



Thema:

Evaluierung und Auswahl eines Testmanagementsystems

Bachelorarbeit

Fakultät für Informatik
Arbeitsgruppe Wirtschaftsinformatik

Themensteller: Dr. Stephanie Nagel-Heyer
Betreuer: Prof. Dr. rer. pol. habil. Hans-Knud Arndt

vorgelegt von: Philipp Ernst

Abgabetermin: 13. August 2012

Inhaltsverzeichnis

Inhaltsverzeichnis	II
Verzeichnis der Abkürzungen und Akronyme	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
1 Einleitung.....	7
2 Grundlagen.....	9
2.1 Das International Software Testing Qualifications Board	9
2.2 Testmanagement.....	10
2.3 Der fundamentale Testprozess	13
2.3.1 Testplanung und Steuerung.....	15
2.3.2 Testanalyse und Testdesign	16
2.3.3 Testrealisierung und Testdurchführung	19
2.3.4 Testauswertung und Bericht.....	21
2.3.5 Abschluss der Testaktivitäten	23
2.4 Optimierung des Testprozesses durch TPI.....	24
2.5 Werkzeugunterstützung.....	27
3 Material und Methoden.....	29
3.1 Strukturierte Systemanalyse	29
3.2 Der Untersuchungsprozess	30
3.3 Das IsoMetrics-Verfahren	33
4 Ergebnisse und Diskussion der analytischen Betrachtungen.....	34
4.1 Analyse der Ist-Situation im Bereich Testmanagement.....	34
4.1.1 Testprozessaktivitäten bei UE/I	34
4.1.2 Wichtige Objekte und Beziehungen im Testprozess von UE/I.....	41
4.1.3 Anwendung des TPI-Modells	44
4.1.4 Methoden und Werkzeugeinsatz.....	48
4.2 Ableitung der Soll-Situation (Anforderungsanalyse).....	49
4.2.1 Anforderungen auf Basis des Testprozesses	49
4.2.2 Anforderungen durch TPI-Maßnahmen.....	54
4.2.3 Anforderungen durch sonstige Betrachtungen.....	59
4.2.4 Anforderungskatalog.....	63
4.3 Evaluierungsprozess.....	65
4.3.1 Marktrecherche und erste Filterung	65
4.3.2 Weitere Vorauswahl der Werkzeuge	71
4.3.3 Bewertung des Anforderungskataloges	75
4.3.4 Usability Evaluierung mittels IsoMetrics-Verfahren	79
4.3.5 Abschließende Bewertung	83

5 Zusammenfassung und Ausblick	86
A Abbildungen des Anhangs	89
B Tabellen des Anhangs	96
Literaturverzeichnis	110

Verzeichnis der Abkürzungen und Akronyme

bzw.	beziehungsweise
CSV	Comma-Separated Values
ISTQB	International Software Testing Qualifications Board
IT	Informationstechnologie
LAMP	Linux-Apache-MySQL-PHP
LDAP	Lightweight Directory Access Protocol
LSY	Lufthansa Systems
LTLS	Lufthansa Technik Logistik Services GmbH
SaaS	Software as a Service
SQL	Structured Query Language
TPI	Test Process Improvement
UE/I	Abteilung für Informationstechnologie der LTLS
z. B.	zum Beispiel

Abbildungsverzeichnis

Abb. 2.1: Entwicklungs- und Testprozess	11
Abb. 2.2: Die Grundstruktur des fundamentalen Testprozesses.....	14
Abb. 2.3: Das TPI-Modell	25
Abb. 3.1: Erzeugung von Ist- und Soll-Modell	29
Abb. 3.2: Der Untersuchungsprozess	30
Abb. 4.1: Von den Teilergebnissen zum Gesamtergebnis.....	83
Abb. A.1: Aktivitätsdiagramm – Initiale Testplanung	89
Abb. A.2: Aktivitätsdiagramm – Teststeuerung	90
Abb. A.3: Aktivitätsdiagramm – Testanalyse und Design	90
Abb. A.4: Aktivitätsdiagramm – Testplanung auf Basis von Zwischenreleases.....	91
Abb. A.5: Aktivitätsdiagramm – Testrealisierung und Durchführung	92
Abb. A.6: Aktivitätsdiagramm – Testauswertung und Bericht.....	93
Abb. A.7: Aktivitätsdiagramm – Abschluss der Testaktivitäten	94
Abb. A.8: ER-Modell – Beziehungen wichtiger Objekte im Testprozess.....	95

Tabellenverzeichnis

Tab. 2.1: Aufbau der TPI-Matrix.....	26
Tab. 4.1: TPI-Matrix – Istzustand bei UE/I vor der Projektdurchführung	45
Tab. 4.2: TPI-Matrix – Sollzustand bei UE/I gemäß Projektzielvereinbarung	46
Tab. 4.3: TPI-Matrix – Istzustand bei UE/I nach der Projektdurchführung.....	47
Tab. 4.4: Marktrecherche: verwendete Symbole und Abkürzungen	69
Tab. 4.5: Marktrecherche: Ergebnis der frühzeitigen Filterung	70
Tab. 4.6: Ergebnis – erste Vorauswahlstufe	72
Tab. 4.7: Ergebnis – zweite Vorauswahlstufe	73
Tab. 4.8: Gewichtungen nach Prioritäten der Bewertungskriterien	75
Tab. 4.9: Nominal- und Ordinalskala als Bewertungsmaßstäbe	76
Tab. 4.10: Nutzungsverteilung der verwendeten Skalen	77
Tab. 4.11: Ergebnis der Bewertung des Anforderungskataloges	79
Tab. 4.12: Bewertung mittels IsoMetrics – ein Beispiel	80
Tab. 4.13: Ergebnis der Usability-Evaluierung.....	82
Tab. 4.14: Gesamtergebnis des Evaluierungsprozesses	84
Tab. B.1: TPI-Matrix – gegenwärtiger Sollzustand bei UE/I.....	96
Tab. B.2: Offene Maßnahmen zur Erreichung von TPI-Stufe 4-6	97
Tab. B.3: Validierter und inhaltlich priorisierter Anforderungskatalog.....	100
Tab. B.4: Vollständig bewerteter Anforderungskatalog.....	109

1 Einleitung

In der Softwareentwicklung nimmt der Bereich des Softwaretests eine besondere Stellung ein. Das effiziente Entwickeln von komplexen Softwareprodukten hoher Qualität ist ohne strukturierte und gut organisierte Testarbeiten nicht möglich. Das Vorhandensein eines effizienten Testmanagements im Softwaretestprozess ist deshalb unabdingbar. Mit dieser Problematik ist auch die Lufthansa Technik Services GmbH (LTLS) konfrontiert.

Bei der LTLS sind viele unterschiedliche IT-Systeme im Einsatz, die stark miteinander vernetzt sind und ständig weiterentwickelt werden. Die umfangreichen Anforderungen an diesen Systemverbund sind ein Grund für die steigende Komplexität im Softwaretestprozess. Um das Risiko in den IT-Projekten zu minimieren, sind deshalb ein strukturierter Testprozess und ein effizientes Testmanagement notwendig. Die Abteilung für Informationstechnologie (UE/I) orientiert sich am standardisierten Testprozess des *International Software Testing Qualifications Board* (ISTQB). Unter Anwendung des *Test Process Improvement Modells* (TPI), wird der vorhandene Testprozess kontinuierlich weiterentwickelt und optimiert. Das Testmanagement von UE/I wird bisher jedoch nicht durch ein Testmanagementsystem unterstützt.

*„Ohne passende Werkzeuge kann der Testprozess nicht optimal durchgeführt werden. Falls diese nicht vorhanden sind, muss eine Auswahl und Beschaffung der Werkzeuge frühzeitig eingeleitet werden.“*¹ Dieses Zitat von Andreas Spillner verdeutlicht bereits die Wichtigkeit des Werkzeugeinsatzes im Testprozess. Es untermauert außerdem die Notwendigkeit und Bedeutung der Durchführung einer Evaluierung von Testmanagementwerkzeugen bei UE/I.

Es wurde viel Aufwand bei UE/I investiert, um einen passenden Prozess zu finden. Deshalb ist es das Ziel der Evaluierung ein Werkzeug (Testmanagementsystem) zu finden, das zum vorhandenen Testprozess von UE/I passt und diesen möglichst gut unterstützen und abbilden kann. Das Werkzeug muss dabei den umfassenden und speziellen Anforderungen des Großunternehmens genügen. Es ist nicht das Ziel einen Prozess zu finden, der zu einem Werkzeug passt. Letztere Situation ist nicht selten der Fall, denn häufig werden die vorhandenen Prozesse aufwändig auf das ausgewählte Werkzeug abgestimmt. Es soll eine fachlich und wissenschaftlich fundierte Auswahl eines Testmanagementsystems erfolgen, die entsprechende Handlungsempfehlungen rechtfertigt. Die Ergebnisse sollen somit als fachliche und analytische Grundlage für die Entscheidungsträger dienen.

In der Arbeit wird zunächst grundlegendes Wissen über den ISTQB-Testprozess und die Aufgaben des Testmanagements vermittelt. Daraufhin werden die Verfahren und Methoden beschrieben, die zur Ergebnisbildung eingesetzt worden sind. Der

¹ Spillner, 2011, S. 13.

Testprozess von UE/I wird durch eine strukturierte Systemanalyse tiefgründig analysiert und die Anforderungen an ein Testmanagementsystem ermittelt. Anschließend wird der Evaluierungsprozess auf Grundlage der Ergebnisse der analytischen Betrachtungen durchgeführt. Ein besonderes Augenmerk liegt zudem auf einer Untersuchung der Benutzerfreundlichkeit der Testmanagementsysteme, um die zukünftige Akzeptanz des Systems bei den Nutzern sicherzustellen. Dabei kommt das standardisierte IsoMetrics-Verfahren zur Beurteilung der Benutzerfreundlichkeit der Werkzeuge zum Einsatz.

In Hinsicht auf existierende Modelle zur Prozessverbesserung wird ausschließlich auf das TPI-Modell eingegangen, da dieses Modell im Testprozess von UE/I Anwendung findet. Die Untersuchungen dieser Arbeit konzentrieren sich auf den Kern des Testmanagements. Es wird deshalb nicht näher auf eng angrenzende Themengebiete, wie Fehlermanagement, Konfigurationsmanagement, Änderungsmanagement oder auch Testautomatisierung eingegangen. Da bei der LTLS ein Fehlermanagementsystem und ein Testautomatisierungssystem eingesetzt werden, finden dennoch Untersuchungen zu den Anbindungsmöglichkeiten der Testmanagementsysteme zu diesen Systemen statt. Weiterhin wird keine Wirtschaftlichkeitsbetrachtung durchgeführt, da es das Ziel der Arbeit ist, die Eignung der Testmanagementsysteme in Hinsicht auf den Testprozess zu untersuchen. Außerdem ist der Nutzen des Einsatzes eines Testmanagementsystems auf monetärer Ebene schwer zu bewerten. Es können einerseits Aussagen über die Kosten der einzelnen Systeme getroffen werden, genauere Abschätzungen über das erreichbare Sparpotential sind jedoch nur schwierig vorzunehmen.

2 Grundlagen

In diesem Kapitel wird ein Überblick über die in der Arbeit verwendeten fachlichen Inhalte vermittelt. Der Grundlagenteil orientiert sich stark an den zwei deutschsprachigen, ISTQB-konformen Standardwerken von Andreas Spillner. Die Inhalte werden weiterhin durch allgemeingültige Auffassungen und Ansätze von Autoren nicht ISTQB-konformer Fachliteratur ergänzt.

Aufgrund der maßgeblichen Bezugnahme auf die ISTQB-Lehrinhalte, wird zunächst ein kurzer Einblick in die Inhalte und Aufgaben der ISTQB-Vereinigung gegeben (siehe Kapitel 2.1). Daraufhin wird auf das Testmanagement eingegangen (siehe Kapitel 2.2) und der fundamentale Testprozess, wie er nach ISTQB definiert ist, beschrieben (siehe Kapitel 2.3). In Kapitel 2.4 ist das Modell zur Optimierung des Testprozesses durch Test Process Improvement dargestellt. Abschließend wird in Kapitel 2.5 auf die Werkzeuge eingegangen, die den Testprozess im Bereich des Testmanagements unterstützen. Im Mittelpunkt steht dabei das Testmanagementsystem selbst.

2.1 Das International Software Testing Qualifications Board

Beim International Software Testing Qualifications Board (ISTQB) handelt es sich um eine im Jahr 2002 gegründete, gemeinnützige Organisation, die auf die Freiwilligenarbeit von internationalen Testingenieuren und Experten aufbaut.² Die Hauptaufgabe des ISTQB ist nach Spillner die einheitliche und vergleichbare Darstellung von Prüfungs- und Lehrinhalten im Bereich Softwaretest.³ Die immer wichtiger werdende Aufgabe des professionellen Prüfens und Testens im Softwareentwicklungsprozess, bedarf einer fundierten Ausbildung. In Folge dieser Erkenntnis hat die ISTQB-Vereinigung mit ihrem drei-stufigen Ausbildungsprogramm zum „ISTQB – Certified Tester“ einen weltweit anerkannten Standard zur Aus- und Weiterbildung für Softwaretester geschaffen.⁴

² Vgl. <http://www.istqb.org/about-istqb.html> (letzter Zugriff: 24.05.2012).

³ Vgl. Spillner, 2011, S. 2.

⁴ Vgl. http://german-testing-board.info/de/certified_tester.shtm (letzter Zugriff: 24.05.2012).

2.2 Testmanagement

In diesem Abschnitt soll ein grundlegender Überblick über die wichtigsten Inhalte und Aufgaben des Testmanagements gegeben werden. Zunächst soll die Bedeutung des Testens geklärt werden, da das Verständnis dieses Begriffes die Grundlage für alle weiteren Betrachtungen ist.

„Testen ist der Prozess, der [...] Auskunft über die Qualität und die damit verbundenen Risiken liefert.“⁵

Die Definition des Testens zeigt, dass die Qualität einer Software durch Softwaretests gemessen wird. Mit der ermittelten Qualität sind außerdem gewisse Risiken verbunden, sodass die Qualität der Software den Erfolg des Projektes bestimmen kann.⁶ Weiterhin sind die Softwareentwickler mit stetig steigenden Qualitätsansprüchen konfrontiert. Testen ist somit ein bedeutender Prozess, der die Sicherstellung einer angemessenen Softwarequalität, über den gesamten Softwarelebenszyklus hinweg, ermöglicht.⁷ Die Qualität einer Software wird nach [ISO 9126] folgendermaßen definiert:

„Gesamtheit der Funktionalitäten und Merkmale eines Softwareprodukts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.“⁸

Um den steigenden Qualitätsanforderungen und der zunehmenden Beschleunigung der Entwicklungszeiten gerecht zu werden, reicht einfaches Testen in der Regel nicht mehr aus. Es ist daher die Aufgabe des Testmanagements die Testaktivitäten zu strukturieren und zu steuern. Durch ein wohldurchdachtes und strukturiertes Vorgehen im Testprozess können einerseits Kosten und Aufwände gesenkt, andererseits kann die Aussagekraft und Zuverlässigkeit der Tests bedeutend erhöht werden. Eine frühzeitige Planung und Strukturierung der Testaktivitäten durch das Testmanagement ist deshalb von besonderer Wichtigkeit.⁹ Durch einen geplanten und durchdachten Testprozess kann sichergestellt werden, dass wichtige Testaktivitäten nicht erst am Ende der Entwicklungsphase durchgeführt werden; und dort eine eventuelle Fehlerbehebung sehr teuer für das Unternehmen werden kann. Weiterhin können Zeit- und Ressourcenbeschränkungen die Qualität einer Software sehr negativ beeinflussen. Zur Sicherstellung der Qualität einer Software muss somit möglichst früh mit dem Testen

⁵ Koomen, et al., 2008, S. 12.

⁶ Vgl. Colditz, 2010, S. 394ff.

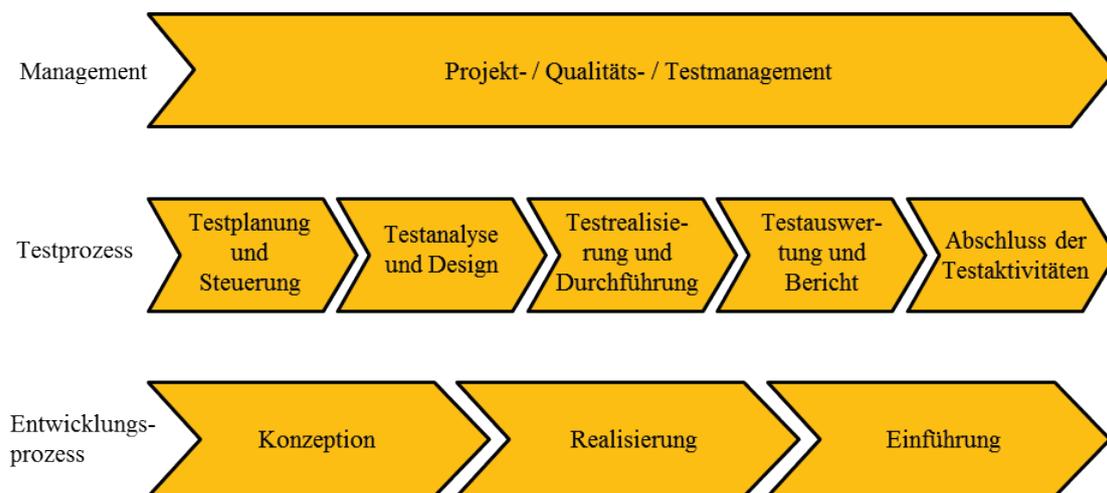
⁷ Vgl. Jackson, 2009, S. 37.

⁸ Hamburg, 2010, S. 46.

⁹ Vgl. Jackson, 2009, S. 37.

begonnen werden. Der Testprozess sollte demnach simultan zum Entwicklungsprozess durchgeführt werden.¹⁰

In Abbildung 2.1 werden der Testprozess, der Entwicklungsprozess und die Managementebene allgemein gegenübergestellt. Es wurde dabei kein beliebiger Ansatz eines Testprozesses für die Abbildung gewählt. Da in Kapitel 2.3 detailliert auf den fundamentalen Testprozess des ISTQB eingegangen wird, ist dieser bereits in der Darstellung abgebildet. Die fünf aufgezeigten Testphasen werden in Kapitel 2.3 ausführlich beschrieben.



Quelle: in Anlehnung an Colditz, 2010, S. 395

Abb. 2.1: Entwicklungs- und Testprozess

Aus Abb. 2.1 wird der zeitliche Zusammenhang der drei Bereiche deutlich. Die Testaktivitäten des Testprozesses werden parallel zum Entwicklungsprozess durchgeführt. Auf Managementebene ist weiterhin zwischen Projekt-, Qualitäts- und Testmanagement unterscheidbar. Alle drei Instanzen haben dabei im Verlauf des gesamten Entwicklungs- und Testprozesses kontrollierenden und steuernden Einfluss auf die Aktivitäten.

Da ein gutes Testmanagement Voraussetzung für einen erfolgreichen Testprozess ist, muss eine sorgfältige Testplanung erfolgen, die von qualifiziertem Personal durchgeführt wird. Ein sinnvoll zusammengestelltes Team und dessen Verfügbarkeit spielt daher eine wesentliche Rolle für die effiziente Testplanung.¹¹ Zwei wichtige Rollen im Testteam übernehmen hierbei der Testmanager und der Tester. Der Testmanager plant, steuert und überwacht im Normalfall den gesamten Testprozess. Das heißt seine Aufgabe ist es dafür zu sorgen, dass alle Testaktivitäten zum richtigen

¹⁰ Vgl. Colditz, 2010, S. 395.

¹¹ Vgl. ebd., S. 395f.

Zeitpunkt stattfinden und hierfür alle notwendigen Ressourcen vorhanden sind.¹² Er ist der Leiter des Testteams und ist für die Erstellung und Umsetzung des Testplans zuständig. Außerdem ist er für die Berichterstattung der Testergebnisse und des Testfortschritts an die Projekt- und Entwicklungsleiter zuständig.¹³ Der Tester hingegen ist für die Testausführung und Abarbeitung des Testplans zuständig. Er dokumentiert die Ergebnisse der Tests und erstellt für den Fall eines auftretenden Fehlers einen Abweichungsbericht.¹⁴

Das wohl bedeutendste Aufgabengebiet des Testmanagements ist der Bereich der Testplanung. Die wesentlichen Aufgaben der Testplanung werden an dieser Stelle kurz aufgelistet.

- Ausarbeitung der Teststrategie
- Ableiten der Testaktivitäten aus der Strategie
- Planung und Aufwandsschätzung der Aktivitäten
- Verteilen der Aktivitäten auf das oder die zur Verfügung stehenden Testteams¹⁵

Zusammenfassend bleibt zu sagen, dass der Testprozess im Unternehmen durch den gezielten Einsatz von Testmanagement einen höheren Grad an Strukturiertheit als auch Professionalität erreichen kann. Der Aufwand für das Management der einzelnen Testprozessphasen und Testaktivitäten steigt zunehmend an, da die Softwareanwendungen immer komplexer und geschäftskritischer werden. Um diesen Aufwand im Unternehmen möglichst zu reduzieren sind ein effizientes Testmanagement und ein strukturierter Testprozess unverzichtbar.¹⁶

¹² Vgl. Colditz, 2010, S. 396.

¹³ Vgl. Spillner, 2011, S. 290.

¹⁴ Vgl. ebd., S. 292.

¹⁵ Vgl. ebd., S. 124.

¹⁶ Vgl. Jackson, 2009, S. 40f.

Die Ergebnisse einer im Jahr 2005 durchgeführten Logica Studie „Testing Times for Board Rooms“ begründen die Notwendigkeit des Testmanagements. Sie verdeutlichen insbesondere in drei Punkten die zentrale Bedeutung des Testmanagements bezüglich der Herausforderungen im Testprozess:¹⁷

- *Struktur und Sicherheit in den Testprozess bringen!*
- *Produktivität und Effizienz im Testprozess steigern!*
- *Testen an den geschäftlichen Anforderungen ausrichten!*¹⁸

2.3 Der fundamentale Testprozess

Der fundamentale Testprozess bildet die Grundlage für sämtliche Überlegungen im Rahmen dieser Arbeit und soll deshalb ausführlich betrachtet werden. In diesem Abschnitt wird der fundamentale Testprozess, wie er im Lehrplan des ISTQB definiert ist, beschrieben. Das Wort fundamental kann in diesem Zusammenhang als gefestigt, etabliert oder standardisiert verstanden werden. Zur Verständlichkeit des Wortgebildes soll an dieser Stelle der Begriff des Testprozesses erneut aufgegriffen werden, da es verschiedene Definitionsansätze gibt. Im Gegensatz zur Definition des Testens von Koomen (siehe Kapitel 2.2) ist Testen nach Brügge „*der Prozess, in dem Unterschiede zwischen dem erwarteten Verhalten, das durch das Systemmodell spezifiziert ist, und dem beobachteten Verhalten des implementierten Systems gefunden werden sollen*“.¹⁹

Der Prozess des Testens stellt selbst eine Entwicklungsaufgabe oder -phase im Softwareentwicklungsprozess dar. Die oftmals sehr einfachen und allgemeinen Darstellungsformen des Testprozesses reichen für eine strukturierte Durchführung von Tests jedoch nicht aus. Es wird daher ein Prozessmodell benötigt, das die Testaktivitäten und Testphasen ausführlich beschreibt.²⁰ Das Modell des fundamentalen Testprozesses beinhaltet die fünf Testphasen: *Testplanung und Steuerung, Testanalyse und Design, Testrealisierung und Durchführung, Testauswertung und Bericht* sowie den *Abschluss der Testaktivitäten*.²¹

¹⁷ Vgl. Spillner, 2011, S. xiii.

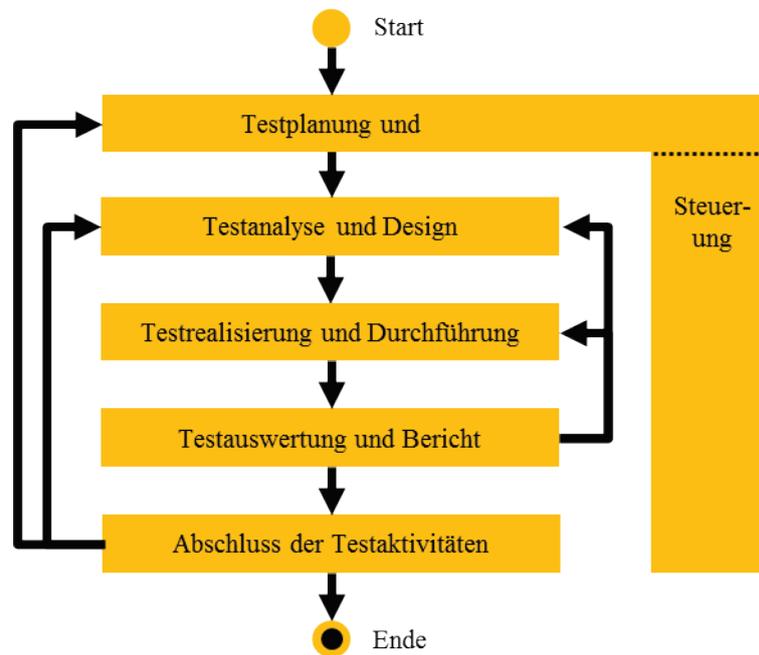
¹⁸ Ebd., S. xiv.

¹⁹ Brügge, 2004, S. 451.

²⁰ Vgl. Spillner, 2010, S. 20.

²¹ Vgl. Spillner, 2011, S. 11.

Die folgende Abbildung zeigt die grundlegende Struktur des Testprozesses nach ISTQB-Vorstellungen:



Quelle: in Anlehnung an Bath, Graham; McKay, Judy, 2011, S. 20.

Abb. 2.2: Die Grundstruktur des fundamentalen Testprozesses

In der Theorie sind die fünf Testphasen in einer sequentiellen Reihenfolge angeordnet, was bedeutet, dass sie im Wesentlichen nacheinander abgearbeitet werden – es ist jedoch möglich, dass Überschneidungen oder parallele Durchführungen von Aktivitäten auftreten.²² Ein möglicher Grund dafür kann Zeitdruck und die damit verbundene Beschleunigung der Testaktivitäten sein. Die sequentielle Reihenfolge gilt jedoch nicht für die steuernden Aktivitäten der *Testplanung und Steuerung*, da diese kontinuierlich und parallel zu allen Aktivitäten im Testprozess durchgeführt werden. Die Darstellung des Testprozesses lässt erkennen, dass gewisse Abhängigkeiten zwischen den Testprozessphasen bestehen und sich bestimmte Aktivitäten im Prozessverlauf gegenseitig beeinflussen. „Der Testprozess ist somit auch als generische Prozessbeschreibung aufzufassen“.²³ Um die Beziehungen und Abhängigkeiten zwischen den Testphasen zu verdeutlichen, soll im Folgenden näher auf die Inhalte und Aufgaben der einzelnen Phasen des fundamentalen Testprozesses eingegangen werden.

²² Vgl. Spillner, 2011, S. 11.

²³ Spillner, 2010, S. 20.

2.3.1 Testplanung und Steuerung

Die Spezifikation einer Software beschreibt dessen Funktion und kann je nach Ausführlichkeit bereits den groben Rahmen für die Testaktivitäten beinhalten. Sobald die Spezifikation (z. B. Fachkonzept) vorhanden ist, kann mit den Tätigkeiten der Testplanung begonnen werden.²⁴ Inhalt der Testplanung ist die Festlegung der Anforderungen, Aufgaben und der Zielsetzung des Testvorhabens. Sämtliche identifizierte Anforderungen bilden dabei die Testbasis. Eine mögliche Definition der Testbasis ist folgende: *„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.“*²⁵

Weiterhin müssen die benötigten Testressourcen, wie Mitarbeiter, Werkzeuge und Hilfsmittel, sowie der zeitliche Ablauf der Tests (z. B. Meilensteine), genau geplant werden. Der Großteil dieser Informationen wird in einem zentralen Dokument der Testplanung und Steuerung, dem Testkonzept, erfasst. Es sollte außerdem die notwendige Organisationsstruktur mit einem angemessenen Testmanagement zur Verfügung stehen.²⁶ Die zeitlichen konkretisierten Planungsdaten für die Durchführung der Tests werden im Testplan festgehalten.²⁷ Weiterhin werden in der Testplanungsphase Risikoeinschätzungen vorgenommen. Verschiedenen Risiken werden dabei unterschiedlichen Testverfahren (Testmethoden) zugeordnet. Diese Risiko-Testverfahren-Kombinationen werden in der Teststrategie (Testhandbuch) festgehalten.²⁸

Bei UE/I stellt das Testhandbuch ein umfassendes, übergeordnetes und innerhalb der Organisation allgemeingültiges Dokument dar. Es enthält die durchzuführenden Testaktivitäten, anzuwendende Testmethoden und alle weiteren Informationen über den Testprozess bei UE/I. Das Testhandbuch dient als Ausgangspunkt für jedes Testkonzept eines Projektes und damit für die Umsetzung einer jeden Testaktivität. Testpläne werden bei UE/I ausgehend vom Testkonzept erstellt und konkretisieren den zeitlichen Ablauf, sowie die zu testenden Objekte und die dafür notwendigen konkreten Tests.²⁹

Im Rückblick auf das zuvor angesprochene Risiko ist es notwendig kritische Systemteile intensiver zu testen als Teile, die als weniger kritisch eingestuft werden.³⁰ Um eine sinnvolle und transparente *„Verteilung der Testaktivitäten auf die »richtigen« Stellen des Softwaresystems zu erreichen“*³¹, sollte diese Aufgabe sehr bedacht und mit

²⁴ Vgl. Thaller, 2004, S. 260.

²⁵ Hamburg, 2010, S. 51 (nach TMap®).

²⁶ Vgl. Spillner, 2010, S. 21.

²⁷ Vgl. Spillner, 2011, S. 27.

²⁸ Vgl. Spillner, 2010, S. 21.

²⁹ Vgl. Testhandbuch – UE/I, S. 8 (interne Quelle).

³⁰ Vgl. Spillner, 2010, S. 21.

³¹ Spillner, 2011, S.13.

großer Sorgfalt durchgeführt werden. Durch Priorisierung werden die durchzuführenden Tests nach ihrer Intensivität geordnet. Für den Fall, dass das Softwareprojekt durch Zeit- oder Ressourcenbeschränkungen in Verzug gerät, werden so die kritischeren Systemteile mit höherer Testintensivität zuerst getestet. Zu beachten sind jedoch auch mögliche Abhängigkeiten zwischen den zu testenden Systemteilen. So kann es sein, dass einige Teile erst getestet werden können, wenn andere bereits erfolgreich getestet wurden, wobei diese Systemteile automatisch eine hohe Risikoklasse besitzen müssen, da andere Systemteile von ihnen abhängen.

Die Aufgabe der Teststeuerung ist es, den Testprozess so zu leiten, dass die in der Testplanung festgelegte Strategie, sowie die Aufgaben und Ziele, bestmöglich erfüllt werden. Dazu gehört ein ständiger Abgleich der Vorgaben der Testplanung mit dem tatsächlichen Verlauf des Projektes über alle Testphasen hinweg. Änderungen, die sich im Laufe der Testaktivitäten an den Planungsvorgaben des Testprojektes ergeben, sind entsprechend aufzunehmen und zu berücksichtigen, um eine effektive Steuerung des Testprozesses zu ermöglichen. Für den Fall, dass in der Testdurchführungsphase vermehrt Fehler an einer unerwarteten Stelle auftreten, ist, entsprechend des veränderten Risikos, die Priorisierung und somit auch die Testintensität anzupassen. Über alle Testprozessphasen hinweg, bildet die Kommunikation der Mitarbeiter die Grundlage für eine erfolgreiche Steuerung und Überwachung des Testprozesses. Sie erfolgt grundsätzlich in Form von Berichterstattungen und wird durch verschiedene Auswertungen unterstützt.³²

2.3.2 Testanalyse und Testdesign

Die Prozessphase der Testanalyse und des Testdesigns ist darauf ausgerichtet, die Testbedingungen festzulegen und Testfälle zu entwerfen. Ausgangspunkt für die Festlegung der Testbedingungen ist die Analyse der, in der Testplanung definierten, allgemeinen Testziele und der Testbasis.³³ Nach Spillner können zur Testbasis „[...] die Anforderungsdokumente, die Architekturdokumente, die Ergebnisse der Risikoanalyse oder auch weitere Dokumente [...], die im Rahmen der Softwareentwicklung genutzt bzw. erstellt werden“³⁴, gehören. Aufbauend auf dem Wissen aus der Testplanungsphase, erfolgt zur Festlegung der Testbedingungen eine weitere Detaillierung bezüglich der Fragestellungen „Was ist zu testen?“ und „Wie ist es zu testen?“. Testkonzept und Strategie dienen hierbei als Basis für die Betrachtungen.

³² Vgl. Spillner, 2011, S. 13.

³³ Vgl. ebd., S. 14.

³⁴ Spillner, 2010, S. 23.

Beispielsweise wird die Frage nach dem „Wie?“ durch die Auswahl der Testverfahren beantwortet. Verbunden mit den gewählten Testverfahren, wird dadurch auch festgelegt, wie intensiv getestet werden soll.³⁵ *„Um Umfang und Struktur der Testbedingungen festzulegen, werden sowohl funktionale als auch nicht-funktionale Eigenschaften des Testobjekts herangezogen.“*³⁶ Zur Verständlichkeit dieser Aussage werden die Begrifflichkeiten funktionale Anforderung, nicht-funktionale Anforderung und das Testobjekt definiert. Das Wort Eigenschaften wird in diesem Zusammenhang durch das Wort Anforderungen substituiert, da die Definitionen von funktionalen und nicht-funktionalen Anforderungen den gleichen Kontext in Bezug auf das Testobjekt widerspiegeln.

