest und Abnahmetest vorgestellt. Dabei wurde deutlich, dass die Teststufen unterschiedliche Testziele aufweisen. Daher ist auch die Testbasis als Ausgangspunkt für die drei Teststufen verschieden und die Testkomplexität ist nicht gleichartig berechenbar. Entsprechend wird die Entwicklung des Modells für Testkomplexität in drei Teilen, die den Teststufen entsprechen, beschrieben.

Die nachfolgenden drei Abschnitte entsprechen jeweils einer der Teststufen (System-)Integrationstest (vgl. Abschnitt 4.1.1), Systemtest (vgl. Abschnitt 4.1.2) und Abnahmetest (vgl. Abschnitt 4.1.3). Der Aufbau der Abschnitte ist gleich. Zunächst werden die Testobjekte und das Testziel für die Teststufe beschrieben. Anschließend wird an einem durchgängigen Beispiel gezeigt, wie die Dokumente in der Testbasis mit UML spezifiziert werden können. Dabei werden Eigenschaften der Testbasis identifiziert, die für die Berechnung der Testkomplexität gemessen werden können. Ergänzend wird auf zugehörige Testverfahren eingegangen, die zum Herleiten und Auswählen von Testfällen eingesetzt werden. Abschließend wird eine Formel entwickelt, deren Ergebnis die

## 4. Entwicklung des Modells für Testkomplexität

Testkomplexität für eine Teststufe darstellt (vgl. Formel 3.9). Zusammen stellen die drei entwickelten Formeln das Modell für Testkomplexität dar.

Geschlossen wird die Definition des Modells für Testkomplexität in Abschnitt 4.1.4 mit einem Fazit. Daran anschließend wird in Abschnitt 4.2 die Evaluation des Modells beschrieben.

### 4.1.1. (System-)Integrationstest

Die Testobjekte für einen Integrationstest sind Schnittstellen zwischen Komponenten, (Sub-)Systemen oder die Interaktion mit anderen Teilen innerhalb eines Systems (Hardware und Software) [Int11, S. 25]. Das Testziel ist das Aufdecken von Fehler in der Implementierung und Nutzung der Schnittstellen. Eine systematische Integration von Softwarekomponenten bis zu einem Systemverbund durchläuft mehrere Stufen. Eine mögliche erste Stufe ist die Integration von Komponenten. Dabei können Komponenten im Fall der objektorientierten Programmierung Klassen im Programmcode sein [SL10, S. 44]. Die zugehörige Teststufe ist der Komponentenintegrationstest.

Nach dem erfolgreichen Komponentenintegrationstest werden die integrierten Komponenten über verschiedene Zwischenstufen weiter bis zu (Sub-)Systemen zusammengesetzt und getestet. Die abschließende Teststufe ist der Systemintegrationstest, bei dem die Schnittstellen zwischen Systemen getestet werden. Die Stufe Systemintegrationstest wird im Folgenden genauer betrachtet, da die TAQ Methode den Integrationstest ab dieser Stufe unterstützen soll. Da die Prinzipien auch für Integrationstests in tieferen Stufen angewendet werden können, wird der Begriff System in der Bezeichnung (System-)Integrationstest eingeklammert.

#### Testbasis für die Teststufe (System-)Integrationstest

Für einen (System-)Integrationstest besteht die Testbasis u. a. aus Dokumenten, die die Architektur des Systemverbundes und die technischen Spezifikationen der Schnittstellen zwischen den beteiligten Systeme beschreiben [Int11, S. 25]. Diese Dokumente können durchgängig mit Hilfe der UML erstellt werden, die dafür verschiedene Diagrammarten anbietet. In diesem Abschnitt werden exem-

platisch drei verschiedene Diagrammarten vorgestellt, die entsprechend in drei Ebenen aufeinander aufbauen. In den Ebenen wird die Beschreibung der Architektur schrittweise zur Spezifikation einzelner Parameter in den Schnittstellen detailliert.

Auf oberster Ebene kann die Architektur eines Systemverbundes in Form eines UML Komponentendiagramms abgebildet werden [Obj11a, S. 146ff][RQ07, S. 211ff]. Ein Beispiel ist in Abbildung 4.2 zu sehen. Dort sind vier Systeme als Komponenten (Online Shop 1, Online Shop 2, Back-End System und Logistik) dargestellt. Die Komponente **Back-End** bietet die Schnittstellen **I-Back-End** und die Komponente **Logistik** die Schnittstelle **I-Logistik** nach außen an. Die Schnittstelle **I-Back-End** wird von der Komponente **Online Shop 1** und **Online Shop 2** genutzt während die Schnittstelle **I-Logistik** von der Komponente **Back-End** verwendet wird. Diese Verwendungsbeziehungen zwischen der Komponente werden in dem Komponentendiagramm über Kompositionskonnektoren (*assembly connector*) gezeigt. Da in einem (System-)Integrationstest sichergestellt werden muss, dass die Systeme über die Schnittstellen korrekt zusammenarbeiten, sind für die Bewertung der Komplexität der Testobjekte auf dieser Ebene die Kompositionskonnektoren zu zählen.

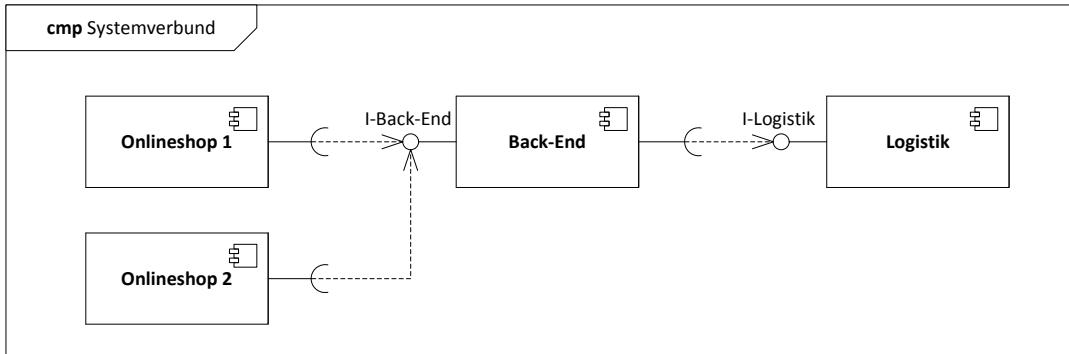


Abbildung 4.2.: Darstellung der Architektur des Systemverbundes mit einem UML Komponentendiagramm

Die zweite Ebene zur Beschreibung besteht aus der technischen Spezifikation der Kommunikation zwischen zwei Systemen über eine Schnittstelle. Mit der UML kann der Ablauf der Kommunikation u. a. in Sequenzdiagrammen spezifiziert werden [Obj11a, S. 473ff][RQ07, S. 397ff]. In Abbildung 4.3 sind zwei UML Sequenzdiagramme für den Ablauf der Kommunikation zur Anlage eines neuen Kunden über die Schnittstelle **I-Back-End** dargestellt. Die

#### 4. Entwicklung des Modells für Testkomplexität

Sequenzdiagramme sind als Detaillierung der Kompositionskonnektoren zwischen den Online Shops und dem Back-End System aus dem Komponentendiagramm zu betrachten. Das linke Sequenzdiagramm enthält die Komponenten **Online Shop 1** und **Back-End**. Die Pfeile in Richtung des **Back-End** Systems sind mit den Namen der Operationen (**einloggen**, **gibFreieKundennummer**, **legeNeuenKundenMitNummerAn**) beschriftet, die von der Komponente **Online Shop 1** nacheinander über die Schnittstelle **I-Back-End** aufgerufen werden. Entsprechend stehen die Pfeile in Richtung der Komponente **Online Shops 1** für die Antworten der Komponente **Back-End**. Da in einem (System-)Integrationstest sichergestellt sein muss, dass die Kommunikation zwischen den Systemen über die genutzten Operationen funktionieren, müssen je Kompositionskonnektor die Operationen getestet werden. Daher sind für die Bewertung der Komplexität der Testobjekte die genutzten Operationen je Kompositionskonnektor zu zählen.

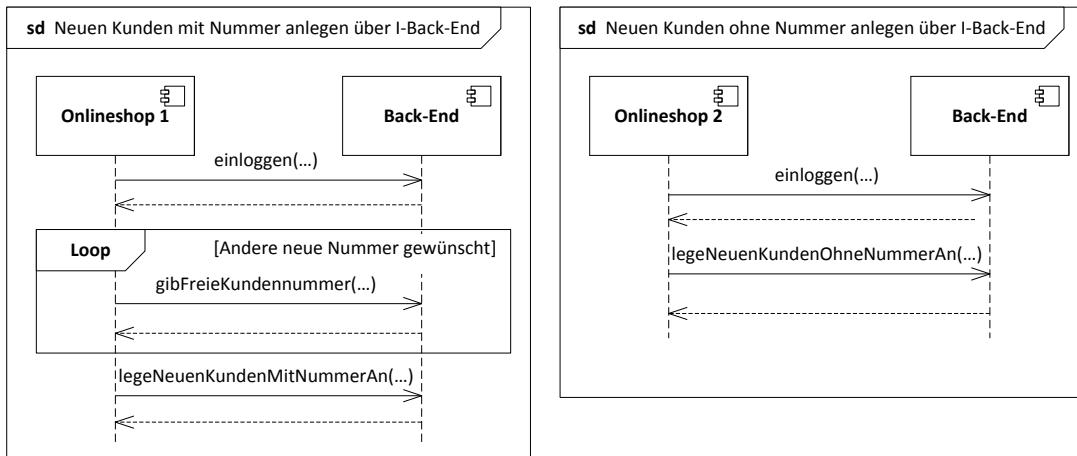


Abbildung 4.3.: Darstellung der Kommunikation der Online Shops mit dem Back-End System in UML Sequenzdiagrammen

Auf der dritten Ebene werden die technischen Spezifikation der Operationen detailliert. Dabei werden die zugehörigen Signaturen<sup>1</sup> vervollständigt, von denen in den vorgestellten Sequenzdiagrammen nur die Operationsnamen erkennbar gewesen sind. Diese Detaillierung ist mit einem UML Klassendiagramm möglich [Obj11a, S. 86][RQ07, S. 122ff]. In Abbildung 4.4 ist ein Klassendiagramm für die Schnittstelle **I-Back-End** zu sehen. Für die Operation **einloggen** sind die zwei Übergabeparameter **Benutzer** und **Passwort** mit dem Typ **String** defi-

<sup>1</sup>Eine Signatur besteht aus dem Operationsnamen und den Über- und Rückgabeparametern nach [RQ07, S. 117].

niert. Die Rückgabe ist vom Typ `Login`, der als separater Datentyp beschrieben ist. Bei der Operation `gibFreieKundennummer` ist kein separater Datentyp notwendig, da die Rückgabe vom Typ `Integer` ist.

Da in einem (System-)Integrationstest die Operationen aufgerufen und ihre Rückgaben ausgewertet werden müssen, sind die Über- und Rückgabeparameter zu berücksichtigen. Für die Bewertung der Komplexität der Testobjekte auf dieser Ebene sind daher die Über- und Rückgabeparameter für jede Operation zu zählen. Wenn die Über- und Rückgabeparameter auf komplexen Datentypen beruhen, sind die Attribute des Datentyps als Parameter zu zählen.

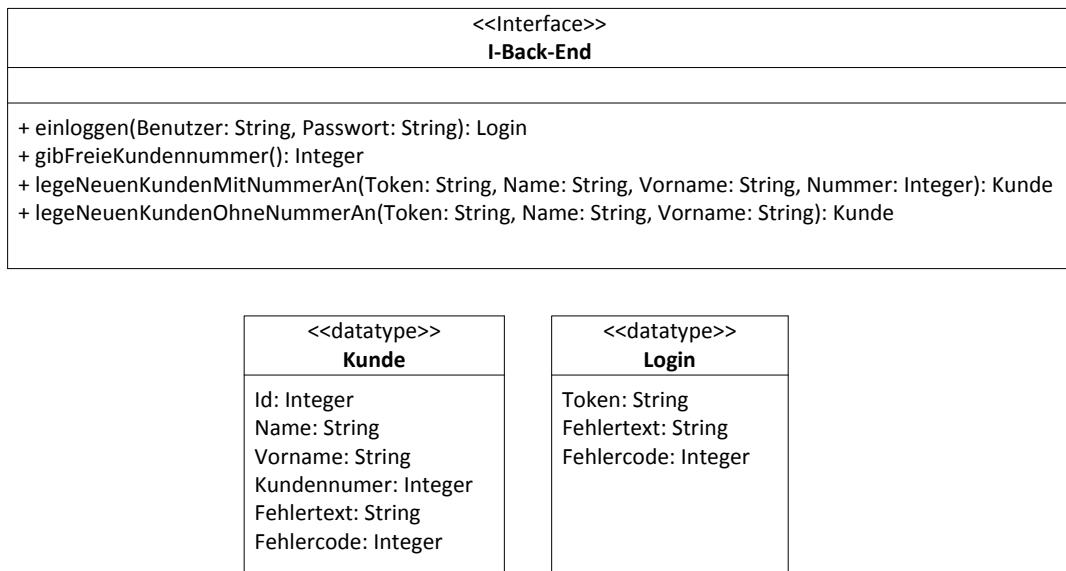


Abbildung 4.4.: Signaturen der Operationen der Schnittstelle I-Back-End in einem UML Klassendiagramm

In Abbildung 4.5 sind die in den drei Ebenen identifizierten Eigenschaften der Testobjekte für den (System-)Integrationstest in einem Metamodell zusammengefasst. Die Testkomplexität für den (System-)Integrationstest ist auf Basis jeder Spezifikation bestimmbar, aus der die im Metamodell enthaltenen Elemente ableitbar sind. Damit die Ableitung nicht von Anwender zu Anwender verschieden sind, müssen Vorgaben bereitgestellt werden. Diese sind in Form einer Abbildung der genutzten Modellierungssprache zur Erstellung der Spezifikation auf das in Abbildung 4.5 gezeigte Metamodell zu definieren.

#### 4. Entwicklung des Modells für Testkomplexität

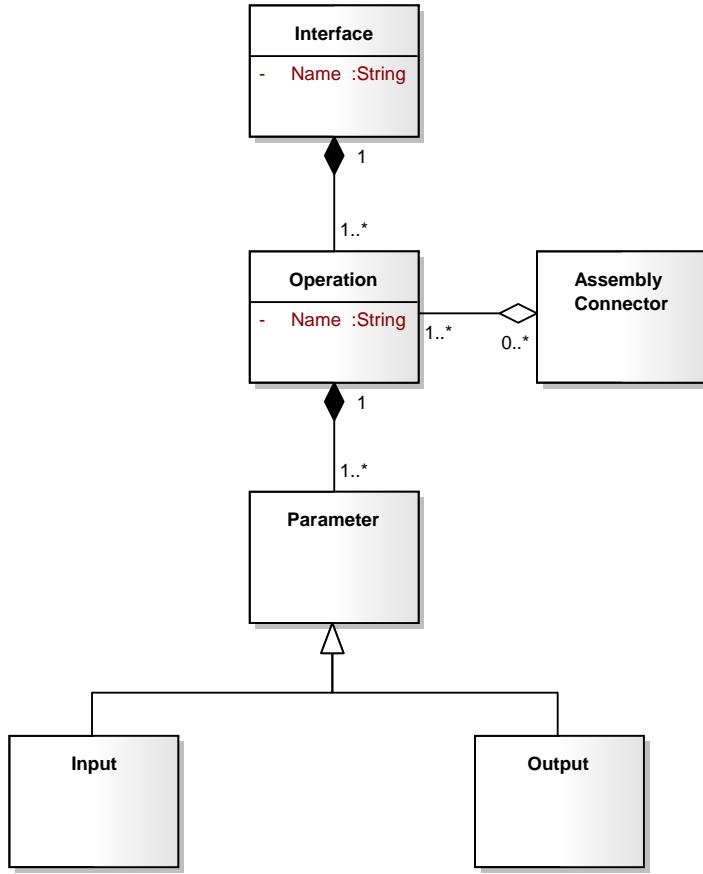


Abbildung 4.5.: Die zu berücksichtigenden Eigenschaften der Testobjekte für den (System-)Integrationstest

#### Standardtestverfahren für die Teststufe (System-)Integrationstest

Für den (System-)Integrationstest eignet sich das *Blackbox* Testverfahren Äquivalenzklassenbildung [SL10, S. 114ff][Int11, S. 40]. Dabei wird die Menge der möglichen konkreten Ein- und Ausgaben in Äquivalenzklassen unterteilt. Zu einer Äquivalenzklasse gehören dabei alle Ein- und Ausgaben bei denen sich das Testobjekt gleich verhält. Für ein systematisches Vorgehen zur Testfallerstellung für einen (System-)Integrationstest müssen für jeden Übergabeparameter einer Operation Äquivalenzklassen mit zulässigen und unzulässigen Werten aus der Testbasis abgeleitet werden. Ein Beispiel ist der Übergabeparameter **Nummer** der Operation `legeNeuenKundenMitNummerAn` in Abbildung 4.4. Die Äquivalenzklasse mit zulässigen Werten für diesen Übergabeparameter umfasst alle ganzen Zahlen. Eine mögliche Äquivalenzklasse für unzulässige Werte sind alle Werte, die nicht eine ganze Zahl darstellen. Wenn das System auf übergebene unzulässige Werte reagiert, kann dies zu einem Fehler führen.

sige Werte (z. B. werden Buchstaben anders als Fließkommazahlen behandelt) mit unterschiedlichen Fehlertexten reagiert, ist die Äquivalenzklasse für unzulässige Werte in zwei Äquivalenzklassen zu verfeinern. Im einfachsten Fall sind für jeden Übergabeparameter zwei Äquivalenzklassen (zulässige/unzulässige Werte) zu definieren. Ähnlich kann für die Ausgabeparameter vorgegangen werden. Bei der Testfallerstellung, die auf Äquivalenzklassen aufbaut, muss berücksichtigt werden, dass eine unzulässige Äquivalenzklasse ausschließlich in Kombination mit zulässigen Äquivalenzklassen getestet werden darf [SL10, S. 120].

### Schätzung der Testkomplexität für den (System-)Integrationstest

Die gezählten Elemente aus den drei Ebenen der Testbasis werden als Messergebnisse in der Formel 4.1 zur Berechnung der Testkomplexität für einen (System-)Integrationstest verwendet. Der Aufbau der Formel folgt der Idee, dass die Mindestzahl der Testfälle, die auf Basis der Äquivalenzklassenanalyse zu erstellen sind, der Anzahl aller Parameter plus Zwei entsprechen. Dabei wird davon ausgegangen, dass alle zulässigen Äquivalenzklassen zusammen in einem Testfall und alle unzulässigen getrennt in jeweils einem Testfall getestet werden müssen. Dies ist für Eingabe- und Ausgabeparameter zu erfüllen. Die Summe über jede Operation und Kompositionskonnektor ergibt die Testkomplexität  $K_{(\text{System-})\text{Integrationstest}}$  für einen (System-)Integrationstest.

$$K_{(\text{System-})\text{Integrationstest}} = \sum_{n=1}^{AKK} \sum_{m=1}^{AO_n} (AEP_m^n + AAP_m^n + 2) \quad (4.1)$$

$AKK$  = Anzahl Kompositionskonnektoren

$AO_n$  = Anzahl Operationen für Kompositionskonnektor  $n$

$AEP_m^n$  = Anzahl Übergabeparameter für Operation  $m$   
von Kompositionskonnektor  $n$

$AAP_m^n$  = Anzahl Rückgabeparameter für Operation  $m$   
von Kompositionskonnektor  $n$

Abhängig vom Zeitpunkt der Schätzung oder der Qualität der Testbasis sind evtl. keine Informationen über die zweite und dritte Ebene vorhanden, die in diesem Abschnitt beschrieben worden sind. Um dennoch eine Bestimmung der Testkomplexität zu ermöglichen, werden die Annahmen in Tabelle 4.1 für die

#### 4. Entwicklung des Modells für Testkomplexität

Beschreibung	Annahme
Operationen je Kompositionskonnektor	3
Anzahl Über- und Rückgabeparameter	10

Tabelle 4.1.: Annahmen für die Berechnung der Testkomplexität für einen (System-)Integrationstest in frühen Projektphasen oder bei nicht vollständiger Dokumentation

Berechnung vorgeschlagen. Diese Annahmen beruhen auf der Auswertung von Testprojekten, die auch später für die Evaluation des Modells für Testkomplexität genutzt worden sind (vgl. Abschnitt 4.2). Dazu wurde der Durchschnitt über die Anzahl der Operationen je Kompositionskonnektor bzw. der Über- und Rückgabeparameter berechnet.

### 4.1.2. Systemtest

Testobjekte für den Systemtest sind u. a. das entwickelte System, die Konfiguration des Systems und die zugehörigen Benutzer- und Betriebshandbücher. Das Testziel ist die Überprüfung, ob das System die mit dem Kunden vereinbarten funktionalen und nicht-funktionalen Anforderungen erfüllt [Int11, S. 26][SL10, S. 60ff]. Die Testbasis besteht aus Dokumenten, die u. a. die Anforderungsspezifikation und die funktionale Spezifikation beinhalten. Die Dokumente können hinsichtlich funktionaler Anforderungen mit Hilfe der UML beschrieben werden. Dazu werden in diesem Abschnitt zwei Diagrammarten vorgestellt, die aufeinander aufbauen. Die nicht-funktionalen Anforderungen sind nach [Rup13, S. 84] am Besten in natürlicher Sprache zu spezifizieren. Da nicht-funktionale Anforderungen normalerweise mit funktionalen Anforderungen verknüpft sind [Rup13, S. 20], wird bei der Vorstellung der UML Diagrammarten darauf eingegangen, wie die nicht-funktionalen Anforderungen bei der Messung der Komplexität eines Testobjektes berücksichtigt werden können.

#### Testbasis für die Teststufe Systemtest

Die funktionalen Anforderungen an ein System können auf oberster Ebene in einem UML Use Case Diagramm spezifiziert werden [Obj11a, S. 597ff][RQ07, S.237ff][Rup13, S. 66ff]. Diese Diagrammart bietet einen schnellen Überblick

#### 4.1. Modell für Testkomplexität

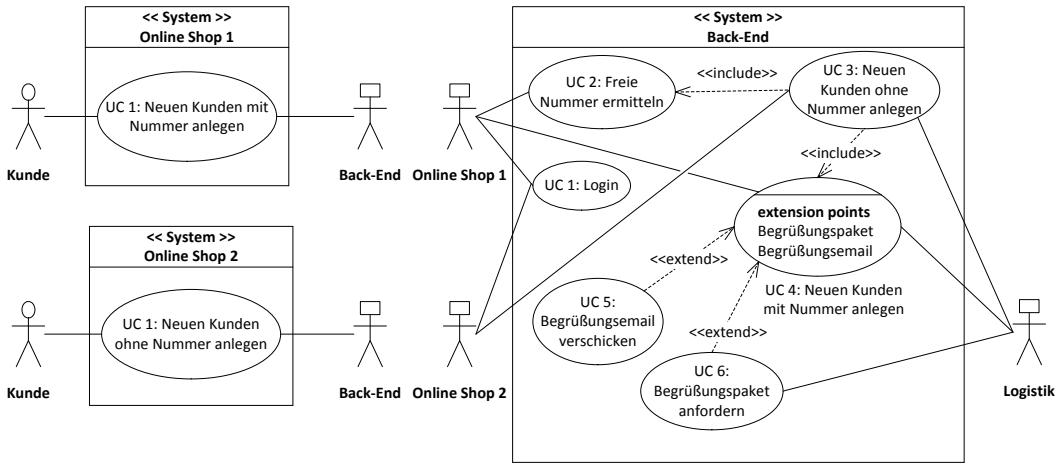


Abbildung 4.6.: Übersicht über die Use-Cases der Online Shops und des Back-End Systems als UML Use Case Diagramm

über die Grenzen, die Funktionalität und die beteiligten Kommunikationspartner. Das zentrale Element in dieser Diagrammart sind Use Cases zur Beschreibung des Systemverhaltens. Dabei stellt ein Use Case eine Kapselung von Aktivitäten dar, die in einer bestimmten Reihenfolge ablaufen.

In Abbildung 4.6 sind für drei von vier Komponenten aus Abbildung 4.2 Use Case Diagramme dargestellt. Die Use Case Diagramme für die Online Shops beinhalten jeweils einen Use Case, der den Ablauf für die Anlage eines neuen Kunden beschreibt. Für die Use Cases sind die beteiligten Kommunikationspartner der Akteur **Kunde** und der Akteur **Back-End**, der für die gleichnamige Komponente steht. Die Beteiligung wird in dem Diagramm durch eine Assoziation zwischen dem Akteur und dem Use Case ausgedrückt.

Das Use Case Diagramm in Abbildung 4.6 für die Komponente **Back-End** enthält sechs Use Cases und die drei Akteure **Online Shop 1**, **Online Shop 2** und **Logistik**. In dem Diagramm sind Abhängigkeiten der Use Cases in Form von Assoziationen zu sehen, die mit **«include»** oder **«extend»** typisiert sind. Mit Assoziationen vom Typ **«include»** wird der Ablauf eines Use Cases in einen anderen integriert. In dem Use Case Diagramm integriert der Use Case **UC 3** die Abläufe der Use Cases **UC 2** und **UC 4**. Ohne die integrierten Use Cases ist der Ablauf des Use Case **UC 3** nicht ausführbar. Assoziation vom Typ **«extend»** zeigen, dass ein Use Case durch einen anderen Use Case in einem bestimmten Punkt erweitert wird. Ein Beispiel ist in dem Use Case Diagramm der **UC 4**, der die zwei Erweiterungspunkte **Begrüßungspaket** und **Begrüßungsemail** beinhaltet.

#### *4. Entwicklung des Modells für Testkomplexität*

tet. An diesen Punkten wird in dem UC 4 zur Ausführungszeit entschieden, ob die Abläufe der Use Cases UC 5 und UC 6 in den Ablauf integriert werden.

Eine weitere Möglichkeit für Beziehungen in UML Use Case Diagrammen ist die Generalisierung [RQ07, S. 243f und S. 251]. Die Generalisierung ist eine gerichtete Beziehung und steht für die Spezialisierung eines Use Cases oder Akteurs. In dem Use Case Diagramm aus Abbildung 4.6 könnten z. B. die Akteure **Student** und **Rentner** eingeführt werden, die den Akteur **Kunde** spezialisieren. Entsprechend könnten Use Cases zur Neukundenanlage erstellt werden, die den Use Case UC 1 spezialisieren. Im Folgenden wird nur die Generalisierung zwischen Akteuren beachtet. Es werden alle Akteure berücksichtigt, die direkt oder indirekt durch eine Generalisierung mit einem Use Case verbunden sind. Die Generalisierung zwischen Use Cases wird nicht beachtet, sondern der spezialisierte Use Case wird als vollständig autarker Use Case betrachtet.

Für einen Systemtest muss überprüft werden, dass die Funktionalität entsprechend der Use Cases umgesetzt ist. Daher wird für die Bewertung der Komplexität der Testobjekte die Anzahl der Use Cases gezählt. Weiterhin wird die Anzahl der Kommunikationspartner gezählt, da die Akteure direkt oder durch Stellvertreter im Systemtest beteiligt sind. Abhängig von der Art der Kommunikation mit den Akteuren können unterschiedlich hohe Aufwände im Testprojekt entstehen. Um dies zu berücksichtigen, werden die Akteure analog zu [Fro09, S. 58] nach der Art der Kommunikation gewichtet. Die Wahl der Gewichtung entspricht der Anzahl Schritte, die in den Testfällen zur Abbildung der Kommunikation notwendig sind. Für die Testausführung notwendige menschliche Akteure werden dreifach gezählt, da ein Schritt zur Kommunikationsaufnahme, einer zu Beschreibung der Tätigkeit des Akteurs und einer zur Verarbeitung/Prüfung der Rückantwort durch den Tester notwendig ist. Die Kommunikation mit einem zweiten System wird entsprechend der Tabelle 4.2 gewichtet. Hier ist für die einfachste Form der Kommunikation nur ein Schritt notwendig, während bei komplexer Kommunikation von drei Schritten ausgegangen wird. Speziell die Kommunikation über zustandsbehaftete Schnittstellen erfordert in einem Testfall drei oder mehr Schritte zur Herstellung des Zustandes, der eigentliche Kommunikation und der Prüfung des Ergebnisses in Form der Antwort und dem Zustand der Schnittstelle.

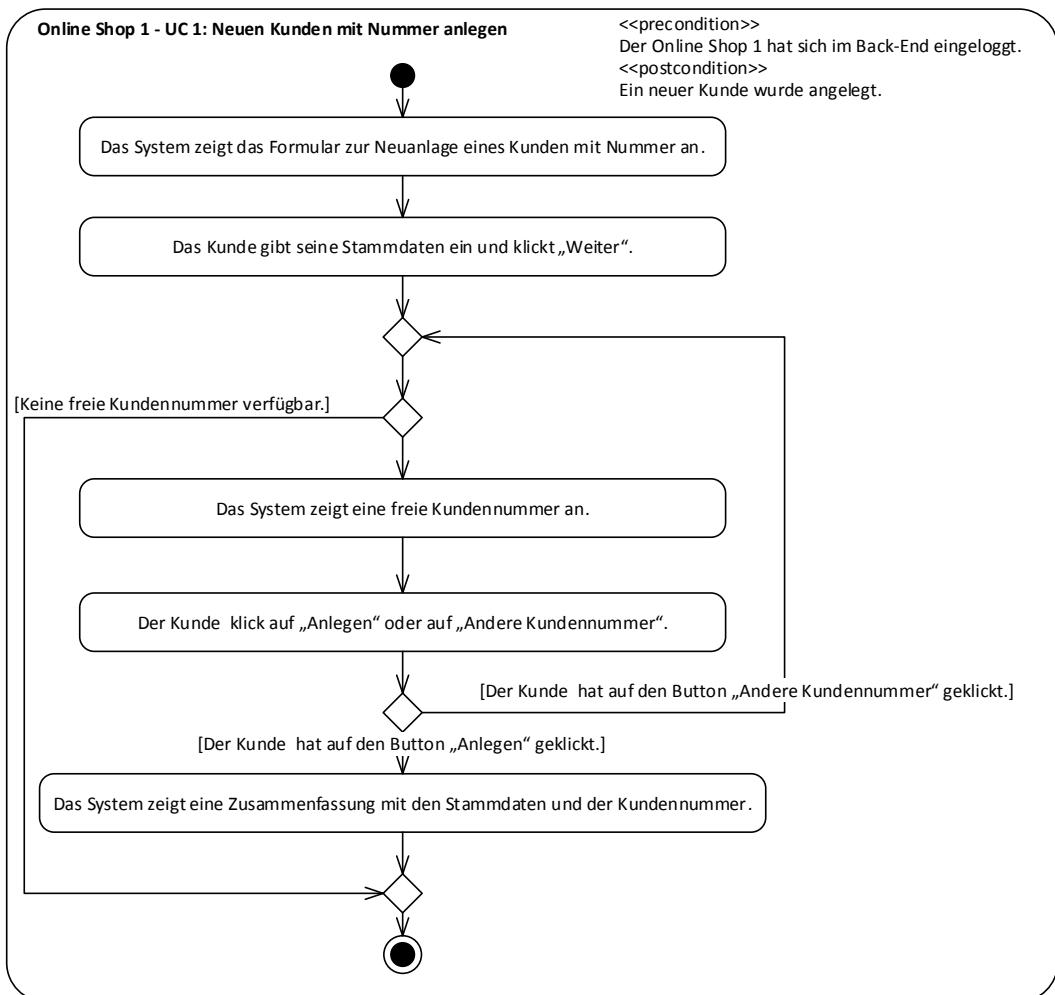


Abbildung 4.7.: Schritte des Use Cases *Online Shop 1 - UC 1 Neuen Kunden mit Nummer anlegen* als UML Aktivitätendiagramm

#### 4. Entwicklung des Modells für Testkomplexität

Klasse	Definition	Gewicht
Einfach	Kommunikation über zustandslose Schnittstellen keine Versionierung der Schnittstellen notwendig es muss kein fachliches Transaktionskonzept beachtet werden	1
Mittel	Mindestens eines der drei Merkmale der Klasse <i>komplex</i>	2
Komplex	Kommunikation über zustandsbehaftete Schnittstellen mehrere Versionen von Schnittstellen müssen angeboten werden Beachtung von langen fachlichen Transaktionen (über mehrere Aufrufe hinweg) erforderlich	3

Tabelle 4.2.: Gewichtung der Akteure in der Methode TAQ

Die zweite Ebene der Beschreibung der funktionalen Anforderungen besteht aus der Detaillierung der Use Cases. Dabei werden die Abläufe und die zugehörigen Aktionen der Use Cases näher betrachtet. Dies ist u. a. mit UML Aktivitätsdiagrammen möglich [Obj11a, S. 303][RQ07, S. 260ff][Rup13, S. 69]. Eine sehr verbreitete Alternative ist die Beschreibung der Use Cases in natürlicher Sprache [Coc03] (vgl. Abschnitt 2.2). In Abbildung 4.7 ist ein Aktivitätsdiagramm für den Use Case UC 1 der Komponente **Online Shop 1** dargestellt. Das äußere Oval wird als Aktivität bezeichnet, die die Aktionen des Use Cases kapselt. Für eine Aktivität können Vor- und Nachbedingungen festgelegt werden. Hier wird als Vorbedingung (**«precondition»**) die Anmeldung des **Online Shop 1** am **Back-End** gefordert. Diese muss zum Startpunkt der Aktivität erfüllt sein. Die Nachbedingung (**«postcondition»**) definiert, dass nach einem erfolgreichen Ablauf ein neuer Kunde angelegt wurde. Die Aktionen innerhalb der Aktivität sind aus der Sicht des Kunden beschrieben, d. h. das interne Verhalten des Systems ist nicht sichtbar, sondern nur die Reaktion auf eine Aktion des Kunden.

Neben den Aktionen gibt es die Kontrollelemente Start-, End-, Verzweigungs- und Verbindungsknoten. Der Startpunkt markiert den Start des Ablaufs, der durch gerichtete Kanten zwischen Aktivitäten und Kontrollelementen zu dem Ende des Ablaufs in Form des Endknoten führt. Die Rauten stehen für die Verzweigungs- und Verbindungsknoten. Verzweigungsknoten besitzen mehr als eine ausgehende Kante und nur eine eingehende Kante. Für Verbindungsknoten gilt, dass sie mit nur einer ausgehenden Kante und mit mehr als einer eingehenden Kante verbunden sind. Durch die Verzweigungs- und Verbindungsknoten

gibt es mehr als einen eindeutigen Ablauf für das dargestellte Aktivitätsdiagramm. In der Regel gibt es einen einfachen Standardablauf für den erfolgreichen Abschluss eines Use Cases, der im Folgenden als Hauptszenario bezeichnet wird [Coc03, S. 116]. Die weiteren Abläufe werden als alternative Erfolgs- und Fehlerszenarien bezeichnet [Coc03, S. 131]. In dem Aktivitätsdiagramm in Abbildung 4.7 wird ein Fehlerszenario durch das Fehlen von freien Kundennummern ausgelöst. Ein alternatives Erfolgsszenario ist mit dem Klick des Kunden auf **Andere Kundennummer** möglich.

Im Systemtest muss die Umsetzung der Hauptszenarien und auch der alternativen Erfolgs- und Fehlerszenarien getestet werden [BM10, S. 59]. Dazu müssen die zugehörigen Aktivitäten durch den Tester ausgeführt und die Reaktion des Systems mit dem erwarteten Verhalten abgeglichen werden. Daher werden für die Bewertung der Komplexität der Testobjekte die Anzahl der Aktionen und die Anzahl der Szenarien je Use Case gezählt. Die Bestimmung der Anzahl der Szenarien ist nur bei azyklischen Abläufen eindeutig. Die Schleife in Abbildung 4.7 über den Klick des Kunden auf **Andere Kundennummer** führt zu mehreren Szenarien, da der Klick wiederholt werden kann. Für eine einheitliche Zählung werden die Szenarien so definiert, dass für die Abdeckung *Use Case Grenze-Inneres-Überdeckung*  $c_{GI}^{UC} = 1$  gilt [Win99, S. 119f]. Danach muss für jede Schleife im Aktivitätsdiagramm jeweils mindestens ein Szenario existieren, das die Schleife keinmal, genau einmal und mindestens zweimal durchläuft<sup>2</sup>. Für das Aktivitätsdiagramm aus Abbildung 4.7 müssten damit die folgenden Szenarien berücksichtigt werden:

- Hauptszenario ohne Klick auf **Andere Kundennummer**
- Alternatives Erfolgsszenario mit einmaligem Klick auf **Andere Kundennummer**
- Alternatives Erfolgsszenario mit zweimaligem Klick auf **Andere Kundennummer**
- Fehlerszenario für den Fall das keine freie Kundennummer verfügbar ist.

---

<sup>2</sup>Es gibt zahlreiche alternative Möglichkeiten mit Schleifen in Aktivitätsdiagramm umzugehen. Eine ist die *All-one-loop-paths* Abdeckung [UL07, S. 118][Mic11, S. 140]. Um die Abdeckung zu erreichen, muss jeweils mindestens ein Szenario existieren, das die Schleife keinmal oder genau einmal durchläuft.

#### 4. Entwicklung des Modells für Testkomplexität

Die gezählten Aktionen und Szenarien werden anschließend analog zur Use Case Point Methode von Frohnhoff bewertet [Fro09, S. 60]. Dabei wird ein Szenario einfach gezählt und eine Aktion mit  $\frac{1}{3}$ . Falls das Ergebnis keine ganze Zahl ist, wird entsprechend abgerundet. Die in den Use Case Diagrammen dargestellten «*include*» und «*extend*» Beziehungen werden bei der Zählung nur einmalig berücksichtigt, da die Funktionalität in der Regel nur einmal getestet wird. Damit werden Use Cases, die von mehreren anderen Use Cases über eine «*include*» oder «*extend*» Beziehung integriert werden, nur einmal in der Zählung berücksichtigt.

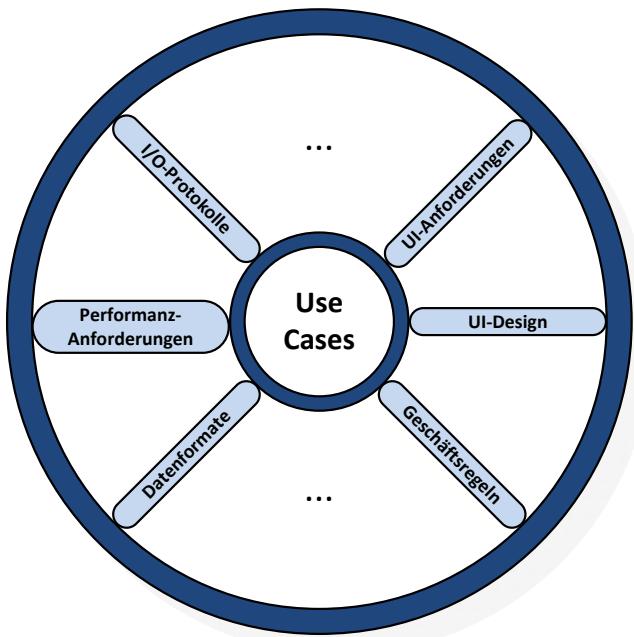


Abbildung 4.8.: Das *Nabe-Speiche*-Modell der Anforderungen nach [Coc03, S. 32]

Neben den spezifizierten Use Cases existieren weitere Anforderungen an das zu testende System. Dies umfasst insbesondere die nicht-funktionalen Anforderungen. Diese weiteren Anforderungen können durch Use Cases verknüpft werden [Coc03, S. 32]. Dabei bildet ein Use Case wie in Abbildung 4.8 das zentrale Element, das durch weitere Anforderungen referenziert wird. Für den Test der Use Cases müssen die mit ihnen verknüpften Anforderungen berücksichtigt werden. Daher muss für die Bewertung der Komplexität der Testobjekte die Anzahl der weiteren Anforderungen gezählt werden. Dabei werden die Anforderungen genauso wie Szenarien einfach gezählt, da von mindestens einem weiteren Szenario je Anforderung ausgegangen wird.

In Abbildung 4.9 sind die in den drei Ebenen identifizierten Eigenschaften der Testobjekte für den Systemtest in einem Metamodell noch einmal zusammengefasst. Die Testkomplexität für den Systemtest ist auf Basis jeder Spezifikation bestimmbar, aus der die im Metamodell enthaltenden Elemente ableitbar sind. Damit die Ableitung nicht von Anwender zu Anwender verschieden sind, müssen Vorgaben bereitgestellt werden. Diese sind in Form einer Abbildung der genutzten Modellierungssprache zur Erstellung der Spezifikation auf das in Abbildung 4.9 gezeigte Metamodell zu definieren.

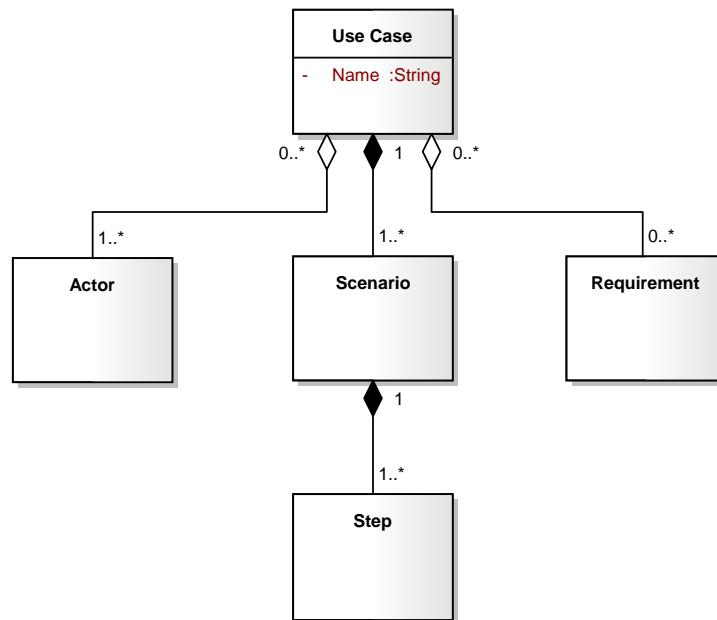


Abbildung 4.9.: Die zu berücksichtigenden Eigenschaften der Testobjekte für den Systemtest

### Standardtestverfahren für die Teststufe Systemtest

Für den Systemtest eignet sich das Use Case basierte Testen [Int11, S. 41][BM10, S. 99f][SL10, S. 145ff]. Dabei sind die Szenarien, Hauptscenario und alternative Erfolgs- und Fehlerszenarien, der Use Cases zu testen. Ein Testfall für einen Use Case bildet jeweils ein Szenario ab und folgt schrittweise den Aktionen, die durch das Aktivitätsdiagramm für das Szenario beschrieben sind. Der Testfall ist ggf. mehrfach mit unterschiedlichen Testdaten auszuführen, um die Umsetzung weiterer Anforderungen zu testen.

## 4. Entwicklung des Modells für Testkomplexität

### Schätzung der Testkomplexität für den Systemtest

Die gezählten Elemente aus den Use Case Diagrammen und Aktivitätsdiagrammen werden als Messergebnisse in die Formel 4.2 eingesetzt. Der Aufbau der Formel folgt der Idee aus [Fro09, S. 58]. Dabei berechnet sich die Testkomplexität für den Systemtest  $K_{Systemtest}$  aus der Summe der gewichteten Akteure  $GA_m$  und der Summe der gezählten Elemente ( $AS_n$ ,  $AAU_n$ ,  $AAF_n$ ) je Use Case.

$$K_{Systemtest} = \sum_{m=1}^{AA} GA_m + \sum_{n=1}^{AUC} \left( AS_n + \left\lfloor \frac{1}{3} \cdot AAU_n \right\rfloor + AAF_n \right) \quad (4.2)$$

$AA$  = Anzahl Akteure

$GA_m$  = Gewicht von Akteur  $m$

$AUC$  = Anzahl Use Cases

$AS_n$  = Anzahl Szenarien für Use Case  $n$

$AAU_n$  = Anzahl Aktionen für Use Case  $n$

$AAF_n$  = Anzahl Anforderungen für Use Case  $n$

Beschreibung	Annahme
Gewichtung Akteur	2
Summe gezählte Elemente je Use Case	2

Tabelle 4.3.: Annahmen für die Berechnung der Testkomplexität für einen Systemtest in frühen Projektphasen oder bei nicht vollständiger Dokumentation

Abhängig vom Zeitpunkt der Schätzung oder der Qualität der Testbasis sind evtl. keine Informationen über die beteiligten Akteure oder keine Detailinformationen zu den Use Cases verfügbar. Um dennoch eine Bestimmung der Testkomplexität zu ermöglichen, werden die Annahmen in Tabelle 4.3 für die Berechnung vorgeschlagen. Die Annahmen orientieren sich an den Vorschlägen, die für die Methode Use Case Point getroffen worden sind [Fro09, S. 67f]. Dabei sind in einem mittelgroßen Softwareprojekt (2.500 - 5.000 Personentage) durchschnittlich der Wert 2 für die Gewichtung von Akteuren und 1,2 für die Summe der gezählten Elemente je Use Case anzunehmen. Der Wert 1,2 wurde hier aus eigener Erfahrung und der Auswertung, der später in der Evaluation betrachte-

ten Testprojekte, auf 2 erhöht, da in der Regel von einer zusätzlich zu testenden Anforderung je Use Case auszugehen ist.

### 4.1.3. Abnahmetest

Testobjekte für den Abnahmetest sind u. a. das entwickelte System und die Geschäftsprozesse im integrierten Systemverbund [Int11, S. 26][SL10, S. 64f]. Das Testziel ist die Überprüfung durch den Kunden, ob das System die vertraglich vereinbarten Ziele erfüllt. Dabei werden Teile des Systemtests wiederholt, so dass der Kunde bewerten kann, ob die Akzeptanzkriterien erfüllt sind. Ergänzend kann die Ausführung von Geschäftsprozessen im integrierten Systemverbund geprüft werden. Die Testbasis besteht genauso wie beim Systemtest aus Dokumenten, die u. a. die Anforderungsspezifikation und die funktionale Spezifikation beinhalten. Für den Teil des Abnahmetests, der eine Wiederholung von Teilen des Systemtests darstellt, wird die Testkomplexität genauso wie für den Systemtest berechnet. Dabei werden nur die Use Cases berücksichtigt, die im Abnahmetest geprüft werden sollen. Die Bestimmung der Komplexität für die ergänzende Ausführung von Geschäftsprozessen wird im Folgenden diskutiert.

#### Testbasis für die Teststufe Abnahmetest

Die im Abnahmetest zu prüfenden Geschäftsprozesse können mit UML Aktivitätsdiagrammen spezifiziert werden [RQ07]. Eine verbreitete Alternative ist die Modellierung der Geschäftsprozesse mit BPMN Diagrammen [Obj11b]. In Abbildung 4.10 ist ein Aktivitätsdiagramm zu sehen, das den Geschäftsprozess GP 1 über die in Abbildung 4.2 gezeigten Systeme darstellt. Die Aktionen des GP 1 sind über die Aktivitätsbereiche **Kunde**, **Online Shop 1**, **Back-End** und **Logistik** verteilt. Damit wird deutlich, wer für die Ausführung der Aktionen verantwortlich ist. Für die Bewertung der Komplexität wird auf das Vorgehen von Frohnhoff zurückgegriffen [Fro09, S. 88]. Dazu werden in dem Geschäftsprozess Use Cases unter der Annahme identifiziert, dass ein Use Case durch einen oder wenige Aktionen im Geschäftsprozess repräsentiert wird. In dem GP 1 stehen z. B. die ersten drei Aktionen für den Use Case UC 1 aus Abbildung 4.7. Falls keine weiteren Beschreibungen zu Aktionen oder Szenarien in den Use Cases vorliegen, sind die Annahmen aus Tabelle 4.3 zu verwenden. Anschließend

#### 4. Entwicklung des Modells für Testkomplexität

müssen die Akteure genauso wie beim Systemtest gezählt und gewichtet werden. Dazu sind die Verantwortlichen aus den Aktivitätsbereichen entsprechend der Tabelle 4.2 zu bewerten.

In Abbildung 4.11 sind die identifizierten Eigenschaften der Testobjekte für den Abnahmetest in einem Metamodell noch einmal zusammengefasst. Der Anteil mit Use Cases wurde aus dem Metamodell für den Systemtest (vgl. Abbildung 4.9) wiederverwendet. Die Testkomplexität für den Abnahmetest ist auf Basis jeder Spezifikation bestimmbar, aus der die im Metamodell enthaltenen Elemente ableitbar sind. Damit die Ableitungen nicht von Anwender zu Anwender verschieden sind, müssen Vorgaben bereitgestellt werden. Diese sind in Form einer Abbildung der genutzten Modellierungssprache zur Erstellung der Spezifikation auf das in Abbildung 4.11 gezeigte Metamodell zu definieren.

#### Standardtestverfahren für den Abnahmetest

Für den Abnahmetest eignet sich ein ähnliches Vorgehen wie beim Use Case basierten Testen [Int11, S.41][BM10, S.99f][SL10, S.145ff]. Dazu sind die Szenarien, das Hauptszenario und alternative Erfolgs- und Fehlerszenarien der Geschäftsprozesse zu testen. Ein Testfall für einen Geschäftsprozess bildet jeweils ein Szenario ab und folgt schrittweise den Aktionen/Use Cases, die durch das Aktivitätsdiagramm für das Szenario beschrieben sind. Der Testfall ist ggf. mehrfach mit unterschiedlichen Testdaten auszuführen, um die Umsetzung weiterer Anforderungen zu testen.

#### Schätzung der Testkomplexität für die Teststufe Abnahmetest

Die Testkomplexität für den Abnahmetest wird mit Formel 4.3 berechnet. Dabei ergibt sich die Testkomplexität für den Abnahmetest  $K_{Abnahmetest}$  aus der Testkomplexität des zu wiederholenden Teils aus dem Systemtest und den dazugekommenen Teilen für den Abnahmetest. Die neuen Anteile für den Abnahmetest werden wie in der Formel 4.2 für den Systemtest berechnet. Zu berücksichtigen ist hier, dass die Use Cases nur einmal für die Berechnung der Testkomplexität des Abnahmetests verwendet werden sollen. Damit dürfen die Use Cases, die schon in der Testkomplexität  $K_{WS}$  berücksichtigt sind, nicht in der Summe über die identifizierten Use Cases berücksichtigt werden.

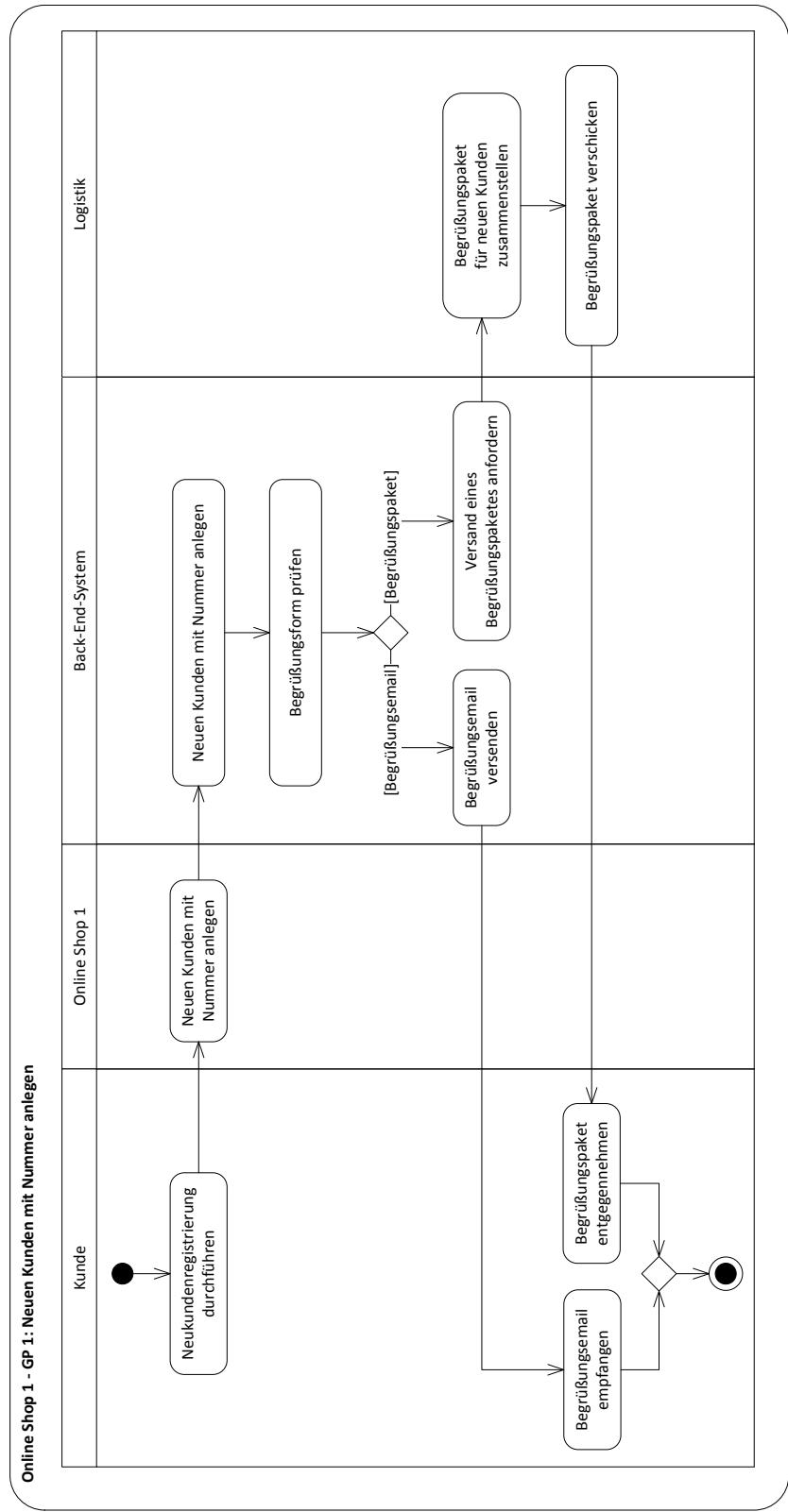


Abbildung 4.10.: Darstellung eines Geschäftsprozesses als UML Aktivitätendiagramm

#### 4. Entwicklung des Modells für Testkomplexität

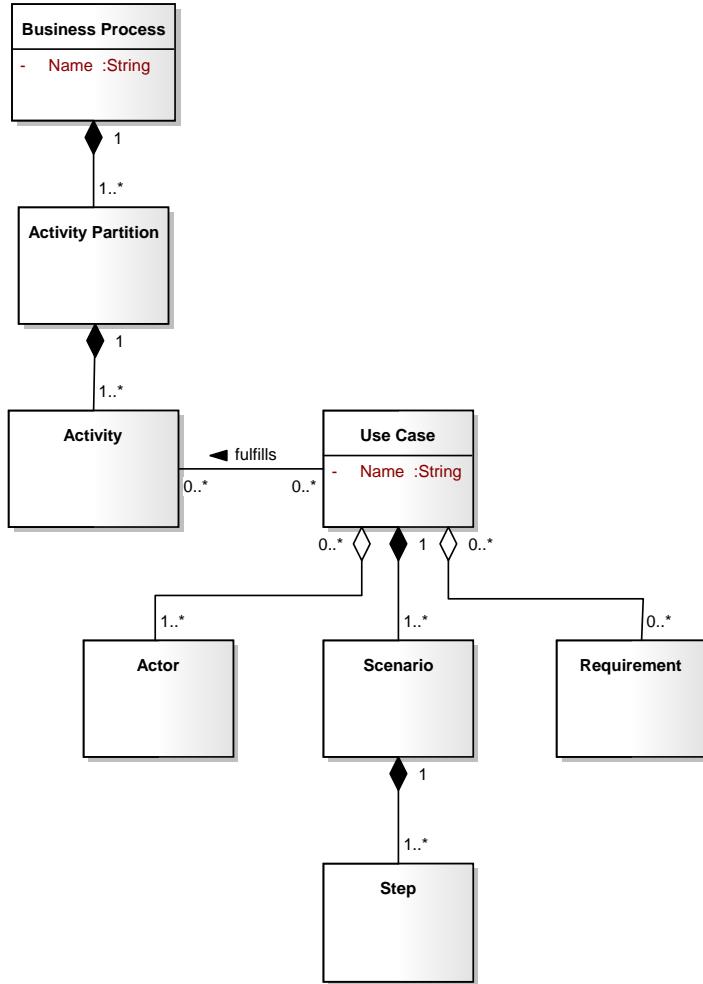


Abbildung 4.11.: Die zu berücksichtigenden Eigenschaften der Testobjekte für den Abnahmetest

$$K_{\text{Abnahmetest}} = K_{WS} + \sum_{m=1}^{AAB} GV_m + \sum_{n=1}^{AIU} \left( AS_n + \left\lfloor \frac{1}{3} \cdot AAK_n \right\rfloor + AAF_n \right) \quad (4.3)$$

Testkomplexität für den Anteil des System-  
 $K_{WS}$  = tests, der für den Abnahmetest wiederholt  
 werden soll.

$AAB$  = Anzahl Aktivitätsbereiche

$GV_m$  = Gewicht von Verantwortlichem  $m$

$AIU$  = Anzahl identifizierter Use Cases

$AS_n$  = Anzahl Szenarien für Use Case  $n$

$AAK_n$  = Anzahl Aktionen für Use Case  $n$

$AAF_n$  = Anzahl Anforderungen für Use Case  $n$

Von Frohnhoff wird der allgemeine Hinweis gegeben, dass bei der Bewertung darauf zu achten ist, ob tatsächliche Geschäftsprozesse vorliegen und nicht Use Cases, wie z. B. im Abschnitt 4.1.2 dargestellt [Fro09, S. 88]. Wenn keine Geschäftsprozesse vorliegen, kann die Bewertung der Komplexität genauso wie für den Systemtest erfolgen, d. h. es müssen keine Use Cases identifiziert werden.

#### 4.1.4. Fazit

Mit dem Modell für Testkomplexität ist die Bewertung der Komplexität für die Testobjekte der Teststufen (System-)Integrationstest, Systemtest und Abnahmetest durchführbar. Voraussetzung ist, dass die Dokumente der Testbasis in entsprechendem Detaillierungsgrad vorliegen, der eine Belegung der Variablen in den drei vorgestellten Formeln ermöglicht.

In den Abschnitten 4.1.1, 4.1.2 und 4.1.3 wurde zur Erläuterung der verwendeten Begriffe durchgängig auf den UML Standard gesetzt. Wenn die Dokumente der Testbasis nicht mit Hilfe der UML erstellt wurden, kann die Bewertung der Testkomplexität trotzdem wie erläutert durchgeführt werden. Dazu muss allerdings eine Abbildung der Begriffe der Spezifikationstechnik auf die oben verwendeten Begriffe geschaffen werden, so dass die Variablen der Formeln belegt werden können. Dies sollte auf einem einheitlichen Abstraktionsgrad erfolgen, so dass die Testkomplexität über verschiedene Spezifikationstechniken hinweg vergleichbar bleibt.

## 4.2. Evaluation des Modells für Testkomplexität

Die Evaluation des Modells für Testkomplexität dient dem Nachweis, dass es als Grundlage für die weiteren Berechnungen in Kapitel 5 ausreicht. Für den Nachweis werden folgende zwei vorbereitende Schritte unternommen, die in Abschnitt 4.2.1 beschrieben sind:

1. Anwendung des Modells für Testkomplexität auf Basis von Daten aus abgeschlossenen Testprojekten des Unternehmens CRM IT (vgl. Abschnitt 4.2.1.1)

#### *4. Entwicklung des Modells für Testkomplexität*

2. Berechnung der globalen Produktivität für das Unternehmen CRM IT und eine detaillierte Analyse der lokalen Produktivität über mehrere Testprojekte (vgl. Abschnitt 4.2.1.2)

Anschließend wird in Abschnitt 4.2.2 das Modell für Testkomplexität zusammenfassend beurteilt. Dabei wird gezeigt, dass ein linearer Zusammenhang zwischen der ermittelten Testkomplexität und den Aufwänden für die Testaktivitäten besteht.

##### **4.2.1. Empirische Untersuchung über Daten aus abgeschlossenen Testprojekten**

Eine empirische Untersuchung, die ihre Beobachtungen anhand realer Softwareprojekte gewinnt, wird als Feldstudie bezeichnet [Pre01, S. 43]. Eine Spezialisierung davon, die als Software-Archäologie bezeichnet wird, leitet die Beobachtungen nachträglich aus einem Softwareprojekt ab [Pre01]. Diese Form der Untersuchung wurde hier auch angewendet, d.h. es findet kein Eingriff in ein laufendes Testprojekt statt, sondern die Beobachtungen werden aus Daten gewonnen, die sowieso während des Testprojektes zur Verfügung standen oder aufgezeichnet wurden.

Für den ersten Bestandteil der Evaluation, die Anwendung des Modells für Testkomplexität, musste zunächst die Testbasis der Testprojekte ermittelt werden. Zusätzlich wurden für die Projekte die erstellten Testmittel, wie Testfälle, Testkonzept und Testpläne, gesammelt, um eine stärkere Eingrenzung auf die tatsächlich getestete Funktionalität vornehmen zu können. Auf den Artefakten der Testbasis und den Testmitteln wurde anschließend das Modell für Testkomplexität angewendet, um die Komplexität in TKP für jedes Testprojekt zu ermitteln.

Für die Erfüllung des zweiten Bestandteils, die Berechnung der durchschnittlichen Produktivität, werden neben der Komplexität in TKP für jedes Testprojekt die tatsächlich geleisteten Aufwände in Personentagen getrennt nach Testaktivität und Teststufe benötigt. Dafür wurden aus dem System, das die Mitarbeiter der CRM IT nutzen, um die Aufwände für jedes durchgeführte Projekt für die

Abrechnung zu protokollieren, die Zeiterfassungen der Mitarbeiter exportiert und analysiert.

In Abschnitt 4.2.1.1 wird die entwickelte Werkzeugunterstützung für die beiden Bestandteile vorgestellt. Danach wird in Abschnitt 4.2.1.2 die Berechnung der lokalen und globalen Produktivität erläutert.

#### **4.2.1.1. Aufbau einer Projektdatenbank**

Für die Analyse und die Berechnung der Produktivität müssen die Daten strukturiert verwaltet werden. Dies war ohne Werkzeugunterstützung bei ca. 200 Testprojekten nicht mehr zu gewährleisten. Daher wurde die TAQ Anwendung implementiert, die die Datenerfassung, die Analyse und die Berechnungen unterstützt, die für diese Dissertation notwendig sind.

In Abbildung 4.12 ist eine Übersicht über die wesentlichen Komponenten zu sehen. Die Hauptkomponenten sind die TAQ Anwendung und die TAQ Datenbank. Die Komponenten der TAQ Anwendung sind über eine graphische Benutzeroberfläche benutzbar. Alle Komponenten der TAQ Anwendung greifen auf die TAQ Datenbank zu, in der die Daten zu den Testprojekten zentral abgelegt sind. Die Datenbank besteht aus einer strukturierten Dateiablage und einer relationalen Datenbank.

In der Dateiablage wurden für jedes Testprojekt die Testmittel in einer vordefinierten Ordnerstruktur abgelegt. Zu den Testmitteln gehörten speziell die Testkonzepte, Testpläne und Testabschlussberichte, da diese eine Eingrenzung der getesteten Funktionalität erlauben und später weiterhin in Kapitel 5 zur Evaluation des Modells für Testeinflussfaktoren genutzt werden.

In der relationalen Datenbank wurden als erstes Stammdaten zu den Testprojekten, wie der Name, die Teststufe, die beteiligten Mitarbeiter und ihre Projektrolle, erfasst. Anschließend wurden zu den Testprojekten über eine Importschnittstelle Daten zu Stundenbuchungen der Mitarbeiter eingelesen. Diese Stundenbuchungen wurden entweder über den mitgelieferten Beschreibungstext oder der Rolle des Mitarbeiters in dem Testprojekt automatisch einer Testaktivität sowie einer Teststufe zugeordnet und in der relationalen Datenbank gespeichert. Eine Übersicht über die absolute Anzahl der importierten Testpro-

#### 4. Entwicklung des Modells für Testkomplexität

jekte, der Mitarbeiter und den Stundenbuchungen ist in diesem Abschnitt in Tabelle 4.4 zu finden. Die Stundenbuchungen werden in aggregierter Form im Anhang B.1 aufgeführt.

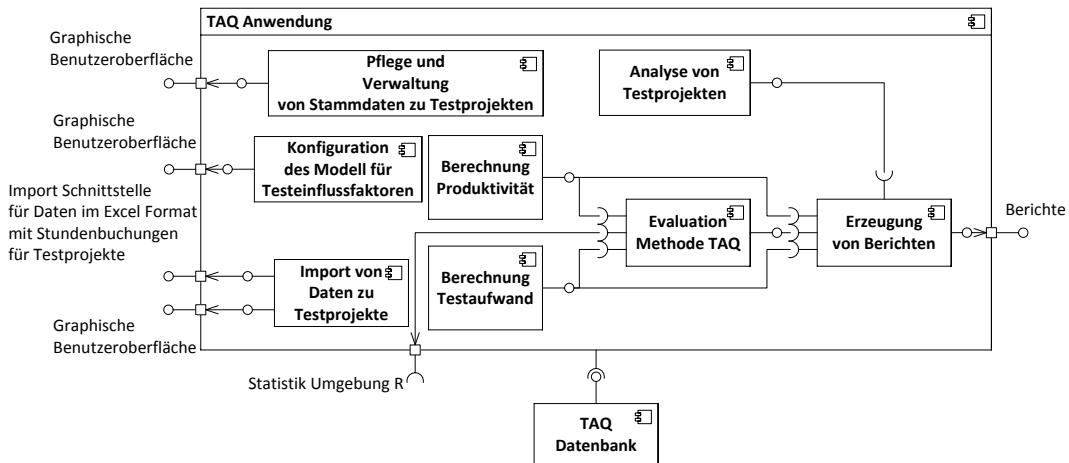


Abbildung 4.12.: Übersicht über die Werkzeugunterstützung als Komponentendiagramm

Nach der Erfassung der Stammdaten eines Testprojektes und dem Import der Stundenbuchungen wurde das Modell für Testkomplexität angewendet, d.h. für jedes Testprojekt wurde die Testkomplexität in TKP ermittelt. Hierfür wurde in der TAQ Anwendung eine Maske für die Erfassung der gemessenen Eigenschaften angeboten. Eine Übersicht über die Gesamtzahl der gemessenen Eigenschaften ist in diesem Abschnitt in Tabelle 4.4 zu finden. Im Anhang B.1 wird neben den aggregierten Stundenbuchungen je Testaktivität auch die Testkomplexität aufgeführt.