Ein Testobjekt ist die *„Komponente oder das System, welches getestet wird“*.³⁷

„Eine funktionale Anforderung definiert eine vom System bzw. von einer Systemkomponente bereitzustellende Funktion [...]“.³⁸

*„Eine nicht-funktionale Anforderung (auch: Qualitätsanforderung) definiert eine qualitative Eigenschaft des gesamten Systems, einer Systemkomponente oder einer Funktion.“*³⁹

Die Festlegung von Aufbau und Umfang der Testbedingungen wird durch verschiedene Faktoren beeinflusst. Sind die Anforderungen in der Testbasis sehr allgemein gehalten, so kann davon ausgegangen werden, dass sich auch keine detaillierten Testbedingungen daraus ergeben. Folglich wirkt sich dies auch auf den Detaillierungsgrad der daraus abzuleitenden Testfälle aus. Der zu bewertende Punkt in diesem Fall ist, ob die Testbasis „gut“ genug ist. Einen weiteren Einflussfaktor stellt das ermittelte Risiko dar. Wurde ein hohes Risiko erkannt, sollten auch Testbedingungen mit entsprechend höherer Detaillierung formuliert werden. Sind die Testbedingungen bestimmt, so bilden diese den Ausgangspunkt für den Entwurf der Testfälle. Die Erarbeitung der Testfälle erfolgt schrittweise und unter Verwendung der, in der Teststrategie für die jeweiligen Testintensitäten festgelegten, Testverfahren.⁴⁰ Die zu erstellenden Testfälle sollten folgende Eigenschaften erfüllen: Wiederholbarkeit, Nachprüfbarkeit und Rückverfolgbarkeit zu den Anforderungen.⁴¹

Nach Spillner und dem ISTQB-Lehrplan sollten folgende drei Punkte beim Entwurf von Testfällen Beachtung finden:

³⁵ Vgl. Spillner, 2011, S. 14.

³⁶ Ebd.

³⁷ Hamburg, 2010, S. 56.

³⁸ Pohl, 2008, S. 15.

³⁹ Ebd., S. 16.

⁴⁰ Vgl. Spillner, 2011, S. 14f.

⁴¹ Vgl. ebd., S. 15 (vgl. auch ISTQB Certified Tester: Advanced Level Syllabus, S. 38).

- *„Vorbedingungen, wie projektbezogene oder lokale Testumgebungen (Testvorrichtungen) und deren geplante Bereitstellung*
- *Anforderungen an die Testdaten*
- *erwartete Ergebnisse aus dem Testfall und dessen Nachbedingungen*⁴²

Genau wie bei der Festlegung der Testbedingungen, wirken sich die Qualität und der Detailgehalt der Testbasis auf den Entwurf der Testfälle aus. Allgemein kann gesagt werden: Je präziser eine Testbasis definiert ist, desto einfacher ist es, daraus Testfälle und deren Sollergebnisse abzuleiten! In der Praxis ist eine präzise Testbasis jedoch meist nicht gegeben. Ungenaue Formulierungen, Lückenhaftigkeit oder strikt veraltete Dokumente sind Beispiele dafür, warum Tester häufig auf ihr Wissen und ihre Erfahrungswerte zurückgreifen müssen.⁴³ Inkonsistente, unpräzise oder unvollständige Anforderungsdokumente der Testbasis können aber auch Ursache für im späteren Verlauf auftretende Fehlerwirkungen sein.⁴⁴ Allerdings ist auch eine qualitativ gute Testbasis kein Garant für die erleichterte Erstellung von Testfällen und bessere Vorhersagbarkeit von dessen Sollergebnissen. Deshalb stellt die Zuhilfenahme des Testorakels einen wesentlichen Schritt dar.⁴⁵ Es dient als Informationsquelle, um wichtige Annahmen über die Sollergebnisse der einzelnen Testfälle treffen zu können und kann beispielsweise in Form einer Anforderungsdefinition oder Spezifikation zur Verfügung stehen.⁴⁶ Beim Testorakel kann es sich auch um ein bereits bestehendes System, das spezielle Wissen einer Person oder ein Benutzerhandbuch handeln.⁴⁷ Eine mögliche qualitätssichernde Maßnahme zur Verbesserung der Aktivitäten, sowie der Analyse und dem Verständnis der Testbasis, ist die Durchführung von Reviews. Durch sie können zum Beispiel Fehler in den entworfenen Testfällen noch vor der Testfallausführung erkannt und korrigiert werden. Neben den Aktivitäten der Testanalyse und des Testdesigns, sollte der Entwurf der Testbedingungen und Testfälle in einer Dokumentation nach IEEE-Standard 829 festgehalten werden. Neben den Aktivitäten der Testanalyse und des –Designs sollte bereits in dieser Testprozessphase die notwendige Testinfrastruktur, also alle Mittel die zur Durchführung der Tests benötigt werden, vorbereitet werden.⁴⁸

⁴² ISTQB Certified Tester: Advanced Level Syllabus, S. 38 (vgl. auch Spillner, 2011, S. 15).

⁴³ Vgl. Spillner, 2011, S. 16.

⁴⁴ Vgl. Pezzè, 2009, S. 423.

⁴⁵ Vgl. Spillner, 2011, S. 16.

⁴⁶ Vgl. Spillner, 2010, S. 25.

⁴⁷ Vgl. ebd., S. 265.

⁴⁸ Vgl. Spillner, 2011, S. 17.

2.3.3 Testrealisierung und Testdurchführung

In dieser Prozessphase werden die Vorbereitungen der Testfälle sowie der notwendigen Testinfrastruktur und der Testumgebung abgeschlossen, um die Tests durchführen zu können. Bei der Testrealisierung wird das fachliche Testfallwissen aus den vorhergehenden Testprozessphasen weiterverwendet. Aufbauend darauf muss bei der Testrealisierung eine genaue Festlegung der technischen Inhalte, also der Testdaten, stattfinden. Neben der Konkretisierung der Testfälle werden diese weiter organisiert und abgestimmt. Beispielsweise wird eine größere Anzahl an Testfällen häufig zu Testszenarien zusammengefasst. Die Testfälle werden dabei in wohlstrukturierten Gruppen organisiert und für die spätere Durchführung, gemäß der Testfallpriorisierung der Testplanung, in eine bestimmte Abarbeitungsreihenfolge gebracht. Durch bestimmte Einschränkungen muss die Priorisierung der Testfälle an dieser Stelle möglicherweise um zusätzliche Betrachtungen erweitert werden. So kann es sein, dass ein Testfall Vorbedingungen besitzt, die es zur Ausführung einzuhalten gilt. Die Einhaltung einer Vorbedingung kann z. B. durch Ausführung eines weiteren Testfalls, der bestimmte Kriterien vorab prüft oder erzeugt, sichergestellt werden. Somit können sich Abhängigkeiten zwischen Testfällen ergeben, die zur korrekten Abarbeitung der Testfälle eine bestimmte Reihenfolge voraussetzen. Diese Betrachtungen können auch auf Ebene der Testszenarien übertragen werden.⁴⁹ Vor Beginn der Testdurchführung sind innerhalb der Testszenarien „*explizit oder implizit gegebene Eingangskriterien zu prüfen oder [...] sicherzustellen*“.⁵⁰ Dies erfolgt dann wiederum durch die Ausführung von entsprechend darauf ausgerichteten Testfällen. Abschließend sollte bei der Organisation von Testfällen und Testszenarien das Ziel nicht außer Acht gelassen werden, die gemäß Teststrategie angestrebten Testziele bestmöglich zu erreichen. Die Menge aller Testszenarien, die darin enthaltenen, geordneten Testfälle und ihre jeweiligen Testdaten werden schließlich im Testausführungsplan festgehalten.⁵¹ Neben der Vorbereitung der Testfälle muss auch die vollständige Verfügbarkeit der Testumgebung sowie der Testwerkzeuge und -mittel sichergestellt werden. Dabei ist die Fehlerfreiheit der Testumgebung (-en) besonders wichtig, um Beeinflussungen auf die Testfalldurchführung durch Fehler in der Testumgebung zu verhindern.⁵²

Sind alle vorbereitenden Maßnahmen abgeschlossen, so kann mit der Durchführung der Tests begonnen werden. Zunächst ist es sinnvoll die Hauptfunktionen zu testen, da dessen Korrektheit die Grundlage für tiefergehende Tests bilden. Es ist ein sorgfältiger Vergleich von Soll- und Ist-Ergebnissen der Testfälle durchzuführen. Tritt während der

⁴⁹ Vgl. Spillner, 2011, S. 18.

⁵⁰ Ebd.

⁵¹ Vgl. ebd.

⁵² Vgl. Ludewig, 2007, S. 460f.

Testausführung eine Abweichung eines Ergebnisses zum Sollergebnis auf, ist dessen Ursache sehr ausführlich zu prüfen.⁵³ Die Ursache eines Fehlers kann „*auch in der fehlerhaften oder ungenauen Testspezifikation, in einer fehlerhaften Testinfrastruktur oder in einem fehlerhaften Testfall oder Testlauf liegen*“.⁵⁴ Die Korrektur eines Fehlers kann außerdem neue Fehler verursachen. Deshalb ist neben der Prüfung der Beseitigung des eigentlichen Fehlers, sicherzustellen, dass dadurch keine weiteren Fehler hinzugekommen sind. In bestimmten Situationen kann dies durch die Erstellung neuer Testfälle überprüft werden.⁵⁵ Sollten sich neue Erkenntnisse bei der Durchführung von Testfällen ergeben, können diese sogar die Spezifikation neuer Testszenarien oder die Verwendung weiterer Testverfahren nach sich ziehen.⁵⁶

In Hinsicht auf die Berichterstattung müssen die Informationen und Ergebnisse der Testrealisierung und -durchführung vollständig dokumentiert werden. Verwendete, im Verlauf entstehende oder ergänzte Dokumente sind beispielsweise die Testfallspezifikation, der Testobjektübergabebericht (Release Dokumentation), das Testprotokoll oder der Testabweichungsbericht. Es gilt mögliche vorausgesetzte Standards bzw. die Vorgaben des Unternehmens zur Dokumentation einzuhalten. Alle soeben genannten Dokumente werden durch den anerkannten und weit verbreiteten IEEE-Standard 829 abgedeckt.⁵⁷

⁵³ Vgl. Spillner, 2010, S. 27f.

⁵⁴ Ebd., S. 28.

⁵⁵ Vgl. ebd., S. 29.

⁵⁶ Vgl. Spillner, 2011, S. 21.

⁵⁷ Vgl. Bath, 2011, S. 25ff.

2.3.4 Testauswertung und Bericht

Die Aktivitäten der Testauswertung und des Testberichtes besitzen eine besondere Bedeutung für das Testmanagement, da sie wichtige Informationen über den Testprozess liefern. Es sind Informationen zusammenzutragen und zu bewerten, die der Überwachung des Testfortschritts, der Berichterstattung als auch der Entscheidung, ab wann ein Test als abgeschlossen gilt, dienen. Hierfür werden viele verschiedene Informationen und auch Metriken benötigt, die durch das Testmanagement zu selektieren oder zu ergänzen sind.⁵⁸ Nach Spillner sind unter anderem folgende wichtige Informationen und Metriken zu berücksichtigen:

„Zu ermitteln sind die Anzahl der geplanten und zu überprüfenden Testbedingungen und die Anzahl der geplanten und ausgeführten Testfälle, wobei zwischen bestandenen und nicht bestandenen Tests zu unterscheiden ist.

Auszuwerten ist die Anzahl aller bei der Testdurchführung aufgedeckten Fehlerwirkungen unter Angabe von Fehlerklasse und Fehlerpriorität, sowohl für die korrigierten als auch für die noch offenen Fehlerwirkungen.

[...]

Der geplante Aufwand ist zu vergleichen mit dem tatsächlichen Aufwand. [...]

Die identifizierten Risiken sind nach der Testausführung dahin gehend zu bewerten, welche Risiken durch den Test abgemildert werden konnten und welche weiterhin uneingeschränkt gelten.“⁵⁹

Anhand der hervorgehobenen Schlüsselwörter können die Aktivitäten und Metriken dieser Prozessphase auf allgemeiner Ebene als ermittelnde, vergleichende und aus- bzw. bewertende Tätigkeiten beschrieben werden.

Diese Phase des Testprozesses wird außerdem von der Zielfrage begleitet, ob das Testende bereits erreicht ist. Dazu müssen die, in der Testplanung definierten, Testendekriterien überprüft und ausgewertet werden. Wurden nach Testdurchführung alle Ende-Kriterien erreicht, so sind die Testaktivitäten abgeschlossen. Weitere Tests sollten durchgeführt werden, wenn ein Kriterium nicht erfüllt wurde. Für den Fall, dass sich Kriterien als nicht erfüllbar herausstellen, sind weitere Tests überflüssig. Die Ursache dafür liegt häufig in ungenau definierten Anforderungen oder Spezifikationen. Darüber hinaus sollten das mit offenen Endkriterien verbundene Risiko und der Testaufwand, die Entscheidung über die Durchführung weiterer Tests rechtfertigen. Bei den zusätzlichen Tests sollte die Konzentration auf der Beobachtung des tatsächlichen

⁵⁸ Vgl. Spillner, 2011, S. 23.

⁵⁹ Ebd., S. 23f.

Testfortschrittes und der Annäherung an das noch offene Testendekriterium liegen, um keine unnötigen Aufwände zu erzeugen. Dabei kann es notwendig sein die Testplanung, aufgrund zusätzlich benötigter Ressourcen, überarbeiten zu müssen. Weitere Hilfsmittel zur Bestimmung des Testendes sind die Betrachtungen von Fehlerüberdeckung und Fehlerfindungsrate. Ist ein bestimmter Prozentsatz des Testobjekts durch Tests abgedeckt worden oder sinkt die Rate (Häufigkeit) der Fehlerfindung im Verlauf der Testaktivitäten unter einen vorgegebenen Zielwert, so können die Testaktivitäten als abgeschlossen betrachtet werden. Bei UE/I werden diese Kriterien im Testkonzept definiert.

Weiterhin ist es wichtig, dass mehrere Korrektur- und Testzyklen (Testdurchläufe) berücksichtigt und eingeplant werden.⁶⁰ Es kann nicht davon ausgegangen werden, „*dass keine Fehlerwirkungen im Test nachgewiesen werden, die zu beheben und in einem weiteren →Testzyklus zu testen sind*“⁶¹ Zum erleichterten Verfassen von Testberichten, gibt der IEEE-Standard 829 nützliche Rahmenbedingungen vor. Ob die Ausgangskriterien eingehalten wurden und das Testende somit erreicht ist, kann der Testmanager anhand dieser Rahmenbedingungen bzw. erfüllten Zahlen überprüfen.⁶²

In diesem Kapitel wurden die fachlichen Einflüsse und Zusammenhänge in Hinsicht auf das Testende betrachtet. In der Praxis ist jedoch Ressourcenknappheit in Form von Zeit- und Kosteneinschränkungen häufig die Ursache für den Abbruch der Testaktivitäten. Dies birgt die Gefahr, dass Fehler, die erst im produktiven Betrieb der Software auftreten, mit viel größerem Kostenaufwand korrigiert werden müssen. Deshalb sollten den Testaktivitäten genügend Ressourcen zur Verfügung gestellt und Pufferreserven eingeplant werden. Zum Abschluss der Aktivitäten ist ein zusammenfassender Testbericht zu erstellen, der eine Gesamtübersicht für die Entscheidungsträger bietet.⁶³

⁶⁰ Vgl. Spillner, 2010, S. 30ff.

⁶¹ Ebd., S. 32.

⁶² Vgl. Spillner, 2011, S. 24.

⁶³ Vgl. Spillner, 2010, S. 32.

2.3.5 Abschluss der Testaktivitäten

Nachdem die Software zur Einführung freigegeben wurde, erfolgt der Abschluss der Testaktivitäten. Oftmals mangelt es dieser Prozessphase an notwendiger Beachtung und zur Verfügung gestelltem Budget.⁶⁴ Unter dem Leitgedanken „*Wenn wir nicht aus unseren Fehlern lernen, ist es wahrscheinlich, dass wir sie im nächsten Projekt wiederholen*“⁶⁵, erlangt diese Prozessphase dennoch ihre Bedeutung. Die während der Testarbeiten gewonnenen Erfahrungen sind zu analysieren und zu sichern, damit auch zukünftige Projekte davon profitieren können. Beispielsweise sind durchgeführte Tätigkeiten, die ursprünglich anders geplant wurden, und die Ursachen für deren Abweichungen im Projektverlauf festzuhalten.⁶⁶ Außerdem müssen die wichtigsten Informationen und Ergebnisse dokumentiert und an die zu benachrichtigenden Personen weitergeleitet werden. Es ist weiterhin zu prüfen, ob tatsächlich alle geplanten Aufgaben abgeschlossen oder begründet ausgelassen worden sind. Letztendlich müssen nicht nur Erfahrungen und Dokumente wie Ergebnisse, Berichte und Protokolle archiviert, sondern auch die für die Testaktivitäten eingesetzten Testmittel, wie Werkzeuge oder Testfälle, gesichert werden. Diese Aufgabe ist wichtig, da nach Einführung der Software weiterhin Fehlerwirkungen auftreten können oder mögliche Änderungen an der Software erneute Tests erfordern. Erst wenn auch die Testmittel archiviert worden sind, kann eine effiziente Wiederaufnahme von Testaktivitäten gewährleistet werden. Der notwendige Aufwand für zukünftige Tests im Rahmen von Wartungsarbeiten kann dadurch deutlich reduziert werden. Da diese Testabschlussaktivitäten von großer Bedeutung sind, sind sie separater Bestandteil des fundamentalen Testprozesses.⁶⁷ Der Testprozess ist somit erst nach Durchführung dieser abschließenden Tätigkeiten als abgeschlossen zu betrachten.

*„Wenn wir dem Abschluss der Testaktivitäten zu wenig Zeit widmen, bedeutet dies auch, dass wir uns weniger Zeit nehmen können, um unsere Prozesse zu verbessern“.*⁶⁸

Diese Aussage kann wie folgt interpretiert werden: Werden die Testaktivitäten nicht gründlich abgeschlossen, so sind Verbesserungen der Prozesse (am System) durch zukünftige Wartungsmaßnahmen mit erhöhtem Zeitaufwand verbunden. Der zusätzliche Aufwand zur Testvorbereitung, der beispielsweise durch nur unvollständig archivierte Testmittel auftritt, hätte eingespart und die Wartungsmaßnahmen schneller durchgeführt werden können.

⁶⁴ Vgl. Bath, 2011, S. 28.

⁶⁵ Ebd.

⁶⁶ Vgl. Spillner, 2010, S. 33.

⁶⁷ Vgl. Spillner, 2011, S. 25ff.

⁶⁸ Bath, 2011, S. 28.

2.4 Optimierung des Testprozesses durch TPI

Mangelnde Prozessqualität ist häufig eine Ursache für das Auftreten verschiedener Arten von Fehlern. Gezielt geplante Prozessänderungen und Optimierungsmaßnahmen zielen darauf ab die Eintrittswahrscheinlichkeit von Fehlern aufgrund dieser Ursache zu senken.⁶⁹ Es gibt verschiedene Verfahren und Vorgehensweisen, um den Testprozess in einem Unternehmen oder Projekt zu verbessern. Im Rahmen dieser Arbeit soll ausschließlich auf das Test Process Improvement (TPI) eingegangen werden, da dieses Modell bei der LTLS in der Abteilung UE/I zur Verbesserung des Testprozesses Anwendung findet. Im Gegensatz zu allgemeingültigen Ansätzen und Modellen zur Verbesserung der Prozesse der Softwareentwicklung, ist TPI speziell und ausschließlich auf die Optimierung des Testprozesses zugeschnitten.⁷⁰

» *Warum wird überhaupt das Ziel verfolgt, den Testprozess zu verbessern?* «

Im Allgemeinen dient die Verbesserung des Testprozesses der „[...] *Optimierung von Qualität, Kosten und Zeitaufwand für den Testprozeß in Bezug zur Bedeutung des gesamten IT-Systems*“.⁷¹ Das bedeutet die Optimierung zielt darauf ab die Qualität des Prozesses zu verbessern sowie entstehende Kosten und Zeitaufwände im Rahmen der Testarbeiten zu senken. Ein optimierter Testprozess verbessert jedoch nicht direkt die Qualität des zu testenden Systems. Er sorgt lediglich dafür, dass die Testergebnisse bzw. die Ergebnisse der Qualitätsmessungen effektiver und effizienter geliefert werden. Zur Erhöhung der Qualität eines Systems müssen anhand der Testergebnisse erst entsprechende Maßnahmen der Entwickler umgesetzt werden. Es besteht somit ein wichtiger Zusammenhang zwischen der Verbesserung des Testprozesses und dem restlichen Entwicklungsprozess (siehe auch Kapitel 2.2). Wird der Entwicklungsprozess nicht im Gesamten weiterentwickelt und optimiert, so ist der Nutzen eines qualitativ optimierten Testprozesses entsprechend geringer.⁷² Die Vorteile eines optimierten und strukturierten Testprozesses sind jedoch weitaus tiefer begründet. Ein strukturierter Ansatz kann beispielsweise breitere Wiederverwendungs- und Einsatzmöglichkeiten, frühe Fehlerfindung und -vermeidung, sowie erhöhte Verständlichkeit und Beherrschbarkeit des Prozesses, bieten.⁷³

Mit Hilfe des TPI-Modells wird der Reifegrad eines Testprozesses bewertet und dessen Stärken und Schwächen, sowie Verbesserungspotential, offengelegt. Dabei erfolgt die Bewertung des Prozesses nach verschiedenen Kriterien, die in bestimmte *Kernbereiche*

⁶⁹ Vgl. Pezzè, 2009, S. 446.

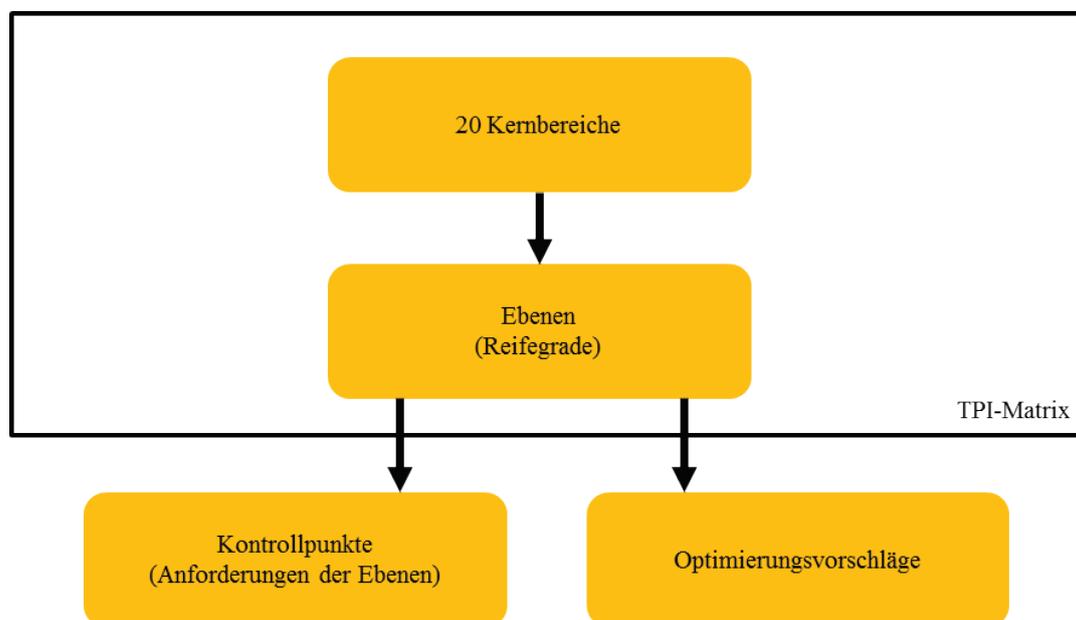
⁷⁰ Vgl. Spillner, 2011, S. 171.

⁷¹ Pol, 2002, S. 23.

⁷² Vgl. ebd., S. 23f.

⁷³ Vgl. Ewijk, 2011, S. 14.

unterteilt sind. Um den Entwicklungsstand des Testprozesses innerhalb der Kernbereiche aufzuzeigen, werden verschiedene *Ebenen* betrachtet, die erreicht werden können. Die Ebenen bauen stufenweise aufeinander auf, wobei durch Erreichung der nächsthöheren Ebene eine Verbesserung in Hinsicht Qualität, Kosten oder Zeitaufwand erreicht wird. Die Verbesserung des Testprozesses erfolgt somit stufen- bzw. schrittweise. Eine Ebene ist mit mindestens einem *Kontrollpunkt* versehen, den bzw. die es jeweils zu erfüllen gilt, damit der Testprozess der entsprechenden Entwicklungsebene innerhalb eines Kernbereiches zugeordnet werden kann. Das TPI-Modell bietet außerdem *Optimierungsvorschläge* zur Erreichung der nächsthöheren Ebene.⁷⁴



Quelle: in Anlehnung an Pol, 2002, S. 32.

Abb. 2.3: Das TPI-Modell

Zwischen vielen Kernbereichen und Ebenen existieren bestimmte Abhängigkeiten und Prioritäten. Der Fortschritt einer Ebene setzt somit häufig voraus, dass sich abhängige Ebenen mitentwickeln. Das allgemeine Ziel von TPI ist es deshalb keine Ebenen auf den unteren Stufen zurückzulassen und eine kontinuierliche Verbesserung in allen Kernbereichen zu erwirken. Die verschiedenen Ebenen der 20 Kernbereiche werden in einer Entwicklungsmatrix zueinander in Beziehung gestellt und einer von *13 Reifegrad bzw. Entwicklungsstufen* zugeordnet. Dabei kennzeichnen die drei Kategorien „beherrschbar“, „effizient“ und „optimierend“ bestimmte Fortschrittsbereiche auf der Entwicklungsskala.⁷⁵

⁷⁴ Vgl. Pol, 2002, S. 31f.

⁷⁵ Vgl. Spillner, 2011, S. 200f.

Folgende Tabelle zeigt die nicht-bewertete TPI-Matrix. Die Ebenen sind durch Buchstaben im Bereich von A bis D gekennzeichnet, wobei Kernbereiche mit zwei, drei oder vier Ebenen existieren.

Kernbereich		Stufe	beherrschbar					effizient					optimierend		
			0	1	2	3	4	5	6	7	8	9	10	11	12
1	Teststrategie		A					B				C		D	
2	Einsatz des Phasenmodells		A			B									
3	Zeitpunkt der Beteiligung			A				B				C		D	
4	Kostenvoranschlag und Planung				A							B			
5	Testspezifikationstechniken		A		B										
6	Statische Testtechniken					A		B							
7	Metriken						A			B			C		D
8	Testautomatisierung				A				B			C			
9	Testumgebung				A				B						C
10	Testarbeitsplatz				A										
11	Engagement und Motivation		A				B						C		
12	Testfunktionen und Ausbildungen				A			B				C			
13	Reichweite der Methodik					A						B			C
14	Kommunikation			A		B							C		
15	Berichterstattung		A			B		C					D		
16	Fehlermanagement		A				B		C						
17	Testwaremanagement			A			B				C				D
18	Testprozessmanagement		A		B								C		
19	Prüfen							A			B				
20	Low-Level-Tests					A		B		C					

Quelle: in Anlehnung an Pol, 2002, S. 44f.

Tab. 2.1: Aufbau der TPI-Matrix

In Kapitel 4.1.3 wird der Reifegrad des Testprozesses von UE/I analysiert. Es wird dort aufgezeigt wie die bewertete TPI-Entwicklungsmatrix für den Testprozess von UE/I aussieht und es werden die Ergebnisse ausgewertet.

2.5 Werkzeugunterstützung

Ein Testwerkzeug ist eine Software, die der Unterstützung und Effizienzsteigerung der Testaktivitäten im Testprozess, sowie der Reduzierung von Informationsverlusten, dient.⁷⁶ Sie erleichtert den Anwendern die Umsetzung und Einhaltung von strukturierten Vorgehensweisen und bildet ein professionelleres Umfeld für die unterstützten Testaktivitäten. Außerdem dient sie dazu den Testaufwand und dadurch anfallende Kosten zu reduzieren. Neben einem angemessenen Kosten-Nutzen-Verhältnis ist bei der Einführung eines Werkzeugs darauf zu achten, dass es nutzerfreundlich, somit leicht erlernbar und ohne großen Schulungsaufwand ist. Die drei letztgenannten Punkte sind von besonderer Bedeutung, um eine angemessene Akzeptanz bei den Anwendern zu erreichen.⁷⁷

Es sei an dieser Stelle darauf hingewiesen, dass die Begriffe „Werkzeug“ und „System“ in der gesamten Arbeit synonym verwendet werden. Es gibt viele verschiedene Arten von Testwerkzeugen mit unterschiedlichen Einsatzzwecken. Testmanagementsysteme bzw. *„Testmanagementwerkzeuge unterstützen uns beim Management des Testprozesses und des gesamten Projekts“*.⁷⁸

In Hinsicht auf Funktionsumfang, Konzept und Ausrichtung der Werkzeuge gibt es unterschiedliche Ansätze der Werkzeughersteller. Deshalb muss zwischen funktionsübergreifenden Systemen und einer Kombination aus mehreren kleineren Werkzeugen, mit jeweils abgegrenzter, aber dafür möglicherweise umfassenderer Funktionsausrichtung, unterschieden werden. Die nachfolgend genannten typischen Aufgabenfelder eines Testmanagementsystems können deshalb sowohl von einem System als auch durch Verknüpfung mehrerer Werkzeuge abgedeckt werden.

Die Kernaufgabe eines Testmanagementsystems ist es, den Testmanager bei der *Testplanung und Steuerung* und dem Management von Testfällen zu unterstützen. Die Anzahl an festgehaltenen Testfällen kann schnell sehr groß und unübersichtlich werden. Deshalb können die Testfälle zum einen erfasst und priorisiert, als auch mit Hilfe entsprechender Sortier- und Gruppierungsfunktionen verwaltet werden. Anhand verschiedener Indikatoren zur Testfallausführung kann außerdem ein Überblick und Kontrolle über den Status der Testfälle bewahrt werden. Einige Testmanagementwerkzeuge besitzen unterstützende Funktionen für die Berichterstattung, indem sie relevante Informationen sammeln, auswerten und in Berichtsform bringen. Je nach Ausrichtung und Umfang dieser Funktionen können selbst vollständige Testdokumentationen erzeugt werden. Zusätzlich können Testmanagementwerkzeuge

⁷⁶ Vgl. Bath, 2011, S. 323.

⁷⁷ Vgl. Jackson, 2009, S. 38.

⁷⁸ Bath, 2011, S. 325.

auch Tätigkeiten auf Projektmanagementebene, wie die Planung notwendiger Ressourcen oder Testzeiträume, unterstützen.⁷⁹

Weitere wichtige Beziehungen zum Testmanagement bilden die Aufgabenbereiche:

- Erfassung von Anforderungen (Anforderungsmanagement)
- Verwaltung von Fehler- und Problemmeldungen bezüglich der durchgeführten Tests (*Fehlermanagement / Abweichungsmanagement*)
- Konfigurations- und Versionsverwaltung (*Konfigurationsmanagement*)

Um alle diese Aufgabenfelder vollständig abdecken zu können, ist meist die Integration mehrerer Werkzeuge notwendig, die sich oft nur auf jeweils einen oder wenige dieser Aufgabenbereiche beschränken. In diesem Fall nimmt häufig das Testmanagementsystem die zentrale Position zur Kommunikation mit den angebotenen Werkzeugen ein. Wird beispielsweise ein separates Werkzeug zum Fehlermanagement verwendet, so besitzt es sinnvollerweise eine direkte Anbindung zum Testmanagementsystem, um im Testprozess aufgedeckte Fehlerwirkungen zur Nachverfolgung und Fehlerbehebung an das Fehlermanagementsystem weiterzugeben. Eventuell bietet das Testmanagementsystem die Funktion zur Erfassung und Verwaltung von Anforderungen, was es zu einem funktionsübergreifendem Werkzeug macht. Darüber hinaus bieten viele Testmanagementwerkzeuge eine Anbindung an Testausführungswerkzeuge zur automatischen Übermittlung der Testergebnisse.⁸⁰

Sind die Funktionen aller genannten Bereiche gegeben, ist es möglich „den Teststatus von den Anforderungen über die Testfälle und Testergebnisse bis zu den Fehlermeldungen und Codeänderungen lückenlos zu verfolgen und nachzuvollziehen“.⁸¹

Dieser Sachverhalt soll durch ein kurzes Beispiel verdeutlicht werden. Ein Testmanagementsystem bietet typische Testmanagementfunktionen, wie Verwaltung der Testfälle und Erfassung der Testergebnisse. Zusätzlich bietet es Funktionen des Anforderungsmanagements sowie des Fehlermanagements und Konfigurationsmanagements. Es ist somit in der Lage alle wichtigen Bereiche des Testmanagements abzubilden und eine transparente Sicht auf den Testprozess zu bieten. Dabei spielt es keine Rolle, ob die Gesamtmenge der Funktionen durch Integration des Testmanagementsystems mit weiteren Werkzeugen oder durch ein funktionsübergreifendes Testmanagementwerkzeug abgebildet wird.

⁷⁹ Vgl. Spillner, 2011, S. 344f.

⁸⁰ Vgl. Spillner, 2010, S. 210f.

⁸¹ Ebd., S. 211.

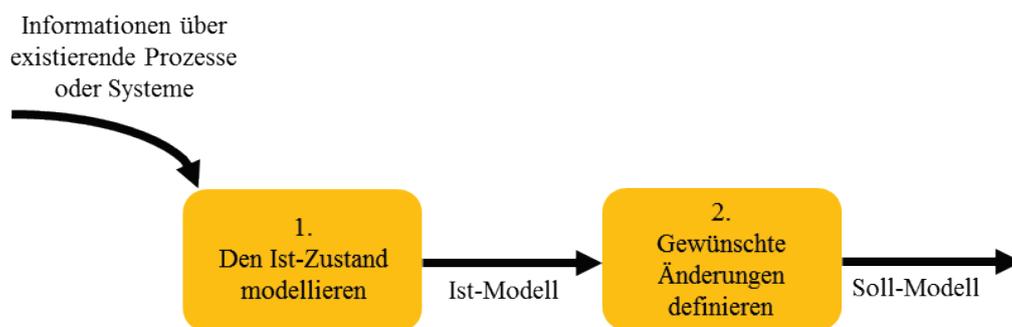
3 Material und Methoden

Die in Kapitel 4 verwendeten Materialien, Methoden und Verfahren, die zum zielorientierten Ermitteln der Ergebnisse eingesetzt wurden, werden in diesem Abschnitt zusammengefasst und allgemein beschrieben.

3.1 Strukturierte Systemanalyse

Um verschiedene Testmanagementwerkzeuge auf ihre Eignung für den Einsatz bei UE/I untersuchen und evaluieren zu können, mussten gewisse Untersuchungsgrundlagen geschaffen werden.

Eine bewährte und allgemein anerkannte Vorgehensweise ist die strukturierte Systemanalyse von DeMarco. Abbildung 3.1 zeigt dieses Vorgehen schemenhaft.



Quelle: in Anlehnung an DeMarco, 1978 (nach Pohl, 2008, S. 26)

Abb. 3.1: Erzeugung von Ist- und Soll-Modell

Nach DeMarco werden Informationen über existierende Prozesse oder Systeme genutzt, um den Istzustand zu ermitteln. Daraufhin wird das Soll-Modell über gewünschte Änderungen am Ist-Modell definiert. In den Betrachtungen dieser Arbeit geht es jedoch weniger darum gewünschte Änderungen zu definieren. Die Konzentration liegt eher auf dem Ziel zu ermitteln, wie gut die Testmanagementwerkzeuge den bereits bestehenden Prozess unterstützen und abbilden können. Dies geschieht auf Grundlage der ermittelten Prozesse, Modelle und Informationen zum Istzustand.

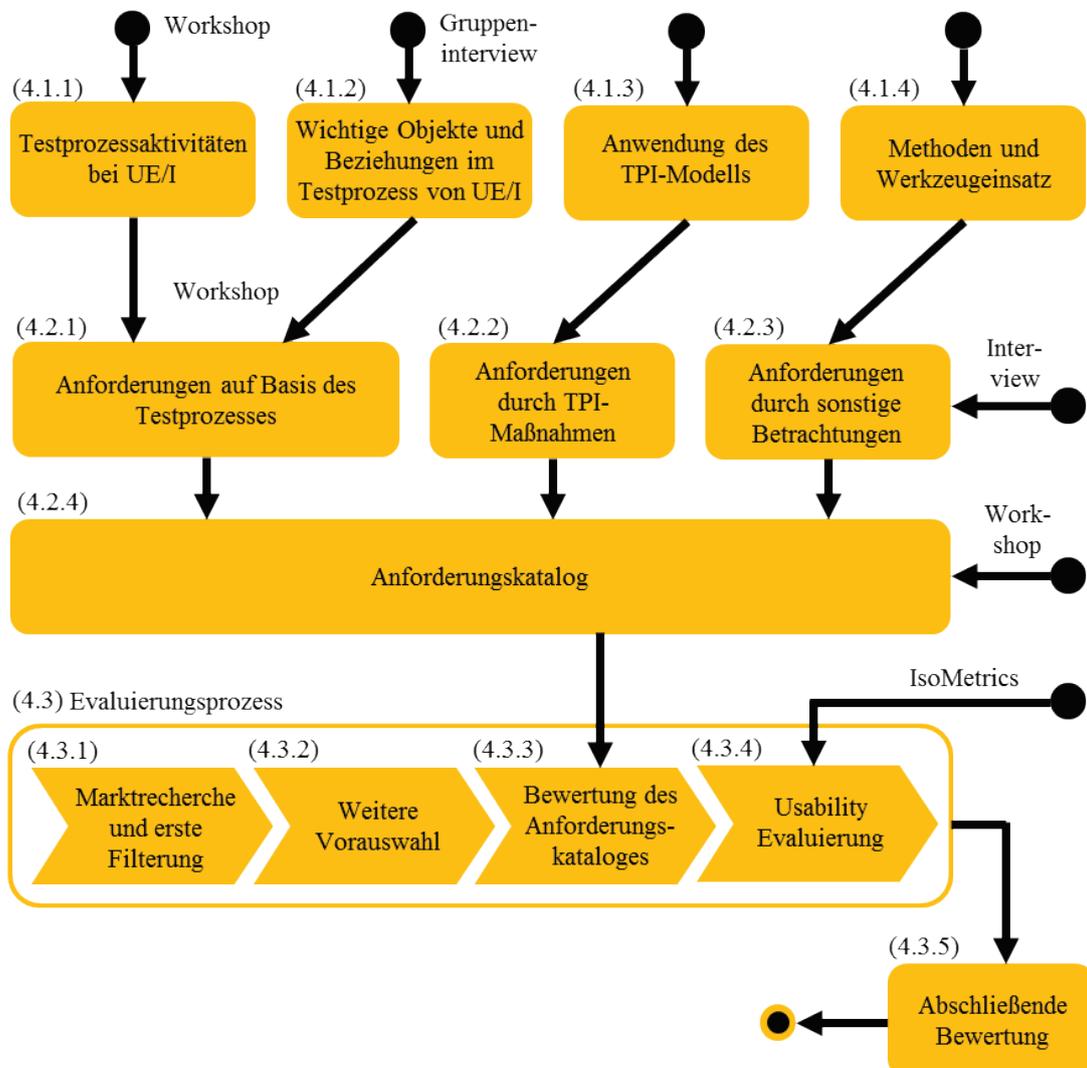
Der eigentliche Sollzustand wird folglich, durch die möglichst umfangreiche Abbildung des Istzustandes, durch ein Testmanagementsystem gekennzeichnet. Dies schließt jedoch nicht aus, dass im Verlauf der Betrachtungen gewünschte Zustände und Anforderungen für den Sollzustand formuliert wurden, die im Ist-Modell nicht existent sind.

Zur Schaffung der Untersuchungsbasis wurden somit zwei grobe Schritte durchgeführt:

1. Analyse der Ist-Situation im Bereich Testmanagement (siehe Kapitel 4.1)
2. Ableitung von Anforderungen und der Soll-Situation auf Grundlage der Ergebnisse der Ist-Analyse (siehe Kapitel 4.2)

3.2 Der Untersuchungsprozess

Das nachfolgende Prozessschaubild gibt eine Übersicht über das in Kapitel 4 angewendete analytische Vorgehen. Es zeigt den Untersuchungsablauf sowie die dabei angewendeten Methoden und Verfahren.



Quelle: Eigene Darstellung

Abb. 3.2: Der Untersuchungsprozess

Das Prozessschaubild stellt eine Art Fahrplan der durchgeführten analytischen Betrachtungen dar, wobei jedes Unterkapitel aus Kapitel 4 in den Prozessfluss einbezogen und dargestellt wird. Es soll dabei helfen den analytischen Ablauf lückenlos nachvollziehen zu können.

Die Inhalte der Unterkapitel 4.1.x und 4.2.x sind Bestandteil der strukturierten Systemanalyse (siehe Kapitel 3.1) und bilden die Bewertungsgrundlage für den Evaluierungsprozess (siehe Kapitel 4.3).

Eine Evaluierung (Synonym: Evaluation) ist im Allgemeinen „*eine Bewertung oder Beurteilung eines Sachverhalts oder Objektes auf der Basis von Informationen*“⁸². Sie erfolgt zielgerichtet in den drei Schritten: *Informationssammlung* und *Bewertung der Informationen* als Grundlage für das *Treffen einer Entscheidung*.⁸³

Diese drei Schritte spiegeln sich folgendermaßen im Prozessschaubild wider. Die Informationssammlung erfolgt sowohl durch die strukturierte Systemanalyse, als auch durch die Marktrecherche (1.). Auf Grundlage der gesammelten Informationen wird die Bewertung der Werkzeuge durchgeführt (2.), dessen Ergebnisse daraufhin als Entscheidungsgrundlage (3.) dienen.

Aus dem Prozessschaubild ist erkennbar, dass zwei wesentliche Methoden – Interviews und Workshops – zur Informationsgewinnung angewendet wurden. Zur Usability-Evaluierung wurde eine weitere Methode – das IsoMetrics-Verfahren – angewendet. Sie wird in Kapitel 3.3 separat beschrieben. Die Methoden Interview und Workshop werden im Folgenden auf allgemeiner Ebene dargestellt. Die Beschreibungen zu den detaillierten Vorgehensweisen finden sich an den entsprechenden Stellen des Methodeneinsatzes in Kapitel 4.1 und 4.2 wieder.

Ein Interview ist eine Befragung, wobei diese Arbeit nur mündlich durchgeführte Experteninterviews beinhaltet. Ein Experte ist eine Person, „*die aufgrund langjähriger Erfahrung über bereichsspezifisches Wissen (...) [oder] Können verfügt*“.⁸⁴ In Experteninterviews werden die Experten deshalb zu Ihrem Wissen befragt.

Für die Untersuchungen wurden verschiedene Arten von Interviews durchgeführt, wobei nach der Anzahl der am Interview beteiligten Personen und der Art der Durchführung des Interviews unterschieden werden muss. Wird nur eine Person befragt, ist die Rede von einem *Einzelinterview*, dessen Resultat die alleinige Meinung eines Befragten ist. Wird eine Gruppe von Personen befragt, so handelt es sich um ein *Gruppeninterview*, dessen Ergebnis die gemeinsamen Antworten der Teilnehmer und

⁸² Stockmann, 2007, S.25.

⁸³ Ebd., S.25f.

⁸⁴ Mieg, 2005, S.7.

die Meinung der Gruppe umfasst. Die Besonderheit beim Gruppeninterview besteht darin, dass sich die Teilnehmer während des Gespräches gegenseitig beeinflussen. Dies kann positive, als auch negative Einflüsse mit sich bringen (siehe Kapitel 4.1.2). Ebenso kann zwischen einem explorativen und einem standardisierten Interview unterscheiden werden. Bei beiden Interviewarten werden Fragen vorbereitet. Ein exploratives Interview ist im Gegensatz zum standardisierten Interview offener bezüglich des Ablaufes des Interviews. Das heißt es kann von den Fragen abgewichen werden, um die gewünschten Informationen, beispielsweise durch Hinterfragung, näher zu erkunden. Beim standardisierten Interview weicht der Interviewer hingegen nicht von den vorbereiteten Fragen ab.⁸⁵ Für die Untersuchungen dieser Arbeit wurden ein standardisiertes Gruppeninterview und ein exploratives Einzelinterview durchgeführt. Weshalb genau diese Interviewarten gewählt wurden, wird an den entsprechenden Stellen der Methodenverwendung vertieft (siehe Kapitel 4.1.2 und 4.2.3).

Ein Workshop wird aus Sicht der Unternehmen folgendermaßen definiert: Workshops sollen „*Werkstätten des Schaffens und des Lernens sein, in denen zu gewissen vordefinierten Thematiken Lösungen erarbeitet werden*“⁸⁶. In den Untersuchungen dieser Arbeit dienten die Workshops primär dem Schaffen von zusammenhängenden Informationen. Ein Workshop hat einen festen Ablaufplan dessen Einhaltung durch den Workshop-Leiter bzw. Moderator geregelt wird.⁸⁷ Drei Aktivitäten wurden bei der Durchführung eines Workshops in jedem Fall durchgeführt. Der Workshop wurde frühzeitig organisiert, um sicher zu gehen, dass alle benötigten Personen teilnehmen können. Weiterhin wurde der Ablaufplan vorab an die Teilnehmer kommuniziert, damit sie sich ein Bild vom anstehenden Workshop machen oder sich bereits im Vorfeld vorbereiten konnten. Im Anschluss und während des Workshops wurden die Ergebnisse sorgfältig dokumentiert. Der Vorteil eines Workshops liegt in der direkten Kommunikation der Teilnehmer und dem dadurch gesteigerten gegenseitigen Verständnis. Gezielt gesteuerte Diskussionen ermöglichen es, im Team abgestimmte Informationen zu erhalten.⁸⁸

⁸⁵ Vgl. Pohl, 2008, S.325.

⁸⁶ Ruedel, 2008, S. 7.

⁸⁷ Vgl. Rupp, 2009, S. 103.

⁸⁸ Vgl. ebd.

3.3 Das IsoMetrics-Verfahren

IsoMetrics ist ein von Gediga, Hamborg und Willumeit entwickelter Frage- und Bewertungsbogen zur Evaluation der Usability von Software bzw. Softwaresystemen.⁸⁹

Das Wort „Usability“ kommt aus dem Englischen und kann mit „Gebrauchstauglichkeit“ übersetzt werden. Neben der Benutzerfreundlichkeit, also der angenehmen Nutzung eines Systems, konzentriert sich Usability auch auf die Unterstützung der Erreichung der Ziele des Nutzers. Sie ist folglich ein Indikator für die Qualität eines Softwaresystems oder allgemein eines technischen Systems.⁹⁰ Der IsoMetrics Fragebogen enthält somit Bewertungspunkte zur Beurteilung der Benutzerfreundlichkeit, als auch zur Erreichung der Ziele des Anwenders.

IsoMetrics basiert auf verschiedenen Bewertungspunkten unterschiedlicher anerkannter Fragebögen, wie QUIS, EVADIS, ISONORM 9241 / 10 oder DIN EN ISO 9241-10 und -11. In der aktuellen Version von IsoMetrics (Version 2.04, Februar 2002) werden 75 Items untersucht. Alle Bewertungspunkte bzw. -elemente (engl. *Items*) sind sieben Bereichen zugeordnet, den Prinzipien des internationalen Standards DIN EN ISO 9241-110.⁹¹ Die sieben Prinzipien bzw. Gestaltungsgrundsätze kennzeichnen die Bereiche *Aufgabenangemessenheit*, *Selbstbeschreibungsfähigkeit*, *Steuerbarkeit*, *Erwartungskonformität*, *Fehlerrobustheit*, *Individualisierbarkeit* und *Erlernbarkeit*.⁹²

Der IsoMetrics Fragebogen steht sowohl in einer langen, als auch in einer kurzen Version zur Verfügung. Beide Versionen enthalten dieselbe fünfstufige Bewertungsskala von „stimmt nicht“ bis „stimmt sehr“ („keine Angabe“ ist auch möglich) und beinhalten die gleichen Items. Der Unterschied zwischen beiden Versionen besteht lediglich darin, dass in der langen Version zusätzlich bei jedem Item nach dessen Wichtigkeit für den Gesamteindruck des Systems, sowie nach einem begründenden Beispiel, gefragt wird.⁹³

Für beurteilende und vergleichende Untersuchungen bereits fertiger Softwaresysteme ist die Anwendung der Kurzform des IsoMetrics-Verfahrens geeignet.⁹⁴ Zur Usability-Evaluierung wurde deshalb die Kurzversion von IsoMetrics in der aktuellen Version 2.04 verwendet. Die genaue Anwendung des Fragebogens wird in Kapitel 4.3.4 beschrieben. Die Verwendung der IsoMetrics Methode ist für die kommerzielle Nutzung kostenpflichtig – für die Nutzung in wissenschaftlichen Projekten und Arbeiten ist sie, unter Einholung einer Erlaubnis der Entwickler, dagegen kostenfrei.⁹⁵

⁸⁹ Vgl. Sarodnick, 2011, S. 188f.

⁹⁰ Vgl. ebd., S. 19.

⁹¹ Vgl. ebd., S. 189.

⁹² Vgl. Gediga, 2000, S. 3.

⁹³ Vgl. Sarodnick, 2011, S. 188f.

⁹⁴ Vgl. Gediga, 2000, S. 6f.

⁹⁵ Vgl. <http://www.isometrics.uni-osnabrueck.de> (letzter Zugriff 30.07.2012).

4 Ergebnisse und Diskussion der analytischen Betrachtungen

4.1 Analyse der Ist-Situation im Bereich Testmanagement

In diesem Kapitel werden verschiedene Bereiche innerhalb des Testmanagementumfeldes analysiert, die den Istzustand bei UE/I charakterisieren. Es wird näher auf die einzelnen Aktivitäten im Testprozess, Beziehungen und Abhängigkeiten wichtiger Objekte im Testprozess, den momentanen Fortschritt durch TPI-Maßnahmen und die im Testumfeld verwendeten Methoden und Werkzeuge eingegangen. Die Durchführung aller Analysen erfolgte unter der Zielsetzung, dass aus den gewonnenen Ergebnissen Anforderungen an ein Testmanagementwerkzeug abgeleitet werden können.

4.1.1 Testprozessaktivitäten bei UE/I

In Kapitel 2.3 wurde der fundamentale Testprozess, wie er nach ISTQB-Standard definiert ist, beschrieben. UE/I orientiert sich zwar an diesen theoretischen Grundlagen, um jedoch eine Vorstellung davon zu bekommen wie genau die Testprozessphasen bei UE/I gelebt und welche Testaktivitäten tatsächlich durchgeführt werden, musste der Testprozess bei UE/I analysiert werden. Der Testmanager hat üblicherweise den umfassendsten Überblick über die einzelnen Testprozessphasen und die durchzuführenden Aktivitäten im Testprozess. Deshalb wurde ein Workshop mit dem Testmanager von UE/I durchgeführt, in dem die einzelnen Testprozessaktivitäten jeder Prozessphase identifiziert wurden. Aufgrund der Erfahrung und des präzisen Wissens in den Bereichen der Testvorbereitung und der Testdurchführung, wurde zusätzlich ein Tester von UE/I zum Workshop eingeladen. Als unterstützendes Dokument, das im Workshop begleitend benutzt wurde, diente das Testhandbuch von UE/I. Das von UE/I verwendete Testhandbuch ist ein sehr komplexes und umfangreiches Dokument, das beim Lesen entsprechendes Hintergrundwissen voraussetzt. Im Testhandbuch sind unter anderem die üblicherweise durchzuführenden Aktivitäten je Testprozessphase und verwendete Objekte bzw. Dokumente, beschrieben. Insbesondere um die Zusammenhänge zwischen den Aktivitäten korrekt abbilden zu können, wurde die Entscheidung für einen gemeinsamen Workshop mit dem Testmanager und einem Tester getroffen, anstatt das Testhandbuch als alleinige Grundlage für die Analyse zu verwenden. Nach Durchführung des Workshops wurden die protokollierten Aktivitäten der jeweiligen Testphasen, in der entsprechenden Reihenfolge, in Aktivitätsdiagrammen zusammengefasst und veranschaulicht. Diese Aktivitätsdiagramme verdeutlichen den Prozessfluss der Aktivitäten (engl. Workflow) und werden durch die Integration wichtiger Objektbeziehungen (z. B. zu Dokumenten) erweitert. Sie bilden somit eine

Art Szenario ab, ohne dabei jedoch zwischen den einzelnen Akteuren und Rollen, die verschiedene Aktivitäten im Testprozess ausführen, zu unterscheiden.

Es wurde davon ausgegangen, dass nicht alle erfassten Aktivitäten für die spätere Anforderungsermittlung benötigt werden. Dennoch wurde an dieser Stelle auf eine möglichst tiefgründige Prozessanalyse Wert gelegt, um die notwendigen Kontextinformationen und ein angemessenes Verständnis über den Testprozess des Unternehmens zu schaffen.

Im Folgenden werden alle Testprozessphasen und Aktivitäten bei UE/I beschrieben und dargestellt. Die Aktivitäten der Teststeuerung wurden getrennt betrachtet, da sich diese über den gesamten Testprozess erstrecken. Außerdem wurde bei den Aktivitäten der Testplanung zwischen der *initialen Testplanung* zu Beginn eines Projektes und der *Testplanung auf Basis von Zwischenreleases* unterschieden. Der Grund dafür ist, dass die Releases bei UE/I in mehreren Zwischenreleases für den Test bereitgestellt werden und die Testplanung auf Ebene der Zwischenreleases erfolgt. Insgesamt sind sieben Aktivitätsdiagramme für die Aktivitäten der Bereiche *Initiale Testplanung*, *Teststeuerung*, *Testanalyse und Design*, *Testplanung auf Basis von Zwischenreleases*, *Testrealisierung und Durchführung*, *Testauswertung und Bericht* sowie *Abschluss der Testaktivitäten* entstanden. Die Aktivitätsdiagramme sind dem Anhang beigelegt (siehe Abb. A.1 bis Abb. A.7). Im nachfolgenden Text werden alle sieben Bereiche der Aktivitätsdiagramme beschrieben. Die einzelnen Aktivitäten wurden für jede Prozessphase entsprechend ihrer Reihenfolge in den Aktivitätsdiagrammen nummeriert. Verwendete Buchstaben referenzieren auf die in den Aktivitätsdiagrammen verwendeten Objekte.

Initiale Testplanung:

Das umfangreiche Testhandbuch (A) von UE/I bildet die Grundlage für die Erstellung des Testkonzepts (B). Die meisten Ausgangsinformationen des Testkonzepts basieren auf dem Inhalt des Testhandbuches. Da Testkonzepte für ein Projekt angefertigt werden, kann das Testkonzept die Gültigkeit der Annahmen aus dem Testhandbuch für ein Projekt überschreiben. Das Testkonzept umfasst außerdem den Testelementekatalog (D) und den Masterplan (E), die aufgrund ihrer Abgeschlossenheit auch separat betrachtet werden können.

1. Die Projektrahmenbedingungen werden identifiziert und im Testkonzept (B) festgehalten. Dazu gehören beispielsweise Informationen über den Testauftrag oder festgelegte Meilensteine.

2. Es werden die Testaufgaben definiert, die ebenfalls im Testkonzept (B) erfasst werden.
3. Jegliche Informationen aus denen Anforderungen ersichtlich sind, wie beispielsweise Fachkonzepte und Spezifikationen, werden gesammelt oder referenziert, um die Testbasis (C) zu identifizieren.
4. Ist die Testbasis identifiziert, so wird deren Qualität untersucht. Dies geschieht durch Reviews der Anforderungsdokumente, wobei die Ergebnisse und eventuell neu gewonnene Erkenntnisse in die Testbasis (C) einfließen. Die Vorgehensweisen zur Qualitätssicherung der Testbasis sind im Testkonzept (B) beschrieben.
5. Die durchzuführenden Teststufen und die Testobjekte je Teststufe werden auf Grundlage der Testbasis (C) und des Testkonzeptes (B) identifiziert. Das Ergebnis fließt in den Katalog (D) ein, der die zu testenden Testelemente je Testobjekt enthält.
6. Auf Grundlage der im Testkonzept (B) definierten Qualitätsmerkmale werden die zu testenden Testobjekte bewertet. Dazu werden auch Betrachtungen der Testbasis (C) vorgenommen und die bewerteten Testobjekte im Testelementekatalog (D) aufgenommen.
7. Durch erneutes Hinzuziehen der Testbasis (C) werden die Testelemente, die es für das entsprechende Testobjekt zu testen gilt, identifiziert. Die identifizierten Testelemente je Testobjekt werden im Testelementekatalog (D) festgehalten.
8. Für jedes Testelement aus dem Testelementekatalog (D) wird eine Risikobewertung mittels der im Testkonzept (B) definierten Methoden vorgenommen. Der Testelementekatalog (D) wird durch das Ergebnis erweitert.
9. Anhand der im Testkonzept (B) beschriebenen Methoden wird für jedes Testelement des Testelementekataloges (D) eine Auswahl der Qualitätsmerkmale getroffen, auf die beim Test des jeweiligen Testelementes Wert gelegt wird. Das Ergebnis wird wiederum im Testelementekatalog (D) festgehalten.
10. Auf Basis der Risikostufe eines Testelementes (Vgl. 8.) und der Gewichtung der Qualitätsmerkmale des Testobjekts (Vgl. 6.) wird die Testintensität entsprechend abgeleitet. Als Ergebnis ist jedem Testelement (aus (D)) eine definierte Testintensität je Qualitätsmerkmal zugeordnet. Die anzuwendenden Methoden sind dem Testkonzept (B) zu entnehmen und das Ergebnis wird im Testelementekatalog (D) festgehalten.

11. Auf Grundlage der im Testkonzept (B) definierten Methoden wird eine Aufwandsschätzung vorgenommen, um eine Aussage über den Ressourcenbedarf zur Durchführung von Tests für ein Testobjekt (Testelemente je Testobjekt in (D) als Basis) bzw. eine Teststufe zu ermöglichen. Darauf basierend kann eine Abschätzung der Testdurchführungskosten erfolgen. Das Ergebnis wird im Mastertestplan (E) festgehalten
12. Auf Grundlage der im Testkonzept (B) beschriebenen Anforderungen an einen Mastertestplan (E), wird eine zeitliche Grobplanung des Testprojektes vorgenommen und das Resultat im Mastertestplan (E) erfasst.
13. Die für das Testprojekt notwendigen Testressourcen werden ermittelt und im Mastertestplan (E) festgehalten.

Teststeuerung:

1. Es erfolgt ein ständiger Abgleich der Sollvorgaben der Testplanung mit dem tatsächlichen Projektverlauf (Istzustand). Grundlage für diesen Abgleich bilden die Berichterstattungen der Mitarbeiter (B) und die Planungsdaten und Informationen aus der Testplanungsphase (A). Diese Aktivität wird kontinuierlich durchgeführt und wiederholt.
2. Tritt eine Abweichung zwischen Soll- und Istzuständen auf, so muss die Testplanung (A) gegebenenfalls angepasst werden.

Testanalyse und Design:

Wie bereits bei der *initialen Testplanung* erwähnt, basiert das Testkonzept (B) auf dem Testhandbuch (A).

1. Die in der Testplanung identifizierten Anforderungen aus der Testbasis (C) werden analysiert.
2. Zu den Testelementen aus dem Testelementekatalog (D) werden Testbedingungen, wie Testabdeckungskriterien oder Abhängigkeiten, festgelegt. Die Ergebnisse werden wiederum im Testelementekatalog (D) aufgenommen. Im Testkonzept (B) sind dabei die Verfahren zur Ermittlung der Testbedingungen beschrieben.
3. In Abhängigkeit der in der Testplanung ermittelten Testintensität je Testelement des Testelementekataloges (D) werden die Entwurfsziele und anzuwendende Entwurfstechniken definiert. Die Grundlagen dafür werden dem Testkonzept (B) entnommen. Der Testelementekatalog (D) wird durch die Ergebnisse erweitert.

Testplanung auf Basis von Zwischenreleases:

Bei der Bereitstellung eines Zwischenreleases wird vom Entwickler eine Dokumentation zu diesem Release (A) zur Verfügung gestellt.

1. Wurde ein Zwischenrelease zum Testen bereitgestellt, so wird ein Testplan (C) für das Zwischenrelease angelegt.
2. Mit Hilfe der Releasedokumentation (A) wird ermittelt, welche Anforderungen bereits im Zwischenrelease umgesetzt wurden. Zu diesen umgesetzten Anforderungen werden aus dem Testfallpool (B) die Testfälle ausgewählt, die diese Realisierungen testen. Die ausgewählten Testfälle werden im Testplan (C) aufgenommen.
3. Auf Basis der Releasedokumentation (A) werden ebenso Testfälle zu erforderlichen Re-Tests aus dem Testfallpool (B) identifiziert und im Testplan (C) festgehalten.

Für einen Re-Test werden dabei Testfälle ausgewählt, in denen beim vorigen Testlauf Fehler aufgetreten bzw. Abweichungen erkannt worden sind.

4. Die Releasedokumentation (A) dient außerdem als Grundlage für die Identifizierung von Testfällen für Regressionstests. Aus dem Testfallpool (B) ausgewählte Testfälle werden in den Testplan (C) aufgenommen.
5. Abschließend wird eine Planung für die im Testplan (A) vorgesehenen Tests durchgeführt. Es werden dabei Priorisierungen der Tests vorgenommen und sämtliche notwendige Ressourcen, wie Mitarbeiter oder Testinfrastruktur, zugewiesen. Diese Planungsdaten werden im Testplan (A) hinterlegt.

Testrealisierung und Durchführung:

1. Unter Berücksichtigung des Testplans (A) werden die Tests in der vorgegebenen Reihenfolge abgearbeitet. Zur konkreteren Beschreibung der Aktivitäten wird im Folgenden von einem einzelnen Testfall ausgegangen, der abgearbeitet werden soll.
2. Der vorerst logische Testfall wird mit Testdaten angereichert und stellt einen konkreten (ausführbaren) Testfall dar. Die Testdaten für den Testfall können dabei bereits im Testplan (A) hinterlegt worden sein.
3. Die Abarbeitung des Testfalls wird mit entsprechenden Vorgehensweisen durchgeführt. Auch die Durchführung mehrerer logisch zusammenhängender Testfälle ist möglich.

4. Das Ergebnis der Testfallausführung wird im Testprotokoll (B) festgehalten.
5. Sind Abweichungen erkannt worden, so muss im genutzten Fehlermanagementsystem – *JIRA*⁹⁶ (C) ein Fehlereintrag angelegt werden.

Solange bis alle Testfälle des Testplans (A) abgearbeitet sind, werden die Aktivitäten für jeden Testfall erneut durchlaufen.

Testauswertung und Bericht:

1. Die Testprotokolle (B) mit den Testergebnissen aus der Testdurchführung werden analysiert.
2. Die analysierten Testprotokolle (B) werden mit dem Testplan (D) (Ursprungsplanung) abgeglichen und der Testfortschritt ermittelt. Das Ergebnis wird im Testbericht (A) festgehalten.
3. Die im Abweichungsbericht (C) dokumentierten Abweichungen werden analysiert.
4. Nach Betrachtung der Abweichungen aus dem Abweichungsbericht (C) wird durch Hinzuziehen der Sollvorgaben aus dem Testplan (D) eine Abweichungsentwicklung ermittelt. Das Resultat wird in einem Testbericht (A) festgehalten.
5. Auf Grundlage der Aktivitäten eins bis vier und einer Prüfung des Testplans (D) kann eine qualitative Beurteilung erstellt werden. Das Ergebnis fließt wiederum in einen Testbericht (A) ein.

Das Objekt Testbericht (A) wird an dieser Stelle als Oberbegriff für verschiedene Arten von Testberichten, wie dem Testfortschrittsbericht oder dem Testabschlussbericht, verwendet. Abhängig davon, in welcher Phase des Testprozesses die Testberichte erstellt werden, haben diese eine spezifische Bezeichnung. Im Kontext der analysierten Testprozessphase „Testauswertung und Bericht“ kann nicht darauf geschlossen werden, für welche Phase des Testprozesses – „Testanalyse und Design“ oder „Testrealisierung und Durchführung“ – ein Testbericht erstellt wird (vgl. Kapitel 2.3, Abb. 2.2). Es wurde an dieser Stelle daher der allgemeine Ausdruck des Testberichtes verwendet.

⁹⁶ Siehe Herstellerlink für weitere Informationen (<http://www.atlassian.com/software/jira>).

Abschluss der Testaktivitäten:

1. Es finden letzte Überprüfungen in Hinsicht auf die Begründbarkeit und Richtigkeit des Testendes statt.
2. Wichtige Ergebnisse und Erfahrungen, die während der im Testprozess durchgeführten Aktivitäten gesammelt werden konnten, müssen identifiziert und analysiert werden. Um Optimierungspotentiale von Aktivitäten aufzudecken, erfolgt hauptsächlich eine Analyse der Aktivitäten der Testphasen „Testplanung und Steuerung“ und „Testanalyse und Design“ (siehe Kapitel 2.3, Abb. 2.2).
3. Die in Aktivität zwei erhaltenen Ergebnisse müssen im Testbewertungsbericht (A) dokumentiert werden. Der Testbewertungsbericht (A) enthält üblicherweise eine Zusammenfassung der Ergebnisse, einen Erfahrungsbericht und eine Bewertung des Testprozesses.
4. Die im Testprozess generierten Dokumente und die verwendeten Testmittel müssen für den Fall der späteren Wiederverwendung konserviert werden.

In Kapitel 2.2 wurde bereits erwähnt, dass sich in der Praxis viele Testaktivitäten überschneiden oder parallel ausgeführt werden können bzw. nicht klar voneinander trennbar sind. Dies ist auch bei UE/I der Fall. Obgleich die Ergebnisse des Workshops eine klare Trennung der Aktivitäten und Testprozessphasen aufzeigen, handelt es sich in Wirklichkeit um eine Vermischung aus theoretischem Soll-Prozess und Ist-Prozess. Bei der praktischen Durchführung treten eben genau diese Fälle der Überschneidung und zeitgleicher Durchführung von Aktivitäten auf. Bei der Testdurchführung ist es beispielsweise möglich, dass während der Ausführung eines Testfalls, bereits parallel weitere Testfälle vorbereitet und mit Testdaten versehen werden. Dieses praxisnahe Beispiel widerlegt die Notwendigkeit der sequentiellen Abarbeitung von Aktivitäten. Abschließend sei außerdem anzumerken, dass die ermittelten Aktivitäten und deren Einordnung in die Testprozessphasen das unternehmensindividuelle Bild des Testprozesses der LTLS widerspiegeln. Beispielsweise wird die Aktivität vier (Review der Testbasis) aus der Phase der Testplanung im fundamentalen Testprozess als eine Aktivität der Phase *Testanalyse und Design* angesehen.

4.1.2 Wichtige Objekte und Beziehungen im Testprozess von UE/I

Zur weiteren Analyse der Ist-Situation wurden die Beziehungen und Abhängigkeiten wichtiger im Testprozess involvierter Objekte untersucht. Dazu wurde ein einstündiges Interview mit zwei Experten, einem Tester und dem Testmanager von UE/I, durchgeführt. Es wurden bewusst zwei Experten, die unterschiedliche Rollen im Testprozess einnehmen, befragt. Der Vorteil dieses Vorgehens ist, dass dadurch das Wissen von zwei Experten, mit unterschiedlichen Sichten auf die Testaktivitäten, in eine gemeinsame Basis überführt werden kann.

Das Ziel war es aus den Aussagen der Experten ein Entity-Relationship-Modell (ER-Modell) ableiten zu können. Ein ER-Modell stellt bestimmte Objekte (Entitäten) zueinander in Beziehung (Relationen) und bietet die Möglichkeit die Objekte und deren Beziehungen durch zusätzliche Informationen und Eigenschaften (Attribute) zu beschreiben.⁹⁷

Bei der Befragung handelte es sich um ein standardisiertes Gruppeninterview. Im Verlauf des Interviews konnten sich dadurch beide Experten in Ihren Aussagen verbessern, ergänzen und gemeinsame Antworten generieren. Deshalb wird an dieser Stelle nicht zwischen den Aussagen beider Experten unterschieden. Außerdem wurde das Ziel „die Erstellung eines ER-Modells zu den Beziehungen und Abhängigkeiten im Testprozess“ zu Beginn des Interviews klar kommuniziert, sodass die Experten ihre Antworten bereits auf deren Verwendungsabsicht ausrichten konnten. Die Aussagen der Experten waren somit qualitativ hochwertig direkt verwendbar. An dieser Stelle soll bereits auf das ER-Modell als nachbereitetes Ergebnis des Interviews verwiesen werden (siehe Abb. A.8). Durch Hinzuziehen des Modells wird somit die Nachvollziehbarkeit des Interviewverlaufs sowie der Interviewergebnisse vereinfacht. Unter Berücksichtigung der Bestandteile und der Struktur eines ER-Modells wurden folgende drei Fragen zielgerichtet für das Interview vorbereitet und beantwortet:

1. *Welche fachlichen Objekte erachten Sie im Testprozess als besonders wichtig?*

Es wurden bewusst nur wesentliche Kernobjekte genannt, um das später entstehende ER-Modell nicht zu überladen und einen übersichtlichen Zusammenhang zwischen den Objekten herstellen zu können. Folgende zehn Objekte wurden als wichtig und zusammenhängend erachtet: *Anforderung, Testfall, Testelement, Testszenario, Testdatensatz, Testplan, Masterplan, Abweichung, Zwischenrelease (engl. Minor-Release), Release (engl. Major-Release / dt. auch Haupt-Release).*

⁹⁷ Vgl. Tabeling, 2006, S. 269.