	Anzahl (System-) Integrationstest	Anzahl		Gesamt
		Systemtest	Abnahmetest	
Testprojekte	20	95	36	151
Mitarbeiter	225	344	163	732
Stundenbuchungen	1.495	7.792	1.224	10.511
Use Cases		2640	160	2.800
Akteure		667	131	798
Kompositionskonnek.	112			112
Operationen	250			250

Tabelle 4.4.: Anzahl Datensätze in der relationalen TAQ Datenbank

## 4.2. Evaluation des Modells für Testkomplexität

Für die im Abschnitt 4.2.2 folgende Beurteilung des Modells und die Evaluierung in Kapitel 5 ist die Validität der Daten von zentraler Bedeutung. Daher wurden mögliche Risiken identifiziert und Maßnahmen zur Risikominimierung getroffen. Für die folgenden Punkte wird die Validität näher betrachtet, da diese die Eingaben für die Anwendung des Modells für Testkomplexität sind:

- Validität der Stundenbuchungen durch die Mitarbeiter
- Validität der Zuordnung der Stundenbuchungen zur Testaktivitäten und Teststufen
- Validität der Testmittel

Die Validität einer Stundenbuchung ist hoch, wenn die Stundenbuchungen den tatsächlichen Aufwand des Mitarbeiters für ein Testprojekt wiedergeben. Anders formuliert müssen für die Höhe der Stundenbuchung mögliche Alternativerklärungen mit hoher Wahrscheinlichkeit ausgeschlossen werden können. Bei den Buchungen gibt es das Risiko, dass die Höhe der Buchung an das vorhandene Budget angepasst worden ist. Dafür kann es sehr unterschiedliche Gründe geben, wie z. B. der Mitarbeiter arbeitete *langsamer*, er hat mehr als notwendig für das Testprojekt gearbeitet, er hat Aufwände eines anderen sich in Schieflage befindlichen Testprojektes für das aktuelle Testprojekt ausgewiesen oder das Testprojekt wurde für beendet erklärt, als das Budget verbraucht gewesen war. Daneben besteht die Möglichkeit, dass der Mitarbeiter versehentlich falsche Buchungen vorgenommen hat. Bei den Stundenbuchungen der Mitarbeiter in der Abteilung für Qualitätssicherung (QM) von dem Unternehmen CRM IT können diese Gründe mit hoher Wahrscheinlichkeit ausgeschlossen werden. Dies liegt in erster Linie daran, dass das Unternehmen als *Cost Center* und interner Dienstleister bei arvato keine Gewinnmaximierung verfolgt, sondern nur kostendeckend agieren muss. Damit besteht zumindest aus Unternehmenssicht kein Grund für *langsameres* Arbeiten oder nicht notwendige Mehrarbeit in einem Testprojekt. Weiterhin gilt für QM, dass der Abschluss eines Testprojektes nicht unnötig verzögert wird. Auch wird durch die enge Zusammenarbeit in relativ kleinen Teams mit gegenseitiger Kontrolle nicht notwendige Mehrarbeit für ein Testprojekt nahezu ausgeschlossen. Das Verschieben von Stundenbuchungen eines Testprojektes zu einem anderen und mögliche Fehler in den Buchungen des Mitarbeiters werden durch das kontinuierliche Kontrollieren der jeweils verantwortlichen Testmanager und auch der Gesamtprojektleitung verhindert.

#### *4. Entwicklung des Modells für Testkomplexität*

Von der Ermittlung der richtigen Testaktivität und Teststufe zu einer Stundenbuchung hängt die Validität der Zuordnung der Stundenbuchungen zu Testaktivitäten und Teststufen ab. Hier besteht als erstes das Risiko der falschen Beschreibung der Stundenbuchung durch den Mitarbeiter. Dieses Risiko wurde durch Vorgaben für zu verwendende Texte bei der Beschreibung der Stundenbuchungen bei QM minimiert. Als zweites besteht das Risiko der falschen automatischen Zuordnung der Stundenbuchung zu einer Testaktivität und Teststufe. Um dieses Risiko zu minimieren wurde die Beschreibung anhand einer hinterlegten Wortliste geprüft und in unklaren Fällen die Zuordnung über die Rolle des Mitarbeiters im Projekt getroffen. Zusätzlich wurde die Zuordnung zu Testaktivität und Teststufe durch eine Sichtprüfung kontrolliert und bei Bedarf nachbearbeitet.

Die Validität der Testmittel ist hoch, wenn sie einen Aufschluss über die getestete Funktionalität geben und die Möglichkeit zur Anwendung des Modells für Testkomplexität geben. Dabei gibt es das Risiko, dass die Artefakte der Testmittel nur unzureichenden Rückschluss über das Testprojekt zulassen. Das Risiko wurde individuell für jedes Testprojekt bewertet. Wenn dabei ein hohes Risiko festgestellt wurde, ist mit den verantwortlichen Testmanagern gesprochen und, wenn möglich, sind die fehlenden Testmittel ergänzt worden. Falls dies nicht möglich war, wurden die Testprojekte nicht in die TAQ Anwendung eingepflegt.

Zusammenfassend kann die Validität der gesammelten Testprojektdaten bestätigt werden. Damit können die weiteren Schritte, wie die Berechnung der Produktivität im folgenden Abschnitt 4.2.2 und auch die Evaluation des in Kapitel 5 beschriebenen Modell für Testeinflussfaktoren auf Basis dieser Daten erfolgen.

##### **4.2.1.2. Berechnung der Produktivität auf Basis der gesammelten Daten**

Nach der Sammlung der Testprojektdaten wurde die Produktivität in den Testprojekten je Testaktivität und Teststufe mit der Formel  $\frac{TKP}{PT}$  berechnet (vgl. Definition 3). Dabei wurden nur die Testprojekte berücksichtigt, die einen Aufwand größer Null für die jeweilige Teststufe und Testaktivität besaßen (vgl. Tabellen im Anhang B.1). Über diese Produktivitäten konnte anschließend eine

## 4.2. Evaluation des Modells für Testkomplexität

durchschnittliche Produktivität je Testaktivität und Teststufe bestimmt werden.

Die berechnete Produktivität je Testaktivität und Teststufe ist in Tabelle 4.5 zu sehen. Bei der Interpretation der Zahlen muss berücksichtigt werden, dass nur für die Teststufe Systemtest Datensätze in höherer Anzahl vorlagen (vgl. Tabelle 4.4) und damit für diese Teststufe das aussagekräftigste Ergebnis vorliegt. Trotzdem bieten die Werte für die anderen Teststufen erste Anhaltspunkte darüber, wie hoch die Produktivität ist. Im Folgenden wird zunächst das Ergebnis in Tabelle 4.5 mit den eigenen Erfahrungen abgeglichen. Anschließend wird gezeigt, wie die Produktivität über Testprojekte zu einer Software schwanken kann.

(System-) Integrationstest	Systemtest	Abnahmetest
Testmanagement	30,59	12,63
Testdesign	57,02	9,69
Testausführung	36,40	13,85

Tabelle 4.5.: Globale Produktivität für das Unternehmen CRM IT in  $\frac{TKP}{PT}$

Die Werte für die Produktivität bestätigen die eigenen Erfahrungen im Bereich des Testmanagement und auch die bisher erstellten Expertenschätzungen. Dabei kann z. B. ein Testdesigner für die Teststufe Systemtest vier bis fünf Use Cases pro Tag bearbeiten und entsprechende Testfälle erstellen bzw. bestehende anpassen. Die Produktivität für die Testaktivitäten Testmanagement und Testausführung liegen im Allgemeinen etwas höher. Dies spiegelt sich auch in den Zahlen für die Produktivität der Testaktivitäten bzgl. der Teststufe Systemtest in Tabelle 4.5 wieder. Die Produktivität für Testdesign liegt bei 9,69  $\frac{TKP}{PT}$  was ungefähr fünf Use Cases entspricht, wenn die Standardwerte für einen Use Case nach Tabelle 4.3 angenommen werden.

Die Produktivität für den Abnahmetest weicht speziell beim Testmanagement und in der Testausführung von den entsprechenden Werten des Systemtest nach unten ab. Auch dies entspricht den eigenen Erfahrungen, da diese Testaktivitäten beim Abnahmetest durch die höhere Anzahl an Ansprechpartnern aufwendiger sind. Ähnliches gilt für den (System-)Integrationstest. Ein Testdesigner kann nach eigener Erfahrung ein bis zwei Schnittstellen pro Tag bearbeiten und

#### *4. Entwicklung des Modells für Testkomplexität*

die zugehörigen Testfälle erstellen oder bestehende anpassen. Die Produktivität 57,02  $\frac{TKP}{PT}$  entspricht nach den Standardwerten in Tabelle 4.1 ca. zwei Schnittstellen.

Abschließend erfolgt eine exemplarische Betrachtung der Schwankungen der Produktivität in mehreren Testprojekten. Die Testprojekte gehören dabei zu einer Software, die regelmäßig aktualisiert wird und damit getestet werden muss. In Abbildung 4.13 ist eine grafische Darstellung der Produktivität mit Anmerkungen zu den Gründen der Schwankungen zu sehen. Deutlich wird, dass die Produktivität sogar bei derselben Software von Testprojekt zu Testprojekt verschieden ist und in vielen Fällen stark von der globalen Produktivität (vgl. Tabelle 4.5) abweicht. Die Gründe dafür können u. a. in den Möglichkeiten der Wiederverwendung von Testmitteln, der Erfahrung des Personals oder der Stabilität der Testumgebung liegen. Daher reicht die globale Produktivität nicht aus, um eine verlässliche Aussage über den zu erwartenden Testaufwand treffen zu können. Für ein Testprojekt müssen weitere (Test-)Einflussfaktoren berücksichtigt werden, die auf die Produktivität wirken. Daher wird in Kapitel 5 das Modell für Testeinflussfaktoren entwickelt, das eine systematische Berücksichtigung von Testeinflussfaktoren erlaubt.

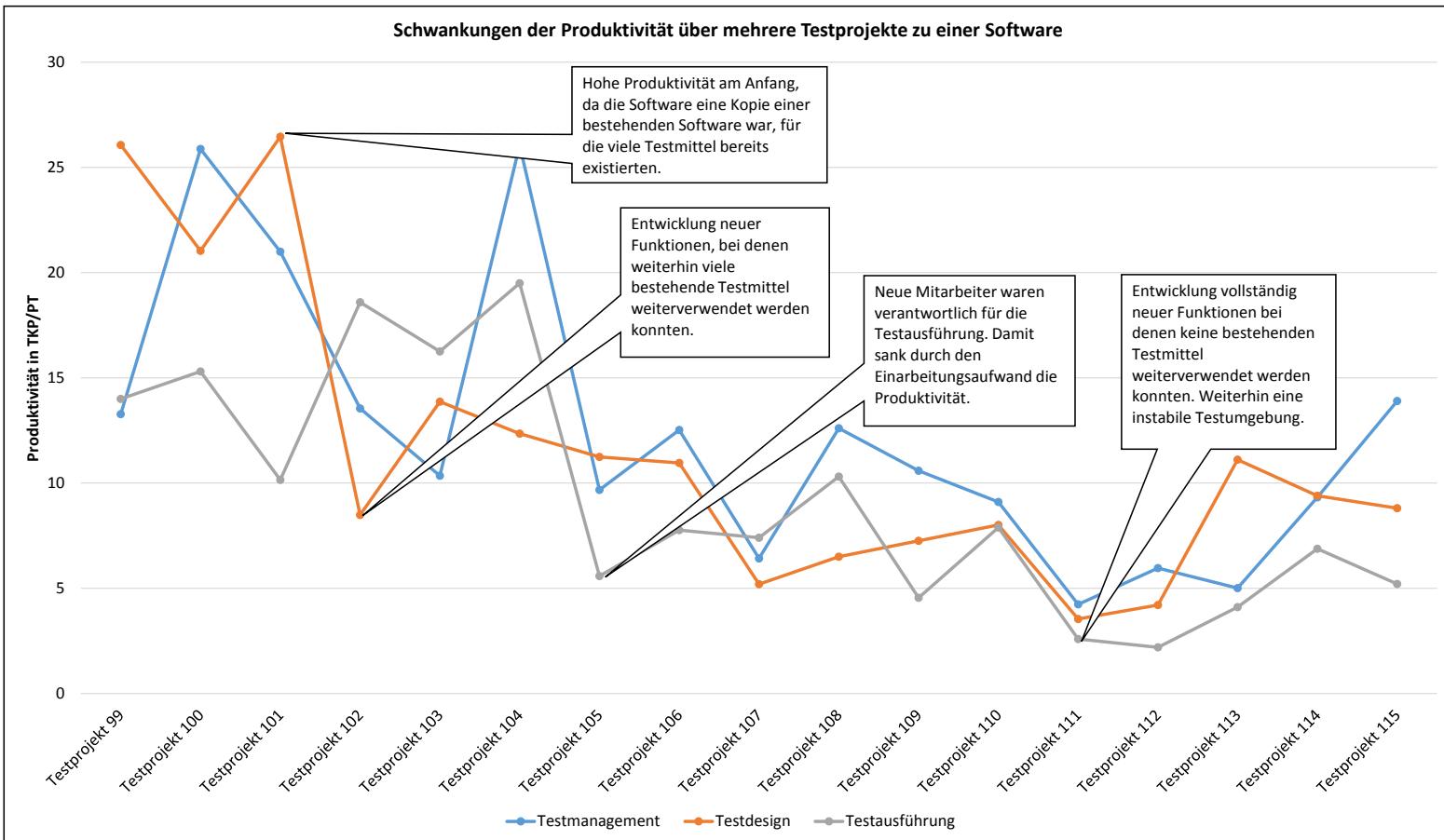


Abbildung 4.13.: Übersicht über Schwankungen der Produktivität über mehrere Testprojekte. Dabei gehören die Testprojekte zu derselben Software in unterschiedlichen Versionen.

### 4.2.2. Beurteilung des Modells

Für die Evaluation in Kapitel 5 wird der lineare Zusammenhang zwischen der Testkomplexität und dem Aufwand der Testaktivitäten, die für den Test der bewerteten Testobjekte notwendig sind, vorausgesetzt. Um dies sicherzustellen, wurde der Zusammenhang zwischen der Testkomplexität und dem Aufwand über die Produkt-Moment-Korrelation [BS10, S. 156ff] analysiert. Für die Berechnung der Korrelation müssen die Werte für die Testkomplexität und das Testmanagement normalverteilt sein [BS10, S. 157]. Daher muss zunächst die Verteilungen der Werte untersucht werden. Anschließend ist die Korrelation zwischen der Testkomplexität und dem Aufwand berechenbar. Im Folgenden wird das Vorgehen exemplarisch für die Testaktivität Testmanagement in der Teststufe Systemtest erläutert.

Die Verteilung der Werte für die Testkomplexität und den Aufwand ist über Histogramme<sup>3</sup> überprüfbar. Wenn die Balken in der grafischen Darstellung zusammen eine gaußsche Glockenkurve ergeben, kann von einer Normalverteilung der Werte ausgegangen werden. In den Abbildungen 4.14 und 4.15 sind jeweils auf der linken Seite Histogramme zu den Werten Testkomplexität und Testmanagement zu sehen. Dabei sind in beiden Histogrammen auf der x-Achse die Kategorien zu sehen, denen die Werte zugeordnet worden sind. An dem Diagramm für Testkomplexität ist zu erkennen, dass die Testprojekte in vielen Fällen zwischen 13 und 184 TKP aufweisen. Gleichzeitig gibt es einige Testprojekte, die deutlich größer sind. Entsprechendes gilt für die Testaktivität Testmanagement, deren Wert in der Mehrzahl der Testprojekte zwischen 1,19 und 17,99 PT liegt. Beide Histogramme sind nach links verzerrt. Dies liegt an den vereinzelten Ausreißern mit hohen Werten. Um dem zu begegnen und sich einer Normalverteilung anzunähern, wird auf die Werte der Logarithmus angewendet. Die darauf basierenden Histogramme sind in den Abbildungen 4.14 und 4.15 auf der rechten Seite zu sehen. Sie sind wesentlich stärker zur Mitte orientiert und der Ansatz einer Normalverteilung ist zu erkennen.

Für eine formale Betrachtung des linearen Zusammenhangs zwischen der Testkomplexität und dem Aufwand der Testaktivitäten wird die Korrelation der Werte über den Korrelationskoeffizienten analysiert. Der Korrelationskoeffizi-

---

<sup>3</sup>Histogramme dienen der Visualisierung von Häufigkeitsverteilungen innerhalb einer Menge an Messwerten. Für eine ausführliche Beschreibung von Histogrammen siehe [BS10, S. 41].

## 4.2. Evaluation des Modells für Testkomplexität

ent beschreibt die Enge des linearen Zusammenhangs zwischen  $x$  und  $y$  [BS10, S. 157]. Bei  $r = 1$  liegt ein perfekter positiver und bei  $r = 0$  liegt kein linearer Zusammenhang vor. Zur Berechnung der Korrelation  $r$  zwischen den beiden Werten  $x$  und  $y$  nach Formel 4.8 ist die Bestimmung des arithmetischen Mittels, der Standardabweichung und der Kovarianz notwendig. Diese grundlegenden Kennzahlen der deskriptiven Statistik werden im Folgenden näher erläutert, da sie bei der Evaluation des Modells für Testeinflussfaktoren in Kapitel 5 wieder verwendet werden.

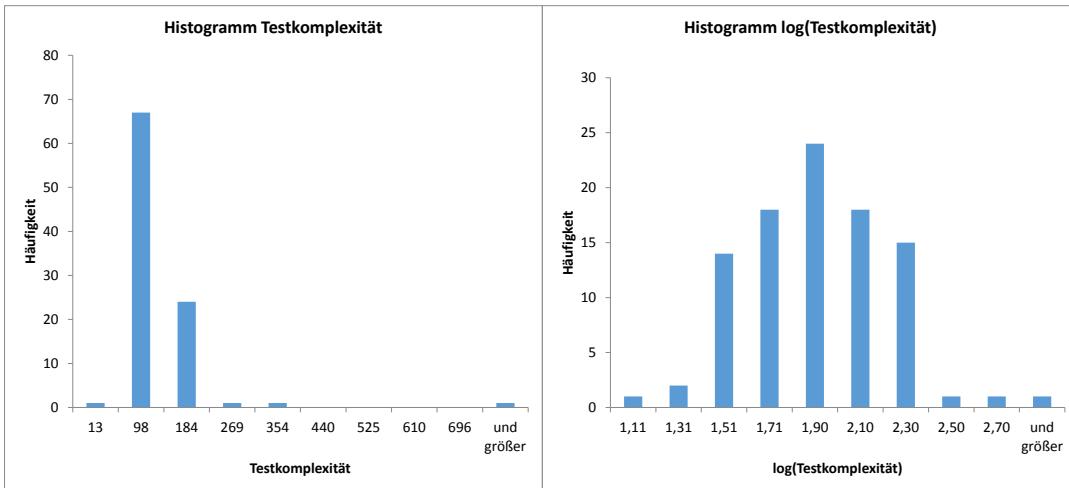


Abbildung 4.14.: Gegenüberstellung der Histogramme für die Testkomplexität (Systemtest) vor und nach der Anwendung des Logarithmus

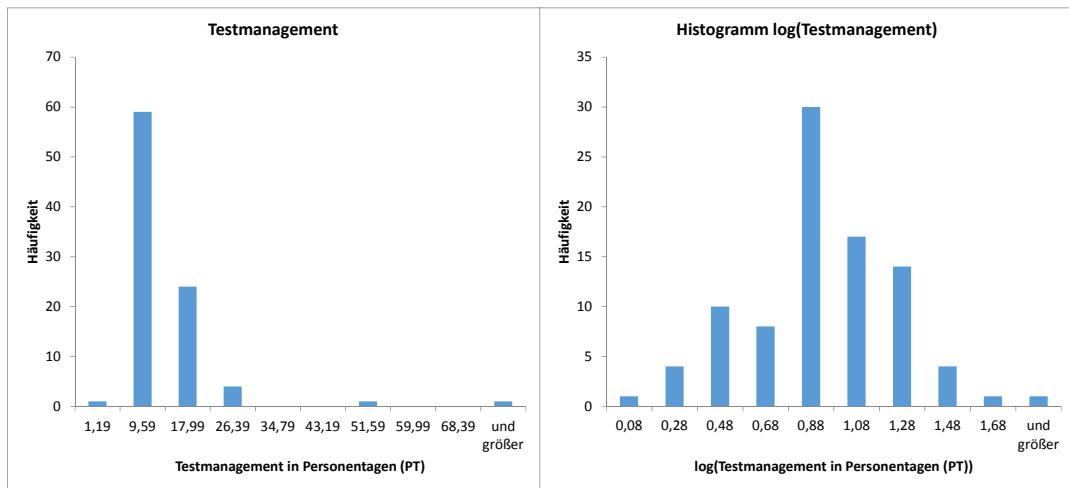


Abbildung 4.15.: Gegenüberstellung der Histogramme für die Testaktivität Testmanagement (Systemtest) vor und nach der Anwendung des Logarithmus

## 4. Entwicklung des Modells für Testkomplexität

### Arithmetisches Mittel

Das arithmetische Mittel dient der Ermittlung der zentralen Tendenz in einer Menge von  $n$  Werten [BS10, S. 25f]. Dazu werden nach Formel 4.4 die Werte addiert und anschließend durch die Anzahl der Werte geteilt. Der Vorteil des arithmetischen Mittels ist, dass jeder Wert in die Berechnung einfließt. Allerdings kann dies gleichzeitig ein Nachteil sein, da Ausreißer in den Werten das arithmetische Mittel verzerren können.

### Standardabweichung

Mit der Standardabweichung kann die Abweichung von der zentralen Tendenz einer Menge von  $n$  Werten angegeben werden (vgl. Formel 4.6). Die Standardabweichung wird aus der Varianz berechnet (vgl. Formel 4.5). „Die Varianz einer Stichprobe des Umfangs  $n$  ist definiert als die Summe der quadrierten Abweichungen aller Messwerte vom arithmetischen Mittel, dividiert durch  $n - 1$ . Wir bezeichnen die Stichprobenvarianz mit  $s^2$ “ [BS10, S. 30f].

### Kovarianz

Die Kovarianz (vgl. Formel 4.7) dient der Beschreibung linearer Zusammenhänge [BS10, S. 153]. Der Wert der Kovarianz gibt ergänzend zu einem Streudiagramm (vgl. z. B. Abbildung 4.16) einen Eindruck über die Stärke und Richtung des linearen Zusammenhangs. Eine positive Kovarianz zeigt, dass hohe  $x$  Werte zu hohen  $y$  Werten führen. Dabei ist der Zusammenhang stärker, umso höher der Wert der Kovarianz ist. Allerdings hängt der Wert der Kovarianz aufgrund der Berechnungsweise vom Maßstab der genutzten Variablen ab. Zu dem Streudiagramm in Abbildung 4.16 gehört z. B. der Wert 0,05, der sehr niedrig erscheint. Allerdings geht der Wertebereich auch nur von 0 bis 3,5.

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.4)$$

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1} \quad (4.5)$$

$$s = \sqrt{s^2} \quad (4.6)$$

$$s_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{n - 1} \quad (4.7)$$

$$r = \frac{s_{xy}}{s_x \cdot s_y} \quad (4.8)$$

Für das Streudiagramm in Abbildung 4.16 wurde der Korrelationskoeffizient  $r$  mit 0,52 bestimmt. Damit liegt ein linearer Zusammenhang vor. Mit diesem Verfahren wurden auch die anderen Teststufen und Testaktivitäten auf den Zusammenhang zur Testkomplexität untersucht. Insgesamt konnte der lineare Zusammenhang zwischen Testkomplexität und Testaufwand bestätigt werden, d.h. das Modell für Testkomplexität wird als Basis für die Evaluation in Kapitel 5 verwendet.

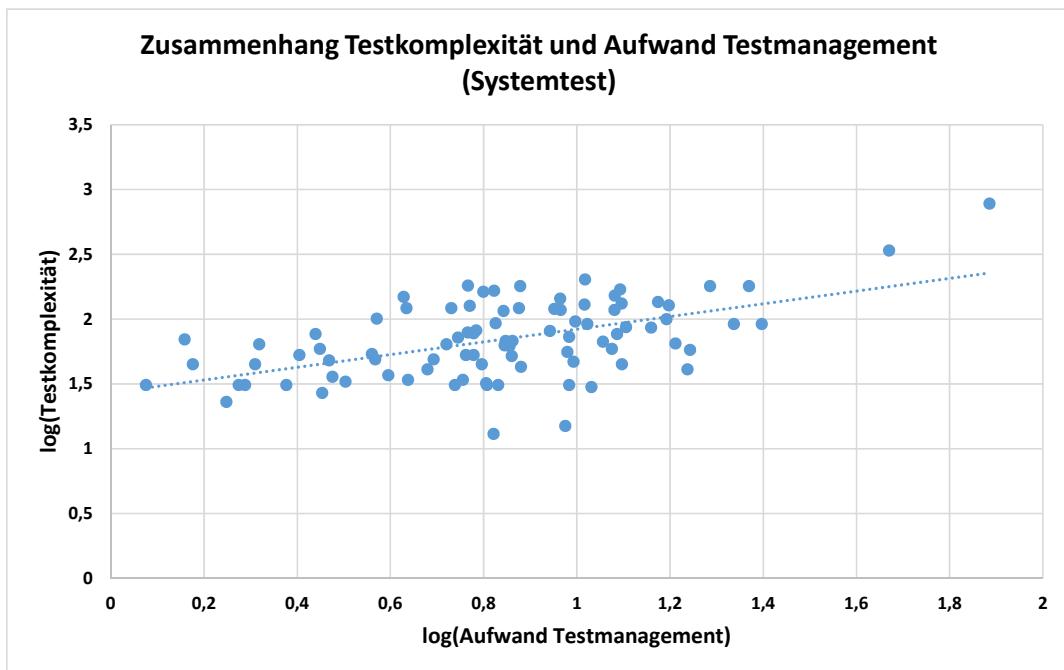


Abbildung 4.16.: Streudiagramm Testkomplexität und Aufwand für die Testaktivität Testmanagement (Systemtest)

## 4.3. Zusammenfassung

In diesem Kapitel wurde das Modell für Testkomplexität beschrieben. Dabei wurde schrittweise für die Teststufen (System-)Integrationstest, Systemtest und Abnahmetest an den Standardmodellierungssprachen der UML gezeigt, welche Eigenschaften die Testobjekte der jeweiligen Teststufe charakterisieren. Anschließend wurde jeweils für jede Teststufe auf typische Testentwurfsverfahren eingegangen und eine Formel zur Schätzung der Testkomplexität entwickelt.

#### *4. Entwicklung des Modells für Testkomplexität*

Damit eine Anwendung auch mit Modellierungssprachen außerhalb der UML möglich ist, wurde je Teststufe ein Metamodell entwickelt, das die strukturellen Abhängigkeiten der zu berücksichtigenden Eigenschaften für die Schätzung der Testkomplexität beschreibt. Die Elemente in den Metamodellen entsprechen Sprachelementen von Modellierungssprachen der UML, worüber sich die Semantik der Elemente ergibt. Über die Metamodelle können nun (formale) Abbildungen von weiteren Modellierungssprachen definiert werden, so dass Spezifikationen, die mit diesen Sprachen erstellt worden sind, als Grundlage für die Schätzung der Testkomplexität verwendet werden können.

Für den Nachweis des linearen Zusammenhangs zwischen der Testkomplexität und dem Testaufwand wurde eine Evaluation über mehr als 120 Testprojekten des Unternehmens CRM IT durchgeführt. Der Schwerpunkt lag hierbei auf den Testprojekten der Teststufe Systemtest, da hier am meisten Daten vorlagen. Die Evaluation zeigte neben dem gewünschten linearen Zusammenhang auch, dass u. a. Kennzahlen, wie die globale Produktivität des Unternehmens, berechenbar und an Einzelprojekten nachvollziehbar sind.

# 5. Entwicklung des Modells für Testeinflussfaktoren

Die größten Probleme bei unserer Arbeit  
sind keine technologischen Probleme,  
sondern soziologische Probleme.

---

(*Tom DeMarco, Timothy Lister [DL99]*)

In diesem Kapitel wird das Modell für Testeinflussfaktoren beschrieben. Mit diesem Modell werden Einflüsse bewertet, die auf die Produktivität in einem Testprojekt wirken. Nach Sneed sind Softwareentwicklungsprojekte immer einmalig und die Produktivität ist in jedem Projekt unterschiedlich [Sne05]. In Form von Einflussfaktoren wird versucht diese projektspezifischen Schwankungen in der Produktivität zu bewerten. Die Bandbreite von Einflussfaktoren ist nahezu unbegrenzt. Von DeMarco werden u. a. Einflüsse von der Büroumgebung, der Leistungsfähigkeit der Mitarbeiter und der Zusammenarbeit im Team genannt [DL99]. Jede Methode zur Aufwandsschätzung in Softwareprojekten berücksichtigt unterschiedliche Einflussfaktoren (vgl. Abschnitt 2.4.3). Für Methoden im Bereich von Testprojekten gilt das Gleiche, wie die Literatur um den de facto Standard des ISTQB zeigt [Spi11, S. 86][Int12c, S. 37].

Die Grundlage für die Bewertung der Produktivität in einem Testprojekt wurde in Kapitel 4 beschrieben. Mit der Kenntnis über die Testkomplexität und die geleisteten Aufwände aus abgeschlossenen Testprojekten sind eine durchschnittliche globale Produktivität und projektspezifische lokale Werte für die Produktivität berechenbar. Die Werte für die globale und lokale Produktivität weichen wegen der genannten Einmaligkeit der Testprojekte voneinander ab. Um die Abweichung zu erklären, müssen Einflussfaktoren identifiziert und deren Auswirkung auf die Produktivität ermittelt werden. Diese Einflussfaktoren werden im Folgenden als Testeinflussfaktoren bezeichnet. Auf Basis der Testein-

## 5. Entwicklung des Modells für Testeinflussfaktoren

flussfaktoren ist über das Modell für Testeinflussfaktoren bei bekannter globaler Produktivität eine projektspezifische lokale Produktivität berechenbar. Mit der berechneten lokalen Produktivität sind anschließend genauere Aussagen über den zu erwartenden Testaufwand möglich.

Als Vorbereitung für den ersten Schritt wurden Testexperten des Unternehmens CRM IT nach Einflussfaktoren auf den Testaufwand befragt [Thi12]. Dabei fiel auf, dass bei dem Schätzen des Testaufwands oft nur implizit auf Testeinflussfaktoren eingegangen wird. Einen Konsens über eine zu berücksichtigende Menge an Testeinflussfaktoren gab es nicht. Vielmehr wurden sehr unterschiedliche Testeinflussfaktoren genannt, die in den kürzlich vorausgegangenen Projekten stark aufgefallen sind. Daher besteht der erste Schritt aus der Identifikation einer allgemeinen Menge an Testeinflussfaktoren. Dies wird in Abbildung 5.1 durch den Block Datenerhebung dargestellt. Zu der Datenerhebung gehört eine Literaturrecherche über allgemeine Einflussfaktoren in Softwareprojekten und eine Literaturrecherche über Einflussfaktoren auf den Testaufwand. Aus den Ergebnissen der beiden Literaturrecherchen wird anschließend eine allgemeine Menge an Testeinflussfaktoren synthetisiert.

Neben dem fehlenden Konsens gibt es bei dem Unternehmen CRM IT und in der Literatur kein einheitliches Verfahren zur Beschreibung und Bewertung von Testeinflussfaktoren. Damit werden die Testeinflussfaktoren sehr unterschiedlich beurteilt und die Vergleichbarkeit der Schätzungen von unterschiedlichen Personen ist nicht gegeben. Neben der fehlenden Vergleichbarkeit der Schätzungen sinkt auch die Nachvollziehbarkeit, da das Verfahren zur Bewertung der Testeinflussfaktoren von Schätzer zu Schätzer variiert. Daher wird das Modell für Testeinflussfaktoren eingeführt, das die identifizierten Testeinflussfaktoren integriert und Vorgaben zur qualitativen Bewertung enthält.

Die Testeinflussfaktoren sollen über eine Formel zur Vorhersage der lokalen Produktivität genutzt werden. Dazu muss für jeden Testeinflussfaktor die quantitative Auswirkung auf die Produktivität ermittelt werden. Diese quantitative Auswirkung wird im Modell für Testeinflussfaktoren durch einen Prozentwert ausgedrückt. Mit dem Wert wird die Erhöhung oder Minderung der Produktivität in Abhängigkeit von der qualitativen Bewertung bestimmt. Da die globale Produktivität getrennt nach den Kombinationen aus Testaktivitäten und Test-

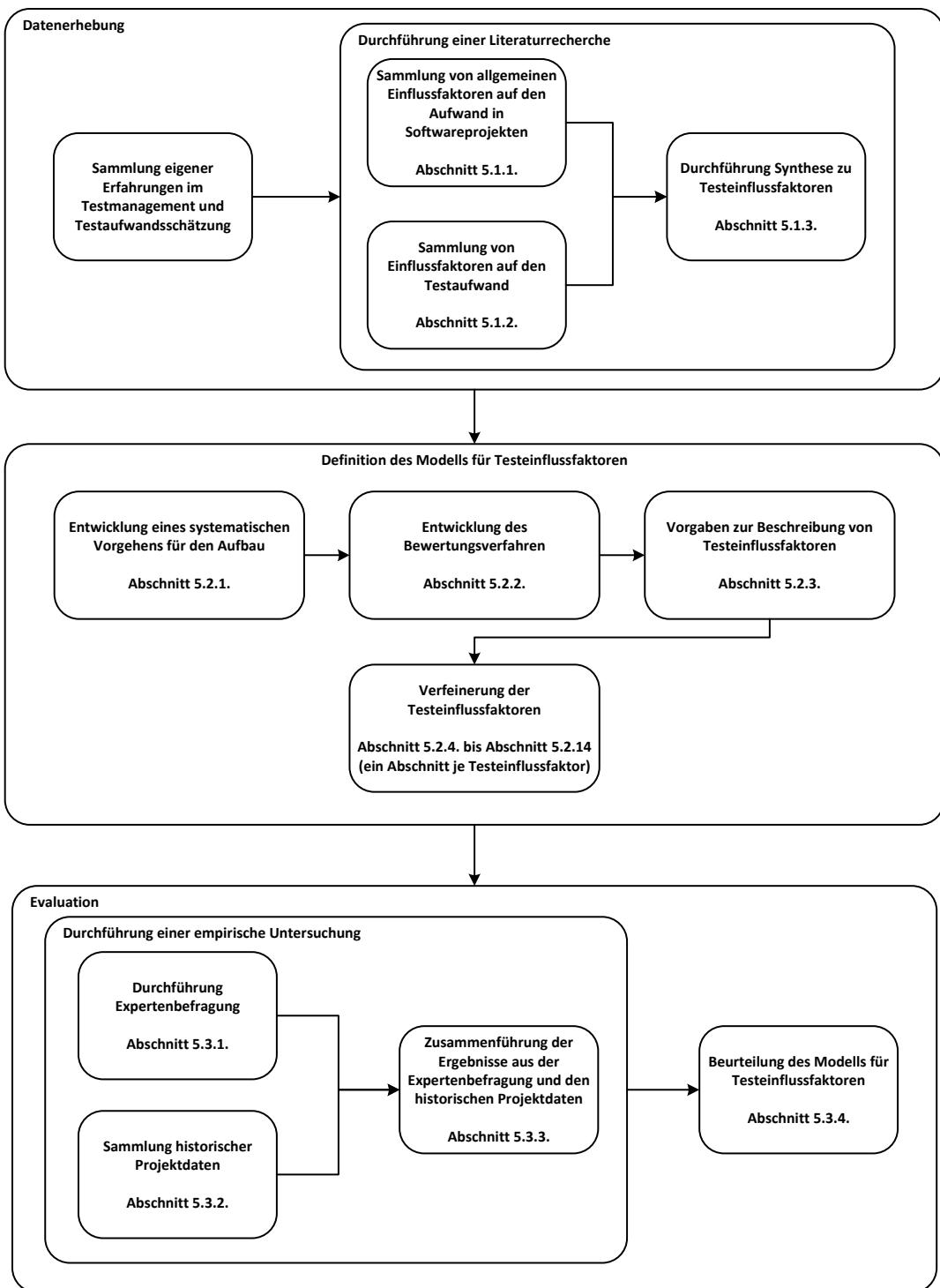


Abbildung 5.1.: Übersicht über den Prozess für die Entwicklung des Modells für Testeinflussfaktoren

## 5. Entwicklung des Modells für Testeinflussfaktoren

stufen bestimmt wird (vgl. Abschnitt 3.2), sind auch die quantitativen Auswirkungen separat für die Kombinationen anzugeben.

Das Ermitteln der quantitativen Auswirkungen wird in Abbildung 5.1 mit dem Block *Durchführung einer empirischen Untersuchung* beschrieben. Die empirische Untersuchung ist in drei Schritte unterteilt. Im ersten Schritt wird eine Expertenbefragung durchgeführt. Damit werden Tendenzen für die quantitativen Auswirkungen ermittelt und Annahmen, die bei der Definition des Modells für Testeinflussfaktoren getroffen worden sind, validiert. Im zweiten Schritt wird die Höhe des Einflusses über Daten aus abgeschlossenen Testprojekten des Unternehmens CRM IT untersucht. Die Ergebnisse aus den ersten beiden Schritten werden anschließend kombiniert und die Höhe der quantitativen Auswirkungen als Prozentwert bestimmt.

Die ermittelten quantitativen Auswirkungen gelten zunächst nur für die Firma CRM IT. Für eine Kalibrierung an einen neuen Kontext werden in Abschnitt 5.4.1 Hinweise gegeben. Danach folgen in Abschnitt 5.4.2 Erklärungen, wie Testeinflussfaktoren entfernt oder hinzugefügt werden können.

### 5.1. Identifikation von Testeinflussfaktoren

Neben dem funktionalen Umfang und der Komplexität bestimmen zahlreiche weitere Einflussfaktoren den Aufwand für Softwareprojekte. Die Einflussfaktoren lassen sich in vielen Fällen direkt oder in veränderter Form auch auf Testprojekte übertragen. Ergänzend gibt es weitere Einflussfaktoren, die speziell für Testprojekte gelten.

Die allgemeinen Einflussfaktoren und diejenigen die direkt für Testprojekte gelten werden in den beiden nachfolgenden Abschnitten zur Übersicht in vier Kategorien eingeordnet: Produkt, Personen, Infrastruktur und Organisation. Die Wahl der Kategorien ist ausgehend von der Kategorisierung der *effort multiplier* in der Methode COCOMO II getroffen worden [Boe00, S. 36].

Die Kategorie Produkt entspricht der Kategorie *product* in der Methode COCOMO II [Boe00, S. 41]. Zu dieser Kategorie gehören Einflussfaktoren, die sich auf das zu entwickelnde Softwareprodukt beziehen. Die Kategorie Personen gleicht

der Kategorie *personnel factors* in der Methode COCOMO II und enthält Einflussfaktoren, die sich auf Charakteristiken der Projektmitglieder und deren Zusammenarbeit beziehen [Boe00, S. 47]. Nach Boehm hat u. a. die Qualifikation der Projektmitglieder nach dem funktionalen Umfang den größten Einfluss auf den Aufwand eines Softwareprojektes [Boe00, S. 47]. Dies trifft auch auf Testprojekte zu, wobei der zu betrachtende Personenkreis von Anforderungsanalytikern, Softwarearchitekten und Programmierern um Personen in den testspezifischen Rollen erweitert werden muss.

Die Kategorie Infrastruktur entspricht der Kategorie *platform factors* in der Methode COCOMO II [Boe00, S. 45]. Zu dieser Kategorie gehören Einflussfaktoren, die sich auf die Zielplattform (Hardware) zum Betrieb der zu entwickelnden Software beziehen. Ergänzend zu der Definition in COCOMO II werden für Testprojekte Einflussfaktoren, die sich auf die Testinfrastruktur<sup>1</sup> beziehen, berücksichtigt. Die Definition der Kategorie Organisation ist analog zu der Kategorie *project factors* in der Methode COCOMO II [Boe00, S. 49]. Zu dieser Kategorie zählen Einflussfaktoren, die sich aus Eigenschaften des Projektes und der beteiligten Organisationen ableiten lassen.

Im anschließenden Abschnitt 5.1.1 wird eine Literaturrecherche über Einflussfaktoren in Softwareprojekten durchgeführt. Dann werden in Abschnitt 5.1.2 Einflussfaktoren auf den Testaufwand ermittelt. Die Gesamtheit der identifizierten Einflussfaktoren werden danach in Abschnitt 5.1.3 in einer Synthese zu Testeinflussfaktoren zusammengeführt.

### 5.1.1. Einflussfaktoren auf den Aufwand in Softwareprojekten

Die Identifikation von Einflussfaktoren bzgl. des Aufwands in einem Softwareprojekt erfolgte im Rahmen einer Literaturrecherche. Die Durchführung der Literaturrecherche orientierte sich an den Prinzipien von Kitchenham [Kit04, KC07]. Im Folgenden sind das Vorgehen und die Ergebnisse zusammengefasst.

Ausgangspunkt für die Literaturrecherche war die Frage, welche Einflussfaktoren auf den Aufwand in einem Softwareentwicklungsprojekt existieren. Die Suche

---

<sup>1</sup>Testinfrastruktur ist im Glossar des ISTQB definiert als „Die organisatorischen Elemente, die für die Durchführung des Tests benötigt werden, bestehend aus Testumgebung, Testwerkzeugen, Büroräumen, Verfahren usw.“ [GTB10, S. 54].

## 5. Entwicklung des Modells für Testeinflussfaktoren

nach Einflussfaktoren ging von einer umfassenden Studie über Methoden zur Aufwandsschätzung in Softwareentwicklungsprojekten aus [BAC00]. Die dort enthaltenden Methoden wurden als Startpunkt für weitere Recherchen genutzt. Einschlusskriterien für zu untersuchende Methoden sind, dass diese einen hohen Bekanntheitsgrad besitzen (z. B. durch eine Norm), Belege für den Einsatz in der Praxis existieren und das sie auf einem algorithmischen Modell basieren. Ausschlusskriterium ist, dass den Methoden Modelle zugrunde liegen, deren Aufbau und Funktionsweise nicht in der Literatur veröffentlicht worden sind. Von jeder gefundenen Methode wurden die Einflussfaktoren extrahiert und den in Abschnitt 5.1 genannten Kategorien zugeordnet.

Die folgenden Methoden sind berücksichtigt worden:

### **Function Point**

Die Methode Function Point (FP) wurde berücksichtigt, da die Methode von der ISO als Standard ISO/IEC 20926 aufgenommen worden ist. Ein weiteres Indiz für die Bekanntheit der Methode sind die große Projektdatenbank der ISBSG, weitere verwandte Standards der ISO und zahlreiche Veröffentlichungen (vgl. Abschnitt 2.4.3.1).

### **COCOMO II**

Die Methode COCOMO II wurde berücksichtigt, da die durch die Methode berücksichtigten Einflussfaktoren aufwendig empirisch untersucht worden sind und an zahlreichen Fallbeispielen die Einsatzfähigkeit der Methode gezeigt worden ist (vgl. Abschnitt 2.4.3.2).

### **Use Case Point 3.0**

Die Use Case Point 3.0 Methode (UCPM 3.0) ist berücksichtigt worden, da für diese Methode Einflussfaktoren empirisch untersucht worden sind und auch Beispiele für den erfolgreichen Einsatz bekannt sind [Fro08] (vgl. Abschnitt 2.4.3.3).

### **SLIM**

Die Methode *Putnam's Software Life-Cycle Model* (SLIM) wurde berücksichtigt, da für diese Methode aufwendige empirische Studien zur Analyse von Aufwandsverläufen in Softwareprojekten durchgeführt worden sind [PM92, S. 17]. Für den Einsatz von SLIM existieren zahlreiche Werkzeuge, die laut dem Hersteller auf der Erfahrung aus ca. 10.000 Projekten beruhen [QSM].

## SEER-SEM

Die Methode *Software Evaluation and Estimation of Resources - Software Estimating Model* (SEER-SEM) beruht teilweise auf einem Modell, dessen Aufbau und Funktionsweise nicht veröffentlicht ist. Trotzdem wird diese Methode berücksichtigt, da das Buch [GE06] Einblick in die berücksichtigten Einflussfaktoren gibt und diese laut dem Autor mit Hilfe einer großen Projektdatenbank evaluiert worden sind [GE06, S. 442].

Die folgenden Methoden sind nicht berücksichtigt worden:

### **ESTIMACS, PRICE-S, CheckPoint und SELECT Estimator**

Die Methoden ESTIMACS, PRICE-S, CheckPoint und SELECT Estimator wurden nicht berücksichtigt, da deren Aufbau und Funktionsweise in der Literatur nur skizziert wird und keine ausreichenden Informationen über die berücksichtigten Einflussfaktoren gefunden wurden. Weiterhin zeigt die Tabelle 2 in [BAC00, S. 191], dass die Methoden keine über die oben genannten Methoden hinausgehenden Einflussfaktoren/Aktivitäten beinhalten.

Die extrahierten Einflussfaktoren sind in den Tabellen 5.1 bis 5.4 getrennt nach ihrer zugeordneten Kategorie aufgeführt. Ähnliche Einflussfaktoren wurden dabei gruppiert und ggf. eine neue Bezeichnung eingeführt (vgl. Spalte Einflussfaktor). Für jeden Einflussfaktor ist aufgeführt, welche Quelle(n) die Grundlage ist/sind und wie die Originalbezeichnung lautet. Zur Verfolgung des Einflussfaktors in der nachfolgenden Synthese zu Testeinflussfaktoren ist jedem Einflussfaktor (EF) ein Kürzel  $EF_{X_i}$  zugeordnet.

In Tabelle 5.1 sind die extrahierten Einflussfaktoren der Kategorie Produkt zu sehen. Die ersten drei Einflussfaktoren beziehen sich auf die Spezifikation der Anforderungen des Produktes. Diese Einflussfaktoren wurde der Kategorie Produkt zugeordnet, da diese die direkte Grundlage für die Entwicklung des Produktes sind. Die Qualität und die Stabilität der Anforderungen hat einen direkten Einfluss auf den Aufwand eines Projektes, da die Zeit, um ein gemeinsames Verständnis der Anforderungen aufzubauen, bei steigender Qualität und Stabilität minimiert wird.

Die Einflussfaktoren  $EF_{PR_4}$  bis  $EF_{PR_8}$  orientieren sich an den Merkmalen, die in der Norm ISO 9126 [ISO01] bzw. dem Nachfolger ISO 25010 [ISO11b] aufgeführt sind (vgl. Abbildung 2.3). Zusätzlich ist explizit der Einflussfaktor Sicherheit

## 5. Entwicklung des Modells für Testeinflussfaktoren

$EF_{PR_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EF_{PR_1}$	Qualität der fachl. Spezifikation	UCPM 3.0 [Fro09, S. 126]	Qualität der Grobspezifikation
$EF_{PR_2}$	Qualität der techn. Spezifikation	UCPM 3.0 [Fro09, S. 126]	Qualität der T-Architektur
$EF_{PR_3}$	Stabilität der Anforderungen	UCPM 3.0 [Fro09, S. 124] SEER-SEM [GE06, S. 463]	Stabile Anforderungen <i>Requirements volatility (change)</i>
$EF_{PR_4}$	Zuverlässigkeit	UCPM 3.0 [Fro09, S. 125ff]	Ausfallsicherheit Verfügbarkeit
$EF_{PR_5}$	Benutzbarkeit	COCOMO II [Boe00, S. 41] SEER-SEM [GE06, S. 465]	<i>Required Software Reliability</i> <i>Test Level</i>
$EF_{PR_6}$	Effizienz	UCPM 3.0 [Fro09, S. 132] FP [BD08, S. 473][Int10] SEER-SEM [GE06, S. 470]	Benutzerfreundlichkeit <i>End-User Efficiency</i> <i>Special display requirements</i>
$EF_{PR_7}$	Übertragbarkeit	UCPM 3.0 [Fro09, S. 132] SEER-SEM [GE06, S. 466ff]	Performanz und Lastanforderungen Performance <i>Time Constraints</i> <i>Real-time code</i>
$EF_{PR_8}$	Sicherheit	UCPM 3.0 [Fro09, S. 132] SEER-SEM [GE06, S. 472]	Portabilität Benutzerfreundlichkeit <i>Target System Complexity</i> <i>Rehost from development to target</i> <i>Target System Volatility</i>
$EF_{PR_9}$	Wiederverwendbarkeit	UCPM 3.0 [Fro09, S. 132] COCOMO II [Boe00, S. 42] SEER-SEM [GE06, S. 467f]	Sicherheitsanforderungen <i>Security requirements</i> Wiederverwendbarkeit <i>Development for Reusability</i> <i>Reusability level required</i> <i>Software impacted by reuse</i>

Tabelle 5.1.: Einflussfaktoren der Kategorie Produkt auf den Aufwand eines Softwareentwicklungsprojektes

aufgeführt, da Anforderungen an die Sicherheit zu höheren Aufwänden bei der Entwicklung des Softwareproduktes führen. In den vorher genannten Normen wären diese Einflussfaktoren dem Merkmal Funktionalität zuzuordnen.

Der Einflussfaktor  $EF_{PR_9}$  beschreibt, wie hoch die Anforderungen an die Wiederverwendbarkeit des Codes oder von Programmkomponenten sind. Bei höheren Anforderungen ist ein höherer Aufwand für die Planung und die Entwicklung zu erwarten.

In Tabelle 5.2 sind die extrahierten Einflussfaktoren der Kategorie Personen zu sehen. Die Einflussfaktoren  $EF_{P_1}$  bis  $EF_{P_7}$  beziehen sich auf die Erfahrung und Qualifikation der am Projekt beteiligten Personen. Dabei wird bei der Qualifikation zwischen den drei Rollen Softwarearchitekt, Analytiker und Entwickler

## 5.1. Identifikation von Testeinflussfaktoren

$EF_{P_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EF_{P_1}$	Erfahrung mit dem Softwaretyp	COCOMO II [Boe00, S. 33 u. 48 ] SLIM [PM92, S. 261] SEER-SEM [GE06, S. 455]	<i>Application Experience Precedentedness</i> <i>Size and Application Analyst's application experience</i>
	Erfahrung mit der Plattform	COCOMO II [Boe00, S. 48] SEER-SEM [GE06, S. 457]	<i>Platform Experience</i> <i>Target system experience</i>
$EF_{P_3}$	Qualifikation der Architekten	UCPM 3.0 [Fro09, S. 126]	Fähigkeit technische Chefdesigner
	Qualifikation der Analytiker	UCPM 3.0 [Fro09, S. 126] COCOMO II [Boe00, S. 47] SEER-SEM [GE06, S. 455]	Fähigkeit fachliche Chefdesigner <i>Analyst Capabilities</i> <i>Analyst Capabilities</i>
$EF_{P_5}$	Qualifikation der Entwickler	COCOMO II [Boe00, S. 47] SEER-SEM [GE06, S. 455f]	<i>Programmer Capability</i> <i>Programmer Capabilities</i> <i>Programmer Language Experience</i> <i>Development System Experience</i>
	Sprach-/Werkzeugerfahrung	COCOMO II [Boe00, S. 48] SLIM [PM92, S. 261] SEER-SEM [GE06, S. 458]	<i>Language and Tool Experience</i> <i>Development Computer Practices and Methods Experience</i>
$EF_{P_7}$	Reifegrad des Projektes	UCPM 3.0 [Fro09, S. 125]	Reifegrad des Projektes
$EF_{P_8}$	Anzahl Ansprechpartner	UCPM 3.0 [Fro09, S. 126]	Anzahl Ansprechpartner
$EF_{P_9}$	Zusammenarbeit	UCPM 3.0 [Fro09, S. 127] COCOMO II [Boe00, S. 34]	Zusammenarbeit <i>Team Cohesion</i>
$EF_{P_{10}}$	Kontinuität Mitarbeiter	UCPM 3.0 [Fro09, S. 127] COCOMO II [Boe00, S. 48]	Kontinuität Mitarbeiter <i>Personnel Continuity</i>
$EF_{P_{11}}$	Anzahl Standorte	UCPM 3.0 [Fro09, S. 127] COCOMO II [Boe00, S. 49] SEER-SEM [GE06, S. 460]	Verteilte Entwicklung <i>Multisite Development</i> <i>Multiple Site Development</i>

Tabelle 5.2.: Einflussfaktoren der Kategorie Personen auf den Aufwand eines Softwareentwicklungsprojektes

## 5. Entwicklung des Modells für Testeinflussfaktoren

$EF_{I_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EF_{I_1}$	Rechenzeitverbrauch	COCOMO II [Boe00, S. 45f]	<i>Execution Time Constraint</i>
$EF_{I_2}$	Speicherverbrauch	COCOMO II [Boe00, S. 46] SLIM [PM92, S. 259]	<i>Main Storage Constraint Memory</i>
$EF_{I_3}$	Verfügbarkeit Ressourcen	SEER-SEM [GE06, S. 461]	<i>Resource dedication Resource and Support Location</i>
		SLIM [PM92, S. 258]	<i>Computer Availability</i>
$EF_{I_4}$	Qualität Infrastruktur	SEER-SEM [GE06, S. 459f]	<i>Turnaround time Response time</i>
$EF_{I_5}$	Werkzeugunterstützung	COCOMO II [Boe00, S. 49] SEER-SEM [GE06, S. 459]	<i>Use of Software Tools Automated tools use</i>

Tabelle 5.3.: Einflussfaktoren der Kategorie Infrastruktur auf den Aufwand eines Softwareentwicklungsprojektes

$EF_{O_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EF_{O_1}$	Risikomanagement	COCOMO II [Boe00, S. 34] SEER-SEM [GE06, S. 466]	<i>Architecture and Risk Resolution Quality Assurance (QA) Level</i>
$EF_{O_2}$	Prozessreife	UCPM 3.0 [Fro09, S. 127] COCOMO II [Boe00, S. 34f]	<i>Prozess-Overhead Process Maturity</i>
		SEER-SEM [GE06, S. 458ff]	<i>Modern Development Practices Use Process Volatility</i>
$EF_{O_3}$	Dokumentationsbedarf	COCOMO II [Boe00, S. 42] SEER-SEM [GE06, S. 463]	<i>Documentation Match to Life-Cycle Needs Specification level-reliability</i>
$EF_{O_4}$	Termindruck	UCPM 3.0 [Fro09, S. 125] COCOMO II [Boe00, S. 50]	<i>Termindruck Required Development Schedule</i>

Tabelle 5.4.: Einflussfaktoren der Kategorie Organisation auf den Aufwand eines Softwareentwicklungsprojektes

unterschieden (vgl. Abschnitt 2.2). Dagegen bezieht sich die Erfahrung auf die Gesamtheit der beteiligten Personen im Softwareprojekt.

Der Einflussfaktor  $EF_{P_8}$  bezieht sich auf die Anzahl der Ansprechpartner. Dabei gehört nicht nur das Personal, das direkt an dem Projekt beteiligt ist, zu den Ansprechpartnern. Stattdessen werden alle technischen und fachlichen Ansprechpartner gezählt. Bei steigender Anzahl erhöht sich u. a. der Kommunikationsaufwand und damit der Gesamtaufwand.

Die drei Einflussfaktoren  $EF_{P_9}$  bis  $EF_{P_{11}}$  beziehen sich auf die Erfahrung und die Möglichkeiten zur Zusammenarbeit der Projektmitglieder. Dabei gibt es den allgemeinen Einflussfaktor Zusammenarbeit, der sich auf die Erfahrung in der Zusammenarbeit im Projekt bezieht. Der Einflussfaktor  $EF_{P_{10}}$  beschreibt dagegen die Möglichkeit von der steigenden Erfahrung des Personals in der Zu-

## 5.1. Identifikation von Testeinflussfaktoren

sammenarbeit in einem Projekt zu profitieren.  $EF_{P_1}$  charakterisiert zusätzlich, welche Auswirkungen eine verteilte Entwicklung auf den Aufwand eines Softwareentwicklungsprojektes hat.

Die extrahierten Einflussfaktoren der Kategorie Infrastruktur sind in Tabelle 5.3 zu sehen. Die ersten beiden Einflussfaktoren beschreiben Beschränkungen bzgl. der Rechenzeit- und des Speicherverbrauchs. Die Annahme ist, dass bei hoher Ausnutzung der verfügbaren Ressourcen der Aufwand für die Entwicklung des Softwareproduktes steigt.

Die drei Einflussfaktoren  $EF_{I_3}$  bis  $EF_{I_5}$  beziehen sich auf die Infrastruktur, die dem Personal für die Entwicklung des Softwareproduktes zur Verfügung steht. Dabei wird mit  $EF_{I_3}$  beschrieben, wie exklusiv das Personal die notwendigen Ressourcen nutzen kann. Mit  $EF_{I_4}$  werden Qualitätseigenschaften der Infrastruktur charakterisiert. Dazu gehören Merkmale, die das Warten bzw. die Effizienz der Infrastruktur beschreiben. Der Einflussfaktor  $EF_{I_5}$  bezieht sich auf die Nutzung von Werkzeugunterstützung mit dem Fokus auf die Automatisierung.

Die extrahierten Einflussfaktoren der Kategorie Organisation sind in Tabelle 5.4 zu sehen. Die ersten beiden Einflussfaktoren beziehen sich allgemein auf die Projektorganisation. Dabei charakterisiert  $EF_{O_1}$  zu welchem Maß durch aktives Risikomanagement Aufwände für die Entwicklung des Softwareproduktes minimiert werden können. Dies kann z. B. die Einführung einer Qualitätssicherung sein, die in frühen Projektphasen durch die statische Überprüfung von Spezifikationen Fehler in späteren Phasen vorbeugt.  $EF_{O_2}$  beschreibt die Stabilität und Reife der eingesetzten Prozesse zur Entwicklung des Softwareproduktes (vgl. Abschnitt 2.1).

Der Einflussfaktor  $EF_{O_3}$  bezieht sich auf den *Overhead* durch den Dokumentationsbedarf im Softwareentwicklungsprojekt. Die Dokumentation in einem Projekt kann z. B. nach festgelegten Vorlagen erfolgen, deren Nutzung Mehraufwand für das Projekt bedeutet. Da dieser Aufwand analog zu COCOMO II signifikant sein kann, wird dies als separater Einflussfaktor aufgeführt [Boe00, S. 42].

$EF_{O_4}$  bezieht sich auf den zeitlichen Planung des Projektes. Ein hoher Termindruck kann in einem Projekt Mehraufwand verursachen, da die Planung komprimiert wird. Mit der Komprimierung wird versucht mehr Aufwand in weniger

## 5. Entwicklung des Modells für Testeinflussfaktoren

Zeit zu leisten. Dies verursacht einen höheren Aufwand in der Planung und erhöht den Abstimmungsbedarf zwischen den Projektmitgliedern.

### 5.1.2. Einflussfaktoren auf den Testaufwand

Bei der Identifikation von Einflussfaktoren bzgl. des Testaufwands (EFT) wurde genauso wie in Abschnitt 5.1.1 vorgegangen. Ausgangspunkt für die Literaturrecherche war die Frage, welche Einflussfaktoren auf den Aufwand in einem Testprojekt existieren.

Die Suche nach Einflussfaktoren ging von einer Expertenumfrage [Thi12] und der Standardliteratur zum de facto Standard des ISTQB aus [SL10, Spi11]. Die dabei ermittelten Methoden wurden als Ausgangspunkt für weitere Recherchen über Einflussfaktoren auf den Testaufwand genutzt. Einschlusskriterien für die gefundenen Methoden sind auch hier, dass diese einen hohen Bekanntheitsgrad besitzen (z. B. durch eine Norm), Belege für den Einsatz in der Praxis existieren und das sie auf einem algorithmischen Modell basieren. Das Ausschlusskriterium proprietäres Modell konnte verworfen werden, da keine der Methoden auf einem proprietären Modell beruht. Von jeder gefundenen Methode wurden die Einflussfaktoren extrahiert und den in Abschnitt 5.1 genannten Kategorien zugeordnet.

Die folgenden Methoden sind berücksichtigt worden:

#### Testpunktanalyse

Die Methode Testpunktanalyse (TPA) wurde berücksichtigt, da sie in der Literatur zur Vorbereitung für die Zertifizierung durch das ISTQB erwähnt wird [Spi11, S. 93][GTB10, S. 57]. Davon ausgehend kann von einer hohen Bekanntheit der Testpunktanalyse ausgegangen werden. Die Methode wurde in dieser Dissertation in Abschnitt 3.1.2 bereits diskutiert.

#### Testaufwandsschätzung in der Softwareentwicklung

Die Studie über Einflussfaktoren auf den Testaufwand in der Dissertation von Dowie wird als Methode *Testaufwandsschätzung in der Softwareentwicklung* (TAS) berücksichtigt, da viele Einflussfaktoren in zwei Organisationen empirisch untersucht worden sind [Dow09]. Die Dissertation stellt keine vollständige Methode zur Testaufwandsschätzung dar, sondern eine Methode zur Selektion von Einflussfaktoren abhängig vom Unternehmens-

und Projektkontext. Durch die empirische Untersuchung ist die Einbeziehung der bestätigten Einflussfaktoren ein wertvoller Beitrag.

### **Standardliteratur zum ISTQB**

Die Standardliteratur zur Vorbereitung für eine Zertifizierung durch das ISTQB (StdLitISTQB) wird berücksichtigt, da dort Einflussfaktoren aufgeführt werden, die im Rahmen einer Schätzung des Testaufwands berücksichtigt werden sollten [Spi11, SL10, Int11, Int12c]. Obwohl diese eher Ausgangspunkte für eine Entwicklung einer Methode zur Testaufwandschätzung sind, werden diese Einflussfaktoren mit berücksichtigt.

### ***Test Process Improvement***

Die Methode *Test Process Improvement* (TPI) wird berücksichtigt, da sie in der Literatur zur Vorbereitung für die Zertifizierung durch das ISTQB erwähnt wird [Spi11, S. 177][GTB10, S. 60]. Zwar ist dies keine Methode zur direkten Testaufwandsschätzung, aber aus der Beschreibung der Methode TPI können Einflussfaktoren auf den Testaufwand abgeleitet werden.

### **Nageswaran Use Case Points**

Die Methode Nageswaran Use Case Points (NUCP) wurde berücksichtigt, obwohl sie keinen hohen Bekanntheitsgrad besitzt und auch keine Belege für den Einsatz in der Praxis. Sie wurde gewählt, da sie die Methode Use Case Points (vgl. Abschnitt 2.4.3.3) auf den Testbereich überträgt.

Die extrahierten Einflussfaktoren sind in den Tabellen 5.5 bis 5.8 getrennt nach ihrer zugeordneten Kategorie aufgeführt. Ähnliche Einflussfaktoren wurden dabei gruppiert und ggf. eine neue Bezeichnung eingeführt (vgl. Spalte Einflussfaktor). Für jeden Einflussfaktor ist aufgeführt, welche Quelle die Grundlage ist und wie die Originalbezeichnung lautet. Zur Verfolgung des Einflussfaktors in der nachfolgenden Synthese zu Testeinflussfaktoren ist jedem Einflussfaktor ein Kürzel  $EFT_{X_i}$  zugeordnet.

In Tabelle 5.5 sind die extrahierten Einflussfaktoren der Kategorie Produkt aufgeführt. Die ersten zwei Einflussfaktoren sind auf die Qualität und die Stabilität der Anforderungsspezifikation bezogen. Der Einflussfaktor  $EFT_{PR_3}$  betrifft auch die Qualität der Spezifikation, aber mit dem Fokus auf die Testbarkeit. Die dahinter stehende Frage ist, wie weit die Spezifikation den Test unterstützt.

## 5. Entwicklung des Modells für Testeinflussfaktoren

$EFT_{PR_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EFT_{PR_1}$	Qualität der Spezifikationen	StdLitISTQB [Spi11, S. 87]	Qualität der Dokumentation
		TPA [vBKV08, S. 471]	Testbasis
		TAS [Dow09, S. 177]	Qualität der Spezifikation und der Entwurfsdokumentation
$EFT_{PR_2}$	Stabilität der Anforderungen	TAS [Dow09, S. 167]	Anzahl und Zeitpunkt der Anforderungsänderungen
		StdLitISTQB [SL10, S. 186]	Rate der Softwareänderungen
$EFT_{PR_3}$	Testbarkeit	StdLitISTQB [SL10, S. 187]	Testbarkeit
$EFT_{PR_4}$	Benutzbarkeit	TPA [vBKV08, S. 466f]	Effektivität und Angemessenheit
$EFT_{PR_5}$	Übertragbarkeit	TPA [vBKV08, S. 466f]	Übertragbarkeit
$EFT_{PR_6}$	Funktionalität	TPA [vBKV08, S. 466f]	Funktionalität
$EFT_{PR_7}$	Effizienz	TPA [vBKV08, S. 466f] NUCP [Nag01]	Leistungsfähigkeit Performance objectives
$EFT_{PR_8}$	Sicherheit	TPA [vBKV08] NUCP [Nag01]	Sicherheit <i>Security Features</i>
$EFT_{PR_9}$	Teststrategie	StdLitISTQB [SL10, S. 186]	Teststrategie
		TPA [vBKV08, S. 458f]	Teststrategie
$EFT_{PR_{10}}$	Bedeutung des Softwareproduktes	[Dow09, S. 156]	Strategische Bedeutung des Softwareproduktes
$EFT_{PR_{11}}$	Wiederverwendbarkeit	StdLitISTQB [Spi11, S. 87]	Wiederverwendbarkeit
		TPA [vBKV08, S. 472]	Testware
		TAS [Dow09, S. 199]	Wiederverwendbare Testdaten
		NUCP [Nag01]	<i>Test-ware reuse</i>

Tabelle 5.5.: Einflussfaktoren der Kategorie Produkt auf den Aufwand eines Testprojektes

Die Einflussfaktoren  $EFT_{PR_4}$  bis  $EFT_{PR_8}$  orientieren sich an den Merkmalen, die in der Norm ISO 9126 [ISO01] bzw. dem Nachfolger ISO 25010 [ISO11b] aufgeführt sind. Sie sind analog zu den Einflussfaktoren, die allgemein in Softwareprojekten gelten (vgl. Tabelle 5.1). Genauso wurde Sicherheit als gesondertes Merkmal aufgeführt, da Anforderungen an die Sicherheit den Testaufwand erhöhen.

Der Einflussfaktor  $EFT_{PR_9}$  beschreibt Charakteristiken der gewählten Teststrategie. Zur Teststrategie gehört das Festlegen der Testmethoden und der Testziele. Wenn als Testmethoden die Äquivalenzklassenbildung und die Grenzwertanalyse verwendet werden sollen, dann ist der Testaufwand höher als beim Test mit nur der Äquivalenzklassenbildung.

Der Einflussfaktor  $EFT_{PR_{10}}$  ist auf die strategische Bedeutung des Softwareproduktes bezogen. Die Annahme ist, dass mehr Testaufwand investiert wird, wenn das Produkt eine hohe strategische Bedeutung besitzt. Letztendlich ist dieser Einflussfaktor auch implizit in den Einflussfaktoren  $EFT_{PR_4}$  bis  $EFT_{PR_8}$

## 5.1. Identifikation von Testeinflussfaktoren

$EFT_{P_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EFT_{P_1}$	Erfahrung mit dem Softwaretyp	TAS [Dow09, S. 160] StdLitISTQB [Spi11, S. 87]	Projekttyp Ähnlichkeit zu Vorgängerprojekten
$EFT_{P_2}$	Erfahrung in der Plattform	TAS [Dow09, S. 165]	Erfahrung mit der Technologie
$EFT_{P_3}$	Erfahrung im firmeninternen Testprozess	TAS [Dow09, S. 182]	Erfahrung der Tester mit ihren Aufgaben
$EFT_{P_4}$	Erfahrung mit der zu testenden Software	TAS [Dow09, S. 182]	Erfahrung der Tester mit dem Produkt
$EFT_{P_5}$	Qualifikation des Testpersonals	StdLitISTQB [Spi11, S. 87] StdLitISTQB [SL10, S. 186] TPI [van11, S. 108]	Erfahrung und Ausbildung des Testpersonals Mitarbeiterqualifikationen Professionalität der Tester
$EFT_{P_6}$	Anzahl Ansprechpartner	TAS [Dow09, S. 197]	Anzahl Kunden, die am Testen teilnehmen
$EFT_{P_7}$	Teamgröße	TPA [vBK08, S. 473]	Teamgröße
$EFT_{P_8}$	Engagement der Stakeholder	TPI Next [van11, S. 41] TAS [Dow09, S. 185]	Engagement der Stakeholder Einstellung des Projektleiters zu Qualität
$EFT_{P_9}$	Anzahl Standorte	TAS [Dow09, S. 191]	Anzahl der Standorte der Projektmitarbeiter
$EFT_{P_{10}}$	Kontinuität Mitarbeiter	TAS [Dow09, S. 194]	Verfügbarkeit der Teammitglieder
$EFT_{P_{11}}$	Anzahl verschiedener Muttersprachen im Team	TAS [Dow09, S. 190]	Anzahl verschiedener Muttersprachen im Team

Tabelle 5.6.: Einflussfaktoren der Kategorie Personal auf den Aufwand eines Testprojektes

enthalten. Trotzdem wird er mit aufgeführt, da er wahrscheinlich leichter einzuschätzen ist, als die einzelnen Qualitätsmerkmale.