Nach Aussagen der Experten, hätten noch weitaus mehr Objekte genannt werden können – die aufgeführten Objekte sind jedoch ausreichend, um überschaubare und bedeutende Zusammenhänge darstellen zu können.

2. Welche Beziehungen und Abhängigkeiten sehen Sie zwischen diesen Objekten, sowohl fachlich als auch mengenmäßig?

Diese Frage ist besonders von Bedeutung, da die Antworten der Experten (Testmanager und Tester) bereits eine zusammenhängende Beschreibung des Modells ergeben. Folgende Beziehungen und Abhängigkeiten wurden zwischen den zehn Objekten identifiziert.

Eine oder mehrere Anforderungen sollen im Rahmen von einem Release bzw. mehreren Zwischenreleases getestet werden. Eine oder mehrere Anforderungen sind dabei mindestens einem oder mehreren Testelementen zugeordnet, die jeweils in einem logischen Verbund testbare Anforderungen beinhalten. Ein Testelement, das beispielsweise eine bestimmte Funktion eines Systems zusammenhängend testen soll, benötigt dazu einen oder mehrere Testfälle. Es besteht jedoch die Möglichkeit, dass ein Testfall zum Test von mehreren Testelementen genutzt wird. Ein Testelement ist genau einem Mastertestplan zugeordnet, wogegen im Masterplan der Test mehrerer Testelemente vorgesehen ist. Zur Ausführung eines Testfalles werden ein oder mehrere Testdatensätze benutzt. Im Umkehrschluss kann ein Testdatensatz von einem oder mehreren Testfällen verwendet werden. Weiterhin stehen Testfälle mit Testplänen in Beziehung. Ein Testfall kann in einem oder mehreren Testplänen referenziert sein. Genauso kann ein Testplan einen oder mehrere Testfälle referenzieren. Weiterhin ist für ein Release genau ein Masterplan definiert. Ein Testplan ist genau einem Mastertestplan zugeordnet, wogegen einem Mastertestplan mehrere Testpläne zugeordnet sein können. Außerdem wird ein Testplan in einem oder mehreren Zwischenreleases ausgeführt bzw. getestet und ein Zwischenrelease besitzt genau einen Testplan. Das Zwischenrelease ist hingegen genau einem Release zugeordnet, wobei im Rahmen des (Haupt-) Releases in der Regel mehrere Zwischenreleases vorgesehen sind. Werden durch einen Testfall eine oder mehrere Abweichungen aufgedeckt, so werden diese im Fehlermanagementsystem angelegt. Wurde keine Abweichung entdeckt, so referenziert der entsprechende Testfall auch nicht auf das Fehlermanagementsystem. Ein Fehlereintrag ist außerdem genau einem Zwischenrelease zugeordnet. Im Gegenzug kann es zu einem Zwischenrelease mehrere Fehler geben – es besteht jedoch auch die Möglichkeit, dass kein Fehler existiert.

Neben diesen fachlichen Beziehungen existieren strukturelle Beziehungen zwischen einigen Objekten. Beispielsweise ist ein Zwischenrelease eine Spezialisierung bzw. eine Unterform eines Release. Im Gegenzug stellt ein Release die Generalisierung, also die Verallgemeinerung eines Zwischenrelease dar. Ein Zwischenrelease ist also implizit auch ein Release und ein Release kann die spezifische Form eines Zwischenrelease annehmen. Abgesehen von Spezialisierungen und Generalisierungen existieren im Modell Objekte mit gruppierendem und zusammenfassendem Hintergrund. Ein Testszenario gruppiert und fasst mehrere Testfälle gemäß einem Testziel zusammen. Ebenso werden mehrere Zwischenreleases im Rahmen eines Release durchgeführt. Im Modell existieren außerdem wichtige transitive Abhängigkeiten. Beispielsweise werden Testfälle in Testszenarien zusammengefasst und einem Testplan zugeordnet. Es kann somit eine transitive Beziehung zwischen dem Testplan und dem Testszenario hergestellt werden. Demnach kann aus dem Modell abgeleitet werden, dass indirekt auch die Möglichkeit besteht, ganze Testszenarien einem Testplan zuzuordnen (transitive Eigenschaft).

3. *Vorausgesetzt die Objekte und deren Beziehungen sollen über ein Testmanagementwerkzeug abgebildet werden: Welche Eigenschaften und Informationen der Objekte erachten Sie als kennzeichnend und wichtig?*

Die gestellte Frage verdeutlicht bereits – nicht zu jedem Objekt wurden Eigenschaften (Attribute) formuliert, da nicht jedes Objekt für den Kontext „Auswahl und Einführung eines Testmanagementsystems“ relevante Inhalte mit sich bringt.

Objekt – „Anforderung“: Anforderungen besitzen einen *Prozessstatus*, der Aufschluss über den aktuellen Zustand der Anforderungen gibt.

Objekt – „Testfall“: Ein Testfall wird in *Testschritten* abgearbeitet. Zur Nachvollziehbarkeit des Testfallzustandes besitzt er einen *Ausführungstatus*, sowie eine *Version*. Weil Struktur und Inhalte von Testfällen oft wiederverwendet werden, kann die Testfallerstellung unter Benutzung von *Testfall-Vorlagen* (engl. *Templates*) hilfreich sein.

Objekt – „Abweichung“: Eine Abweichung bzw. ein Fehlereintrag besitzt einen *Bearbeitungsstatus*, sowie eine Systemversion in welcher der Fehler entdeckt wurde (engl. *affects version*) und eine Zielversion in welcher der Fehler behoben werden soll (engl. *fix version*).⁹⁸

⁹⁸ Vgl. Testhandbuch - UE/I, S. 61 (interne Quelle).

Objekt – „Release / Zwischenrelease“: Ein Release bzw. Zwischenrelease besitzt eine *Version* und ein *Release-Datum*, das angibt für welchen Termin ein Release geplant ist bzw. bis wann es abgeschlossen sein soll.

Im Ergebnis des Interviews wurden zehn Objekte identifiziert, die zueinander in Beziehung gestellt wurden. Werden die gewonnenen Erkenntnisse zusammengefasst, so lässt sich daraus das in Abbildung A.8 dargestellte ER-Modell in (min, max)-Notation generieren. Die Entitäten sind rechteckig, Beziehungen in Form einer Raute und Attribute kreisförmig dargestellt.

4.1.3 Anwendung des TPI-Modells

Das TPI-Modell wird bei UE/I verwendet, um das Testvorgehen kontinuierlich zu verbessern. Es gibt dabei Auskunft über den Reifegrad des bei UE/I vorhandenen Testprozesses und verdeutlicht dessen Stärken und Schwächen. Das TPI-Modell wird bei UE/I projektbegleitend angewendet, das heißt im Zuge der Durchführung eines Projektes soll der Testprozess mit Hilfe von TPI verbessert werden. Dies kann unter Umständen sogar zwingend notwendig sein, beispielsweise um abhängig von der Komplexität des Projektes überhaupt wirtschaftlich akzeptable und effiziente Testarbeiten durchführen zu können. Zu Projektbeginn wird das Ergebnis der TPI-Matrix ermittelt, die den aktuellen Fortschritt des Testprozesses aufzeigt. Häufig dient das Ergebnis des letzten durch TPI-Betrachtungen begleiteten Projektes als Einstieg. Net^{WG} ist das neueste Projekt der LTLS, welches Prozessverbesserungen durch TPI-Maßnahmen mit sich brachte. An dieser Stelle wird kurz auf den erreichten Fortschritt im Rahmen des Projektes eingegangen. Die nachfolgende TPI-Matrix stellt den Istzustand des Testprozesses bei UE/I vor Projektbeginn dar. Bereits erreichte Bausteine sind hierbei grün markiert.

Kernbereich	Stufe	beherrschbar					effizient					optimierend			
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Teststrategie		A					B				C		D	
2	Einsatz des Phasenmodells		A			B									
3	Zeitpunkt der Beteiligung			A				B				C		D	
4	Kostenvoranschlag und Planung				A							B			
5	Testspezifikationstechniken		A		B										
6	Statische Testtechniken					A		B							
7	Metriken						A			B			C		D
8	Testautomatisierung				A				B			C			
9	Testumgebung				A				B						C
10	Testarbeitsplatz				A										
11	Engagement und Motivation		A				B						C		
12	Testfunktionen und Ausbildungen				A			B				C			
13	Reichweite der Methodik					A						B			C
14	Kommunikation			A		B							C		
15	Berichterstattung		A			B		C					D		
16	Fehlermanagement		A				B		C						
17	Testwaremanagement			A			B				C				D
18	Testprozessmanagement		A		B								C		
19	Prüfen							A			B				
20	Low-Level-Tests					A		B		C					

Quelle: Interne Quelle – Ergebnispräsentation des Projektes Net^{WG}

Tab. 4.1: TPI-Matrix – Istzustand bei UE/I vor der Projektdurchführung

Aus der TPI-Matrix ist erkennbar, dass der Testprozess bei UE/I vor Projektbeginn noch nicht sehr entwickelt war. Da auf Stufe eins keine vollständige Erfüllung der Kontrollpunkte aller Ebenen vorliegt, kann der gesamte Testprozess nur mit Stufe 0 (Initialstufe) klassifiziert werden. Ausgehend vom Istzustand wurde im Projekt ein, als angemessen befundener Sollzustand, des Testprozesses vereinbart. Es galt die Stufe 3 zu erreichen und den Testprozess damit gezielt und schrittweise zu verbessern. Die folgende TPI-Matrix zeigt den angestrebten Sollzustand des Testprozesses gemäß den Zielvereinbarungen im Rahmen von Net^{WG}. Die fehlenden Bausteine zur Erreichung von Stufe eins sind gelb, von Stufe zwei – orange, und Stufe drei – rot markiert.

Kernbereich	Stufe	beherrschbar					effizient					optimierend			
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Teststrategie		A					B				C		D	
2	Einsatz des Phasenmodells		A			B									
3	Zeitpunkt der Beteiligung			A				B				C		D	
4	Kostenvoranschlag und Planung				A							B			
5	Testspezifikationstechniken		A		B										
6	Statische Testtechniken					A		B							
7	Metriken						A			B		C		D	
8	Testautomatisierung				A				B			C			
9	Testumgebung				A				B						C
10	Testarbeitsplatz				A										
11	Engagement und Motivation		A				B					C			
12	Testfunktionen und Ausbildungen				A			B				C			
13	Reichweite der Methodik					A						B			C
14	Kommunikation			A		B							C		
15	Berichterstattung		A			B		C					D		
16	Fehlermanagement		A				B		C						
17	Testwaremanagement			A			B				C				D
18	Testprozessmanagement		A		B								C		
19	Prüfen							A			B				
20	Low-Level-Tests					A		B			C				

Quelle: Interne Quelle – Ergebnispräsentation des Projektes Net^{WG}

Tab. 4.2: TPI-Matrix – Sollzustand bei UE/I gemäß Projektzielvereinbarung

Insgesamt galt es neun Ebenen zu erreichen, was bedeutet, dass eine Vielzahl von Kontrollpunkten erfüllt werden mussten. Im Verlauf des Projektes konnten jedoch nicht alle Kontrollpunkte der Ebenen erfüllt werden, sodass nach Abschluss des Projektes zwei von neun Ebenen nach wie vor nicht erreicht waren. Obwohl es Verbesserungen des Testprozesses in sieben Kernbereichen gab, konnte der Gesamtprozess weiterhin nur auf Stufe null klassifiziert werden. Dies ist ein großer Nachteil des TPI-Modells, denn um einen Fortschritt auf der Stufenskala zu erzielen, dürfen keine Kernbereiche vernachlässigt werden. Die TPI-Stufe ist somit, bezüglich des allgemeinen Entwicklungsstandes des Testprozesses, nicht aussagekräftig. Durch nachträgliche Verbesserungsmaßnahmen konnte das Defizit in den zwei, im Projekt nicht erreichten Ebenen „Kostenvoranschlag und Planung“ und „Engagement und Motivation“, ausgeglichen werden. Der Testprozess von UE/I konnte somit auf TPI-Stufe drei

klassifiziert werden. Die Tabelle 4.3 zeigt die Ergebnismatrix nach der Projektdurchführung und enthält bereits die im Nachgang des Projektes durchgeführten Verbesserungsmaßnahmen zur Erreichung von Stufe drei.

Kernbereich		Stufe	beherrschbar					effizient					optimierend		
			0	1	2	3	4	5	6	7	8	9	10	11	12
1	Teststrategie		A					B				C		D	
2	Einsatz des Phasenmodells		A			B									
3	Zeitpunkt der Beteiligung			A				B				C		D	
4	Kostenvoranschlag und Planung				A							B			
5	Testspezifikationstechniken		A		B										
6	Statische Testtechniken					A		B							
7	Metriken						A			B			C		D
8	Testautomatisierung				A				B			C			
9	Testumgebung				A				B						C
10	Testarbeitsplatz				A										
11	Engagement und Motivation		A				B						C		
12	Testfunktionen und Ausbildungen				A			B				C			
13	Reichweite der Methodik					A						B			C
14	Kommunikation			A		B							C		
15	Berichterstattung		A			B		C					D		
16	Fehlermanagement		A				B		C						
17	Testwaremanagement			A			B				C				D
18	Testprozessmanagement		A		B								C		
19	Prüfen							A			B				
20	Low-Level-Tests					A		B		C					

Quelle: Interne Quelle – Ergebnispräsentation des Projektes Net^{WG}

Tab. 4.3: TPI-Matrix – Istzustand bei UE/I nach der Projektdurchführung

Die Ergebnismatrix stellt den aktuellen Stand bei UE/I in Bezug auf Prozessverbesserungen durch TPI-Maßnahmen dar. Die Matrixübersicht lässt erkennen, dass noch viele Möglichkeiten für weitere Verbesserungen des Testprozesses vorhanden sind. Das Ergebnis dient in Abschnitt 4.2.2 außerdem als Ausgangspunkt für die Ableitung von Anforderungen an das Testmanagementwerkzeug.

4.1.4 Methoden und Werkzeugeinsatz

Obgleich das Ist-Modell im Wesentlichen durch die Ergebnisse der Prozessanalyse (siehe Kapitel 4.1.1 und 4.1.2) charakterisiert wird, wurde untersucht mit welchen Methoden und Werkzeugen im Testmanagementumfeld gearbeitet wird. Auch die Ergebnisse dieser Betrachtungen konnten Informationen für die Ermittlung von weiteren Anforderungen bieten. Hauptsächlich ging es jedoch darum, einen Praxisbezug zu den in Kapitel 4.1.1 und 4.1.2 dargestellten Ergebnissen herzustellen. Zur Informationsgewinnung wurden vom Testmanager bereitgestellte interne Dokumente und das Testhandbuch von UE/I untersucht. Unter Beachtung folgender Zielfragestellung galt es die gegenwärtige Arbeitssituation des Testmanagers zu beschreiben:

» *Mit Hilfe welcher Mittel (Werkzeuge) und Methoden wird bisher gearbeitet?* «

Der Testmanager von UE/I arbeitet mit verschiedenen Werkzeugen. In erster Linie handelt es sich dabei um Microsoft Excel Tabellen in einfacher bis sehr komplexer Struktur. Viele Tätigkeiten, die mittels dieser Tabellen stattfinden, spiegeln bereits bekannte Aktivitäten (siehe Kapitel 4.1.1) und Objektbeziehungen (siehe Kapitel 4.1.2) wider. Beispielsweise werden bestimmte Testfälle in verschiedenen Tabellen zusammengefasst und geordnet. Sie bilden den Testfallpool aus dem die Testfälle für die Erstellung eines Testplanes ausgewählt werden (siehe Abb. A.4). Ein Großteil dieser Tätigkeiten, die auf Grundlage der Excel Tabellen durchgeführt werden, sollen zukünftig im Testmanagementwerkzeug erfolgen.

Eine separate und zentrale Erfassung von Anforderungen, etwa in einem *Anforderungsmanagementsystem*, existiert bei UE/I bisher nicht. Stattdessen werden die Anforderungen in verschiedenen Dokumenten erfasst, gesammelt und z. B. in einer Ordnerstruktur organisiert und verwaltet.

Bei der LTLS wird außerdem das *Fehlermanagementsystem JIRA* zur Dokumentation sowie der gezielten Steuerung und Nachverfolgung von Abweichungen und Fehlern verwendet.

In der Phase der Testdurchführung steht zusätzlich die Software „HP Quicktest“ zur Testautomatisierung zur Verfügung, das bisher jedoch nicht in vielen Testprojekten genutzt wird.

4.2 Ableitung der Soll-Situation (Anforderungsanalyse)

Ziel und Inhalt dieses Kapitels ist das Ableiten von Anforderungen und die Erstellung eines Anforderungskataloges. Die in 4.1 erhaltenen Analyseergebnisse bilden die Grundlage für das Ableiten von Anforderungen an das Testmanagementwerkzeug. Die Ermittlung der Anforderungen basierte dabei auf:

- Den Testprozessaktivitäten bei UE/I (4.1.1)
- Wichtigen Objektbeziehungen im Testprozess von UE/I (4.1.2)
- Dem aktuellen Entwicklungsstand durch TPI-Maßnahmen (4.1.3) und den zukünftig zu erreichenden TPI-Kontrollpunkten und Ebenen mit Hilfe bereits geplanter TPI-Maßnahmen
- Den verwendeten Methoden und Werkzeugen (4.1.4)

Die ermittelten Anforderungen wurden fortlaufend nummeriert, um eine spätere Identifizierung im Anforderungskatalog zu gewährleisten.

4.2.1 Anforderungen auf Basis des Testprozesses

Das Grundlagenmodell über den Testprozesses von UE/I wurde durch die Aktivitätsbeschreibungen der einzelnen Testprozessphasen (siehe Kapitel 4.1.1) und das Identifizieren wichtiger Objekte, Beziehungen und Eigenschaften (siehe Kapitel 4.1.2) gebildet. Durch weitere Analyse dieser Ausgangsbasis konnte ein Großteil der Anforderungen an das Testmanagementwerkzeug ermittelt werden. Diese Analyse wurde in einem erneuten Workshop mit dem Testmanager von UE/I durchgeführt, da er im Testprozess über das umfangreichste Hintergrundwissen zu den einzelnen Aktivitäten verfügt. Die vorhandenen Modelle aus 4.1.1 und 4.1.2 bildeten eine gute Ausgangsbasis für das methodische und schrittweise Vorkommen im Workshop. Die Testprozessphasen wurden der Reihe nach abgearbeitet und die Aktivitäten der Aktivitätsdiagramme Schritt für Schritt betrachtet. Diese Vorgehensweise stellte eine Art Fahrplan für die Durchführung des Workshops dar. Außerdem wurde das ER-Modell (siehe Abb. A.8) parallel zu den Aktivitätsdiagrammen betrachtet. Das konkrete Ziel der systematischen Anforderungsermittlung konnte im Workshop somit ohne größere Probleme umgesetzt werden.

Die Anforderungen konnten auf verschiedene Art und Weise abgeleitet werden.

Anforderung wurden...

- ...direkt aus den Aktivitäten abgeleitet.
- ...auf Basis von Kontextinformationen zu Aktivitäten geschlussfolgert.
- ...aus den Aktivitäten abgeleitet und durch Objektbeziehungen des ER-Modells belegt.
- ...direkt aus Objekten, Beziehungen oder Attributen des ER-Modells abgeleitet.

Im Folgenden wird die Vorgehensweise zur Ermittlung der Anforderungen, exemplarisch für die Phasen der *initialen Testplanung* und der *Testplanung auf Basis von Zwischenreleases*, dargestellt. Die Wahl fiel auf diese beiden Testprozessphasen, da sie den größten Stellenwert bezüglich der Aktivitäten des Testmanagements besitzen. Anschließend werden alle Anforderungen dargestellt, die allein durch die Analyse des ER-Modells (siehe Abb. A.8) ermittelt werden konnten. Die Anforderungen wurden außerdem in der Möglichkeitsform (Konjunktiv) formuliert, da sie zum Zeitpunkt der Analyse noch keine Gewichtung bzw. Priorisierung besaßen.

Die im Text verwendete Struktur zur Beschreibung der ermittelten Anforderungen soll zunächst anhand eines Musters erklärt werden:

(Aus X) Zunächst wird der Kontext, der zur Ermittlung der Anforderung diente, hergestellt. X kennzeichnet die verwendete Quelle, z. B. eine Aktivitäten-Nummer aus den Aktivitätsdiagrammen (siehe Abb. A.1 und A.4 sowie Kapitel 4.1.1) oder das ER-Modell aus Abbildung A.8. In den Testprozessphasen verwendete Objekte werden durch die in den Aktivitätsdiagrammen verwendeten Buchstaben gekennzeichnet (siehe Abb. A.1 und A.4)

(A.N) Die Anforderung (A) wird mit einer identifizierenden Nummer (N) formuliert.

Aktivitäten der initialen Testplanung:

(Aus I.) Planerische, zu Beginn eines Projektes erhobene Daten, werden im Testkonzept für ein Release dokumentiert.

(A.1) Mit dem Testmanagementwerkzeug können Releases und deren Zeitplan definiert werden.

(Aus 3.) In der Testbasis (C) werden sämtliche identifizierte Anforderungsdokumente gesammelt und verwaltet. Eine Abbildung oder das Referenzieren der Testbasis (Anforderungen) kann in einem Testmanagementwerkzeug sehr unterschiedlich flexibel gehandhabt werden.

- (A.2) Das Werkzeug bietet eine Verwaltung für Anforderungen mit Funktionen, wie Anlegen, Ändern und Löschen.
- (A.3) Anforderungen sollen strukturiert und gruppiert werden können, um diese z. B. nach Anforderungstyp oder Zweck ordnen zu können (z. B. in Ordnern bzw. einer Baumstruktur).
- (A.4) Anforderungen sind außerdem in Prosaform erfassbar.
- (A.5) Anforderungen können miteinander verknüpft werden, um Abhängigkeiten abzubilden.
- (A.6) Den Anforderungen können Anhänge, beispielsweise in Form von Dokumenten oder Bildern, angefügt werden.
- (A.7) Anforderungen können mittels gängigem Excel- oder CSV-Format importiert oder exportiert werden.

(Aus 4. / Abb. A.8) Ein Anforderungsdokument, das bereits auf Qualität geprüft wurde, sollte kenntlich gemacht werden. Untermuert durch das Attribut „Status“ aus dem ER-Modell ergibt sich folgende Anforderung:

- (A.8) Zu den Anforderungen ist ein Prozessstatus definierbar oder wählbar (z. B. NEW, DESIGN, to-Be-Reviewed, READY, COMMITED).

(Aus 11.) Die ermittelten Bedarfe der Testaufwandsschätzung müssen dokumentiert werden.

- (A.9) Das Werkzeug bietet Funktionen zur Unterstützung der Aufwandsplanung bzw. -zuweisung, indem es beispielsweise zeitliche Daten erfassen kann (z. B. Dauer der Tests auf Testfallebene oder Testzyklus- bzw. Testphasenebene).

(Aus Abb. A.8) Anforderungen die implizit aus dem ER-Modell abgeleitet werden können:

- (A.10) Anforderungen können einem Release zugeordnet werden.
- (A.11) Anforderungen können einem Verantwortlichen zugewiesen werden.

Testplanung auf Basis von Zwischenreleases:

(Aus Abb. A.8) Im ER-Modell besteht eine direkte Beziehung zwischen den Anforderungen und dem Release, sowie zwischen dem Release und Zwischenrelease. Außerdem ergibt sich eine transitive Abhängigkeit zwischen den Anforderungen und dem Zwischenrelease.

(A.12) Einem Release können mehrere Zwischenreleases zugeordnet werden.

(A.13) Anforderungen können einem Zwischenrelease zugeordnet werden.

(Aus 1. / Abb. A.8) Releases werden in mehreren Zwischenreleases für den Test bereitgestellt. Die Testplanung erfolgt auf Ebene der Zwischenreleases. Testpläne werden bei UE/I also für jedes Zwischenrelease angelegt.

(A.14) Für ein Zwischenrelease können Testpläne angelegt werden.

(A.15) Der Import und Export von Testplänen ist beispielsweise im Excel- oder CSV-Format möglich.

(Aus 2. / 3. / Abb. A.8) Angelegte Testpläne müssen mit relevanten Testfällen gefüllt werden. Das ER-Modell verdeutlicht diese Beziehung.

(A.16) Je Testplan können bereitgestellte Anforderungen bzw. Testfälle, die die Anforderungen prüfen, sowie Testfälle für den Re-Test ausgewählt und zugeordnet werden.

(Aus 5. / Abb. A.8) Im Testplan erfolgt eine Verteilung der Tests auf die Mitarbeiter.

(A.17) Testfälle und Testfallgruppierungsobjekte können Benutzern oder Gruppen (Testern / Verantwortlichen) zugeordnet werden.

(A.18) Benutzer können, z. B. über Email, automatisch benachrichtigt werden, wenn ihnen Testaufgaben (z. B. Testfälle) zugewiesen worden sind.

Die folgenden Anforderungen konnten durch die Analyse des ER-Modells und ohne Betrachtung weiterer Aktivitäten abgeleitet werden.

(Aus Abb. A.8) Anforderungen, die direkt oder durch transitive Abhängigkeiten abgeleitet worden sind.

(A.19) Testfälle können, beispielsweise mittels Testfall-Gruppierungsobjekten (siehe „Testszenarien“ in Kapitel 2.3.3) in einer Ordner- oder Baumstruktur, gruppiert und verwaltet werden.

Es gibt keinen standardisierten Begriff für Gruppierungselemente von Testfällen. Es werden häufig verschiedene Begriffe wie Testsuite oder Testszenario in dieser Hinsicht verwendet.

(A.20) Jeder Testfall kann mehreren Testfall-Gruppierungsobjekten (z. B. Testsuiten bzw. Testszenarien) zugeordnet werden.

(A.21) Ein Testfall kann mehreren Testplänen zugeordnet werden.

(A.22) Jeder Testfall kann mehreren Anforderungen zugeordnet werden.

Hintergrund im Testprozess bei UE/I: Bestimmte Anforderungen werden in Testelementen logisch gruppiert getestet. Den Testelementen zugeordnete Testfälle stehen somit über die Testelemente in Verbindung mit den Anforderungen.

(A.23) An den Testfällen ist ein Prozessstatus wählbar oder definierbar (z. B. NEW, DESIGN, READY, COMMITTED).

(A.24) Jeder Testfall kann beliebig viele Testschritte enthalten.

(A.25) Ausgehend von (A.24) kann ein Testschritt eine Beschreibung, ein Sollergebnis und Testdaten besitzen.

(A.26) Abweichungen können einem Verantwortlichen zugeordnet werden.

Es wurde beispielhaft gezeigt, wie verschiedene Anforderungen an das Testmanagementwerkzeug ermittelt werden konnten. Das gleiche analytische Vorgehen wurde für jede weitere Testprozessphase aus Kapitel 4.1.1 angewendet. Aus den Aktivitäten der Teststeuerung sind keine Anforderungen an das Testmanagementwerkzeug hervorgegangen, da sich der steuernde Charakter erst durch eine entsprechend häufige Interaktion von Benutzern mit dem System bzw. Werkzeug ergibt. Dies gilt insbesondere für die steuernden Absichten und Funktionen des Testmanagers. Ein steuernder Charakter ergibt sich beispielsweise, indem das Werkzeug bestimmte Auswertungsmöglichkeiten (Anforderungen aus der Phase *Testauswertung und Bericht*) bietet, die dem Testmanager bei der Überwachung des Testprozesses helfen oder ihm eine frühzeitige Aktualisierung der Testplanung ermöglichen.

4.2.2 Anforderungen durch TPI-Maßnahmen

Auf Grundlage der in Kapitel 4.1.3 dargestellten Ist-Situation in Bezug auf Testprozessverbesserungen durch TPI, werden Anforderungen an das Testmanagementsystem abgeleitet. Ein gegenwärtig durch TPI-Maßnahmen begleitetes Großprojekt der LTLS sieht als realistisches Ziel vor, durch weitere Prozessverbesserungen die TPI Stufe vier zu erreichen. Ausgehend von der Ergebnismatrix (Tab 4.3) stellt sich im Kontext dieses Kapitels die Frage:

» *Wie kann ein Testmanagementwerkzeug zur Erreichung weiterer Ebenen bzw. zur Erfüllung weiterer Kontrollpunkte beitragen?* «

Dazu hätten zunächst alle nicht erfüllten Kontrollpunkte bis Stufe vier betrachtet werden müssen. Es ist jedoch davon auszugehen, dass auch Ebenen der Stufen größer als vier durch ein Testmanagementwerkzeug unterstützt werden können. Deshalb wurden vorausschauend auch alle Ebenen und Kontrollpunkte der Stufen fünf und sechs in die Betrachtungen einbezogen. Die korrespondierende Zielmatrix ist im Anhang (Tab. B.1) dargestellt. Die fehlenden Bausteine zur Erreichung von Stufe vier sind rot – die zusätzlich untersuchten Ebenen grau markiert. Eine ausführliche Beschreibung aller Kontrollpunkte und Ebenen kann in „*Management und Optimierung des Testprozesses: ein praktischer Leitfaden für erfolgreiches Testen mit TPI und TMap*“⁹⁹ eingesehen werden. Zum Untersuchungszeitpunkt sind von insgesamt 134 Kontrollpunkten (Stufe 1-6) bereits 68 Kontrollpunkte erreicht und 66 offen bzw. noch nicht erreicht. Die tiefgründige Untersuchung aller Ebenen und Kontrollpunkte bis Stufe sechs wäre mit erheblichem Aufwand verbunden gewesen. Außerdem wurden im Rahmen des gegenwärtigen Großprojektes der LTLS bereits alle speziell auf UE/I abgestimmten Maßnahmen zur Erreichung jeder Ebene bis einschließlich Stufe sechs festgelegt. Es musste dadurch lediglich geprüft werden, welche Maßnahmen durch ein Testmanagementsystem unterstützt werden können, weshalb der notwendige analytische Aufwand maßgeblich reduziert werden konnte. Zum Erreichen einiger Ebenen wurden Maßnahmen zur Durchführung vorbereitet oder bereits eingeleitet, die zum Erreichen der entsprechenden Ebene verhelfen. Die mögliche Unterstützung eines Testmanagementsystems musste an diesen Stellen nicht weiter untersucht werden, weshalb nicht alle Ebenen bis Stufe sechs näher betrachtet werden mussten. Eine Liste der noch durchzuführenden oder zu unterstützenden Maßnahmen, zum Erreichen jeder Ebene bis einschließlich Stufe sechs, ist dem Anhang beigelegt (siehe Tab. B.2).

Maßnahmen, die durch ein Testmanagementsystem unterstützt werden können oder erfüllbar sind, stellen implizit auch Anforderungen an das Werkzeug dar. Im Folgenden werden diese Maßnahmen identifiziert. Außerdem wird allgemein beschrieben, warum

⁹⁹ Pol, 2002, S. 83-178.

ein Testmanagementwerkzeug diese Maßnahmen unterstützen kann. Anschließend wird beispielhaft an einer Maßnahme erläutert, inwiefern diese nicht durch ein Testmanagementsystem unterstützt werden kann. Die Maßnahmen werden außerdem über ihre Identifikationsnummern (ID) referenziert (siehe Tab. B.2).

(M.1) *Verfassung von Testabschlussbericht inkl. Betrachtung der offenstehenden Abweichungen, Risiken und Empfehlungen*

Ein Testfall ist für das Prüfen von einer oder mehreren Anforderungen zuständig und kann im Testverlauf Abweichungen identifizieren. Weiterhin sind mit verschiedenen Anforderungen bestimmte Risiken verbunden. Folglich sind auch mit offenstehenden Abweichungen gewisse Risiken verknüpft, da bestimmte Anforderungen nicht erfüllt sind.

Diese Maßnahme kann durch ein Testmanagementwerkzeug unterstützt werden, indem es entsprechende Auswertungsmöglichkeiten (Reporting) über gelistete Abweichungen und den damit verbundenen Risiken bietet.

(M.4) *Testfallausführung: Erfassung von Abarbeitungszeit (Durchlaufzeit); Was wurde getestet? Was ist korrekt verlaufen? Was ist noch zu testen?*

Das Testmanagementsystem kann diese Maßnahme durch geeignete Auswertungen der Tests unterstützen, indem es Aufwände erfassen kann und entsprechende Informationen zu Testfällen, Testplänen, sowie dem Testfortschritt bereitstellt.

(M.5) *Erfassung und Reporting von projektorientierten Metriken*

Metriken werden dazu eingesetzt den Testprozess zu überwachen und ermöglichen einen Vergleich von Systemen und Prozessen. Des Weiteren können sie zur Begründung von Empfehlungen herangezogen werden. Einige Metriken dienen als Grundlage für die Erstellung von Testberichten, die für die Projektleitung bestimmt sind. Eine projektorientierte Metrik kann dabei beispielsweise Informationen zur Bewertung des Projektfortschrittes liefern.¹⁰⁰ Ein Testmanagementsystem, das diese Metriken abbilden und dementsprechende Auswertungsergebnisse liefern kann, würde entscheidend zur Umsetzung dieser Maßnahme beitragen.

(M.8) *Fehlermanagement umfassend und konsequent betreiben*

Durch eine Integration des Testmanagementwerkzeuges mit dem Fehlermanagementwerkzeug der LTLS (JIRA), sind die Abweichungen, je nach Tiefe der Integration, auch im Testmanagementwerkzeug referenziert und für dessen Benutzer komfortabel

¹⁰⁰ Vgl. Testhandbuch – UE/I, S. 45 (interne Quelle).

abrufbar. Außerdem sind die Abweichungen dort direkt zu den Testfällen, sowie indirekt zu den Anforderungen, zugeordnet. Die Integration mit JIRA wurde durch sonstige Betrachtungen in Kapitel 4.2.3 als Anforderung (A.42) ermittelt – sie hätte jedoch auch in Bezug auf diese Maßnahme (M.8) ermittelt werden können.

Darüber hinaus kann ein Testmanagementwerkzeug, das verschiedene Auswertungen zu aufgetretenen Abweichungen bereitstellt, zur Erstellung von Trendberichten beitragen und das Fehlermanagement unterstützen.

(M.11) *Berichte um Trendbewertung und Risikoabschätzung, inkl. Empfehlungen, erweitern*

Diese Maßnahme kann ebenfalls durch entsprechende Auswertungsmöglichkeiten (Reporting) des Testmanagementwerkzeuges, die als Grundlage für Bewertungen und Schätzungen von Trends und Risiken dienen, unterstützt werden.

Im Folgenden wird anhand der Maßnahme (M.7) beschrieben, weshalb diese nicht durch ein Testmanagementsystem unterstützt werden kann.

(M.7) *Abstimmung zwischen Test- und Entwicklungsabteilung*

Der Inhalt eines Release wird zwischen der Entwicklungs- und der Testabteilung abgestimmt. Außerdem hat die Testplanung Einfluss auf die inhaltliche Reihenfolge der Release-Lieferungen. Da es sich um eine rein organisatorische und kommunikative Angelegenheit zwischen der Entwicklungs- und Testabteilung handelt, kann das Testmanagementsystem keinen direkten Beitrag zur Umsetzung der Maßnahme liefern.

Im Zusammenhang mit diesem Sachverhalt konnte jedoch bereits in Kapitel 4.2.1 eine Anforderung formuliert werden. Die von der Entwicklungsabteilung umgesetzten Änderungen werden an das Testteam kommuniziert, wobei das Testmanagementwerkzeug die Funktion bietet, relevante Testfälle zu den umgesetzten Anforderungen auszuwählen und einem entsprechenden Testplan zuzuweisen (siehe Kapitel 4.2.1, Anforderung (A.18)).

Zusammenfassend konnte festgestellt werden, dass fünf von elf offenstehenden Maßnahmen durch ein Testmanagementsystem unterstützt werden können. Alle fünf Maßnahmen sind im Bereich *Testauswertung und Bericht* einzuordnen. Dies zeigt, dass der gegenwärtige Bedarf an Unterstützung im Bereich der Berichterstattung und Testauswertung (Metriken) als hoch einzustufen ist. Diesen Bereich kann ein Testmanagementsystem unterstützen, indem es Auswertungen, beispielsweise über Anforderungen, Testfälle, Testpläne oder Abweichungen, zur Verfügung stellen kann.

Dabei kann eine rein zahlenmäßige Auswertung oder eine Veranschaulichung der Auswertungen und Metriken erfolgen.

Da die unterstützbaren Maßnahmen potentielle Anforderungen an ein Testmanagementwerkzeug darstellen, mussten die Auswertungsmöglichkeiten durch Metriken als Anforderungen aufgenommen werden. Im Testhandbuch von UE/I sind alle Metriken dokumentiert, die im Testprozess Anwendung finden sollen. Um untersuchen zu können welche dieser Metriken durch ein Testmanagementsystem abgebildet werden können, wurden alle im Testhandbuch aufgelisteten Metriken in die weiteren Betrachtungen einbezogen. Hinter jeder Metrik verbirgt sich jeweils eine mathematische Formel, die einen Sachverhalt beschreibt. Für die Untersuchungen im Rahmen dieser Arbeit wird jedoch nicht näher auf den mathematischen Hintergrund der Metriken eingegangen. Es wird lediglich der Sachverhalt bzw. Kontext jeder Metrik geschildert, den es jeweils zu untersuchen galt.

Nachfolgend werden alle Metriken, unterteilt nach deren Verwendung für Testfortschritts-, Testabschluss- und Testbewertungsbericht, aufgelistet und durch eine Anforderungsnummer identifiziert:

Unterstützung der Erstellung von Testfortschrittsberichten:

- (A.27) Wie viele Anforderungen sind dem Testobjekt zugeordnet?
- (A.28) Wie hoch ist der Anteil an erfassten Anforderungen, denen wenigstens ein Testfall zugeordnet ist (Anforderungsüberdeckungsgrad)?
- (A.29) Wie ist der Fortschritt der Testfallerstellung?
- (A.30) Wie ist der aktuelle Testfortschritt (aktueller Status der Testfälle über Zeitraum x)?
- (A.31) Wie ist der Fehlerfindungstrend (Anzahl offener Abweichungen über Zeitraum x)?
- (A.32) Wie viel Test-Restaufwand ist noch für den Abschluss der Tests für ein Testobjekt nötig?
- (A.33) Wie ist der Trend für ein Testobjekt beim Abarbeiten von Testfällen?
- (A.34) Kann der vereinbarte Zieltermin gehalten werden?

Unterstützung der Erstellung von Testabschlussberichten:

- (A.35) Wie ist die Verteilung der Anforderungen hinsichtlich der Testintensität?
- (A.36) Wie groß ist der durchschnittliche Aufwand zur Spezifikation eines Testfalls für das Testobjekt (betrachtet je Testintensität)?
- (A.37) Wie viele Testzyklen (Re-Tests) waren im Mittel je Testfall notwendig (Gegebenenfalls je Testintensitätsklasse)?
- (A.38) Wie ist der aktuelle Status der eingeplanten Testfälle?
- (A.39) Wie ist der aktuelle Status der identifizierten Abweichungen?

Unterstützung der Erstellung von Testbewertungsberichten:

- (A.40) Wie viele Testfälle sind im Mittel einer Anforderung zugewiesen (Testüberdeckungsgrad)?
- (A.41) Wie groß ist der durchschnittliche Aufwand zur Durchführung eines Testfalls für das Testobjekt (betrachtet je Testintensität)?

Die Metriken der Anforderungen A.28, A.36 und A.37 dienen auch als Inhalt von Testbewertungsberichten.

4.2.3 Anforderungen durch sonstige Betrachtungen

In Kapitel 2.5 wurde auf die oftmals auftretende Notwendigkeit zur Integration von verschiedenen Werkzeugen und der zentralen Position eines Testmanagementsystems dabei hingewiesen. Im Kontext dieser Arbeit wird darauf Wert gelegt, dass das auszuwählende Testmanagementwerkzeug eine Integration mit bereits vorhandenen Werkzeugen der LTLS bietet. Dadurch werden vorhandene Potentiale dieser Werkzeuge genutzt und der Systemverbund gestärkt. Wie in Kapitel 4.1.4 erwähnt wurde, wird bei der LTLS das Fehlermanagementwerkzeug JIRA eingesetzt. Das Testmanagementwerkzeug muss dementsprechend die Möglichkeit zur Integration bieten, oder zumindest bestimmte Schnittstellen zur Anbindung an JIRA bereitstellen. Im Testprozess erkannte Fehlerwirkungen bzw. Abweichungen sollen somit direkt an das Fehlermanagementsystem weitergeleitet werden, um eine effektive Bearbeitung und Fehlerbehebung zu gewährleisten. Weiterhin könnten behobene Fehler an das Testmanagementwerkzeug rückgemeldet werden, damit das Testmanagement stets einen aktuellen Überblick über den Fortschritt der Fehlerbehebung besitzt.

Auf Grundlage der Betrachtungen und Ergebnisse aus Kapitel 4.1.4 konnten weitere wichtige Anforderungen abgeleitet werden.

(A.42) Das Testmanagementwerkzeug bietet Möglichkeiten zur Anbindung bzw. Integration an JIRA (Diese Anforderung ist auch trivial aus Abb. A.5 durch Aktivität fünf und Objekt (C) ableitbar).

(A.43) Das Werkzeug stellt Möglichkeiten zur Anbindung an „HP-Quicktest“ zur Testautomatisierung bereit.

Wie das Beispiel der Anforderung (A.42) zeigt, können einige Anforderungen auf Grundlage verschiedener Quellen ermittelt und belegt werden.

Nicht jedes Werkzeug kann problemlos bei der LTLS eingeführt und eingesetzt werden. Es gibt bestimmte infrastrukturelle Einschränkungen im IT Umfeld des Unternehmens, die eingehalten werden müssen. Es galt herauszufinden, welche systemseitigen Voraussetzungen die Werkzeuge mitbringen müssen, um bei der LTLS eingesetzt werden zu können. Darum wurde ein 60 minütiges, exploratives Interview mit dem IT Service Manager von UE/I durchgeführt. Ziel des Interviews war es „harte“ systemseitige Anforderungen an die zu bewertenden Werkzeuge zu identifizieren, die im späteren Vorauswahlprozess Anwendung finden sollten. Sie sind keine Anforderungen, die es während der Evaluierung zu überprüfen gilt, sondern stellen unumgängliche Einschränkungskriterien dar, die noch vor der eigentlichen Evaluierung überprüft werden müssen. Die Antwort bei der Prüfung auf eine harte Anforderung

kann nur mit ja oder nein bzw. erfüllt oder nicht erfüllt beantwortet werden.¹⁰¹ Die Fragen, die während des Interviews gestellt wurden, konnten im Nachhinein zu vier Schwerpunkten zusammengefasst werden. Nachfolgend werden diese vier Schwerpunkte in Kernfragen formuliert und die Inhalte der Antworten beschrieben.

1. Wo sind vorhandene Applikationen installiert und wer ist für den Betrieb zuständig?

Diese initial gestellte Frage sollte den technischen Hintergrund des Applikationsbetriebes ergründen und dient an dieser Stelle lediglich der Verständlichkeit der Fragestellung „Wo werden die verschiedenen Werkzeuge der LTLS betrieben“?

Die meisten Applikationen werden bei der Lufthansa Systems (LSY) betrieben, die der größte IT Service Partner der LTLS ist. Neue Anwendungen sollen bevorzugt bei der LSY betrieben werden.

2. Werden die Applikationen ausschließlich über die LSY betrieben (Eigenbetrieb im Konzern) oder besteht auch die Möglichkeit der Nutzung einer „Software as a Service“ Lösung (SaaS)?

SaaS ist ein transparentes Dienstmodell des Cloud-Computing. Mittels SaaS-Modell werden Applikationen in der Cloud (Datenwolke im Internet) bereitgestellt und der Anwender greift über einen Web-Client auf die Applikation zu. Der Anbieter ist für den Betrieb der Applikation verantwortlich. Auf der Anwenderseite wird daher keine IT-Infrastruktur benötigt, um die Applikation zu betreiben. Der Zugriff erfolgt somit zeit- und ortsunabhängig.¹⁰²

Prinzipiell ist auch der Fremdbetrieb von Applikationen in Form einer Serviceleistung möglich. Einen Schwerpunkt würde die Integration zu Systemen des Unternehmens (z. B. JIRA) darstellen. Dies ist schwierig zu realisieren, da eine hohe Wahrscheinlichkeit besteht, dass konzeptionelle und technische Probleme bei der Integration auftreten oder zusätzliche enorme Kostenaufwände entstehen. Außerdem sind intensive Analysen der Anbieter und umfangreiche vertragliche Abstimmungen notwendig. Im Rahmen dieses Projektes und dieser Arbeit wird deshalb auf SaaS-Lösungen verzichtet.

¹⁰¹ Vgl. Ludewig, 2007, S. 349.

¹⁰² Vgl. Schill, 2012, S. 31f.

3. Welche Möglichkeiten zur Inbetriebnahme eines neuen Werkzeuges sind bei der LTLS gegeben bzw. zulässig?

Es muss zunächst zwischen der Serverseite und dem Client unterschieden werden. Auf Seiten des Serverbetriebes sind folgende vier Plattformen der LSY zugelassen, die zusätzlich nach dem Einsatzbereich des Datenbank- und Applikationsbetriebs unterschieden werden:

Datenbankenbetrieb:

Eine Plattform namens *L/baseDATA* steht für den Betrieb von Oracle Datenbanken zur Verfügung, wobei nur Versionen ab Oracle 11g – Version 11.2.0.2 – unterstützt werden.

Applikationsserver:

Auf der zweiten Plattform *L/baseAPPS*, die *HP Unix* in der Version 11.23 als Betriebssystem verwendet, können Applikationen, die auf Unix-Betriebssystemen lauffähig sind, betrieben werden.

Außerdem steht als dritte Plattform ein virtueller *VMWare* Server mit *Linux* (Red Hat oder SUSE) als Betriebssystem zur Verfügung, der flexibel bezüglich der verwendbaren Linux Versionen ist. Linux-basierte Applikationen können auf dieser Plattform betrieben werden.

Darüber hinaus existiert als vierte Plattform eine LAMP Web-Applikations-Plattform. LAMP steht für Linux (Betriebssystem), Apache (Webserver), MySQL (Datenbank) und PHP (Skriptsprache). Diese Plattform ist ein Komplettpaket für Webapplikationen, die auf den genannten Techniken basieren, und ist ebenfalls sehr flexibel in den unterstützten Versionen.

Werkzeuge die auf einer Microsoft Infrastruktur basieren, werden bei der LTLS nur in Ausnahmefällen eingesetzt. Ausnahmefall bedeutet in diesem Zusammenhang, dass es keine nicht auf Microsoft Infrastruktur basierende Alternative für ein Werkzeug gibt. Da auf dem Markt für Testmanagementwerkzeuge viele verschiedene Werkzeuge existieren, die nicht auf Microsoft Infrastruktur setzen, spiegelt diese Anforderung ein K.O.-Kriterium wider. Ein Werkzeug, das die Verwendung einer Microsoft Infrastruktur (z. B. Microsoft SQL-Server) voraussetzt, muss somit nicht weiter betrachtet werden.

4. Welche Einschränkungen gibt es bei den Clients?

Auf der Client-Seite ist Komplexität durch den Applikationsbetrieb zu vermeiden. Um die PCs der Clients möglichst „schlank“ zu halten, sollen Installationen auf den Geräten der Clients vermieden werden und keine lokale Datenhaltung erfolgen. Da Applikationen mit Browserzugriff keine Installation von Client-Software voraussetzen, sind sie in dieser Hinsicht optimal. Browserbasierte Applikationen bieten jedoch noch weitere Vorteile. Sie sind orts- und plattformunabhängig über jedes Endgerät erreichbar, das über einen Browser und entsprechende Zugriffsberechtigungen verfügt.

Es ist anzumerken, dass der Betrieb von Applikationen, die komplexere Anforderungen an die Clients stellen, bei der LTLS generell möglich ist und in Erwägung gezogen wird. Die infrastrukturellen Bedingungen dieses Projektes sind lediglich mit dem IT Service Manager von UE/I, in Hinsicht auf das Vorhaben und die Größenordnung des Projektes „Einführung eines Testmanagementsystems“, abgestimmt und angepasst worden. Für die Betrachtungen zur Auswahl eines Testmanagementsystems wird die Anbindung von Clients über einen Browser als flexibelste und optimalste Lösung angesehen und stellt eine verbindliche Anforderung dar.

Die infrastrukturellen Bedingungen für den Betrieb eines neuen Werkzeuges bei UE/I werden nachfolgend noch einmal zusammengefasst.

- Verzicht auf SaaS-basierte Testmanagementlösungen
- Unterstützte Datenbanken sind MySQL und Oracle 11g (ab Version 11.2.0.2). Weiterhin sind Softwarelösungen mit integrierter Datenbank (eingebettet) möglich, weil somit keine Anforderungen an die Datenbanken-Infrastruktur existieren.
- Zugelassene Betriebssysteme sind Linux oder HP Unix (ab Version 11.23).
- Nicht zugelassen sind Applikationen, die auf einer Microsoft-Infrastruktur aufbauen (z. B. Microsoft SQL Server als Datenbank, Microsoft Windows Server als Betriebssystem).
- Zugelassen sind Applikationen bei denen die Clients die Verbindung über einen Browser herstellen.

4.2.4 Anforderungskatalog

„Ein Anforderungskatalog ist eine Auflistung aller funktionalen Anforderungen“ und stellt eine „komplette Beschreibung aller gewünschten und benötigten Funktionen und Fähigkeiten“ dar.¹⁰³

In den Kapiteln 4.2.1, 4.2.2 und 4.2.3 wurden funktionale Anforderungen an das Testmanagementwerkzeug ermittelt. Außerdem wurde die Anforderungsermittlung auf Basis der Testaktivitäten nur exemplarisch für die beiden Testplanungsphasen durchgeführt (siehe 4.2.1). Die Anwendung der dort gezeigten Methodik führte, ebenso für alle weiteren Testprozessphasen, zur Entstehung weiterer Anforderungen. Diese Anforderungen werden im Folgenden über das Kürzel (A.Z) referenziert. Werden die Ergebnisse aus den Kapiteln 4.2.1, 4.2.2 und 4.2.3 mit den Anforderungen (A.Z) zusammengefasst, so ergibt sich ein erster Entwurf des Anforderungskataloges.

Dieser Katalog ist jedoch nur als Zwischenergebnis zu betrachten, da eine Qualitätssicherung der ermittelten Anforderungen erfolgen musste, um zu gewährleisten, dass diese Anforderungen den Qualitätsvorstellungen genügen. Diese Erkenntnis kann mit folgendem Zitat von Pohl belegt werden: „An definierten Stellen des Prozesses ist es notwendig, die erreichte Qualität der entwickelten Anforderungen zu überprüfen“.¹⁰⁴

Die Qualitätssicherungsmaßnahmen konnten nur im Zusammensein mit Experten realisiert werden. Deshalb wurde ein weiterer Workshop mit dem Testmanager von UE/I und zwei Testern durchgeführt. Inhalt und Ziel des Workshops war zum einen die Validierung der einzelnen Anforderungen und die Prüfung des Kataloges auf Vollständigkeit – zum anderen wurden im Anschluss die einzelnen Anforderungen inhaltlich priorisiert.

Für die Gewichtung (Priorisierung) wurde zwischen Muss-, Soll- und Kann-Kriterien unterschieden. Muss-Kriterien „muss“ das Werkzeug in jedem Fall erfüllen. Soll-Kriterien sind wichtig, aber kein Ausscheidkriterium. Kann-Kriterien stellen eher einen netten Zusatznutzen (engl. *nice to have*) dar und besitzen die niedrigste Priorität.

Das Ergebnis des Workshops war somit ein geprüfter, vollständiger sowie um inhaltliche Prioritäten erweiterter Anforderungskatalog.

¹⁰³ Andler, 2012, S. 224.

¹⁰⁴ Pohl, 2008, S.419.

Anforderungen, die im Verlauf des Workshops ergänzt wurden, werden im Folgenden kurz dargestellt.

- (A.44) Das Testmanagementwerkzeug muss mehrere Projekte oder mehrere Mandanten unterstützen.
- (A.45) Das Testmanagementwerkzeug muss eine Benutzerverwaltung mit Rollen- und Berechtigungsmodell besitzen, sodass alle Benutzer über entsprechende Rechte verfügen und gleichzeitig mit dem Werkzeug arbeiten können.
- (A.46) Zur nicht-redundanten Pflege der Benutzer gibt es die Möglichkeit einer LDAP-Anbindung (engl. *Lightweight Directory Access Protocol*).
- (A.47) Zu den erfassten Releases sollen Zeitpläne hinterlegt werden können.
- (A.48) Anforderungen sollen anhand von Attributen klassifiziert werden können, sodass sie z. B. nach Typ oder Priorität gefiltert werden können.
- (A.49) Testfälle sollen anhand von Attributen klassifiziert werden können, sodass sie z. B. nach Typ oder Priorität filterbar sind.
- (A.50) In Prosa beschriebene Anforderungen sollen formatiert werden können.
- (A.51) Das Testmanagementsystem soll eine Suche nach Anforderungen bieten.
- (A.52) In Prosa beschriebene Testfälle sollen formatiert werden können.
- (A.53) Das Testmanagementsystem soll eine Suche nach Testfällen ermöglichen.
- (A.54) Das Testmanagementsystem soll eine Suche nach Abweichungen bieten.

Insgesamt konnten 70 Anforderungen ermittelt werden, die sinngemäß geordnet und im zentralen Dokument, dem Anforderungskatalog, zusammengefasst wurden. Diese 70 Anforderungen unterteilen sich in 17 „Muss-“, 39 „Soll-“ und 14 „Kann-Kriterien“. Die Anforderungen haben allesamt funktionalen Charakter und stellen somit funktionale Anforderungen an das Werkzeug. Um die Anforderungen zu kategorisieren, wurde der Katalog zunächst in Gruppen unterteilt. Da bei der Ermittlung der Anforderungen überwiegend auf die Testprozessaktivitäten der einzelnen Phasen Bezug genommen wurde, bilden diese die Gruppen des Anforderungskataloges. Die Anforderungen, die sich aus den Aktivitäten der *initialen Testplanung* und der *Testplanung auf Basis der Zwischenreleases* ergeben haben, wurden in einer Gruppe „Testplanung“ zusammengefasst. Weiterhin wurde die Gruppe „Systemeigenschaften“ in den Katalog

aufgenommen, die die allgemeinen und sonstigen Anforderungen an das Testmanagementwerkzeug beschreibt. Die folgenden fünf Anforderungsgruppen sind somit entstanden:

1. *Systemeigenschaften*
2. *Testplanung*
3. *Testanalyse und Design*
4. *Testdurchführung*
5. *Testauswertung und Bericht*

Tabelle B.3 (siehe Anhang) zeigt den vollständigen Anforderungskatalog nach Durchführung des Workshops. Im Anforderungskatalog enthalten sind alle validierten (geprüften und bestätigten) und zusätzlich gewonnenen Anforderungen, inklusive deren inhaltlicher Priorisierungen.

Außerdem werden alle in den Kapiteln 4.2.1, 4.2.2 und 4.2.3 ermittelten Anforderungen durch ihre fortlaufende Nummer im Anforderungskatalog identifiziert. Zusätzlich wurde jede Anforderung mit einer Identifikationsnummer (ID) versehen, die die Anforderungen für die weiteren Betrachtungen eindeutig identifiziert.

4.3 Evaluierungsprozess

Auf Grundlage des in Kapitel 4.2.4 entwickelten Anforderungskataloges konnte die Evaluierung der Testmanagementsysteme durchgeführt werden. Es wurde zunächst eine Marktrecherche mit erster Eingrenzung der Werkzeuge durchgeführt, woraufhin eine weitere Vorauswahl die wenigen relevanten Werkzeuge identifizierte.

4.3.1 Marktrecherche und erste Filterung

In diesem Kapitel werden das Vorgehen und die Ergebnisse der Marktrecherche beschrieben. Die Schaffung des Marktüberblicks ging mit einer frühzeitigen Filterung der Werkzeuge einher, um die Vielzahl der am Markt verfügbaren Testmanagementwerkzeuge auf eine überschaubare Menge zu reduzieren.

Auf allgemeiner Ebene ist feststellbar, dass der Funktionsumfang, das Nutzungskonzept und die Philosophie der Produkthersteller meist sehr unterschiedlich sind. Weiterhin ist

zwischen kommerziellen Produkten und Open-Source Produkten zu unterscheiden. Open-Source Produkte sind, im Gegensatz zu kommerziellen Produkten, frei am Markt verfügbare Produkte, die keine Lizenz- und Nutzungskosten verursachen und deren Programmcode (Quellcode) offengelegt ist.¹⁰⁵ Auch wenn davon ausgegangen werden kann, dass Open-Source Software nicht an den Funktionsumfang und die Professionalität von kommerzieller Software heranreicht, sollen diese zunächst nicht aus den Betrachtungen ausgeschlossen werden.

Um einen Überblick über die am Markt verfügbaren Werkzeuge zu erhalten, wurde eine ausgiebige Internetrecherche betrieben. Einige Organisationen und Interessengruppen, die sich mit dem Thema Softwaretest befassen, stellen im Internet Übersichten zu den am Markt verfügbaren Testmanagementwerkzeugen zur Verfügung. Zur Marktsichtung wurden mehrere dieser Auflistungen genutzt. Einige Auflistungen stellten dabei sehr bedeutende Quellen für die Marktrecherche dar. Eine eher kleine Werkzeugübersicht auf *imbus.de*¹⁰⁶ bezieht sich auf rein kommerziell vertriebene Softwareprodukte, wogegen die Zusammenstellung auf *opensourcetesting.de*¹⁰⁷ eine Auflistung von frei verfügbaren Werkzeugen darstellt. Auf *xqual.com*¹⁰⁸ wird eine sehr umfangreiche Auflistung zur Verfügung gestellt, die den Großteil der am Markt verfügbaren Werkzeuge beinhaltet. Dort wird bereits in frei verfügbare und kommerzielle Produkte unterschieden, sowie ein tabellarischer Überblick über die groben Funktionsbereiche jedes Werkzeuges gegeben. Weiterhin wurde eine Studie über Testmanagementwerkzeuge aus dem Jahr 2010 in die Betrachtungen einbezogen. In dieser Studie von *Software Quality Lab*¹⁰⁹ wurde eine Produktevaluierung mit neun Testmanagementwerkzeugen durchgeführt. Deshalb wurden zusätzlich alle in der Studie betrachteten Werkzeuge in die Untersuchungen aufgenommen. Darüber hinaus konnten weitere Testmanagementwerkzeuge durch intensive Internetrecherche ausfindig gemacht werden.

Um vorab eine Filterung der am Markt verfügbaren Produkte durchzuführen, galt es bei der Recherche eine erste Sichtprüfung der Produkte vorzunehmen, bevor diese in die Menge der relevanten Werkzeuge aufgenommen werden konnten. Dies war überwiegend bei den frei verfügbaren Produkten (Open-Source) notwendig, da sie in sehr großer Anzahl am Markt zur Verfügung stehen. Die Recherche und Sichtprüfung wurde durch Marktrecherchefragen begleitet, die als Kontrollpunkte zum frühzeitigen Aussortieren der Produkte dienten.

¹⁰⁵ Vgl. Buxmann, 2008, S. 185f.

¹⁰⁶ Vgl. http://www.imbus.de/tool-liste/#cat_11 (letzter Zugriff: 17.07.2012).

¹⁰⁷ Vgl. <http://opensourcetesting.org/testmgt.php> (letzter Zugriff 17.07.2012).

¹⁰⁸ Vgl. <http://www.xqual.com/qa/tools.html> (letzter Zugriff 18.07.2012).

¹⁰⁹ Für mehr Informationen siehe (<http://www.software-quality-lab.com>).

Nachfolgend werden die Marktrecherchefragen und deren inhaltliche Kernpunkte genannt.

Wie ist der Hersteller aufgestellt?

- Erfahrung und Reputation des Herstellers
- Größe und Aktivität der Community (entscheidend bei Open-Source Produkten)
- Produktpflege und Änderungen (Intensivität und Häufigkeit der Produktverbesserungen)

Welchen Grad an Professionalität vermittelt der erste Eindruck der Software?

- Eindrücke über die Webseite des Herstellers (z. B. durch Beschreibungen und Grafiken)
- Begutachten von Screenshots und Demovideos über die Werkzeuge

Wie ist die Produktdokumentation gestaltet?

- Zugänglichkeit
- Grad der Ausführlichkeit

Diese Fragen konnten auf Grundlage der frei verfügbaren Herstellerinformationen und durch gezielte Internetrecherche beantwortet werden. Beispielsweise wurde ein Open-Source Werkzeug, das über keine große und aktive Community verfügt und nicht kontinuierlich gepflegt und verbessert wird, aus den weiteren Betrachtungen ausgeschlossen. Die Wahrscheinlichkeit, dass ein Werkzeug, hinter dem eine starke Community steht, weiterentwickelt und gepflegt wird, ist in der Regel weitaus höher als bei Werkzeugen mit geringerem Beliebtheits- und Verbreitungsgrad.

Neben der Beantwortung der oben stehenden Marktrecherchefragen, wurden bei der Produktrecherche weitere Einschränkungskriterien ausgewählt, um eine noch fundiertere und frühzeitige Filterung der Werkzeuge vorzunehmen. In Anlehnung an die Funktionsübersicht auf *xqual.com*¹¹⁰ und durch Abstimmung mit den groben Funktionsbereichen des Anforderungskataloges, wurden folgende vier allgemeine Filterkriterien der Marktrecherche aufgestellt.

¹¹⁰ Vgl. <http://www.xqual.com/qa/tools.html> (letzter Zugriff am 18.07.2012).

Der allgemeine Funktionsumfang des Werkzeugs beinhaltet Funktionen...

1. ...des Anforderungsmanagements.
2. ...zur Erstellung und zum Umgang mit Testplänen (Element der Testplanung).
3. ...zur Erstellung von Auswertungen (Metriken) und Berichten über das Testprojekt und den Testprozess.
4. ...zur Unterstützung der manuellen Testdurchführung.

Zusätzlich wurde die in Kapitel 4.2.3 ermittelte Anforderung (A.42) als fünftes Kriterium der Marktrecherche gewählt.

Demnach bietet das Werkzeug Möglichkeiten...

5. ...zur Anbindung bzw. Integration an das Fehlermanagementsystem JIRA.

Es ist außerdem zu beachten, dass viele Hersteller von Testmanagementwerkzeugen eine unterschiedliche Auffassung über die Benennung, Inhalt und Sinnhaftigkeit der Elemente des Testprozesses und der Arbeitsabläufe innerhalb des Testprozesses besitzen. Dies wird in Punkt drei deutlich, denn nicht jedes Werkzeug verwendet den Begriff eines Testplanes im gleichen Zusammenhang oder verwendet andere Begrifflichkeiten. In einer einfachen und allgemeinen Form ist ein Testplan ein Objekt, dem auszuführende Testfälle für eine bestimmte Testintention bzw. für einen bestimmten Testinhalt, zugewiesen werden können. Aufgrund unterschiedlicher Ansichten der Hersteller über den Testprozess und die Arbeitsabläufe, unterscheiden sich jedoch viele Werkzeuge in der Umsetzung dieser und weiterer Beziehungen.

Anhand der auf *xqual.com* zur Verfügung gestellten Funktionsliste konnte bereits ein Großteil der ersten vier Filterkriterien für die einzelnen Werkzeuge bewertet werden. Es hat sich dennoch herausgestellt, dass die Auflistung unvollständig ist und nicht alle Angaben dem aktuellen Stand der Werkzeuge entsprachen, da einige Hersteller ihre Produkte bereits weiterentwickelt hatten. Die Betrachtung der Funktionsliste auf *xqual.com* ging deshalb mit der kontinuierlichen Überprüfung von aktuellen Herstellerangaben einher. Für die allgemeine Überprüfung, ob die Werkzeuge eine JIRA Integration bzw. Anbindung bieten, wurde ebenfalls nach Herstellerangaben recherchiert – wenn diese jedoch keine Informationen bereitstellten, musste eine weitere und gezielte Internetrecherche betrieben werden.

Nach ausgiebiger Prüfung vieler am Markt verfügbarer Produkte hat sich der Bestand vor allem bei den Open-Source Produkten erheblich reduziert. Nur ein Open-Source

Produkt – *TestLink* – konnte die Kriterien der Marktrecherche erfüllen und hinterließ bei der Beantwortung der Marktrecherchefragen einen zumindest mittelmäßigen Eindruck. Alle weiteren Open-Source Produkte konnten die Funktionsgruppen nicht abbilden oder hinterließen durch die Beantwortung der Marktrecherchefragen kein positives Ergebnis.

Insgesamt konnten 17 Testmanagementwerkzeuge als hinreichend verwendbar identifiziert werden. Zusätzlich wurden acht bekannte Werkzeuge, die die Kriterien nicht erfüllen konnten, in das Ergebnis der Marktübersicht einbezogen. Es soll exemplarisch gezeigt werden, aufgrund welcher Marktrecherche-Kriterien diese Werkzeuge nicht in die weiteren Betrachtungen aufgenommen wurden. Sie dienen somit lediglich dazu, einen Einblick über die Anwendung der fünf Filterkriterien zu geben. Ein Großteil der Werkzeuge musste aussortiert werden, da sie keine Funktionen zum Anforderungsmanagement boten.

Für die Ergebnismatrix wurden verschiedene Abkürzungen und Zeichen (Symbole) verwendet. Die folgende Übersicht fasst alle verwendeten Symbole und Abkürzungen, sowie deren Bedeutungen zusammen.

AM	Anforderungsmanagement
TP	Testpläne
BE	Berichterstattung
MT	Manuelle Testdurchführung
JIRA	JIRA-Integrationsfähigkeit
✓	Kriterium erfüllt
✓✘	Kriterium nicht direkt erfüllt; Funktion über nicht erwünschte Umwege realisierbar
✘	Kriterium nicht erfüllt
?	Erfüllung des Kriteriums ungewiss

Quelle: Eigene Darstellung

Tab. 4.4: Marktrecherche: verwendete Symbole und Abkürzungen

Tabelle 4.5 zeigt das Ergebnis der Marktrecherche unter Anwendung einer ersten Filterung, die während der Marktsichtung vorgenommen wurde und aus der Beantwortung der Marktrecherchefragen, sowie der Prüfung der fünf Marktrecherche-Kriterien, bestand.