Der letzte Einflussfaktor in Tabelle 5.5 bezieht sich auf die Wiederverwendbarkeit von Testmitteln aus vorherigen Testphasen oder Projekten. Dabei gehören zu den Testmitteln „Alle Artefakte, die während des Testprozesses erstellt werden und die erforderlich sind, um die Tests zu planen, zu entwerfen oder auszuführen.“ [GTB10, S. 56]. Mit einer möglichen Wiederverwendung wird der Aufwand in dem aktuellen Testprojekt geringer.

In Tabelle 5.5 sind die extrahierten Einflussfaktoren der Kategorie Personen aufgeführt. Die Einflussfaktoren  $EF_{P_1}$  bis  $EF_{P_5}$  beziehen sich auf die Erfahrung und Qualifikation der am Projekt beteiligten Personen. Dabei wurden möglichst gleichen Bezeichnungen für die Einflussfaktoren gewählt, wie in Tabelle 5.2. Ergänzend ist die Erfahrung des beteiligten Personals mit dem Testprozess und der zu testenden Software aufgenommen worden. Wenn die Tester Erfahrung in

## 5. Entwicklung des Modells für Testeinflussfaktoren

$EFT_{I_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EFT_{I_1}$	Entwicklungsumgebung	TPA [vBKV08, S. 471], NUCP [Nag01]	Entwicklungsumgebung <i>Development Environment</i>
$EFT_{I_2}$	Stabilität u. Verfügbarkeit der Testumgebung	TPA [vBKV08, S. 471f] TAS [Dow09, S. 200] StdLitISTQB [SL10, S. 186] TPI [van11, S. 124] StdLitISTQB [Spi11, S. 87]	Vorheriger Test Testumgebung Verfügbarkeit und Stabilität des Testsystems Verfügbarkeit von Testumgebung Abstimmung der Testumgebung Testreife
$EFT_{I_3}$	Verfügbarkeit Ressourcen	StdLitISTQB [SL10, S. 186]	Verfügbarkeit von Testwerkzeugen
$EFT_{I_4}$	Managementwerkzeuge	TPA [vBKV08, S. 474]	Managementwerkzeuge
$EFT_{I_5}$	Testmittelmanagement	TPI [van11, S. 108]	Testwaremanagement
$EFT_{I_6}$	Werkzeugunterstützung	TPA [vBKV08, S. 470], NUCP [Nag01] TPI [van11, S. 119]	Testwerkzeuge <i>Test Tools</i> Testwerkzeuge

Tabelle 5.7.: Einflussfaktoren der Kategorie Infrastruktur auf den Aufwand eines Testprojektes

$EFT_{O_i}$	Einflussfaktor	Quelle	Originalbezeichnung
$EFT_{O_1}$	Permanente Testorganisation	TPA [vBKV08, S. 474]	Permanente Testorganisation
$EFT_{O_2}$	Prozessreife	StdLitISTQB [Spi11, S. 87] StdLitISTQB [SL10, S. 186] TPI [van11, S. 47 u. 103]	Reife des Testprozesses Reife des Entwicklungsprozesses Grad der Beteiligung Methodisches Vorgehen
$EFT_{O_3}$	Termintreue von Testobjektlieferung	StdLitISTQB [Spi11, S. 87]	Termintreue von Testobjektlieferung
$EFT_{O_4}$	Dokumentationsbedarf	StdLitISTQB [Spi11, S. 87]	Geforderte Qualität und Exaktheit der zu erstellenden Dokumente
$EFT_{O_5}$	Termindruck	TAS [Dow09, S. 205] StdLitISTQB [SL10, S. 186]	Zu Projektbeginn verfügbare Zeit Zeitdruck

Tabelle 5.8.: Einflussfaktoren der Kategorie Organisation auf den Aufwand eines Testprojektes

dem Vorgehen und mit der zu testenden Software besitzen, ist der Testaufwand geringer.

Die Einflussfaktoren  $EFT_{P_6}$  und  $EFT_{P_7}$  beschreiben den Kommunikationsaufwand, der im Testprojekt zu erwarten ist. Die Definition von Dowie für den Einflussfaktor  $EFT_{P_6}$  war auf die Anzahl der Kunden beschränkt, die am Test teilnehmen [Dow09, S. 197]. Dies wird erweitert zu der Anzahl von Ansprechpartnern, die z. B. Unterstützungsleistungen für die Erstellung von Testfällen oder die Testausführung leisten müssen. Dies schließt auch die Unterstützungsleistungen durch Entwickler mit ein. Dagegen bezieht sich der Einflussfaktor

$EFT_{P_7}$  auf die Größe des Testteams. Dazu gehört nur das Personal in den Rollen Testmanager, Testdesigner oder Tester.

Die Einflussfaktoren  $EFT_{P_8}$  bis  $EFT_{P_{11}}$  charakterisieren die Zusammenarbeit der Projektmitglieder. Dazu gehört auch das Engagement der *Stakeholder*, da deren Einstellung zu dem Testprojekt die Zusammenarbeit stark beeinflusst [van11, S. 41][Dow09, S. 185].

In Tabelle 5.7 sind die extrahierten Einflussfaktoren der Kategorie Infrastruktur aufgeführt. Die ersten beiden Einflussfaktoren beziehen sich auf die Stabilität der Testumgebung. Dabei werden mit dem Einflussfaktor  $EFT_{I_1}$  die Charakteristiken der Entwicklungsumgebung beschrieben, die Einfluss auf die Stabilität der Testumgebung haben. Dies können z. B. Mechanismen zur Fehlervermeidung bei der Kompilierung der Software sein.

Die Einflussfaktoren  $EFT_{I_3}$  bis  $EFT_{I_6}$  beziehen sich auf die Verwendung von Werkzeugunterstützung und deren Verfügbarkeit für das Testprojekt. Dabei wird mit dem Einflussfaktor  $EFT_{I_4}$  auf Werkzeuge zur (automatisierten) Unterstützung der Testmanagementaktivitäten eingegangen. Zusätzliche dazu wird mit dem Einflussfaktor  $EFT_{I_6}$  die (automatisierte) Werkzeugunterstützung für die Testaktivitäten Testdesign und Testausführung beschrieben.

In Tabelle 5.8 sind die extrahierten Einflussfaktoren der Kategorie Organisation aufgeführt. Dabei beziehen sich die ersten drei Einflussfaktoren auf die gesamte Organisation des Projektes. Ergänzend zu dem allgemeinen Einflussfaktor  $EFT_{O_2}$  für die Charakterisierung der Prozessreife beschreibt der Einflussfaktor  $EFT_{O_1}$ , ob die Testorganisation eine dauerhafte Einrichtung ist (z. B. eine eigenständige und dauerhafte Abteilung in einem Unternehmen).

Der Einflussfaktor  $EFT_{O_4}$  beschreibt den Dokumentationsbedarf in dem Testprojekt. Dies ist u. a. bei der Testaktivität Testausführung von hoher Bedeutung, da der Aufwand für die Ausführung mit von den Anforderungen an die Dokumentation der Ausführung abhängt.

Der Einflussfaktor  $EFT_{O_5}$  bezieht sich auf die Zeitplanung und Termine in dem Projekt und beschreibt wie hoch der Termindruck in dem Projekt ist.

## 5. Entwicklung des Modells für Testeinflussfaktoren

### 5.1.3. Synthese zu Testeinflussfaktoren

Im diesem Abschnitt wird die Synthese der extrahierten Einflussfaktoren aus den Abschnitten 5.1.1 und 5.1.2 beschrieben. Ziel für die Synthese ist die Bestimmung einer Menge an Testeinflussfaktoren, die durch Gruppen von extrahierten Einflussfaktoren beschrieben sind. Diese Menge ist anschließend eine Eingabe für die Beschreibung des Modells für Testeinflussfaktoren in Abschnitt 5.2.

	$TF_i$	Testeinflussfaktor	Allgemeine Einflussfaktoren	Einflussfaktoren auf den Testaufwand
Produkt	$TF_1$	Qualität der Testbasis	$EF_{PR_1}, EF_{PR_2}, EF_{PR_3}$	$EFT_{PR_1}, EFT_{PR_2}, EFT_{PR_3}$
	$TF_2$	Qualitätsziele	$EF_{PR_4}, EF_{PR_5}, EF_{PR_6}, EF_{PR_7}, EF_{PR_8}$	$EFT_{PR_4}, EFT_{PR_5}, EFT_{PR_6}, EFT_{PR_7}, EFT_{PR_8}, EFT_{PR_9}, EFT_{PR_{10}}$
	$TF_3$	Wiederverwendbarkeit von Testmitteln	$EF_{PR_9}$	$EFT_{PR_{11}}$
Personal	$TF_4$	Qualifikation und Erfahrung	$EF_{P_1}, EF_{P_2}, EF_{P_3}, EF_{P_4}, EF_{P_5}, EF_{P_6}, EF_{P_7}$	$EFT_{P_1}, EFT_{P_2}, EFT_{P_3}, EFT_{P_4}, EFT_{P_5}$
	$TF_5$	Kommunikationsaufwand	$EF_{P_8}$	$EFT_{P_6}, EFT_{P_7}$
	$TF_6$	Zusammenarbeit der Projektmitglieder	$EF_{P_9}, EF_{P_{10}}, EF_{P_{11}}$	$EFT_{P_8}, EFT_{P_9}, EFT_{P_{10}}, EFT_{P_{11}}$
Infrastruktur	$TF_7$	Stabilität und Verfügbarkeit der Testumgebung	$EF_{I_1}, EF_{I_2}$	$EFT_{I_1}, EFT_{I_2}$
	$TF_8$	Testinfrastruktur	$EF_{I_3}, EF_{I_4}, EF_{I_5}$	$EFT_{I_3}, EFT_{I_4}, EFT_{I_5}, EFT_{I_6}$
Organisation	$TF_9$	Gesamttestorganisation	$EF_{O_1}, EF_{O_2}$	$EFT_{O_1}, EFT_{O_2}, EFT_{O_3}$
	$TF_{10}$	Grad des Dokumentationsbedarfs	$EF_{O_3}$	$EFT_{O_4}$
	$TF_{11}$	Zeitplanung und Termine	$EF_{O_4}$	$EFT_{O_5}$

Tabelle 5.9.: Synthese zu Testeinflussfaktoren

In Tabelle 5.9 ist das Ergebnis der Synthese in Form von 11 Testeinflussfaktoren zu sehen ( $TF_1$  bis  $TF_{11}$ ). Für die Definition der Testeinflussfaktoren wurden die extrahierten Einflussfaktoren zunächst nach Ähnlichkeit gruppiert. Anschließend wurden die Gruppen mit Bezeichnungen versehen, die die Namen der Testeinflussfaktoren darstellt. Ergänzend wurden die Testeinflussfaktoren den Kategorien Produkt, Personal, Infrastruktur und Organisation zugeordnet.

Die Testeinflussfaktoren werden in Abschnitt 5.2 aufgegriffen und in das Modell für Testeinflussfaktoren überführt. Bei der Überführung wird jeder Testeinflussfaktor mit Rückgriff auf die Gruppe an zugeordneten Einflussfaktoren (vgl. Tabelle 5.9) beschrieben. Dazu gehören die Überführung von allgemeinen

Einflussfaktoren auf die Testdomäne und die Beschreibung von Vorgaben für die Bewertung der Testeinflussfaktoren.

## 5.2. Modell für Testeinflussfaktoren

Für die Bewertung der in Abschnitt 5.1.3 identifizierten Testeinflussfaktoren fehlt ein einheitliches Verfahren und Vorgaben zur Beschreibung der Testeinflussfaktoren. Ohne ein einheitliches Bewertungsverfahren werden die Testeinflussfaktoren unterschiedlich interpretiert und angewendet (vgl. Problemstellung 2 in Abschnitt 1.1). Zur Bestimmung eines solchen Bewertungsverfahren wird in Abschnitt 5.2.1 zunächst der *Goal Question Metric* (GQM) Ansatz erläutert. Die Prinzipien des GQM Ansatzes werden anschließend aufgegriffen, um ein systematisches Vorgehen für den Aufbau des Modells für Testeinflussfaktoren zu entwickeln. Aufbauend darauf wird in Abschnitt 5.2.2 das Bewertungsverfahren der Testeinflussfaktoren definiert.

Danach werden in Abschnitt 5.2.3 formale Vorgaben für die einheitliche Beschreibung der Testeinflussfaktoren gegeben. Abschließend werden in den Abschnitten 5.2.4 bis 5.2.14 die in Abschnitt 5.1.3 identifizierten Testeinflussfaktoren mit dem entwickelten systematischen Vorgehen unter Berücksichtigung der Vorgaben zur Bewertung und Beschreibung der Testeinflussfaktoren verfeinert.

### 5.2.1. Entwicklung eines systematischen Vorgehens für den Aufbau des Modells für Testeinflussfaktoren

Das Vorgehen für den Aufbau des Modells für Testeinflussfaktoren folgt dem *Goal Question Metric* (GQM) Ansatz [BR88, BCR94, BRS<sup>+</sup>07]. Der GQM Ansatz beschreibt ein *top-down* Vorgehen, um aus gegebenen Zielen strukturiert Fragen und Metriken abzuleiten. Das Ergebnis der Anwendung des Ansatzes ist ein hierarchisches GQM Modell, das die drei Ebenen Ziele, Fragen und Metriken beinhaltet. Die Ziele werden im ersten Schritt des GQM Ansatzes definiert. Dabei besteht ein Ziel aus einem Vorhaben (z. B. Verbesserung), einem Problem (z. B. unzureichende Zeitplanung), einem zu messenden Objekt (z. B. Prozess

## 5. Entwicklung des Modells für Testeinflussfaktoren

für Änderungsmanagement) und einer Sicht (z.B. Projektmanager). Die Ziele werden mit entsprechenden Fragen verfeinert (z.B. Wie schnell werden Änderungen abgearbeitet?). Danach werden den Fragen Metriken zugeordnet (z.B. abgearbeitete Änderungen pro Stunde).

Nach der Definition eines GQM Modells werden die Metriken angewendet und die Ergebnisse gesammelt. Diese Ergebnisse werden interpretiert und *bottom-up* zur Beantwortung der Fragen und letztendlich zum Erreichen des übergeordneten Ziels verwendet. Für die Interpretation und Bewertung der Ergebnisse aus den Messungen mit den Metriken müssen vorher verbindliche Vorgaben festgelegt worden sein. Wenn dieselben Ziele, dieselben zu messenden Objekte und dieselbe Sicht vorliegen, ist ein GQM Modell wiederverwendbar. Für das Erreichen der übergeordneten Ziele müssen nur erneut die gegebenen Metriken angewendet, die Ergebnisse der Anwendung anhand entsprechender Vorgaben interpretiert und zur Beantwortung der Fragen genutzt werden.

Die Prinzipien des GQM Ansatzes sind auf Testeinflussfaktoren übertragbar. Damit ist ein Testeinflussfaktor über Fragen und Metriken zu bewerten. Die Gesamtheit der Testeinflussfaktoren inklusive der Fragen und der Metriken stellt das GQM Modell dar, das dem Modell für Testeinflussfaktoren entspricht. Im Folgenden wird dies beginnend mit der Zieldefinition erläutert.

In dem (GQM) Modell für Testeinflussfaktoren besteht die Zieldefinition aus zwei Teilen. Der erste Teil der Zieldefinition ist für alle Testeinflussfaktoren gleich. Das Vorhaben ist immer die Bewertung eines Testeinflussfaktors, um die quantitativen Auswirkungen in Prozent auf die globale Produktivität in einem Projekt zu ermitteln. Weiterhin ist die Sicht immer die des Testmanagers, der die Aufwandsschätzung durchführt.

Der zweite Teil der Zieldefinition, die zu betrachtenden Probleme und die zu messenden Objekte, variieren dagegen. In Abschnitt 5.1.3 wurde die Synthese der Menge aller Einflussfaktoren zu Testeinflussfaktoren beschrieben. Dabei wurden ähnliche Einflussfaktoren zu Testeinflussfaktoren gruppiert. Aus dieser Gruppierung von Einflussfaktoren sind für jeden Testeinflussfaktor die zu betrachtenden Probleme und zu messenden Objekte für die Zieldefinition des GQM Modell abzuleiten.

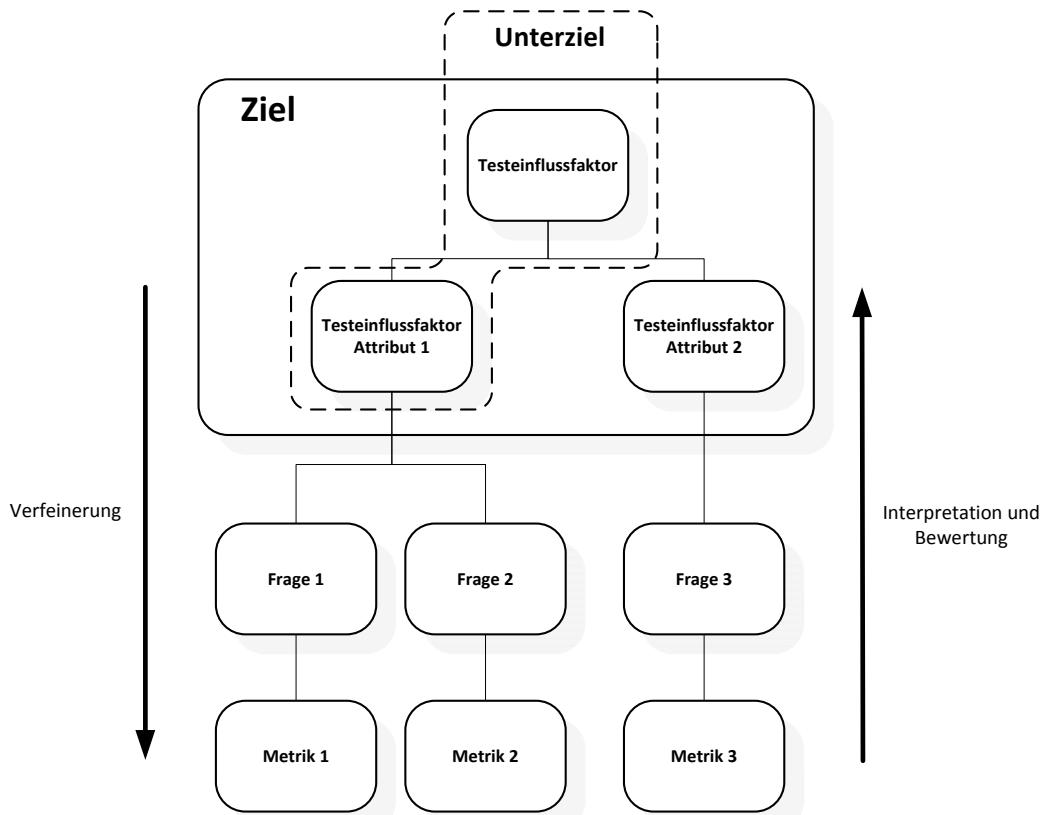


Abbildung 5.2.: Verbindung des Modells für Testeinflussfaktoren zu GQM

Innerhalb der Gruppen von Einflussfaktoren zu einem Testeinflussfaktor weichen das zu betrachtende Problem und das zu messende Objekt für eine gemeinsame Zieldefinition zu stark ab. Daher erfolgt der zweite Teil der GQM Zieldefinition über eine schrittweise Verfeinerung. Dies wird in Abbildung 5.2 dargestellt. Die Beschreibung des zu betrachtenden Problems und des zu messenden Objektes erfolgt abstrakt auf der Ebene des Testeinflussfaktors. Auf der Ebene des Attributes wird die Beschreibung verfeinert, so dass eine vollständige Zieldefinition erreicht wird. Die vollständige Zieldefinition für eine Kombination aus Testeinflussfaktor und Attribut wird in Abbildung 5.2 mit Unterziel bezeichnet.

Die Unterziele sind für jeden Testeinflussfaktor mit Fragen und Metriken zu verfeinern. Damit das Bewertungsverfahren der Testeinflussfaktoren einheitlich durchführbar ist, werden vereinfachend gegenüber dem GQM Ansatz die Antwortmöglichkeiten für alle Fragen vorgegeben. Die (qualitativen) Bewertungen der Fragen erfolgt auf der Ordinalskala:

## 5. Entwicklung des Modells für Testeinflussfaktoren

$$\text{sehr niedrig} < \text{niedrig} < \text{normal} < \text{hoch} < \text{sehr hoch} \quad (5.1)$$

Aus den vorgebenden Antwortmöglichkeiten folgt, dass alle Messergebnisse der Metriken entsprechend der genannten Skala interpretierbar sein müssen. Dies erfordert die Einführung von Vorgaben, die beschreiben, welche Kategorie der Skala zu welchem Ergebnis der Metrik gehört. Diese Vorgaben werden im Folgenden als Indikatoren bezeichnet.

### 5.2.2. Bewertungsverfahren für Testeinflussfaktoren

Über die Fragen und Metriken inklusive der Indikatoren ist das Bewertungsverfahren für Testeinflussfaktoren definierbar. Dabei wird wieder den Prinzipien des GQM Ansatzes gefolgt, wonach die Bewertungen der Testeinflussfaktoren *bottom-up* erfolgt. Im ersten Schritt sind die Metriken anzuwenden. Die Ergebnisse der Metriken ergeben zusammen mit den Vorgaben der Indikatoren die Antworten der zugehörigen Fragen.

Die Antworten der Fragen müssen in einem zweiten Schritt zu einer qualitativen Bewertung des zugehörigen Attributes zusammengefasst werden. Dabei wird die gleiche Ordinalskala wie bei der Bewertung der Fragen genutzt (vgl. Formel 5.1). Wie die qualitativen Bewertungen der Fragen zu interpretieren und zu kombinieren sind, wird in einem Bewertungsmodell am Attribut festgehalten.

Im dritten Schritt wird die qualitative Bewertung des Testeinflussfaktors aus den qualitativen Bewertungen der zugeordneten Attribute ermittelt. Die Skala ist wieder die oben genannte Ordinalskala. Die Vorgaben für die Interpretation werden in einem Bewertungsmodell am Testeinflussfaktor hinterlegt.

Für die Erreichung des Ziels muss die qualitative Bewertung abschließend in eine quantitative Auswirkung auf die globale Produktivität transformiert werden. Die Regeln hierfür werden am Testeinflussfaktor definiert. Sie haben die Form der Abbildung  $R$  in Formel 5.2.  $R$  bildet das Tupel  $T$  aus Formel 5.3 auf eine reelle Zahl ab. Diese reelle Zahl stellt die Auswirkung des Testeinflussfaktors auf die Produktivität in Prozent dar.

$$R : T \rightarrow \mathbb{R} \quad (5.2)$$

$$T = (\text{Teststufe}, \text{Testaktivität}, \text{Qual. Bewertung}) \quad (5.3)$$

Teststufe  $\in \{(\text{System-})\text{Integrationstest}, \text{Systemtest}, \text{Abnahmetest}\}$

Testaktivität  $\in \{\text{Testmanagement}, \text{Testdesign}, \text{Testausführung}\}$

Qual. Bewertung  $\in \{\text{sehr niedrig}, \text{niedrig}, \text{normal}, \text{hoch}, \text{sehr hoch}\}$

Für jeden Testeinflussfaktor muss ein Modell analog zu dem in Abbildung 5.2 aufgebaut werden. Die Gesamtheit der Modelle ergibt anschließend das Modell für Testeinflussfaktoren. Nach der Definition des Modells für Testeinflussfaktoren kann ein Einsatz im Kontext der Testaufwandsschätzung erfolgen.

### 5.2.3. Vorgaben zur Beschreibung der Testeinflussfaktoren

Neben dem beschriebenen Vorgehen analog zum GQM Ansatz sind verbindliche Vorgaben u. a. für die Zusammenhänge und Eigenschaften der Elemente Testeinflussfaktor, Testeinflussfaktor Attribut und Frage notwendig. Ohne entsprechende Vorgaben würden die Testeinflussfaktoren unterschiedlich beschrieben und könnten nicht auf die gleiche Weise in Formel 3.8 berücksichtigt werden. Weiterhin würde die Anwendbarkeit des Modells für Testeinflussfaktoren erschwert, da der strukturelle Aufbau der Testeinflussfaktoren nicht einheitlich ist. Daher werden in zwei Schritten verbindliche Vorgaben festgehalten. Im ersten Schritt wird die abstrakte Syntax des Modells für Testeinflussfaktoren in einem Metamodell definiert. Darauf basierend wird im zweiten Schritt eine konkrete Syntax vorgestellt, die anschließend zur Beschreibung der einzelnen Testeinflussfaktoren genutzt wird.

Die abstrakte Syntax des Modells für Testeinflussfaktoren ist in dem Metamodell in Abbildung 5.3 zu sehen. In dem Aufbau des Metamodells spiegelt sich die vorgestellte Struktur aus Zielen und Unterzielen. Das Element Testeinflussfaktor wird durch die Klasse *Test Cost Driver* repräsentiert. Zu einem Testeinflussfaktor gehören beliebig viele Attribute in Form der Klasse *Test Cost Driver Attribute*. Die Attribute selbst werden durch Fragen verfeinert (vgl. Klasse *Question*). Die qualitative Bewertung einer Frage für den Kontext einer Testauf-

## 5. Entwicklung des Modells für Testeinflussfaktoren

wandsschätzung wird in der Form der Klasse *Question Rating* festgehalten. Die möglichen Ausprägungen sind in der Aufzählung *Qualitative Rating Scale* definiert. Die Werte der Aufzählung entsprechen den Kategorien der Ordinalskala aus Formel 5.1.

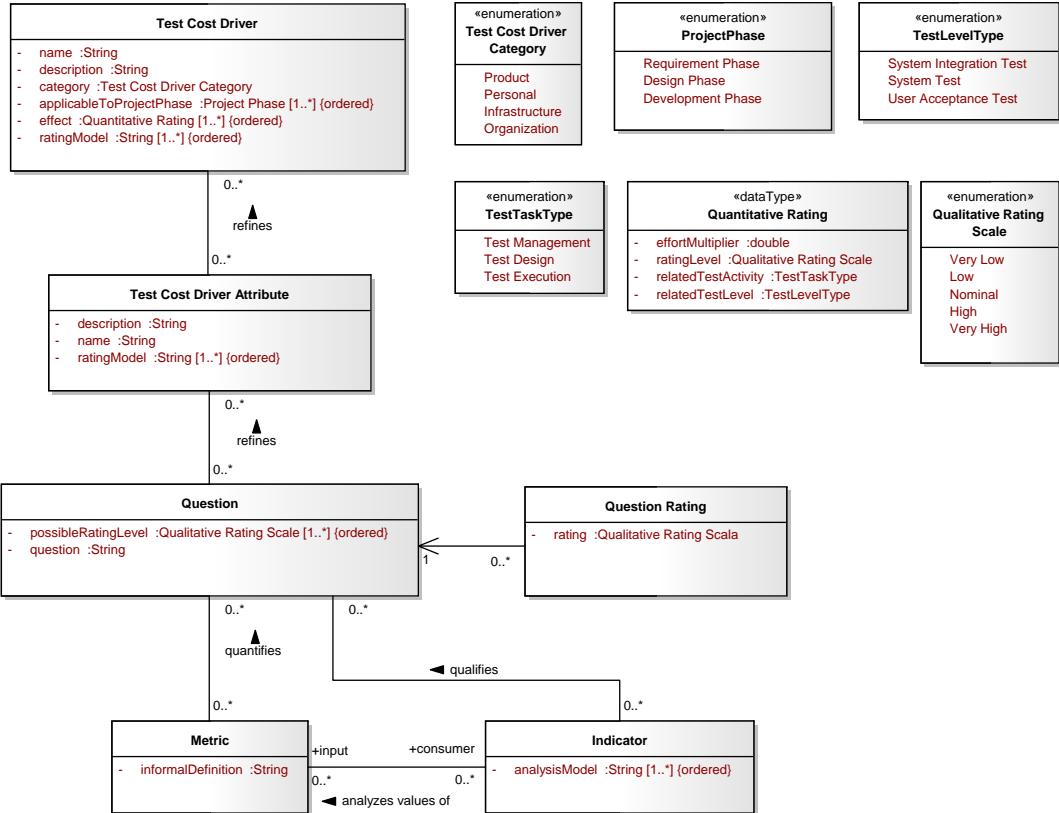


Abbildung 5.3.: Struktur des Modells für Testeinflussfaktoren

Zu jedem Testeinflussfaktor gehören ein Name und eine natürlich sprachliche Beschreibung. Weiterhin wird mit der Eigenschaft *category* die Zuordnung zu den in Abschnitt 5.1 vorgestellten Kategorien festgehalten. Die Kategorien sind in dem Metamodell in der Aufzählung *Test Cost Driver Category* hinterlegt.

Über die Eigenschaft *applicableToProjectPhase* des Testeinflussfaktors ist zu definieren, in welchen Projektphasen dieser anwendbar ist. Auswahlbar sind hier die Projektphasen aus der Aufzählung *ProjectPhase*. Die Eigenschaft *effect* beschreibt, welche Auswirkung ein Testeinflussfaktor auf die Produktivität besitzt. Die Auswirkung wird über den Datentyp *Quantitative Rating* je Teststufe, Testaktivität und qualitativer Ausprägung ein Prozentwert in der Eigenschaft *effortMultiplier* hinterlegt.

Die möglichen Belegungen für Teststufen, Testaktivitäten und qualitativer Skala werden in den zugehörigen Aufzählungen *TestLevelType*, *TestTaskType* und Aufzählung *Qualitative Rating Scale* festgelegt. Die Vorgaben zur Interpretation der qualitativen Bewertung der zugehörigen Attribute werden in einem Bewertungsmodell über die Eigenschaft *ratingModel* abgebildet. Das Ergebnis der Anwendung des Bewertungsmodells ist die qualitative Bewertung des Testeinflussfaktors.

Die Attribute in Form der Klasse *Test Cost Driver Attribute* werden durch einen Namen und eine natürlich sprachliche Beschreibung charakterisiert. Die Vorgaben zur Interpretation der Fragen, die zu einem Attribut gehören, werden wie beim Testeinflussfaktor in einem Bewertungsmodell definiert. Das Ergebnis der Anwendung ist die qualitative Bewertung des Attributes auf der Skala aus der Formel 5.1, die im der Aufzählung *Qualitative Rating Scale* entsprechend hinterlegt ist.

Den Attributen sind Fragen in der Form *Question* zugeordnet. Zu jeder Frage gehört die Frage selbst in der Eigenschaft *question* und die zugelassenen qualitativen Ausprägungen in der Eigenschaft *possibleRatingLevel*. Auch hier entsprechen die möglichen Ausprägungen denen, die in der Aufzählung *Qualitative Rating Scale* hinterlegt sind.

Die Bewertung der Fragen erfolgt in zwei Schritten: quantitativ und qualitativ. Grundlage für beide Schritte ist das Ergebnis einer Metrik, die der Frage zugeordnet ist. Das direkte Ergebnis der Metrik quantifiziert die Frage. Für eine Einordnung des Ergebnisses in qualitativer Form sind weitere Vorgaben erforderlich. Diese werden in einem Indikator in der Form der Klasse *Indicator* in der Eigenschaft *analysisModel* festgehalten.

Die Bezeichnung Indikator stammt aus der ISO-Norm *Measurement process*, die die Auswertung des Ergebnisses von Metriken bis zur Beantwortung der beschriebenen Informationsbedürfnisse als Prozess dargestellt [JTC07]. Der Indikator ist dabei definiert als “[...] a measure that provides an estimate or evaluation of specified attributes [...] with respect to defined information needs. Indicators are the basis for analysis and decision-making. These are what should be presented to measurement users. [...]“ [JTC07, S. 23]. Für die Übersetzung der Ergebnisse aus Messungen mit Metriken in Antworten auf die Informationsbedürfnisse nutzt ein Indikator (Analyse-)Modelle. Ein (Analyse-)Modell ist

## 5. Entwicklung des Modells für Testeinflussfaktoren

dabei “An algorithm or calculation combining one or more base and/or derived measures with associated decision criteria. It is based on an understanding of, or assumptions about, the expected relationship between the component measures and/or their behaviour over time. Models produce estimates or evaluations relevant to defined information needs.“ [JTC07, S. 24]. Dieser Definition entspricht die Eigenschaft *analysisModel* der Klasse *Indicator*.

---

### ***TF***<Nr. des Testeinflussfaktors> <Name des Testeinflussfaktors>

---

B	B ist die Beschreibung des Einflussfaktors
KA	KA ∈ {Produkt, Personal, Infrastruktur, Organisation}
V	V ∈ {Anforderungsphase, Designphase, Entwicklungsphase}
BM	BM ist die Beschreibung des Bewertungsmodells für den Testeinflussfaktor

---

### ***A***<Nr. des Testeinflussfaktors>.<Nr. des Attributes> <Name des Attributes>

---

B	B ist die Beschreibung des Attributes
BM	BM ist die Beschreibung des Bewertungsmodells für das Attribut.

---

### ***F***<Nr. des Testeinflussfaktors>.<Nr. des Attributes>.<Nr. der Frage> <Frage>

---

QA	QA ∈ {sehr niedrig, niedrig, normal, hoch, sehr hoch, extra hoch}
I	I ist die Übersetzung des Metrikergebnisses in eine QA
M	Beschreibung der Metrik

---

Tabelle 5.10.: Vorlage für die Beschreibung der Testeinflussfaktoren

In Tabelle 5.10 ist eine mögliche konkrete Syntax zu dem in Abbildung 5.3 dargestellten Metamodell zu sehen. Die Bestimmung der quantitative Auswirkung auf die Produktivität ist Teil der Evaluation in Abschnitt 5.3. Daher wird die Eigenschaft *effect* der Klasse *Test Cost Driver* nicht in der konkreten Syntax gezeigt. Die tabellarische Form wurde für die konkrete Syntax gewählt, da diese einen schnellen Überblick über einen Testeinflussfaktor gibt. Mit den Konventionen für die Nummerierung innerhalb der Benennung der Attribute und Fragen ist eine eindeutige Identifizierung der Elemente innerhalb von Bewertungsmodellen der Testeinflussfaktoren und Attribute möglich. Zur Vereinfachung gilt bei der Definition der Bewertungsmodelle:

- $A_{<\text{Nr. des Testeinflussfaktors}>.<\text{Nr. des Attributes}>}$  gibt das Ergebnis der Anwendung des Bewertungsmodells BM für das Attribut zurück.

- $F_{<\text{Nr. des Testeinflussfaktors}>.<\text{Nr. des Attributes}>.<\text{Nr. der Frage}>}$  gibt die qualitative Ausprägung QA zurück, die über die Anwendung der Metrik und des Indikators bestimmt worden ist.

Bei der Formalisierung der Bewertungsmodelle sollen Gewichtungen von Fragen und Attributen vorgenommen werden. Dazu ist eine Übersetzung der Skala aus Formel 5.1 in Zahlenwerte notwendig. Diese wird mit der bijektiven Funktion  $S$  in Formel 5.4 definiert.

$$\begin{aligned}
 S : \text{Qualitative Bewertung} &\rightarrow \mathbb{N} & (5.4) \\
 S(\text{sehr niedrig}) &= 1 \\
 S(\text{niedrig}) &= 2 \\
 S(\text{normal}) &= 3 \\
 S(\text{hoch}) &= 4 \\
 S(\text{sehr hoch}) &= 5
 \end{aligned}$$

In den Abschnitten 5.2.4 bis 5.2.14 werden die Testeinflussfaktoren aus Abschnitt 5.1.3 in die konkrete Syntax überführt.

### 5.2.4. Qualität der Testbasis

#### **TF<sub>1</sub>** Qualität der Testbasis

B	Der Testeinflussfaktor <i>Qualität der Testbasis</i> umfasst die Attribute <i>Qualität der Spezifikationen</i> und die <i>Stabilität der Spezifikationen</i> zusammen. Umso höher dieser Testeinflussfaktor bewertet wird, desto niedriger ist der Testaufwand.
KA	Produkt
V	Anforderungsphase, Designphase, Entwicklungsphase
BM	$\min(A_{1.1}, A_{1.2})$

#### **A<sub>1.1</sub>** Qualität der Spezifikationen

B	<p>Die Qualität der Spezifikationen wird in den drei Dimensionen Verständlichkeit, Konsistenz und Vollständigkeit bewertet. Verständlichkeit ergibt sich hier aus der Verwendung von Vorlagen und Standards bei der Erstellung der Spezifikationen. Wenn die Testbasis aus Dokumenten standardisierten Vorlagen besteht, reduziert sich der Einarbeitungsbedarf und die Anzahl der Rückfragen durch das Testpersonal. Damit wird insgesamt der Testaufwand verringert.</p> <p>Die Konsistenz bezieht sich auf die einheitliche Verwendung von Begriffen. Wenn Begriffe konsistent in den Spezifikationen verwendet werden, reduziert sich die Anzahl der Missverständnisse. Die Missverständnisse sind dabei nicht nur auf das Testpersonal beschränkt, sondern die Missverständnisse können schon bei der Entwicklung des Softwareproduktes zu mehr Fehlern geführt haben. Wenn diese Missverständnisse vermieden werden, sinkt der Testaufwand durch weniger Rückfragen durch das Testpersonal bei der Ausführung der Testaktivitäten und weniger Fehler in der Software.</p> <p>Vollständigkeit ergibt sich aus der Berücksichtigung der Anforderungen in den Spezifikationen. Wenn alle Anforderungen berücksichtigt worden sind, ist der Abstimmungsbedarf für die Ausführung von Testaktivitäten geringer und es werden Fehler bei der Entwicklung der Software vermieden. Damit sinkt insgesamt der Aufwand für den Test.</p>
BM	$\min(F_{1.1.1}, F_{1.1.2}, F_{1.1.3})$

---

*F<sub>1.1.1</sub>* Zu welchem Grad wurden für die Erstellung der Spezifikationen

Vorlagen und Standards verwendet?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	0	<0,25	0,26-0,59	0,6-0,75	>0,75
M	Verhältnis Anzahl erstellter Spezifikationsdokumente zu Anzahl erstellter Spezifikationsdokumente unter Berücksichtigung von Vorlagen und Standards.				

*F<sub>1.1.2</sub>* Wie konsistent sind die Anforderungen beschrieben?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	0	<0,25	0,26-0,59	0,6-0,75	>0,75
M	Verhältnis Anzahl verwendeter Begriffe zu Anzahl erklärter Begriffe in einem Glossar oder (Fach-)Klassendiagramm.				

*F<sub>1.1.3</sub>* Wie vollständig sind die Anforderungen beschrieben?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	0	<0,25	0,26-0,59	0,6-0,75	>0,75
M	Verhältnis Anzahl erwarteter Anforderungen gesamt zu Anzahl dokumentierter Anforderungen.				

### A<sub>1.2</sub> Stabilität der Spezifikationen

---

B	Die Rate und die Reichweite von Änderungen (müssen Teile der Software neu entwickelt werden) bestimmen die Stabilität der Spezifikationen. Wenn viele Änderungen mit großer Reichweite vorliegen, müssen Testaktivitäten wiederholt werden. Daher steigt mit sinkender Stabilität der Testaufwand.
BM	<i>F<sub>1.2.1</sub></i>

*F<sub>1.2.1</sub>* Wie stabil sind die Anforderungen?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	nicht	wenig	mittelmäßig	ziemlich	überwiegend
M	Subjektive Einschätzung der Stabilität der Anforderungen.				

### 5.2.5. Qualitätsziele

---

#### **TF<sub>2</sub>** Qualitätsziele

---

B	Der Testeinflussfaktor <i>Qualitätsziele</i> umfasst die Attribute <i>Funktionalität</i> und <i>Weitere Qualitätsmerkmale</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto höher ist der Testaufwand.
KA	Produkt
V	Designphase, Entwicklungsphase
BM	$\max(A_{2.1}, A_{2.2})$

---

#### **A<sub>2.1</sub>** Funktionalität

---

B	Dieses Attribut bezieht sich auf den gewünschten Abdeckungsgrad der Anforderungen im Test. Die Abdeckung wird von der Teststrategie bestimmt, die festlegt, wie welche Anforderung getestet werden soll. Wenn ein höherer Abdeckungsgrad gewünscht ist, müssen mehr Testfälle spezifiziert, koordiniert und ausgeführt werden. Damit erhöht sich bei steigendem Abdeckungsgrad der Testaufwand.  Für die Vergleichbarkeit der Einschätzung dieses Testeinflussfaktors muss festgelegt werden auf welchem Abstraktionsgrad der Begriff Anforderung genutzt wird. Beispielsweise kann eine Anforderung einem Use Case entsprechen oder einem Pfad in einem Use Case.
BM	$F_{2.1.1}$

---

*F<sub>2.1.1</sub>* Zu welchem Grad sollen die Anforderungen abgedeckt werden?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	0-0,24	0,25-0,49	0,5-0,74	0,75-0,9	>0,9
M	Verhältnis von Anzahl der Anforderungen zu zu testenden Anforderungen.				

---

---

**A<sub>2.2</sub>** Weitere Qualitätsmerkmale

---

B	Dieses Attribut bezieht sich auf die weiteren Qualitätsmerkmale Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit. Wenn Anforderungen hinsichtlich dieser weiteren Qualitätsmerkmale vorliegen, steigt der Aufwand für den Test.
BM	$F_{2.2.2}$

---

$F_{2.2.1}$  Wie viele weitere Qualitätsmerkmale sollen im Test berücksichtigt werden?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	0	1	2	3	> 3
M	Anzahl zu berücksichtigender Qualitätsmerkmale aus der Menge: Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit				

---

### 5.2.6. Wiederverwendbarkeit

#### **TF<sub>3</sub>** Wiederverwendbarkeit von Testmitteln

B	Der Testeinflussfaktor <i>Wiederverwendbarkeit von Testmitteln</i> umfasst die Attribute <i>Wiederverwendbarkeit von Testmitteln</i> und <i>Ähnlichkeit der zu testenden Funktionalität</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto niedriger ist der Testaufwand.
KA	Produkt
V	Designphase, Entwicklungsphase
BM	$\max(A_{3.1}, A_{3.2})$

#### **A<sub>3.1</sub>** Wiederverwendbarkeit von Testmitteln

B	Dieses Attribut bezieht sich auf die Wiederverwendbarkeit von Testmitteln in dem aktuellen Testprojekt. Wenn Testmittel (Testfälle, Testdaten, Dokumente usw.) aus vorherigen Projekten oder Teststufen wiederverwendbar sind, können diese vollständig oder in leicht veränderter Form genutzt werden. Damit sinkt der Aufwand für die Testaktivitäten, die für die Erstellung der jeweiligen Testmittel zuständig sind.
BM	$F_{3.1.1}$

$F_{3.1.1}$  Zu welchen Grad sind Testmittel aus vorherigen Projekten oder Teststufen wiederverwendbar?

QA	normal	hoch	sehr hoch
I	nicht	mittelmäßig	sehr
M	Subjektive Einschätzung der Frage, zu welchem Grad Testmittel wiederverwendbar sind.		

---

**A<sub>3.2</sub> Ähnlichkeit der zu testenden Funktionalität**

---

B	Dieses Attribut bezieht sich auf die Ähnlichkeit der zu testenden Funktionalität. Wenn die Funktionalität hinsichtlich der Struktur und dem Verhalten ähnlich ist, ist Wiederverwendbarkeit bei den Testaktivitäten Testdesign und Testausführung möglich. Damit ist der Aufwand für Testdesign und Testausführung niedriger, wenn die Wiederverwendbarkeit steigt. Zusätzlich sinkt auch der Aufwand für das Testmanagement, da weniger Koordinierung der anderen Testaktivitäten notwendig ist.
BM	<i>F<sub>3.2.1</sub></i>

---

*F<sub>3.2.1</sub>* Wie ähnlich ist die zu testende Funktionalität?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I			nicht	mittelmäßig	sehr
M	Subjektive Einschätzung der Frage, wie ähnlich die zu testende Funktionalität ist.				

---

### 5.2.7. Qualifikation und Erfahrung

---

#### **TF<sub>4</sub> Qualifikation und Erfahrung**

---

B	Der Testeinflussfaktor <i>Qualifikation und Erfahrung</i> umfasst die Attribute <i>Qualifikation des Personals</i> und <i>Erfahrung des Personals</i> . Der Begriff Personal bedeutet hier das Testpersonal und das Entwicklungspersonal. Umso höher dieser Testeinflussfaktor bewertet wird, desto niedriger ist der Testaufwand.
KA	Personal
V	Designphase, Entwicklungsphase
BM	$\min(A_{4.1}, A_{4.2})$

---

#### **A<sub>4.1</sub> Qualifikation des Personals**

---

B	<p>Dieses Attribut bezieht sich auf die Qualifikation der Projektmitglieder. Dabei wird die Qualifikation der Analytiker, der Architekten, der Entwickler und des Testpersonals getrennt bewertet. Im Bewertungsmodell wird anschließend eine stärkere Gewichtung der Qualifikation des Testpersonals berücksichtigt, da dieses hauptverantwortlich für die Durchführung des Testprojektes ist.</p> <p>Die Qualifikationen der Analytiker, der Architekten und der Entwicklern bestimmen die Effizienz und Effektivität von Unterstützungsleistungen für das Testpersonal. Wenn z. B. Fehler, die während der Durchführung von Testaktivitäten aufgedeckt worden sind, effizient und effektiv behoben werden, ist eine zeitliche Verzögerung durch diese Fehler unwahrscheinlicher. Damit sinkt bei steigender Qualifikation der Analytikern, der Architekten und der Entwicklern der Testaufwand.</p> <p>Die Qualifikation des Testpersonals bestimmt die Effizienz und Effektivität der Durchführung der Testaktivitäten. Wenn das Testpersonal z. B. vertraut mit den Methoden für die Ableitung von Testfällen ist, kann die Testaktivität Testdesign effektiver und effizienter ausgeführt werden. Damit sinkt bei steigender Qualifikation des Testpersonals der Testaufwand.</p>
BM	$\min(F_{4.1.1}, F_{4.1.2}, F_{4.1.3}, F_{4.1.4})$

---

---

*F<sub>4.1.1</sub>* Wie qualifiziert sind die Analytiker?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	$\leq 15\%$	35 %	55 %	75 %	90 %
M	Subjektive Bewertung in Prozent, die u. a. die Beurteilung der Fähigkeiten zur Analyse, Gründlichkeit, Kommunikation und Kooperation einschließen sollte.				

---

*F<sub>4.1.2</sub>* Wie qualifiziert sind die Architekten?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	$\leq 15\%$	35 %	55 %	75 %	90 %
M	Subjektive Bewertung in Prozent, die u. a. die Beurteilung der Fähigkeiten zur Analyse, Gründlichkeit, Kommunikation und Kooperation einschließen sollte.				

---

*F<sub>4.1.3</sub>* Wie qualifiziert sind die Entwickler?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	$\leq 15\%$	35 %	55 %	75 %	90 %
M	Subjektive Bewertung in Prozent, die u. a. die Beurteilung der Fähigkeiten zur Analyse, Gründlichkeit, Kommunikation und Kooperation einschließen sollte.				

---

*F<sub>4.1.4</sub>* Wie qualifiziert ist das Testpersonal?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	$\leq 15\%$	35 %	55 %	75 %	90 %
M	Subjektive Bewertung in Prozent, die u. a. die Beurteilung der Fähigkeiten zur Analyse, Gründlichkeit, Kommunikation und Kooperation einschließen sollte.				

---

---

#### A4.2 Erfahrung des Personals

---

- B Die Erfahrung des Personals wird getrennt von der Qualifikation betrachtet, da die Erfahrung in keinem direkten Zusammenhang mit der Qualifikation steht. Ein qualifiziertes Personal ist z. B. nicht automatisch erfahren. Die Erfahrung des Personal wird in vier Dimensionen bewertet: Erfahrung mit dem Softwaretyp, Erfahrung in der Plattform, Erfahrung im firmeneigenen Testprozess und Sprach-/Werkzeugerfahrung. Weiterhin wird die direkte Erfahrung des Personals mit der zu testenden Software berücksichtigt. Durch das entsprechende Bewertungsmodell wird die direkte Erfahrung stärker bei der Gesamtbewertung des Attributes gewichtet.
- BM In diesem Bewertungsmodell wird die Erfahrung des Personals mit der zu testenden Software stärker gewichtet. Wenn die Tester Erfahrung mit dem Produkt haben, entfällt u. a. der Einarbeitungsaufwand durch andere Projektmitglieder. Die Tester können dann von Beginn effektiver testen und finden Abweichungen effizienter [Dow09, S. 94]. Für die Berechnung der qualitativen Ausprägung wird zunächst die niedrigste qualitative Bewertung der Fragen  $F_{4.2.1}$  bis  $F_{4.2.4}$  ermittelt. Diese wird durch die qualitative Bewertung der Frage  $F_{4.2.5}$  einen Schritt nach oben oder unten auf der qualitativen Skala verschoben, falls die Bewertungen abweichen.

$$m = \min(F_{4.2.1}, F_{4.2.2}, F_{4.2.3}, F_{4.2.4})$$

$$BM = \begin{cases} S^{-1}((S(m) - 1)) & \text{wenn } m > F_{4.2.5} \\ S^{-1}((S(m) + 1)) & \text{wenn } m < F_{4.2.5} \\ F_{4.2.5} & \text{sonst} \end{cases}$$

---

$F_{4.2.1}$  Wie hoch ist die durchschnittliche Erfahrung mit dem Softwaretyp?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	< 1 Jahr	1-2 Jahre	3-5 Jahre	6-9 Jahre	$\geq 10$ Jahre
M	Durchschnittliche Erfahrung des Personals in Anzahl Monaten bzw. Jahren mit dem Softwaretyp. Softwaretyp $\in \{\text{Finanzsoftware, Kundenbetreuung, Onlineshop, Kundenbewertung, Kundenbindung, ...}\}$				

---

---

F<sub>4.2.2</sub> Wie hoch ist die durchschnittliche Erfahrung mit der Plattform?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	< 6 Monate	6-11 Monate	1-2 Jahr	3-5 Jahre	≥6 Jahre
M	Durchschnittliche Erfahrung des Personals in Anzahl Monaten bzw. Jahren mit der Plattform auf der die Software betrieben/getestet wird. Plattform ∈ {Web, Desktop, Mobiles Gerät,...}				

---

F<sub>4.2.3</sub> Wie hoch ist die durchschnittliche Erfahrung im firmeneigenen

Testprozess?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	< 4 Monate	4-11 Monate	1-2 Jahr	3 Jahre	> 3 Jahre
M	Durchschnittliche Erfahrung des Personals in Anzahl Monaten bzw. Jahren mit dem firmeneigenen Testprozess				

---

F<sub>4.2.4</sub> Wie hoch ist die durchschnittliche Sprach-/Werkzeugerfahrung?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	< 2 Monate	3-6 Monate	7 M. - 2 J.	3-5 Jahre	≥6 Jahre
M	Durchschnittliche Erfahrung des Personals in Anzahl Monaten bzw. Jahren mit den im Projekt genutzten Modellierungssprachen und Werkzeugen				

---

F<sub>4.2.5</sub> Wie hoch ist die durchschnittliche Erfahrung mit der zu testenden

Software?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	<4 Monate	4-11 Monate	1-2 Jahr	3 Jahre	> 3 Jahre
M	Durchschnittliche Erfahrung des Personals in Anzahl Monaten bzw. Jahren mit der zu testenden Software				

---

### 5.2.8. Kommunikationsaufwand

#### **TF<sub>5</sub>** Kommunikationsaufwand

B	Der Testeinflussfaktor <i>Kommunikationsaufwand</i> umfasst die Attribute <i>Anzahl Ansprechpartner</i> und <i>Größe des Testteams</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto höher ist der Testaufwand.
KA	Personal
V	Anforderungsphase, Designphase, Entwicklungsphase
BM	$\max(A_{5.1}, A_{5.2})$

#### **A<sub>5.1</sub>** Anzahl Ansprechpartner

B	Dieses Attribut bezieht sich auf die Anzahl der Ansprechpartner, die z. B. Unterstützungsleistungen für die Erstellung von Testfällen oder die Testausführung leisten müssen. Dies schließt auch die Unterstützungsleistungen durch die Entwickler mit ein. Wenn nur wenige Ansprechpartner zur Durchführung der Testaktivitäten erforderlich sind, ist weniger Kommunikation zur Koordination oder zur Klärung von Rückfragen notwendig. Damit ist bei einer kleinen Anzahl von Ansprechpartnern der Kommunikationsaufwand geringer und damit auch der gesamte Testaufwand. Die Definition des Indikators folgt der Definition des Einflussfaktors <i>Anzahl Ansprechpartner</i> in der Methode UCP 3.0 [Fro09, S. 126].
BM	$F_{5.1.1}$

$F_{5.1.1}$  Wie viele Ansprechpartner sind für die Durchführung der

Testaktivitäten relevant?

QA	niedrig	normal	hoch
I	$\leq 5$	6-14	$\geq 15$
M	Anzahl fachlicher und technischer Ansprechpartner		

---

**A5.2 Größe des Testteams**

---

B	Dieses Attribut bezieht sich auf die Größe des Testteams. Dazu gehört nur Personal in den Rollen Testmanager, Testdesigner oder Tester. Wenn das Testteam klein ist, ist weniger Kommunikation zur Abstimmung innerhalb des Testteams notwendig. Damit ist bei einer kleinen Anzahl Projektmitglieder der Kommunikationsaufwand geringer und damit auch der gesamte Testaufwand. Die Definition des Indikators folgt der Definition des Einflussfaktors Teamgröße in der Methode TPA [vBKV08, S. 473].
BM	$F_{5.2.1}$

---

$F_{5.2.1}$  Wie groß ist das Testteam?

---

QA	niedrig	normal	hoch
I	$\leq 4$	5-9	$\geq 10$
M	Anzahl Personen in den Rollen Testmanager, Testdesigner oder Tester		

### 5.2.9. Zusammenarbeit

---

#### **TF<sub>6</sub>** Zusammenarbeit der Projektmitglieder

---

B	Der Testeinflussfaktor <i>Zusammenarbeit der Projektmitglieder</i> umfasst die Attribute <i>Anzahl Standorte</i> und <i>Zusammenarbeit</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto höher ist der Testaufwand.
KA	Personal
V	Designphase, Entwicklungsphase
BM	In diesem Bewertungsmodell wird die Zusammenarbeit im Team stärker gewichtet. Damit wird der Argumentation aus der Methode COCOMO II gefolgt, wonach die Zusammenarbeit im Team exponentiell auf die Produktivität wirkt [Boe00]. Dabei ist für die Bewertung nach COCOMO II die Anzahl der Standorte nicht relevant. Die Anzahl der Standorte wird als multiplikativer Faktor ( <i>Multisite Development</i> ) separat betrachtet und damit weniger hoch gewichtet [Boe00, S. 49]). Für die Berechnung der qualitativen Ausprägung wird die Bewertung des Attributes 2, durch die qualitative Bewertung des Attributes 1 einen Schritt nach oben oder unten auf der qualitativen Skala verschoben wird, falls die Bewertungen der Attribute um zwei Stufen abweichen.

$$BM = \begin{cases} S^{-1}((S(A_{6.2}) - 1)) & \text{wenn } S(A_{6.1}) + 1 < S(A_{6.2}) \\ S^{-1}((S(A_{6.2}) + 1)) & \text{wenn } S(A_{6.1}) - 1 > S(A_{6.2}) \\ A_{6.2} & \text{sonst} \end{cases}$$

---

#### **A<sub>6.1</sub>** Anzahl Standorte

---

B	Dieses Attribut bezieht sich auf die räumliche Verteilung der fachlichen/technischen Ansprechpartner und des Testteams. Wenn die Ansprechpartner und das Testteam auf viele Standorte verteilt sind, ist der Aufwand zur Zusammenarbeit höher. Damit ist bei steigender Anzahl Standorte der Testaufwand höher.
BM	$F_{6.1.1}$

---

*F<sub>6.1.1</sub>* Wie viele Standorte sind für die Durchführung der Testaktivitäten relevant?

QA	niedrig	normal	hoch
I	1	2	>2
M	Anzahl der Standorte		

**A<sub>6.2</sub> Zusammenarbeit**

B	Dieses Attribut bezieht sich auf die Zusammenarbeit des am Projekt beteiligten Personals. Bei der Bewertung wird auch die Zusammenarbeit mit den fachlichen/technischen Ansprechpartnern, dem Personal für weitere Unterstützungsleistungen und den Entwicklern berücksichtigt. Im Kern geht es dabei um die Zusammenarbeit, die notwendig ist, um die Testaktivitäten durchführen zu können. Wenn die Zusammenarbeit schwierig ist, ist der Aufwand für die Erfüllung der Testaktivitäten höher. Damit steigt bei schwieriger Zusammenarbeit der Testaufwand.  Eine hohe Personalfluktuation führt zu Schwierigkeiten in der Zusammenarbeit, da neues Personal immer wieder eingearbeitet werden muss. Damit ist bei hoher Personalfluktuation der Aufwand für die Durchführung der Testaktivitäten höher.
BM	$\max(F_{6.2.1}, F_{6.2.2})$

*F<sub>6.2.1</sub>* Wie schwierig ist die erwartete Zusammenarbeit?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	nicht	wenig	mittelmäßig	ziemlich	sehr
M	Subjektive Bewertung der erwarteten Schwierigkeiten bei der Zusammenarbeit. Zu berücksichtigen sind auch: Anzahl Muttersprachen innerhalb des Projektteams, Engagement der Stakeholder, Erfahrung in der Teamarbeit				

*F<sub>6.2.2</sub>* Wie hoch ist die Fluktuation des Personals im gesamten Projektteam?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	$\leq 3\%$	6 %	7-12 %	13-24 %	$\geq 48\%$
M	Wechsel von Personal während der Projektlaufzeit				

### 5.2.10. Stabilität und Verfügbarkeit der Testumgebung

#### **TF<sub>7</sub>** Stabilität und Verfügbarkeit der Testumgebung

B	Der Testeinflussfaktor <i>Stabilität und Verfügbarkeit der Testumgebung</i> umfasst die Attribute <i>Stabilität der Testumgebung</i> und <i>Verfügbarkeit der Testumgebung</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto niedriger ist der Testaufwand.
KA	Infrastruktur
V	Entwicklungsphase
BM	$\min(A_{7.1}, A_{7.2})$

#### **A<sub>7.1</sub>** Stabilität der Testumgebung

B	Dieses Attribut bezieht sich auf die Stabilität der Testumgebung. Dabei wird die vorherige Erfahrung mit der Testumgebung, die rechtzeitige Verfügbarkeit und die Abstimmung der Testumgebung bewertet. Im Bewertungsmodell wird anschließend eine stärkere Gewichtung der Frage 1.1 berücksichtigt, da bei einer steigenden Anzahl von durchgeführten Testprojekten auf einer Testumgebung die Stabilität der Testumgebung steigt.  Die Reife der Testumgebung bestimmt die Stabilität der Testumgebung mit. Wenn schon mehrere Testprojekte auf der gleichen Testumgebung durchgeführt worden sind, kann von einer höheren Stabilität ausgegangen werden. Dabei ist eine Testumgebung als gleich zu betrachten, wenn die gleichen Systeme in gleicher Konfiguration vorhanden sind.
BM	$\min(F_{7.1.1}, F_{7.1.2})$

#### **F<sub>7.1.1</sub>** Wie reif ist die Testumgebung?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	0	1	2	$\leq 5$	$> 10$
M	Anzahl vorheriger Testprojekte auf gleicher Testumgebung				

---

*F<sub>7.1.2</sub>* Wie wahrscheinlich wird die Konfiguration der Testumgebung vor und während des Testprojektes mit dem Testmanager abgestimmt?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	keinesfalls	wahrscheinlich	vielleicht	ziemlich	ganz
		nicht		wahrscheinlich	wahrscheinlich
M	Subjektive Einschätzung der Wahrscheinlichkeit, dass Änderungen an der Konfiguration vor und während der Durchführung des Tests mit dem Testmanager abgestimmt werden.				

---

#### A<sub>7.2</sub> Verfügbarkeit der Testumgebung

- |    |  |
|----|--|
| B  | <p>Dieses Attribut bezieht sich auf die Verfügbarkeit der Testumgebung. Dabei sind die rechtzeitige Verfügbarkeit und der Grad der Verfügbarkeit während der Testdurchführung von der geplanten Verfügbarkeit berücksichtigt.</p> <p>Die verzögerte oder nicht abgestimmte Bereitstellung und Konfiguration der Testumgebung führt zu Verzögerungen bei der Durchführung von Testaktivitäten. Damit steigt der Aufwand für den Test. Wenn die Testumgebung weniger als geplant zur Verfügung steht, entstehen Wartezeiten während der Testausführung. Damit steigt der Aufwand für den Test.</p> |
| BM | <p>Im Bewertungsmodell wird berücksichtigt, dass die Verfügbarkeit nicht über 100 % der geplanten Zeit hinausgehen kann. Daher wird die Antwort auf die Frage <i>F<sub>7.2.1</sub></i> nur berücksichtigt, wenn die Verfügbarkeit von 100 % abweicht.</p>  |
-

## 5. Entwicklung des Modells für Testeinflussfaktoren

$$BM = \begin{cases} \min(F_{7.2.1}, F_{7.2.2}) & \text{wenn } S(F_{7.2.1}) = 2 \\ \min(F_{7.2.1}) & \text{sonst} \end{cases}$$

*F<sub>7.2.1</sub>* Wie wahrscheinlich steht die Testumgebung rechtzeitig zur Verfügung?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	keinesfalls	wahrscheinlich	vielleicht	ziemlich	ganz
M	nicht wahrscheinlich wahrscheinlich wahrscheinlich				

Subjektive Einschätzung der Wahrscheinlichkeit für die rechtzeitige Bereitstellung und Konfiguration der Testumgebung.

*F<sub>7.2.2</sub>* Zu welchem Grad ist die Testumgebung während der Testdurchführung verfügbar?

QA	sehr niedrig	niedrig	normal
I	40 %	70 %	100 %
M	Verhältnis tatsächlicher Verfügbarkeit zu geplanter Verfügbarkeit in Prozent		

### 5.2.11. Testinfrastruktur

---

#### **TF<sub>8</sub>** Testinfrastruktur

---

B	Der Testeinflussfaktor <i>Testinfrastruktur</i> umfasst die Attribute <i>Verfügbarkeit von Ressourcen</i> , <i>Qualität der Infrastruktur</i> und <i>Werkzeugunterstützung</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto niedriger ist der Testaufwand.
KA	Infrastruktur
V	Entwicklungsphase
BM	Im Bewertungsmodell wird berücksichtigt, dass die Verfügbarkeit von Ressourcen nicht über 100 % der geplanten Zeit hinausgehen kann. Daher wird die Antwort auf des Attributes $A_{8.1}$ nur berücksichtigt, wenn die Verfügbarkeit von 100 % abweicht.

$$BM = \begin{cases} \min(A_{8.2}, A_{8.3}) & \text{wenn } S(A_{8.1}) = 2 \\ \min(A_{8.1}, A_{8.2}, A_{8.3}) & \text{sonst} \end{cases}$$


---

#### **A<sub>8.1</sub>** Verfügbarkeit von Ressourcen

---

B	Dieses Attribut bezieht sich auf die Verfügbarkeit der Ressourcen, die für die Durchführung der Testaktivitäten notwendig sind. Dabei umfasst der Begriff die Ressourcen Hardware und Software, die außerhalb der Testumgebung genutzt werden. Die Testumgebung wird in Testeinflussfaktor $TF_7$ separat betrachtet. Wenn die Ressourcen weniger als geplant zur Verfügung stehen, entstehen Wartezeiten. Dies führt zu einer Steigerung des Aufwandes für den Test.
BM	$F_{8.1.1}$

---

$F_{8.1.1}$  Zu welchem Grad stehen die Ressourcen zur Durchführung der Testaktivitäten bereit?