Allgemeine Produktinformationen				Marktrecherche-Kriterien					Ergebnis
Nr.	Werkzeug	Hersteller	Herstellerlink	AM	TP	BE	MT	JIRA	
Kommerzielle Produkte									
1	ApTest Manager	ApTest	www.aptest.com	✓	✓	✓	✓	✓	✓
2	Aqua	Andagon	www.andagon.com	✓	✓	✓	✓	✓	✓
3	Enterprise Tester	Catch Limited	www.enterprisetester.com	✓	✓	✓	✓	✓	✓
4	TestBench	Imbus AG	www.imbus.de	✗	✓	✓	✓	✓	✗
5	Orcanos QPack ALM-Suite	Orcanos	www.orcanos.com	✓	✓	✓	✓	✗	✗
6	Polarion QA	Polarion	www.polarion.com	✓	✓	✓	✓	✓✗	✗
7	PractiTest	PractiTest	www.practitest.com	✓	✓	✓	✓	✓	✓
8	QABook	NMQA	www.nmqa.com	✓	?	✓	✓	✗	✗
9	QAComplete	SmartBear	www.smartbear.com	✓	✓	✓	✓	✓	✓
10	Quality Center	HP	www.hp.com	✓	✓	✓	✓	✓✗	✗
11	Rational Clear Quest	IBM	www.ibm.com	✓	✓	✓	✓	✗	✗
12	SilkCentral Test Manager	Micro Focus (Borland)	www.microfocus.com	✓	✓	✓	✓	✓	✓
13	SpiraTest	Inflectra	www.inflectra.com	✓	?	✓	✓	✓	✓
14	TEMPPO (SITEMPOO)	Siemens	http://www.siemens.com	✓	✓	✓	✓	✓	✓
15	TEST® Professional Suite	SQS	www.sqs.com	✓	✓	✓	✓	✓	✓
16	TestRail	Gurock Software	www.gurock.com	✗	✓	✓	✓	✓	✗
17	Testuff	Testuff Ltd.	www.testuff.com	✓	✓	✓	✓	✓	✓
18	TestWave	TestWave	www.testwave.co.uk	✓	✓	✓	✓	?	✓
19	TOMOS	TOMOS	www.reachsimplicity.com	✓	✓	✓	✓	?	✓
20	TOSCA Testsuite	TRICENTIS	www.tricentis.com	✓	✓	✓	✓	?	✓
21	Total Test Management (Enterprise)	Qmetry	www.qmetry.com	✓	✓	✓	✓	✓	✓
22	Xstudio Enterprise	Xqual	www.xqual.com	✓	✓	✓	✓	✓	✓
23	Zephyr Enterprise Edition	Zephyr	www.getzephyr.com	✓	✓	✓	✓	✓	✓
Open-Source Produkte									
24	TestLink	TestLink	www.teamst.org	✓	✓	✓	✓	✓	✓
25	Testopia	Mozilla	www.mozilla.org	✗	✓	✓	✓	✗	✗

Quelle: Eigene Darstellung

Tab. 4.5: Marktrecherche: Ergebnis der frühzeitigen Filterung

Marktrecherche-Kriterien, deren Erfüllung aufgrund von fehlenden Informationen nicht eindeutig bewertet werden konnte, wurden zunächst mit einem Fragezeichen versehen.

Waren alle weiteren Kriterien erfüllt, wurden die Zeilenergebnisse temporär mit dem Status „Kriterium erfüllt“ gekennzeichnet. Darüber hinaus wurden sie in die weiteren Betrachtungen einbezogen, da noch weitere Vorauswahlschritte durchgeführt wurden. Bei zwei sehr bekannten und etablierten Werkzeugen – „HP Quality Center“ und „Polarion QA“ – ist eine JIRA-Integration nur über separate und lizenzgebührenpflichtige Erweiterungen (Plugins) eines Drittanbieters bzw. nur über zusätzliche und kostenintensive Integrationsplattformen möglich. Alle weiteren Werkzeuge werden bereits mit lauffähigen JIRA-Integrationsmöglichkeiten (out-of-the-box) ausgeliefert. Die Anbindung erfolgt dabei über die Schnittstelle (API – *engl.* application programming interface) des Werkzeuges oder ein eigenes Plugin des Herstellers. Aufgrund der genannten Vorteile der anderen Hersteller gegenüber den Produkten von HP und Polarion, wurde das „HP Quality Center“ und „Polarion QA“ nicht in die weiteren Betrachtungen einbezogen.

4.3.2 Weitere Vorauswahl der Werkzeuge

Aufbauend auf den Ergebnissen der Marktrecherche (siehe Kapitel 4.3.1) erfolgte eine weitere Reduzierung der 17 ausgewählten Werkzeuge. Dazu mussten die Testmanagementsysteme einen zweistufigen Vorauswahlprozess durchlaufen, der aus folgenden Schritten bzw. Stufen bestand.

1. Vorauswahl durch Prüfung infrastruktureller Einschränkungen
2. Vorauswahl durch Prüfung der 17 Muss-Kriterien des Anforderungskataloges

Die erste Vorauswahlstufe basierte auf den Informationen, die in Kapitel 4.2.3 ermittelten harten Anforderungen, bezüglich der infrastrukturellen Einschränkungen beim Einsatz eines Testmanagementwerkzeuges bei UE/I. Innerhalb der zweiten Vorauswahlstufe wurden alle 17 Muss-Kriterien des Anforderungskataloges (siehe Tab. B.3) vorab auf Erfüllung geprüft. Ein Werkzeug, das ein Muss-Kriterium nicht erfüllen konnte, durfte nicht weiter betrachtet werden.

Um an die erforderlichen Informationen zur Bewertung der Vorauswahlkriterien zu gelangen, wurden zwei wesentliche Tätigkeiten zur Informationsbeschaffung durchgeführt:

- Lesen von Dokumentationen und Beschreibungen der Werkzeughersteller
- Gezielte Gespräche und Schriftverkehr mit dem Vertrieb und der Kundenbetreuung der Produkthersteller

Die nachfolgende Tabelle zeigt das Ergebnis der ersten Vorauswahlstufe unter Weiterverwendung der in Kapitel 4.3.1 (siehe Tab. 4.4) definierten Symbole.

		Vorauswahl-Kriterien Stufe 1								Ergebnis
Nr.	Werkzeug	Eigenbetrieb (E) / SaaS (S)		Datenbank (Oracle 11g oder MySQL)		Server-Betriebssystem (Linux, HP Unix)		Browser-zugriff?		
Kommerzielle Produkte										
1	ApTest Manager	✓	E	-	eingebettet	✓	Linux / Unix	✓	Browser	✓
2	Aqua	✓	E / S	✓	Oracle	✗	Microsoft	✗		✗
3	Enterprise Tester	✓	E	✓	Oracle / MySQL	✗	Microsoft	✓	Browser	✗
4	PractiTest	✗	S	-		-		✓	Browser	✗
5	QAComplete	✓	E / S	✗	Microsoft	✗	Microsoft	✓	Browser	✗
6	SilkCentral Test Manager	✓	E	✓	Oracle	✗	Microsoft	✗		✗
7	SpiraTest	✓	E / S	✗	Microsoft	✗	Microsoft	✓	Browser	✗
8	TEMPPO (SiTEMPOO)	✓	E	✓	Oracle	✗	Microsoft	✗		✗
9	TEST® Professional Suite	✓	E	-	eingebettet	✗	Microsoft	?		✗
10	Testuff	✗	S	-		-		✗		✗
11	TestWave	✗	S	-		-		✓	Browser	✗
12	TOMOS	✗	S	-		-		✓	Browser	✗
13	TOSCA Testsuite	✓	E	✗	Oracle 10g	✗	Microsoft	✗		✗
14	Qmetry Total Test Management (Enterprise)	✓	E / S	✓	MySQL	✓	Linux	✓	Browser	✓
15	Xstudio Enterprise	✓	E / S	✓	Oracle / MySQL	✓	Linux	✗		✗
16	Zephyr Enterprise Edition	✓	E / S	✓	MySQL	✓	Linux	✓	Browser	✓
Open-Source Produkte										
17	TestLink	✓	E	✓	Oracle / MySQL	✓	Linux	✓	Browser	✓

Quelle: Eigene Darstellung

Tab. 4.6: Ergebnis – erste Vorauswahlstufe

Zu erkennen ist, dass viele Testmanagementwerkzeuge, aufgrund des Voraussetzens von Microsoft-Infrastruktur auf Serverseite, negativ bewertet werden mussten. Weiterhin mussten vier Werkzeuge aussortiert werden, da sie allein über das Cloud-basierte Nutzungskonzept „SaaS“ bezogen werden können. Insgesamt mussten 13 von

17 Testmanagementwerkzeuge negativ bewertet werden, weshalb nur vier Werkzeuge die erste Vorauswahlstufe bestehen konnten.

Die vier übrig gebliebenen Testmanagementwerkzeuge mussten anschließend die zweite Vorauswahlstufe durchlaufen und die 17 Muss-Kriterien des Anforderungskataloges erfüllen. Tabelle 4.7 zeigt das Ergebnis der zweiten Vorauswahlstufe.

Muss-Kriterien des Anforderungskataloges		Werkzeuge			
ID	Anforderung	ApTest	Qmetry	TestLink	Zephyr
1	Das Testmanagementsystem muss mehrere Projekte oder mehrere Mandanten unterstützen.	✓	✓	✓	✓
2	Das Testmanagementwerkzeug muss eine Benutzerverwaltung mit Rollen- und Berechtigungsmodell besitzen, sodass alle Benutzer über entsprechende Rechte verfügen und gleichzeitig mit dem Werkzeug arbeiten können.	✓	✓	✓	✓
3	Das Testmanagementsystem muss Möglichkeiten zur Anbindung bzw. Integration an JIRA besitzen.	✓	✓	✓	✓
6	Anforderungen müssen verwaltet werden können (Anlegen, Ändern, Löschen).	✓	✓	✓	✓
9	Anforderungen müssen in Prosa beschrieben werden können.	✓	✓	✓	✓
17	Anforderungen müssen einem Release zugeordnet werden können.	✓	✓	✗	✓
18	Mit dem Testmanagementsystem müssen Releases definiert werden können.	✓	✓	✓	✓
22	Für ein Release / Zwischenrelease müssen mehrere Testpläne angelegt werden können.	✓	✓	✗	✓
24	Je Testplan müssen bereitgestellte Anforderungen bzw. Testfälle, die die Anforderungen prüfen, sowie Testfälle für den Re-Test ausgewählt und zugeordnet werden können.	✓	✓	✓	✓
28	Testfälle müssen verwaltet werden können (Anlegen, Ändern, Löschen).	✓	✓	✓	✓
29	Testfälle müssen strukturiert und gruppiert werden können, um diese z. B. nach Zweck, Teststufe oder Testart ordnen zu können (Gruppierungsobjekte, z. B. in Ordern oder einer Baumstruktur).	✓	✓	✓	✓
32	Ein Testfall muss mehreren Testplänen zugeordnet werden können.	✓	✓	✓	
33	Jeder Testfall muss mehreren Anforderungen zugeordnet werden können.	✓	✓	✓	✓
34	Testfälle müssen in Prosa beschrieben werden können.	✓	✓	✓	✓
37	An den Testfällen muss ein Prozessstatus wählbar oder definierbar sein (z. B. NEW, DESIGN, READY, COMMITTED).	✓	✓	✓	✓
45	Abweichungen müssen im Testmanagementsystem angelegt, in Prosa beschrieben und mit Eigenschaften versehen werden können.	✓	✓	✓	✓
63	Wie ist der aktuelle Status der eingeplanten Testfälle?	✓	✓	✓	✓

Quelle: Eigene Darstellung

Tab. 4.7: Ergebnis – zweite Vorauswahlstufe

TestLink kann den Testprozess und das damit verbundene Vorgehen bei den Testaktivitäten nicht abbilden. Bei UE/I werden Testpläne für ein Release bzw. Zwischenrelease angelegt, wogegen in TestLink mehrere Releases für einen Testplan angelegt werden können. Die Testplanung bei UE/I findet auf Basis der Releases und Zwischenreleases statt, wobei ausgewählte Anforderungen für jedes Zwischenrelease getestet werden. In TestLink können den Release-Objekten keine Anforderungen zugeordnet werden, weshalb sich die gewünschte Struktur nicht abbilden lässt.

Den gesamten zweistufigen Vorauswahlprozess konnten nur drei von 17 Testmanagementwerkzeugen bestehen. Die Testmanagementwerkzeuge „ApTest Manager“ – Version 2.25.01.02, „Qmetry Total Test Management (Enterprise)“ in der Version 5.2 und „Zephyr Enterprise“ Version 4.0 waren somit die einzigen Werkzeuge, die die Vorauswahlkriterien erfüllten und detailliert auf alle 70 Anforderungen geprüft werden konnten.

4.3.3 Bewertung des Anforderungskataloges

Um eine mathematische Bewertungsgrundlage anhand der inhaltlich priorisierten Anforderungen (siehe Tab. B.3) zu schaffen, musste zunächst ein äquivalenter Zahlenwert für die Prioritäten ermittelt werden. Da ein Muss-Kriterium die höchste Priorität besitzt, muss es durch einen höheren Zahlenwert repräsentiert werden als ein Soll-Kriterium. Ebenso muss ein Soll-Kriterium eine höhere Zahlengewichtung besitzen als ein Kann-Kriterium. Die logische Abstufung der Prioritäten musste sich entsprechend in den festgelegten Zahlenwerten widerspiegeln. Bei der genauen Differenzierung der Zahlenwerte ist jedoch ein hoher Freiheitsgrad gegeben. Für die Betrachtungen dieser Arbeit wurde von dem einfachen Fall ausgegangen, dass ein Soll-Kriterium doppelt so wichtig ist wie ein Kann-Kriterium. Analog dazu wurde ein Muss-Kriterium als doppelt so wichtig wie ein Soll-Kriterium interpretiert. Dennoch sollte ein Muss-Kriterium durch seine bindende Eigenschaft geringfügig von diesem Verhältnis abweichen. Weiterhin wurden Zahlenwerte kleiner gleich zehn gewählt, um überschaubare Zahlenauswertungen zu bekommen. Die festgelegten Gewichtungen der gegebenen Anforderungsprioritäten (Wichtigkeit des Kriteriums) sind in Tabelle 4.8 dargestellt.

Wichtigkeit des Kriteriums	Gewichtung
Muss	10
Soll	6
Kann	3

Quelle: Eigene Darstellung

Tab. 4.8: Gewichtungen nach Prioritäten der Bewertungskriterien

Für einen mathematisch fundierten Vergleich und die damit verbundene Bewertung der Testmanagementsysteme, musste eine weitere mathematische Bewertungsgrundlage zur Messung des Erfüllungsgrades der Anforderungen geschaffen werden. Eine typische Methode ist die Verwendung des Schulnotensystems zur Bewertung des Erfüllungsgrades von „sehr gut“ bis „ungenügend“, erweitert um die Möglichkeit der Nichterfüllung, was einer Punkteskala von null bis fünf entspricht. Die alleinige Verwendung dieser allgemeinen Skala verleiht den Evaluierungsergebnissen jedoch einen sehr subjektiven Charakter, da die evaluierende Person eine subjektive Einschätzung des Erfüllungsgrades jeder Anforderung vornimmt.

Deshalb mussten für möglichst viele Anforderungen hinreichend objektive Bewertungs- und Messkriterien ermittelt werden, die eine entsprechende Punktevergabe zur Bewertung des Erfüllungsgrades rechtfertigen. Als Grundlage für die Bewertung anhand dieser Bewertungskriterien dienten zwei Skalenniveaus. Es wurde zwischen einem

zwei-stufigem Erfüllungsgrad – „erfüllt“ und „nicht erfüllt“ – mittels einer Nominalskala und einem vier-stufigen Erfüllungsgrad – „gut“, „mittel“, „schlecht“ und „nicht erfüllt“ – mittels einer Ordinalskala, unterschieden. Die zwei-stufige Nominalskala und die vier-stufige Ordinalskala sind in Tabelle 4.9 in einer Skalenmatrix dargestellt.

Erfüllungsgrad (2-stufig)	erfüllt	-	-	nicht erfüllt
Erfüllungsgrad (4-stufig)	gut	mittel	schlecht	nicht erfüllt
Bewertungspunkte	3	2	1	0

Quelle: Eigene Darstellung

Tab. 4.9: Nominal- und Ordinalskala als Bewertungsmaßstäbe

Viele Anforderungen sind sehr genau und ohne großen Interpretationsspielraum formuliert worden, sodass ein Großteil der Anforderungen nur auf die Zustände „erfüllt“ und „nicht erfüllt“ untersucht werden konnte. Sie stellen folglich rein objektiv bewertbare Anforderungen dar, es ist jedoch schwierig eine nähere Unterscheidung zwischen den zu bewertenden Werkzeugen vorzunehmen. Alle inhaltlich näher differenzierbaren Anforderungen vermindern den Grad an Subjektivität durch die Verwendung von Bewertungs- bzw. Messkriterien. Eine zunächst subjektive Ordinalskala kann somit messbarer und nachvollziehbarer verwendet werden. Die Schwierigkeit bei der Ermittlung der Kriterien bestand darin, dass sie aufgestellt werden mussten, ohne dass tiefgründiges Wissen über die Funktions- und Umsetzungsmöglichkeiten im Bereich von Testmanagementwerkzeugen vorlag. Außerdem konnte nicht in jedem Fall für alle Abstufungen der Ordinalskala ein Messkriterium ermittelt werden. So wurde in einigen Fällen beispielsweise nur zwischen „gut“ und „schlecht“ oder „gut“ und „mittel“ unterschieden. Für einige Anforderungen, für die keine detaillierten Bewertungs- und Messkriterien gefunden werden konnten, wurde dennoch die allgemeine Ordinalskala zur Bewertung verwendet. Obgleich dies eine erhöhte Subjektivität zur Folge hatte, konnte somit eine differenziertere Bewertung, im Gegensatz zur Verwendung der Nominalskala, vorgenommen werden.

Die nachfolgende Tabelle listet detailliert alle Anforderungen der Tabelle B.3 auf, ordnet diese den jeweils verwendeten Skalen zu, und gibt Aufschluss über die Häufigkeit der Verwendung der beiden Skalen. Bezüglich der Ordinalskala wird zusätzlich zwischen der Verwendung der allgemeinen, subjektiveren Form und der objektiveren Form, durch Hinzuziehen von Bewertungs- und Messkriterien, unterschieden.

	Anforderungen nach ID	Anzahl	in %
Nominalskala (2-stufig)	1, 4, 5, 6, 9, 13, 15, 17, 22, 24, 27, 28, 31, 32, 33, 34, 38, 40, 42, 48, 49, 50, 67, 68, 70	25	35,71%
Ordinalskala (4-stufig) allgemein	10, 11, 35, 36, 47, 51, 69	7	10,00%
Ordinalskala (4-stufig) mit Messkriterien	2, 3, 7, 8, 12, 14, 16, 18, 19, 20, 21, 23, 25, 26, 29, 30, 37, 39, 41, 43, 44, 45, 46, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66	38	54,29%

Quelle: Eigene Darstellung

Tab. 4.10: Nutzungsverteilung der verwendeten Skalen

Die Tabelle zeigt, dass für rund 64% der Anforderungen die Ordinalskala verwendet wurde, um eine differenziertere Unterscheidung der Werkzeuge zu ermöglichen. Für 54% aller Anforderungen konnten Bewertungs- und Messkriterien aufgestellt werden, um den Grad an Subjektivität zu vermindern. Bei etwa 36% der Anforderungen wurde die Bewertung über die Nominalskala vorgenommen, wogegen nur zehn Prozent aller Anforderungen mittels der allgemeinen Ordinalskala bewertet wurden.

Im Folgenden wird beispielhaft auf ausgewählte Anforderungen des Anforderungskataloges eingegangen, um das Vorgehen der weiteren Detaillierung in Hinsicht auf den Erfüllungsgrad zu beschreiben. Es wird jeweils ein Beispiel für die zwei, in Tabelle 4.10 dargestellten, Verwendungsarten der Ordinalskala gegeben. Dabei werden der Bewertungskontext und, soweit gegeben, die Messkriterien beschrieben.

In Bezug auf die Anforderung drei (siehe Tab. B.3, ID=3) konnte die Ordinalskala, durch weitere Detaillierung mittels geeigneter Messkriterien, verwendet werden. Als Bewertungskontext wurde näher nach der „Art der Integration“, als Maßstab für die Integrationstiefe und die Kommunikationsrichtungen, unterschieden. Bietet das Werkzeug eine vollautomatische und bidirektionale Synchronisation der Abweichungsdaten, so wurde das Werkzeug mit drei Punkten (3) als „gut“ bewertet. Bidirektional bedeutet in diesem Zusammenhang, dass die Datenübertragung in beide Richtungen, das heißt in Richtung des Fehlermanagementsystems und zurück in Richtung des Testmanagementsystems, stattfindet. Für den Fall, dass die bidirektionale Synchronisation nicht vollautomatisch geschieht und eine Form von manueller Aktualisierung notwendig ist, wurde das Werkzeug mit zwei Punkten (2) als „mittel“ bewertet. Im „schlechtesten“ Fall bietet das Testmanagementwerkzeug nur eine unidirektionale Synchronisation in Richtung des Fehlermanagementsystems, wobei die Abweichungsdaten nur in diese Richtung übertragen werden und kein Rückkanal

besteht. Das Werkzeug wurde unter diesen Voraussetzungen nur mit einem Punkt (1) bewertet. Der Fall, dass ein Werkzeug die Anforderung nicht erfüllt, wurde bereits in der Vorauswahl ausgeschlossen, da alle drei geprüften Werkzeuge die Muss-Kriterien erfüllen „mussten“. Zur Vollständigkeit der Skala wurde die Bewertungsstufe „nicht erfüllt“ (0) jedoch mit festgehalten.

Die allgemeine und subjektivere Ordinalskala wurde beispielsweise für die Anforderung 36 (siehe Tab. B.3, ID=36) genutzt. Bei der Suche nach Testfällen wurden die Werkzeuge nach dem „Umfang der Filtermöglichkeiten“ differenziert, da dies ein Indikator für die Präzision und Anpassbarkeit der Suche darstellt. Dabei musste der Umfang der Filtermöglichkeiten auf der allgemeinen Skala als „gut“ (3), „mittel“ (2) oder „schlecht“ (1) eingeschätzt werden. Weiterhin bestand die Möglichkeit, dass ein Werkzeug keine Filtermöglichkeiten bietet, was mit null (0) Punkten bewertet wurde.

Der Anforderungskatalog (siehe Tab. B.3) konnte anhand der zwei Skalen (siehe Tab. 4.9) und der gewichteten Anforderungsprioritäten (siehe Tab. 4.8) vollständig und für alle drei vorausgewählten Testmanagementwerkzeuge (siehe Kapitel 4.3.2) bewertet werden. Der vollständig bewertete Anforderungskatalog inklusive aller Anforderungen, Bewertungsmaßstäbe und Einzelbewertungen, ist in Tabelle B.4 im Anhang dargestellt. Für jeden der fünf Bereiche des Anforderungskataloges wurde ein Teilergebnis berechnet. Die nachfolgende Formel ist für die Berechnung der Teilergebnisse zuständig.

$$TE = \sum_i^j P \times E$$

Ein Teilergebnis (TE) besteht aus der Summe der Zeilenprodukte. Ein Zeilenprodukt ist dabei der Erfüllungsgrad eines Kriteriums (E) im Punktebereich null bis drei, multipliziert mit der Zahlengewichtung (P) des Kriteriums. Die Indizes (i) und (j) kennzeichnen den Bereich über den summiert werden soll, wobei (i) die erste Anforderung des Bereiches und (j) die letzte Anforderung des betrachteten Bereiches darstellt.

Ausgehend von den ermittelten Teilergebnissen konnte die Gesamtzahl der erreichten Punkte für jedes Werkzeug berechnet werden. Dazu musste lediglich die Summe der Teilergebnisse gebildet werden. Gemessen werden die Ergebnisse am Maximalwert der zu erreichenden Punkte, der sowohl für die Teilbereiche, als auch für die Menge aller Anforderungen berechnet wurde. Der Maximalwert eines Teilbereiches lässt sich ebenso mit der obenstehenden Formel berechnen, jedoch ist der Erfüllungsgrad stets drei und somit maximal. Im Maximum konnten 1338 Punkte erreicht werden. Die nachfolgende Tabelle zeigt das detaillierte Ergebnis der Werkzeugbewertung.

	ApTest	Qmetry	Zephyr
Erfüllte Kriterien	56	66	56
Nicht erfüllte Kriterien	14	4	14
Erreichbare Punktezahl	1338	1338	1338
Erreichte Punktzahl	981	1191	1020
Erreichte Punktzahl in %	73,32%	89,01%	76,23%

Quelle: Eigene Darstellung

Tab. 4.11: Ergebnis der Bewertung des Anforderungskataloges

Das Testmanagementsystem „Qmetry Total Test Management (Enterprise) – Version 5.2“ hat mit 1191 von 1338 Punkten (89%) die höchste Punktzahl erreicht. Das „Zephyr Enterprise – Version 4.0“ erreichte 76% der maximal erreichbaren Punkte und hat damit rund drei Prozent besser als „ApTest Manager – Version 2.25.01.02“ abgeschnitten. Qmetry konnte sich in der Gesamtbewertung deutlich von Zephyr und ApTest absetzen, was sich in einem Unterschied von rund dreizehn Prozent auf Zephyr widerspiegelt. Außerdem konnte es zehn Kriterien mehr erfüllen als ApTest und Zephyr. Durch eine hohe Flexibilität und gute Konfigurierbarkeit konnte sich Qmetry insbesondere im Bereich *Testauswertung und Bericht* von Zephyr und ApTest absetzen. Da der Großteil der Anforderungen dieses Bereiches aus den TPI-Maßnahmen (siehe Kapitel 4.2.2) abgeleitet wurde, liefert Qmetry den größten Beitrag zur Erreichung weiterer TPI-Ebenen und Kontrollpunkte.

4.3.4 Usability Evaluierung mittels IsoMetrics-Verfahren

Um die Benutzerfreundlichkeit der Werkzeuge zu bewerten, wurde die Kurzversion des IsoMetrics-Verfahrens angewendet. Eine angemessene Benutzerfreundlichkeit ist von hoher Bedeutung, da sie großen Einfluss auf zukünftige Akzeptanz der Software bei den Anwendern hat. Deshalb war es wichtig eine Usability-Evaluierung mit Personen durchzuführen, die im Testprozess involviert sind und zukünftige Anwender des Testmanagementsystems darstellen. Die Usability-Evaluierung wurde mit drei zukünftigen Anwendern durchgeführt. Bei den Personen handelte es sich um den Testmanager von UE/I, einen Tester und eine Person, die als Tester eingesetzt ist, jedoch partiell auch Funktionen des Testmanagements übernimmt. Um die Evaluierung durchführen zu können, benötigten alle Testpersonen einen Zugang zu den drei zu evaluierenden Testmanagementwerkzeugen, weshalb die Zugänge angelegt und vorab kommuniziert werden mussten. Ein weiterer Grund für die frühzeitige Kommunikation war, dass sich die drei zukünftigen Anwender vorab mit den Werkzeugen vertraut machen sollten. Dies war wichtig, da zur Bewertung des IsoMetrics Fragebogens grundlegende Kenntnisse über das bzw. die Werkzeuge vorhanden sein sollten. Neben

der Bewertung der Benutzerfreundlichkeit der Werkzeuge, untersucht IsoMetrics die Unterstützung der Erfüllung der Aufgaben des Anwenders. Die Testpersonen mussten deshalb versuchen den Inhalt und den Ablauf ihrer Aufgaben und Tätigkeiten, während des Testprozesses, mit den Werkzeugen abzubilden.

Zur Durchführung der Evaluierung wurden die Fragebögen an die beteiligten Personen verteilt, wobei jeder Teilnehmer alle drei Testmanagementsysteme bewerten musste. Insgesamt wurde der Fragebogen neunmal ausgefüllt – dreimal je Testperson. Außerdem wurde die Evaluierung an nur einem Tag durchgeführt, wodurch der Zeitraum zwischen den Bewertungen der einzelnen Werkzeuge sehr gering war. Dies war von Vorteil für die Testpersonen und die Aussagekraft der Ergebnisse, weil die zeitlich aktuellen Erfahrungen und Eindrücke der Anwender bessere Vergleichsmöglichkeiten zwischen den einzelnen Werkzeugen ermöglichten.

Um die Struktur und das Vorgehen der Bewertung zu verdeutlichen, sind nachfolgend drei Beispielkriterien (Items) des ersten Bewertungsbereiches „Aufgabenangemessenheit“ und die IsoMetrics-Bewertungsskala dargestellt.

		stimmt nicht	stimmt wenig	stimmt mittelmäßig	stimmt ziemlich	stimmt sehr	keine Angabe
		1	2	3	4	5	k. A.
Index	Aufgabenangemessenheit						
A.1	Die Software zwingt mich, überflüssige Arbeitsschritte durchzuführen.		x				
A.3	Mit der Software kann ich zusammenhängende Arbeitsabläufe vollständig bearbeiten.				x		
A.4	Die Software bietet mir alle Möglichkeiten, die ich für die Bearbeitung meiner Aufgaben benötige.			x			

Quelle: Eigene Darstellung

Tab. 4.12: Bewertung mittels IsoMetrics – ein Beispiel

Beim Lesen der Bewertungskriterien wird deutlich, dass im Fragebogen positiv und negativ gepolte Items existieren. Im Beispiel sind die Kriterien A.3 und A.4 positiv gepolt, wogegen das Item A.1 negativ gepolt ist. Bei positiv gepolten Items ist eine Bewertung im oberen Skalenbereich (stimmt ziemlich / sehr) notwendig, um das Kriterium positiv zu bewerten. Für eine positive Bewertung ist bei negativ gepolten Items dagegen eine Bewertung im unteren Skalenbereich (stimmt wenig / nicht) notwendig. Im Beispiel sind somit Item A.1 und A.3 positiv bewertet wurden. Die

Polung muss bei der Auswertung des Fragebogens entsprechend berücksichtigt werden. Insgesamt sind neun von 75 Bewertungskriterien negativ gepolt.

Die vollständige Liste mitsamt allen 75 Bewertungskriterien (Items) ist durch Kontaktaufnahme mit den Entwicklern des *IsoMetrics*-Fragebogens¹¹¹ einholbar.

Zur Auswertung des Fragebogens kann eine Mittelwertbildung oder die Durchführung einer Cutoff-Analyse genutzt werden. Da nur drei Personen an der Usability-Evaluierung teilgenommen haben, ist die Auswertung über eine Mittelwertbildung nicht repräsentativ und aussagekräftig genug. Deshalb wurde die Cutoff-Analyse als Auswertungsmethode gewählt. Die Cutoff-Analyse nutzt die Struktur des Antwortschemas um Aussagen darüber treffen zu können, ob die evaluierte Software als „mittelmäßig“ oder „besser“ eingestuft wird. Es wird dazu für jede Person (1 bis m) der prozentuale Anteil (p_i) der Antworten, in denen eine Auswahl der beiden positiven Beurteilungen („besser als mittelmäßig“) getroffen wurde, bestimmt. Jede Person (i) beantwortet dabei (n_i) Fragen.¹¹²

Der prozentuale Anteil der positiv bewerteten Items (p_i) berechnet sich aus dem Verhältnis der Anzahl der positiv bewerteten Items (N) und der Anzahl der beantworteten Fragen (n_i).

$$p_i = \frac{N}{n_i}$$

Die prozentualen Anteile positiv bewerteter Items werden über alle teilnehmenden Personen aufsummiert und durch die Anzahl der Personen geteilt. Als Ergebnis ergibt sich der prozentuale Mittelwert der positiv bewerteten Items (mittlere Cutoff-Beurteilung).

Die mittlere Cutoff-Beurteilung wird durch folgende Formel ausgedrückt.

$$p = \frac{1}{m} \sum_{i=1}^m p_i$$

In Tabelle 4.13 ist das konsolidierte Ergebnis der Usability – Evaluierung dargestellt. Es wurden folgende Daten aus den Einzelauswertungen der Anwender erhoben: Anzahl bewerteter Items (n_i), Anzahl der Antworten mit keiner Angabe, Anzahl positiv bewerteter Items (N), Anzahl positiv bewerteter Items in Prozent (p_i) und Gesamtergebnis (p). Die Spalte „Anzahl k. A.“ hat keinen Einfluss auf die Bewertung, wurde zur Vollständigkeit aber mit in die Tabelle aufgenommen.

¹¹¹ Für mehr Informationen siehe (<http://www.isometrics.uni-osnabrueck.de>).

¹¹² Vgl. Gediga, 2000, S. 22.

	Rolle der befragten Person	Anzahl bewerteter Items (n _i)	Anzahl "k. A."	positiv bewertete Items (N)	positiv bewertete Items in % (p _i)	Usability Gesamtergebnis (p)
Aptest	Testmanager	70	5	29	41,43%	34,03%
	Tester	68	7	31	45,59%	
	Tester / Testmanager	73	2	11	15,07%	
Qmetry	Testmanager	72	3	55	76,39%	64,61%
	Tester	69	6	47	68,12%	
	Tester / Testmanager	73	2	36	49,32%	
Zephyr	Testmanager	71	4	52	73,24%	59,52%
	Tester	69	6	42	60,87%	
	Tester / Testmanager	72	3	32	44,44%	

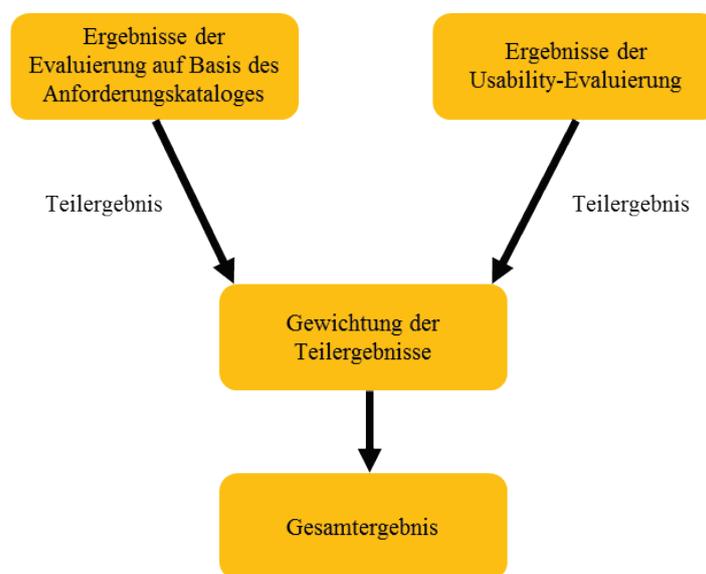
Quelle: Eigene Darstellung

Tab. 4.13: Ergebnis der Usability-Evaluierung

Analog zu den Ergebnissen in Kapitel 4.3.3 ergibt sich die gleiche Reihenfolge der Werkzeuge. Qmetry liegt mit rund 64,6 Prozent vor Zephyr mit rund 59,5 Prozent. Im Mittel wurde Qmetry rund fünf Prozent besser bewertet als Zephyr. ApTest hat mit 34 Prozent nur ein schlechtes Usability-Ergebnis erreicht. Auffällig ist außerdem, dass eine Testperson (Tester / Testmanager) eine durchweg zurückhaltende und eher skeptische Bewertungseinschätzung vorgenommen haben muss, da die Anzahl der positiv bewerteten Items deutlich unter denen der anderen beiden Teilnehmer liegt. Dies hat jedoch keinen großen Einfluss auf das Verhältnis der Werkzeugergebnisse untereinander. Es wird hingegen deutlich, welchen Ergebniseinfluss die Einstellungen und Erwartungshaltungen der Teilnehmer besitzen.