---

QA	sehr niedrig	niedrig	normal
I	40 %	70 %	100 %
M	Verhältnis tatsächlicher Verfügbarkeit zu geplanter Verfügbarkeit in Prozent		

---

## 5. Entwicklung des Modells für Testeinflussfaktoren

---

### A<sub>8.2</sub> Qualität der Infrastruktur

---

B	Dieses Attribut bezieht sich auf die Qualität der Ressourcen, die für die Durchführung der Testaktivitäten notwendig sind. Dabei wird Qualität auf die Wartezeit und Verzögerung während Aktionen für die Durchführung von Testaktivitäten bezogen. Wenn hohe Wartezeiten auf Testergebnisse erforderlich sind, können z. B. Folgeschritte in Testaktivitäten blockiert sein. Zusätzlich sind die Wiederholungen von Testfällen bei behobenen Abweichungen zeitintensiver. Damit erhöht sich bei steigender Wartezeit der Testaufwand.
BM	$\min(F_{8.2.1}, F_{8.2.2})$

---

F<sub>8.2.1</sub> Wie schnell sind die Testergebnisse für einen Testfall verfügbar?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	>8	4-7	2-3	0,5-1	<0,5
M	Wartezeit in Stunden				

---

F<sub>8.2.2</sub> Wie schnell reagiert die Infrastruktur bei Aktionen

für die Durchführung von Testaktivitäten?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	>2	1-2	0,5-0,9	0,25-0,4	<0,25
M	Höhe der durchschnittlichen Verzögerung in Sekunden				

---

---

**A8.3 Werkzeugunterstützung**

---

B	Dieses Attribut bezieht sich auf die Werkzeugunterstützung, die im Testprojekt genutzt wird. Wenn Werkzeuge zur Unterstützung der Durchführung von Testaktivitäten eingesetzt werden, kann z. B. die Konsistenz zwischen Testmitteln automatisch sichergestellt, Teile der Testausführung automatisiert oder der Fortschritt von Testaktivitäten besser überwacht werden. Damit sinkt bei steigendem Grad der Werkzeugunterstützung der Aufwand für die Durchführung der Testaktivitäten.
BM	<i>F<sub>8.3.1</sub></i>

---

*F<sub>8.3.1</sub>* Welchen Typen von Werkzeugunterstützung werden eingesetzt?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I		DTA	DTA	DTA	DTA
			TMM	TMM	TMM
				MW	MW
					AW
Der Indikator gibt an, welche Typen von Werkzeugunterstützung mindestens für eine qualitative Ausprägung genutzt werden müssen. Dies bedeutet, dass das Vorhandensein der den Werkzeugtypen TMM und MW zu der qualitative Ausprägung sehr niedrig führt.					
M	Typ von Werkzeugunterstützung = {Managementwerkzeuge (MW), Testmittelmanagement (TMM), Automatisierungswerkzeuge (AW), dedizierte Testarbeitsplätze (DTA)}				

---

### 5.2.12. Gesamttestorganisation

---

#### **TF<sub>9</sub>** Gesamttestorganisation

---

B	Der Testeinflussfaktor <i>Gesamttestorganisation</i> umfasst die Attribute <i>Management der Projektrisiken</i> und <i>Prozessreife</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto niedriger ist der Testaufwand.
KA	Organisation
V	Anforderungsphase, Designphase, Entwicklungsphase
BM	$\min(A_{9.1}, A_{9.2})$

---

#### **A<sub>9.1</sub>** Management der Projektrisiken

---

B	Dieses Attribut bezieht sich auf das Management der Projektrisiken. Wenn kein aktives Management betrieben wird, kann die Durchführung von Testaktivitäten durch das Eintreten von unvorhergesehenen Ereignissen behindert oder blockiert werden. Durch entsprechende Vorbereitung können diese Ereignisse im Vorfeld identifiziert und entsprechende Reaktionen definiert werden, so dass die Durchführung der Testaktivitäten möglichst nicht behindert oder blockiert wird. Dies könnte z. B. durch entsprechende Umgruppierung von Testfällen realisiert werden. Daraus folgt, dass bei fehlendem Management der Projektrisiken der Aufwand für den Test steigt.
BM	$F_{9.1.1}$

---

##### *F<sub>9.1.1</sub>* Mit welchen Techniken und Hilfsmitteln werden Projektrisiken erfasst?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	keine Technik	1	2	3	$\geq 4$
	keine Hilfsmittel				
M	Anzahl Techniken und Hilfsmittel zur Risikoidentifikation $\in \{\text{Experten (E)}, \text{Verwendung Vorlagen (VV)}, \text{Checklisten (CL)}, \text{Risikoworkshop (RW)}, \text{abgeschlossene Projekte (AP)}\}$				

---

---

**A9.2 Prozessreife**

---

B	<p>Dieses Attribut bezieht sich auf die Reife des Testprozesses. Zu einem Testprozess gehören die Definition der Testaktivitäten und die Techniken sowie Hilfsmittel zur Durchführung der Testaktivitäten. Die Reife eines Testprozesses steht in der Verbindung mit der Effizienz und der Effektivität der durchgeführten Testaktivitäten. Wenn die Reife niedrig ist, werden die Testaktivitäten weniger effizient und effektiv durchgeführt.</p> <p>Ein reifer Testprozess trägt z. B. dazu bei, dass die Ansprechpartner für den Test koordiniert, mögliche Änderungen an den Anforderungen oder der Konfiguration der Testumgebung entsprechend abgesprochen und Testwerkzeuge richtig verwendet werden. Damit steigt bei höherer Reife des Testprozesses die Effizienz und Effektivität der Durchführung von Testaktivitäten. Daraus folgt, dass eine höhere Reife zu weniger Testaufwand führt.</p>
BM	$F_{9.2.1}$

---

$F_{9.2.1}$  Welche Reife hat der Testprozess nach TPI Next?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	kein	I	K	E	O
M	Prozess				
M	Einstufung nach TPI Next in Stufen $\in \{\text{initial (I)}, \text{kontrolliert (K)}, \text{effizient (E)}, \text{optimierend (O)}\}$				

---

### 5.2.13. Dokumentationsbedarf

---

#### **TF<sub>10</sub>** Grad des Dokumentationsbedarfs

---

B	Der Testeinflussfaktor <i>Grad des Dokumentationsbedarfs</i> umfasst das Attribut <i>Dokumentationsbedarf</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto höher ist der Testaufwand.
KA	Organisation
V	Designphase, Entwicklungsphase
BM	$A_{10.1}$

---

#### **A<sub>10.1</sub>** Dokumentationsbedarf

---

B	Dieses Attribut bezieht sich auf den Dokumentationsbedarf bzgl. der Testmittel bei der Durchführung der Testaktivitäten. Bei höherem Dokumentationsbedarf, z. B. durch formale Anforderungen des Kunden oder gesetzliche Rahmenbedingungen, steigt der Aufwand für die Durchführung der Testaktivitäten, da die Dokumente entsprechend erstellt und gepflegt werden müssen. Damit steigt bei höheren Anforderungen an die Dokumentation der Aufwand für den Test.
BM	$F_{10.1.1}$

---

$F_{10.1.1}$  Wie hoch sind die Anforderungen an die Dokumentation der Testmittel?

---

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	nicht	wenig	mittelmäßig	ziemlich	sehr
M	Subjektive Einschätzung der notwendigen Intensität für die Dokumentation der Testmittel				

### 5.2.14. Zeitplanung

---

#### **TF<sub>11</sub>** Zeitplanung und Termine

B	Der Testeinflussfaktor <i>Zeitplanung und Termine</i> umfasst das Attribut <i>Termindruck</i> . Umso höher dieser Testeinflussfaktor bewertet wird, desto höher ist der Testaufwand.
KA	Organisation
V	Designphase, Entwicklungsphase
BM	<i>A<sub>11.1</sub></i>

---

#### **A<sub>11.1</sub>** Termindruck

B	Dieses Attribut bezieht sich auf den Termindruck in einem Testprojekt. Wenn die Zeitplanung stark komprimiert wird, steht wenig Zeit für die Erfüllung der notwendigen Testaktivitäten zur Verfügung. Daher muss zusätzliches Personal eingesetzt werden, um die Testaktivitäten zu bearbeiten. Speziell am Projektanfang fällt dadurch ein hoher zusätzlicher Koordinierungsaufwand an, Wartezeiten bei dem Personal bis zur Zuweisung der Aufgaben entstehen und der Kommunikationsaufwand für Abstimmungen steigt. Damit steigt bei höherem Termindruck bzw. der Komprimierung der Zeitplanung der Aufwand für den Test.
BM	<i>F<sub>11.1.1</sub></i>

---

#### *F<sub>11.1.1</sub>* Wie hoch ist die Ausdehnung der Zeitplanung?

QA	sehr niedrig	niedrig	normal	hoch	sehr hoch
I	$\leq 75\%$	85 %	100 %	130 %	$\geq 160\%$
Der Wert 75 % bedeutet eine Komprimierung der Zeitplanung und damit eine Beschleunigung des Projektes.					
M	Erwartete Ausdehnung der Zeitplanung im Testprojekt mit Bezug zu einer normalen Zeitplanung				

## 5.3. Evaluation des Modells für Testeinflussfaktoren

Zur vollständigen Definition der in den Abschnitten 5.2.4 bis 5.2.14 beschriebenen Testeinflussfaktoren fehlt die Kalibrierung der quantitativen Auswirkung (Eigenschaft *effect* des Testeinflussfaktors) auf die Produktivität. Dabei ist die quantitative Auswirkung für jede Kombination aus Testaktivität und Teststufe zu definieren (siehe Datentyp *Quantitative Rating*), sofern der Testeinflussfaktor eine relevante Auswirkung auf den Aufwand der Testaktivität hat. Dies sind insgesamt  $\leq 99$  Kombinationen bei drei Teststufen, drei Testaktivitäten und insgesamt 11 Testeinflussfaktoren.

Eine ausschließliche Befragung von Experten für die Kalibrierung ist auf Grund dieser hohen Anzahl an Kombinationen nicht möglich. Das zeigte sich auch bei der Vorbereitung der Expertenumfrage (vgl. Abschnitt 5.3.1.1) mit 11 Kombinationen. Die Teilnehmer einer Testumfrage (Pretests) brauchten 20 bis 30 Minuten zur vollständigen Beantwortung.

Auch wenn der Aufwand in eine Expertenbefragung investiert würde, sind nur Tendenzen bzgl. der Auswirkungen ableitbar [Fro09, S. 115]. Hierfür ist der Hauptgrund, dass die Antworten der Experten zu sehr streuen und keine eindeutigen Werte ermittelbar sind. Trotzdem ist eine Expertenbeurteilung erforderlich, da für eine ausschließlich mathematische Kalibrierung statistische Daten in hoher Anzahl vorliegen müssen. Dabei müssen für die 99 Kombinationen jeweils mehrere Datensätze aus Testprojekten vorliegen. Dies ist mit den verfügbaren Daten der Firma CRM IT nicht realisierbar.

Die Ermittlung einer entsprechenden quantitativen Auswirkung von nahezu 99 Kombinationen ist für eine Abdeckung der Teststufen und Testaktivitäten in jedem Fall erforderlich, um verlässliche Aussagen über die Auswirkungen einer Testeinflussfaktors für eine bestimmte Teststufe treffen zu können. Ein Beispiel sind die erhöhten Auswirkungen der Testeinflussfaktoren Kommunikation und Zusammenarbeit bei steigender Teststufe, da der Bedarf an Kommunikation und guter Zusammenarbeit bei einem Abnahmetest höher ist als bei einem (System-)Integrationstest. Genauso verändert sich auch die Bedeutung der Aktivität Testmanagement in den Teststufen, so dass die eine Ermittlung der quantitativen Auswirkung für die Kombinationen notwendig ist.

Um die Ermittlung der quantitativen Auswirkung für die 99 Kombinationen zu ermöglichen, erfolgt eine empirische Untersuchung in zwei Teilen. Der erste Teil besteht

aus einer Expertenumfrage, die Tendenzen der quantitativen Auswirkung aufdecken soll (vgl. Abschnitt 5.3.1). Dabei ist von Experten zu beurteilen, ob die Testeinflussfaktoren einen relevanten Einfluss auf die Testaktivitäten haben. Diese Beurteilung findet ohne Berücksichtigung der Teststufen statt, um die Anzahl der zu bewertenden Kombinationen zu reduzieren. Wenn keine relevante Auswirkung zu erwarten ist, wird die Kombination nicht weiter betrachtet und keine Gewichtung ermittelt.

Im zweiten Teil wird eine empirische Untersuchung über Daten aus abgeschlossenen Testprojekten durchgeführt, um unter zur Hilfenahme der Tendenzen aus der Expertenumfrage die quantitativen Auswirkungen zu bestimmen (vgl. Abschnitt 5.3.2). Abschließend erfolgt in Abschnitt 5.3.3 eine Beurteilung des Modells für Testeinflussfaktoren.

### 5.3.1. Empirische Untersuchung über eine Expertenumfrage

Die Durchführung einer Umfrage zählt nach Prechelt zu den empirischen Forschungsmethoden [Pre01]. Der entscheidende Vorteil der Umfrage ist, dass auf Seiten des Teilnehmers geringer Aufwand entsteht und ein großer Anzahl Teilnehmer gleichzeitig erreicht werden können. Damit ergibt sich, eine sorgfältige Teilnehmerauswahl vorausgesetzt, eine gute Querschnittsübersicht über die Meinungen von Experten im Untersuchungsbereich. Der Nachteil einer Expertenumfrage ist, dass die Verlässlichkeit der Ergebnisse unklar ist [Pre01, S. 45]. Trotzdem eignet sich diese Form der empirischen Untersuchung, um erste Tendenzen bzgl. der Auswirkungen eines Testeinflussfaktors auf die Produktivität abzuleiten<sup>2</sup>.

Die zu untersuchenden Forschungsfragen für die Umfrage lauten:

1. Welche Auswirkung hat ein Testeinflussfaktor durchschnittlich auf die Produktivität in einem Testprojekt?
2. Auf welche Testaktivitäten (Testmanagement, Testdesign und Testausführung) hat der Testeinflussfaktor einen relevanten Einfluss?
3. In welchen Projektphasen (Anforderungs-, Design- und Entwicklungsphase) ist der Testeinflussfaktor gut einzuschätzen?

Die erste Forschungsfrage soll Tendenzen bzgl. der Höhe der Auswirkung auf die Produktivität aufdecken. Darüber können später die Ergebnisse der Untersuchung über

---

<sup>2</sup> „Die Auswertung einer Umfrage erlaubt vorsichtige Rückschlüsse auf die subjektive Wirklichkeit der Teilnehmer in Bezug auf die Forschungsfrage.“ [Pre01, S. 46]

## 5. Entwicklung des Modells für Testeinflussfaktoren

Daten aus abgeschlossenen Testprojekten validiert und ggf. ergänzt werden. Von den Experten soll losgelöst von der Testaktivität und Teststufe die Auswirkung auf die Produktivität in ihren Projekten beantwortet werden.

Mit der zweiten Forschungsfrage wird die Ermittlung der Gewichte über Daten aus abgeschlossenen Testprojekten in Abschnitt 5.3.2.2 unterstützt. Nur die Testaktivitäten, auf die der jeweilige Testeinflussfaktor nach Meinung der Probanden einen relevanten Einfluss hat, werden bei der Ermittlung der Gewichtungen für diesen Testeinflussfaktor berücksichtigt. Die Experten werden hier nach ihrer subjektiven Einschätzung befragt.

Die dritte Forschungsfrage wird für die Validierung der bei den Testeinflussfaktoren angegebenen Phasen genutzt. Im Modell für Testeinflussfaktoren wurde für jeden Testeinflussfaktor festgelegt, ob dieser bei der Testaufwandsschätzung in einer bestimmten Projektphase berücksichtigt werden soll. Auch hier werden die Experten nach ihrer subjektiven Einschätzung gefragt.

Eine Beantwortung der Forschungsfragen in Form einer Umfrage kann schriftlich (Fragebogen) oder mündlich (strukturiertes Interview) erfolgen [Pre01, S. 46]. Um eine größere Teilnehmeranzahl zu erreichen, wurde die Umfrage in Form eines (Online-)Fragebogens realisiert. Die dafür notwendigen Vorbereitungen werden in Abschnitt 5.3.1.1 beschrieben. Anschließend wird das Ergebnis der Umfrage in Abschnitt 5.3.1.2 analysiert.

### 5.3.1.1. Vorbereitung Expertenumfrage

Die Vorbereitung der Expertenumfrage<sup>3</sup> umfasst die Konstruktion des Fragebogens und das Durchführen eines Pretests, um den Fragebogen mit ausgewählten Teilnehmern zu testen. Das Vorgehen zur Konstruktion des Fragebogens orientiert sich an dem Vorgehen, das in den Büchern *Die Fragebogen-Methode* von Mummentey [Mum08, S. 61ff] und *Einführung in die Test- und Fragebogenkonstruktion* von Bühner [Büh11, S. 84ff] beschrieben wird. Dabei sind u. a. die nachfolgenden Schritte auszuführen:

1. Bestimmung der Form des Fragebogens

---

<sup>3</sup>Die Konstruktion des Fragebogens ist in Zusammenarbeit mit Yavuz Sancar durchgeführt worden. Yavuz Sancar forscht im Bereich der Testfreigabebewertung und hat darauf bezogen eine getrennte Gruppe an Items hinzugefügt. Die Probanden haben die Items zusammen mit den Items zur Testaufwandsschätzung beantwortet. Die Beschreibung und Auswertung der Items zur Testfreigabebewertung ist nicht Teil dieser Dissertation und wird daher nicht weiter behandelt.

2. Bestimmung der Itemsammlung<sup>4</sup>
3. Itemrevision
  - Reihenfolge der Items festlegen
  - Art und Anzahl der Antwortmöglichkeiten bestimmen
  - Auswahl und grafisches Design der Antwortskala vornehmen
4. Formulierung der Instruktion für den Fragenbogen
5. Pretest mit einer ausgewählten Gruppe von Teilnehmern

Im ersten Schritt bei der Konstruktion eines Fragebogens muss die Form festgelegt werden. Dabei wird darüber entschieden, wie die Items dem Probanden präsentiert werden sollen. Hier soll der Fragebogen online über das Open-Source-Werkzeug LimeSurvey [Sch] abgebildet. Damit können viele Experten gleichzeitig erreicht werden und sie sind flexibel in der Wahl des Zeitpunktes für die Teilnahme. Als zeitliche Obergrenze wurden 20 Minuten für die Beantwortung der Items gesetzt. 20 Minuten schrecken potentielle Probanden nach unserer Meinung noch nicht ab und können in den Arbeitstag der Probanden integriert werden. Weiterhin besteht bei einer zu hohen Dauer die Gefahr, dass die Motivation der Probanden zur genauen Beantwortung nachlässt. Damit sinkt die Genauigkeit der Itembeantwortung [Büh11].

Weiterhin wurde für die Expertenumfrage allgemein festgelegt, dass die Items als geschlossene Fragen oder als Feststellung formuliert werden sollen, da die Vorteile hinsichtlich der Auswertung und Interpretation der Ergebnisse überwiegen [Mum08, S. 75]. Außerdem können nach den Vorarbeiten in diesem Kapitel die Antwortmöglichkeiten fest vorgegeben werden, so dass eine offene Beantwortung nicht notwendig ist.

Der zweite Schritt zur Konstruktion eines Fragebogens ist die Sammlung von möglichen Items. Dazu wurden zunächst drei Gruppen an Items identifiziert:

1. Bei der Analyse der Ergebnisse muss die Stichprobenzusammensetzung angegeben werden können. Daher ist eine Gruppe mit einleitenden Items notwendig, um allgemeine Informationen über die berufliche Erfahrung des Probanden zu sammeln. Dazu gehören Items zu den wahrgenommenen Projektrollen und dem Unternehmen des Probanden. Nur wenn die bei der Umfrage Beteiligten Experten mit möglichst hoher Berufserfahrung sind, können sie verlässliche Aussagen

---

<sup>4</sup>Ein Item ist eine Frage, ein Begriff oder eine Behauptung, die dem Probanden vorgelegt werden. [Mum08, S. 13]

## 5. Entwicklung des Modells für Testeinflussfaktoren

zu Testaufwandsschätzungen treffen. Außerdem sollten die (Projekt-)Rollen derer, die an der Umfrage teilnehmen, als wesentlich für den Bereich der Qualitätssicherung gelten.

2. Bei der Beschreibung der Stichprobenzusammensetzung ist neben den allgemeinen Informationen, die mit Items aus der ersten Gruppe ermittelt worden sind, die Erfahrung des Probanden mit Testaufwandsschätzungen von wesentlicher Bedeutung. Zur Ermittlung der Erfahrung mit Testaufwandsschätzungen wurde eine zweite Gruppe von Items eingeführt, die sich auf die Erfahrung mit Testaufwandsschätzungen bezieht. Nur wenn die Probanden Erfahrung im Bereich der Testaufwandsschätzung haben, können sie Aussagen zu Auswirkungen von Testeinflussfaktoren auf den Testaufwand treffen.
3. Die direkte Beantwortung der Forschungsfragen aus dem Abschnitt 5.3.1 wird mit einer dritten Gruppe von Items adressiert. Diese Gruppe enthält zu jedem Testeinflussfaktor, der in den Abschnitten 5.2.4 bis 5.2.14 beschrieben worden ist, eine Untergruppe mit Items. Die Items der Untergruppe beziehen sich dann jeweils auf eine der Forschungsfragen aus dem vorherigen Abschnitt. Die Items und Antwortmöglichkeiten sind für jeden Testeinflussfaktor ähnlich zu formulieren, so dass keine Abweichungen in der Beantwortung der Items entstehen.

Anschließend wurden Items für jede Gruppe gesammelt. Dabei wurde die Menge minimal gehalten, da die Beantwortung der Items im Schnitt nicht über 20 Minuten dauern sollte. Für die erste Gruppe wurden insgesamt sechs Fragen aufgenommen (vgl. Abbildungen in Abschnitt A.2.1 und Abbildung A.2) und für die zweite Gruppe drei (vgl. Abbildungen in Abschnitt A.2.2). Wegen der zeitlichen Restriktionen wurden von 11 möglichen nur 10 Testeinflussfaktoren in der dritten Gruppe berücksichtigt, da die Beantwortung der Items für einen Testeinflussfaktor sehr aufwendig ist. Der Testeinflussfaktor Wiederverwendbarkeit aus Abschnitt 5.2.6 wurde nicht in der Umfrage berücksichtigt. Für diesen Testeinflussfaktor wurde angenommen, dass eine Einschätzung gut möglich ist. Mit der Beschränkung auf 10 Testeinflussfaktoren konnte, wie auch der Pretest gezeigt hat, die Zeit von 20 Minuten im Durchschnitt eingehalten werden.

Für jeden Testeinflussfaktor wurden drei Items identifiziert (vgl. Abbildung A.3):

1. Das erste Item ist auf die Auswirkung des Testeinflussfaktors auf den Testaufwand bezogen. Das Ziel ist die Ermittlung eines durchschnittlichen Zahlenwertes für jede qualitative Ausprägung des Testeinflussfaktors, um die erste Forschungsfrage aus dem vorherigen Abschnitt beantworten zu können. Da-

zu musste der Proband auf der (bipolaren) Antwortskala {-40 %, -20 %, -10 %, -5 %, +0,5 %, +10 %, +20 %, +40 %} für jede qualitative Ausprägung des Testeinflussfaktors eine Bewertung vornehmen. Das Vorzeichen *Plus* (+) steht für eine Erhöhung und das *Minus* (-) für eine Verringerung des Testaufwands.

Die Skala für die qualitative Ausprägung eines Testeinflussfaktors wurde für ein besseres Verständnis durch den Probanden in leicht modifizierter Form in den Items verwendet (sehr niedrig → sehr gering/schlecht, niedrig → eher gering/schlecht, hoch → eher hohe/gute, sehr hoch → sehr hohe/gute). Weiterhin wurden die Skalenwerte *außergewöhnlich gering* und *außergewöhnlich hohe/gute* ergänzt, um extreme Tendenzen in beide Wirkungsrichtungen eines Testeinflussfaktors zu erkennen. Der Skalenwert *normal* wurde weggelassen, da dieser für die Tendenzen in der Höhe der Wirkung nicht relevant ist.

Die bipolare Antwortskala des Items bietet den Vorteil, dass das Item sehr kompakt dargestellt werden kann und der Proband die Bewertungen aller qualitativen Ausprägungen auf einer Seite dargestellt bekommt. So kann er die Bewertungen vergleichen und aufeinander abstimmen. Der Nachteil der Antwortskala ist die Anfälligkeit für eine Fehlinterpretation von - und +, die durch geeignete Instruktionen zu minimieren sind. Weiterhin ist bei falschem Verständnis der Skala auch eine Nachbearbeitung der Ergebnisse für einzelne Probanden möglich.

Da eine Antwortskala direkt in numerischer Form mit Bezug zu der Veränderung des Testaufwands gewählt worden ist, kann ein durchschnittlicher Wert berechnet werden. Daraus kann eine Tendenz für die Stärke des Einflusses ermittelt werden und die Forschungsfrage beantwortet werden.

2. Das zweite Item ist auf die Auswirkung des Testeinflussfaktors auf die Testaktivitäten Testmanagement, Testdesign, und Testausführung bezogen. Das Ziel ist die Ermittlung von einem Wahrscheinlichkeitswert je Testaktivität, um die zweite Forschungsfrage aus dem vorherigen Abschnitt beantworten zu können. Dazu müssen die Probanden für die drei Testaktivitäten separat die Wahrscheinlichkeit auf der Skala *keinesfalls–wahrscheinlich nicht–vielleicht–ziemlich wahrscheinlich–ganz sicher* bewerten. Die Begriffe der gewählten Skala sind gleichabständig (vgl. [Roh78, S. 239] und [Mum08, S. 79]). Damit kann die Antwortskala

## 5. Entwicklung des Modells für Testeinflussfaktoren

in quantitative Werte transformiert und angenähert als Intervallskala<sup>5</sup> aufgefasst werden. Daher ist die Berechnung eines Mittelwertes für die Wahrscheinlichkeit zulässig und die Forschungsfrage kann beantwortet werden [Mum08, S. 78][BS10, S. 20].

3. Das dritte Item ist auf die Möglichkeit zur Bewertung des Testeinflussfaktors in den drei Projektphasen Anforderungs-, Design- und Entwicklungsphase bezogen. Das Ziel ist die Ermittlung der durchschnittlichen Häufigkeit je Phase in Prozent, so dass die dritte Forschungsfrage aus dem vorherigen Abschnitt bewertet werden kann. Dazu müssen die Probanden separat für jede Phase die Häufigkeit einer guten Möglichkeit zur Bewertung des Testeinflussfaktors auf der Skala *nie–selten–gelegentlich–oft–immer* vornehmen. Die Begriffe der gewählten Skala sind wieder gleichabständig (vgl. [Roh78, S. 231] und [Mum08, S. 79]) und die Skala kann damit als Intervallskala aufgefasst werden. Somit ist die Berechnung eines Mittelwertes zulässig und die Forschungsfrage kann beantwortet werden.

Im vorletzten Schritt muss die Instruktion für den Fragebogen formuliert werden. Dies wurde anhand der Vorschläge aus dem Buch *Die Fragebogen-Methode* vorgenommen [Mum08, S. 86]. Das Ergebnis ist im Anhang in Abbildung A.1 zu sehen.

Im letzten Schritt wurde ein Pretest mit drei Kollegen aus dem s-lab Software Quality Lab vorgenommen. Dabei wurde noch einmal verifiziert, dass die durchschnittliche Zeit für die Beantwortung aller Items bei 20 Minuten liegt. Ein weiteres Ergebnis war, dass die bipolare Skala unterschiedlich interpretiert wurde: *Plus* (+) als Verringerung und *Minus* (-) als Erhöhung des Testaufwands. Daher wurden die Instruktionen für die Beantwortung des Items erweitert und Hilfetexte ergänzt. Gleichzeitig wurde auch deutlich, dass das Umdrehen der Antworten bei abweichendem Verständnis möglich ist, da einzelne Probanden die Skala konsistent umgekehrt aufgefasst haben.

### 5.3.1.2. Analyse der Ergebnisse der Expertenumfrage

Die Umfrage wurde zunächst über soziale Medien verbreitet. Dazu zählen die Gruppe *Agiles Testmanagement* bei XING [San] und ein Blog zu modellbasiertem Testen [Gü]. Anschließend wurden Kontaktpersonen bei verschiedenen Unternehmen an-

---

<sup>5</sup> „Eine Intervallskala ordnet den Objekten eines empirischen Relativs Zahlen zu, die so geartet sind, dass die Rangordnung der Zahlendifferenzen zwischen je zwei Objekten der Rangordnung der Merkmalsunterschiede zwischen je zwei Objekten entspricht.“ [BS10, S. 20].

gesprochen: arvato AG, corporate quality consulting GmbH, SQS Software Quality Systems AG, Sogeti Deutschland GmbH und s-lab Software Quality Lab. Dadurch haben insgesamt 92 Probanden die Umfrage begonnen. Davon haben 40 Probanden die Umfrage vollständig abgeschlossen. Im Folgenden werden die Ergebnisse zu den in Abschnitt 5.3.1.1 definierten Itemgruppen vorgestellt.

#### **Berufserfahrung des Probanden**

Zur Erlangung eines aussagekräftigen Ergebnisses war es wichtig, dass es sich bei den an der Studie Beteiligten um Experten mit möglichst hoher Berufserfahrung handelte. Außerdem mussten die (Projekt-)Rollen derer, die an der Umfrage teilnahmen, als wesentlich für den Bereich der Qualitätssicherung gelten. Im Rahmen der Umfrage ist beiden Bedingungen hinreichend entsprochen worden (vgl. Abbildung A.9). 15 Probanden waren Testmanager mit mehrjähriger Berufserfahrung. Des Weiteren haben 7 Projekt- bzw. Teamleiter und 8 Testdesigner/Testanalysten/Tester mit mehr als 5 Jahren Berufserfahrung teilgenommen.

Für ein repräsentatives Ergebnis der Umfrage war weiterhin eine möglichst gute Verteilung über unterschiedliche Größenklassen von Projekten sowie deren Dauer notwendig. Dem konnte mit einer relativ gleichen Verteilung über die verschiedenen Größenklassen und Dauern entsprochen werden. (vgl. Abbildungen A.6 und A.7). Nur Projekte ab der Größe von 20 Personenjahren oder Projekte mit einer Dauer von mehr als 2 Jahren waren nicht ausreichend vertreten.

#### **Erfahrung des Probanden mit Testaufwandsschätzungen**

Neben der Berufserfahrung war die direkte Erfahrung der Probanden mit Testaufwandsschätzungen für ein aussagekräftiges Ergebnis wichtig. Für die für diese Dissertation relevanten Teststufen Abnahmetest, Systemtest und (System-)Integrationstest hat fast jeder zweite Proband angegeben, dass er Testaufwandsschätzungen *oft* oder *immer* durchführt. Damit ist ein aussagekräftiges Ergebnis gewährleistet.

Weiterhin ist für ein repräsentatives Ergebnis wichtig, dass möglichst viele Probanden mit Erfahrung im dedizierten Schätzen von Aufwänden für die Qualitätssicherung teilnehmen, da die Items der dritten Gruppe Kenntnisse in diesem Bereich voraussetzen. Als optimal gelten dabei Erfahrungen hinsichtlich der Schätzung einzelner Aktivitäten bei der Qualitätssicherung. Diesen Vorgaben konnte entsprochen werden (vgl. Abbildung A.10). Für mehr als jeden zweiten Probanden gilt, dass der Aufwand für die Qualitätssicherung insgesamt oder sogar für einzelne Maßnahmen der Quali-

## 5. Entwicklung des Modells für Testeinflussfaktoren

tätssicherung bzw. Dokumente geplant wird. Damit schätzt mehr als jeder zweite der Probanden dediziert den Aufwand für die Qualitätssicherung und es haben ausreichend Probanden mit Erfahrung in diesem Bereich an der Umfrage teilgenommen.

### Bewertungen der Testeinflussfaktoren

Im Folgenden werden die Ergebnisse für die Bewertungen der Testeinflussfaktoren vorgestellt. Dazu werden zunächst notwendige statistischen Begriffe definiert und erläutert. Anschließend wird eine Übersicht über die Ergebnisse für die Bewertungen der Testeinflussfaktoren gegeben. Danach werden die Ergebnisse für jeden Testeinflussfaktoren einzeln bewertet. Abschließend werden die Ergebnisse zusammengefasst und ein Fazit präsentiert.

Für die Auswertung der Antworten für die einzelnen Testeinflussfaktoren sind einige statistische Grundlagen notwendig. In Abschnitt 4.2.2 wurden schon das arithmetische Mittel, die Varianz und Standardabweichung eingeführt. Ergänzend wird hier die Berechnung des Medianwerts vorgestellt. Mit den Kennwerten aus Abschnitt 4.2.2 und dem Medianwert können die Antworten hinsichtlich der Forschungsfragen aus Abschnitt 5.3.1 ausgewertet werden. Weiterhin werden Kenntnisse für die Analyse von Häufigkeiten mit dem  $\chi^2$ -Test benötigt. Damit lassen sich Unregelmäßigkeiten in der Verteilung der Antworthäufigkeiten identifizieren, um z. B. starke Unterschiede in den Expertenmeinungen aufzudecken.

#### Medianwert

Als Ergänzung zum arithmetischen Mittel kann der Median einer Menge von  $n$  Werten angegeben werden [BS10, S. 26f]. Hierfür werden die Werte zunächst nach ihrer Größe sortiert. Diese Werte notiert man als

$$x_{(1)}, x_{(2)}, \dots, x_{(n)}.$$

Wobei  $x_{(1)}$  der kleinste und  $x_{(n)}$  der größte Wert aus der gegebenen Menge ist. Anschließend kann über die Vorschrift 5.5 der Median ermittelt werden. Da in die Berechnung nicht jeder Wert unmittelbar einfließt ist der Median stabil gegenüber Ausreißern.

$$Md = \begin{cases} x_{(\frac{n+1}{n})}, & \text{wenn } n \text{ ungerade,} \\ (x_{(\frac{n}{2})} + x_{(\frac{n}{2})} + 1)/2, & \text{wenn } n \text{ gerade.} \end{cases} \quad (5.5)$$

**$\chi^2$ -Test**

Der  $\chi^2$ -Test wird für die Analyse von Häufigkeiten eingesetzt, wenn Objekte entsprechend eines oder mehrerer Merkmale kategorisiert sind [BS10, S. 137]. Eine Variante des  $\chi^2$ -Test kann dabei zur Analyse der Häufigkeiten eines Merkmals genutzt werden, wenn dieses Merkmal  $k$ -fach gestuft ist [BS10, S. 142]. Im Fall des Fragebogens wäre das Merkmal ein Item und die Antwortskala die Stufung.

Für die Analyse der Antworthäufigkeiten mit dem  $\chi^2$ -Test wird die Hypothese  $H_0$  aufgestellt, dass die Antworten für ein Item gleichverteilt sind. Wenn die Antworten gleichverteilt sind, stimmen die Expertenmeinungen wenig bis gar nicht überein. Für diese Hypothese wird mit Formel 5.6 die Prüfgröße  $\chi^2$  berechnet [BS10, S. 143]. Für die Prüfgröße kann mit Berücksichtigung der  $\chi^2$ -Verteilung eine Signifikanz  $p$  berechnet werden. Wenn  $p < 0,05$  gilt, kann die Hypothese  $H_0$  mit mehr als 95 % Wahrscheinlichkeit verworfen werden. Bei  $p \geq 0,05$  gilt die Hypothese mit mehr als 5 % Wahrscheinlichkeit.

$$\chi^2 = \sum_{i=1}^k \frac{b_i - e_i}{e_i} \quad (5.6)$$

**Übersicht über die Ergebnisse der Testeinflussfaktor Bewertungen**

In den Tabellen 5.22, 5.23 und 5.24 sind die Ergebnisse für die Testeinflussfaktoren zusammengefasst. Mit jeder Tabelle wird eine Forschungsfrage aus Abschnitt 5.3.1 adressiert. Die Tabelle 5.22 enthält die Analyseergebnisse der Antworten für die Frage nach der Wahrscheinlichkeit der Auswirkungen der Testeinflussfaktoren auf die Testaktivitäten. Tabelle 5.23 enthält die Analyse der Antworten für die Frage nach der Möglichkeit einer guten Einschätzung der Testeinflussfaktoren in den Projektphasen. Die dritte Tabelle 5.24 fasst die Analyseergebnisse der Antworten für die Frage nach der Höhe der Auswirkungen eines Testeinflussfaktors auf den Testaufwand zusammen.

An einigen Stellen waren die Ergebnisse der Expertenbefragung nicht eindeutig. Diese Stellen sind in den Tabellen 5.22, 5.23 und 5.24 bei einem  $p$ -Wert  $\geq 0,05$  grau hinterlegt. Ergänzend sind die Einträge in den Tabellen 5.22 und 5.23 bei nicht eindeutigen Ergebnissen und einer Standardabweichung  $\geq 1$  grau eingefärbt.

Die Bezeichnungen für die Testaktivitäten Testmanagement (TM), Testdesign (TD) und Testausführung (TA) wurden in Tabelle 5.22 entsprechend gekürzt. In Tabel-

## 5. Entwicklung des Modells für Testeinflussfaktoren

Testeinflussfaktor	Test-aktivität	Md	$\bar{x}$	s	$\chi^2$ -Test p
Qualität der Testbasis	TM	3,00	2,95	0,99	0,07
	TD	4,00	3,80	0,56	0,00
	TA	3,00	3,03	0,89	0,00
Qualitätsziele	TM	4,00	3,18	1,01	0,00
	TD	4,00	3,70	0,72	0,00
	TA	4,00	3,30	0,94	0,00
Qualifikation und Erfahrung	TM	4,00	3,48	0,72	0,00
	TD	4,00	3,50	0,78	0,00
	TA	4,00	3,33	0,80	0,03
Kommunikationsaufwand	TM	4,00	3,60	0,59	0,00
	TD	3,00	2,85	0,98	0,13
	TA	3,00	2,83	1,06	0,31
Zusammenarbeit	TM	3,00	3,30	0,79	0,00
	TD	3,00	3,00	0,75	0,00
	TA	4,00	3,40	0,90	0,00
Stabilität der Testumgebung	TM	3,00	2,93	1,07	0,00
	TD	2,00	1,83	1,13	0,03
	TA	4,00	3,90	0,44	0,00
Testinfrastruktur	TM	3,00	3,05	0,96	0,01
	TD	3,00	2,90	0,90	0,03
	TA	4,00	3,65	0,62	0,00
Gesamttestorganisation	TM	4,00	3,65	0,62	0,00
	TD	3,00	2,88	0,88	0,01
	TA	3,00	3,05	0,81	0,00
Zeitplanung	TM	4,00	3,63	0,74	0,00
	TD	3,00	3,33	0,76	0,00
	TA	4,00	3,58	0,75	0,00

Tabelle 5.22.: Analyse der Antworten für die Frage nach der Wahrscheinlichkeit der Auswirkungen der Testeinflussfaktoren auf die Testaktivitäten

Testeinflussfaktor	Phase	Md	$\bar{x}$	s	$\chi^2$ -Test p
Qualität der Testbasis	A	3,00	2,50	1,01	0,00
	D	3,00	2,93	0,94	0,00
	E	3,00	3,03	0,97	0,00
Qualitätsziele	A	2,00	2,18	1,06	0,01
	D	3,00	2,60	0,90	0,02
	E	3,00	2,83	0,81	0,00
Qualifikation und Erfahrung	A	3,00	2,35	1,05	0,00
	D	3,00	2,93	1,00	0,00
	E	3,00	3,00	0,96	0,00
Kommunikationsaufwand	A	2,50	2,33	0,97	0,00
	D	3,00	2,73	0,93	0,00
	E	3,00	2,83	0,96	0,00
Zusammenarbeit	A	2,00	2,28	1,01	0,01
	D	3,00	2,75	0,81	0,00
	E	3,00	2,95	0,90	0,00
Stabilität der Testumgebung	A	1,00	1,23	0,92	0,00
	D	1,50	1,63	0,98	0,00
	E	2,50	2,43	1,06	0,01
Testinfrastruktur	A	2,00	1,83	1,22	0,01
	D	2,50	2,35	1,14	0,03
	E	3,00	2,83	0,96	0,00
Gesamttestorganisation	A	3,00	2,63	1,00	0,00
	D	3,00	2,83	0,90	0,01
	E	3,00	3,08	0,92	0,00
Dokumentationsbedarf	A	2,50	2,40	0,81	0,00
	D	3,00	2,75	0,71	0,00
	E	3,00	2,95	0,78	0,00
Zeitplanung	A	2,00	1,83	1,06	0,01
	D	2,00	2,38	0,93	0,00
	E	3,00	2,88	0,91	0,00

Tabelle 5.23.: Analyse der Antworten für die Frage nach der Möglichkeit einer guten Einschätzung der Testeinflussfaktoren in den Projektphasen

## 5. Entwicklung des Modells für Testeinflussfaktoren

$TF_i$	Statistik	außerge. niedrig	sehr niedrig	niedrig	hoch	sehr hoch	extra hoch
$TF_1$	Md	40,00	20,00	10,00	-5,00	-10,00	
	$\bar{x}$	31,50	19,50	8,63	-5,00	-12,63	
	$s$	13,31	12,13	6,89	7,25	11,66	
	$\chi^2$ -Test $p$	0,00	0,00	0,56	0,00	0,47	
$TF_2$	Md	-10,00	-10,00	-5,00	10,00	20,00	
	$\bar{x}$	-16,13	-9,13	-4,63	8,75	23,63	
	$s$	13,75	8,23	4,58	6,86	15,40	
	$\chi^2$ -Test $p$	0,17	0,03	0,00	0,36	0,00	
$TF_4$	Md	20,00	10,00	5,00	-5,00	-20,00	
	$\bar{x}$	26,50	15,25	6,00	-7,50	-18,63	
	$s$	12,72	9,20	4,96	7,84	14,05	
	$\chi^2$ -Test $p$	0,00	0,00	0,02	0,01	0,14	
$TF_5$	Md		-10,00	-5,00	5,00	10,00	20,00
	$\bar{x}$		-12,13	-6,25	3,50	10,13	19,75
	$s$		12,90	8,30	3,79	8,51	14,89
	$\chi^2$ -Test $p$		0,10	0,00	0,07	0,03	0,10
$TF_6^*$	Md		-7,50	0,00	10,00	20,00	40,00
	$\bar{x}$		-10,75	-4,25	9,25	19,00	30,50
	$s$		12,07	7,30	7,81	11,45	12,60
	$\chi^2$ -Test $p$		0,40	0,00	0,00	0,00	0,00
$TF_7$	Md	40,00	20,00	10,00	0,00	-7,50	
	$\bar{x}$	34,63	19,63	10,25	-5,75	-12,13	
	$s$	11,40	9,63	7,92	8,81	13,77	
	$\chi^2$ -Test $p$	0,00	0,00	0,03	0,00	0,43	
$TF_8$	Md	40,00	20,00	10,00	-5,00	-10,00	
	$\bar{x}$	32,25	20,38	10,13	-6,25	-15,50	
	$s$	11,87	10,71	7,80	8,38	15,06	
	$\chi^2$ -Test $p$	0,00	0,00	0,03	0,00	0,90	
$TF_9$	Md	40,00	20,00	10,00	-2,50	-10,00	
	$\bar{x}$	28,75	17,50	8,63	-4,50	-10,88	
	$s$	13,43	10,74	6,40	7,32	11,76	
	$\chi^2$ -Test $p$	0,00	0,00	0,22	0,00	0,40	
$TF_{10}$	Md		-20,00	-10,00	5,00	10,00	20,00
	$\bar{x}$		-20,50	-10,25	3,88	10,50	19,63
	$s$		13,29	8,32	4,60	9,04	14,29
	$\chi^2$ -Test $p$		0,00	0,01	0,00	0,07	0,11
$TF_{11}$	Md		-10,00	-5,00	5,00	10,00	20,00
	$\bar{x}$		-14,25	-8,13	6,13	13,75	22,38
	$s$		15,26	10,42	7,55	11,02	15,23
	$\chi^2$ -Test $p$		0,13	0,00	0,00	0,09	0,02

Tabelle 5.24.: Analyse der Antworten für die Frage nach der Höhe der Auswirkungen eines Testeinflussfaktors auf den Testaufwand.

### 5.3. Evaluation des Modells für Testeinflussfaktoren

le 5.23 wurden die Bezeichnungen für die Phasen Anforderungsphase (A), Designphase (D) und Entwicklungsphase abgekürzt.

Die Skala *außergewöhnlich niedrig–sehr niedrig–niedrig–hoch–sehr hoch* in der Tabelle 5.24 steht für die qualitativen Stufen eines Testeinflussfaktors im Modell für Testeinflussfaktoren (vgl. Abschnitt 5.2). Die Abweichungen von der Skala, die für die qualitativen Stufen im Modell für Testeinflussfaktoren beschrieben ist, wurden im Abschnitt 5.3.1.1 vorgestellt.

Die Wirkungsrichtung des Testeinflussfaktors *Zusammenarbeit* war in der Umfrage abweichend von der Richtung im Modell für Testeinflussfaktoren. Zur Korrektur wurde die Skala für die Auswertung umgedreht. Der entsprechende Testeinflussfaktor  $TF_6$  wurde in der Tabelle 5.24 mit einem Stern markiert.

Durch eine fehlerhafte Konfiguration der Umfrage konnten für den Testeinflussfaktor *Dokumentationsbedarf* keine Antworten bzgl. der Testaktivitäten gespeichert werden. Daher fehlt der Testeinflussfaktor in der Tabelle 5.22. Von einer weiteren Befragung der Probanden wird abgesehen, da für den *Dokumentationsbedarf* Auswirkungen auf alle Testaktivitäten zu erwarten sind. Dies wird bei Ermittlung der Gewichte aus Daten von abgeschlossenen Testprojekten validiert.

#### Bewertung des Testeinflussfaktors Qualität der Testbasis

Der Testeinflussfaktor Qualität der Testbasis ist mit hoher Wahrscheinlichkeit für die Testaktivitäten Testdesign und Testausführung relevant. Für die Aktivität Testmanagement ist die Streuung der Antworten sehr groß (vgl.  $p$ -Wert in Tabelle 5.22 und Abbildung A.20). Daher wird der Testeinflussfaktor bzgl. der Testaktivität Testmanagement nicht weiter betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für jeden zweiten Probanden *oft* oder *immer* in allen Projektphasen gegeben (vgl. Abbildung A.21). Für die Anforderungsphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass der Testeinflussfaktor für fast die Hälfte der Probanden nur *selten* oder *gelegentlich* gut in der Anforderungsphase einschätzbar ist. Da die Anzahl der Antworten für die Skalenwerte *oft* und *immer* knapp über der Hälfte liegen, wird die Anforderungsphase für diesen Testeinflussfaktor beibehalten. Hiermit ist auch auch die Definition des Testeinflussfaktors im Modell für Testeinflussfaktoren bestätigt.

## 5. Entwicklung des Modells für Testeinflussfaktoren

Aus der Verteilung der Mediane ist eine deutliche Zweiteilung der Bewertungen für die Auswirkungen des Testeinflussfaktors auf den Testaufwand ablesbar (vgl. Tabelle 5.24 und Abbildung A.19). Eine *außergewöhnlich niedrige* bis *niedrige* Qualität der Testbasis haben die Befragten mit einer Tendenz von +40 % bis +10 % bewertet. Dagegen wurde eine *eher hohe* bis *sehr hohe* Qualität der Testbasis mit nur -5 % bis -10 % bewertet. Daraus ist zu schließen, dass eine hohe Qualität der Testbasis kaum einen Einfluss auf den Testaufwand hat. Im Kontrast dazu kann eine *niedrige* Qualität der Testbasis eine deutliche Erhöhung des Testaufwands bedeuten.

### Bewertung des Testeinflussfaktors Qualitätsziele

Der Testeinflussfaktor Qualitätsziele ist mit hoher Wahrscheinlichkeit für jede Testaktivität relevant. Jeder zweite Proband hat die Wahrscheinlichkeit mit *ganz sicher* bewertet (vgl. Abbildung A.23). Die Streuung der Antworten ist nur für die Testaktivität Testmanagement über der definierten Grenze. Da allerdings jeder zweite Proband die Wahrscheinlichkeit mit *ganz sicher* bewertet hat, wird der Testeinflussfaktor Qualitätsziele auch bzgl. der Testaktivität Testmanagement weiter betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für jeden zweiten Probanden *oft* oder *immer* in der Design- und Entwicklungsphase gegeben (vgl. Abbildung A.24). Für die Anforderungsphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass die Tendenz zu einer *gelegentlich gut* möglichen Einschätzung geht. Daher wird der Testeinflussfaktor nicht in der Anforderungsphase berücksichtigt und die Definition im Modell für Testeinflussfaktoren bestätigt.

Aus der Verteilung der Mediane ist eine schwache Zweiteilung der Bewertung für die Auswirkungen des Testeinflussfaktors auf den Testaufwand ablesbar (vgl. Tabelle 5.24 und Abbildung A.22). *Außergewöhnliche geringe* Qualitätsziele führen nur zu einer leichten Reduktion des Testaufwands. Dagegen haben *sehr hohe* Qualitätsziele eine deutlichere Erhöhung des Testaufwands zur Folge.

### Bewertung des Testeinflussfaktors Qualifikation und Erfahrung

Der Testeinflussfaktor Qualifikation und Erfahrung ist mit hoher Wahrscheinlichkeit für jede Testaktivität relevant. Die Streuung der Antworten ist sehr gering (vgl. *p*-Wert in Tabelle 5.22 und Abbildung A.26). Daher wird der Testeinflussfaktor für alle Testaktivitäten weiter betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für jeden zweiten Probanden *oft* oder *immer* in allen Projektphasen gegeben (vgl. Abbildung A.27). Für die An-

forderungsphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass der Testeinflussfaktor für fast die Hälfte der Probanden nur *selten* oder *gelegentlich* gut in der Anforderungsphase einschätzbar ist. Da die Anzahl der Antworten für die Skalenwerte *oft* und *immer* knapp über der Hälfte liegen, wird die Anforderungsphase für diesen Testeinflussfaktor beibehalten. Damit ist auch die Definition im Modell für Testeinflussfaktoren bestätigt.

Die Probanden haben die Auswirkungen des Testeinflussfaktors Qualifikation und Erfahrung auf den Testaufwand hauptsächlich im Bereich von -20 % bis +20 % bewertet (vgl. Tabelle 5.24 und Abbildung A.25). Die Verteilung der Mediane zeigt im Gegensatz zu den vorausgehend untersuchten Testeinflussfaktoren eine gleiche Verteilung. Eine *sehr hohe* Erfahrung und Qualifikation von Projektmitgliedern hat demgemäß prozentual die gleiche Auswirkung auf den Testaufwand wie eine *aufßergewöhnlich geringe* Erfahrung.

#### Bewertung des Testeinflussfaktors Kommunikationsaufwand

Der Testeinflussfaktor Kommunikationsaufwand ist mit hoher Wahrscheinlichkeit nur für die Testaktivität Testmanagement relevant. Für die Testaktivitäten Testdesign und Testausführung liegt annähernd eine Gleichverteilung der Antworten vor. Dabei entfallen trotzdem mehr als 50 % der Antworten auf die Kategorien *ganz sicher* und *ziemlich wahrscheinlich*. Daher werden alle Testaktivitäten weiterhin bzgl. des Testeinflussfaktors Kommunikationsaufwand betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für mehr als jeden zweiten Probanden *oft* oder *immer* in allen Projektphasen gegeben (vgl. Abbildung A.30). Damit ist auch die Definition im Modell für Testeinflussfaktoren bestätigt.

Dem Testeinflussfaktor Kommunikationsaufwand wird insgesamt weniger Auswirkung auf den Testaufwand zu gerechnet. Zwar gibt es, wie in Abbildung A.28 zu sehen ist, Ausreißer in die Extreme, allerdings liegen die Mediane nicht tiefer als -10 % und nicht höher als +20 % (vgl. Tabelle 5.24). Genauso wie bei den vorherigen Testeinflussfaktoren liegt eine *schiefe* Verteilung vor. Eine negative Ausprägung (Hoher Kommunikationsaufwand) hat damit wesentlich höhere Auswirkungen als eine positive Ausprägung.

#### Bewertung des Testeinflussfaktors Zusammenarbeit

Der Testeinflussfaktor Zusammenarbeit ist mit hoher Wahrscheinlichkeit für jede Testaktivität relevant. Die Streuung der Antworten ist sehr gering (vgl. *p*-Wert in Tabel-

## 5. Entwicklung des Modells für Testeinflussfaktoren

le 5.22 und Abbildung A.32). Daher wird der Testeinflussfaktor für alle Testaktivitäten weiter betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für jeden zweiten Probanden *oft* oder *immer* in der Design- und Entwicklungsphase gegeben (vgl. Abbildung A.33). Für die Anforderungsphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass die Tendenz zu einer *selten* bis *gelegentlich* gut möglichen Einschätzung geht. Daher wird der Testeinflussfaktor nicht in der Anforderungsphase berücksichtigt und die Definition im Modell für Testeinflussfaktoren bestätigt.

Aus der Verteilung der Mediane für die Auswirkung des Testeinflussfaktors auf den Testaufwand ist eine deutliche Zweiteilung der Bewertung ablesbar (vgl. Tabelle 5.24 und Abbildung A.31). Eine schlechte Zusammenarbeit wirkt sich nach Auffassung der Befragten stärker negativ aus, als sich eine gute Zusammenarbeit positiv auswirkt.

### Bewertung des Testeinflussfaktors Stabilität und Verfügbarkeit der Testumgebung

Der Testeinflussfaktor Stabilität und Verfügbarkeit der Testumgebung ist mit hoher Wahrscheinlichkeit für die Testaktivitäten Testmanagement und Testausführung relevant. Zwar ist keine Gleichverteilung der Antworten für die Testaktivität Testdesign zu erwarten, aber die Streuung der Antworten und die Lage des Medians zeigen das der Testeinflussfaktor *wahrscheinlich nicht* bis *vielleicht* relevant für die Testaktivität Testdesign ist. Daher werden nur die Testaktivitäten Testmanagement und Testausführung weiterhin bzgl. des Testeinflussfaktors Stabilität und Verfügbarkeit der Testumgebung betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für fast jeden zweiten Probanden *oft* oder *immer* in der Entwicklungsphase gegeben (vgl. Abbildung A.36). Für die Anforderungs- und Designphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass die Tendenz zu einer *nie* bis *selten* gut möglichen Einschätzung geht. Daher wird der Testeinflussfaktor nur in der Entwicklungsphase berücksichtigt und die Definition im Modell für Testeinflussfaktoren bestätigt.

Bei dem Testeinflussfaktor Stabilität der Testumgebung ergibt sich nach der Lage der Mediane und der geringen Standardabweichung ein klares Bild bzgl. der Erhöhung bei geringer Stabilität 10 %, 20 % und 40 % (vgl. Tabelle 5.24 und Abbildung A.34). *Hohe* und *sehr hohe* Stabilität liegen nach dem Median bei 0–10 % Verringerung Test-

aufwand. In Abbildung A.34 ist ablesbar, dass die Probanden speziell bei geringer Stabilität ähnliche oder gleiche Bewertungen abgegeben haben.

#### Bewertung des Testeinflussfaktors Testinfrastruktur

Der Testeinflussfaktor Testinfrastruktur ist mit hoher Wahrscheinlichkeit für jede Testaktivität relevant. Die Streuung der Antworten ist sehr gering (vgl. *p*-Wert in Tabelle 5.22 und Abbildung A.38). Daher wird der Testeinflussfaktor für alle Testaktivitäten weiter betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für mehr als jeden zweiten Probanden *oft* oder *immer* in der Entwicklungsphase gegeben (vgl. Abbildung A.39). Für die Anforderungs- und Designphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass die Tendenz zu einer *nie* bis *gelegentlich* gut möglichen Einschätzung geht. Daher wird der Testeinflussfaktor nur in der Entwicklungsphase berücksichtigt und die Definition im Modell für Testeinflussfaktoren bestätigt.

Dem Testeinflussfaktor Testinfrastruktur wird insgesamt weniger Auswirkung auf den Testaufwand zugerechnet. Zwar liegt der Median für eine *außergewöhnlich schlechte* Infrastruktur bei +40 %, allerdings ist z. B. an dem arithmetischen Mittel eine Tendenz zur Mitte zu sehen (vgl. Tabelle 5.24 und Abbildung A.37). Auch verringert eine gute Testinfrastruktur den Testaufwand nach der Bewertung der Probanden kaum.

#### Bewertung des Testeinflussfaktors Gesamttestorganisation

Der Testeinflussfaktor Gesamttestorganisation ist mit hoher Wahrscheinlichkeit für jede Testaktivität relevant. Die Streuung der Antworten ist sehr gering (vgl. *p*-Wert in Tabelle 5.22 und Abbildung A.41). Daher wird der Testeinflussfaktor für alle Testaktivitäten weiter betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für mehr als jeden zweiten Probanden *oft* oder *immer* in allen Projektphasen gegeben (vgl. Abbildung A.30). Damit ist auch die Definition im Modell für Testeinflussfaktoren bestätigt.

Aus der Verteilung der Mediane ist eine deutliche Zweiteilung der Bewertung ablesbar (vgl. Tabelle 5.24 und Abbildung A.40). Eine schlechte Gesamttestorganisation wirkt sich nach Auffassung der Befragten stärker negativ aus, als sich eine gute Gesamttestorganisation positiv auswirkt.

## 5. Entwicklung des Modells für Testeinflussfaktoren

### Bewertung des Testeinflussfaktors Dokumentationsbedarf

Wie vorher dargestellt, konnten für den Testeinflussfaktor Dokumentationsbedarf keine Antworten bzgl. der Testaktivitäten gespeichert werden. Daher wird angenommen, dass der Testeinflussfaktor eine Auswirkung auf alle Testaktivitäten hat. Dies wird entsprechend bei der Ermittlung der Gewichte validiert.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für jeden zweiten Probanden *oft* oder *immer* in der Design- und Entwicklungsphase gegeben (vgl. Abbildung A.33). Für die Anforderungsphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass die Tendenz zu einer *selten* bis *gelegentlich* gut möglichen Einschätzung geht. Daher wird der Testeinflussfaktor nicht in der Anforderungsphase berücksichtigt und die Definition im Modell für Testeinflussfaktoren bestätigt.

Die Probanden rechnen dem Testeinflussfaktor Dokumentationsbedarf einen mittleren Einfluss auf den Testaufwand zu. Dies ist an der Tendenz zu dem Bewertungsbereich -20 % bis +20 % zu erkennen (vgl. Tabelle 5.24 und Abbildung A.43).

### Bewertung des Testeinflussfaktors Zeitplanung

Der Testeinflussfaktor Zeitplanung ist mit hoher Wahrscheinlichkeit für jede Testaktivität relevant. Die Streuung der Antworten ist sehr gering (vgl. *p*-Wert in Tabelle 5.22 und Abbildung A.46). Daher wird der Testeinflussfaktor für alle Testaktivitäten weiter betrachtet.

Die Wahrscheinlichkeit für eine gute Einschätzung ist für jeden zweiten Probanden *oft* oder *immer* in der Design- und Entwicklungsphase gegeben (vgl. Abbildung A.33). Für die Anforderungsphase ist an dem arithmetischen Mittel und der Standardabweichung zu erkennen (vgl. Tabelle 5.23), dass der Schwerpunkt bei einer *selten* bis *gelegentlich* gut möglichen Einschätzung liegt. Daher wird der Testeinflussfaktor nicht in der Anforderungsphase berücksichtigt und die Definition im Modell für Testeinflussfaktoren bestätigt.

Die Verteilung der Mediane für den Testeinflussfaktor Zeitplanung bzgl. der Auswirkungen auf den Testaufwand liegen im Bereich von -10 % bis 20 % (vgl. Tabelle 5.24 und Abbildung A.45). Damit wird dem Testeinflussfaktor Zeitplanung ein mittlerer Einfluss auf den Testaufwand zugerechnet. Dabei wirkt sich ein geringer Termindruck weniger stark auf den Testaufwand aus, als ein hoher Termindruck.

## Fazit

Die Forschungsfragen aus dem Abschnitt 5.3.1 konnten alle beantwortet werden. Es sind für 10 von 11 Testeinflussfaktoren Tendenzen in der Höhe des Einflusses auf den Testaufwand identifiziert worden. Die Tendenzen werden zusammenfassend in der Tabelle 5.24 dargestellt und können schnell anhand der Mediane und Mittelwerte abgelesen werden. Weiterhin ist für jeden Testeinflussfaktor unter Berücksichtigung der Abbildungen aus Abschnitt A.2.3 noch mal eine dedizierte Bewertung vorgenommen worden.

Die Kombinationen aus Testeinflussfaktoren und Testaktivitäten, die eine Gewichtung erfordern, konnten eingeschränkt werden. Für den Testeinflussfaktor Qualität der Testbasis wird die Testaktivität Testmanagement nicht mehr betrachtet. Bei dem Testeinflussfaktor Stabilität der Testumgebung muss die Testaktivität Testdesign nicht mehr berücksichtigt werden.

Die Annahmen aus dem Modell für Testeinflussfaktoren hinsichtlich der Möglichkeit zur guten Einschätzung in den unterschiedlichen Projektphasen konnten alle bestätigt werden (vgl. Tabelle 5.23). Mit Unterstützung der Abbildungen im Anhang A.2.3 kann schnell ein Überblick über die Einschätzungen der Probanden gewonnen werden.

### **5.3.2. Empirische Untersuchung über Daten aus abgeschlossenen Projekten**

In dem zweiten Teil der Evaluation wird eine empirische Untersuchung über Daten aus abgeschlossenen Testprojekten durchgeführt. Mit dem Ergebnis sollen unter zur Hilfenahme der Tendenzen aus der Expertenumfrage die quantitativen Auswirkungen der Testeinflussfaktoren bestimmt werden. Die empirische Untersuchung über abgeschlossene Projekte wird analog dem Vorgehen für das Modell für Testkomplexität (vgl. Abschnitt 4.2.1) vorgenommen, d. h. es wird dem Prinzip der Software-Archäologie gefolgt. Dies hat auch hier den Vorteil, dass kein Eingriff in ein Testprojekt notwendig gewesen ist und auch länger zurückliegende Testprojekte berücksichtigt werden können.

Die Grundlage für die empirische Untersuchung bilden dieselben Testprojekte des Unternehmens CRM IT, die bei der Evaluation des Modells für Testkomplexität verwendet worden sind (vgl. Abschnitt 4.2.1). Da die Testprojekte rückwirkend betrachtet

## 5. Entwicklung des Modells für Testeinflussfaktoren

werden, sind für die Testprojekte jeweils alle Testeinflussfaktoren bewertbar. Dies entspricht dem Setzen der Projektphase *V* auf *Entwicklungsphase* (vgl. auch Vorlage für die Beschreibung der Testeinflussfaktoren in Tabelle 5.10 und die Einträge für *V* in den Tabellen für die Beschreibung der Testeinflussfaktoren in den Unterabschnitten zu Abschnitt 5.2).

Zunächst wird in Abschnitt 5.3.2.1 beschrieben, wie die in Abschnitt 4.2.1.1 eingeführte TAQ Anwendung zur Datensammlung genutzt worden ist. Anschließend erfolgt in Abschnitt 5.3.2.2 die Zusammenführung mit den Ergebnissen der Expertenumfrage und die Bestimmung der quantitativen Auswirkung für jede qualitative Ausprägung der Testeinflussfaktoren getrennt nach Teststufe und Testaktivität.

### 5.3.2.1. Sammlung von Daten aus abgeschlossenen Testprojekten

Die große Anzahl an abgeschlossenen Testprojekten erforderte eine strukturierte werkzeuggestützte Verwaltung der Daten, um Berechnungen auf den gesammelten Daten zur ermöglichen. Daher wurde die TAQ Anwendung entsprechend erweitert, um projektspezifische Bewertungen der Testeinflussfaktoren eingeben zu können. Die eingegebenen Bewertungen werden im relationalen Teil der TAQ Datenbank gespeichert und ermöglichen somit komplexe Berechnungen auf Datenbankebene (vgl. Übersicht über die Werkzeugunterstützung in Abbildung 4.12).

Die notwendigen Informationen zur Bewertung der Testeinflussfaktoren bzw. der Beantwortung der zugehörigen Fragen wurden entweder aus den Testmitteln gewonnen, die in der Ordnerstruktur der TAQ Datenbank abgelegt sind, oder durch Gespräche mit den Beteiligten des Testprojektes ermittelt. Insgesamt sind für 151 Testprojekte die Testeinflussfaktoren bewertet worden. Das Ergebnis ist in Tabelle B.4 im Anhang zusammengefasst. In der Tabelle sind für jedes Testprojekt die Bewertungen der Testeinflussfaktoren auf der qualitativen Skala von *sehr niedrig* bis *sehr hoch* aufgeführt, wobei die Bewertungen aus Platzgründen über die bijektive Funktion aus Formel 5.4 transformiert worden sind.

Genauso wie bei den gesammelten Daten für das Modell für Testkomplexität ist die Validität der Daten für die nachfolgenden Berechnungen von zentraler Bedeutung. Daher wurden mögliche Risiken identifiziert und Maßnahmen zur Risikominimierung getroffen. Für die folgenden Punkte wird die Validität näher betrachtet, da diese die Grundlage für die Anwendung des Modells für Testeinflussfaktoren sind:

### *5.3. Evaluation des Modells für Testeinflussfaktoren*

- Validität der Testmittel und anderer zur Verfügung stehenden Informationen über das Testprojekt
- Validität der Antworten auf die Fragen zur Bewertung der Testeinflussfaktoren

Die Validität der Testmittel ist für das Modell für Testeinflussfaktoren hoch, wenn aus ihnen Informationen abgeleitet werden können, die zur Beantwortung der Fragen für die Bewertung der Testeinflussfaktoren genutzt werden können. Dabei gibt es das Risiko, dass die Testmittel falsch interpretiert und falsche Antworten auf Fragen erfasst werden. Das Risiko ist hier höher als beim Modell für Testkomplexität, da der Kontext oder Rahmenbedingungen eines Testprojektes bei der Firma CRM IT implizit erfasst worden sind. Um dem zu begegnen wurde an vielen Stellen auf eigene Erfahrungen in den Testprojekten zurückgegriffen. Weiterhin wurden fehlende Informationen in Gesprächen mit Projektbeteiligten ergänzt. Mit der eigenen Erfahrung und den ergänzenden Gesprächen konnten in den meisten Fällen ausreichend verlässliche Informationen gesammelt werden, da die Personalfliktuation der Abteilung QM über den betrachteten Zeitraum sehr gering war und in vielen Testprojekten eine eigene Beteiligung vorgelegen hat. Testprojekte, über die keine ausreichenden Informationen zur Verfügung standen, wurden nicht in die TAQ Datenbank aufgenommen.

Von der objektiven Beantwortung der Fragen hängt die Validität der Bewertung der Testeinflussfaktoren ab. Umso größer die Objektivität bei der Beantwortung, desto höher ist die Validität. Eine wichtige Voraussetzung für die Objektivität ist, dass die Fragen ohne Einflussnahme des Unternehmens beantwortet werden. Dies war während der gesamten Zeit gewährleistet.

Insgesamt kann die Validität der gesammelten Testprojektdaten bestätigt werden. Daher können auf Basis dieser Daten und den Ergebnissen der Expertenumfrage in Abschnitt 5.3.2.2 die quantitativen Auswirkungen der Testeinflussfaktoren bestimmt werden.

#### **5.3.2.2. Zusammenführung Ergebnisse der Expertenumfrage mit Daten aus abgeschlossenen Testprojekten**

In diesem Abschnitt wird beschrieben, wie die quantitativen Auswirkungen der Testeinflussfaktoren aus den Ergebnissen der Expertenumfrage und den gesammelten Projektdaten berechnet wird. Im Kern ist eine quantitative Auswirkung ein Gewicht einer qualitativen Ausprägung eines Testeinflussfaktors. Mit dem Gewicht wird ausgedrückt, um wie viel Prozent sich der Testaufwand z. B. bei einer niedrigen Bewertung

## 5. Entwicklung des Modells für Testeinflussfaktoren

des Testeinflussfaktors *Qualität der Testbasis* erhöht. Diese Gewichte müssen separat für jede Teststufe aufgeschlüsselt nach Testmanagement, Testdesign und Testausführung für alle qualitativen Ausprägungen der Testeinflussfaktoren bestimmt werden.

Als Grundlage werden im Folgenden zunächst die mathematischen Grundlagen über die Technik Regressionsanalyse zusammengefasst. Anschließend wird auf Basis dieser Grundlagen ein Überblick über den Einsatz der Regressionsanalyse im Modell für Testeinflussfaktoren für die Bestimmung der Gewichte der Testeinflussfaktoren gegeben. Darauf folgt die tiefere Erläuterung der Evaluation am Beispiel der Teststufe Systemtest.

### Mathematischer Exkurs zur Technik Regressionsanalyse

Die Regressionsanalyse ist eine Technik, um die Beziehung zwischen mindestens zwei Merkmalen zu untersuchen [BS10, S. 183ff]. Mit dem Wissen über diesen Zusammenhang kann mit einer entsprechenden Regressionsgleichung bei Kenntnis über die Ausprägung eines Merkmals die zugehörige Ausprägung des anderen Merkmals berechnet werden. Bei der einfachen linearen Regression wird der Zusammenhang der Merkmale  $x$  und  $y$  als Regressionsgerade beschrieben (vgl. Formel 5.7). Der Fehler, der sich aus der Differenz der beobachteten Merkmalsausprägung und der vorhergesagten Ausprägung aufgrund der Regressionsgeraden ergibt, wird als Residuum bezeichnet. Die Regressionsgerade ist dann die Gerade, die die Summe der quadrierten Residuen minimiert.