4.3.5 Abschließende Bewertung

Das Ziel der abschließenden Bewertung ist es, anhand der Ergebnisse aus Kapitel 4.3.3 und 4.3.4, ein Gesamtergebnis des Evaluierungsprozesses zu ermitteln. Dabei müssen die Ergebnisse der Evaluierung auf Basis des Anforderungskataloges mit den Ergebnissen der Usability-Evaluierung kombiniert werden. Dieser Schritt setzt eine weitere Gewichtung der beiden Teilergebnisse voraus. Abbildung 4. zeigt die Gewichtung der Teilergebnisse als essenziellen Schritt bei der Bildung des Gesamtergebnisses.



Quelle: eigene Darstellung

Abb. 4.1: Von den Teilergebnissen zum Gesamtergebnis

Die Bedeutung der Wahl der richtigen Gewichtung soll in Form eines kurzen Beispiels verdeutlicht werden. Es werden zwei Werkzeuge miteinander verglichen, wobei Werkzeug eins das bessere Teilergebnis eins besitzt und Werkzeug zwei das bessere Teilergebnis zwei besitzt. Wird eine Gewichtung von 70 Prozent (Teilergebnis eins) zu 30 Prozent (Teilergebnis zwei) gewählt, beeinflusst diese Gewichtung das Gesamtergebnis schwerwiegend. Liegen die Teilergebnisse jeweils eng beieinander, so können kleinste Änderungen an der Gewichtung einen großen Einfluss auf das Gesamtergebnis haben. In diesem Fall müsste eine Empfindlichkeitsanalyse durchgeführt werden, die ermittelt wie sensibel das Gesamtergebnis auf die Veränderung der Gewichtung reagiert.

Die Teilergebnisse der Untersuchungen aus Kapitel 4.3.3 und 4.3.4 sind hingegen durch eine einheitliche Abstufung der einzelnen Werkzeugergebnisse charakterisiert. Sowohl bei der Usability-Evaluierung, als auch bei der Evaluierung auf Basis des

Anforderungskataloges, kann sich „Qmetry Total Test Management (Enterprise) – Version 5.2“ von „Zephyr Enterprise – Version 4.0“ absetzen. Unabhängig von der Wahl der Gewichtung, ist aufgrund dieser Erkenntnis feststellbar, dass Qmetry das beste Gesamtergebnis erreichen wird. „ApTest Manager – Version 2.25.01.02“ hat in beiden Bereichen am schlechtesten abgeschnitten, wobei das Ergebnis im Bereich der Usability deutlich stärker von den Ergebnissen der Konkurrenzprodukte abweicht.

Die Usability-Evaluierung geht neben der Benutzerfreundlichkeit auch auf die Erreichung der Ziele und Aufgaben der Anwender ein. Deshalb decken die Ergebnisse der Usability-Evaluierung zu einem gewissen Grad die Ergebnisse der Evaluierung, auf Basis des Anforderungskataloges, ab. Die Unterstützung und Abbildungsfähigkeit des Testprozesses spiegelt sich somit in beiden Teilergebnissen wider. Ein Werkzeug, das durch Prüfung des Anforderungskataloges gut bewertet wurde, darf theoretisch bei den Usability-Fragen zur Zielerfüllung der Nutzer nicht schlecht bewertet worden sein.

Auf Grundlage der genannten Überlegungen wurde eine Gewichtung von 50 zu 50 festgelegt. Beide Teilergebnisse beeinflussen das Gesamtergebnis somit gleich stark. Folglich musste für jedes Werkzeug nur der Mittelwert aus den Teilergebnissen gebildet werden. Tabelle 4.14 fasst die Teilergebnisse der Kapitel 4.3.3 und 4.3.4 zusammen und stellt die daraus resultierenden Gesamtergebnisse dar.

	ApTest	Qmetry	Zephyr
Ergebnis der Evaluierung auf Basis des Anforderungskataloges	73,32%	89,01%	76,23%
Ergebnis der Usability-Evaluierung	34,03%	64,61%	59,52%
Gesamtergebnis	53,67%	76,81%	67,88%

Quelle: Eigene Darstellung

Tab. 4.14: Gesamtergebnis des Evaluierungsprozesses

Qmetry konnte mit 76,81% das beste Gesamtergebnis erreichen, gefolgt von Zephyr mit 67,88% und ApTest mit 53,67%. Es sind deutliche Unterschiede in den Teil- und Gesamtergebnissen erkennbar. Folglich bietet Qmetry den höchsten Funktionsumfang auf Basis der Betrachtungen des Kapitels 4.3.3. Ebenso bietet es den höchsten Usability-Wert auf Grundlage der Ergebnisse des Kapitels 4.3.4.

Auffällig ist das schlechte Usability-Ergebnis von ApTest. Die Unterstützung bei der Erfüllung der Aufgaben der Nutzer scheint für dieses Ergebnis weniger negativ ausschlaggebend zu sein, da ApTest in der Bewertung auf Basis des Anforderungskataloges nicht schlecht abgeschnitten hat. Aus dem direkten Vergleich mit Qmetry und Zephyr lässt sich somit schlussfolgern, dass ApTest Probleme bei der Benutzerfreundlichkeit aufweist.

Die Ergebnisse zeigen auch, dass jedes der getesteten Testmanagementwerkzeuge Schwächen aufweist. Funktionale Schwächen sind durch nicht oder nur schlecht erfüllte Kriterien des Anforderungskataloges nachprüfbar (siehe Tab. B.4). Das Erreichen von annähernd 100% im Gesamtergebnis stellt hingegen ein unrealistisches Szenario dar. Allein die Usability-Bewertung hat dabei einen zu subjektiven Einfluss auf das Gesamtergebnis. Wie in den Teilergebnissen des Kapitels 4.3.4 bereits aufgezeigt wurde, beeinflussen die Einstellungen und Erwartungshaltungen der Testpersonen die Höhe der prozentualen Usability-Werte.

Zusammenfassend ist „Qmetry Total Test Management“ das Werkzeug, das den Testprozess von UE/I am besten unterstützen und abbilden kann. Es bietet gleichzeitig den höchsten Usability-Wert und vermindert somit das Risiko der Nicht-Akzeptanz bei den zukünftigen Nutzern.

5 Zusammenfassung und Ausblick

Durch eine strukturierte Systemanalyse konnte der Testprozess von UE/I tiefgründig analysiert werden. Die komplexe Struktur des Ist-Prozesses konnte in Form von verschiedenen Aktivitätsdiagrammen und einem ER-Modell abgebildet werden. Dazu waren viele detaillierte Informationen über den vorhandenen Testprozess notwendig. Darüber hinaus wurde der gegenwärtige TPI-Fortschritt ermittelt, der den Reifegrad des Testprozesses von UE/I charakterisiert sowie dessen Entwicklungspotential veranschaulicht. Die Anwendung der Methoden des Interviews und des Workshops war nicht nur ein besonders wichtiger Schritt zur Gewinnung von qualitativen Informationen – insbesondere die durchgeführten Workshops konnten dazu beitragen ein grundlegendes Verständnis über den Testprozess zu erlangen. Mit diesem Wissen über den Testprozess war es möglich funktionale Anforderungen an ein Testmanagementsystem zu ermitteln, die präzise auf die Testaktivitäten und die Bedürfnisse des Testpersonals abgestimmt sind. Auf Basis des Anforderungskataloges konnte der wohlstrukturierte Evaluierungsprozess durchgeführt werden.

Eine ausgiebige Marktrecherche identifizierte zunächst alle grundsätzlich relevanten Testmanagementsysteme. Durch die Wahl geeigneter Vorauswahlkriterien konnte eine schrittweise Selektion derjenigen Werkzeuge erfolgen, die den Grundvoraussetzungen für einen Einsatz bei UE/I genügten. Die drei vorausgewählten Werkzeuge wurden daraufhin einer detaillierten Prüfung anhand der Kriterien des Anforderungskataloges unterzogen. Darüber hinaus wurden Untersuchungen zur Usability durchgeführt, um die Akzeptanz bei den zukünftigen Nutzern sicherzustellen. Dazu wurde das standardisierte IsoMetrics-Verfahren angewendet. Die Ergebnisse der Untersuchungen wurden daraufhin zu einem Gesamtergebnis kombiniert, wobei den Teilergebnissen beider Bereiche die gleiche Wichtigkeit und Priorität zugewiesen wurde.

Es konnte ein Testmanagementsystem ausfindig gemacht werden, das zum Testprozess von UE/I passt und mit dem sich dieser möglichst gut abbilden lässt. Das Ergebnis ist dahingehend optimal, da dieses Werkzeug in beiden Teilbereichen – funktionale Anforderungen und Usability – die besten Ergebnisse erzielt hat. Es erfüllt mit rund 89% den Großteil der funktionalen Anforderungen und konnte sich damit deutlich von den Konkurrenzprodukten absetzen. Es ist jedoch anzumerken, dass jedes der getesteten Werkzeuge seine eigenen Stärken und Schwächen besitzt. Die Relevanz dieser Stärken und Schwächen wird hingegen durch die Vorgaben des Anforderungskataloges eingegrenzt. Eine bestimmte Stärke eines evaluierten Werkzeuges kann durch die gezielte Prüfung des Anforderungskataloges „schlicht und einfach“ nicht betrachtet werden. Im Gegenzug dazu kann sich eine bestimmte Schwäche eines Werkzeuges deutlich im bewerteten Anforderungskatalog widerspiegeln, wenn dieser die Prüfung

entsprechender Kriterien voraussetzt. Letztendlich ist es jedoch von Bedeutung, dass das gewählte Werkzeug den spezifischen Anforderungen der Organisation genügt und es den Testprozess so gut wie möglich abbilden kann.

Das gewählte Werkzeug ist daher „Qmetry Total Test Management“. Es kann den Testprozess von UE/I bestmöglich abbilden und bietet gleichzeitig den höchsten Usability-Wert. Die Wahrscheinlichkeit, dass das Testmanagementsystem von den zukünftigen Nutzern abgelehnt bzw. unzureichend genutzt wird, ist deshalb bei Qmetry am geringsten. Es war nicht das Ziel der Arbeit einen Prozess für ein Werkzeug zu finden, sodass die Prozesse hätten für das neue Werkzeug angepasst werden müssen. Diese Voraussetzung ist durch Qmetry grundsätzlich erfüllt, da es sich ohne großen Aufwand in den vorhandenen Testprozess einbinden lässt. Zu einem späteren Zeitpunkt, nach der Einführung des Testmanagementsystems, kann es hingegen sogar gewollt sein die Prozesse für das Werkzeug anzupassen. Dies hängt davon ab, wie sehr sich das Werkzeug innerhalb der Organisation etablieren kann und wie groß der wirkliche Nutzen ist, der aus dem Testmanagementwerkzeug gezogen werden kann. Da sich der Testprozess bei UE/I indessen nicht grundlegend ändern wird, würde die Einführung von „Qmetry Total Test Management“ die Einführung eines fortwährenden und beständigen Systems bedeuten. Nicht zuletzt hat Qmetry aufgrund der hohen Konfigurierbarkeit und Anpassbarkeit viele Punkte bei der Evaluierung auf Basis des Anforderungskataloges bekommen. Qmetry besitzt somit einen gewissen Grad an Flexibilität, um sich an ein verändertes Systemumfeld und an verändernde Prozesse anzupassen.

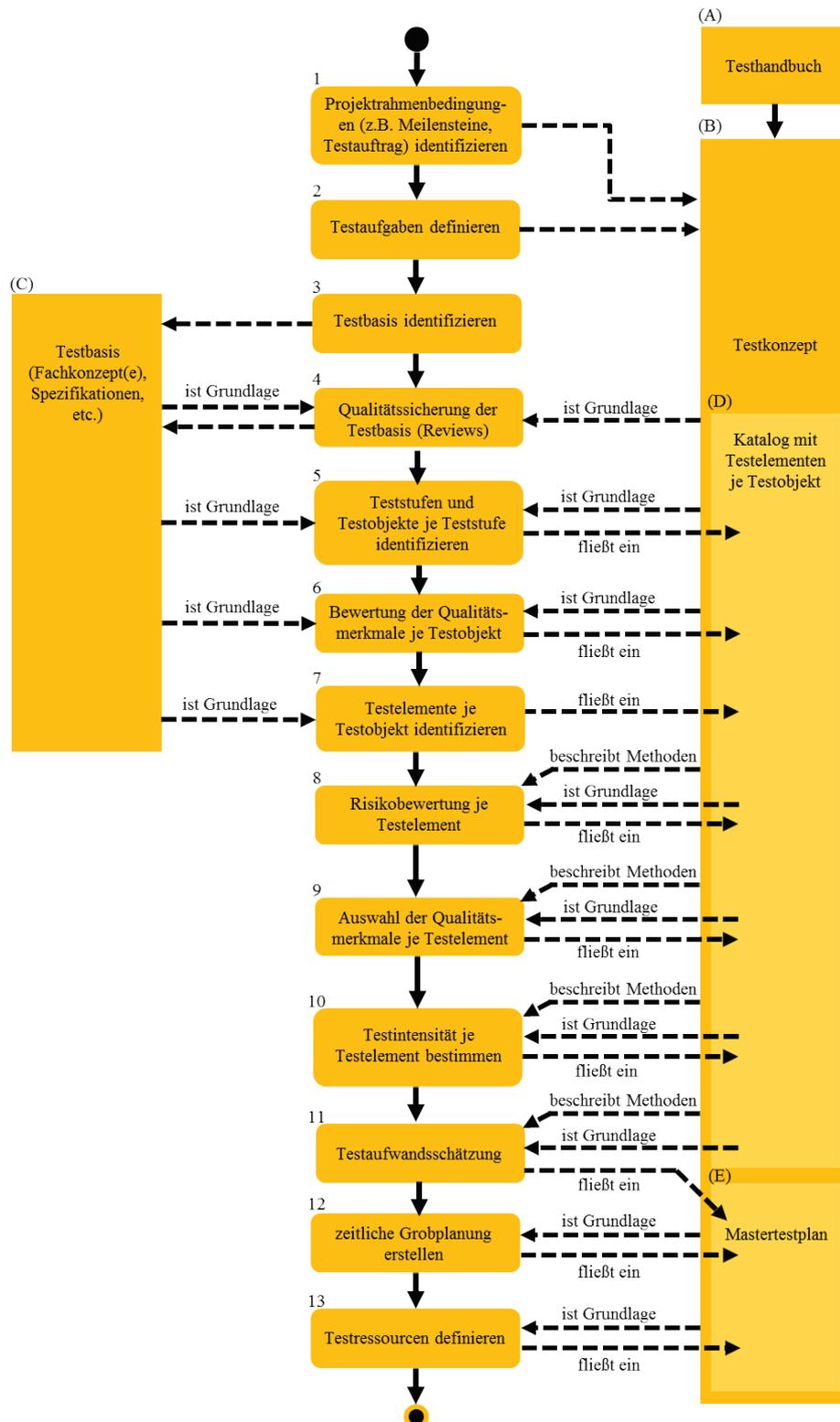
In der Arbeit wurden Werkzeuge untersucht, die bereits im Testprozess von UE/I eingebunden sind, bisher hingegen ohne ein zentrales Testmanagementsystem auskommen mussten. Nach der Einführung des Testmanagementsystems sollte zunächst die Integration mit dem intensiv genutzten Fehlermanagementsystem JIRA durchgeführt werden. Qmetry besitzt eine vollautomatische bidirektionale Integration mit JIRA, wodurch ein großer Mehrwert für das Testmanagement und dessen Kontroll- und Steuerungsfunktionen generiert wird. Da Qmetry offene Schnittstellen zur Anbindung von Testautomatisierungswerkzeugen besitzt, sollte bei UE/I auch über die Anbindung des bereits aktiv eingesetzten „HP Quicktest“-Systems nachgedacht werden. Testautomatisierungswerkzeuge bieten enormes Einsparpotential bei der Testdurchführung und sollten in Kombination mit einem Testmanagementsystem noch kontrollierbarer und effizienter einsetzbar sein.

Der Auswahlprozess für ein Testmanagementsystem hat ein klares Ergebnis gezeigt. Bevor das Werkzeug eingeführt wird, sollte eine Wirtschaftlichkeitsbetrachtung erfolgen, da der Einsatz des Werkzeuges grundsätzlich finanziell gerechtfertigt sein muss.

Qmetry konnte in dem Bereich des Anforderungskataloges – *Testauswertung und Bericht* – die meisten Kriterien erfüllen. Da diese Anforderungen aus den TPI-Untersuchungen abgeleitet wurden, wird das zukünftige Erreichen weiterer TPI-Stufen durch Qmetry am umfassendsten von allen drei betrachteten Werkzeugen unterstützt. Die infrastrukturellen Einschränkungen, insbesondere der Verzicht auf den Einsatz von Microsoft Infrastruktur, haben die Untersuchungsbasis während des Evaluierungsprozesses deutlich verkleinert. Ob abgesehen von dieser Einschränkung ein besseres Testmanagementsystem hätte gefunden werden können, bleibt offen.

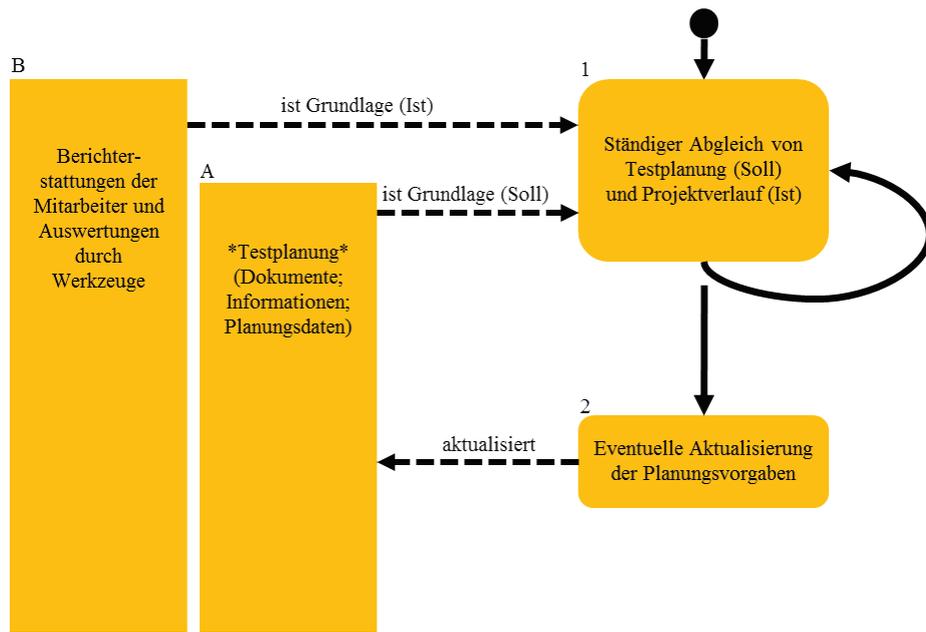
Anhang

A Abbildungen des Anhangs



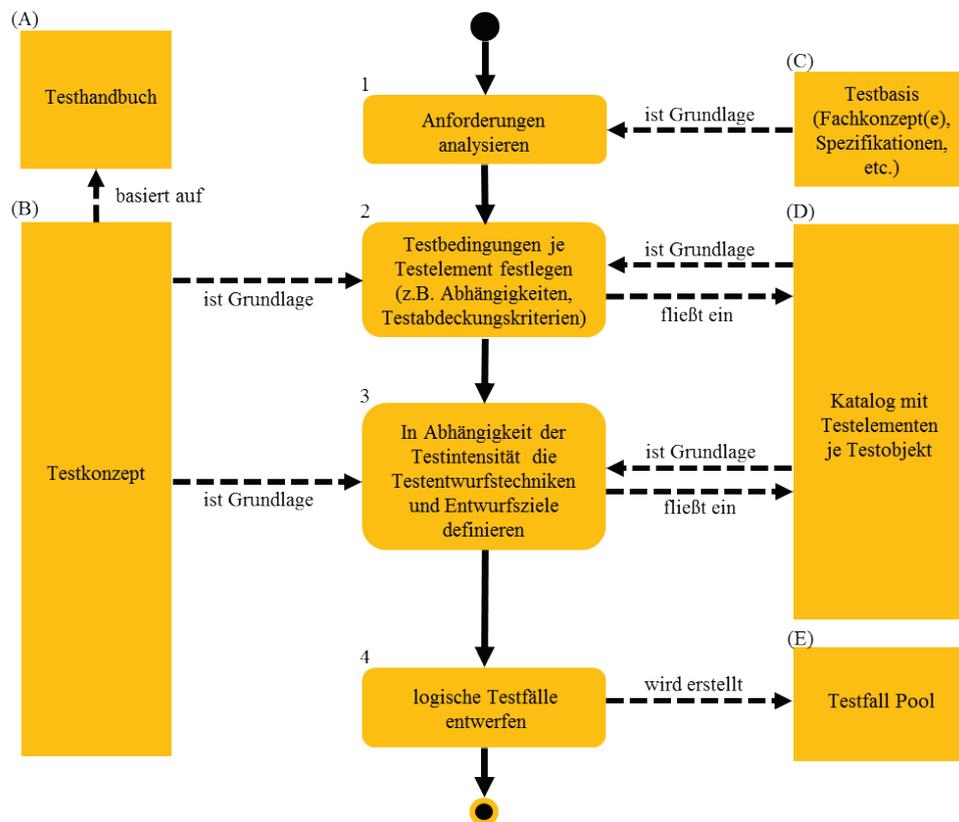
Quelle: Eigene Darstellung

Abb. A.1: Aktivitätsdiagramm – Initiale Testplanung



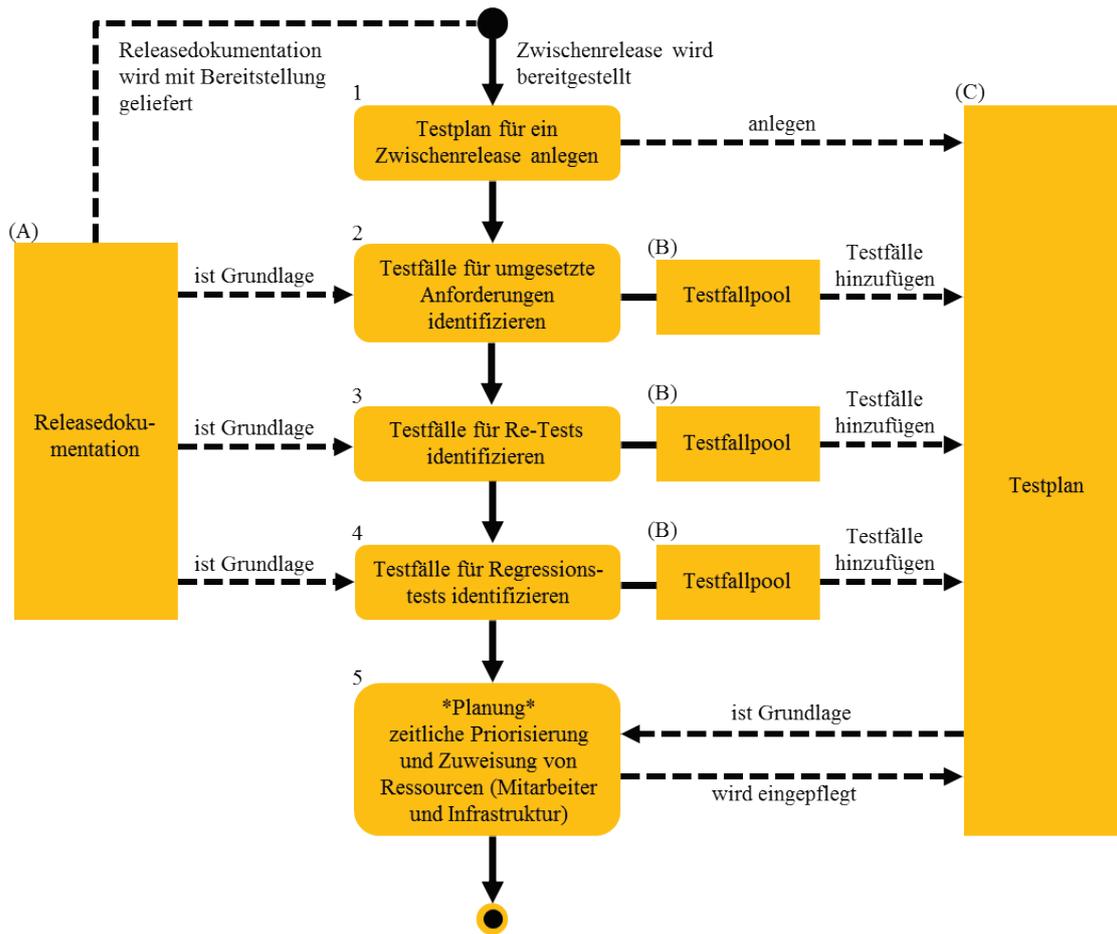
Quelle: Eigene Darstellung

Abb. A.2: Aktivitätsdiagramm – Teststeuerung



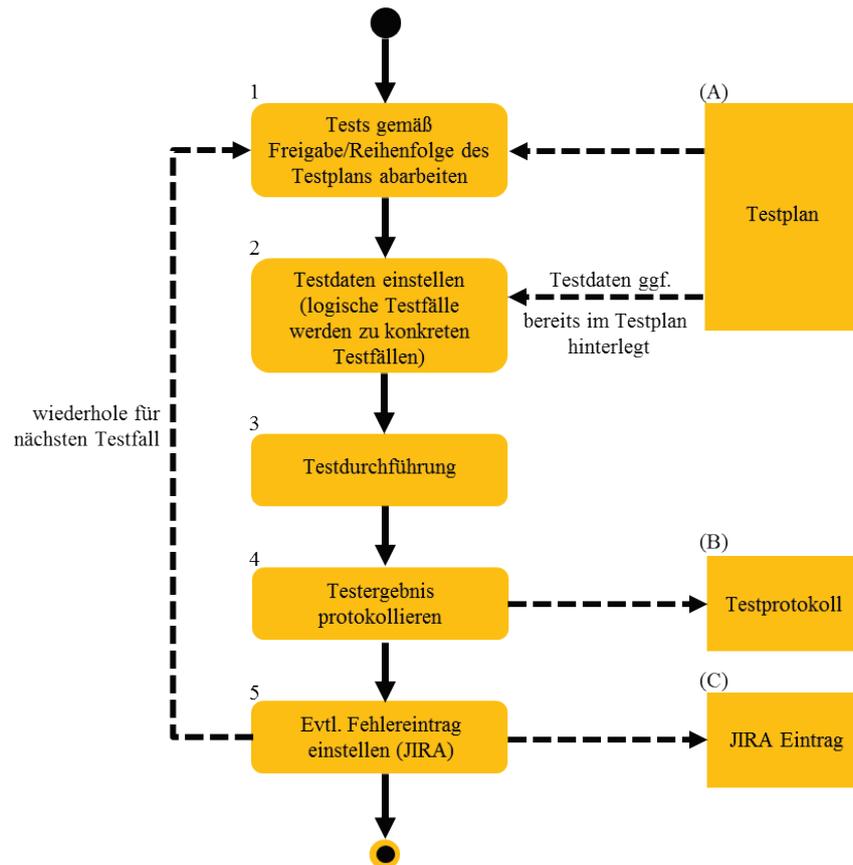
Quelle: Eigene Darstellung

Abb. A.3: Aktivitätsdiagramm – Testanalyse und Design



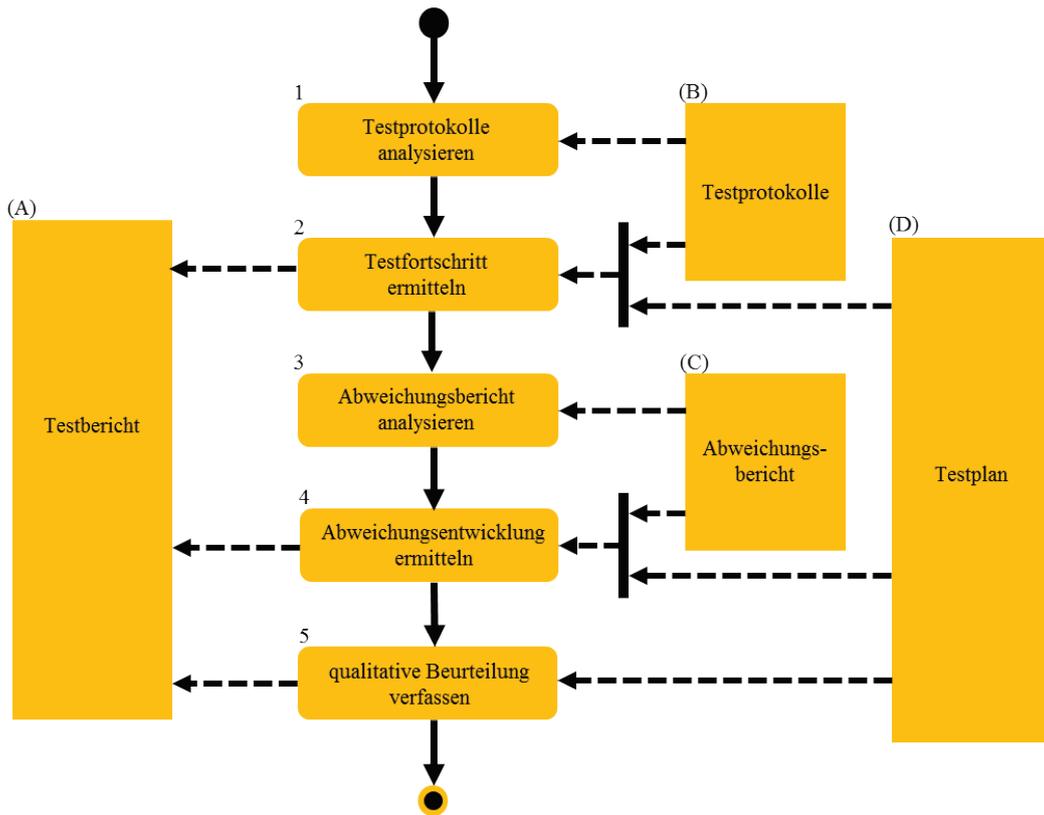
Quelle: Eigene Darstellung

Abb. A.4: Aktivitätsdiagramm – Testplanung auf Basis von Zwischenreleases



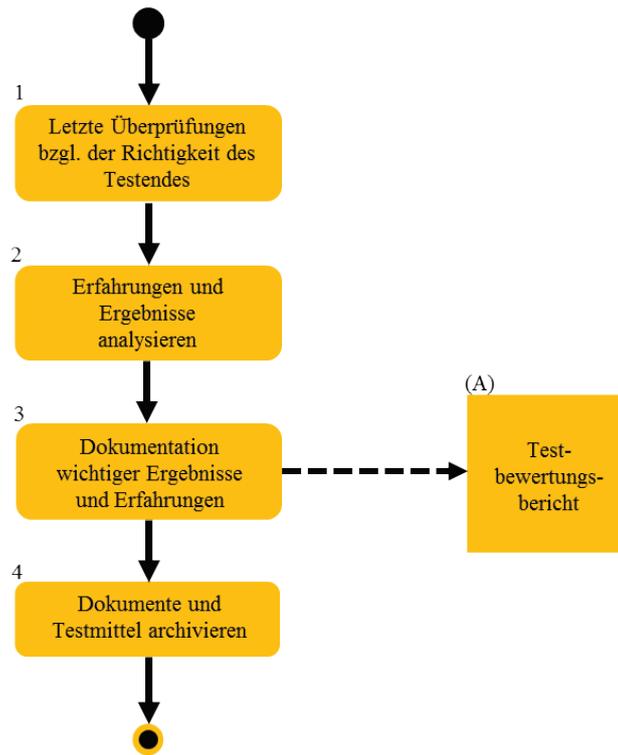
Quelle: Eigene Darstellung

Abb. A.5: Aktivitätsdiagramm – Testrealisierung und Durchführung



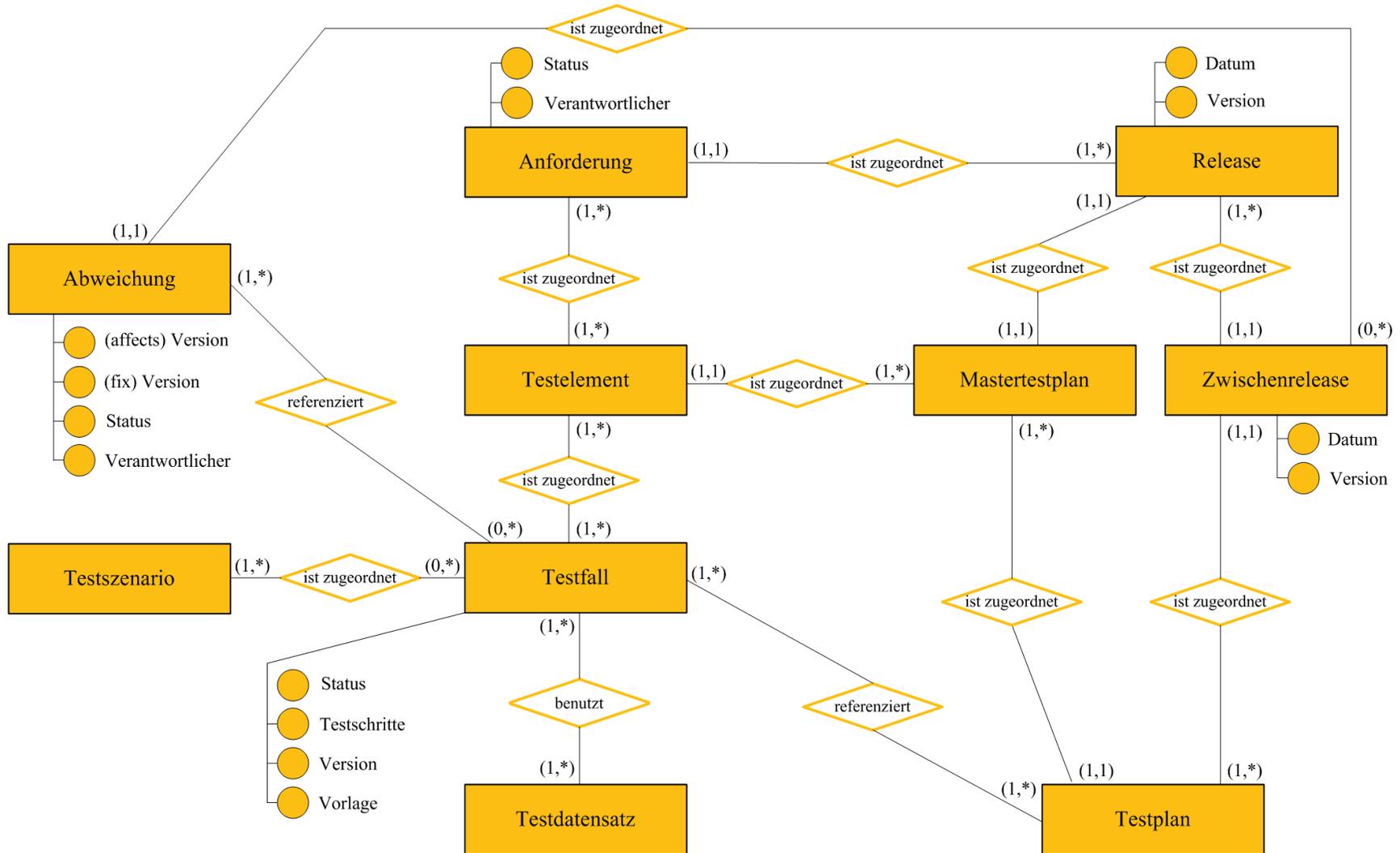
Quelle: Eigene Darstellung

Abb. A.6: Aktivitätsdiagramm – Testauswertung und Bericht



Quelle: Eigene Darstellung

Abb. A.7: Aktivitätsdiagramm – Abschluss der Testaktivitäten



Quelle: Eigene Darstellung

Abb. A.8: ER-Modell – Beziehungen wichtiger Objekte im Testprozess

B Tabellen des Anhangs

Kernbereich	Stufe	beherrschbar					effizient					optimierend			
		0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Teststrategie		A					B				C		D	
2	Einsatz des Phasenmodell		A			B									
3	Zeitpunkt der Beteiligung			A				B				C		D	
4	Kostenvoranschlag und Planung				A							B			
5	Testspezifikationstechniken		A		B										
6	Statische Testtechniken					A		B							
7	Metriken						A			B			C		D
8	Testautomatisierung				A				B			C			
9	Testumgebung				A				B						C
10	Testarbeitsplatz				A										
11	Engagement und Motivation		A					B						C	
12	Testfunktionen und Ausbildungen				A				B			C			
13	Reichweite der Methodik					A						B			C
14	Kommunikation			A		B								C	
15	Berichterstattung		A			B			C					D	
16	Fehlermanagement		A					B		C					
17	Testwaremanagement			A				B				C			D
18	Testprozessmanagement		A		B									C	
19	Prüfen							A			B				
20	Low-Level-Tests					A			B			C			