Die Variable  $x$  in Formel 5.7 wird dabei als Prädiktorvariable,  $y$  als Kriteriumsvariable und  $a$  sowie  $b$  als Regressionskoeffizienten bezeichnet. Im Allgemeinen sind Prädiktorvariablen leichter zu messen und können zur Vorhersage einer schwerer zu ermittelten Kriteriumsvariable genutzt werden. Mit der multiplen linearen Regression wird die einfache lineare Regression erweitert, so dass mehrere Prädiktorvariablen<sup>6</sup> gleichzeitig berücksichtigt werden [BS10, S. 339ff]. In Formel 5.8 ist die entsprechende Regressionsgleichung dargestellt. Die  $b_i$  stellen die Regressionskoeffizienten dar, die zugehörigen  $x_i$  die Prädiktorvariablen und  $\hat{y}$  die Kriteriumsvariable. Genauso wie bei der einfachen linearen Regression sollen die  $x_k$  einfacher zu messen sein und dann mit der Regressionsgleichung zur Vorhersage von  $\hat{y}$  eingesetzt werden.

---

<sup>6</sup> „In einer multiplen Regression mit  $k$  Prädiktoren gibt die Steigung  $b_j$  die erwartete Veränderung des Kriteriums  $y$  an, die einer Erhöhung des  $j$ -ten Prädiktors um eine Einheit entspricht, wenn die anderen  $k - 1$  Prädiktoren konstant sind.“ [BS10, S. 346]

$$y = a + b \cdot x \quad (5.7)$$

$$\hat{y} = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_k \cdot x_k \quad (5.8)$$

Die Voraussetzung für die Anwendung der Technik lineare Regression sind nach [BS10, S. 348]:

### **Linearität**

Die Abhängigkeit zwischen den Erwartungswerten des Kriteriums und den Prädiktorwerten sind durch die Regressionsgleichung 5.7 bzw. 5.8 bestimmt.

### **Homoskedastizität**

Die Varianz der  $y$ -Werte ist für alle Kombination von Prädiktorwerten konstant, d. h. der Vorhersagefehler verändert sich nicht mit kleinen oder großen Werten in der Stichprobe. Dies lässt sich an einem Streudiagrammen über die Residuen nach Durchführung der Regressionsanalyse zeigen [BS10, S. 197].

### **Normalität**

Die Verteilung der  $y$ -Werte ist für jede Kombination von Prädiktorwerten eine Normalverteilung, d. h. im Fall der einfachen linearen Regression sind die meisten beobachteten Werte in der Nähe der Regressionsgeraden und verteilen sich für ein festes  $x$  analog einer Normalverteilung um die Regressionsgerade. Dies lässt sich nach Durchführung der Regressionsanalyse z. B. in einem Histogramm über die (standardisierten) Residuen überprüfen [BS10, S. 198].

### **Stichprobenumfang**

Der Stichprobenumfang  $n$  darf nicht kleiner sein als die Anzahl der Prädiktoren.

### **Lineare Abhängigkeit der Prädiktoren**

Es darf keine lineare Abhängigkeit unter den Prädiktoren existieren.

## **Die Technik Regressionsanalyse im Modell für Testeinflussfaktoren**

Für das Modell für Testeinflussfaktoren wurde die Technik multiple lineare Regression eingesetzt, um Gewichte für die Testeinflussfaktoren zu bestimmen. Der Ausgangspunkt für die Regressionsanalyse ist die Formel 5.9, die sich aus Formel 3.8 ableitet. Dabei steht PT für den Testaufwand, K für Testkomplexität, P für Produktivität und  $TF_i$  für Testeinflussfaktor  $i$  aus  $k$  Testeinflussfaktoren. Hier wird bewusst von Formel 3.8 abstrahiert, da die Gleichungen für alle Teststufen und Testaktivitäten gleich aufgebaut sind.

## 5. Entwicklung des Modells für Testeinflussfaktoren

In Abschnitt 4.2.2 wurde für das Modell für Testkomplexität gezeigt, dass ein linearer Zusammenhang zwischen dem Testaufwand und der Testkomplexität besteht, wenn beide Werte logarithmiert werden. Daher wird auf die Formel 5.9 der Logarithmus angewendet mit dem Ergebnis in Formel 5.10. Weiterhin wird  $\log(\frac{K}{P})$  auf die linke Seite gezogen, da kein Regressionskoeffizient für die Testkomplexität und die Produktivität berechnet werden soll. Wenn nun die Regressionsanalyse angewendet wird, werden die Regressionskoeffizienten  $b_i$  für alle  $\log(TF_i)$  und eine Konstante  $b_0$  berechnet (vgl. Formel 5.12). Das Ergebnis entspricht damit der Formel 5.8 der multiplen Regressionsanalyse.

$$PT = \frac{K}{P} \cdot \prod_{i=1}^k TF'_i \quad (5.9)$$

$$\log(PT) = \log\left(\frac{K}{P}\right) + \sum_{i=1}^k \log(TF'_i) \quad (5.10)$$

$$\log(PT) - \log\left(\frac{K}{P}\right) = \sum_{i=1}^k \log(TF'_i) \quad (5.11)$$

### Tiefere Erläuterung der Evaluation am Beispiel der Teststufe Systemtest

In Abbildung 5.4 ist der vollständige Ablauf der Evaluation dargestellt. Grundlegende Bestandteile sind die wiederholte Durchführung einer Regressionsanalyse mit Kreuzvalidierung. Nach  $n$ -maliger Wiederholung wird die nach Vorhersagegenauigkeit beste quantitative Auswirkung ausgewählt und ausgegeben. Dabei sollte  $n$  so gewählt werden, dass die durchschnittliche Vorhersagegenauigkeit stabil über alle berechneten Gewichtungen ist. Im Folgenden wird der Ablauf aus Abbildung 5.4 schrittweise erläutert. Begonnen wird dabei mit der Überprüfung der Voraussetzungen für die Anwendung der Technik Regressionsanalyse.

Die im mathematischen Exkurs beschriebenen Voraussetzung für die Anwendung der Technik lineare Regression wurden über die Ausgaben der verwendeten Statistikumgebungen R<sup>7</sup> und SPSS<sup>8</sup> geprüft. Für die Teststufe Systemtest konnten die Voraussetzungen bestätigt werden. Dagegen waren für den (System-)Integrationstest und den

---

<sup>7</sup>R ist eine freie Statistikumgebung, die u. a. die Ausführung der Technik multiple lineare Regression unterstützt, als auch die Prüfungen auf die Voraussetzungen [Pro].

<sup>8</sup>IBM SPSS Statistics ist eine Software zur Analyse von statistischen Daten der Firma IBM, die genauso wie die Statistikumgebung R die multiple lineare Regression unterstützt [IBM].

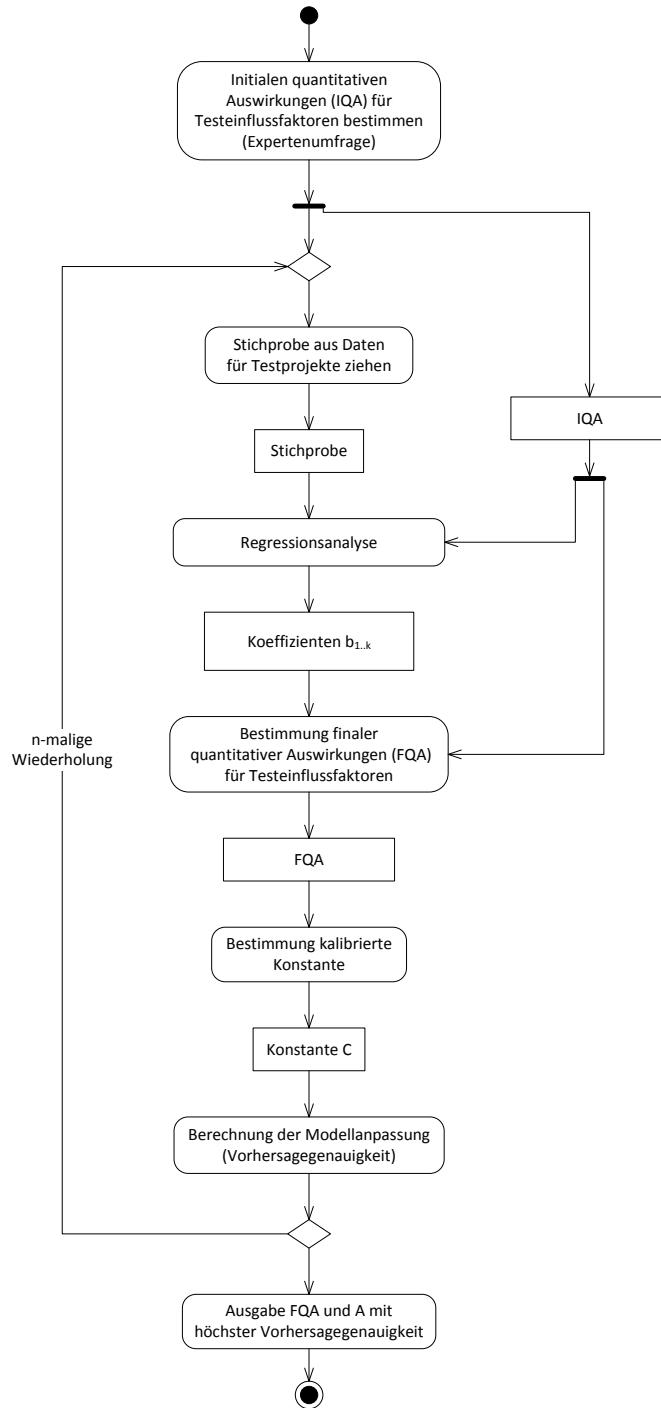


Abbildung 5.4.: Ablauf der Zusammenf hrung der Expertenumfrage mit den Daten aus den abgeschlossenen Testprojekten

## 5. Entwicklung des Modells für Testeinflussfaktoren

Abnahmetest nicht genügend Datensätze vorhanden (vgl. Tabelle 4.4). Daher wird im Folgenden exemplarisch der Systemtest betrachtet.

	sehr niedrig	niedrig	normal	hoch	sehr hoch
Qualität der Testbasis	1,20	1,09	1,00	0,95	0,87
Qualitätsziele	0,91	0,95	1,00	1,09	1,23
Wiederverwendbarkeit von Testmitteln			1,00	0,75	0,50
Erfahrung des Personals	1,15	1,06	1,00	0,93	0,76
Kommunikationsaufwand		0,94	1,00	1,04	
Zusammenarbeit der Projektmitglieder	0,89	0,95	1,00	1,09	1,19
Stabilität und Verfügbarkeit der Testumgebung	1,20	1,10	1,00	0,94	0,88
Testinfrastruktur	1,20	1,10	1,00	0,94	0,85
Gesamttestorganisation	1,18	1,09	1,00	0,95	0,89
Grad des Dokumentationsbedarfs	0,80	0,90	1,00	1,04	1,11
Zeitplanung und Termine	1,13	1,06	1,00	0,92	0,86

Tabelle 5.25.: Initialen quantitativen Auswirkungen (IQA), die aus den Ergebnissen der Expertenumfrage abgeleitet worden ist

Für die Regressionsanalyse müssen die Variablen der Formel 5.11 mit Daten belegt werden. Die Daten für die Variablen PT, K und P können der empirischen Untersuchung für das Modell für Testkomplexität entnommen werden (vgl. Abschnitt 4.2.1). Für die Variablen  $TF_i$  werden die Daten genutzt, die im vorherigen Abschnitt für jedes Testprojekt erfasst worden sind. Dabei ist eine Transformation der qualitativen Ausprägung auf der Skala von *sehr niedrig* bis *sehr hoch* in ein Format notwendig, das für die Regressionsanalyse genutzt werden kann.

Für die Transformation der qualitativen Ausprägungen wird das gleiche Verfahren wie bei COCOMO II angewendet, wobei aus Expertenmeinungen die initialen quantitativen Auswirkungen der Testeinflussfaktoren (IQA) je qualitativer Ausprägung (QA) bestimmt werden [Boe00, S.166ff]. Dazu wird das arithmetische Mittel der Expertenmeinungen zur quantitativen Auswirkung der jeweiligen qualitativen Ausprägung von den Testeinflussfaktoren (vgl. Tabelle 5.24) genutzt. Damit die verwendeten qualitativen Ausprägungen zu denjenigen im Modell für Testeinflussfaktoren passen, wird die Stufe *normal* mit der neutralen quantitativen Auswirkung 1 ergänzt und auf die extrem Positionen *außergewöhnlich niedrig* und *extra hoch* verzichtet.

### 5.3. Evaluation des Modells für Testeinflussfaktoren

Bei den IQA wird nicht zwischen Testaktivitäten unterschieden, da die Expertenmeinungen sich immer auf den Gesamttestaufwand und nicht auf eine einzelne Testaktivität bezogen haben. Die daraus resultierenden IQA sind in Tabelle 5.25 aufgeführt. Damit sind die Werte der Variablen aus Formel 5.11 bestimmt und bilden je Testprojekt einen Datensatz für die Regressionsanalyse.

Zur Überprüfung der Stabilität der Regressionsanalyse wird die Kreuzvalidierung angewendet. Dabei wird eine Stichprobe aus den verfügbaren Datensätzen gezogen und darauf die Regressionsanalyse angewendet. Das Ergebnis der Regressionsanalyse wird zur Vorhersage der Kriteriumsvariable bzw. des Testaufwands, der nicht in der Stichprobe enthaltenden Datensätze, verwendet. Anhand der Genauigkeit der Vorhersage wird anschließend das Ergebnis beurteilt (vgl. [BS10, S. 356]). Die Größe der Stichprobe wurde auf 50 % der Gesamtzahl der Datensätze gesetzt.

Aus dem Ergebnis der Regressionsanalyse in Form der berechneten  $b_i$  wurde mit den IQA in zwei Schritten die finalen quantitativen Auswirkungen (FQA) berechnet. Im ersten Schritt wurde auf die Formel 5.11 die Exponentialfunktion  $e$  als Umkehrfunktion zum Logarithmus angewendet. Das Ergebnis der Umformung ist in Formel 5.13 zu sehen.

Danach wurde in einem zweiten Schritt eine zufällige Gewichtung  $g$  mit  $0 \leq g \leq 1$  gewählt und mit Formel 5.14 für jeden Testeinflussfaktor je qualitativer Ausprägungen die FQA berechnet. Umformuliert bedeutet dies, dass die IQA zu  $g\%$  in die FQA einfließen. Dies folgt dem Beispiel aus COCOMO II, wo die Expertenmeinungen in einem vereinfachten Modell zu 90 % in der endgültigen Gewichtung der Einflussfaktoren berücksichtigt werden, da die Ergebnisse der Regressionsanalyse nicht aussagekräftig genug waren [Boe00]. Speziell werden kontraintuitive quantitative Auswirkungen vermieden, d. h. das z. B. eine hohe Qualität der Testbasis zu einem erhöhten Testaufwand führt. Ähnlich wie bei COCOMO II wurden solche Ergebnisse bei einer ausschließlichen Berücksichtigung des Ergebnisses der Regressionsanalyse zur Bestimmung der FQA beobachtet [Boe00].

$$\log(PT) - \log\left(\frac{K}{P}\right) = b_0 + \sum_{i=1}^k b_i \cdot \log(TF'_i) \quad (5.12)$$

$$PT = e^{b_0} \cdot \frac{K}{P} \cdot \prod_{i=1}^k TF'^{b_i} \quad (5.13)$$

$$TF_i = (1 - g) \cdot TF'^{b_i} + g \cdot TF'_i \quad (5.14)$$

## 5. Entwicklung des Modells für Testeinflussfaktoren

Die berechneten FQA für die Testaktivität Testdesign sind in Tabelle 5.27 aufgeführt. Entsprechende Tabellen sind für die Testaktivitäten Testmanagement und Testausführung im Anhang C.1 zu finden.

Die Berechnung der FQA nach Formel 5.14 erfordert eine Neukalibrierung der Konstante  $b_0$ , da die Regressionskoeffizienten nicht direkt übernommen werden, sondern mit den IQA verrechnet werden. Dazu wird die Formel 5.15 genutzt, die auf Formel 5.13 basiert. Das Ergebnis ist die Konstante  $C$ . Die berechneten Werte für die Konstante sind in Tabelle 5.26 getrennt nach den Testaktivitäten aufgeführt.

$$\varnothing(\log(PT) - \log(\frac{K}{P} \cdot \prod_{i=1}^k TF_i)) = C \quad (5.15)$$

Konstante $C$	
Testmanagement	0,89
Testdesign	0,77
Testausführung	1,13

Tabelle 5.26.: Konstante  $C$

Mit den FQA wurde anschließend die Modellanpassung in Form der Vorhersagegenauigkeit berechnet (vgl. Tabelle 5.28). Dazu wurden zunächst die 50 % der Datensätze verwendet, die nicht zur Regressionsanalyse genutzt worden sind und anschließend alle Datensätze. Nach der Speicherung des Ergebnisses der Vorhersagegenauigkeit und der FQA sowie der Konstante  $C$  wurde der gesamte Ablauf mit einer neuen zufälligen Stichprobe aus den Datensätzen wiederholt. Diese Schleife wurde nach n-maliger Wiederholung beendet und die beste FQA zusammen mit der Konstante  $C$  ausgegeben. Dabei sind die besten FQA diejenige mit der höchsten Vorhersagegenauigkeit für alle Datensätze.

Anhand der durchschnittlichen Vorhersagegenauigkeit über alle FQA wurde die Validität des Ergebnisses untersucht. Die Validität ist dabei hoch, wenn die durchschnittliche Vorhersagegenauigkeit über alle Datensätze nah an der Vorhersagegenauigkeit über die Datensätze ist, die nicht in der Stichprobe für die Regressionsanalyse enthalten gewesen sind. Die Einträge in Tabelle 5.28 zeigen, dass dies für alle Testaktivitäten gegeben ist. Die Abweichungen zum jeweils besten Modell erklären sich durch Ausreißer in den Testprojektdaten. Speziell bei der Testaktivität Testdesign führt dies zu

	sehr niedrig	niedrig	normal	hoch	sehr hoch
Qualität der Testbasis	1,69	1,28	1,00	0,87	0,69
Qualitätsziele	0,78	0,87	1,00	1,27	1,80
Wiederverwendbarkeit von Testmitteln			1,00	0,65	0,37
Qualifikation und Erfahrung	1,14	1,06	1,00	0,94	0,78
Kommunikationsaufwand		0,74	1,00	1,23	
Zusammenarbeit	0,89	0,95	1,00	1,09	1,20
Stabilität und Verfügbarkeit der Testumgebung					
Testinfrastruktur	2,45	1,57	1,00	0,77	0,52
Gesamttestorganisation	2,84	1,68	1,00	0,76	0,55
Grad des Dokumentationsbedarfs	0,74	0,87	1,00	1,05	1,15
Zeitplanung und Termine	1,04	1,02	1,00	0,98	0,96

Tabelle 5.27.: Berechnete FQA je qualitativer Ausprägung der Testeinflussfaktoren für die Testaktivität Testdesign. Dabei wurden die IQA zu 30 % berücksichtigt.

dem deutlichen Unterschied zwischen den durchschnittlichen Werten und dem besten Modell. Hier zeigt sich die Stärke des gewählten Vorgehens für die Evaluation, da auch für die Testaktivität Testdesign FQA gefunden werden konnten, bei denen 47 % der Testprojekte innerhalb von 30 % des tatsächlichen Aufwands geschätzt werden konnten.

### 5.3.3. Beurteilung des Modells

Das Modell für Testeinflussfaktoren konnte mit der zweistufigen Evaluation bestätigt werden. Dabei wurden in der Expertenumfrage die Testeinflussfaktoren allgemein bestätigt und erste Tendenzen für die Höhe der Auswirkungen auf den Testaufwand aufgezeigt. Die darauf aufbauende empirische Untersuchung über Daten aus abgeschlossenen Testprojekten belegte und konkretisierte die Tendenzen an realen Projektdaten des Unternehmens CRM IT.

Für die empirische Untersuchung über abgeschlossene Testprojekte wurde das Modell in mehr als 120 Fällen angewendet. Dies zeigt, dass die notwendigen Informationen zur Bewertung der Testeinflussfaktoren in den Testprojekten vorhanden sind und

## 5. Entwicklung des Modells für Testeinflussfaktoren

	Durchschnitt			Bestes Modell		
	Innerhalb ... des tatsächlichen Aufwands			Innerhalb ... des tatsächlichen Aufwands		
	10 %	20 %	30 %	10 %	20 %	30 %
Testmanagement (Alle)	15 %	30 %	43 %	19 %	31 %	48 %
Testmanagement (Nicht in Stichprobe)	14 %	28 %	40 %	13 %	29 %	47 %
Testdesign (Alle)	13 %	24 %	35 %	9 %	32 %	47 %
Testdesign (Nicht in Stichprobe)	12 %	23 %	33 %	9 %	16 %	41 %
Testausführung (Alle)	9 %	17 %	28 %	9 %	16 %	26 %
Testausführung (Nicht in Stichprobe)	8 %	17 %	25 %	8 %	13 %	28 %

Tabelle 5.28.: Durchschnittliche Vorhersagegenauigkeit von 10.000 berechneten Modellen und die höchste erreichte Vorhersagegenauigkeit eines Modells je Testaktivität

genutzt werden können. Gleichzeitig bewiesen die Gespräche für die Bewertung der Testeinflussfaktoren die Anwendbarkeit des Modells für Testeinflussfaktoren.

Expertenbasierte Schätzung			
Innerhalb ... des tatsächlichen Aufwands			
	10 %	20 %	30 %
Testmanagement	17 %	29 %	41 %
Testdesign	19 %	27 %	48 %
Testausführung	9 %	14 %	30 %

Tabelle 5.29.: Vorhersagegenauigkeit für die Teststufe Systemtest des expertenbasierten Ansatzes des Unternehmens CRM IT

Der direkte Vergleich mit dem bestehenden expertenbasierten Vorgehen (vgl. Abschnitt 3.1.4) belegt, dass mit der Kombination aus dem Modell für Testeinflussfaktoren und dem Modell für Testkomplexität vergleichbare Ergebnisse<sup>9</sup> erzielt werden (vgl. Tabellen 5.28 und 5.29). Speziell die Vorhersagegenauigkeit der Testaktivität

<sup>9</sup>Für den Vergleich konnte nur eine Teilmenge der Testprojekte betrachtet werden, da nicht für alle Testprojekte expertenbasierte Schätzungen durchgeführt oder nachträglich auffindbar gewesen sind. Im Anhang ist in Tabelle B.5 aufgeführt, für welche Testprojekte Daten aus expertenbasierten Schätzungen vorlagen.

## *5.4. Hinweise zur organisationsspezifischen Kalibrierung*

Testmanagement ist deutlich besser, während die Genauigkeit für Testdesign und Testausführung nur leicht nach unten abweichen.

Basierend auf den Erkenntnissen wird angenommen, dass für die Teststufen (System-)Integrationstest und Abnahmetest bei mehr verfügbaren Datensätzen ähnliche Ergebnisse erzielt werden können. Allgemein sind für alle Teststufen steigende Vorhersagegenauigkeiten bei einer steigenden Anzahl von Datensätzen zu erwarten, da dann die Ergebnisse der Regressionsanalyse belastbarer sind.

## **5.4. Hinweise zur organisationsspezifischen Kalibrierung**

Für den Einsatz des Modells für Testeinflussfaktoren in einem neuen Kontext ist eine organisationsspezifische Kalibrierung vorzunehmen, da für die berechneten quantitativen Auswirkungen mit den Daten des Unternehmens CRM IT keine Allgemeingültigkeit besteht. Dabei können entweder die Konstante  $C$  der Formel 3.8 oder die quantitativen Auswirkungen neu bestimmt werden. Darauf wird im nächsten Abschnitt näher eingegangen. Weiterhin können auch Testeinflussfaktoren hinzugefügt oder entfernt werden. Dazu werden die Anforderungen und die Auswirkungen in Abschnitt 5.4.2 diskutiert.

### **5.4.1. Nutzung von Daten aus abgeschlossenen Testprojekten**

Im optimalen Fall stehen für eine Kalibrierung des Modells für Testeinflussfaktoren so viele Datensätze zur Verfügung, dass die quantitativen Auswirkungen, die in diesem Kapitel beschriebenen Testeinflussfaktoren, nach dem Verfahren in Abschnitt 5.3.2 neu bestimmt werden können. Allerdings stehen vielen Unternehmen diese Datensätze nicht zur Verfügung und auch wenn sie Verfahren zur Sammlung der Daten implementieren brauchen, dauert es in der Regel mehrere Jahre, um genügend Datensätze zu sammeln. Trotzdem können auch mit wenigen Projektdaten Kalibrierungen vorgenommen werden. Dies ist über die Konstante  $C$  in der Formel 3.8 möglich. Hierfür kann das Verfahren aus Abschnitt 5.3.2.2 für die Bestimmung der Konstante über die Formel 5.15 verwendet werden.

## 5. Entwicklung des Modells für Testeinflussfaktoren

Die Vorhersagegenauigkeit wird neben dem Modell für Testeinflussfaktoren auch durch das Modell für Testkomplexität bestimmt. Die Projektdaten, die für die Bestimmung der durchschnittlichen Produktivität notwendig sind, sind in vielen Fällen leichter zu ermitteln. Daher reicht die Berechnung einer durchschnittlichen Produktivität, die auch geschätzt werden kann und die Kalibrierung der Konstante  $C$  zunächst für Vorhersagen des Testaufwands aus.

Für weitere Kalibrierungen muss ein Programm zur Sammlung von Projektdaten und zu darauf basierenden Schätzungen initiiert werden. Dies kann z. B. analog dem Prozess nach [BD08, S. 87] geschehen. Allerdings gibt ist allgemein vor der Einführung eines (formalen) Schätzprogramms zu beachten, dass die Entwicklung der Software nach standardisierten Prozessen innerhalb des Unternehmens verläuft und Kennzahlen (z. B. Aktivität und Aufwand) zu den Projekten gleichartig erfasst werden [BD08, S. 94].

### 5.4.2. Hinzufügen und Entfernen von Testeinflussfaktoren

Das Hinzufügen von Testeinflussfaktoren sollte in einem Prozess aus drei Schritten erfolgen. Der Prozess ähnelt dabei dem des Modells für Testeinflussfaktoren (vgl. Abbildung 5.1). Zunächst sind Informationen über mögliche neue Testeinflussfaktoren zu sammeln. Dazu können Testmanager zu abgeschlossenen oder auch laufenden Testprojekten befragt werden. Anschließend müssen die Testeinflussfaktoren spezifiziert und mögliche qualitative Ausprägungen der Testeinflussfaktoren bestimmt werden. Danach sind die quantitativen Auswirkungen über eine oder mehrere empirische Untersuchungen festzulegen.

Das Hinzufügen von Testeinflussfaktoren setzt im optimalen Fall immer die Nutzung von Daten aus abgeschlossenen Testprojekten voraus, da somit u. a. eine objektive Bestimmung der Höhe der quantitativen Auswirkungen gewährleistet ist. Trotzdem kann bei fehlenden Daten auch ein Vorgehen ausschließlich auf der Basis von Expertenbefragungen gewählt werden.

Testeinflussfaktoren können einfach aus dem Modell für Testeinflussfaktoren entfernt werden, da keine Abhängigkeiten unter den Testeinflussfaktoren bestehen. Damit sind außer dem Herausnehmen aus der Bewertung und Berechnung des Testaufwands keine weiteren Schritte notwendig.

## 5.5. Zusammenfassung

In diesem Kapitel wurde das Modell für Testeinflussfaktoren beschrieben. Dabei wurden zunächst über eine zweistufige Literaturrecherche allgemeine und testspezifische Einflussfaktoren identifiziert und anschließend zu Testeinflussfaktoren synthetisiert. Mit diesen Testeinflussfaktoren wurde dann das Modell für Testeinflussfaktoren aufgebaut und die Testeinflussfaktoren im Detail beschrieben.

Mit dem Modell für Testeinflussfaktoren ist ein Basisset an Testeinflussfaktoren definiert worden, die die nahezu unbegrenzte Menge der Einflussfaktoren auf den Testaufwand auf 11 einschränkt. Da diese 11 Testeinflussfaktoren auf der Basis einer Analyse von bekannten Methoden zur (Test-)Aufwandsschätzung und Expertenbefragungen entstanden sind, decken sie mit hoher Wahrscheinlichkeit die wichtigsten Bereiche ab. Damit ist über die identifizierten Testeinflussfaktoren ein erster Konsens über Einflussfaktoren auf den Testaufwand möglich, die nach dem jeweiligen Unternehmenskontext kalibriert werden können.

Das Modell für Testeinflussfaktoren zeigt, wie Testeinflussfaktoren beschrieben werden müssen, damit sie durch unterschiedliche Personen gleich verwendet werden können. Dies beginnt mit der Definition der qualitativen Skala von *sehr niedrig* bis *sehr hoch*. Die darauf aufbauende Verknüpfung mit Metriken und Indikatoren ermöglicht eine einheitliche Bewertung der Testeinflussfaktoren über unterschiedliche Testprojekte und Personen hinweg. Damit steigen die Vergleichbarkeit von Testprojekten und die Nachvollziehbarkeit der Testaufwandsschätzungen.

Die Bestimmung der Höhe des Einflusses der Testeinflussfaktoren in Form der quantitativen Auswirkung gibt einen Eindruck, welche Einflussfaktoren besonders in den Testprojekten wirken. Da hierfür zahlreiche Experten an der Umfrage zur empirischen Untersuchung teilnahmen und reale Daten aus abgeschlossenen Testprojekten zur Verfügung standen, ist hier speziell für die Teststufe Systemtest eine belastbare Aussage möglich gewesen. Für die anderen Teststufen (System-)Integrationstest und Abnahmetest werden bei der Verfügbarkeit von weiteren Daten aus abgeschlossenen Testprojekten ähnliche Bewertungen möglich sein.

Für die Teststufe Systemtest sind nun in der Kombination aus dem Modell für Testkomplexität und dem Modell für Testeinflussfaktoren über Formel 3.8 Vorhersagen über den Testaufwand möglich. Dies wird in Kapitel 6 noch einmal zusammenfassend mit Werkzeugunterstützung vorgestellt.



# 6. Die Methode TAQ

In den Kapitel 4 und 5 wurde schrittweise die Methode TAQ als Lösung für die Problemstellung aus Abschnitt 1.1 entwickelt. Mit der Methode TAQ ist der Testaufwand aufgeschlüsselt nach Testaktivitäten berechenbar. Dabei wird insbesondere auf die Testziele und Testobjekte jeder Teststufe eingegangen.

In diesem Kapitel wird in Abschnitt 6.1 die Methode TAQ zusammengefasst. Dazu wird die Basisformel 3.8 der Methode TAQ aus dem Lösungsansatz in Abschnitt 3.2 aufgegriffen und gezeigt, wie über die Formel der Testaufwand vorhersagbar ist. Da die Evaluation der Modelle für Testkomplexität und Testeinflussfaktoren den Fokus auf die Stufe Systemtest hatte, erfolgen die Erläuterungen in diesem Kapitel auch an der Teststufe Systemtest. Dabei wird analog dem Aufbau der Dissertation zunächst auf die Bestimmung der Testkomplexität eingegangen (vgl. Kapitel 4). Die Testkomplexität steht für die Größe eines Testprojektes und bestimmt maßgebend den Testaufwand. Anschließend wird das Modell für Testeinflussfaktoren erläutert (vgl. Kapitel 5), über das zusätzliche Einflussfaktoren bei der Berechnung des Testaufwands berücksichtigt werden.

Für die Unterstützung der Durchführung einer Testaufwandsschätzung über die Methode TAQ wurde eine Werkzeugunterstützung entwickelt. Diese wird in Abschnitt 6.2 im Detail vorgestellt.

## 6.1. Die Testaufwandsschätzung mit der Methode TAQ

In Abbildung 6.1 ist eine Übersicht über die Ziele der Methode TAQ in Form von Use Cases dargestellt. Der zentrale Use Case ist die Schätzung des Testaufwands. Die Ausführung des Use Cases erfordert die Schätzung der Testkomplexität und die Bewertung der Testeinflussfaktoren. Der beteiligte Akteur ist der Schätzer oder dessen Spezialisierung der Testmanager. Im Folgenden wird zunächst die Schätzung der

## 6. Die Methode TAQ

Testkomplexität mit der Methode TAQ und danach die Bewertung der Testeinflussfaktoren erläutert. Anschließend wird die Formel zur Testaufwandsschätzung vorgestellt. Da für die Teststufe Systemtest in den vorangegangenen Kapiteln 4 und 5 eine detaillierte Evaluation durchgeführt worden ist, erfolgen die Erläuterungen am Beispiel der Teststufe Systemtest.

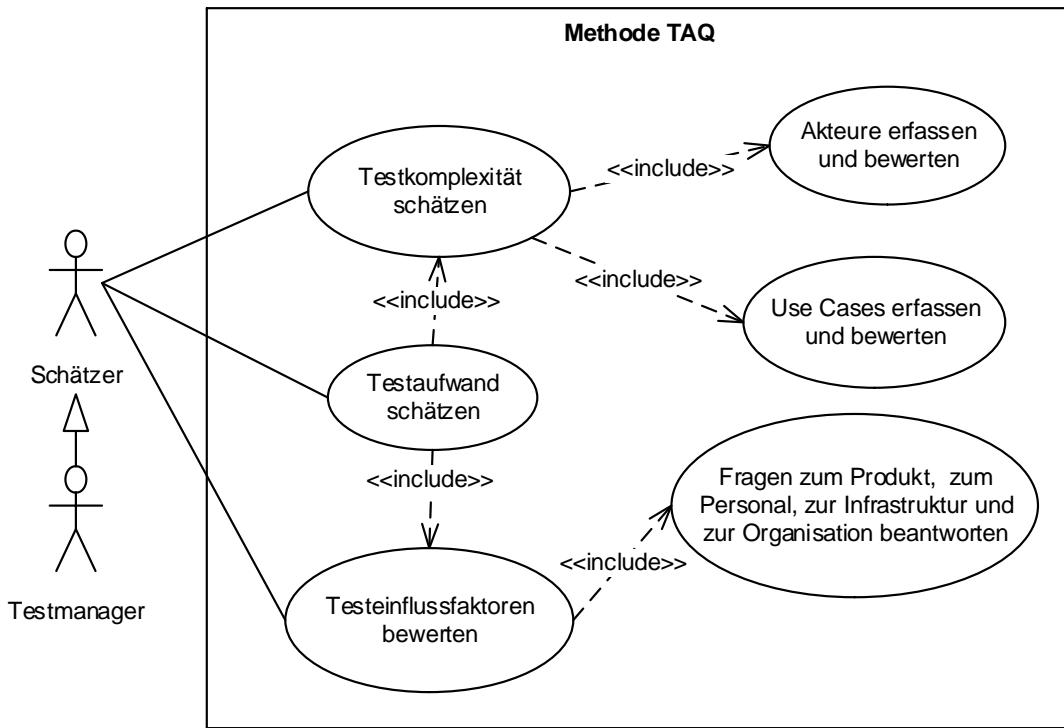


Abbildung 6.1.: Übersicht über Use Cases der Methode TAQ

Die Testkomplexit\u00e4t  $K$  f\u00fcr die zu testenden Testobjekte ist die Grundlage f\u00fcr eine Testaufwandssch\u00e4tzung mit der Methode TAQ. Dabei erfolgt die Sch\u00e4tzung der Testkomplexit\u00e4t  $K$  f\u00fcr den Systemtest in zwei Schritten, die in Abbildung 6.1 als separate Use Cases dargestellt sind. Die Grundlage f\u00fcr beide Schritte ist die Testbasis, die die funktionale Spezifikation des zu entwickelnden Softwareproduktes enth\u00e4lt. Aus dieser Basis werden im ersten Schritt durch den Sch\u00e4tzer die beteiligten Akteure aufgenommen und klassifiziert. Anschlie\u00e3end sind die zu testenden Use Cases zu identifizieren und zu bewerten.

Die Klassifikation der Akteure erfolgt nach dem Schema, das mit dem Modell f\u00fcr Testkomplexit\u00e4t zur Einstufung von Akteuren definiert worden ist. Dabei werden Akteure nach der erwarteten Anzahl der Schritte in Testf\u00e4llen gewichtet, die zur Abbildung der Kommunikation mit diesem Akteur notwendig sind. Die Gewichtung

## 6.1. Die Testaufwandsschätzung mit der Methode TAQ

erfolgt im Intervall von 1 – 3, wobei z. B. menschliche Akteure immer mit 3 bewertet werden. Die Regeln zur Einstufung sind in Tabelle 4.2 nachzulesen.

Für die zu testenden Use Cases sind vom Schätzer jeweils die Anzahl der Aktionen und die Anzahl der möglichen Erfolgs- und Fehlerszenarien zu ermitteln. Da durch weitere Anforderungen, die sich auf den Use Case beziehen, zusätzliche Testfälle erforderlich sein können, müssen diese auch je Use Case aufgenommen werden. In Abschnitt 4.1.2 wurden mit dem *Nabe-Speiche*-Modell mögliche Quellen für derartige Anforderungen gegeben.

Nach der Klassifikation der Akteure und der Bewertung der Use Cases durch den Schätzer ist die Testkomplexität  $K$  über die Formel 6.1 schätzbar. Die unter der Formel angegebenen Erläuterungen der Parameter sind noch einmal der Hinweis, welche Kennzahlen speziell bei der Bewertung der Use Cases durch den Schätzer erhoben werden müssen.

$$K = \sum_{m=1}^{AA} (GA_m) + \sum_{n=1}^{AUC} \left( AS_n + \left\lfloor \frac{1}{3} \cdot AAU_n \right\rfloor + AAF_n \right) \quad (6.1)$$

$AA$  = Anzahl Akteure

$GA_m$  = Gewicht von Akteur  $m$

$AUC$  = Anzahl Use Cases

$AS_n$  = Anzahl Szenarien für Use Case  $n$

$AAU_n$  = Anzahl Aktionen für Use Case  $n$

$AAF_n$  = Anzahl Anforderungen für Use Case  $n$

Im Anschluss an die Schätzung der Testkomplexität werden Testeinflussfaktoren auf der Skala *sehr niedrig* bis *sehr hoch* bewertet. Dazu sind Fragen in den Kategorien Produkt, Personal, Infrastruktur und Organisation zu beantworten. Die Anzahl  $k_{A,V}$  der zu bewertenden Testeinflussfaktoren und damit auch der zu beantwortenden Fragen hängt von der zu schätzenden Testaktivität  $A$  und der Projektphase  $V$  zum Zeitpunkt der Testaufwandsschätzung ab. In Tabelle 6.1 sind die empfohlenen Testeinflussfaktoren für die entsprechenden Testaktivitäten und Projektphasen aufgeführt. Dabei sind die Angaben für die Projektphasen als Mindestmenge an zu bewertenden Testeinflussfaktoren zu interpretieren, d. h. je nach der Verfügbarkeit von Informationen über den Kontext des Testprojektes können auch mehr Testeinflussfaktoren einbezogen werden.

## 6. Die Methode TAQ

	Phase V			Testaktivität A		
	AP	DP	EP	TM	TD	TA
Qualität der Testbasis	X	X	X		X	X
Qualitätsziele		X	X	X	X	X
Wiederverwendbarkeit von Testmitteln		X	X	X	X	X
Qualifikation und Erfahrung		X	X	X	X	X
Kommunikationsaufwand	X	X	X	X	X	X
Zusammenarbeit		X	X	X	X	X
Stabilität und Verfügbarkeit der Testumgebung			X	X		X
Testinfrastruktur				X	X	X
Gesamtprojektorganisation	X	X	X	X	X	X
Grad des Dokumentationsbedarfs		X	X	X	X	X
Zeitplanung und Termine		X	X	X	X	

Tabelle 6.1.: Übersicht über die Anwendbarkeit der Testeinflussfaktoren in Projektphasen (Anforderungsphase (AP), Designphase (DP), Entwicklungsphase (EP)) und für Testaktivitäten (Testmanagement (TM), Testdesign (TD) und Testausführung (TA))

Die bestimmte Testkomplexität und die bewerteten Testeinflussfaktoren sind nun durch den Schätzer in die Formel 6.2 zu überführen, um den Testaufwand in Personentagen (PT) zu bestimmen. Dabei wird der Wert für Testkomplexität als natürliche Zahl  $K$  direkt übernommen, während für die Testeinflussfaktoren jeweils eine Transformation der Bewertung auf der Skala *sehr niedrig* bis *sehr hoch* in eine reelle Zahl  $TF_{A_i}$  vorgenommen werden muss. Die jeweiligen Zahlenwerte für die Stufe Systemtest sind den Tabellen C.1 (Testmanagement), 5.27 (Testdesign) und C.2 (Testausführung) zu finden.

$$PT_A = C_A \cdot K \cdot \frac{1}{P_A} \cdot \prod_{i=1}^{k_{A,V}} TF_{A_i} \quad (6.2)$$

$$A \in \{\text{Testmanagement, Testdesign, Testausführung}\}$$

$$V \in \{\text{Anforderungsphase, Designphase, Entwicklungsphase}\}$$

$$(6.3)$$

Anschließend an das Überführen der Testkomplexität und der Testeinflussfaktoren sind die Werte der Konstante  $C_A$  und der Produktivität  $P_A$  in Abhängigkeit von der Testaktivität  $A$  zu wählen. Für die Teststufe Systemtest wurden die Werte für  $C_A$  für

die Evaluation des Modells für Testeinflussfaktoren bestimmt (vgl. Abschnitt 5.3.2.2 Tabelle 5.26). Entsprechend wurden die Werte für die globale Produktivität  $P_A$  während der Evaluation des Modell für Testkomplexität berechnet (vgl. Abschnitt 4.2.1.2 Tabelle 4.5).

Nach der Belegung aller Parameter der Formel 6.2 ist der Testaufwand in Personentagen (PT) schätzbar. Ein PT entspricht dabei einem Arbeitstag von 8 Stunden. Wenn für mehrere Testaktivitäten der Testaufwand geschätzt wird, muss die Formel ggf. mit unterschiedlichen Testeinflussfaktoren, Konstanten  $C$  und Produktivität  $P_A$  erneut angewendet werden. Hierbei können die bestimmte Testkomplexität  $K$  und schon vorhandene Bewertungen von Testeinflussfaktoren wiederverwendet werden. Zur Unterstützung und Beschleunigung des Vorgehens wird im nächsten Abschnitt die Werkzeugunterstützung vorgestellt.

## 6.2. Werkzeugunterstützung für die Methode TAQ

Zur Unterstützung der Berechnung des Testaufwands mit der Methode TAQ wurde entsprechende Werkzeugunterstützung entwickelt. Diese wurde in die TAQ Anwendung integriert, um die Ergebnisse der Evaluation auf Basis der Projektdaten direkt nutzen zu können. Weiterhin konnten Teile der TAQ Anwendung, die für die Ausführung der Evaluation benötigt wurden, für die Vorhersage des Testaufwands wieder verwendet werden. Die TAQ Anwendung wurde ohne Lokalisierung für die deutsche Sprache entwickelt, daher enthalten die in diesem Kapitel gezeigten Bildschirmfotos englische Begriffe. Für eine bessere Verknüpfung zu den Inhalten der vorangegangen Kapitel wurden die konfigurierbaren Inhalte der TAQ Anwendung in Deutsch hinterlegt. Im Folgenden wird zunächst darauf eingegangen, wie das Modell für Testeinflussfaktoren mit der TAQ Anwendung konfiguriert werden kann und anschließend wird die Nutzung der TAQ Anwendung zur Durchführung einer Testaufwandsschätzung vorgestellt.

### 6.2.1. Konfiguration des Modells für Testeinflussfaktoren

Während der Durchführung der Evaluation für das Modell für Testeinflussfaktoren wurde deutlich, dass immer wieder Anpassungen am Modell für Testeinflussfaktoren notwendig werden. Da sich die Rahmenbedingungen unter denen Testprojekte ausgeführt werden stetig weiterentwickeln, sind auch in der weiteren geplanten Nutzung

## 6. Die Methode TAQ

des Modells für Testeinflussfaktoren wiederholte Anpassungen sehr wahrscheinlich notwendig. Daher ist ein großer Teil des Modells für Testeinflussfaktoren direkt über die graphische Benutzeroberfläche konfigurierbar. Dies betrifft das Hinzufügen und das Entfernen von Testeinflussfaktoren inklusive der Eingabe der Bewertungsmodelle, dem Kalibrieren der Regressionskoeffizienten, dem Einstellen der Projektphasen und der relevanten Testaktivitäten. Weiterhin können die zugehörigen Testeinflussfaktoren Attribute, Fragen und Indikatoren hinterlegt werden. In Abbildung 6.2 ist die Einstiegsmaske für die Konfiguration des Modells für Testeinflussfaktoren zu sehen. Die gezeigten Tabellen bieten die Möglichkeiten Einträge hinzuzufügen und zu entfernen. Damit können dynamisch Testeinflussfaktoren und die zugehörigen Attribute, Fragen und Indikatoren angepasst werden.

Zur Konfiguration des Modells für Testeinflussfaktoren gehören auch Anpassungen der Bewertungsmodelle. In Abbildung 6.2 ist in der Spalte für das Bewertungsmodell (*Rating Model*) in den oberen beiden Tabellen zu sehen, wie ein Bewertungsmodell für einen Testeinflussfaktor und Attribute angegeben werden. Am Testeinflussfaktor ist Bewertungsmodell  $\min(\#11, \#12)$  hinterlegt, d. h. nehme die niedrigste qualitative Bewertung der Attribute mit den Ids 11 und 12 als Gesamtbewertung für den Testeinflussfaktor. Um komplexere Bewertungsmodelle abbilden zu können (vgl. Testeinflussfaktor Testinfrastruktur in Abschnitt 5.2.11), können Bewertungsmodelle mehrzeilig und mit Bedingungen versehen sein. Ein Beispiel für eine Zeile mit Bedingung ist *if #27 = 3 then min(#25, #26)*. Hier wird geprüft, ob das Ergebnis des Bewertungsmodells von Attribut mit Id 27 mit *normal (3)* (vgl. Funktion *S* in Formel 5.4 für die Übersetzung der Skala in Zahlenwerte) ist. Wenn ja, wird der Teil nach *then* ausgewertet. Für die Auswertung mehrzeiliger Bewertungsmodelle werden die Bedingungen zeilenweise geprüft und die erste Zeile mit erfüllter Bedingung für die Bewertung verwendet. Bei der Bestimmung der qualitativen Gesamtbewertung des Testeinflussfaktor wird ein *bottom-up* Vorgehen verwendet, d. h. es werden zunächst alle Bewertungsmodelle der Testeinflussfaktor Attribute angewendet und abschließend das Bewertungsmodell des Testeinflussfaktors.

Über die Einstiegsmaske in Abbildung 6.2 sind weitere Masken zur Konfiguration des Modells für Testeinflussfaktoren erreichbar. Zwei Beispiele sind in Abbildung 6.3 zu sehen. Auf der linken Seite ist der Dialog zum Einstellen der Projektphasen abgebildet, in denen der Testeinflussfaktor angewendet werden soll. Rechts in der Abbildung ist die Maske zum Kalibrieren der Regressionskoeffizienten dargestellt. Darüber hat der Benutzer die Möglichkeit die aktuell hinterlegten Koeffizienten in Abhängigkeit von der Testaktivität und Teststufe einzusehen und zu verändern.

## 6.2. Werkzeugunterstützung für die Methode TAQ

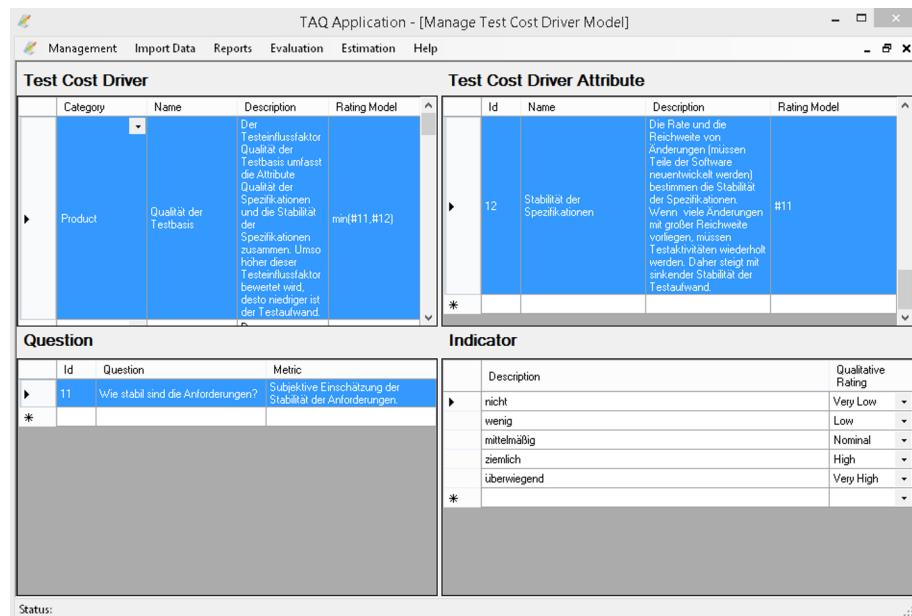


Abbildung 6.2.: Konfiguration des Modells für Testeinflussfaktoren

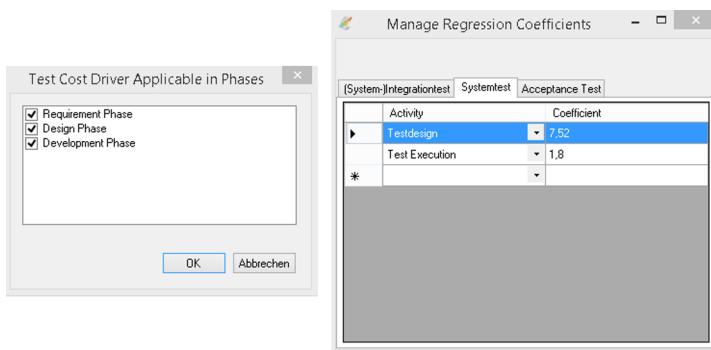


Abbildung 6.3.: Kalibrieren der Regressionskoeffizienten und Einstellen der Projektphasen für den Testeinflussfaktor Qualität der Testbasis

Neben der graphischen Benutzeroberfläche sind weitere Einstellungsmöglichkeiten direkt in der TAQ Datenbank möglich, da z. B. die initialen quantitativen Auswirkungen (vgl. Ergebnis der Expertenumfrage in Abschnitt 5.3.1.2) in einer Datenbanktabelle hinterlegt sind. Dieser Konfigurationsweg wurde für alle Elemente gewählt, die sehr selten geändert werden müssen.

Mit den in diesem Abschnitt beschriebenen Mitteln ist das Modell für Testeinflussfaktoren zur großen Teilen direkt über die graphische Benutzeroberfläche konfigurierbar und für Elemente mit seltenerem Änderungsbedarf über die Datenbank. Damit waren Änderungen während der Evaluation schnell möglich und das Modell für Testeinflussfaktoren ist in der Zukunft leicht anpassbar.

### 6.2.2. Durchführung einer Testaufwandsschätzung

Nachdem das Modell für Testeinflussfaktoren konfiguriert worden ist, können über die TAQ Anwendung Testaufwandsschätzungen durchgeführt werden. In Abbildung 6.4 ist das Vorgehen zur Erstellung einer Testaufwandsschätzung als Aktivitätsdiagramm dargestellt. Die ersten beiden Aktivitäten sind das Anlegen eines neuen Softwareprojektes und eines Testprojektes. Ein Softwareprojekt dient als logische Klammer um alle Testprojekte, die zu derselben Software durchgeführt werden. Testprojekte beziehen sich immer auf einer bestimmte Version<sup>1</sup> der Software. Dabei können mehrere Teststufen zu einer Version gehören und in einem Testprojekt zusammengefasst werden. Für das Softwareprojekt und das Testprojekt werden bei der Anlage Stammdaten, wie ein Name, ein Zeitraum und ggf. schon (Test-)Projektmitglieder hinterlegt.

Nach der Anlage des Testprojektes sind die Testeinflussfaktoren zu bewerten. Die Menge der Testeinflussfaktoren wird dabei durch die Auswahl der aktuellen Projektphase und den zu schätzenden Testaktivitäten eingeschränkt. In Abbildung 6.5 ist die dafür zu verwendende Eingabemaske dargestellt. Auf der linken oberen Seite werden ein Softwareprojekt (*Project*) und eine Version (*Release*) ausgewählt sowie die genannten Einschränkungen vorgenommen. Basierend auf den Einschränkungen wird die Liste der Testeinflussfaktoren auf der linken Seite gefüllt, so dass der Benutzer nur die für die Projektphase und die Testaktivität relevanten Testeinflussfaktoren bewerten muss. Auf der rechten Seite werden nacheinander die Testeinflussfaktoren und die zugehörigen Testeinflussfaktor Attribute mit den Fragen dargestellt. Dabei wird nur ein Testeinflussfaktor Attribut auf einmal angezeigt, da ansonsten die Maske unübersichtlich wird. Die Beantwortung der Fragen erfolgt über das Setzen des Häkchens in der entsprechenden *Checkbox*. Der Benutzer hat die Möglichkeit über Schaltflächen am unteren Rand zwischen den Testeinflussfaktor zu navigieren oder durch einen Klick auf das gewünschte Testeinflussfaktor Attribut in der linken Liste.

Im Anschluss an die Bewertung der Testeinflussfaktoren ist die Testkomplexität zu erfassen. Hierfür wird von der TAQ Anwendung die in Abbildung 6.6 dargestellte Maske angeboten. Der Benutzer kann hier getrennt nach der gewünschten Teststufe, die Parameter für die Berechnung der Testkomplexität eingeben (vgl. Formel 6.1). Die Werte für die Teststufe Systemtest werden in den beiden Tabellen in der Mitte auf der rechten Seite eingegeben. Die Eingabe ist möglichst einfach gestaltet, d. h. es können

---

<sup>1</sup>Die Version bezieht sich immer auf die Hauptversion einer Software, die in dem Testprojekt geprüft werden soll. Untergeordnete Versionen, die z. B. der Behebung von Fehlern dienen, die während der Durchführung des Testprojektes gefunden wurden, werden durch die Hauptversion zusammengefasst.

## 6.2. Werkzeugunterstützung für die Methode TAQ

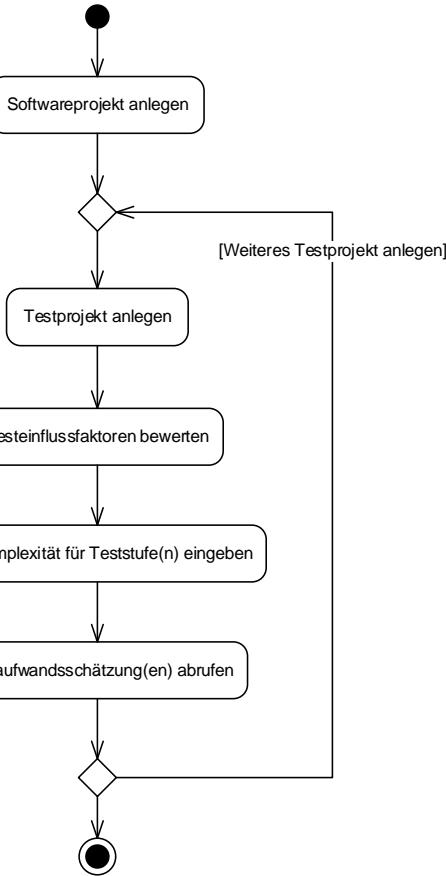


Abbildung 6.4.: Übersicht über den Ablauf zur Erstellung einer neuen Testaufwandsschätzung mit der TAQ Anwendung

über Kopieren und Einfügen Werte aus Excel übernommen, Zeilen durch Drücken der Entfernen-Taste gelöscht und Zeilen durch Eingabe von Text in der letzten Zeile der Tabellen hinzugefügt werden.

Wenn die Testeinflussfaktoren und die Testkomplexität erfasst worden sind, ist der Testaufwand über die TAQ Anwendung berechenbar. Dabei wird für den Systemtest die Formel 6.2 genutzt. In Abbildung 6.7 ist die Ausgabe der Berechnung zu sehen. Der Testaufwand wird immer für den Testkomplexitätsbereich  $0,8 \cdot K$  bis  $1,2 \cdot K$  berechnet (grüne Linie). Damit wird dem Erstellenden der Testaufwandsschätzung ein Überblick über den Verlauf des Testaufwands bei einer Abweichung von der gemessenen Testkomplexität gegeben, so dass dieser die Folgen abschätzen kann. Neben dem berechneten Testaufwand wird eine Abschätzung des Konfidenzintervall (rote Linien) eingezeichnet, das aus dem Standardschätzfehler der Regressionsanalyse des Modells für Testkomplexität (vgl. Tabelle C.3 im Anhang C.2) berechnet wird. Da-

## 6. Die Methode TAQ

**Test Cost Driver: Qualitätsziele**

Der Testeinflussfaktor Qualitätsziele umfasst die Attribute Funktionalität und weitere Qualitätsmerkmale. Umso höher dieser Testeinflussfaktor bewertet wird, desto höher ist der Testaufwand.

**Test Cost Driver Attribute: Funktionalität**

Dieses Attribut bezieht sich auf den gewünschten Abdeckungsgrad der Anforderungen im Test. Die Abdeckung wird von der Teststrategie bestimmt, die festlegt, wie welche Anforderung getestet werden soll. Wenn ein höherer Abdeckungsgrad gewünscht ist, müssen mehr Testfälle spezifiziert, koordiniert und ausgeführt werden. Damit erhöht sich bei steigendem Abdeckungsgrad der Testaufwand.

Für die Vergleichbarkeit der Einschätzung dieses Testeinflussfaktors muss festgelegt werden auf welchem Abstraktionsgrad der Begriff Anforderung genutzt wird. Beispielsweise kann eine Anforderung einem Use Case entsprechen oder einem Pfad in einem Use Case.

**Question: Zu welchem Grad sollen die Anforderungen abgedeckt werden?**

Very Low	Low	Nominal	High	Very High
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
0-0,24	0,25-0,49	0,5-0,74	0,75-0,9	>0,9

**Metric:**

Verhältnis der Anzahl von Anforderungen zu zu testenden Anforderungen.

Abbildung 6.5.: Bewertung der Testeinflussfaktoren

mit wird dem Testaufwandsschätzer die eine Prognose über die Vorhersagegenauigkeit gegeben.

## 6.2. Werkzeugunterstützung für die Methode TAQ

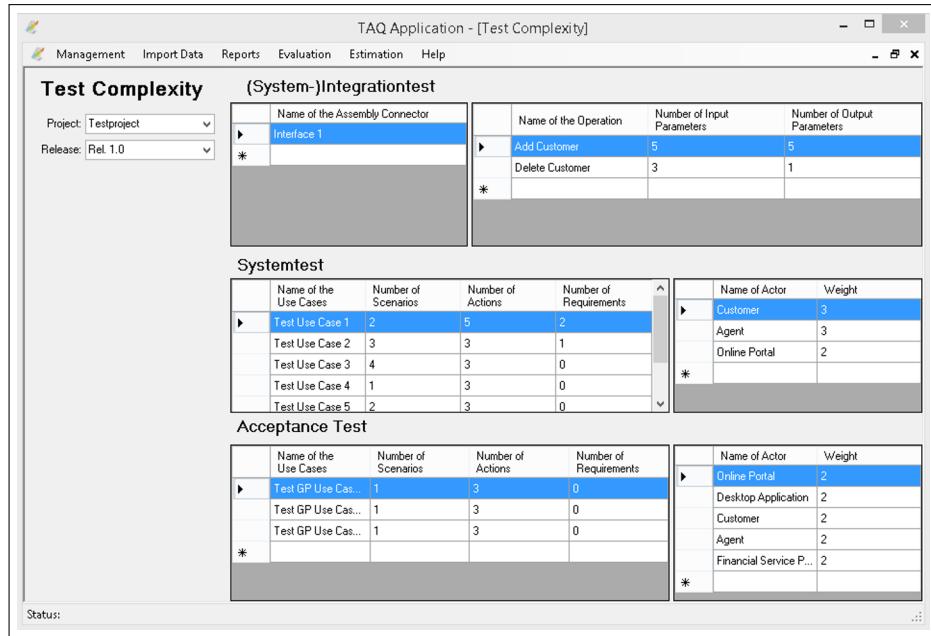


Abbildung 6.6.: Eingabe der Parameter für die Berechnung der Testkomplexität

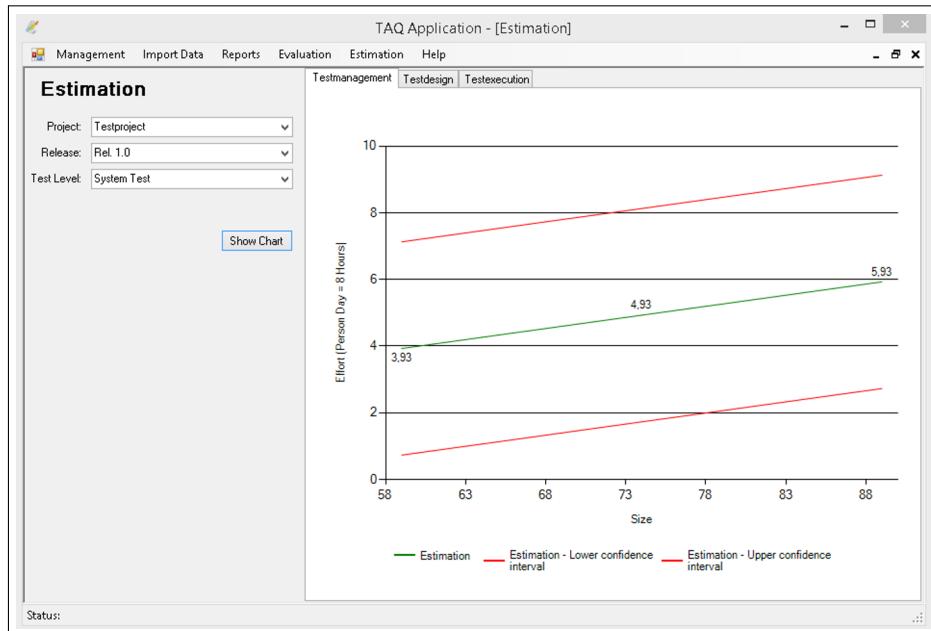


Abbildung 6.7.: Ausgabe der Testaufwandsschätzung für die Teststufe Systemtest und die Testaktivität Testdesign



# **7. Zusammenfassung, Bewertung und Ausblick**

In dieser Dissertation wurde eine neue Methode zur Testaufwandsschätzung in der analytischen Qualitätssicherung (TAQ) entwickelt. In diesem Kapitel werden zunächst die Ergebnisse zusammengefasst und anschließend auf Grundlage der in Kapitel 2 definierten Anforderungen bewertet. Abschließend erfolgt ein Ausblick auf mögliche weiterführende Forschungsthemen.

## **7.1. Zusammenfassung**

Das Thema Testaufwandsschätzung gewinnt gemeinsam mit den steigenden Ansprüchen an Qualitätssicherungsmaßnahmen stetig an Bedeutung. Gerade in den Teststufen (System-)Integrationstest, Systemtest und Abnahmetest, bei denen dediziertes Testpersonal zum Einsatz kommt, ist eine Ressourcenplanung (Personal- und Infrastruktur) auf Basis einer Testaufwandsschätzung von hoher Bedeutung. Daher war das Ziel dieser Arbeit die Definition einer Methode zur Testaufwandsschätzung unter Berücksichtigung der Testkomplexität und der Testeinflussfaktoren (vgl. Abschnitt 1.1).

Für die Ermittlung von Anforderungen an die Methode zur Testaufwandsschätzung wurden in Kapitel 2 zunächst Vorgehensmodelle in der Softwareentwicklung näher betrachtet, um den Kontext eines Testprojektes zu untersuchen. Davon ausgehend wurden Techniken zur Anforderungsspezifikation erläutert, da diese die Grundlage für die Formalisierung der Erfassung des Umfangs und der Komplexität der zu testenden Funktionalität sind. Danach wurde die dynamische Analyse am Beispiel des Testens diskutiert, da für die Aktivitäten bezüglich des Testens der Aufwand geschätzt wird. Abschließend wurde in die Domäne Aufwandsschätzung eingeführt, um aus etablierten Methoden weitere Anforderungen zu ermitteln.

## *7. Zusammenfassung, Bewertung und Ausblick*

In Kapitel 3 wurden die Anforderungen aus Kapitel 2 mit bestehenden Ansätzen zur Testaufwandsschätzung abgeglichen. Es zeigte sich, dass kein Ansatz diese Anforderungen ausreichend erfüllte. Daher wurde im Anschluss an den Vergleich der Lösungsansatz mit der Methode TAQ entwickelt.

Die Methode TAQ besteht aus den zwei Hauptbestandteilen: Testkomplexität und Testeinflussfaktoren. Mit der Testkomplexität wird die Komplexität des Testobjektes als natürliche Zahl ausgedrückt, die in einem linearen Zusammenhang zum erwarteten Testaufwand steht. Die Testeinflussfaktoren stehen für positive und negative Einflüsse auf den Testaufwand, die sich u. a. aus Eigenschaften des Testpersonals oder der Organisation ergeben.

Für die beiden Teile Testkomplexität und Testeinflussfaktoren wurde ein eigenständiges Modell entwickelt. Das Modell für Testkomplexität wurde schrittweise für die Teststufen (System-)Integrationstest, Systemtest und Abnahmetest in Kapitel 4 entworfen. Dabei wurden die Anwendung des Modells und die Verbindung zu Testentwurfsverfahren für jede Teststufe an Standardmodellierungssprachen der UML gezeigt. Für die Verwendung des Modells mit weiteren Modellierungssprachen wurde je Teststufe ein Metamodell entwickelt, das die strukturellen Abhängigkeiten der zu berücksichtigenden Eigenschaften für die Schätzung der Testkomplexität beschreibt. Diese Metamodelle können mit den entsprechenden Erläuterungen aus dem Kapitel 4 zur Anbindung einer weiteren Modellierungssprache genutzt werden.

Für den Nachweis des linearen Zusammenhangs zwischen der Testkomplexität und dem Testaufwand wurde eine Evaluation über mehr als 120 Testprojekten des Unternehmens CRM IT durchgeführt. Der Schwerpunkt lag hierbei auf Testprojekten der Teststufe Systemtest, da hier am meisten aussagekräftige Daten vorlagen. Die Evaluation zeigte neben dem gewünschten linearen Zusammenhang auch, dass u. a. Kennzahlen wie die durchschnittliche Produktivität des Unternehmens berechenbar und an Einzelprojekten nachvollziehbar sind.

Nach der Beschreibung des Modells für Testkomplexität wurde in Kapitel 5 das Modell für Testeinflussfaktoren erläutert. Hier wurden zunächst über Recherchen zu Einflussfaktoren in Softwareprojekten und in Testprojekten Testeinflussfaktoren identifiziert. Die Testeinflussfaktoren wurden dann in das Modell für Testeinflussfaktoren überführt, das über eine formalisierte Beschreibung die gleiche Nutzung der Testeinflussfaktoren durch verschiedene Anwender garantiert.

Danach wurde das Modell für Testeinflussfaktoren in einer zweigeteilten Evaluation überprüft. Der erste Teil bestand aus einer Expertenumfrage, die eine erste Einschät-

zung zu den beschriebenen Testeinflussfaktoren über die Höhe des Einflusses auf den Testaufwand gab. Mit den Rückmeldungen von ca. 90 Teilnehmern konnten auch Fragen zur Anwendbarkeit der Testeinflussfaktoren in einer bestimmten Projektphase oder Testaktivität beantwortet werden. Im zweiten Teil der Evaluation wurden abgeschlossene Testprojekte des Unternehmens CRM IT analysiert, um mit den Bewertungen aus der Expertenumfrage die Höhe des Einflusses jedes Testeinflussfaktors auf den Testaufwand zu determinieren. Genauso wie bei der Evaluation des Modells für Testkomplexität lag der Schwerpunkt auf Testprojekten der Teststufe Systemtest, da nur für diese Teststufe ausreichend Daten vorlagen.

Abschließend wurde die Methode TAQ in Kapitel 6 mit den Detaillierungen aus den vorherigen Kapiteln dargestellt. Da die Evaluation der Modelle für Testkomplexität und Testeinflussfaktoren hauptsächlich anhand von Daten aus Testprojekten der Stufe Systemtest durchgeführt worden ist, wird die Methode TAQ auch hier an dem Beispiel der Teststufe Systemtest zusammengefasst. Ergänzend wurde in dem Kapitel die entwickelte Werkzeugunterstützung vorgestellt, die eine einfache Vorhersage des Testaufwands für Testprojekte ermöglicht.

## 7.2. Bewertung der Methode TAQ

In Kapitel 2 wurden ausgehend von den in Abschnitt 1.1 beschriebenen Problemstellungen Anforderungen definiert. Basierend auf diesen Anforderungen wurden bestehende Methoden zur Testaufwandsschätzung in Kapitel 3 untersucht. Da keine der Methoden die aufgestellten Anforderungen ausreichend erfüllen konnte, wurde in Kapitel 3 als Lösung die Methode TAQ definiert und in den darauf folgenden Kapitel ausgearbeitet. Im Folgenden wird in Abschnitt 7.2.1 die Methode TAQ zunächst in Bezug zu den Problemstellungen bewertet, um deren Lösung durch die Methode zu zeigen. Danach wird die Methode in Abschnitt 7.2.2 anhand der formulierten Anforderungen im Detail diskutiert. Hiermit wird gezeigt, dass die Methode alle Anforderungen erfüllt. Anschließend werden in Abschnitt 7.2.3 die Vorteile und die Gültigkeitsbeschränkung der Lösung diskutiert.

### 7.2.1. Bewertung in Bezug zu den Problemstellungen

Die erste Problemstellung forderte die Formalisierung der Erfassung von Testkomplexität, um durch vergleichbare Werte für die Testkomplexität den Interpretations-

## *7. Zusammenfassung, Bewertung und Ausblick*

spielraum zu verringern. Die Lösung hierfür ist das Modell für Testkomplexität, das separat für jede Teststufe die einheitliche Erfassung der Testkomplexität nach formalen Vorgaben ermöglicht. Dies erleichtert die Prüfung einer Testaufwandsschätzung auf Vollständigkeit, da die Testkomplexität für einen zweiten Experten mit wenig Aufwand überprüfbar ist. Basierend auf dem berechneten Wert für Testkomplexität sind aussagekräftige Kennzahlen für die Produktivität berechenbar. Damit werden die Schwankungen der Produktivität verfolgbar und Auswirkungen von Maßnahmen zur Verbesserung der Produktivität belegbar.

Mit der zweiten Problemstellung sollten Testeinflussfaktoren identifiziert und formalisiert werden, um die Testeinflussfaktoren einheitlich zu beschreiben und zu nutzen. Die Identifikation wurde mit einer Recherche nach Testeinflussfaktoren (vgl. Kapitel 5 Abschnitt 5.1) umgesetzt. Das Ergebnis der Recherche ist eine Basismenge von 11 Testeinflussfaktoren, die über das Modell für Testeinflussfaktoren formalisiert worden sind. Über das Modell ist nun die einheitliche Nutzung der Testeinflussfaktoren durch verschiedene Testmanager möglich. Damit ist der Testaufwand bei der die Prüfung auf Plausibilität durch einen zweiten Experten nachvollziehbar.

Für die Lösung der dritten Problemstellung war die Formalisierung der Verbindung von Testkomplexität mit den Testeinflussfaktoren gefordert. Dies ist mit der Methode TAQ gegeben. Die Methode TAQ basiert auf einem algorithmischen Modell, das mit einer mathematischen Formel beschrieben ist. In der Formel werden die Modelle für Testkomplexität und Testeinflussfaktoren zusammengeführt. Damit sind über die Methode TAQ erstellte Testaufwandsschätzungen nachvollziehbar und vergleichbar.

Zusammenfassend erfüllt die entwickelte Methode TAQ die drei Problemstellungen dieser Dissertation vollständig.

### **7.2.2. Bewertung in Bezug zu den Anforderungen**

In Kapitel 2 wurden insgesamt 13 Anforderungen an die Lösung der Problemstellungen aus Abschnitt 1.1 identifiziert. Nachfolgend wird für jede Anforderung beschrieben, wie sie durch die Methode TAQ abgedeckt ist.

#### **Anforderung 1: Beliebiger Zeitpunkt für die Durchführung der Testaufwandsschätzung**

Die Methode TAQ bietet die Möglichkeit, den Testaufwand zu jedem möglichen Zeitpunkt in einem Softwareprojekt zu berechnen. Damit ist es jederzeit möglich, ein kaufmännisches Angebot für ein Testprojekt auf Grundlage einer

Testaufwandsschätzung zu erstellen. Jede der drei Phasen Anforderungsphase, Designphase und Entwicklungsphase, die in Abschnitt 2.1 bei der Erläuterung des V-Modells eingeführt wurden, wird durch die Methode TAQ unterstützt. Der unterschiedliche Informationsstand in den jeweiligen Phasen wird über Testeinflussfaktoren berücksichtigt.

### **Anforderung 2: Aktualisierbare Testaufwandsschätzung**

Eine über die Methode TAQ erstellte Testaufwandsschätzung ist aktualisierbar. Die Aktualisierung wird entweder durch ein verändertes/neues Testobjekt oder durch Veränderungen im Kontext des Testprojektes notwendig. Bei der Aktualisierung ist der Testaufwand über die Methode TAQ neu zu berechnen. Dabei können die ermittelte Testkomplexität und auch die schon bewerteten Testeinflussfaktoren wiederverwendet und partiell aktualisiert werden. Hierüber sind Testaufwandsschätzungen, die zu einer frühen Projektphase erstellt worden sind, teilweise wiederverwendbar und es werden damit Aufwände zur Erstellung der Schätzung vermieden.

Bei der Aktualisierung der Testkomplexität müssen nur die Teile aktualisiert werden, die in Verbindung mit dem sich veränderten oder neuen Testobjekt stehen. Dies ist möglich, da für die Schätzung der Testkomplexität jedes Testobjekt separat erfasst werden muss. Bei der Neubewertung der Testeinflussfaktoren müssen nur diejenigen berücksichtigt werden, für die sich der Kontext des Testprojektes geändert haben. Dasselbe gilt auch für eine Wiederholung der Testaufwandsschätzung zu einer späteren Projektphase, wie der Entwicklungsphase, bei vorhandener Schätzung aus der Anforderungsphase.

### **Anforderung 3: Berücksichtigung der Qualifikationen von Personen in den Rollen Anforderungsanalytiker des Auftraggebers sowie des Auftragnehmers und des Softwarearchitekten**

In der Methode TAQ werden über den Testeinflussfaktor *Qualifikation und Erfahrung* die Qualifikationen der Personen in den Rollen Anforderungsanalytiker des Auftraggebers sowie des Auftragnehmers und des Softwarearchitekten berücksichtigt (vgl. Abschnitt 5.2.7). Dabei werden die Qualifikationen zunächst getrennt nach Rollen erfasst und anschließend zu einer Gesamtbewertung zusammengefasst. Diese Gesamtbewertung fließt direkt in die Berechnung des Testaufwands ein (vgl. Abschnitt 3.2).

### **Anforderung 4: Unterstützung von Standardmodellierungssprachen der UML**

Mit dem Modell für Testkomplexität bietet die Methode TAQ die Möglichkeit, die zu testende Funktionalität in einer UML basierten Spezifikation zu

## 7. Zusammenfassung, Bewertung und Ausblick

identifizieren. Dabei werden je nach Teststufe entsprechende Standardmodellierungssprachen der UML unterstützt. Auf der Stufe des Systemtests und des Abnahmetests werden UML Use Case Diagramme und Aktivitätendiagramme zur Identifikation der zu testenden Funktionalität genutzt. Daraus ist anschließend über teststufenspezifische Formeln die Testkomplexität schätzbar.

Auf der Stufe (System-)Integrationstest sind UML Komponentendiagramme der Ausgangspunkt, um die zu testenden Schnittstellen zu identifizieren. Die zu testende Interaktion wird über Sequenzdiagramme erfasst. Die detaillierteste Ebene bieten Klassendiagramme, über die zu testenden Ein- und Ausgabeparameter der Schnittstellenoperationen identifiziert werden. Aus diesen Diagrammen der genannten drei Standardmodellierungssprachen der UML ist dann genauso wie beim Systemtest bzw. Abnahmetest über eine Formel die Testkomplexität schätzbar.

### **Anforderung 5: Möglichkeit zur Anbindung weiterer Modellierungssprachen**

Das Modell für Testkomplexität bietet die Möglichkeit neben den Standardmodellierungssprachen der UML weitere Modellierungssprachen, wie z. B. BPMN, zu verwenden. Dazu wurde je Teststufe ein Metamodell entwickelt, das die strukturellen Abhängigkeiten der zu berücksichtigenden Eigenschaften für die Schätzung der Testkomplexität beschreibt. Diese Metamodelle können mit den entsprechenden Erläuterungen aus dem Kapitel 4 zur Definition einer Abbildung weiterer Modellierungssprachen genutzt werden.

### **Anforderung 6: Aufschlüsselung des Testaufwands nach Testaktivitäten**

Die Methode TAQ unterstützt die Aufschlüsselung des Testaufwands nach Testaktivitäten. Dabei wird keine prozentuale Aufschlüsselung des geschätzten Testaufwands auf Testaktivitäten vorgenommen, sondern der Testaufwand wird separat für die Testaktivitäten Testmanagement, Testdesign und Testausführung geschätzt. Mit der Aufschlüsselung nach Testaktivitäten wird der Testmanager in der Personal- und Projektplanung unterstützt.

Der Hauptunterschied bei der Schätzung des Testaufwands für die Testaktivitäten liegt im Modell für Testeinflussfaktoren. Hier werden zum Einen für jede Testaktivität spezifische Testeinflussfaktoren berücksichtigt. Zum Anderen sind die Gewichte der Testeinflussfaktoren für jede Testaktivität angepasst und werden dementsprechend in der Schätzung berücksichtigt.

**Anforderung 7: Unterstützung der Teststufen (System-)Integrationstest, Systemtest und Abnahmetest**

Die Methode TAQ unterstützt die Teststufen (System-)Integrationstest, Systemtest und Abnahmetest. Dazu bietet das Modell für Testkomplexität explizit für jede Teststufe eine Formel zur Berechnung der Testkomplexität an. Bei der Berechnung werden, wie bei Anforderung 4 in diesem Abschnitt beschrieben, abhängig von der Teststufe unterschiedliche Teile der Spezifikation berücksichtigt. Zusätzlich werden die Unterschiede der Teststufen über unterschiedliche Gewichte der Testeinflussfaktoren abgebildet.

**Anforderung 8: Algorithmisches Modell mit *Top-Down* Vorgehen**

Die Methode TAQ basiert auf einem algorithmischen Modell, da eine mathematische Formel zur Bestimmung des Testaufwands verwendet wird. Laut der Definition in Abschnitt 2.4.4 folgt die Methode TAQ dem *Top-Down* Vorgehen, da die Testaufwandsschätzungen auf dem Wert für Testkomplexität und Eigenschaften des Testprojektkontextes über Testeinflussfaktoren basiert. Dabei sind die Testkomplexität und die Testeinflussfaktoren sogenannten globalen Eigenschaften eines Softwareprojektes zuzuordnen.

**Anforderung 9: Multiplikative Form der Formel zur Berechnung des Testaufwands**

Die grundlegende Formel 3.8 entspricht der multiplikativen Form, d. h. der Testaufwand wird über ein Produkt berechnet. Dabei sind die Gewichte der Testeinflussfaktoren als prozentuale Veränderung der Produktivität interpretierbar. Hiermit wird eine einfachere Interpretierbarkeit für den Endanwender gegenüber dem Aufsummieren von Gewichten erreicht, wie bei der Einführung in die Domäne Aufwandsschätzung in Abschnitt 2.4 dargestellt.

**Anforderung 10: Berücksichtigung von Projektdaten aus abgeschlossenen Projekten**

Für das Kalibrieren der Formel 3.8 an einen Unternehmenskontext wird explizit auf Daten aus abgeschlossenen Testprojekten des jeweiligen Unternehmens zurückgegriffen. Hiermit beruht die Testaufwandsschätzung direkt auf den Erfahrungen des Unternehmens. Die Kalibrierung der Formel 3.8 betrifft die durchschnittlichen Produktivität  $P$ , die Konstante  $C$  und die Gewichte der Testeinflussfaktoren jeweils in Abhängigkeit von den Teststufen und Testaktivitäten.

Die Kalibrierung der Formel wurde in der vorliegenden Dissertation durchgängig am Beispiel der Teststufe Systemtest für alle drei Testaktivitäten demonstriert. Dazu wurde zunächst bei der Evaluation des Modells für Testkomplexität erläutert, wie aus vorliegenden Testprojektdaten die durchschnittliche Produktivität

## 7. Zusammenfassung, Bewertung und Ausblick

$P$  zu berechnen ist (vgl. Abschnitt 4.2.1.2). Anschließend wurde in Kapitel 5 bei der empirischen Untersuchung des Modells für Testeinflussfaktoren gezeigt, wie aus Testprojektdaten die Gewichte der Testeinflussfaktoren bestimmt sind (vgl. Abschnitt 5.3.2). Bei der Gewichtsbestimmung der Testeinflussfaktoren wurde außerdem ein Verfahren diskutiert, wie die Konstante  $C$  zu berechnen ist (vgl. 5.3.2.2).

Für die Kalibrierung an einen neuen Unternehmenskontext müssen dieselben Schritte wie in dieser Dissertation durchgeführt werden. Da oft nicht genügend Projektdaten zur Verfügung stehen, wurden für diesen Fall spezielle Hinweise für ein Vorgehen bei der Kalibrierung gegeben (vgl. Abschnitt 5.4). Dabei wird zunächst z. B. nur die globale Produktivität bestimmt und Konstante  $C$  anhand der Daten kalibriert.

### **Anforderung 11: Formale Vorgaben zur Bewertung von Testeinflussfaktoren**

Da in der Methode TAQ die Beschreibung der Testeinflussfaktoren über Attribute und deren zugehörigen Fragen formalisiert worden ist, wird die Bewertung der Testeinflussfaktoren über formale Vorgaben direkt aus den Antworten der Fragen abgeleitet (vgl. Abschnitt 5.2). Für die Beantwortung der Fragen wurden mehrheitlich Metriken formuliert und festgelegt, wie die Ergebnisse der Metriken auf der Skala *sehr niedrig* bis *sehr hoch* einzurichten sind. Mit den genannten formalen Vorgaben, den Metriken und der vorgebenden Auswertung der Metriken ist diese Anforderung erfüllt.

### **Anforderung 12: Stabil gegenüber neuen/wechselnden Technologien zur Realisierung des Softwareproduktes**

Für die Stabilität einer Methode zur Testaufwandsschätzung gegenüber neuen oder wechselnden Technologien muss die Methode möglichst unabhängig von der Technologie der zu testenden Software sein. Dies ist bei der Methode TAQ mit der Kalibrierung anhand von Daten aus abgeschlossenen Testprojekten gewährleistet (vgl. Anforderung 10). Über die regelmäßige Kalibrierung mit Daten aus Testprojekten fließen Veränderungen des Testaufwands durch den Einsatz neuer/wechselnder Technologien in die Testaufwandsschätzung ein.

Damit die Kalibrierung der Parameter in der Methode TAQ bei neuen/wechselnden Technologien möglich ist, müssen die Modelle für Testkomplexität und Testeinflussfaktoren unabhängig von der Technologie sein. Da die Testkomplexität direkt über die Spezifikation der Software geschätzt wird, ist sie per Definition unabhängig von der Technologie zu Realisierung des eigentlichen Testaufwands.

chen Softwareproduktes. Im Modell für Testeinflussfaktoren wurden die Fragen zur Bewertung der Testeinflussfaktoren ohne Bezug zur Technologie formuliert. Darüber ist das Modell für Testeinflussfaktoren unabhängig von der zugrundeliegenden Technologie der Software.

Mit der Unabhängigkeit der Modelle für Testkomplexität und Testeinflussfaktoren von der Technologie der zu testenden Software wird garantiert, dass in einem sich stetig verändernden Umfeld eine regelmäßige Neukalibrierung der Methode TAQ möglich ist. Somit entwickelt sich die Methode zusammen mit ihrem Einsatzkontext weiter und ist insgesamt stabil gegenüber neuen/wechselnden Technologien.

### **Anforderung 13: Aktualisierbare Testeinflussfaktoren**

In der Methode TAQ wurde das Modell für Testeinflussfaktoren so gestaltet, dass die Testeinflussfaktoren hinsichtlich ihrer Gewichte aktualisierbar sind. Dazu wurden Hinweise zur Kalibrierung der Gewichte über Daten aus abgeschlossenen Testprojekten gegeben (vgl. Abschnitt 5.4.1). Darüber hinaus können Testeinflussfaktoren dem Modell hinzugefügt oder entfernt werden. Auch hierfür wurden Voraussetzungen und ein mögliches Vorgehen beschrieben (vgl. Abschnitt 5.4.2).

Zusammenfassend wurden mit der Methode TAQ alle Anforderungen erfüllt, die für eine Lösung im Rahmen dieser Dissertation gefordert gewesen sind.

### **7.2.3. Vorteile und Gültigkeitsbeschränkung**

Die Methode TAQ ist universell einsetzbar mit der Einschränkung, dass eine Kalibrierung an den jeweiligen Unternehmenskontext vorgenommen werden muss. Hierfür wurden speziell bei der Bestimmung der Gewichte der Testeinflussfaktoren in Kapitel 5 Hinweise gegeben. Damit ist, wie die Evaluierung an den Testprojekten der Stufe Systemtest des Unternehmens CRM IT gezeigt hat, eine vergleichbare Genauigkeit zu einer Expertenschätzung erreichbar (vgl. Abschnitt 5.3.3). In Teilen schneidet die Methode sogar besser ab. Mit einer steigenden Anzahl von Daten aus abgeschlossenen Testprojekten ist hier noch mit einer Steigerung zu rechnen, da die Gewichte der Testeinflussfaktoren, die globale Produktivität  $P$  und die Konstante  $C$  aus Formel 3.10 besser kalibriert werden können (vgl. Erläuterungen zur Anforderung 10 in Abschnitt 7.2.2).

## 7. Zusammenfassung, Bewertung und Ausblick

Das Modell für Testkomplexität bietet über Unternehmensgrenzen hinweg die Möglichkeit, einheitlich vergleichbare Werte für die Testkomplexität zu bestimmen. Hier ist nur bei der Verwendung abweichender Modellierungssprachen eine Abbildung auf die im Modell für Testkomplexität beschriebenen Elemente zu definieren. Damit ist die Testkomplexität auch über verschiedene Modellierungssprachen vergleichbar.

Die Basismenge von 11 Testeinflussfaktoren im Modell für Testeinflussfaktoren bietet einen guten Ausgangspunkt, um Einflüsse auf den Testaufwand in einem Testprojekt abzuschätzen. Da diese Testeinflussfaktoren aus einer Synthese von einer hohen Anzahl bekannter Einflussfaktoren in Software- und Testprojekten abgeleitet worden sind, werden sie in vielen Unternehmenskontexten mit nur leichten Anpassungen in der Gewichtung verwendbar sein. Die allgemeine Akzeptanz zeigte sich auch in der Expertenumfrage in Abschnitt 5.3.1, an der Mitarbeiter aus verschiedenen Unternehmen teilgenommen haben.

### 7.3. Ausblick

Die Planung von Testprojekten hängt in hohem Maß von der Qualität der Testaufwandsschätzung ab. Dabei sind vor allem die Vollständigkeit der Schätzung, die transparente Berücksichtigung von Einflussfaktoren und die Nachvollziehbarkeit der Berechnung des Testaufwands von entscheidender Bedeutung. Diese Eigenschaften bietet die Methode TAQ in einem *Top-Down* Vorgehen, das auf einem algorithmischen Modell beruht.

Mit der Methode TAQ ist die Vollständigkeit einer Testaufwandsschätzung hinsichtlich der zu testenden Funktionalität überprüfbar, da mit dem Modell für Testkomplexität formale Vorgaben zur Erfassung der Testkomplexität existieren. Das integrierte Modell für die Testeinflussfaktoren garantiert eine transparente Berücksichtigung des Testprojektkontextes und über verbindliche mathematische Formeln zur Ermittlung des Testaufwands ist die Transparenz der Berechnung gewährleistet.

Damit bietet die Methode TAQ einen guten Ausgangspunkt für die Lösung weiterer verwandter Problemstellungen bzw. Erweiterungen und Verbesserungen. Wie diese aussehen können, wird in diesem Abschnitt in zwei Teilen beschrieben. Im Abschnitt 7.3.1 werden direkte Erweiterungen und Verbesserungen der Methode TAQ erläutert. Danach wird im Abschnitt 7.3.2 beschrieben, welche Perspektiven sich für den Einsatz der Methode TAQ in anderen Kontexten ergeben und wie dabei ausgehend von der Methode TAQ weitere Problemstellungen lösbar sind.

### 7.3.1. Erweiterungen und Verbesserungen der Methode TAQ

Um die Methode TAQ in der Praxis für die Teststufen (System-)Integrationstest und Abnahmetest einzusetzen, müssen noch mehr Daten aus Testprojekten gesammelt werden. Da die Sammlung der Projektdaten auch nach Abschluss dieser Dissertation bei dem Unternehmen CRM IT fortgesetzt wird, ist in zwei bis drei Jahren eine deutlich größere Datenbasis für alle Teststufen sowie Testaktivitäten verfügbar. Damit sind weitergehende Untersuchungen über die Testeinflussfaktoren und deren Auswirkungen auf den Testaufwand möglich.

Zur tieferen Prüfung der Methode TAQ ist ein Einsatz in einem weiteren Unternehmenskontext sinnvoll. Hier sollten entsprechende Daten aus abgeschlossenen Projekten gesammelt und zur Kalibrierung der Methode TAQ genutzt werden. Das Ergebnis, z. B. die Gewichtung der Testeinflussfaktoren, ist anschließend mit der Kalibrierung auf Basis der Daten aus dem Unternehmen CRM IT zu prüfen.

Die mit der Methode TAQ eingeführten Testeinflussfaktoren bieten eine ausreichende Basis für die Durchführung von Testaufwandsschätzungen. Dies wurde am Beispiel des Systemtests in Abschnitt 5.3.3 im Unternehmen CRM IT gezeigt. Jedoch sind auch hier weitere Forschungen empfehlenswert, da insbesondere die Testeinflussfaktoren von Teststufe zu Teststufe verschieden sein können. Weiterhin können von dem Unternehmen CRM IT abweichende Unternehmenskontakte zu grundsätzlich unterschiedlichen Testeinflussfaktoren führen. Für eine Untersuchung müssen in weiteren Unternehmen strukturierte Interviews mit dem Testpersonal geführt und mögliche Testeinflussfaktoren identifiziert werden. Davon ausgehend sind Daten aus Testprojekten in den Unternehmen zu sammeln, um Erkenntnisse über die Höhe des Einflusses der Testeinflussfaktoren zu sammeln. Anschließend können die Testeinflussfaktoren verschiedener Unternehmen miteinander abgeglichen werden, um allgemeine Testeinflussfaktoren zu identifizieren.

Zur Bewertung der Genauigkeit einer Testaufwandsschätzung über die Methode TAQ ist die Ausweisung der Wahrscheinlichkeit sinnvoll, wie wahrscheinlich der tatsächliche Testaufwand z. B. in einem Bereich von  $\pm 20\%$  des geschätzten Aufwands liegt. Der Bereich und die Wahrscheinlichkeit sind auf der Basis der (Standard-)Abweichungen der Eingangsparameter (vgl. Konfidenzintervall in Zusammenhang mit linearer Regression [BS10, S. 351]) berechenbar. Dazu sind u. a. weitere Forschungen über die einzelnen Testeinflussfaktoren notwendig, da die Auswirkungen jedes Testeinflussfaktors auf die gesamte Genauigkeit der Testaufwandsschätzung unterschiedlich sind. Für diese Forschungen sind für jeden Testeinflussfaktor und zugehörigen qualitativen Aus-

## 7. Zusammenfassung, Bewertung und Ausblick

prägungen eine höhere Anzahl Daten aus abgeschlossenen Testprojekten notwendig. Für derartige Forschungen sind ca. 20 Datensätze je Einflussfaktor und Ausprägung notwendig [Fro09, S. 165]. Damit sind mehr als 1.100 Projekte für die Testeinflussfaktoren erforderlich.

Der Aufwand für ein Testprojekt ist in den meisten Fällen proportional zu den erstellten Testfällen. Bei der Entwicklung des Modells für Testkomplexität wurde daher speziell auf Testentwurfsverfahren eingegangen, die für die Teststufen standardmäßig eingesetzt werden. Ergänzend wird im Modell für Testeinflussfaktoren über den Testeinflussfaktor Qualitätsziele die gewünschte Testabdeckung grob eingeordnet. Hier bieten sich weitere Forschungen an, da gerade von dem eingesetzten Testentwurfsverfahren und der Testabdeckung die Anzahl der erstellten Testfälle abhängt. Hier müssen Daten aus Testprojekten über die Anzahl erstellter Testfälle, die eingesetzten Testentwurfsverfahren und die erreichte Testabdeckung gesammelt werden. Darüber ist dann eine Anpassung des Modells für Testkomplexität hinsichtlich einer Berücksichtigung von Testentwurfsverfahren und geplanter Abdeckung ableitbar.

Eine weitere Untersuchung sollte sich im Bereich der Wiederverwendung von Testmitteln anschließen. Wiederverwendung ist in zwei Fällen zu betrachten. Der erste Fall ist die Wiederverwendung von Testmitteln aus abgeschlossenen Testprojekten. Dies kann beispielsweise Testfälle betreffen, die aus einem Vorgängerprojekt in das aktuelle Projekt übernommen werden können. Im zweiten Fall ist die zu testende Funktionalität so ähnlich, dass Testfälle im aktuellen Testprojekt mehrfach wiederverwendet werden können. In der Methode TAQ wurde die Wiederverwendung über den Testeinflussfaktor *Wiederverwendbarkeit* berücksichtigt, der linear auf den Testaufwand wirkt. Hier ist zu untersuchen, ob die Wiederverwendung exponentiell auf den Umfang des Testprojektes wirken sollte. Dies ist z. B. bei COCOMO II der Fall (vgl. Abschnitt 2.4.3). Entsprechend wäre der Testeinflussfaktor *Wiederverwendbarkeit* als Exponent an der Testkomplexität zu verrechnen. Zur Untersuchung müssen mehr Daten aus Testprojekten gesammelt werden. Speziell sind mehr Testprojekte erforderlich bei denen die Wiederverwendung von Testmitteln in hohem Maß genutzt worden ist.

Der Einsatz von Testautomatisierung verändert den Testaufwand für alle Testaktivitäten. Testautomatisierung bedeutet nach dem ISTQB Glossar [GTB10, S. 51]: „Einsatz von Softwarewerkzeugen zur Durchführung oder Unterstützung von Testaktivitäten, z.B. Testmanagement, [...], Testausführung und Soll/Ist-Vergleich.“ Die Höhe und Richtung der Veränderung des Aufwands für die Testaktivitäten sind unterschiedlich. Der Aufwand für die Testaktivität Testdesign steigt erheblich, wenn der Aufwand für die Entwicklung einer Testautomatisierung miteinbezogen wird. Gleich-

zeitig sinkt der Testausführungsauflauf, da ein Automat die definierten Tests ausführt und ein Mensch lediglich das Ergebnis kontrollieren muss. Die Auswirkung auf die Testaktivität Testmanagement können stagnieren, da z. B. mehr Koordinationsaufwände während der Erstellung von Testfällen notwendig sind. Um die Auswirkungen des Einsatzes von Testautomatisierung besser bewerten zu können, müssen darüber Daten aus Testprojekten gesammelt werden.

Zusammenfassend lassen sich die Akzeptanz und Anwendbarkeit der Methode TAQ in der Praxis mit den genannten Erweiterungen und Verbesserungen erhöhen.

### **7.3.2. Weitere Perspektiven für den Einsatz der Methode TAQ**

Die Methode TAQ ist ein guter Ausgangspunkt, um im Bereich Testaufwandsschätzungen in nah verwandten Kontexten weiterzuforschen. Davon werden zwei in diesem Abschnitt näher vorgestellt und damit eine Perspektive für den Einsatz der Methode TAQ aufgezeigt.

#### **Die Methode TAQ und das modellbasierte Testen (MBT)**

Der erste Kontext ist das modellbasierte Testen (MBT), das als Unterstützung der Aktivität Testdesign durch Automatisierung aufgefasst wird (vgl. Dissertation von Mlynarski [Mic11]). Dazu wird in mehreren Schritten aus einer modellbasierten Spezifikation der Softwarelösung ein Testmodell generiert, aus dem anschließend automatisiert konkrete Testfälle erzeugbar sind. Damit verändert MBT die Aufgaben eines Testdesigners deutlich, da das (direkte) Erstellen der Testfälle wegfällt. Stattdessen muss der Testdesigner die bestehenden Modelle aus der Spezifikation analysieren und um Informationen zur Testmodellgenerierung anreichern.

In einem Testprojekt mit dem Einsatz von MBT muss auch der Testaufwand für die entsprechende Personal- und Ressourcenplanung geschätzt werden. Dafür eignet sich die Methode TAQ mit einigen Anpassungen. Die Hauptanpassungen betreffen die Testeinflussfaktoren, da der Aufwand für den Einsatz von MBT im Testdesign stärker von der Qualität der (modellbasierten) Spezifikation abhängt als der Aufwand für das direkte Erstellen von Testfällen. Die Qualität der (modellbasierten) Spezifikationen wird in der Methode TAQ über den Testeinflussfaktor *Qualität der Testbasis* hinsichtlich der Dimensionen Verständlichkeit, Konsistenz und Vollständigkeit bewertet. Diese Dimensionen reichen für eine Schätzung des Testaufwands mit Berücksichtigung

## *7. Zusammenfassung, Bewertung und Ausblick*

im Kontext von MBT nicht aus, da weitere Eigenschaften der Modelle in den Spezifikationen für den Einsatz von MBT zu betrachten sind. Wann eine Spezifikation in Bezug auf MBT qualitativ gut ist und welche Eigenschaften der Spezifikationen dafür analysiert werden müssen, wird im Folgenden dargestellt.

Wie von Mlynarski in [Mic11, S. 130ff] ausgeführt, besteht der erste Schritt zur Generierung des Testmodells aus einer Analyse der Testbarkeit der modellbasierten Spezifikationen. Darüber sind Fehler und Lücken aufzudecken, die eine Generierung des Testmodells verhindern. Die Fehler und Lücken in der Spezifikation sind über entsprechende Gegenmaßnahmen behebbar [Mic11, S. 130ff]. Wie aufwendig dieser Schritt ist, hängt direkt von der Qualität der Spezifikationen ab. Dabei ist eine Spezifikation in Bezug auf MBT qualitativ gut, wenn eine hohe Testbarkeit vorliegt. Die Kriterien für eine hohe Testbarkeit sind systematisch über einen Modell-Qualitäts-Plan (MQP) definierbar [Voi09]. Nach der Bestimmung eines MQP für die Testbarkeit muss dieser als Einflussfaktor in das Modell für Testeinflussfaktoren aufgenommen werden. Dies erfordert eine Erweiterung des Bewertungsverfahrens der Testeinflussfaktoren, da die Prüfung der Qualitätsziele des MQP berücksichtigt werden muss.

Die Bestimmung des MQP für die Testbarkeit, die Verknüpfung mit dem Modell für Testeinflussfaktoren und letztendlich auch die Bestimmung der Auswirkungen auf den Testaufwand erfordern tiefere Forschungen. Wie bei den direkten Erweiterungen der Methode TAQ sind dafür Daten aus Testprojekten erforderlich. Dabei sind speziell Testprojekte notwendig, die MBT einsetzen.

### **Die Methode TAQ in agilen Projekten**

Der zweite Kontext ist der Einsatz der Methode TAQ in agilen Projekten. Das Testen in agilen Projekten ist an den Entwicklungsiterationen auszurichten, da nicht nur das vollständige Softwareprodukt zum Schluss sondern auch Inkremeante nach jeder Iteration zu testen sind [GG12]. Damit muss der Testaufwand für jedes Inkrement und den Test des vollständigen Softwareproduktes bestimmt werden. Dafür ist die Methode TAQ mit Anpassungen in den Modellen für Testkomplexität und Testeinflussfaktoren einsetzbar.

Die Anpassungen für das Modell für Testkomplexität sind optional, d. h. das bestehende Verfahren zur Schätzung der Testkomplexität könnte weiterverwendet werden. Für einen besseren Anschluss an das Vorgehen in einem agilen Projekte sollte das Verfahren allerdings angepasst werden. In dieser Dissertation wurde bei der Vorstellung von Methoden zur Expertenschätzung (vgl. Abschnitt 2.4.2) darauf eingegangen, wie

die Komplexität in einem agilen Projekt z.B. über Planning Poker im Projektteam gemeinsam geschätzt wird. Die Schätzung der Testkomplexität ist hieran anzuschließen, so dass erstens der Aufwand für die Bestimmung der Testkomplexität verringert wird und zweitens das Wissen des gesamten Teams einfließt.

Für das Testen der Inkremeante und des vollständigen Produktes werden die Testmittel, wie Testfälle und Testplanung, sukzessiv entwickelt. Dabei wird jeweils ein großer Teil der Testmittel aus einer vorangegangenen Iteration entweder direkt oder in abgeänderter Form wiederverwendet. Um dies zu berücksichtigen, ist das Modell für Testeinflussfaktoren zu erweitern. In der aktuellen Version der Methode TAQ wird nur die Wiederverwendung insgesamt über den Testeinflussfaktor *Wiederverwendbarkeit* betrachtet. Dieser ist mindestens in zwei Testeinflussfaktoren aufzuteilen, so dass Wiederverwendung und Änderungsbedarf von Testmitteln getrennt bewertet werden können.

Mit den Erweiterungen der Modelle für Testkomplexität und Testeinflussfaktoren ist die Methode TAQ auch in agilen Projekten verwendbar. Für die Umsetzung der Erweiterungen sind dennoch weitere Forschungen notwendig. Dabei muss erstens untersucht werden, wie die Testkomplexität bei der bereits stattfindenden Komplexitätsbestimmung schätzbar ist. Zweitens muss das Modell für Testeinflussfaktoren überarbeitet werden. Dabei sind zunächst die Bereiche Wiederverwendung und Änderungsbedarf über weitere Testeinflussfaktoren zu berücksichtigen. Danach sind weitere Testeinflussfaktoren zu definieren, die sich auch dem Vorgehen in einem agilen Projekt ergeben. Beispielsweise ist in agilen Projekten die Kommunikation intensiver und die Nähe zum Kunden höher.



# Literaturverzeichnis

- [AAM09] ALMEIDA, Érika R. ; ABREU, de Bruno T. ; MORAES, Regina: An Alternative Approach to Test Effort Estimation Based on Use Cases. In: *ICST 2009, Second International Conference on Software Testing Verification and Validation, 1-4 April 2009, Denver, Colorado, USA*, IEEE Computer Society, 2009. – ISBN 978–0769536019, S. 279–288
- [AB07a] ARANHA, Eduardo ; BORBA, Paulo: An Estimation Model for Test Execution Effort. In: *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, IEEE, 2007. – ISBN 978–0769528861, S. 107–116
- [AB07b] ARANHA, Eduardo ; BORBA, Paulo: Test Effort Estimation Models Based on Test Specifications. In: *TAIC PART-MUTATION 2007 : Testing: Academic and Industrial Conference Practice and Research Techniques*, 2007, S. 67–71
- [ACM] ACM: *ACM Digital Library*. <http://dl.acm.org/>, Abruf: 27.09.2014
- [AD SJ01] ANDA, Bente ; DREIEM, Hege ; SJØBERG, Dag I. K. ; JØRGENSEN, Magne: Estimating Software Development Effort Based on Use Cases-Experiences from Industry. In: *Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools*, Springer-Verlag, 2001 («UML» '01). – ISBN 3–540–42667–1, S. 487–502
- [AG83] ALBRECHT, Allan J. ; GAFFNEY, John E. J.: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. In: *IEEE Transactions on Software Engineering* 9 (1983), Nr. 6, S. 639–648. – ISSN 0098–5589
- [Alb79] ALBRECHT, Allan J.: Measuring Application Development Productivity. In: PRESS, I. B. M. (Hrsg.): *IBM Application Development Symp*, 1979, S. 83–92

## Literaturverzeichnis

- [And02] ANDA, Bente: Comparing Effort Estimates Based on Use Case Points with Expert Estimates. In: *In Empirical Assessment in Software Engineering (EASE 2002)*, 2002, S. 8–10
- [AR96] ABRAN, A. ; ROBILLARD, P. N.: Function points analysis: an empirical study of its measurement processes. In: *Software Engineering, IEEE Transactions on* 22 (1996), Nr. 12, S. 895–910. – ISSN 0098–5589
- [ASQ14] ASQF E.V.: *ASQF-Umfrage 2014 Branchenreport für die deutschsprachige IT-Branche: Qualitätstrends in der Softwareentwicklung.* [https://www.asqf.de/pressemitteilungen/items/nie-zuvor-so-viel-investiert-status-der-qualitaetssicherung-steigt.html?file=tl\\_files/asqf/downloads/ASQF%20Branchenreport%202014.pdf](https://www.asqf.de/pressemitteilungen/items/nie-zuvor-so-viel-investiert-status-der-qualitaetssicherung-steigt.html?file=tl_files/asqf/downloads/ASQF%20Branchenreport%202014.pdf). Version: 2014, Abruf: 04.10.2014
- [Baa09] BAARDA, Rob: *Test Estimation and Metrics.* <http://www.sogeti.com/upload/COM/Employees%20only/Testing%20Academy%202009/TA%2009%2007%20Test%20estimation%20and%20Metrics%20Rob%20B%200.9.ppt>. Version: 2009, Abruf: 30.09.2013
- [BAC00] BOEHM, Barry W. ; ABTS, Chris ; CHULANI, Sunita: Software development cost estimation approaches - A survey. In: *Annals of Software Engineering* 10 (2000), Nr. 1-4, S. 177–205. – ISSN 1022–7091
- [BB01] BOEHM, Barry W. ; BASILI, Victor R.: Software Defect Reduction Top 10 List. In: *IEEE Computer* 34 (2001), Nr. 1, S. 135–137
- [BCR94] BASILI, Victor ; CALDIERA, Gianluigi ; ROMBACH, Dieter: The goal question metric approach. In: MARCINIAK, J. J. (Hrsg.): *Encyclopedia of Software Engineering.* John Wiley & Sons Inc, 1994. – ISBN 978–0471540014, S. 528–532
- [BD08] BUNDSCHEUH, Manfred ; DEKKERS, Carol: *The IT measurement compendium: Estimating and benchmarking success with functional size measurement.* Springer, 2008. – ISBN 978–3540681885
- [BDC14] BDC: *Acceptance Monitor 2014: Studienergebnisse.* [http://www.bdc.at/fileadmin/downloads/Studienergebnisse-Acceptance\\_Monitor\\_2014.pdf](http://www.bdc.at/fileadmin/downloads/Studienergebnisse-Acceptance_Monitor_2014.pdf). Version: 2014, Abruf: 04.10.2014
- [BF04] BUNDSCHEUH, Manfred ; FABRY, Axel: *Aufwandschätzung von IT-Projekten: Zeit und Kosten im Griff ; Planungssicherheit durch zuverlässige Schätzung*

- ; *Function Point und andere Methoden*]. 2. mitp, 2004. – ISBN 978–3826605345
- [BM10] BATH, Graham ; MCKAY, Judy: *Praxiswissen Softwaretest - Test Analyst und Technical Test Analyst: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard*. 1. dpunkt.verlag GmbH, 2010. – ISBN 978–3898645911
- [Boe79] BOEHM, Barry W.: Guidelines for Verifying and Validating Software Requirements and Design Specifications. In: SAMET, P. A. (Hrsg.): *Euro IFIP 79*. North-Holland Pub. Co. and Sole distributors for the USA and Canada, Elsevier North-Holland, 1979. – ISBN 978–0444853707, S. 711–719
- [Boe81] BOEHM, Barry W.: *Software engineering economics*. Prentice-Hall, 1981. – ISBN 0–13–822122–7
- [Boe88] BOEHM, Barry W.: A spiral model of software development and enhancement. In: *Computer* 21 (1988), Nr. 5, S. 61–72. – ISSN 0018–9162
- [Boe00] BOEHM, Barry W.: *Software cost estimation with Cocomo II*. Prentice Hall, 2000
- [BR88] BASILI, V. R. ; ROMBACH, H. D.: The TAME project: towards improvement-oriented software environments. In: *IEEE Transactions on Software Engineering* 14 (1988), Nr. 6, S. 758–773. – ISSN 0098–5589
- [Bro95] BROOKS, Frederick P.: *The Mythical man-month: Essays on software engineering*. Anniversary ed. with four new chapters. Addison-Wesley, 1995. – ISBN 978–0201835953
- [BRS<sup>+</sup>07] BASILI, Victor R. (Hrsg.) ; ROMBACH, H. Dieter (Hrsg.) ; SCHNEIDER, Kurt (Hrsg.) ; KITCHENHAM, Barbara A. (Hrsg.) ; PFAHL, Dietmar (Hrsg.) ; SELBY, Richard W. (Hrsg.): *Empirical Software Engineering Issues. Critical Assessment and Future Directions, International Workshop, Dagstuhl Castle, Germany, June 26-30, 2006. Revised Papers*. Bd. 4336. Springer, 2007 (Lecture Notes in Computer Science). – ISBN 978–3540713005
- [BS10] BORTZ, Jürgen ; SCHUSTER, Christof: *Statistik für Human- und Sozialwissenschaftler: Mit ... 163 Tabellen*. 7. Springer, 2010. – ISBN 978–3642127694
- [BT13] BUENEN, Mark ; TEJE, Makarand: *World Quality Report 2013-14*. 2013

## Literaturverzeichnis

- [Büh11] BÜHNER, Markus: *Einführung in die Test- und Fragebogenkonstruktion*. 3. Pearson Studium, 2011 (PS Psychologie). – ISBN 978–3868940336
- [CDM02] CANGUSSU, J.W ; DECARLO, R.A ; MATHUR, A.P: A formal model of the software test process. In: *IEEE Transactions on Software Engineering* 28 (2002), Nr. 8, S. 782–796
- [Coc03] COCKBURN, Alistair: *Use Cases effektiv erstellen: [das Fundament für gute Software-Entwicklung, Geschäftsprozesse mit Use Cases modellieren, die Regeln für Use Cases sicher beherrschen]*. 1. mitp, 2003. – ISBN 978–3826613449
- [Coh06] COHN, Mike: *Agile estimating and planning*. Prentice Hall Professional Technical Reference, 2006 (Robert C. Martin series). – ISBN 978–0131479418
- [Coh10] COHN, Mike: *Agile Softwareentwicklung: Mit Scrum zum Erfolg!* 1. Addison Wesley, 2010 (Sonstige Bücher AW). – ISBN 978–3827329875
- [CYT<sup>+</sup>96] COLLOFELLO, J.S ; YANG, Zhen ; TVEDT, J.D ; MERRILL, D. ; RUS, I.: Modeling software testing processes. In: *Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications, 1996*, 1996, S. 289–293
- [Dal67] DALKEY, Norman C. ; RAND CORPORATION (Hrsg.): *Delphi*. <http://www.rand.org/content/dam/rand/pubs/papers/2006/P3704.pdf>. Version: 1967, Abruf: 11.09.2013
- [DeM82] DEMARCO, Tom: *Controlling software projects: Management, measurement & estimation*. Yourdon Press, 1982. – ISBN 978–0131717114
- [DH06] DOWIE, Ulrike ; HERZWURM, Georg: Teststeuerung und -schätzung im Umfeld dynamischer Anforderungen. In: *Multikonferenz Wirtschaftsinformatik 2006*. Gito, 2006. – ISBN 3–936771–62–6, S. 249–264
- [DL99] DEMARCO, Tom ; LISTER, Timothy: *Wien wartet auf Dich! Der Faktor Mensch im DV-Management*. 2. Hanser, 1999. – ISBN 978–3446212770
- [Dow09] DOWIE, Ulrike: *Testaufwandsschätzung in der Softwareentwicklung: Modell der Einflussfaktoren und Methode zur organisationsspezifischen Aufwandschätzung*. 1. Eul, 2009. – ISBN 978–3899367898

- [Dum01] DUMKE, Reiner: *Software-Engineering: Eine Einführung für Informatiker und Ingenieure: Systeme, Erfahrungen, Methoden, Tools.* 3. Vieweg, 2001.  
– ISBN 978-3528253554
- [Dum03] DUMKE, Reiner: *Software engineering: Eine Einführung für Informatiker und Ingenieure ; Systeme, Erfahrungen, Methoden, Tools.* 4. Vieweg, 2003.  
– ISBN 978-3528353551
- [ED07] EBERT, Christof ; DUMKE, Reiner: Best Practices in Software Measurement: Establish - Extract - Evaluate - Execute. In: *Software measurement* (2007)
- [edJ09] E SILVA, Daniel G. ; DE ABREU, Bruno T. ; JINO, Mario: A Simple Approach for Estimation of Execution Effort of Functional Test Cases. In: *International Conference on : Software Testing Verification and Validation, 2009. ICST '09*, IEEE Computer Society, 2009, S. 289–298
- [Fro08] FROHNHOFF, Stephan: Große Softwareprojekte. In: *Informatik-Spektrum* 31 (2008), S. 566–575. – ISSN 0170–6012
- [Fro09] FROHNHOFF, Stephan: *Use Case Points 3.0: Implementierung einer Use Case bezogenen Schätzmethode für das Software-Engineering betrieblicher Informationssysteme*, Universität Paderborn, Diss., 2009
- [GE06] GALORATH, Daniel D. ; EVANS, Michael W.: *Software sizing, estimation, and risk management: When performance is measured performance improves.* Auerbach Publications, 2006. – ISBN 0-8493-3593-0
- [GG12] GEISEN, Silke ; GÜLDALI, Baris: Agiles Testen in Scrum – Testtypen und Abläufe. In: *OBJEKTSPEKTRUM (Online Themenspecials)* (2012), Oktober, Nr. Agility/2012, S. 1–4
- [Gooa] GOOGLE: *Google*. <http://www.google.de/>, Abruf: 27.09.2014
- [Goob] GOOGLE: *Google Scholar*. <http://scholar.google.de/>, Abruf: 27.09.2014
- [Gre02] GREENING, James: *Planning Poker: How to avoid analysis paralysis while release planning.* <http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.pdf>. Version: 2002, Abruf: 11.09.2013
- [GTB10] GTB WORKING PARTY GLOSSARY ; GERMAN TESTING BOARD E.V. (Hrsg.) ; HAMBURG, Matthias (Hrsg.) ; HEHN, Uwe (Hrsg.): *ISTQB/GTB Standardglossar der Testbegriffe: Deutsch/Englisch: Version 2.1.* 30. September 2010

## Literaturverzeichnis

- [Gü] GÜLDALI, Baris: *Model-Based Testing Community.* <http://model-based-testing.info/2013/05/19/online-survey-on-test-effort-estimation/>, Abruf: 02.01.2014
- [HJE<sup>+</sup>14] HAUPTMANN, Benedikt ; JUNKER, Maximilian ; EDER, Sebastian ; AMANN, Christian ; VAAS, Rudolf: An expert-based cost estimation model for system test execution. In: *International Conference on Software and Systems Process 2014, ICSSP '14, Nanjing, China - May 26 - 28, 2014*, 2014. – ISBN 978–1450327541, S. 159–163
- [Hos14] HOSSEINI, Seyed Morteza: *A Model for Determining the Cost of Test Case A Model for Determining the Cost of Test Case Execution (or Test Execution Effort Estimation Model)*, School of Innovation, Design and Engineering, Masterarbeit, 2014
- [HR58] HELMER, Olaf ; RESCHER, Nicholas ; RAND CORPORATION (Hrsg.): *ON THE EPISTMELOGY OF THE INEXACT SCIENCES*. <http://www.rand.org/content/dam/rand/pubs/papers/2005/P1513.pdf>. Version: 1958, Abruf: 11.09.2013
- [Hum11] HUMMEL, Oliver: *Aufwandsschätzungen in der Software- und Systementwicklung kompakt*. Spektrum, 2011. – ISBN 978–3827427526
- [IBM] IBM: *SPSS*. <http://www-01.ibm.com/software/de/analytics/spss/>, Abruf: 12.05.2014
- [IBS] IBSG: *International Software Benchmarking Standards Group*. <http://www.isbsg.org/>, Abruf: 12.11.2013
- [IEE] IEEE: *IEEE Xplore Digital Library*. <http://ieeexplore.ieee.org/Xplore/home.jsp>, Abruf: 27.09.2014
- [IEE86] IEEE: *IEEE Standard for Software Unit Testing: IEEE Std 1008-1987*. Institute of Electrical and Electronics Engineers, 1986. – ISBN 1–55937–672–4
- [IEE91] IEEE: IEEE Standard Computer Dictionary. A Compilation of IEEE Standard Computer Glossaries: IEEE Std 610. In: *IEEE Std 610* (1991), S. 1
- [IEE98a] IEEE: Industry implementation of International Standard ISO/IEC 12207: 1995. (ISO/IEC 12207 standard for information technology - software life

- cycle processes - implementation considerations: IEEE/EIA 12207.2-1997. In: *IEEE/EIA 12207.2-1997* (1998)
- [IEE98b] IEEE: Industry implementation of International Standard ISO/IEC 12207: 1995. (ISO/IEC 12207) Standard for Information Technology - Software life cycle processes - Life cycle data: IEEE/EIA 12207.1-1997. In: *IEEE/EIA 12207.1-1997* (1998)
- [IEE08] IEEE: IEEE Standard for Software and System Test Documentation: IEEE Std 829-2008. In: *IEEE Std 829-2008* (2008), S. 1–118
- [IEE11a] IEEE: IEEE Guide—Adoption of the Project Management Institute (PMI(R)) Standard A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide)—Fourth Edition: IEEE Std 1490-2011. In: *IEEE Std 1490-2011* (2011), S. 1–508
- [IEE11b] IEEE: *ISO/IEC/IEEE 29148: Systems and software engineering: Life cycle processes : requirements engineering = Ingénierie des systèmes et du logiciel : processus de cycle de vie : ingénierie des exigences*. First edition, 2011-12-01. ISO and IEC and Institute of Electrical and Electronics Engineers, 2011. – ISBN 978-0738165912
- [Int10] INTERNATIONAL FUNCTION POINT USERS GROUP (IFPUG): *Function Point Counting Practices Manual: Release 4.3.1*. IFPUG, 2010. – ISBN 978-0975378342
- [Int11] INTERNATIONAL SOFTWARE TESTING QUALIFICATION BOARD: *Certified Tester: Foundation Level Syllabus*. <http://www.istqb.org/downloads/finish/16/15.html>. Version: 2011, Abruf: 18.01.2014
- [Int12a] INTERNATIONAL SOFTWARE TESTING QUALIFICATION BOARD: *Certified Tester: Advanced Level Syllabus Technical Test Analyst*. <http://www.istqb.org/downloads/finish/46/94.html>. Version: 2012, Abruf: 25.06.2014
- [Int12b] INTERNATIONAL SOFTWARE TESTING QUALIFICATION BOARD: *Certified Tester: Advanced Level Syllabus Test Analyst*. <http://www.istqb.org/downloads/finish/46/95.html>. Version: 2012, Abruf: 25.06.2014
- [Int12c] INTERNATIONAL SOFTWARE TESTING QUALIFICATION BOARD: *Certified Tester: Advanced Level Syllabus Test Manager*. <http://www.istqb.org/downloads/finish/46/96.html>. Version: 2012, Abruf: 18.01.2014

## Literaturverzeichnis

- [ISO85] ISO: *ISO 5807:1985 Information processing - Documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts.* 1985
- [ISO01] ISO/IEC: *ISO/IEC 9126-1:2001 Software engineering - Product quality - Part 1: Quality model.* 2001
- [ISO09] ISO/IEC: *ISO/IEC 20926:2009 Software and systems engineering - Software measurement - IFPUG functional size measurement method 2009.* 2009
- [ISO11a] ISO/IEC: *ISO/IEC 19761:2011 Software engineering - COSMIC: a functional size measurement method.* 2011
- [ISO11b] ISO/IEC: *ISO/IEC 25010:2011 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.* 2011
- [IST] ISTQB: Startseite International Software Testing Qualification Board. <http://www.istqb.org/>, Abruf: 28.07.13
- [JTC07] JTC 1/SC 7 SOFTWARE AND SYSTEMS ENGINEERING: *Systems and software engineering - Measurement process.* 2007
- [Kar93a] KARNER, Gustav: Resource Estimation for Objectory Projects. In: *Objective Systems SF AB* (1993)
- [Kar93b] KARNER, Gustav: *Metrics for Objectory*, University of Linköping, Schweden, Diplomarbeit, Dezember 1993. – Nr. LiTHIDA-Ex-9344:21
- [KC07] KITCHENHAM, Barbara ; CHARTERS, Stuart: *Guidelines for performing Systematic Literature Reviews in Software Engineering.* 2007
- [Kit04] KITCHENHAM, Barbara ; KEELE UNIVERSITY (Hrsg.): *Procedures for Performing Systematic Reviews.* 2004
- [KK03] KROLL, Per ; KRUCHTEN, Philippe: *The rational unified process made easy: A practitioner's guide to the RUP.* Addison-Wesley, 2003 (Addison-Wesley object technology series). – ISBN 0-321-16609-4
- [KM08] KUSHWAHA, Dharmender Singh ; MISRA, Arun Kumar: Software Test Effort Estimation. In: *ACM SIGSOFT Software Engineering Notes* 33 (2008), Nr. 3, S. 6:1–6:5. – ISSN 0163-5948

- [KS13] KUMARI, Abhilasha ; SHARMA, Ashish: Test effort estimation in regression testing. In: *IEEE International Conference in MOOC : Innovation and Technology in Education (MITE), 2013*, IEEE Computer Society, 2013. – ISBN 978–1479916252, S. 343–348
- [LP92] LEDERER, Albert L. ; PRASAD, Jayesh: Nine management guidelines for better cost estimating. In: *Communications of the ACM* 35 (1992), Nr. 2, S. 51–59. – ISSN 0001–0782
- [MAC05] MOHAGHEGHI, Parastoo ; ANDA, Bente ; CONRADI, Reidar: Effort estimation of use cases for incremental large-scale software development. In: *Proceedings of the 27th international conference on Software engineering - ICSE '05*, ACM Press, 2005. – ISBN 1–5959–3963–2, S. 303–311
- [McC06] MCCONNELL, Steve: *Aufwandschätzung für Softwareprojekte*. Microsoft Press, 2006. – ISBN 3–86645–612–3
- [Mic11] MICHAEL MLYNARSKI: *Holistic Use of Analysis Models in Model-Based System Testing*, Universität Paderborn, Diss., 2011
- [MSTK02] MIZUNO, Osamu ; SHIGEMATSU, Eijiro ; TAKAGI, Yasunari ; KIKUNO, Tohru: On estimating testing effort needed to assure field quality in software development. In: *13th International Symposium on Software Reliability Engineering (ISSRE 2002), 12-15 November 2002, Annapolis, MD, USA*, IEEE Computer Society, 2002. – ISBN 0–7695–1763–3, S. 139–146
- [Mum08] MUMMENDEY, Hans Dieter: *Die Fragebogen-Methode*. 5. Hogrefe, 2008. – ISBN 978–3801719487
- [Nag01] NAGESWARAN, Suresh: *Test Effort Estimation Using Use Case Points*. 2001
- [Nel12] NELISSEN, Guido ; SOGETI NEDERLAND B.V. (Hrsg.): *MANUAL TEST CASE POINTS*. 2012
- [NPL13] NGUYEN, Vu ; PHAM, Vu ; LAM, Vu: qEstimation: A Process for Estimating Size and Effort of Software Testing. In: *Proceedings of the 2013 International Conference on Software and System Process*, ACM, 2013 (ICSSP 2013). – ISBN 978–1450320627, S. 20–28
- [Obj08] OBJECT MANAGEMENT GROUP: *Software & Systems Process Engineering Meta-Model Specification: Version 2.0*. <http://www.omg.org/spec/SPEM/2.0/PDF>. Version: April 2008, Abruf: 24.10.2013

## Literaturverzeichnis

- [Obj11a] OBJECT MANAGEMENT GROUP: *OMG Unified Modeling Language (OMG UML), Superstructure: Version 2.4.1.* <http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF>. Version: August 2011, Abruf: 24.10.2013
- [Obj11b] OBJECT MANAGEMENT GROUP: *Business Process Model and Notation (BPMN): Version 2.0.* <http://www.omg.org/spec/BPMN/2.0/PDF>. Version: Januar 2011, Abruf: 24.10.2013
- [Par57] PARKINSON, C. N.: *Parkinson's Law and Other Studies in Administration. Illustrated by Robert C. Osborn.* Houghton Mifflin, 1957
- [PM92] PUTNAM, Lawrence H. ; MYERS, Ware: *Measures for excellence: Reliable software on time, within budget.* Yourdon Press, 1992. – ISBN 0–135–67694–0
- [PR11] POHL, Klaus ; RUPP, Chris: *Requirements engineering fundamentals: A study guide for the Certified Professional for Requirements Engineering exam : foundation level, IREB compliant.* 1. Rocky Nook, 2011 (Rocky Nook computing). – ISBN 978–1933952819
- [Pre01] PRECHELT, Lutz: *Kontrollierte Experimente in der Softwaretechnik: Potenzial und Methodik.* Springer, 2001. – ISBN 3–540–41257–3
- [Pro] PROJECT, R: *The R Project for Statistical Computing.* <http://www.r-project.org>, Abruf: 12.05.2014
- [QSM] QSM: *Quantitative Software Management.* <http://www.qsm.com/>, Abruf: 21.11.2013
- [Res02] RESEARCH TRIANGLE INSTITUTE: *The Economic Impacts of Inadequate Infrastructure for Software Testing: Final Report.* <http://www.nist.gov/director/planning/upload/report02-3.pdf>. Version: 2002
- [Roh78] ROHRMANN, Bernd: Empirische Studien zur Entwicklung von Antwortskalen für die sozialwissenschaftliche Forschung. In: *Zeitschrift für Sozialpsychologie* (1978), Nr. 9, S. 222–245
- [RQ07] RUPP, Chris ; QUEINS, Stefan: *UML 2 glasklar: Praxiswissen für die UML-Modellierung.* 3., aktualisierte Aufl. Hanser, 2007. – ISBN 978–3446411180
- [RR06] ROBERTSON, Suzanne ; ROBERTSON, James: *Mastering the requirements process.* 2. Addison-Wesley, 2006. – ISBN 0–321–41949–9

- [Rup13] RUPP, Chris: *Systemanalyse kompakt*. 3. Springer Vieweg, 2013 (IT kompakt). – ISBN 978–3642354465
- [San] SANCAR, Yavuz: *Gruppe Agiles Testmanagement*. <https://www.xing.com/net/pri4956bax/agiletm/>, Abruf: 02.01.2014
- [Sau11] SAUER, Stefan: *Systematic development of model-based software engineering methods*, Universität Paderborn, Diss., 2011
- [Sch] SCHMITZ, Carsten: *LimeSurvey*. <http://www.limesurvey.org/de>, Abruf: 19.12.2013
- [SJ06] SNEED, Harry M. ; JUNGMAYR, Stefan: Produkt- und Prozessmetriken für den Softwaretest. In: *Informatik-Spektrum* 29 (2006), Nr. 1, S. 23–38. – ISSN 0170–6012
- [SK11] SHARMA, Ashish ; KUSHWAHA, Dharmender Singh: A metric suite for early estimation of software testing effort using requirement engineering document and its validation. In: *2nd International Conference on Computer and Communication Technology (ICCCT), 2011*, IEEE Computer Society, 2011. – ISBN 978–1457713859, S. 373–378
- [SK12] SHARMA, Ashish ; KUSHWAHA, Dharmender Singh: Applying requirement based complexity for the estimation of software development and testing effort. In: *ACM SIGSOFT Software Engineering Notes* 37 (2012), Nr. 1, S. 1–11. – ISSN 0163–5948
- [SL10] SPILLNER, Andreas ; LINZ, Tilo: *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester - Foundation Level nach ISTQB-Standard*. 4., überarb. und aktualisierte Aufl. Dpunkt, 2010. – ISBN 978–3898646420
- [Smi99] SMITH, John ; RATIONAL SOFTWARE (Hrsg.): *The Estimation of Effort Based on Use Cases*. Oktober 1999
- [Sne05] SNEED, Harry M.: *Software-Projektkalkulation: Praxiserprobte Methoden der Aufwandsschätzung für verschiedene Projektarten*. Hanser, 2005. – ISBN 978–3446400054
- [Soga] SOGETI: Tool "Estimating the test effort". [http://www.tmap.net/sites/tmap.net/files/attachments/Tool\\_\\_Estimating\\_the\\_test\\_effort\\_.xls](http://www.tmap.net/sites/tmap.net/files/attachments/Tool__Estimating_the_test_effort_.xls), Abruf: 25.09.2013

## Literaturverzeichnis

- [Sogb] SOGETI: *Tool "Test Point Analysis"*. [http://www.tmap.net/sites/tmap.net/files/attachments/Tool\\_\\_Test\\_Point\\_Analysis\\_.xls](http://www.tmap.net/sites/tmap.net/files/attachments/Tool__Test_Point_Analysis_.xls), Abruf: 25.09.2013
- [Spi11] SPILLNER, Andreas: *Praxiswissen Softwaretest - Testmanagement: Aus- und Weiterbildung zum certified Tester ; advanced Level nach ISTQB-Standard*. 3. dpunkt.verlag GmbH, 2011. – ISBN 978–3898645577
- [Spr] SPRINGER: *Springer Link*. <http://link.springer.com/>, Abruf: 27.09.2014
- [Sti06] STIKKEL, Gábor: Dynamic model for the system testing process. In: *Information and Software Technology* 48 (2006), Nr. 7, S. 578–585. – ISSN 0950–5849
- [SVWH12] SPILLNER, Andreas ; VOSSEBERG, Karin ; WINTER, Mario ; HABERL, Peter: *Umfrage 2011: Softwaretest in der Praxis*. 1. dpunkt.verlag GmbH, 2012
- [Thi12] THIESSEN, Aron: *Empirische Untersuchung von Testaufwandsschätzungen zur Identifikation von Testeinflussfaktoren*, Fachhochschule der Wirtschaft, Bachelorarbeit, 2012
- [UL07] UTTING, Mark ; LEGEARD, Bruno: *Practical model-based testing: A tools approach*. Morgan Kaufmann Publishers, 2007. – ISBN 978–0123725011
- [Van09] VANHOVE, Geert: *Test estimation and metrics*. <http://www.sogeti.be/beluxconference09/documents/Test2.pdf>. Version: 2009, Abruf: 30.09.2013
- [van11] VAN EWINK, Alexander: *TPI NEXT: Geschäftsbasierte Verbesserung des Testprozesses*. 1. dpunkt.verlag GmbH, 2011. – ISBN 978–3898646857
- [vBKV08] VAN DER AALST, Leo ; BROEKMAN, Bart ; KOOMEN, Tim ; VROON, Michiel: *TMap Next: Ein praktischer Leitfaden für ergebnisorientiertes Softwaretesten*. 1. dpunkt.verlag GmbH, 2008. – ISBN 978–3898644617
- [vD99] VAN VEENENDAAL, Erik ; DEKKERS, Ton: *Testpointanalysis: a method for test estimation*. 1999
- [Voi09] VOIGT, Hendrik: *Kontextsensitive Qualitätsplanung von Softwaremodellen*, Universität Paderborn, Diss., 2009
- [vW10] VON DER MASSEN, Thomas ; WÜBBEKE, Andreas: Verteiltes Testen heterogener Systemlandschaften bei arvato services. In: ENGELS, Gregor (Hrsg.) ; LUCKEY, Markus (Hrsg.) ; SCHÄFER, Wilhelm (Hrsg.): *Software Engineering*

*ring 2010 - Fachtagung des GI-Fachbereichs Softwaretechnik, 22.-26.2.2010  
in Paderborn Bd. 159, GI, 2010 (LNI). – ISBN 978-3885792536, S. 17–18*

- [WEI09] WEIT e.V.: *V-Modell XT, Version 1.3.* 2009
- [Win99] WINTER, Mario: *Qualitätssicherung für objektorientierte Software: Anforderungsermittlung und Test gegen die Anforderungsspezifikation*, FernUniversität - Gesamthochschule - in Hagen, Diss., 1999
- [Wol74] WOLVERTON, R.W: The Cost of Developing Large-Scale Software. In: *IEEE Transactions on Computers* 23 (1974), Nr. 6, S. 615–636. – ISSN 0018-9340
- [ZZH<sup>+</sup>08] ZHU, Xiaochun ; ZHOU, Bo ; HOU, Li ; CHEN, Junbo ; CHEN, Lu: An Experience-Based Approach for Test Execution Effort Estimation. In: *Proceedings of the 9th International Conference for Young Computer Scientists, ICYCS 2008, Zhang Jia Jie, Hunan, China, November 18-21, 2008*, IEEE Computer Society, 2008, S. 1193–1198
- [ZZW<sup>+</sup>08] ZHU, Xiaochun ; ZHOU, Bo ; WANG, Fan ; QU, Yi ; CHEN, Lu: Estimate Test Execution Effort at an Early Stage: An Empirical Study. In: *Proceedings of the International Conference on Cyberworlds 2008, Hangzhou, China, 22-24 September 2008*, IEEE Computer Society, 2008. – ISBN 978-0769533810, S. 195–200



# **A. Expertenumfrage**

Im Folgenden wird der Fragebogen in Auszügen vorgestellt. Anschließend werden die Detailergebnisse für die Fragen zur Erfahrung des Probanden und bzgl. des Modells für Testeinflussfaktoren aufgeführt. Der zweite Block an Fragen zur Testfreigabebewertung wird in dieser Dissertation nicht weiter betrachtet.

## **A.1. Fragebogen**

Die Umfrage wurde online über das Open-Source-Werkzeug LimeSurvey [Sch] als Fragebogen realisiert. In den nachfolgenden Abbildungen sind exemplarisch die Startseite der Umfrage, eine Seite mit Fragen zur Erfahrung des Probanden und eine Seite mit Fragen zu einem Testeinflussfaktor.

## A. Expertenumfrage

 LimeSurvey Studie zur Testaufwandsschätzung und (Test-) Freigabebewertung

Wir bedanken uns, dass Sie an dieser Studie des s-lab der Universität Paderborn teilnehmen. Es wäre schön, wenn Sie sich ca. 20-30 Minuten Zeit nehmen könnten, die folgenden Fragen zum Thema Testaufwandsschätzung und Freigabebewertung zu beantworten.

Ziel der Studie ist es, die Testaufwandsschätzung und den Prozess der Freigabebewertung für Testprojekte zu verbessern. Davon können Sie kurzfristig durch einen Bericht über die Ergebnisse der Studie (ca. August '13) und mittelfristig durch Teilhabe an unseren Forschungsergebnissen (ca. Dezember '13) profitieren. Bei Rückfragen können Sie sich gerne an uns, Hendrik Schreiber ([hschreiber@s-lab.uni-paderborn.de](mailto:hschreiber@s-lab.uni-paderborn.de)) oder an Yavuz Sancar ([ysancar@s-lab.uni-paderborn.de](mailto:ysancar@s-lab.uni-paderborn.de)) wenden.

Gehen Sie bei der Beantwortung der Fragen bitte wie folgt vor:

- Bitte lesen Sie die Fragen aufmerksam durch, da sich die Fragen stark ähneln.
- Bitte schauen Sie im Glossar der ISTQB (z.B. beim [German Testing Board](#)) nach, wenn Ihnen ein Begriff nicht eindeutig ist.
- Antworten Sie aufrichtig und nach Ihrer persönlichen Erfahrung. Es gibt keine falschen Antworten und die Anonymität Ihrer Antworten wird garantiert.
- Bitte beantworten Sie die Fragen zügig und lassen Sie keine Frage aus.
- Bitte im Zweifelsfall die Antwort angekreuzen, die am ehesten zutrifft.

Wir bedanken uns für Ihre Unterstützung! Unter allen Teilnehmern, die Ihre E-Mailadresse angeben, verlosen wir zwei Amazon Gutscheine über 25 Euro.

Abbildung A.1.: Expertenumfrage: Startseite der Umfrage

\* In Ihrem Unternehmen werden ...  
Bitte wählen Sie eine der folgenden Antworten:

sehr unterschiedliche Projekte durchgeführt  
 überwiegend gleichartige Projekte durchgeführt  
 weiß ich nicht

\* Existiert in Ihrem Unternehmen eine selbstständige Qualitätssicherungsgruppe/-abteilung?

Ja       Nein

Abbildung A.2.: Expertenumfrage: Exemplarische Seite für Fragen zur Berufserfahrung des Probanden

## A.1. Fragebogen

Diese und die nächsten Fragegruppen sind ähnlich aufgebaut. Es wird zunächst nach der Stärke des Einflusses des jeweiligen Testeinflussfaktors gefragt. Dabei soll nach Gefühl in Prozent angegeben werden, um wie viel sich der Testaufwand bei einer bestimmten Bewertung (z.B. "Sehr hoch") verringert "-" oder erhöht ("+").

Lesen Sie bitte hierbei immer die Beispiele für die Bewertung bevor Sie antworten, da die Bewertung "Sehr hoch" bei manchen Testeinflussfaktoren eine positive und bei anderen eine negative Auswirkung zur Folge hat.

- \* Wie stark beeinflusst die Stabilität der Testumgebung den Testaufwand?

Beispiele für die Bewertung der Stabilität der Testumgebung:

- **Außergewöhnlich geringe Stabilität:**
  - Seltener bis nie verfügbar, keine vorherige Verwendung der Umgebung
- **Sehr hohe Stabilität:**
  - Immer verfügbar, mehrmalige vorherige Verwendung der Umgebung

	- 40%	- 20%	- 10%	- 5%	0	+ 5%	+ 10%	+ 20%	+ 40%
Außergewöhnlich geringe Stabilität	<input type="radio"/>								
Sehr geringe Stabilität	<input type="radio"/>								
Eher geringe Stabilität	<input type="radio"/>								
Eher hohe Stabilität	<input type="radio"/>								
Sehr hohe Stabilität	<input type="radio"/>								

? "-" bedeutet Verringerung des Testaufwandes und "+" bedeutet Erhöhung des Testaufwandes; Merkmale: Reife der Testumgebung, Verfügbarkeit Testumgebung.

- \* Hat die Stabilität der Testumgebung relevanten Einfluss auf die nachfolgenden Testaktivitäten?

	keinesfalls	wahrscheinlich	vielleicht	ziemlich wahrscheinlich	ganz sicher
Testmanagement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testdesign	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Testausführung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

? Testdesign umfasst Analyse, Design und Realisierung - Testausführung umfasst Durchführung und Bewertung

- \* Wie häufig ist in Ihren Projekten die Stabilität der Testumgebung während der nachfolgenden Projektphasen gut einschätzbar?

	nie	selten	gelegentlich	oft	immer
Anforderungsphase	<input type="radio"/>				
Designphase	<input type="radio"/>				
Entwicklungsphase	<input type="radio"/>				

? Allgemeiner formuliert: Wie häufig ist eine Einschätzung des Testeinflussfaktors mit guter Zuverlässigkeit (Reliabilität) in der jeweiligen Projektphase möglich?

Abbildung A.3.: Expertenumfrage: Exemplarische Seite für einen Testeinflussfaktor. Gezeigt wird hier die Seite für den Testeinflussfaktor die Stabilität der Testumgebung.

## A. Expertenumfrage

### A.2. Detailergebnisse

Die Detailergebnisse werden getrennt nach den in Abschnitt 5.3.1.1 definierten Gruppen aufgeführt. Insgesamt haben 92 Probanden die Umfrage begonnen. Davon haben 40 Probanden die Umfrage vollständig abgeschlossen. In den nachfolgenden Abbildungen werden nur die vollständig abgeschlossenen Umfragen berücksichtigt. Bei Fragen, die nicht von allen Probanden beantwortet worden sind, wird die abweichende Anzahl von Probanden angegeben.

#### A.2.1. Auswertung der Fragen zu der Berufserfahrung des Probanden

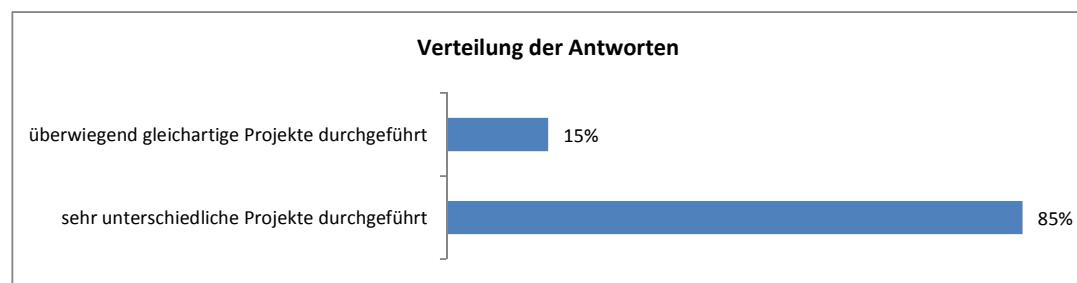


Abbildung A.4.: Expertenumfrage: Wie gleichartig sind die Projekte im Unternehmen des Probanden?

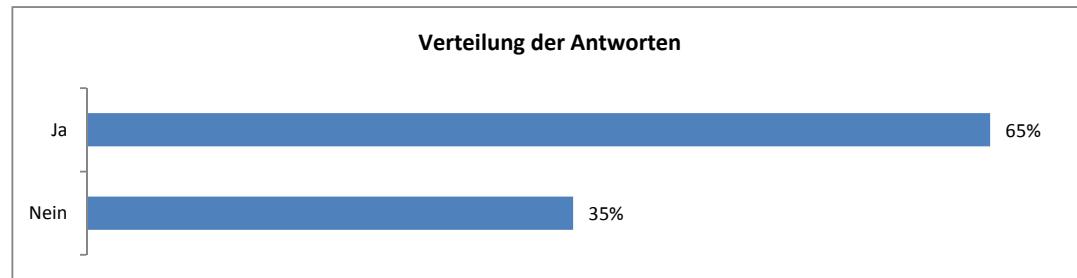


Abbildung A.5.: Expertenumfrage: Existiert in dem Unternehmen des Probanden eine selbstständige Qualitätssicherungsgruppe/-abteilung?

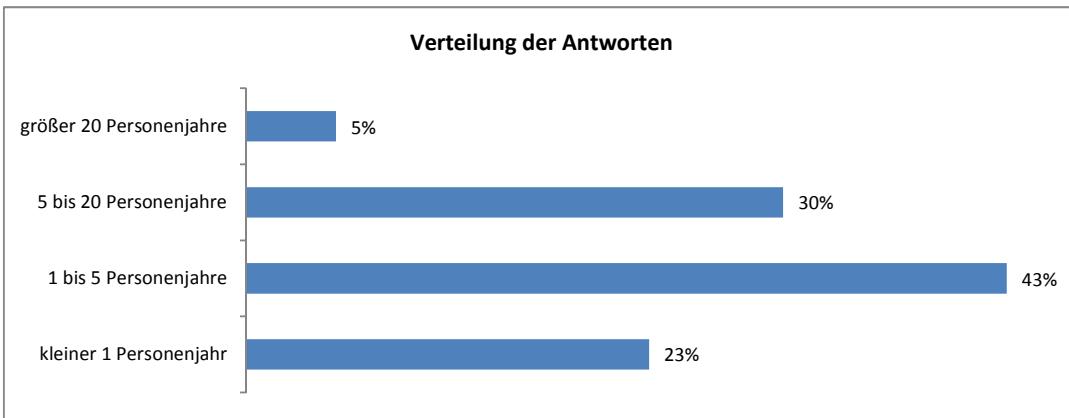


Abbildung A.6.: Expertenumfrage: Welchen Größenklassen sind den durchgeführten Projekte des Probanden zugeordnet?

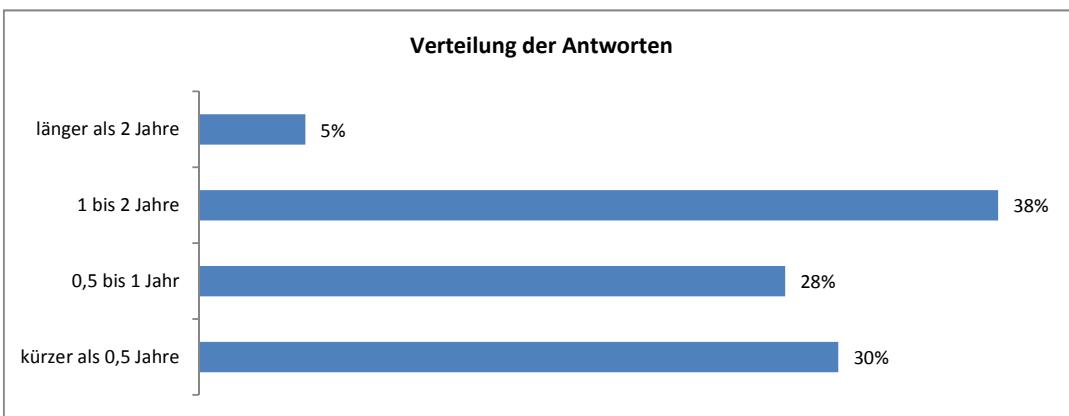


Abbildung A.7.: Expertenumfrage: Wie lange dauern die durchgeföhrten Projekte des Probanden durchschnittlich?

## A. Expertenumfrage

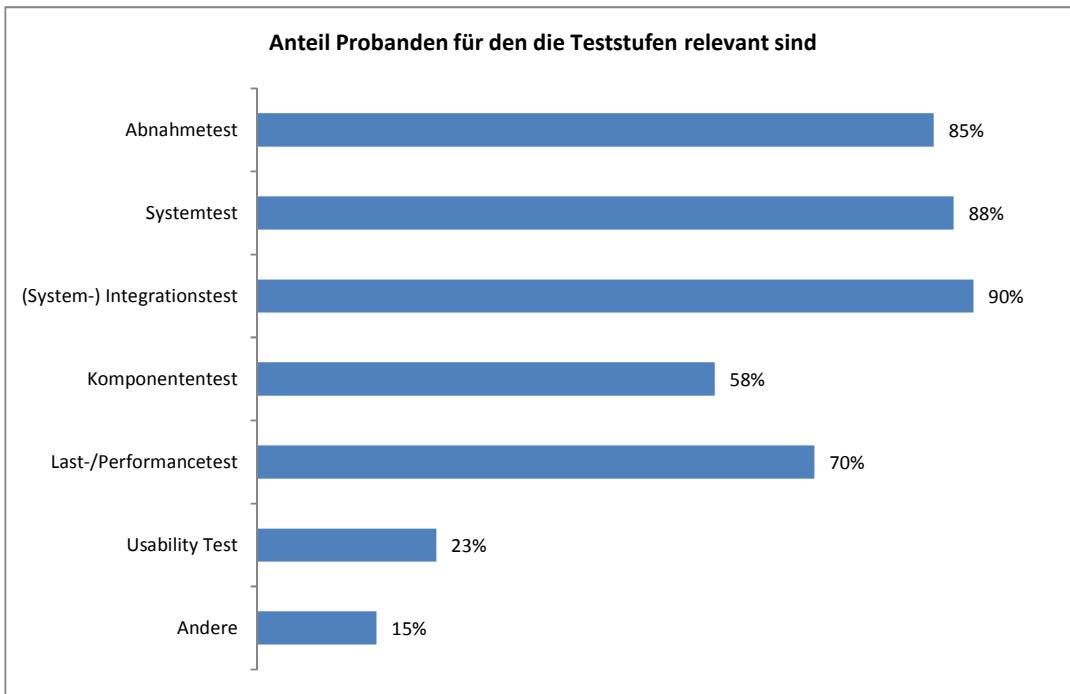


Abbildung A.8.: Expertenumfrage: Welche Teststufen sind in den Projekten des Probanden relevant?

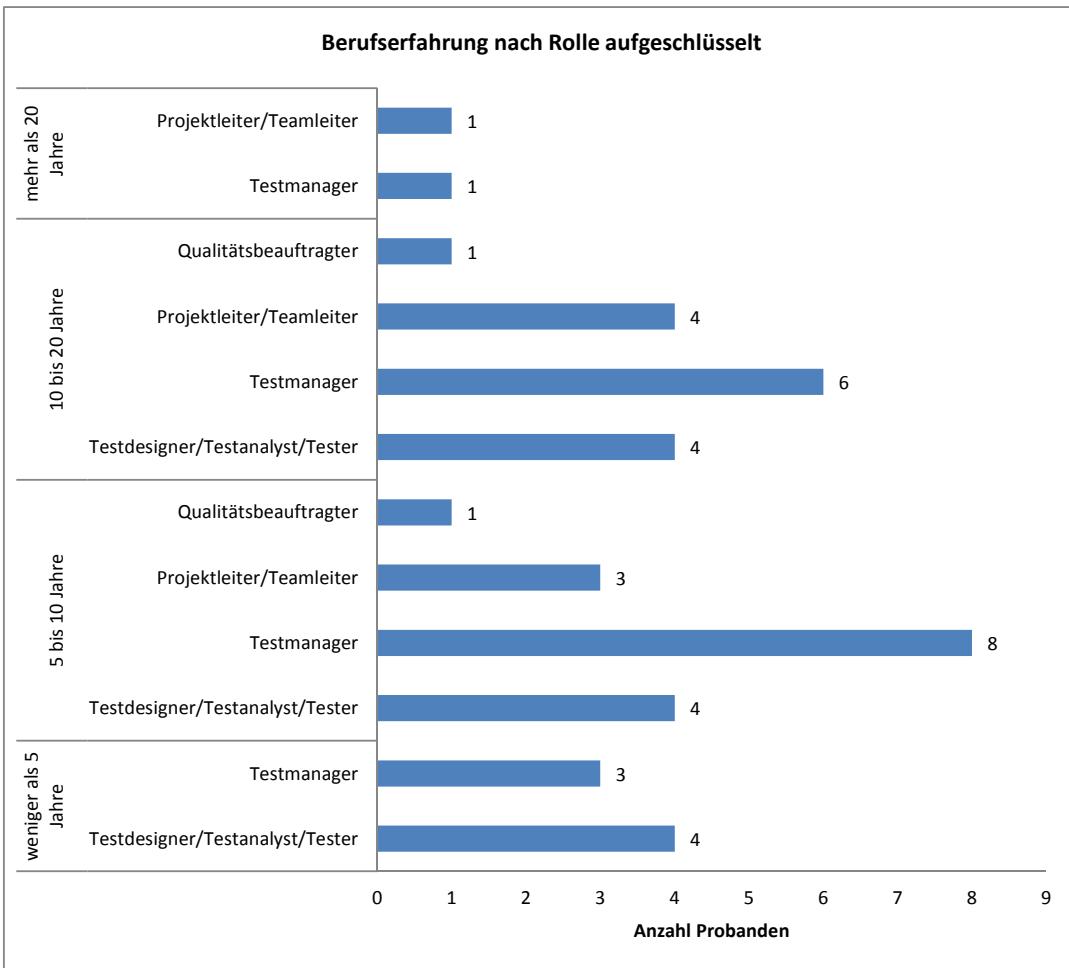


Abbildung A.9.: Expertenumfrage: Welcher Rolle ordnet der Proband seine Hauptaufgaben zu? Wie umfangreich ist die Berufserfahrung des Probanden in der Industrie?

## A. Expertenumfrage

### A.2.2. Auswertung der Fragen zur Erfahrung des Probanden mit Testaufwandsschätzungen

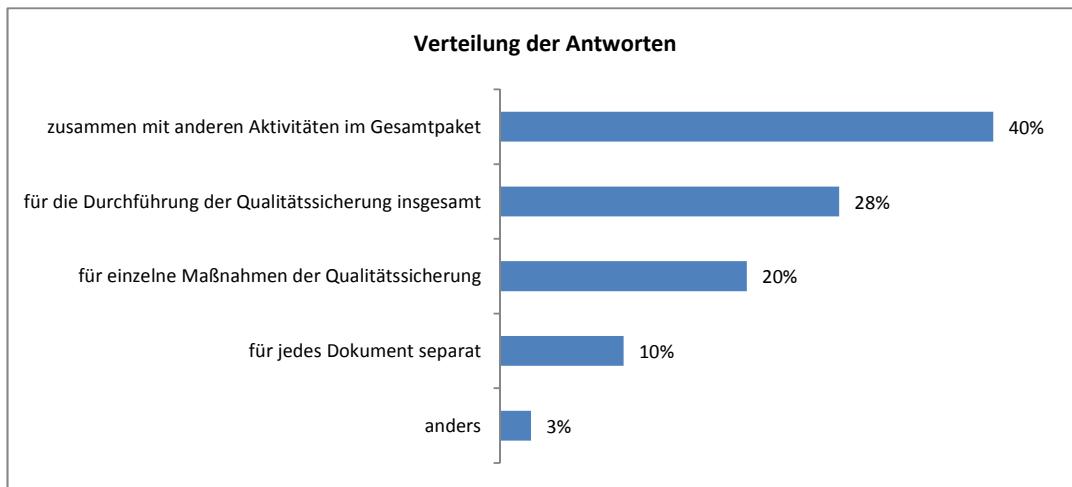


Abbildung A.10.: Expertenumfrage: Wie wird der Aufwand (Budget und Zeit) der Qualitätssicherung in den Projekten des Probanden geplant?

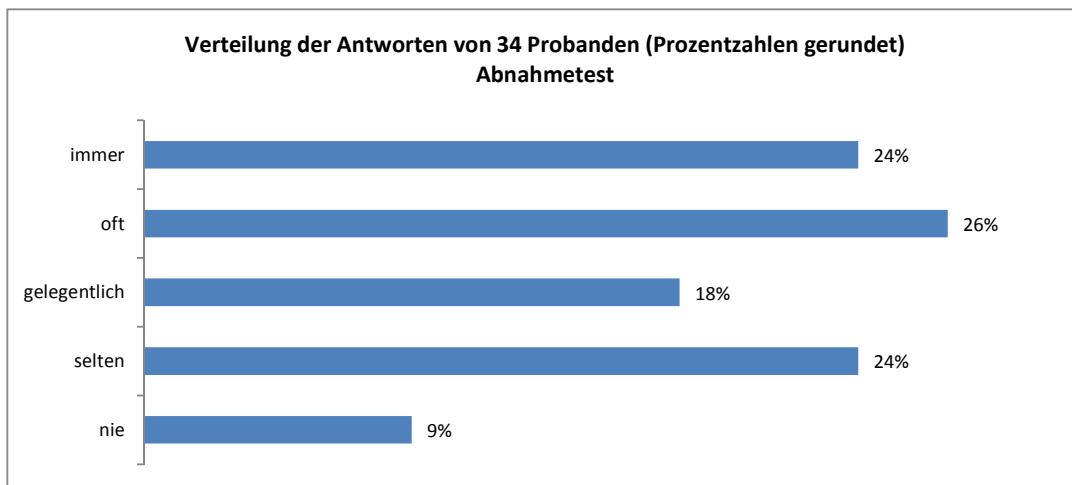


Abbildung A.11.: Expertenumfrage: Wie häufig führt der Proband Aufwands schätzungen für die Teststufe Abnahmetest durch?

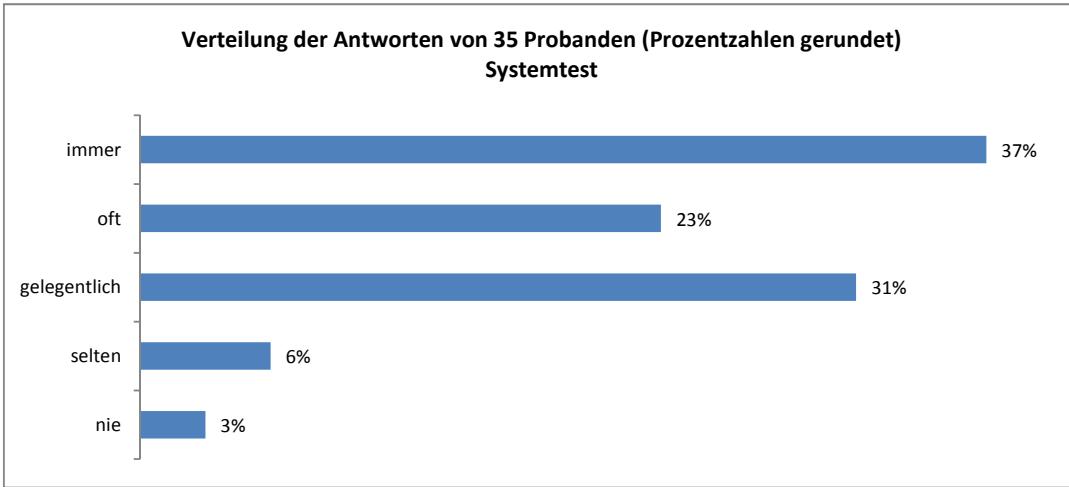


Abbildung A.12.: Expertenumfrage: Wie häufig führt der Proband Aufwands-schätzungen für die Teststufe Systemtest durch?

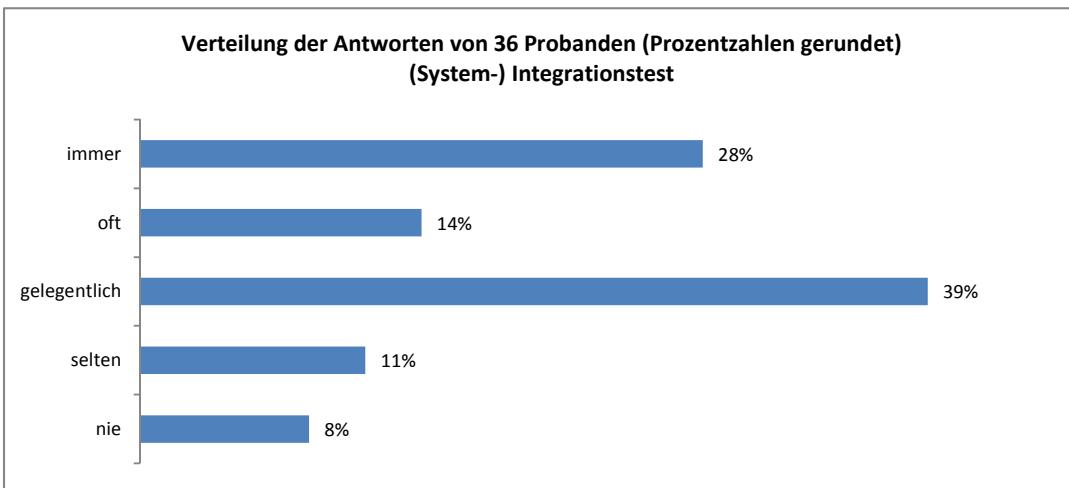


Abbildung A.13.: Expertenumfrage: Wie häufig führt der Proband Aufwands-schätzungen für die Teststufe (System-)Integrationstest durch?

#### A. Expertenumfrage

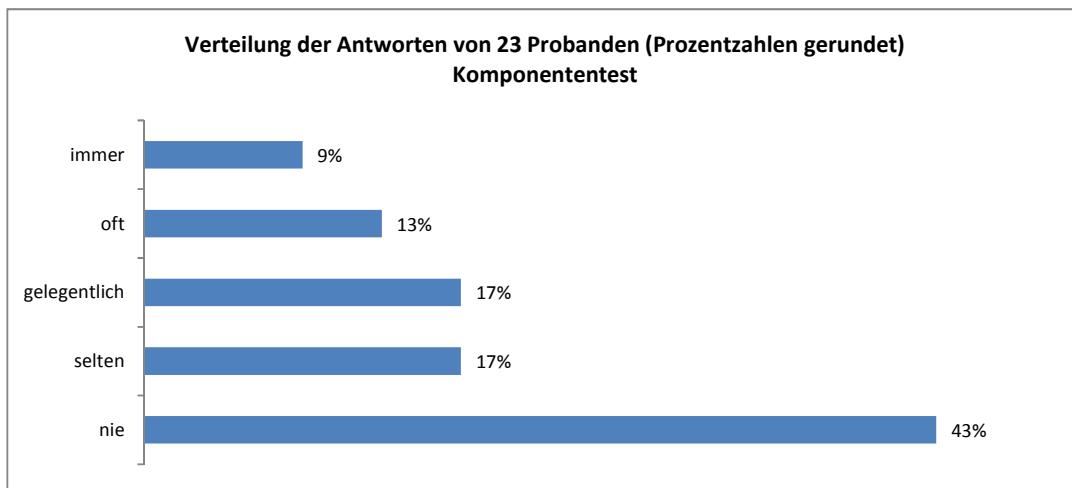


Abbildung A.14.: Expertenumfrage: Wie häufig führt der Proband Aufwands-schätzungen für die Teststufe Komponententest durch?

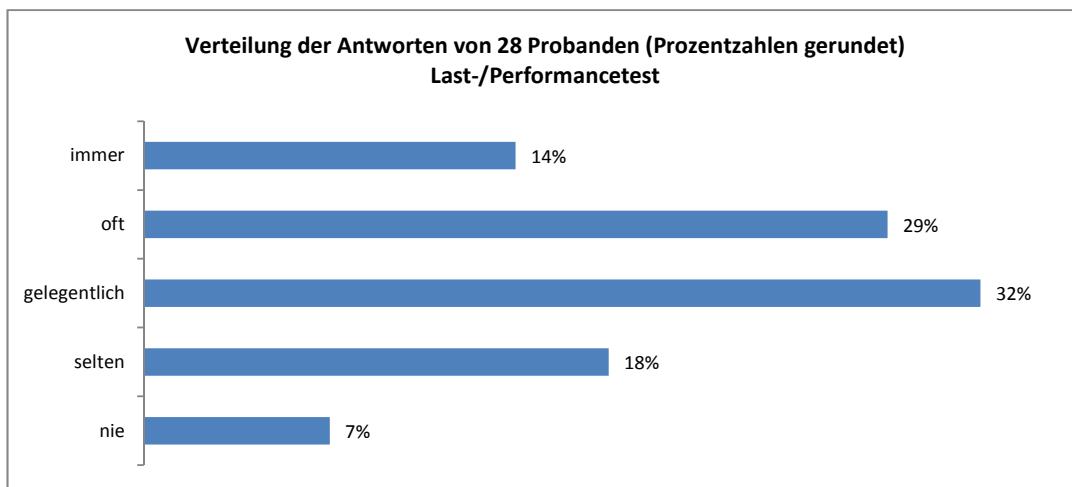


Abbildung A.15.: Expertenumfrage: Wie häufig führt der Proband Aufwands-schätzungen für die Teststufe Last-/Performancetest durch?

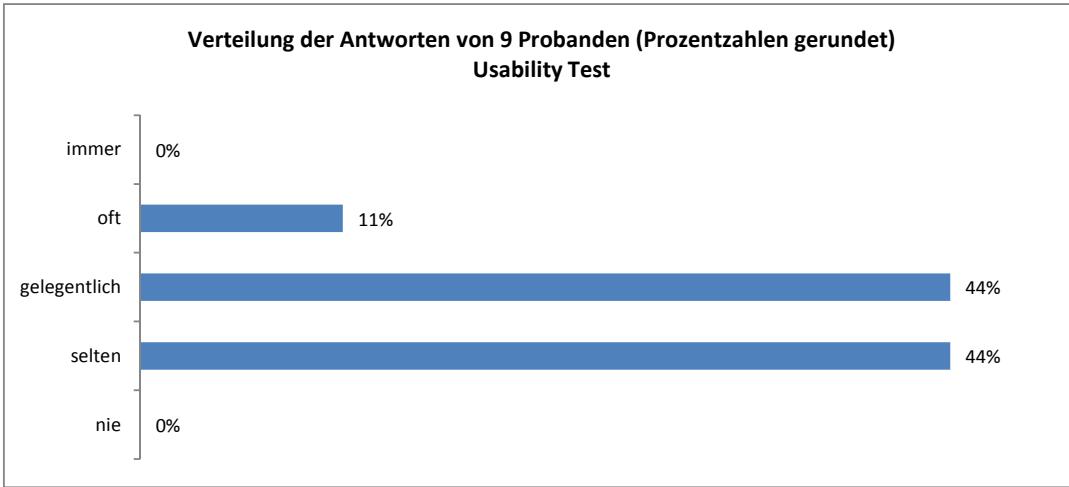


Abbildung A.16.: Expertenumfrage: Wie häufig führt der Proband Aufwands-schätzungen für die Teststufe Usability Test durch?

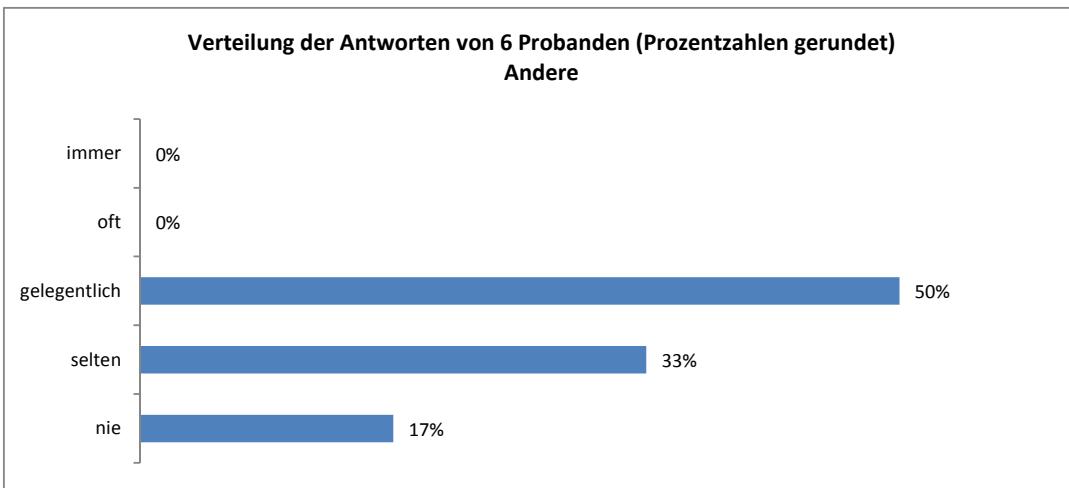


Abbildung A.17.: Expertenumfrage: Wie häufig führt der Proband Aufwands-schätzungen für andere Teststufen durch?

## A. Expertenumfrage

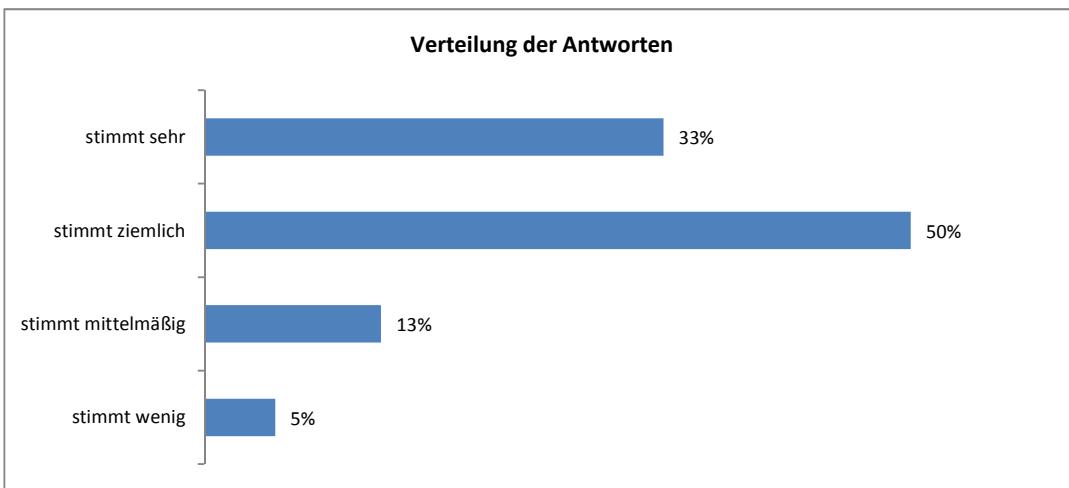


Abbildung A.18.: Expertenumfrage: Ist für eine Testaufwandsschätzung eine Betrachtung des Kontextes z. B. über Testeinflussfaktoren erforderlich?

### A.2.3. Auswertung der Fragen zu den Testeinflussfaktoren

Die Fragegruppen bestanden im Rahmen der Studie immer aus drei Fragen:

- Wie stark beeinflusst der jeweilige Testeinflussfaktor den Testaufwand? Hierzu war für jede qualitative Stufe des Testeinflussfaktors (z.B. *außergewöhnlich gering, sehr hoch* etc.) eine quantitative Bewertung auf einer Skala von -40 % bis +40 % abzugeben.
- Hat der Testeinflussfaktor relevante Auswirkungen auf die Testaktivitäten *Testmanagement, Testdesign, und Testausführung*? Dazu musste für die drei Testaktivitäten separat die Relevanz bewertet werden.
- Wie häufig ist in Ihren Projekten der Testeinflussfaktor während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar? Hierfür konnte für die drei Phasen, nämlich *Anforderungsphase, Designphase, und Entwicklungsphase*, separat eine Bewertung vorgenommen werden.

Zum besseren Nachvollzug der Boxplots seien folgenden Hinweise gegeben:

- Die *Boxen* enthalten 50 % der Antworten. Wenn keine Box oder nur eine Seite der Box für eine qualitative Stufe des Testeinflussfaktors eingezeichnet wurde, fallen die Antworten auf einer Seite oder beiden auf denselben Wert.

- Die *Fühler* an einer Box werden gezeichnet, wenn Werte außerhalb der Box bis zu einer Entfernung der eineinhalbsechsfachen Boxlänge liegen. Die Länge des *Fühlers* ist dabei der maximale Wert innerhalb dieser Spanne.
- Die *Kreise* sind Ausreißer, die größer gleich die eineinhalbsechsfache Boxlänge von der Box entfernt sind.
- Die *Sternchen* sind starke Ausreißer, die größer gleich die dreifache Boxlänge von der Box entfernt sind.
- Die *Schwarze Balken* sind die Mediane.

#### A.2.3.1. Qualität der Testbasis

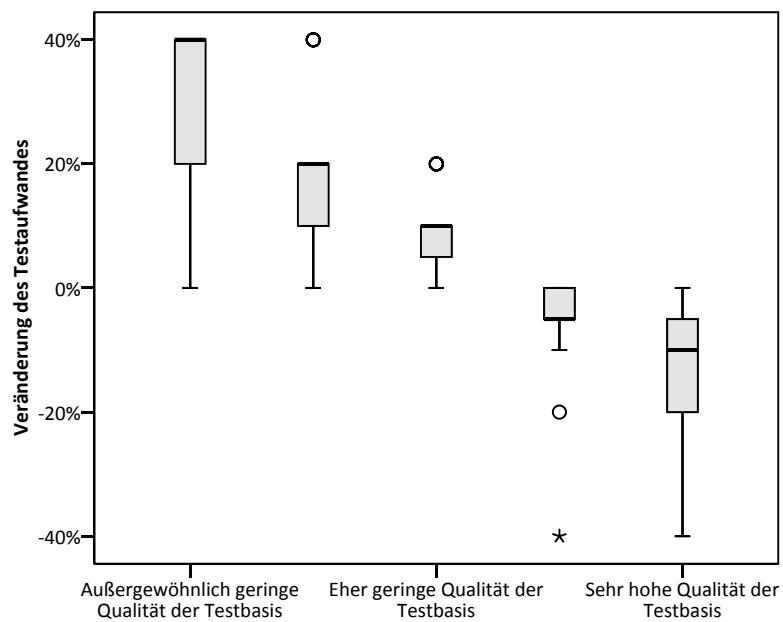


Abbildung A.19.: Expertenumfrage: Boxplot für den Testeinflussfaktor Qualität der Testbasis

## A. Expertenumfrage

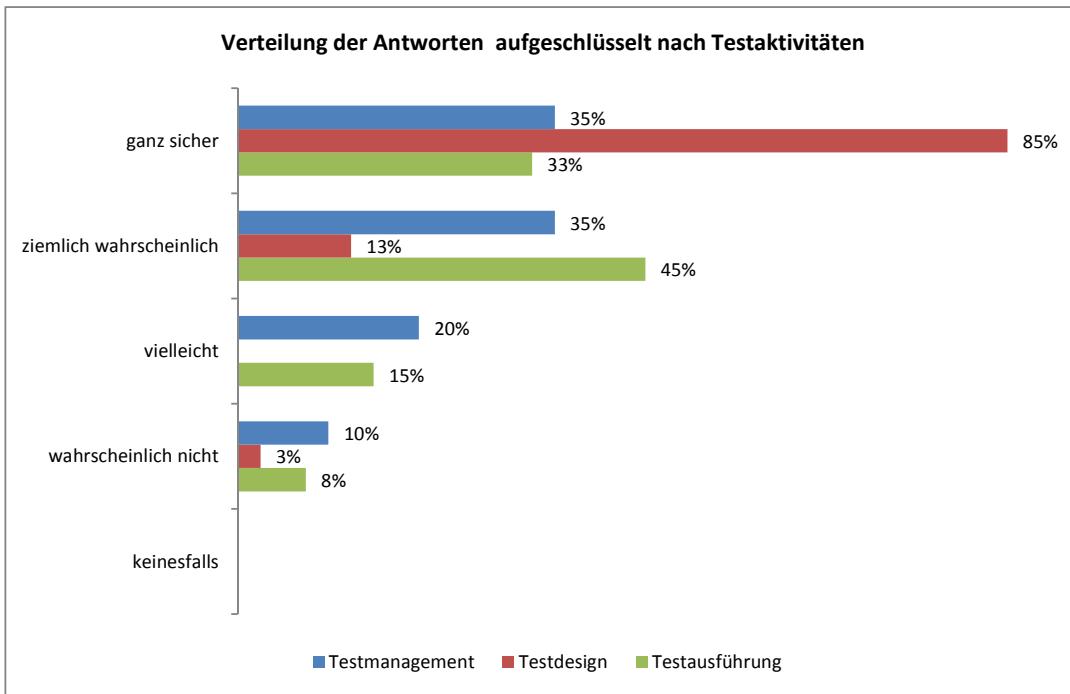


Abbildung A.20.: Expertenumfrage: Hat die Qualität der Testbasis relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

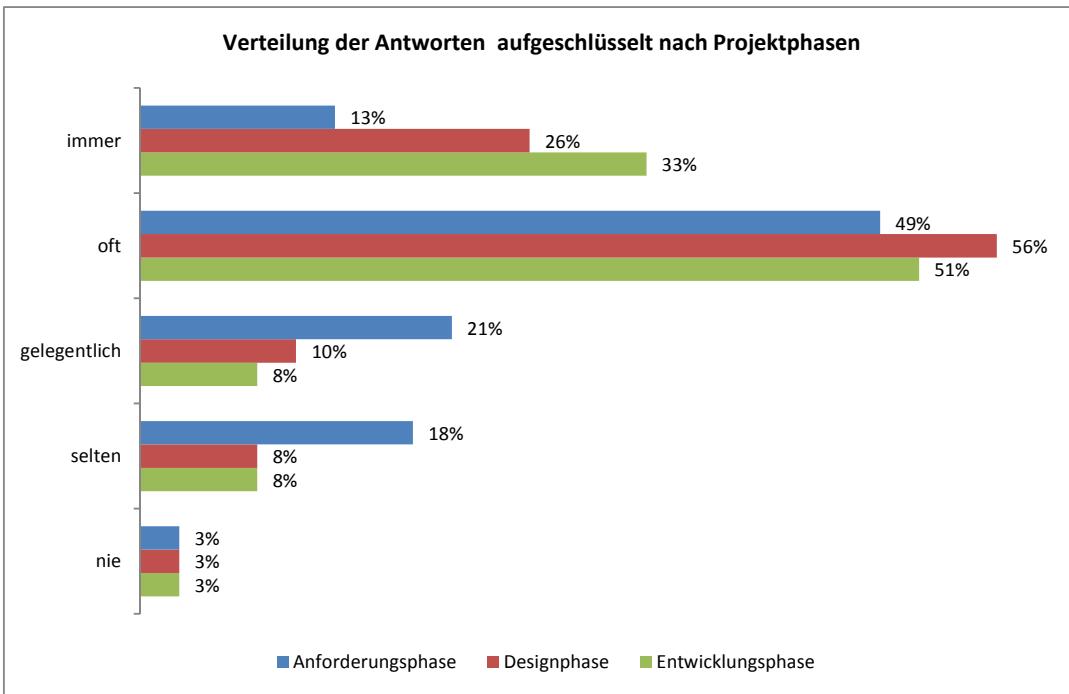


Abbildung A.21.: Expertenumfrage: Wie häufig ist in Ihren Projekten die Qualität der Testbasis während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

### A.2.3.2. Qualitätsziele

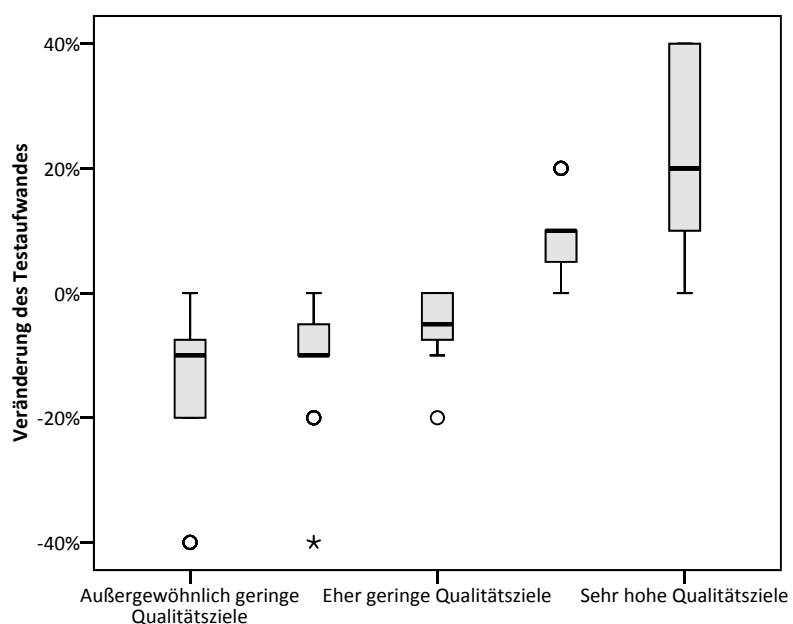


Abbildung A.22.: Boxplot für den Testeinflussfaktor Qualitätsziele

## A. Expertenumfrage

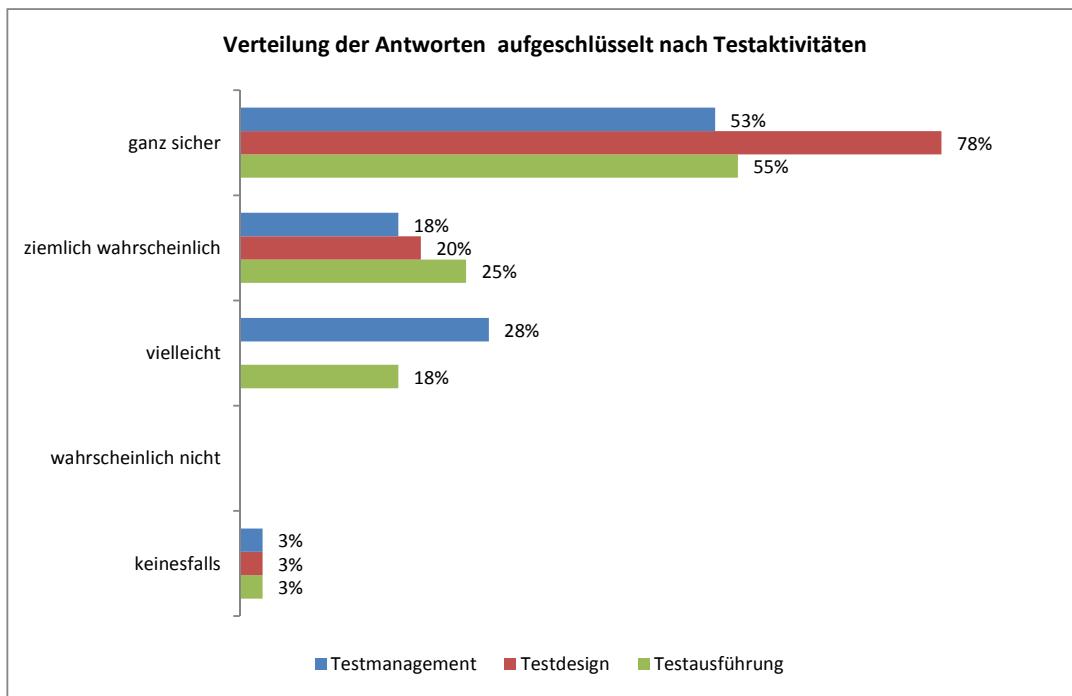


Abbildung A.23.: Expertenumfrage: Haben die Qualitätsziele relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

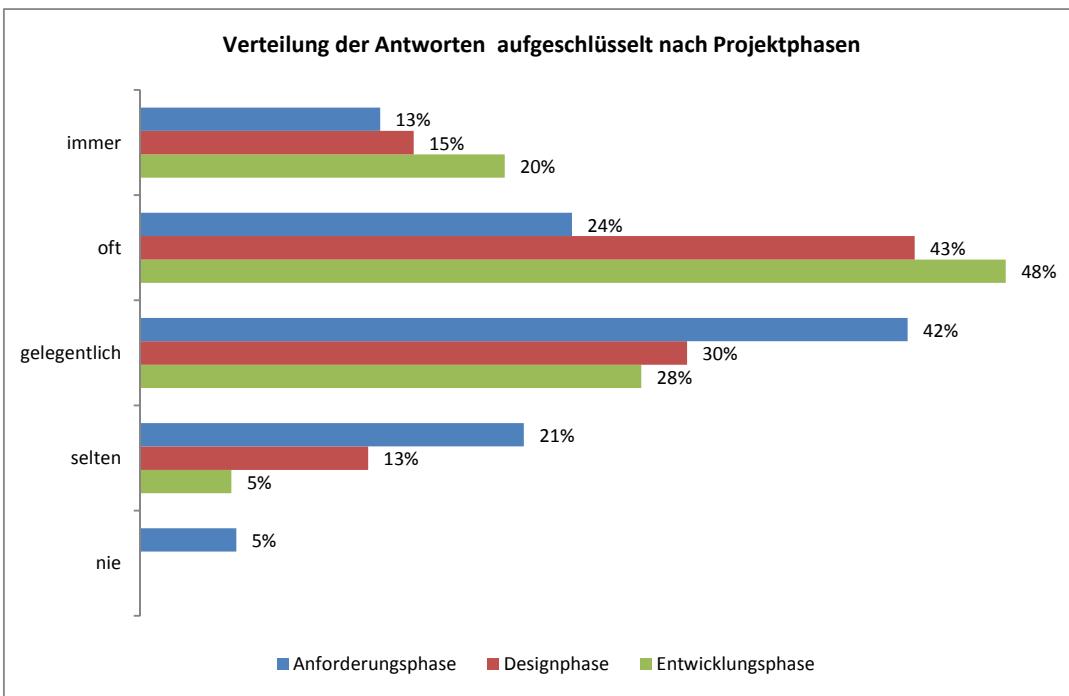


Abbildung A.24.: Expertenumfrage: Wie häufig sind in Ihren Projekten die Qualitätsziele während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

A. Expertenumfrage

**A.2.3.3. Qualifikation und Erfahrung**

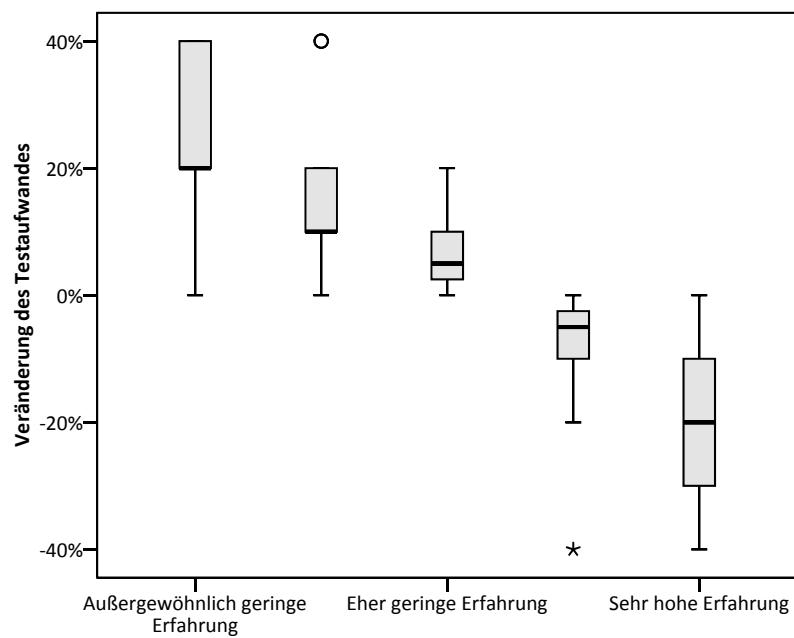


Abbildung A.25.: Boxplot für den Testeinflussfaktor Qualifikation und Erfahrung

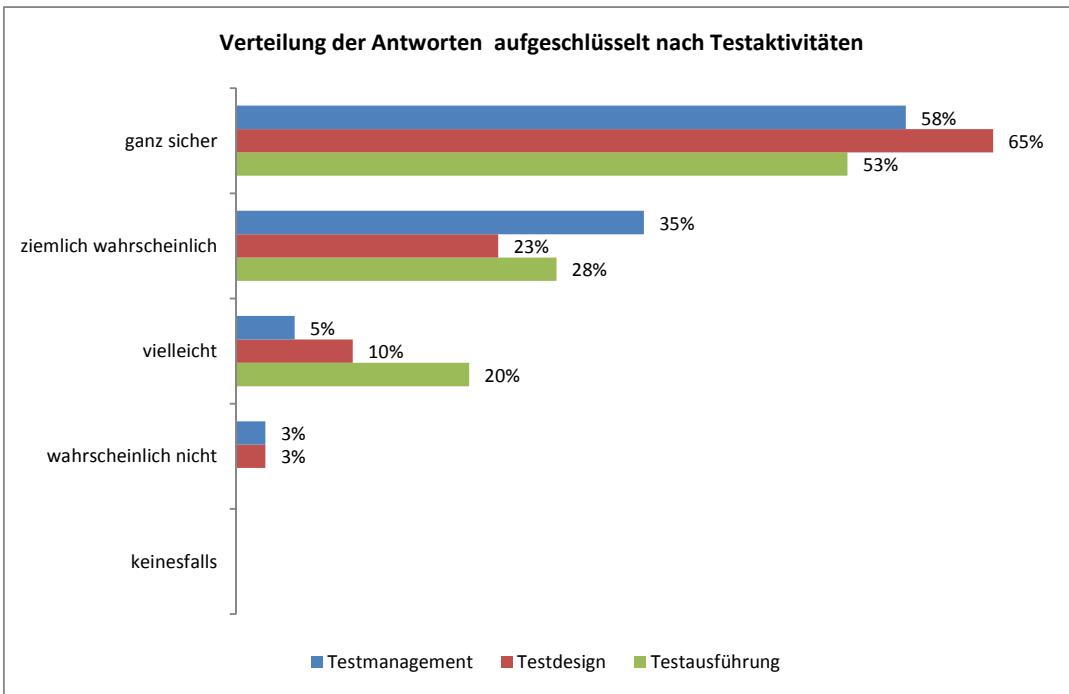


Abbildung A.26.: Expertenumfrage: Hat die Qualifikation und die Erfahrung der Projektmitglieder relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

## A. Expertenumfrage

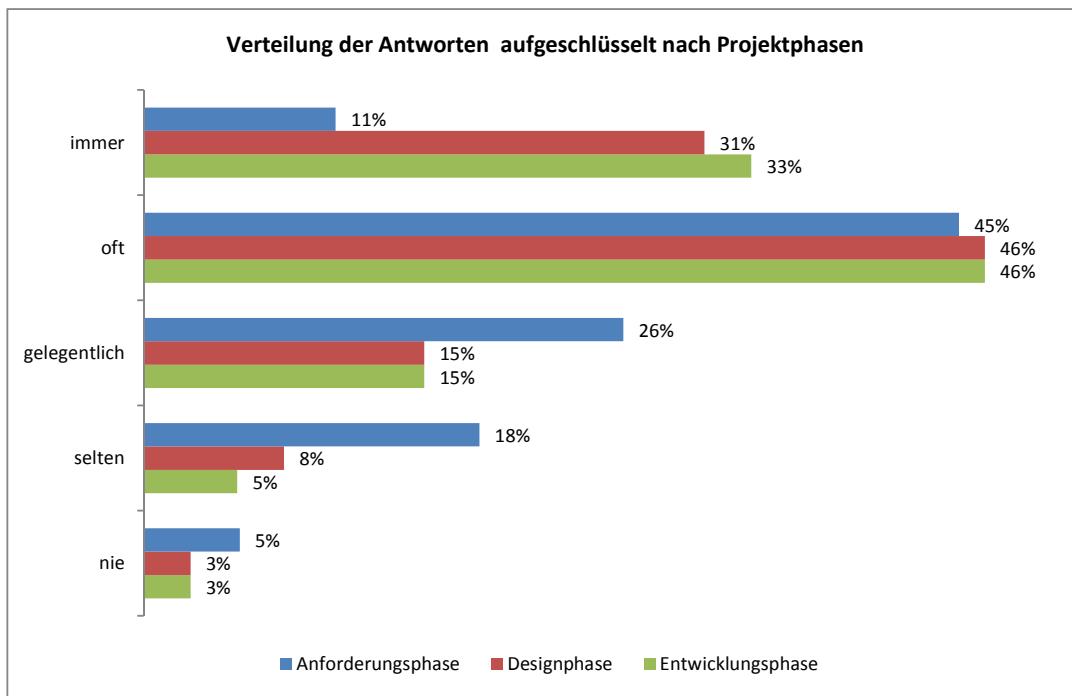


Abbildung A.27.: Expertenumfrage: Wie häufig ist in Ihren Projekten die Qualifikation und die Erfahrung der Projektmitglieder während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

#### A.2.3.4. Kommunikationsaufwand

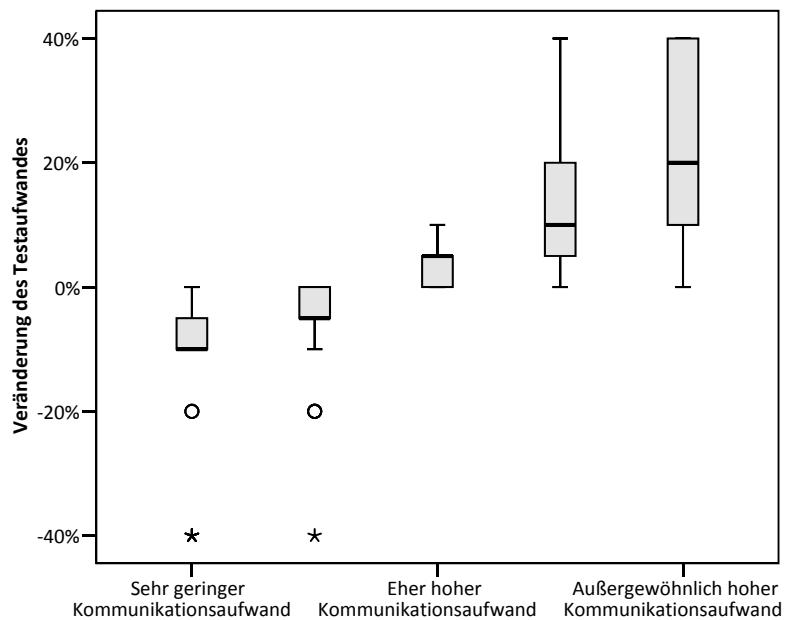


Abbildung A.28.: Boxplot für den Testeinflussfaktor Kommunikationsaufwand

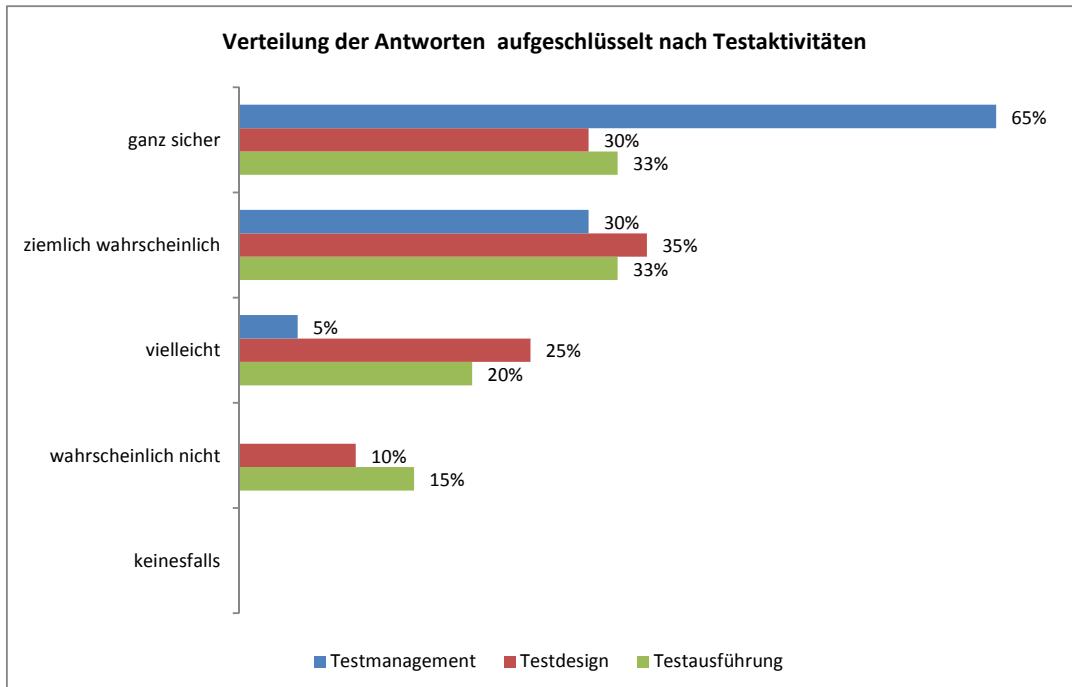


Abbildung A.29.: Expertenumfrage: Hat der Kommunikationsaufwand relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

## A. Expertenumfrage

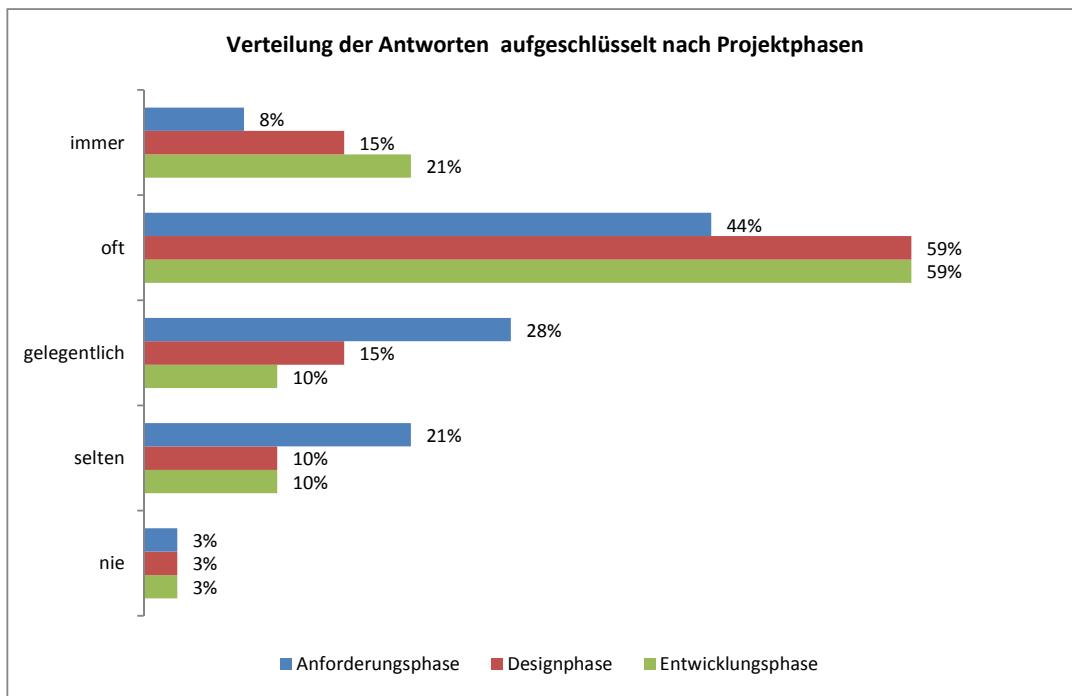


Abbildung A.30.: Expertenumfrage: Wie häufig ist in Ihren Projekten der Kommunikationsaufwand während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

### A.2.3.5. Zusammenarbeit

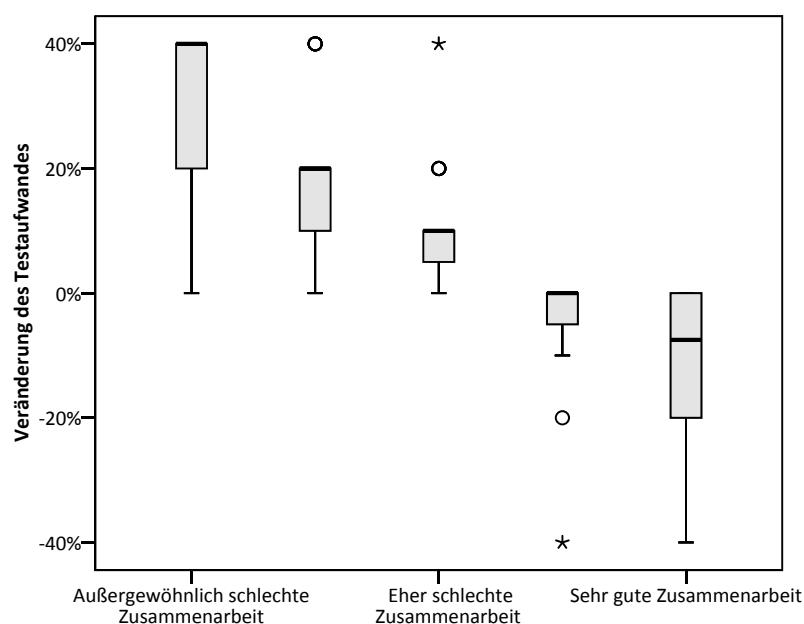


Abbildung A.31.: Boxplot für den Testeinflussfaktor Zusammenarbeit

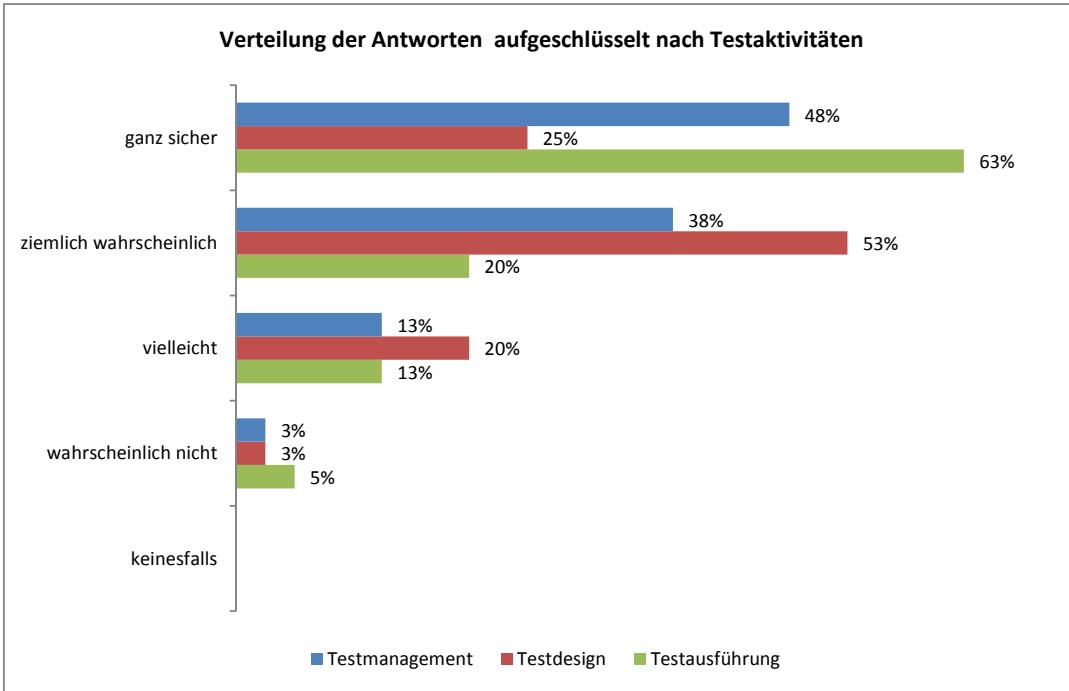


Abbildung A.32.: Expertenumfrage: Hat die Zusammenarbeit der Projektmitglieder relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

## A. Expertenumfrage

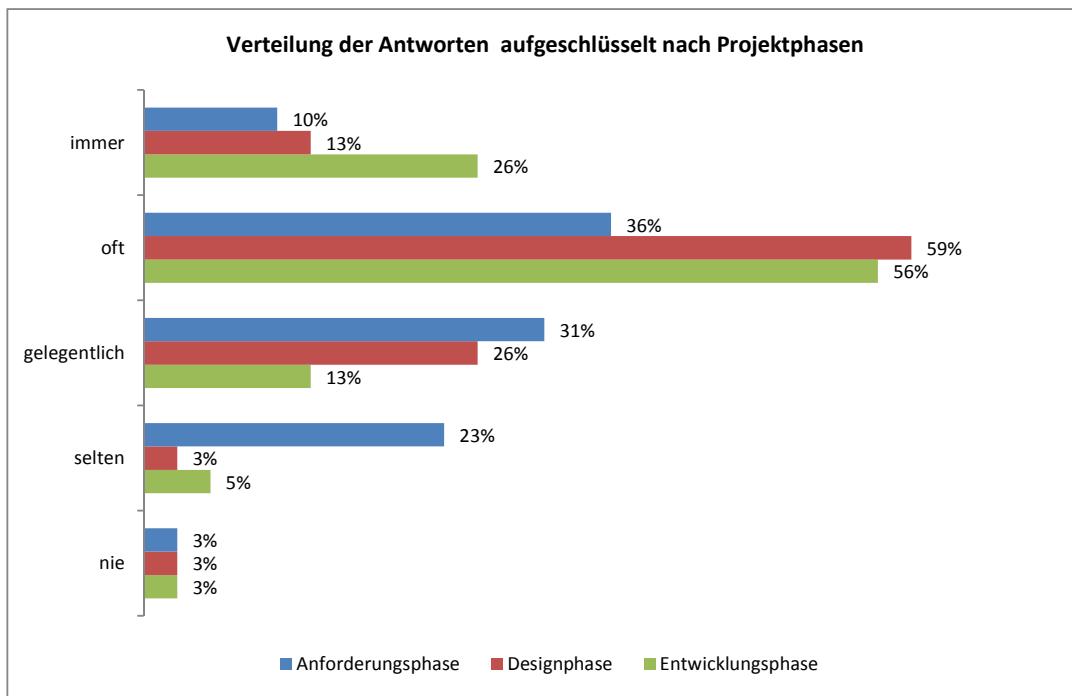


Abbildung A.33.: Expertenumfrage: Wie häufig ist in Ihren Projekten die Zusammenarbeit der Projektmitglieder während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

#### A.2.3.6. Stabilität der Testumgebung

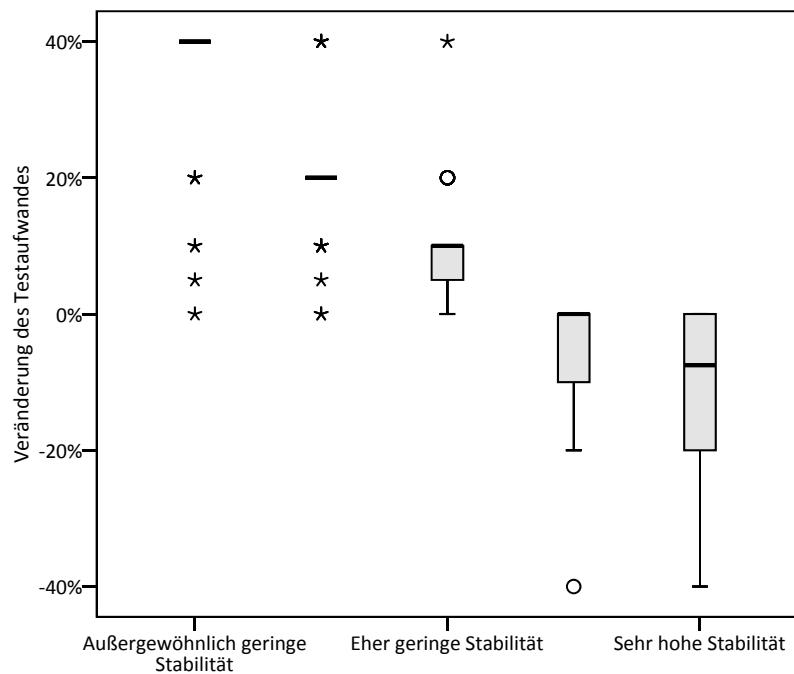


Abbildung A.34.: Boxplot für den Testeinflussfaktor Stabilität der Testumgebung

## A. Expertenumfrage

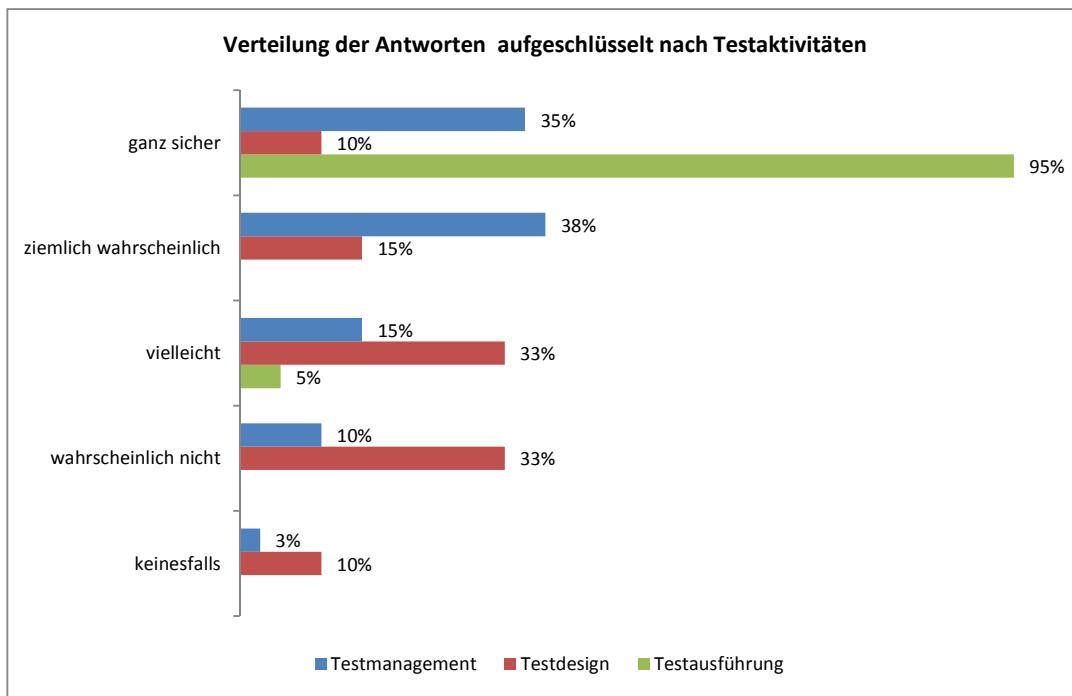


Abbildung A.35.: Expertenumfrage: Hat die Stabilität der Testumgebung relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

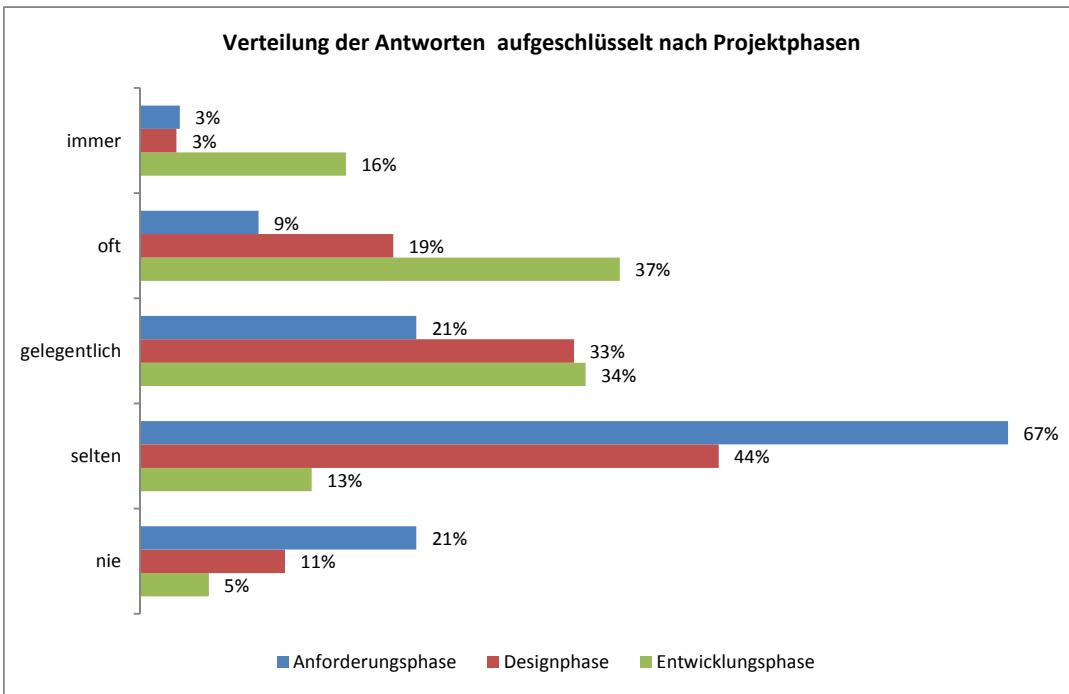


Abbildung A.36.: Expertenumfrage: Wie häufig ist in Ihren Projekten die Stabilität der Testumgebung während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

#### A.2.3.7. Testinfrastruktur

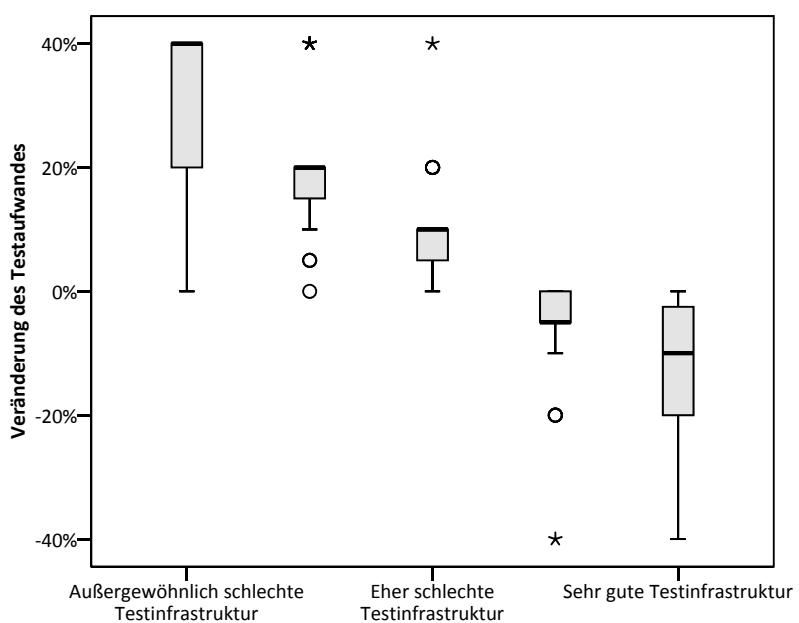


Abbildung A.37.: Boxplot für den Testeinflussfaktor Testinfrastruktur

## A. Expertenumfrage

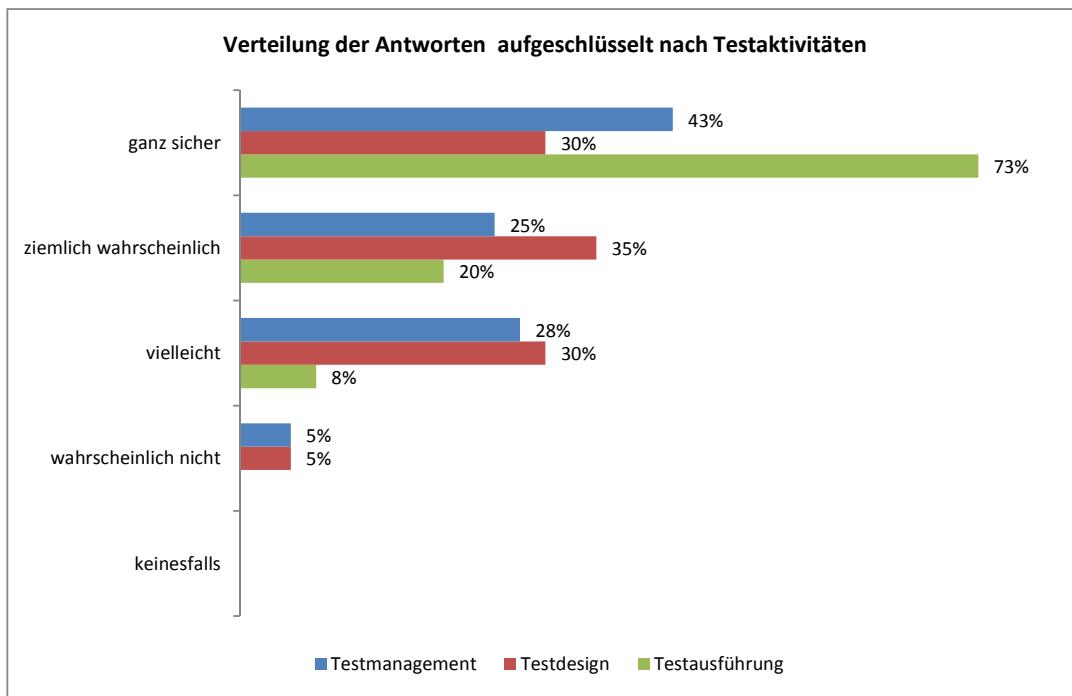


Abbildung A.38.: Expertenumfrage: Hat die Testinfrastruktur relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

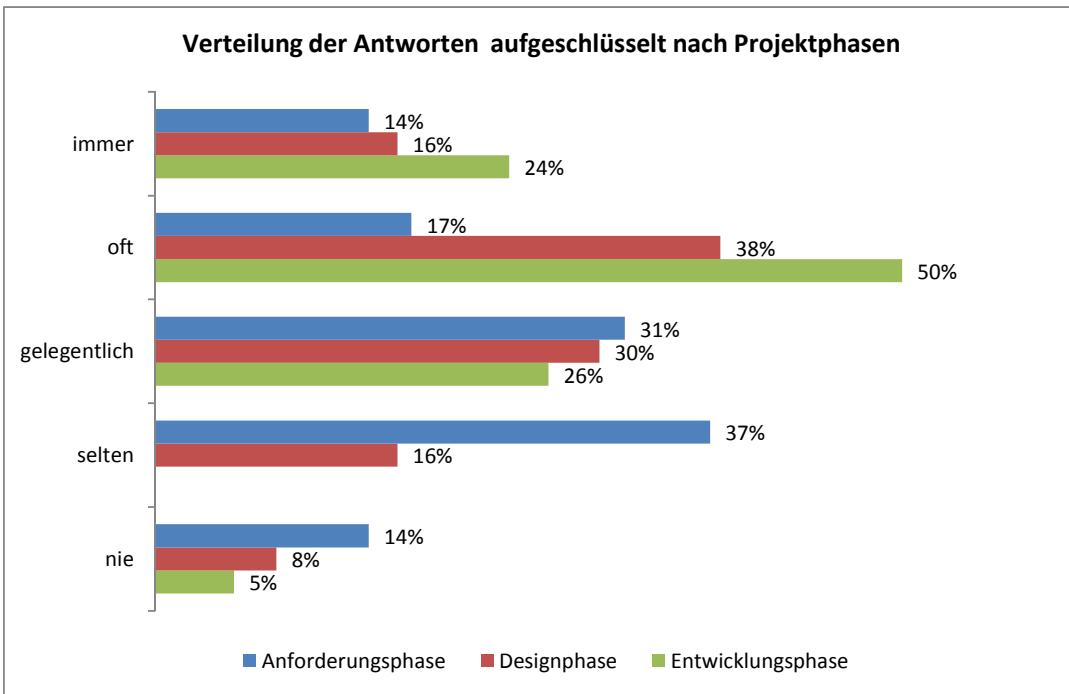


Abbildung A.39.: Expertenumfrage: Wie häufig ist in Ihren Projekten die Testinfrastruktur während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

## A. Expertenumfrage

### A.2.3.8. Gesamttestorganisation

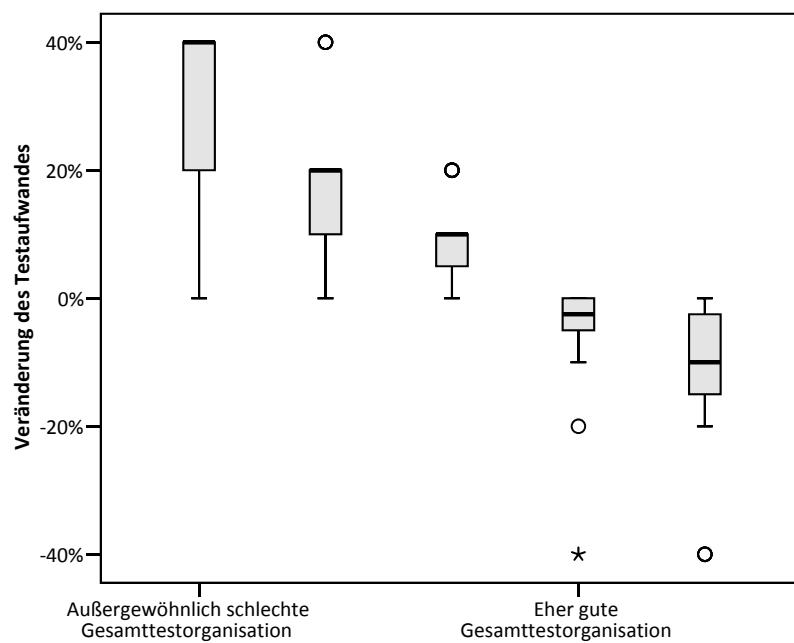


Abbildung A.40.: Boxplot für den Testeinflussfaktor Gesamttestorganisation

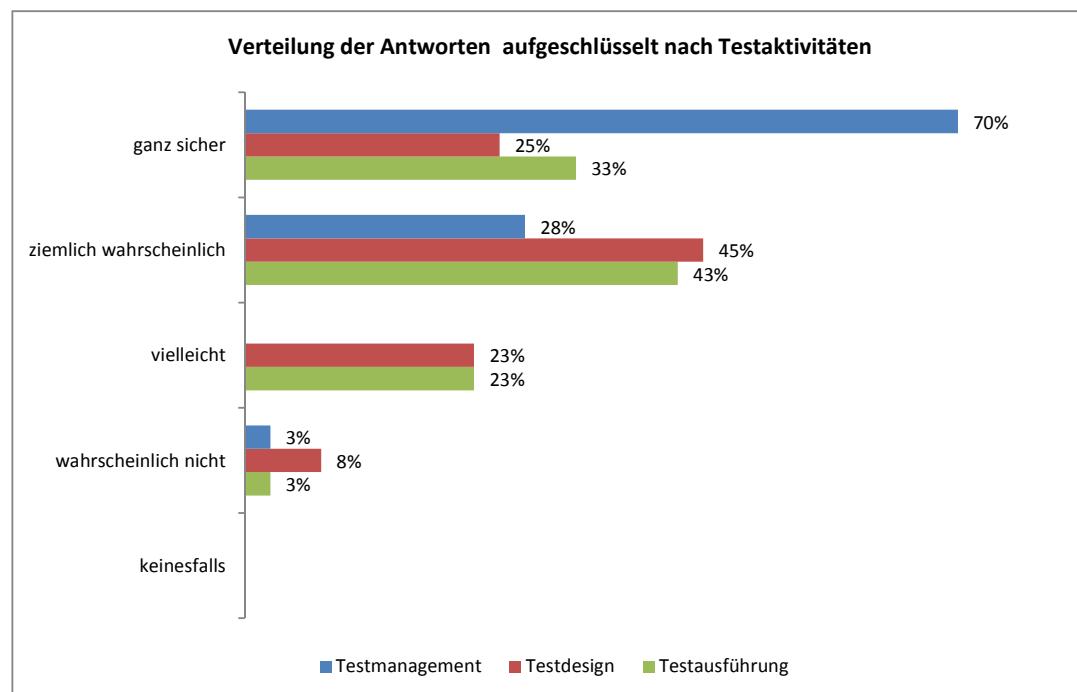


Abbildung A.41.: Expertenumfrage: Hat die Gesamttestorganisation relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

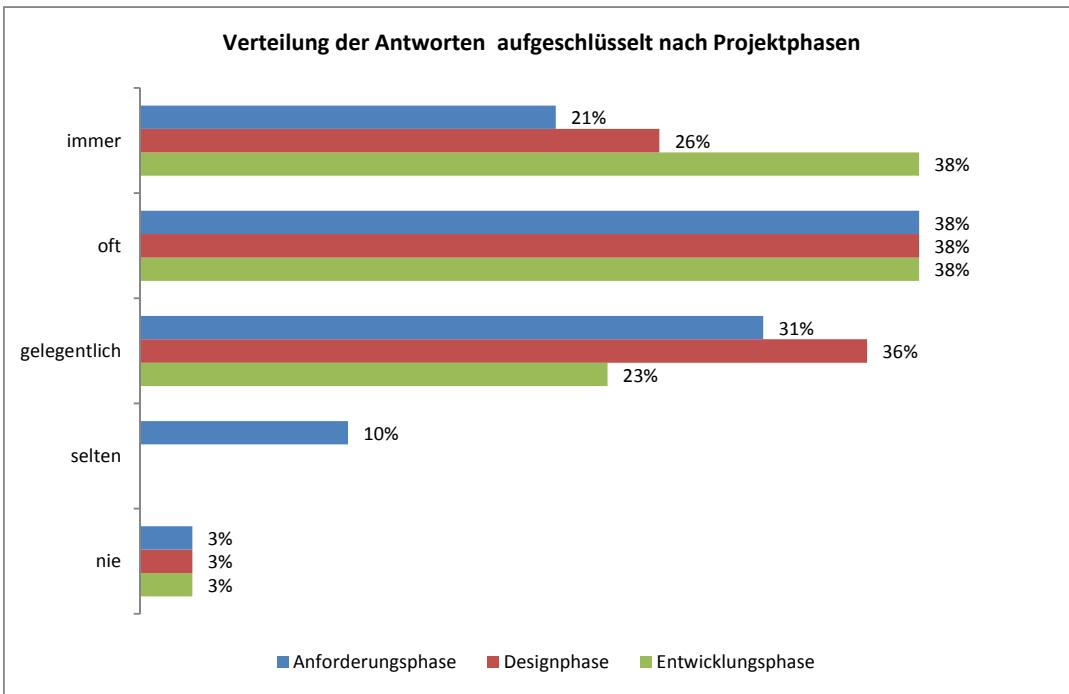


Abbildung A.42.: Expertenumfrage: Wie häufig ist in Ihren Projekten die Testinfrastruktur während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

## A. Expertenumfrage

### A.2.3.9. Dokumentationsbedarf

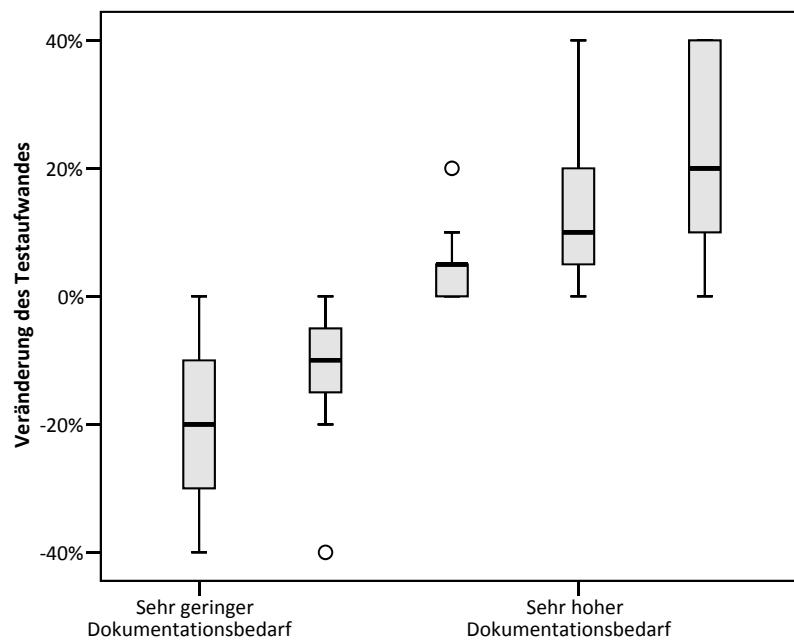


Abbildung A.43.: Boxplot für den Testeinflussfaktor Dokumentationsbedarf

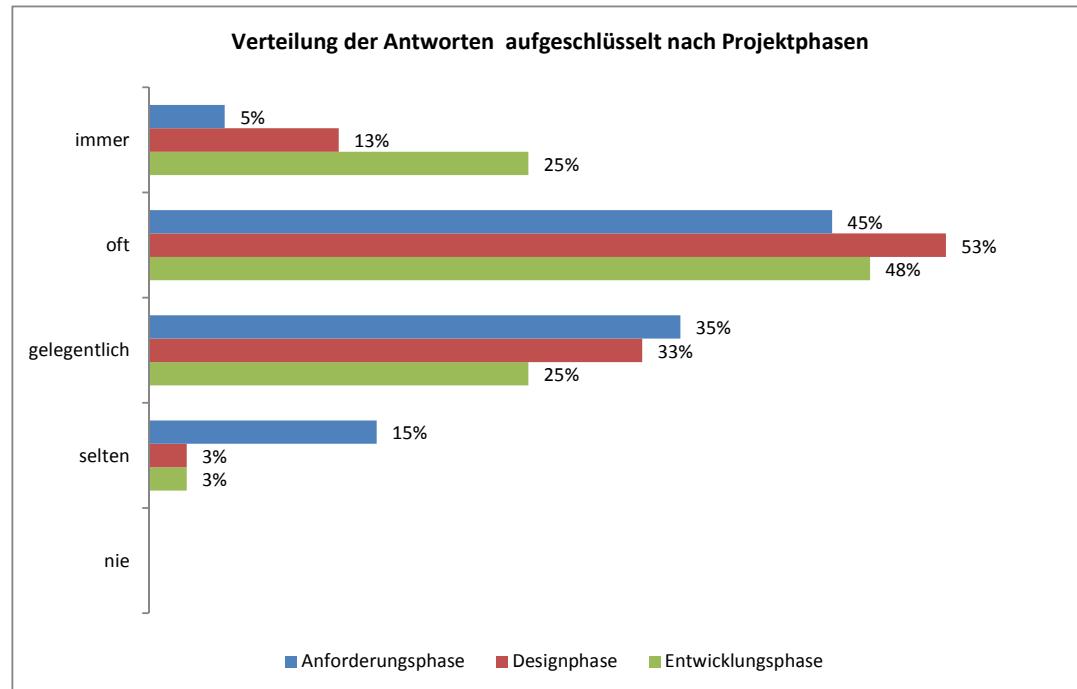


Abbildung A.44.: Expertenumfrage: Wie häufig ist in Ihren Projekten der Grad des Dokumentationsbedarfs während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

### A.2.3.10. Zeitplanung

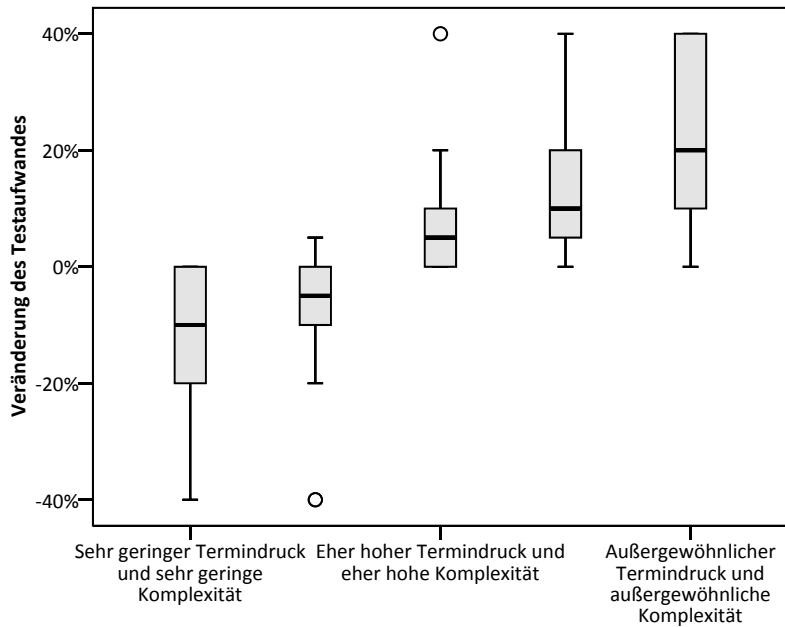


Abbildung A.45.: Boxplot für den Testeinflussfaktor Zeitplanung

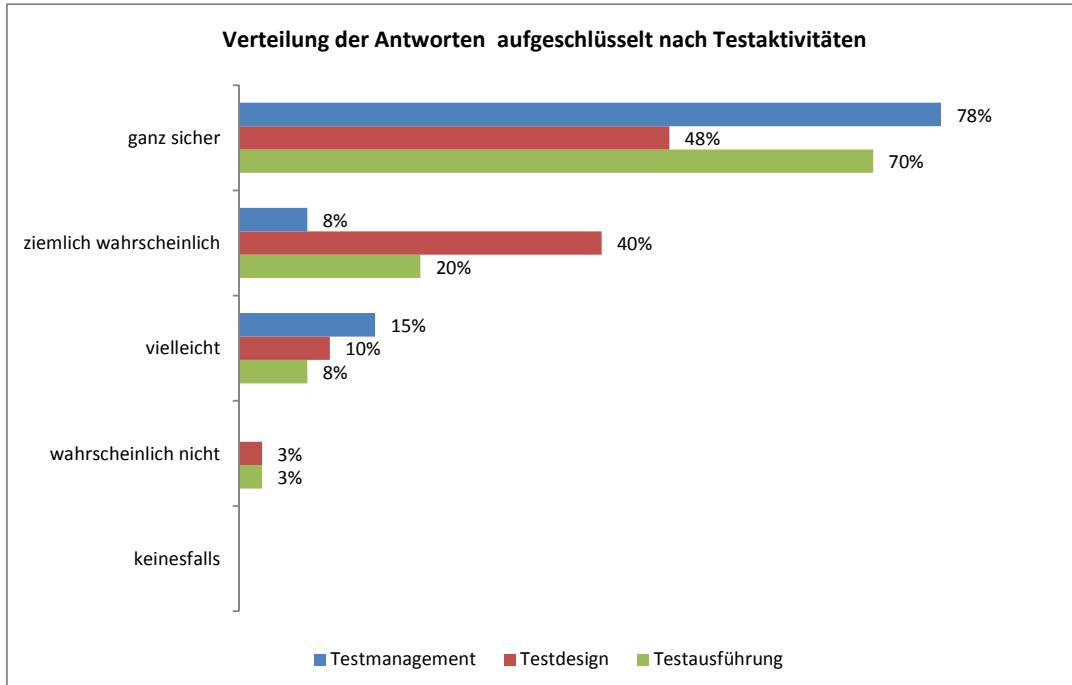


Abbildung A.46.: Expertenumfrage: Hat die Zeitplanung einen relevanten Einfluss auf die Testaktivitäten Testmanagement, Testdesign und Testausführung?

## A. Expertenumfrage

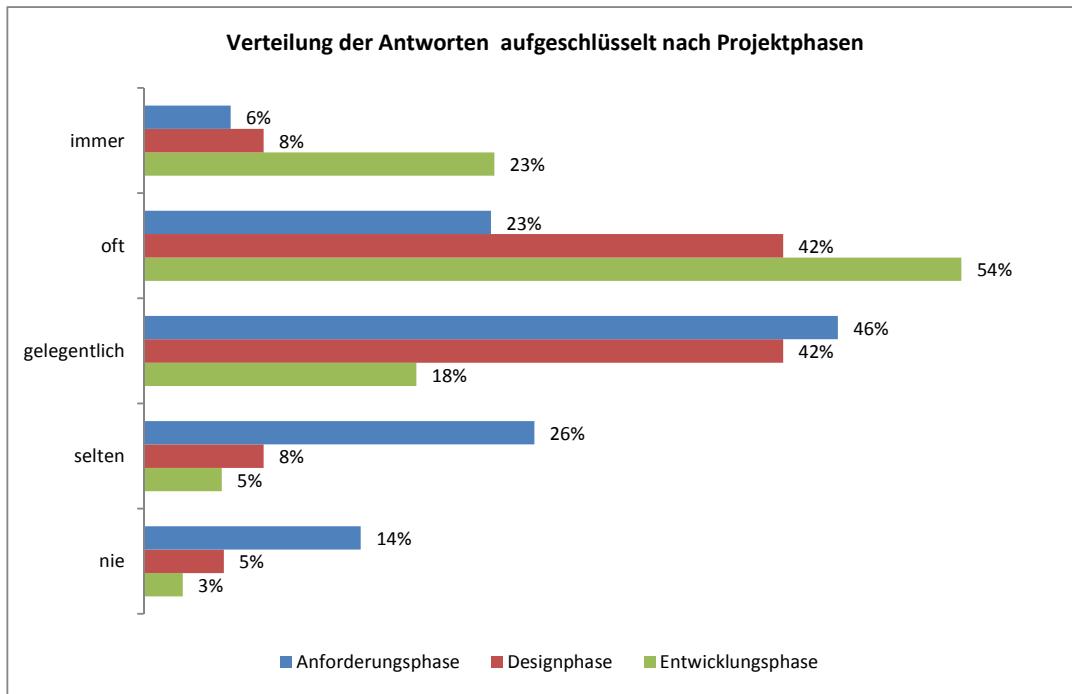


Abbildung A.47.: Expertenumfrage: Wie häufig ist in Ihren Projekten die Zeitplanung während der Anforderungs-, Design- und Entwicklungsphase gut einschätzbar?

## B. Projektdaten

Die Projektdaten basieren auf Testprojekten des Unternehmens CRM IT. Um den Datenschutz zu gewährleisten, werden die Aufwände ohne Bezug auf Personen in aggregierter Form aufgeführt. Weiterhin sind die Namen der Projekte selbst und der zugehörigen Testprojekte ersetzt worden. Dabei ist über die verschiedenen Tabellen hinweg darauf geachtet worden, dass die Namensersetzung konsistent erfolgt ist, d. h. das sich z. B. *Testprojekt 1* immer auf dasselbe Testprojekt bezieht.

## B.1. Testkomplexität und Testaufwände nach Testaktivität getrennt

### B.1.1. (System-)Integrationstest

Projekt	Version = Testprojekt	TKP	Test	Test	Test
			-mgmt.	-design	-ausführung
Projekt 6	Testprojekt 9	202	8,91	7,63	8,13
Projekt 7	Testprojekt 10	52	0,00	3,69	0,00
Projekt 7	Testprojekt 11	52	6,04	1,78	2,63
Projekt 7	Testprojekt 12	346	9,09	2,00	10,53
Projekt 7	Testprojekt 13	318	4,13	1,94	5,85
Projekt 8	Testprojekt 14	532	14,78	5,16	9,69
Projekt 9	Testprojekt 15	309	11,69	4,05	2,81
Projekt 10	Testprojekt 16	106	0,00	1,06	0,00
Projekt 11	Testprojekt 17	74	0,00	1,50	1,25
Projekt 12	Testprojekt 18	156	1,81	3,63	0,00
Projekt 13	Testprojekt 19	42	3,22	6,63	3,56
Projekt 16	Testprojekt 27	82	27,03	39,53	5,06
Projekt 17	Testprojekt 28	282	7,13	49,07	8,96
Projekt 24	Testprojekt 67	252	5,13	1,31	4,09
Projekt 24	Testprojekt 71	32	5,84	6,44	3,28
Projekt 25	Testprojekt 72	132	0,00	13,75	0,00
Projekt 25	Testprojekt 73	132	0,00	6,78	0,00
Projekt 26	Testprojekt 77	82	3,44	7,94	5,98
Projekt 26	Testprojekt 78	82	3,50	8,00	9,00
Projekt 26	Testprojekt 79	62	9,78	1,16	0,00

Tabelle B.1.: Überblick über die Testprojekte für die Stufe - Testkomplexität in TKP und Aufwände in PT

### B.1.2. Systemtest

Projekt = Testprojekt	Version	TKP	Test -mgmt.	Test -design	Test -ausführung
Projekt 1	Testprojekt 1	180	7,56	17,34	0,00
Projekt 1	Testprojekt 2	149	4,25	7,00	4,00
Projekt 1	Testprojekt 3	68	7,28	27,22	7,25
Projekt 2	Testprojekt 4	120	8,94	32,47	23,28
Projekt 3	Testprojekt 5	170	12,38	24,19	14,08
Projekt 3	Testprojekt 6	30	10,75	15,73	19,31
Projekt 4	Testprojekt 7	56	9,54	10,56	0,00
Projekt 5	Testprojekt 8	116	6,96	12,19	0,00
Projekt 7	Testprojekt 10	77	2,75	9,73	16,09
Projekt 7	Testprojekt 11	59	11,89	13,44	12,44
Projekt 13	Testprojekt 19	92	10,53	20,06	0,00
Projekt 14	Testprojekt 20	43	7,59	9,00	0,00
Projekt 15	Testprojekt 21	49	4,93	1,50	1,38
Projekt 15	Testprojekt 22	72	5,56	4,25	8,25
Projekt 15	Testprojekt 23	45	1,50	0,00	0,00
Projekt 16	Testprojekt 24	180	23,41	25,49	6,03
Projekt 16	Testprojekt 25	68	7,03	9,19	7,06
Projekt 16	Testprojekt 26	48	2,94	3,00	2,50
Projekt 17	Testprojekt 28	31	5,47	27,46	17,83
Projekt 17	Testprojekt 29	41	17,28	47,01	28,88
Projekt 17	Testprojekt 30	15	9,44	13,81	6,94
Projekt 18	Testprojekt 31	31	9,63	27,56	28,63
Projekt 18	Testprojekt 32	47	9,84	23,13	22,25
Projekt 18	Testprojekt 33	45	6,25	13,06	16,75
Projekt 18	Testprojekt 34	67	11,38	19,06	18,38
Projekt 18	Testprojekt 35	77	12,20	25,44	29,16
Projekt 18	Testprojekt 36	81	8,75	13,84	30,69
Projekt 18	Testprojekt 37	53	5,78	0,00	21,94
Projekt 18	Testprojekt 38	45	12,50	42,00	39,94
Projekt 18	Testprojekt 39	53	6,00	8,41	11,19
Projekt 18	Testprojekt 40	41	4,78	0,00	3,79
Projekt 18	Testprojekt 41	781	76,79	103,73	32,34
Projekt 18	Testprojekt 42	37	3,94	0,00	0,00

## B. Projektdaten

Projekt 18	Testprojekt 43	58	17,49	29,71	2,76
Projekt 18	Testprojekt 44	182	5,84	15,66	5,25
Projekt 18	Testprojekt 45	53	2,54	7,81	3,44
Projekt 19	Testprojekt 46	101	3,72	7,19	1,38
Projekt 19	Testprojekt 47	203	10,41	9,25	3,13
Projekt 19	Testprojekt 48	122	4,31	9,25	4,38
Projekt 19	Testprojekt 49	70	1,44	2,00	0,00
Projekt 19	Testprojekt 50	31	1,94	3,50	1,88
Projekt 19	Testprojekt 51	92	24,94	28,34	2,25
Projekt 19	Testprojekt 52	31	6,78	6,31	1,38
Projekt 19	Testprojekt 53	31	6,41	6,81	0,00
Projekt 23	Testprojekt 61	13	6,63	5,82	2,19
Projekt 23	Testprojekt 62	63	7,00	20,45	10,38
Projekt 23	Testprojekt 63	31	1,88	2,55	2,48
Projekt 23	Testprojekt 64	31	2,38	1,44	2,63
Projekt 23	Testprojekt 65	27	2,84	2,71	5,13
Projekt 24	Testprojekt 66	23	1,77	4,81	4,44
Projekt 24	Testprojekt 68	64	2,08	0,00	1,03
Projekt 24	Testprojekt 69	45	2,04	5,50	0,00
Projekt 24	Testprojekt 70	34	4,34	4,25	2,52
Projekt 25	Testprojekt 74	73	9,63	7,88	0,00
Projekt 25	Testprojekt 75	21	0,00	7,28	0,00
Projekt 26	Testprojekt 76	45	0,00	22,16	0,00
Projekt 25	Testprojekt 72	27	0,00	33,84	0,00
Projekt 25	Testprojekt 73	19	0,00	19,48	2,50
Projekt 26	Testprojekt 77	65	16,28	10,44	13,31
Projekt 26	Testprojekt 78	87	12,75	4,00	21,05
Projekt 26	Testprojekt 80	54	3,63	0,00	1,75
Projekt 26	Testprojekt 81	52	7,25	1,56	10,62
Projekt 26	Testprojekt 82	34	0,00	0,00	3,44
Projekt 26	Testprojekt 83	118	9,23	14,44	14,16
Projekt 27	Testprojekt 84	36	2,99	1,81	4,98
Projekt 27	Testprojekt 85	339	46,76	76,54	20,83
Projekt 27	Testprojekt 86	62	7,17	4,19	0,00
Projekt 27	Testprojekt 87	144	9,21	8,00	4,78
Projekt 27	Testprojekt 88	78	6,00	4,00	5,19
Projekt 28	Testprojekt 89	34	5,69	5,38	5,38
Projekt 28	Testprojekt 90	122	5,38	6,36	15,41

B.1. Testkomplexität und Testaufwände nach Testaktivität getrennt

Projekt 28	Testprojekt 91	82	6,08	16,23	8,11
Projekt 28	Testprojekt 92	128	15,75	54,32	3,06
Projekt 28	Testprojekt 93	64	5,25	3,94	2,84
Projekt 28	Testprojekt 94	118	12,05	6,14	7,60
Projekt 28	Testprojekt 95	166	6,64	22,95	22,00
Projekt 28	Testprojekt 96	122	7,51	12,92	18,06
Projekt 29	Testprojekt 98	127	5,89	8,75	11,31
Projekt 29	Testprojekt 99	49	3,69	1,88	3,50
Projekt 29	Testprojekt 100	163	6,30	7,75	10,65
Projekt 29	Testprojekt 101	59	2,81	2,23	5,81
Projekt 29	Testprojekt 102	79	5,83	9,31	4,25
Projekt 29	Testprojekt 103	33	3,19	2,38	2,03
Projekt 29	Testprojekt 104	31	1,19	2,51	1,59
Projekt 29	Testprojekt 105	96	9,92	8,54	17,20
Projekt 29	Testprojekt 106	130	10,38	11,87	16,75
Projekt 29	Testprojekt 107	100	15,58	19,25	13,51
Projekt 29	Testprojekt 108	152	12,06	23,37	14,75
Projekt 29	Testprojekt 109	132	12,47	18,19	29,00
Projekt 29	Testprojekt 110	136	14,94	16,98	17,25
Projekt 29	Testprojekt 111	92	21,72	25,96	35,48
Projekt 29	Testprojekt 112	86	14,43	20,46	39,19
Projekt 29	Testprojekt 113	32	6,39	2,88	7,80
Projekt 29	Testprojekt 114	180	19,30	19,15	26,16
Projekt 29	Testprojekt 115	93	6,69	10,56	17,87

Tabelle B.2.: Überblick über die Testprojekte für die Stufe Systemtest - Testkomplexität in TKP und Aufwände in PT

### B.1.3. Abnahmetest

Projekt	Version = Testprojekt	TKP	Test	Test	Test
			-mgmt.	-design	-ausführung
Projekt 13	Testprojekt 19	92	0,00	2,00	0,00
Projekt 19	Testprojekt 54	22	12,13	1,94	0,00
Projekt 19	Testprojekt 55	46	39,50	23,66	0,00
Projekt 20	Testprojekt 56	62	15,44	15,22	0,00
Projekt 21	Testprojekt 57	56	19,17	6,59	0,00
Projekt 21	Testprojekt 58	40	11,84	6,47	0,00
Projekt 22	Testprojekt 59	32	7,27	14,97	0,00
Projekt 23	Testprojekt 60	110	54,40	22,03	0,00
Projekt 28	Testprojekt 97	120	52,39	48,59	40,78

Tabelle B.3.: Überblick über die Testprojekte für die Stufe Abnahmetest - Testkomplexität in TKP und Aufwände in PT

## B.2. Bewertung der Testeinflussfaktoren

Projekt	Version = Testprojekt	<i>TF<sub>1</sub></i>	<i>TF<sub>2</sub></i>	<i>TF<sub>3</sub></i>	<i>TF<sub>4</sub></i>	<i>TF<sub>5</sub></i>	<i>TF<sub>6</sub></i>	<i>TF<sub>7</sub></i>	<i>TF<sub>8</sub></i>	<i>TF<sub>9</sub></i>	<i>TF<sub>10</sub></i>	<i>TF<sub>11</sub></i>
Projekt 1	Testprojekt 1	5	3	3	2	2	1	1	4	1	4	2
Projekt 1	Testprojekt 2	4	5	4	4	2	1	3	5	3	3	3
Projekt 1	Testprojekt 3	4	5	4	3	3	3	4	5	1	4	3
Projekt 2	Testprojekt 4	4	5	3	2	2	3	1	2	2	4	3
Projekt 3	Testprojekt 5	3	4	3	1	2	3	1	4	2	3	3
Projekt 3	Testprojekt 6	3	3	3	2	3	3	1	4	2	3	3
Projekt 4	Testprojekt 7	5	4	3	1	3	4	1	4	2	3	3
Projekt 5	Testprojekt 8	4	4	4	2	3	3	1	4	2	3	3
Projekt 6	Testprojekt 9	4	4	4	4	3	4	1	5	2	4	2
Projekt 7	Testprojekt 10	3	4	3	2	3	3	5	4	2	3	3
Projekt 7	Testprojekt 10	3	4	3	2	3	3	5	4	2	3	3
Projekt 7	Testprojekt 11	3	4	4	2	3	3	3	4	2	3	3
Projekt 7	Testprojekt 11	3	4	4	2	3	3	3	4	2	3	3
Projekt 7	Testprojekt 12	4	5	4	2	3	3	3	4	2	3	3
Projekt 7	Testprojekt 13	4	5	3	2	3	3	3	4	2	3	3
Projekt 8	Testprojekt 14	4	5	3	3	3	3	4	4	2	3	3
Projekt 9	Testprojekt 15	4	5	3	3	3	3	4	4	2	3	3
Projekt 10	Testprojekt 16	4	5	3	3	3	3	4	4	2	3	3
Projekt 11	Testprojekt 17	4	5	3	3	3	3	4	4	2	3	3
Projekt 12	Testprojekt 18	4	5	3	3	3	3	4	4	2	3	3
Projekt 13	Testprojekt 19	4	3	3	1	2	4	1	4	2	5	3

Projekt 13	Testprojekt 19	4	3	3	1	2	4	1	4	2	5	3
Projekt 14	Testprojekt 20	3	3	3	3	3	3	3	4	2	4	3
Projekt 15	Testprojekt 21	5	2	5	4	3	3	3	4	2	3	4
Projekt 15	Testprojekt 22	5	3	5	4	3	3	4	4	2	4	3
Projekt 15	Testprojekt 23	3	3	5	3	3	3	4	2	2	4	3
Projekt 16	Testprojekt 24	3	4	3	1	3	3	2	2	2	4	3
Projekt 16	Testprojekt 25	5	4	4	4	3	4	1	5	2	3	3
Projekt 16	Testprojekt 26	5	5	5	4	3	4	2	5	2	3	3
Projekt 16	Testprojekt 27	4	5	4	1	2	2	1	5	2	5	3
Projekt 17	Testprojekt 28	2	5	3	1	3	2	2	5	2	5	3
Projekt 17	Testprojekt 28	2	5	3	1	3	2	2	5	2	5	3
Projekt 17	Testprojekt 29	4	5	3	2	3	2	3	5	2	5	3
Projekt 17	Testprojekt 30	3	5	3	2	3	2	3	5	2	5	3
Projekt 18	Testprojekt 31	4	5	4	1	3	2	4	5	2	5	3
Projekt 18	Testprojekt 32	5	5	4	2	3	2	4	5	2	5	3
Projekt 18	Testprojekt 33	4	5	4	1	3	2	4	5	2	5	3
Projekt 18	Testprojekt 34	4	5	4	1	2	2	1	5	2	5	3
Projekt 18	Testprojekt 35	4	5	5	1	2	2	4	5	2	5	3
Projekt 18	Testprojekt 36	4	5	5	1	2	2	4	5	2	5	3
Projekt 18	Testprojekt 37	4	5	5	1	2	2	4	5	2	5	3
Projekt 18	Testprojekt 38	4	5	4	1	2	2	5	5	2	5	3
Projekt 18	Testprojekt 39	4	5	5	1	2	2	4	5	2	5	3
Projekt 18	Testprojekt 40	4	5	5	1	2	2	4	5	2	5	3
Projekt 18	Testprojekt 41	5	4	3	2	4	2	1	4	3	3	4

## B.2. Bewertung der Testeinflussfaktoren

Projekt 18	Testprojekt 42	5	4	4	2	4	3	3	3	3	4	3
Projekt 18	Testprojekt 43	4	4	3	2	3	3	4	3	3	4	3
Projekt 18	Testprojekt 44	5	4	4	3	3	3	4	3	3	4	3
Projekt 18	Testprojekt 45	4	5	4	2	3	3	1	3	3	4	3
Projekt 19	Testprojekt 46	5	4	4	3	3	3	4	3	3	4	3
Projekt 19	Testprojekt 47	5	4	4	3	3	3	1	3	3	4	3
Projekt 19	Testprojekt 48	5	4	4	3	3	3	4	3	3	4	3
Projekt 19	Testprojekt 49	5	4	4	3	3	3	4	3	3	4	3
Projekt 19	Testprojekt 50	5	4	4	4	3	3	4	3	3	4	3
Projekt 19	Testprojekt 51	3	5	4	1	3	4	1	4	2	5	3
Projekt 19	Testprojekt 52	3	4	3	1	2	2	1	4	2	3	3
Projekt 19	Testprojekt 53	3	4	3	1	2	2	1	4	2	3	3
Projekt 19	Testprojekt 54	2	5	3	1	3	3	1	2	2	5	3
Projekt 19	Testprojekt 55	2	5	3	1	3	3	2	2	2	5	3
Projekt 20	Testprojekt 56	2	5	3	1	3	3	3	2	2	5	3
Projekt 21	Testprojekt 57	2	5	3	2	3	3	3	2	2	5	3
Projekt 21	Testprojekt 58	2	5	3	2	3	3	3	2	2	5	3
Projekt 22	Testprojekt 59	2	5	3	2	3	3	4	2	2	5	3
Projekt 23	Testprojekt 60	2	5	3	2	3	3	4	2	2	5	3
Projekt 23	Testprojekt 61	3	3	3	1	2	2	1	5	2	3	4
Projekt 23	Testprojekt 62	3	5	5	2	2	2	2	5	3	3	2
Projekt 23	Testprojekt 63	5	5	5	1	2	2	3	5	2	5	3
Projekt 23	Testprojekt 64	5	5	5	3	2	2	4	5	2	5	4
Projekt 23	Testprojekt 65	5	5	5	1	3	3	3	5	2	5	3

Projekt 24	Testprojekt 66	5	5	5	2	2	2	4	5	2	5	3
Projekt 24	Testprojekt 67	2	4	3	1	2	3	2	2	2	3	3
Projekt 24	Testprojekt 68	2	4	4	1	2	3	3	2	2	3	3
Projekt 24	Testprojekt 69	2	4	4	1	2	3	3	2	2	3	3
Projekt 24	Testprojekt 70	2	4	4	1	2	4	3	2	2	3	3
Projekt 25	Testprojekt 74	3	4	3	2	2	2	1	4	2	5	3
Projekt 25	Testprojekt 75	3	4	3	2	2	2	2	4	2	3	3
Projekt 24	Testprojekt 71	3	4	3	2	2	2	3	4	2	3	3
Projekt 26	Testprojekt 76	3	4	3	2	2	2	3	4	2	3	3
Projekt 25	Testprojekt 72	3	4	3	2	2	3	4	4	2	5	3
Projekt 25	Testprojekt 72	3	4	3	2	2	3	4	4	2	5	3
Projekt 25	Testprojekt 73	3	5	3	2	2	3	3	4	2	5	3
Projekt 25	Testprojekt 73	3	5	3	2	2	3	3	4	2	5	3
Projekt 26	Testprojekt 77	2	3	3	1	3	5	1	2	3	3	3
Projekt 26	Testprojekt 77	2	3	3	1	3	5	1	2	3	3	3
Projekt 26	Testprojekt 78	4	3	4	1	3	5	2	2	3	3	3
Projekt 26	Testprojekt 78	4	3	4	1	3	5	2	2	3	3	3
Projekt 26	Testprojekt 79	2	3	3	1	3	5	1	2	3	3	3
Projekt 26	Testprojekt 80	5	5	5	4	3	3	1	4	2	3	3
Projekt 26	Testprojekt 81	5	4	5	4	3	2	2	5	2	3	3
Projekt 26	Testprojekt 82	5	5	5	4	2	3	3	5	2	3	3
Projekt 26	Testprojekt 83	5	5	5	3	4	3	3	5	2	3	3
Projekt 27	Testprojekt 84	5	5	5	3	3	3	3	4	2	3	3
Projekt 27	Testprojekt 85	2	4	3	1	4	3	1	3	2	3	3

## B.2. Bewertung der Testeinflussfaktoren

Projekt 27	Testprojekt 86	3	4	4	2	3	3	3	4	2	3	3
Projekt 27	Testprojekt 87	4	4	4	1	3	3	2	2	2	3	2
Projekt 27	Testprojekt 88	4	3	5	2	3	3	2	2	2	4	3
Projekt 28	Testprojekt 89	2	4	3	1	2	3	2	2	2	3	2
Projekt 28	Testprojekt 90	4	5	5	1	2	3	2	2	2	3	3
Projekt 28	Testprojekt 91	2	4	4	1	3	3	2	2	2	3	2
Projekt 28	Testprojekt 92	2	4	4	1	4	3	2	4	2	4	2
Projekt 28	Testprojekt 93	2	4	4	3	3	3	2	4	2	3	3
Projekt 28	Testprojekt 94	4	4	3	3	4	3	2	2	2	3	4
Projekt 28	Testprojekt 97	2	4	3	1	4	3	1	2	2	3	1
Projekt 28	Testprojekt 95	2	4	3	2	4	3	2	2	2	4	2
Projekt 28	Testprojekt 96	2	4	4	3	4	3	2	2	2	4	2
Projekt 29	Testprojekt 98	2	2	4	2	3	3	1	1	2	2	1
Projekt 29	Testprojekt 99	3	2	5	2	3	3	2	1	2	2	3
Projekt 29	Testprojekt 100	2	2	4	1	3	3	2	1	2	2	3
Projekt 29	Testprojekt 101	2	2	5	1	2	3	1	1	2	2	4
Projekt 29	Testprojekt 102	2	2	4	1	3	3	1	1	2	2	3
Projekt 29	Testprojekt 103	2	2	4	1	2	3	2	1	2	2	3
Projekt 29	Testprojekt 104	2	2	4	2	3	4	2	1	2	2	3
Projekt 29	Testprojekt 105	2	2	4	2	3	3	2	1	2	2	3
Projekt 29	Testprojekt 106	2	2	4	1	3	4	2	1	2	2	1
Projekt 29	Testprojekt 107	2	2	3	1	4	4	2	1	2	4	1
Projekt 29	Testprojekt 108	2	2	3	2	4	5	2	1	2	4	1
Projekt 29	Testprojekt 109	2	2	4	2	4	4	2	1	2	2	2

Projekt 29	Testprojekt 110	2	2	4	2	4	4	2	1	2	2	3
Projekt 29	Testprojekt 111	2	2	3	2	4	5	2	1	2	3	1
Projekt 29	Testprojekt 112	2	2	4	1	3	5	2	1	2	3	3
Projekt 29	Testprojekt 113	2	2	5	1	2	4	2	1	2	2	2
Projekt 29	Testprojekt 114	2	2	4	2	3	4	2	1	2	2	2
Projekt 29	Testprojekt 115	2	2	4	2	3	4	2	1	2	2	2

Tabelle B.4.: Bewertung der Testeinflussfaktoren entsprechend der bijektiven Funktion  $S$  in Formel 5.4

## B.3. Expertenschätzungen

Projekt	Version = Testprojekt	Test	Test	Test
		-management	-design	-ausführung
Projekt 1	Testprojekt 2	2,55	5,3	3
Projekt 1	Testprojekt 3	4,5	11,5	5
Projekt 2	Testprojekt 4	12	13,7	16,2
Projekt 3	Testprojekt 5	7,93	9,18	8,1
Projekt 4	Testprojekt 7	10,52	10,8	9,3
Projekt 5	Testprojekt 8	8,5	6	14
Projekt 7	Testprojekt 10	7,84	9,7	9,7
Projekt 7	Testprojekt 11	13,75	10,5	14,5
Projekt 13	Testprojekt 19	15,1	14	14
Projekt 15	Testprojekt 21	4,75	1	2,5
Projekt 15	Testprojekt 23	5	1,1	5
Projekt 16	Testprojekt 24	17,03	17,2	15,6
Projekt 17	Testprojekt 28	17,7	51,7	21,9
Projekt 17	Testprojekt 29	12	15	15
Projekt 17	Testprojekt 30	14,2	13,4	14,7
Projekt 18	Testprojekt 33	5	7	15,5
Projekt 18	Testprojekt 41	59,1	118,4	45,2
Projekt 18	Testprojekt 42	4,2	0,5	0
Projekt 18	Testprojekt 43	10,2	21,55	0
Projekt 18	Testprojekt 44	8,8	12,7	0
Projekt 18	Testprojekt 45	6,7	4,85	0
Projekt 19	Testprojekt 46	6,6	9,4	0
Projekt 19	Testprojekt 47	10,45	11,55	3
Projekt 19	Testprojekt 48	1	5	1
Projekt 19	Testprojekt 49	1,3	2,05	1
Projekt 19	Testprojekt 52	11,3	11	11,1
Projekt 19	Testprojekt 53	10	7,7	9,7
Projekt 23	Testprojekt 61	10,2	3	3
Projekt 23	Testprojekt 62	11,2	11,55	14,25
Projekt 23	Testprojekt 63	2,75	2	2,5
Projekt 23	Testprojekt 64	2,25	1,95	3
Projekt 23	Testprojekt 65	2,25	5	4

## B. Projektdaten

Projekt 24	Testprojekt 66	3	3,5	3
Projekt 24	Testprojekt 68	2	7,1	2
Projekt 24	Testprojekt 69	5,25	7,6	5
Projekt 24	Testprojekt 70	0,3	1,3	0,5
Projekt 26	Testprojekt 77	8,5	10	10
Projekt 26	Testprojekt 78	10,73	10	11
Projekt 26	Testprojekt 80	7,05	1,7	5,4
Projekt 26	Testprojekt 81	5,15	1,45	3,3
Projekt 27	Testprojekt 85	60,1	72	72
Projekt 27	Testprojekt 86	4,31	1,8	3,6
Projekt 27	Testprojekt 87	10,7	12,2	9,2
Projekt 27	Testprojekt 88	3,8	3,5	4
Projekt 28	Testprojekt 91	6	8,25	4,5
Projekt 28	Testprojekt 92	4,7	15	0
Projekt 28	Testprojekt 93	3,2	4,7	0
Projekt 28	Testprojekt 94	3,5	4,6	0,5
Projekt 28	Testprojekt 97	21,5	24,6	13
Projekt 29	Testprojekt 100	6,4	3,4	6,6
Projekt 29	Testprojekt 106	13,3	12,6	8,4
Projekt 29	Testprojekt 107	15,7	8,8	10
Projekt 29	Testprojekt 108	20,3	20,3	24
Projekt 29	Testprojekt 109	22,8	23,6	18
Projekt 29	Testprojekt 110	19,7	26	12,3
Projekt 29	Testprojekt 111	18,45	25,95	14,8
Projekt 29	Testprojekt 112	17	15,2	14,8
Projekt 29	Testprojekt 113	0,56	2,2	4,5
Projekt 29	Testprojekt 114	20,3	20	25

Tabelle B.5.: Expertenschätzungen in PT der Firma CRM IT für die Stufe Systemtest

# C. Evaluation des Modells für Testeinflussfaktoren

## C.1. Finalen quantitative Auswirkungen der Testeinflussfaktoren

Die Tabellen in diesem Abschnitt sind eine Ergänzung zu der Tabelle 5.27 für die Testaktivität Testdesign. Aufgeführt sind die finalen quantitativen Auswirkungen (FQA) der Testeinflussfaktoren für die Testaktivitäten auf der Stufe Systemtest. Das Verfahren zur Bestimmung der FQA wurde in Abschnitt 5.3.2.2 vorgestellt.

	sehr niedrig	niedrig	normal	hoch	sehr hoch
Qualitätsziele	0,82	0,90	1,00	1,21	1,57
Wiederverwendbarkeit von Testmitteln			1,00	0,78	0,55
Qualifikation und Erfahrung	1,54	1,19	1,00	0,81	0,45
Kommunikationsaufwand		1,06	1,00	0,97	
Zusammenarbeit	0,73	0,87	1,00	1,27	1,64
Stabilität und Verfügbarkeit der Testumgebung	0,98	0,99	1,00	1,01	1,02
Testinfrastruktur	0,90	0,95	1,00	1,04	1,11
Gesamttestorganisation	0,90	0,95	1,00	1,04	1,08
Grad des Dokumentationsbedarfs	0,87	0,94	1,00	1,02	1,07
Zeitplanung und Termine	1,24	1,11	1,00	0,87	0,77

Tabelle C.1.: Berechnete FQA je qualitativer Ausprägung der Testeinflussfaktoren für die Testaktivität Testmanagement. Dabei wurden die IQA zu 60 % berücksichtigt.

### C. Evaluation des Modells für Testeinflussfaktoren

	sehr niedrig	niedrig	normal	hoch	sehr hoch
Qualität der Testbasis	1,47	1,17	1,00	0,92	0,82
Qualitätsziele	0,90	0,95	1,00	1,10	1,26
Wiederverwendbarkeit von Testmitteln			1,00	0,78	0,57
Qualifikation und Erfahrung	1,10	1,04	1,00	0,96	0,92
Kommunikationsaufwand		0,97	1,00	1,02	
Zusammenarbeit	0,93	0,97	1,00	1,06	1,14
Stabilität und Verfügbarkeit der Testumgebung	1,13	1,06	1,00	0,97	0,95
Testinfrastruktur	1,15	1,07	1,00	0,96	0,91
Gesamttestorganisation	1,50	1,20	1,00	0,92	0,84
Grad des Dokumentationsbedarfs	0,87	0,93	1,00	1,03	1,08
Zeitplanung und Termine	1,07	1,03	1,00	0,98	1,00

Tabelle C.2.: Berechnete FQA je qualitativer Ausprägung der Testeinflussfaktoren für die Testaktivität Testausführung. Dabei wurden die IQA zu 90 % berücksichtigt.

## C.2. Ausgabe der Statistikumgebung R

In diesem Abschnitt sind die Ergebnisse der Regressionsanalysen zusammengefasst, die zur Berechnung der FQA mit der höchsten Vorhersagegenauigkeit genutzt worden sind. In Tabelle C.3 sind die Ergebnisse für die quadrierte multiple Korrelation  $R^2$  und das korrigierte  $R^2$  aufgeführt. Mit  $R^2$  wird ausgedrückt, wie stark die Korrelation zwischen dem Kriterium und dem vorhergesagten Wert ist bzw. wie stark die Beobachtungen von der Regressionsgeraden abweichen [BS10, S. 192 u. S. 347]. Dabei ist das korrigierte  $R^2$  stabil gegenüber der Anzahl der Prädiktoren. Der Wert von  $R^2$  liegt im Intervall von 0 bis 1. Bei der Annahme des Wertes 1 zeigt die quadrierte multiple Korrelation einen perfekten linearen Zusammenhang an.

	Testmanagement	Testdesign	Testausführung
$R^2$	0,43	0,69	0,33
Korrigiertes $R^2$	0,35	0,64	0,21

Tabelle C.3.: Übersicht über die  $R^2$  und die korrigierten  $R^2$  für die Testaktivitäten Testmanagement, Testdesign und Testausführung

## C.2. Ausgabe der Statistikumgebung R

Die Tabellen C.4 bis C.6 zeigen die direkten Ausgaben der Statistikumgebung R. Die Werte sind dabei nach den allgemeinen Methoden zur multiplen Regression berechnet worden [BS10].

	Koeffizient	Standard-abweichung	t-Test-statistik	Signifikanz
Konstante	-0,23	0,33	-0,69	0,49
Qualitätsziele	2,43	2,89	0,84	0,41
Wiederverwendbarkeit von Testmitteln	0,82	0,44	1,86	0,07
Qualifikation und Erfahrung	3,49	1,79	1,95	0,06
Kommunikationsaufwand	-1,36	3,22	-0,42	0,68
Zusammenarbeit	3,18	1,85	1,72	0,09
Stabilität und Verfügbarkeit der Testumgebung	-0,37	1,16	-0,32	0,75
Testinfrastruktur	-0,99	2,26	-0,44	0,66
Gesamttestorganisation	-1,05	2,66	-0,39	0,7
Grad des Dokumentationsbedarfs	0,53	2,11	0,25	0,8
Zeitplanung und Termine	1,9	2,99	0,64	0,53

Tabelle C.4.: Ausgabe der Statistikumgebung R für die Testaktivität Testmanagement

C. Evaluation des Modells für Testeinflussfaktoren

	<b>Koeffizient</b>	<b>Standard-abweichung</b>	<b>t-Test-statistik</b>	<b>Signifikanz</b>
Konstante	-0,28	0,47	-0,61	0,55
Qualität der Testbasis	7,52	2,83	2,65	0,01
Qualitätsziele	2,02	2,78	0,73	0,47
Wiederverwendbarkeit von Testmitteln	-0,28	0,68	-0,41	0,69
Qualifikation und Erfahrung	-3,13	2,58	-1,21	0,24
Kommunikationsaufwand	-3,03	4,36	-0,69	0,49
Zusammenarbeit	-2,04	3,79	-0,54	0,59
Stabilität und Verfügbarkeit der Testumgebung	-3,45	1,55	-2,23	0,03
Testinfrastruktur	-2,19	2,81	-0,78	0,44
Gesamtprojektorganisation	8,94	6,27	1,43	0,16
Grad des Dokumentationsbedarfs	-1,75	4,16	-0,42	0,68
Zeitplanung und Termine	-5,28	5,11	-1,03	0,31

Tabelle C.5.: Ausgabe der Statistikumgebung R für die Testaktivität Testdesign

	Koeffizient	Standard-abweichung	t-Test-statistik	Signifikanz
Konstante	-0,38	0,34	-1,11	0,27
Qualität der Testbasis	1,80	2,06	0,87	0,39
Qualitätsziele	-0,14	2,01	-0,07	0,94
Wiederverwendbarkeit von Testmitteln	-0,17	0,47	-0,36	0,72
Qualifikation und Erfahrung	2,46	1,86	1,33	0,19
Kommunikationsaufwand	-0,36	3,28	-0,11	0,91
Zusammenarbeit	5,57	2,20	2,53	0,01
Stabilität und Verfügbarkeit der Testumgebung	-1,67	1,27	-1,31	0,19
Testinfrastruktur	-3,28	1,46	-2,24	0,03
Gesamttestorganisation	5,80	3,14	1,84	0,07
Grad des Dokumentationsbedarfs	1,09	2,79	0,39	0,70
Zeitplanung und Termine	1,10	2,64	0,42	0,68

Tabelle C.6.: Ausgabe der Statistikumgebung R für die Testaktivität Testausführung