Quelle: Eigene Darstellung

Tab. B.1: TPI-Matrix – gegenwärtiger Sollzustand bei UE/I

TPI-Stufe	Kernbereich	Ebene	ID	Kernbereich-Name / Ebenen-Name / Maßnahme je Ebene	Unterstützung?
4	2	B		Einsatz des Phasenmodells / <i>Vollständiges Phasenmodell Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss</i>	
			1	Verfassung von Testabschlussbericht inkl. Betrachtung der offenstehenden Abweichungen, Risiken und Empfehlungen.	✓
4	14	B		Kommunikation / <i>Projektkommunikation, Analyseforum, Änderungsüberwachung</i>	
			2	Teststatusmeetings beinhalten stetige Betrachtung zur Qualität des Testprozesses, regelmäßige Feststellungsrunden, Ergebnisse werden schriftlich festgehalten.	✗
			3	Der Testmanager wird rechtzeitig über Änderungen in den geplanten / vereinbarten Abnahmedaten in Kenntnis gesetzt.	✗
4	15	B		Berichterstattung / <i>Fortschritt einschließlich Prioritätenzuweisung und Berichterstattung über Zeitaufwand und Testfortschritt</i>	
			4	Testfallausführung: Erfassung von Abarbeitungszeit (Durchlaufzeit); Was wurde getestet? Was ist korrekt verlaufen? Was ist noch zu testen?	✓
5	7	A		Metriken / <i>Projektmetriken (über Produkt)</i>	
			5	Erfassung und Reporting von projektorientierten Metriken.	✓
5	11	B		Engagement und Motivation / <i>Testen in Projektorganisation integriert</i>	
			6	In der (Test-) Projektplanung sind die Zyklen Testen, Fehlerkorrektur und Re-Test (erneutes Testen) zu berücksichtigen.	✗
			7	Abstimmung zwischen Test- und Entwicklungsabteilung.	✗
5	16	B		Fehlermanagement / <i>Testen in Projektorganisation integriert</i>	
			8	Fehlermanagement umfassend und konsequent betreiben.	✓
6	1	B		Teststrategie / <i>Strategie für High-Level-Tests</i>	
			9	Design einer teststufenübergreifenden Strategie zur Identifizierung von Löchern und redundanter Abdeckung.	✗
6	12	B		Testfunktionen und Ausbildungen / <i>(Formale) methodische, technische und funktionale Unterstützung, Management</i>	
			10	Explizite Einbeziehung und Beteiligung von Experten (methodisch, fachlich, technisch).	✗
6	15	C		Berichterstattung / <i>Risiken und Empfehlungen anhand von Metriken</i>	
			11	Berichte um Trendbewertung und Risikoabschätzung inkl. Empfehlungen erweitern.	✓

Quelle: Eigene Darstellung

Tab. B.2: Offene Maßnahmen zur Erreichung von TPI-Stufe 4-6

A.-Nr.	ID	Anforderung	m:s:k
Systemeigenschaften			
A.44	1	Das Testmanagementsystem muss mehrere Projekte oder mehrere Mandanten unterstützen.	m
A.45	2	Das Testmanagementwerkzeug muss eine Benutzerverwaltung mit Rollen- und Berechtigungsmodell besitzen, sodass alle Benutzer über entsprechende Rechte verfügen und gleichzeitig mit dem Werkzeug arbeiten können.	m
A.42	3	Das Testmanagementsystem muss Möglichkeiten zur Anbindung bzw. Integration an JIRA besitzen.	m
A.43	4	Zur Testautomatisierung stellt das Testmanagementsystem Möglichkeiten zur Anbindung an „HP-Quicktest“ bereit.	k
A.46	5	Zur nicht-redundanten Pflege der Benutzer gibt es die Möglichkeit einer LDAP-Anbindung.	k
Testplanung			
A.2	6	Anforderungen müssen verwaltet werden können (Anlegen, Ändern, Löschen).	m
A.3	7	Anforderungen sollen strukturiert und gruppiert werden können, um diese z. B. nach Anforderungstyp oder Zweck ordnen zu können (Gruppierungsobjekte, z. B. in Ordern bzw. einer Baumstruktur).	s
A.48	8	Anforderungen sollen anhand von Attributen klassifiziert werden können, sodass sie z. B. nach Typ oder Priorität gefiltert werden können.	s
A.4	9	Anforderungen müssen in Prosa beschrieben werden können.	m
A.50	10	In Prosa beschriebene Anforderungen sollen formatiert werden können.	s
A.51	11	Das Testmanagementsystem soll eine Suche nach Anforderungen bieten.	s
A.8	12	An den Anforderungen soll ein Prozessstatus wählbar oder definierbar sein (z. B. NEW, DESIGN, to-Be-Reviewed, READY, COMMITTED).	s
A.11	13	Anforderungen sollen einem Verantwortlichen zugewiesen werden können.	s
A.5	14	Anforderungen sollen miteinander verknüpfbar sein um Abhängigkeiten abbilden zu können.	s
A.6	15	Anforderungen sollen Anhänge (z. B. Dokumente und Bilder) angefügt werden können.	s
A.7	16	Der Im- und Export von Anforderungen soll z. B. mittels Excel- oder CSV-Dateien möglich sein.	s
A.10	17	Anforderungen müssen einem Release zugeordnet werden können.	m
A.1	18	Mit dem Testmanagementsystem müssen Releases definiert werden können.	m
A.47	19	Zu den erfassten Releases sollen Zeitpläne hinterlegt werden können.	s
A.12	20	Einem Release sollen mehrere Zwischenreleases zugeordnet werden können.	s
A.13	21	Anforderungen können einem Zwischenrelease zugeordnet werden.	k
A.14	22	Für ein Release/Zwischenrelease müssen mehrere Testpläne angelegt werden können.	m
A.15	23	Der Im- und Export von Testplänen soll z. B. mittels Excel- oder CSV-Dateien möglich sein.	s
A.16	24	Je Testplan müssen bereitgestellte Anforderungen bzw. Testfälle, die die Anforderungen prüfen, sowie Testfälle für den Re-Test, ausgewählt und zugeordnet werden können.	m
A.9	25	Das Werkzeug soll Funktionen zur Unterstützung der Aufwandsplanung bzw. -zuweisung bieten (z. B. Dauer der Tests auf Testfallebene oder Testzyklus- bzw. Testphasenebene).	s
A.17	26	Testfälle und Testfallgruppierungsobjekte sollen Benutzern oder -gruppen (Testern bzw. Verantwortliche) zugewiesen werden können.	s
A.18	27	Benutzer können, z. B. über Email, automatisch benachrichtigt werden, wenn ihnen Testaufgaben (z. B. Testfälle) zugewiesen worden sind.	k

Testanalyse und Design			
	28	Testfälle müssen verwaltet werden können (Anlegen, Ändern, Löschen).	m
A.19	29	Testfälle müssen strukturiert und gruppiert werden können, um sie z. B. nach Zweck, Teststufe oder Testart ordnen zu können (Gruppierungsobjekte, z. B. in Ordnern oder einer Baumstruktur).	m
A.49	30	Testfälle sollen anhand von Attributen klassifiziert werden können, sodass sie z. B. nach Typ oder Priorität filterbar sind.	s
A.20	31	Jeder Testfall soll mehreren Testfallgruppierungsobjekten (Gruppierungsobjekten) zugeordnet werden können.	s
A.21	32	Ein Testfall muss mehreren Testplänen zugeordnet werden können.	m
A.22	33	Jeder Testfall muss mehreren Anforderungen zugeordnet werden können.	m
	34	Testfälle müssen in Prosa beschrieben werden können.	m
A.52	35	In Prosa beschriebene Testfälle sollen formatiert werden können.	s
A.53	36	Das Testmanagementsystem soll eine Suche nach Testfällen ermöglichen.	s
A.23	37	An den Testfällen muss ein Prozessstatus wählbar oder definierbar sein (z. B. NEW, DESIGN, READY, COMMITTED).	m
A.24	38	Jeder Testfall kann beliebig viele Testschritte enthalten.	k
A.25	39	Jeder Testschritt besitzt eine Beschreibung, ein Sollergebnis und die Möglichkeit zur Beschreibung bzw. Angabe von Testdaten.	k
	40	Testfällen sollen Anhänge (z. B. Dokumente und Bilder) angefügt werden können.	s
	41	Der Im- und Export von Testfällen soll z. B. mittels Excel- oder CSV-Dateien möglich sein.	s
	42	Testfälle sollen mit Parametern (Variablen) erstellt werden können, die erst während der Testausführung konkretisiert werden (z. B. fachliche Schlüssel und Referenzen).	s
	43	Die Risikoklassifizierung bzw. -definition von Anforderungen soll unterstützt werden.	s
	44	Die Bestimmung bzw. Definition der Testintensität einer Anforderung wird unterstützt.	k
Testdurchführung			
	45	Abweichungen müssen im Testmanagementsystem angelegt, in Prosa beschrieben und mit Eigenschaften versehen werden können.	m
	46	Während der Testausführung soll eine neue Abweichung angelegt werden können.	s
A.54	47	Das Testmanagementsystem soll eine Suche nach Abweichungen bieten.	s
A.26	48	Abweichungen sollen einem Verantwortlichen zugeordnet werden können.	s
	49	Die manuelle Testausführung soll durch eine Testausführungs-Komponente bzw. durch einen Agenten unterstützt werden, wobei Testanweisungen präsentiert und die Erfassung der Testergebnisse unterstützt werden.	s
	50	Der Testausführungsstatus soll auch ohne Ausführung des kompletten Tests änderbar sein (z. B. bei Anwenderfehlern bei der manuellen Testdurchführung).	s
	51	Das Testmanagementwerkzeug unterstützt Testautomatisierungsmethoden bzw. -werkzeuge (abgesehen von HP Quicktest).	k
Testauswertung und Bericht			
		Unterstützung der Erstellung von Testfortschrittsberichten:	
A.27	52	Wie viele Anforderungen sind dem Testobjekt zugeordnet?	s
A.28	53	Wie hoch ist der Anteil an erfassten Anforderungen, denen wenigstens ein Testfall zugeordnet ist (Anforderungsüberdeckungsgrad)?	s
A.29	54	Wie ist der Fortschritt der Testfallerstellung?	s
A.30	55	Wie ist der aktuelle Testfortschritt (aktueller Status der Testfälle über Zeitraum x)?	s
A.31	56	Wie ist der Fehlerfindungstrend (Anzahl offener Abweichungen über Zeitraum x)?	k

A.32	57	Wie viel Test-Restaufwand ist noch für den Abschluss der Tests für ein Testobjekt nötig?	k
A.33	58	Wie ist der Trend für ein Testobjekt beim Abarbeiten von Testfällen?	k
A.34	59	Kann der vereinbarte Zieltermin gehalten werden?	k
		<i>Unterstützung der Erstellung von Testabschlussberichten:</i>	
A.35	60	Wie ist die Verteilung der Anforderungen hinsichtlich der Testintensität?	s
A.36	61	Wie groß ist der durchschnittliche Aufwand zur Spezifikation eines Testfalls für das Testobjekt (betrachtet je Testintensität)?	s
A.37	62	Wie viele Testzyklen (Re-Tests) waren im Mittel je Testfall notwendig (Gegebenenfalls je Testintensitätsklasse)?	s
A.38	63	Wie ist der aktuelle Status der eingeplanten Testfälle?	m
A.39	64	Wie ist der aktuelle Status der identifizierten Abweichungen?	s
		<i>Unterstützung der Erstellung von Testbewertungsberichten:</i>	
A.40	65	Wie viele Testfälle sind im Mittel einer Anforderung zugewiesen (Testüberdeckungsgrad)?	s
A.41	66	Wie groß ist der durchschnittliche Aufwand zur Durchführung eines Testfalls für das Testobjekt (betrachtet je Testintensität)?	s
	67	Der Export von Zahlenwerten und grafischen Darstellungen einzelner Auswertungen (Metriken) soll in gängigen Dateiformaten (z. B. Excel, Word oder PDF) möglich sein.	s
	68	Der Export von Berichten, mindestens als Abbildung, soll möglich sein.	s
	69	Berichte sind frei konfigurierbar, z. B. für die Entitäten Testfall, Anforderung, Test oder Abweichung.	k
	70	Berichte werden automatisch, z. B. zu bestimmten Zeiten oder Ereignissen, an konfigurierbare Benutzer und -gruppen verteilt werden (z. B. per Email).	k

Quelle: Eigene Darstellung

Tab. B.3: Validierter und inhaltlich priorisierter Anforderungskatalog

ID	Anforderung	Punkte	Max.	Bewertungsmaßstab	Bewertung		
					ApTest	Qmetry	Zephyr
Systemeigenschaften							
1	Das Testmanagementsystem muss mehrere Projekte oder mehrere Mandanten unterstützen.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
2	Das Testmanagementwerkzeug muss eine Benutzerverwaltung mit Rollen- und Berechtigungsmodell besitzen, sodass alle Benutzer über entsprechende Rechte verfügen und gleichzeitig mit dem Werkzeug arbeiten können.	10	30	<i>Granularität der Rollen- und Rechtevergabe:</i> Rollen sind vordefiniert (v) und selbst anlegbar (a) und die Rechtevergabe erfolgt auf Modulebene (m) oder Lese-Schreib-Recht-Ebene (r) = (3); nur (a) und (m) oder (r) = (2); nur (v) und (m) oder (r) = (1); nicht mehrere Benutzer oder nicht (m) oder (r) = (0)	3	3	3
3	Das Testmanagementsystem muss Möglichkeiten zur Anbindung bzw. Integration an JIRA besitzen.	10	30	<i>Art der Integration:</i> vollautomatische bidirektionale Synchronisation (3); manuelle bidirektionale Synchronisation (Aktualisierung notwendig) (2); unidirektionale Synchronisation vom Testmanagementwerkzeug nach JIRA (1); nicht erfüllt (0)	1	3	3
4	Zur Testautomatisierung stellt das Testmanagementsystem Möglichkeiten zur Anbindung an „HP-Quicktest“ bereit.	3	9	erfüllt (3); nicht erfüllt (0)	3	3	3
5	Zur nicht-redundanten Pflege der Benutzer gibt es die Möglichkeit einer LDAP-Anbindung.	3	9	erfüllt (3); nicht erfüllt (0)	3	3	3
			108		88	108	108
Testplanung							
6	Anforderungen müssen verwaltet werden können (Anlegen, Ändern, Löschen).	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
7	Anforderungen sollen strukturiert und gruppiert werden können, um diese z. B. nach Anforderungstyp oder Zweck ordnen zu können (Gruppierungsobjekte, z. B. in Ordnern bzw. einer Baumstruktur).	6	18	<i>Grad der Strukturierungsfähigkeit:</i> Anforderungen sind untereinander strukturierbar und beliebig viele Ordner Ebenen anlegbar (3); beliebig viele Ordner Ebenen (2); eingeschränkte Strukturierungsfähigkeit, z. B. durch nur eine oder wenige Ordner Ebenen (1); nicht erfüllt (0)	2	3	2

8	Anforderungen sollen anhand von Attributen klassifiziert werden können, sodass sie z. B. nach Typ oder Priorität gefiltert werden können.	6	18	<i>Umfang der Eigenschaften und Klassifizierungsmöglichkeiten:</i> Attributfelder sind frei konfigurierbar (3); Attributfelder sind vorgegeben und ausreichend (2); Attributfelder sind vorgegeben und eingeschränkt nutzbar (1); nicht erfüllt (0)	3	3	3
9	Anforderungen müssen in Prosa beschrieben werden können.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
10	In Prosa beschriebene Anforderungen sollen formatiert werden können.	6	18	<i>Umfang der Formatierungsmöglichkeiten:</i> gut (3); mittel (2); schlecht (1); nicht erfüllt (0)	3	3	2
11	Das Testmanagementsystem soll eine Volltextsuche nach Anforderungen bieten.	6	18	<i>Umfang der Filtermöglichkeiten:</i> gut (3); mittel (2); schlecht (1); nicht erfüllt (0)	3	3	1
12	An den Anforderungen soll ein Prozessstatus wählbar oder definierbar sein (z. B. NEW, DESIGN, to-Be-Reviewed, READY, COMMITTED).	6	18	<i>Prozessstatus:</i> frei konfigurierbar (3); vorgegeben und ausreichend (2); vorgegeben und eingeschränkt nutzbar (1); nicht erfüllt (0)	3	3	3
13	Anforderungen sollen einem Verantwortlichen zugewiesen werden können.	6	18	erfüllt (3); nicht erfüllt (0)	3	3	0
14	Anforderungen sollen miteinander verknüpfbar sein um Abhängigkeiten abbilden zu können.	6	18	<i>Grad der Verknüpfbarkeit:</i> Verknüpfung über direkte Verlinkung, um Abhängigkeiten auch getrennt gruppierter bzw. erfasster Anforderungen darstellen zu können(3); Darstellung durch optische Verknüpfung von Anforderungen, z. B. in einer Baumstruktur (2); nicht erfüllt (0)	0	2	0
15	Anforderungen sollen Anhänge (z. B. Dokumente und Bilder) angefügt werden können.	6	18	erfüllt (3); nicht erfüllt (0)	3	3	3
16	Der Im- und Export von Anforderungen soll z. B. mittels Excel- oder CSV-Dateien möglich sein.	6	18	<i>Flexibilität:</i> frei konfigurierbarer Import (Zuordnen der Spalten) und Filterfunktionen für den Export (3); fest vorgegebene Struktur der Daten beim Import oder keine Filterfunktionen für den Export (2); fest vorgegebene Struktur beim Import und keine Filtermöglichkeiten für den Export (1); nicht erfüllt (0)	3	3	3
17	Anforderungen müssen einem Release zugeordnet werden können.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3

18	Mit dem Testmanagementsystem müssen Releases definiert werden können.	10	30	<i>Abbildungsfähigkeit:</i> Releases sind als eigenständiges Objekt anlegbar (3); Abbilden eines Releases über die Benennung von nicht-funktionalen Objekten möglich, z. B. Ordner in dem Testpläne strukturiert werden (2); Abbilden eines Releases über die Benennung von funktionalen Objekten, z. B. Testpläne oder Testsuiten mit entsprechender Versionsnummer (1); nicht erfüllt (0)	2	3	3
19	Zu den erfassten Releases sollen Zeitpläne hinterlegt werden können.	6	18	<i>Abbildungsfähigkeit:</i> Komponente mit Projektplanungsfunktionen, z. B. für Meilensteinpläne und Testzeiträume (3); einfaches Hinterlegen des Releasedatums (1); nicht erfüllt (0)	1	1	3
20	Einem Release sollen mehrere Zwischenreleases zugeordnet werden können.	6	18	<i>Abbildungsfähigkeit:</i> direktes Abbilden der Releasestruktur durch eigenständige Zwischenrelease-Objekte möglich (3); durch Benennung der Releaseobjekte bzw. nicht-funktionalen Objekte identifizierbar (2); über Benennung von funktionalen Objekten, z. B. Testpläne oder Testsuiten mit Versionsnummer (1); nicht erfüllt (0)	2	3	2
21	Anforderungen können einem Zwischenrelease zugeordnet werden.	3	9	<i>Abbildungsfähigkeit:</i> Anforderungen sind direkt dem Zwischenrelease-Objekt zuweisbar (3); Anforderungen sind an Testfälle geknüpft, die für ein Zwischenrelease ausgewählt worden sind (1); nicht erfüllt (0)	1	3	1
22	Für ein Release / Zwischenrelease müssen mehrere Testpläne angelegt werden können.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
23	Der Im- und Export von Testplänen soll z. B. mittels Excel- oder CSV-Dateien möglich sein.	6	18	<i>Im- und Exportfähigkeit der werkzeugspezifischen Testpläne:</i> Import und Export möglich (3); nur Im- oder Export möglich (1); nicht erfüllt (0)	0	3	1
24	Je Testplan müssen bereitgestellte Anforderungen bzw. Testfälle, die die Anforderungen prüfen, sowie Testfälle für den Re-Test, ausgewählt und zugeordnet werden können.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
25	Das Werkzeug soll Funktionen zur Unterstützung der Aufwandsplanung bzw. -zuweisung bieten (z. B. Dauer der Tests auf Testfallebene oder Testzyklus- bzw. Testphasenebene).	6	18	<i>Möglichkeiten der Planung und Zuweisung von Aufwänden:</i> auf Projekt-, Testphasen-, und Testfallebene (3); Testphasen- oder Testplanebene und auf Testfallebene (2); nur auf Testfall- oder Testphasenebene (1); nicht erfüllt (0)	2	1	2

26	Testfälle und Testfallgruppierungsobjekte sollen Benutzern oder -gruppen (Testern bzw. Verantwortliche) zugewiesen werden können.	6	18	<i>Möglichkeiten der Zuweisung:</i> Testfälle und Testfallgruppierungen (3); nur Testfälle (2); nur Testfallgruppierungen (1); nicht erfüllt (0)	2	2	3
27	Benutzer können, z. B. über Email, automatisch benachrichtigt werden, wenn ihnen Testaufgaben (z. B. Testfälle) zugewiesen worden sind.	3	9	erfüllt (3); nicht erfüllt (0)	3	3	0
			450		362	414	351
Testanalyse und Design							
28	Testfälle müssen verwaltet werden können (Anlegen, Ändern, Löschen).	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
29	Testfälle müssen strukturiert und gruppiert werden können, um sie z. B. nach Zweck, Teststufe oder Testart ordnen zu können (Gruppierungsobjekte, z. B. in Ordern oder einer Baumstruktur).	10	30	<i>Grad der Strukturierungsfähigkeit:</i> es sind beliebig viele Ordner Ebenen zur Strukturierung anlegbar (3); begrenzte, aber ausreichende Anzahl an Ebenen (2); eingeschränkte Strukturierungsfähigkeit, z. B. durch nur eine oder wenige Ordner Ebenen (1); nicht erfüllt (0)	3	3	3
30	Testfälle sollen anhand von Attributen klassifiziert werden können, sodass sie z. B. nach Typ oder Priorität filterbar sind.	6	18	<i>Umfang der Eigenschaften und Klassifizierungsmöglichkeiten:</i> Attributfelder sind frei konfigurierbar (3); Attributfelder sind vorgegeben und ausreichend (2); Attributfelder sind vorgegeben und eingeschränkt nutzbar (1); nicht erfüllt (0)	3	3	3
31	Jeder Testfall soll mehreren Testfallgruppierungsobjekten (Gruppierungsobjekten) zugeordnet werden können.	6	18	erfüllt (3); nicht erfüllt (0)	3	3	3
32	Ein Testfall muss mehreren Testplänen zugeordnet werden können.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
33	Jeder Testfall muss mehreren Anforderungen zugeordnet werden können.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
34	Testfälle müssen in Prosa beschrieben werden können.	10	30	erfüllt (3); nicht erfüllt (0)	3	3	3
35	In Prosa beschriebene Testfälle sollen formatiert werden können.	6	18	<i>Umfang der Formatierungsmöglichkeiten:</i> gut (3); mittel (2); schlecht (1); nicht erfüllt (0)	3	3	0
36	Das Testmanagementsystem soll eine Volltextsuche nach Testfällen bieten.	6	18	<i>Umfang der Filtermöglichkeiten:</i> gut (3); mittel (2); schlecht (1); nicht erfüllt (0)	3	3	1

37	An den Testfällen muss ein Prozessstatus wählbar oder definierbar sein (z. B. NEW, DESIGN, READY, COMMITTED).	10	30	<i>Prozessstatus</i> : frei konfigurierbar (3); vorgegeben und ausreichend (2); vorgegeben und eingeschränkt nutzbar (1); nicht erfüllt (0)	3	3	3
38	Jeder Testfall kann beliebig viele Testschritte enthalten.	3	9	erfüllt (3); nicht erfüllt (0)	3	3	3
39	Jeder Testschritt besitzt eine Beschreibung, ein Sollergebnis und die Möglichkeit zur Beschreibung bzw. Angabe von Testdaten.	3	9	<i>Beschreibungsgrad der Testschritte</i> : es können Beschreibung, Sollergebnis und Testdaten hinterlegt werden (3); nur 2 der Eigenschaften sind hinterlegbar (2); nur eine Eigenschaft ist hinterlegbar (1); nicht erfüllt (0)	3	3	2
40	Testfällen sollen Anhänge (z. B. Dokumente und Bilder) angefügt werden können.	6	18	erfüllt (3); nicht erfüllt (0)	3	3	3
41	Der Im- und Export von Testfällen soll z. B. mittels Excel- oder CSV-Dateien möglich sein.	6	18	<i>Flexibilität</i> : frei konfigurierbarer Import (Zuordnen der Spalten) und Filterfunktionen für den Export (3); fest vorgegebene Struktur der Daten beim Import oder keine Filterfunktionen für den Export (2); fest vorgegebene Struktur beim Import und keine Filtermöglichkeiten für den Export (1); nicht erfüllt (0)	3	3	3
42	Testfälle sollen mit Parametern (Variablen) erstellt werden können, die erst während der Testausführung konkretisiert werden (z. B. fachliche Schlüssel und Referenzen).	6	18	erfüllt (3); nicht erfüllt (0)	0	3	0
43	Die Risikoklassifizierung bzw. -definition von Anforderungen soll unterstützt werden.	6	18	<i>Abbildungsfähigkeit</i> : eine Risikoberechnung ist direkt im Werkzeug möglich (3); das Werkzeug bietet unterstützende Informationen zur Berechnung des Risikos und erfasst das ermittelte Risiko (2); das separat berechnete bzw. ermittelte Risiko kann in einem Feld hinterlegt werden (1); nicht erfüllt (0)	1	1	1
44	Die Bestimmung bzw. Definition der Testintensität einer Anforderung wird unterstützt.	3	9	<i>Abbildungsfähigkeit</i> : eine Intensitätsberechnung ist direkt im Werkzeug möglich (3); das Werkzeug bietet unterstützende Informationen zur Berechnung der Intensität und erfasst die ermittelte Intensität (2); die separat berechnete bzw. ermittelte Intensität kann in einem Feld hinterlegt werden (1); nicht erfüllt (0)	1	1	1
			351		315	333	282

Testdurchführung							
45	Abweichungen müssen im Testmanagementsystem angelegt, in Prosa beschrieben und mit Eigenschaften versehen werden können.	10	30	<i>Detailgehalt der erstellten Abweichungen:</i> die wichtigsten JIRA-Felder (Eigenschaften) sind vorgegeben und zusätzliche, benutzerdefinierte JIRA-Felder können konfiguriert werden (3); Die wichtigsten JIRA-Felder sind vorhanden (2); JIRA Felder sind nur in ausreichendem Maß vorhanden (1); nicht erfüllt(0)	3	3	3
46	Während der Testausführung soll eine neue Abweichung angelegt werden können.	6	18	<i>Flexibilität:</i> Abweichungen sind für Testfälle und sonstige Abweichungen, die nicht in Bezug zu einem Testfall stehen, anlegbar (3); Abweichungen sind für Testfälle anlegbar (2); nicht erfüllt (0)	2	2	3
47	Das Testmanagementsystem soll eine Suche nach Abweichungen bieten.	6	18	<i>Umfang der Filtermöglichkeiten:</i> gut (3); mittel (2); schlecht (1); nicht erfüllt (0)	0	3	2
48	Abweichungen sollen einem Verantwortlichen zugeordnet werden können.	6	18	erfüllt (3); nicht erfüllt (0)	3	3	3
49	Die manuelle Testausführung soll durch eine Testausführungs-Komponente bzw. durch einen Agenten unterstützt werden, wobei Testanweisungen präsentiert und die Erfassung der Testergebnisse unterstützt werden.	6	18	erfüllt (3); nicht erfüllt (0)	3	3	0
50	Der Testausführungsstatus soll auch ohne Ausführung des kompletten Tests änderbar sein (z. B. bei Anwenderfehlern bei der manuellen Testdurchführung).	6	18	erfüllt (3); nicht erfüllt (0)	0	3	3
51	Das Testmanagementwerkzeug unterstützt Testautomatisierungsmethoden bzw. -werkzeuge (abgesehen von HP Quicktest).	3	9	<i>Offenheit der Systemschnittstellen zur Anbindung und Anzahl der offiziell unterstützten Werkzeuge:</i> gut (3); mittel (2); schlecht (1); nicht erfüllt (0)	2	3	3
			129		84	123	105
Testauswertung und Bericht							
	<i>Unterstützung der Erstellung von Testfortschrittsberichten:</i>						
52	Wie viele Anforderungen sind dem Testobjekt zugeordnet?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	3	3	3

53	Wie hoch ist der Anteil an erfassten Anforderungen, denen wenigstens ein Testfall zugeordnet ist (Anforderungsüberdeckungsgrad)?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	1	3	3
54	Wie ist der Fortschritt der Testfallerstellung?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	3	2	3
55	Wie ist der aktuelle Testfortschritt (aktueller Status der Testfälle über Zeitraum x)?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	3	3	3
56	Wie ist der Fehlerfindungstrend (Anzahl offener Abweichungen über Zeitraum x)?	3	9	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	0	2	3
57	Wie viel Test-Restaufwand ist noch für den Abschluss der Tests für ein Testobjekt nötig?	3	9	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	3	2	0
58	Wie ist der Trend für ein Testobjekt beim Abarbeiten von Testfällen?	3	9	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	0	3	3
59	Kann der vereinbarte Zieltermin gehalten werden?	3	9	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	3	0	0
	<i>Unterstützung der Erstellung von Testabschlussberichten:</i>						
60	Wie ist die Verteilung der Anforderungen hinsichtlich der Testintensität?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	2	2	0

61	Wie groß ist der durchschnittliche Aufwand zur Spezifikation eines Testfalls für das Testobjekt (betrachtet je Testintensität)?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	0	0	0
62	Wie viele Testzyklen (Re-Tests) waren im Mittel je Testfall notwendig (gegebenenfalls je Testintensitätsklasse)?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	0	0	0
63	Wie ist der aktuelle Status der eingeplanten Testfälle?	10	30	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	3	3	3
64	Wie ist der aktuelle Status der identifizierten Abweichungen?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	0	3	3
<i>Unterstützung der Erstellung von Testbewertungsberichten:</i>							
65	Wie viele Testfälle sind im Mittel einer Anforderung zugewiesen (Testüberdeckungsgrad)?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	1	2	0
66	Wie groß ist der durchschnittliche Aufwand zur Durchführung eines Testfalls für das Testobjekt (betrachtet je Testintensität)?	6	18	<i>Abbildungsfähigkeit:</i> durch das Werkzeug darstellbar (3); erstellbar durch benutzerspezifische Auswertungen, z. B. durch freie SQL-Abfragen auf die Datenbank (2); aus der Darstellung ableitbar (1); nicht darstellbar (0)	0	2	0
67	Der Export von Zahlenwerten und grafischen Darstellungen einzelner Auswertungen (Metriken) soll in gängigen Dateiformaten (z. B. Excel, Word oder PDF) möglich sein.	6	18	erfüllt (3); nicht erfüllt (0)	0	3	0
68	Der Export von Berichten, mindestens als Abbildung, soll möglich sein.	6	18	erfüllt (3); nicht erfüllt (0)	0	3	3
69	Berichte sind frei konfigurierbar, z. B. für die Entitäten Testfall, Anforderung, Test oder Abweichung.	3	9	<i>Grad der Konfigurierbarkeit und inhaltlicher Umfang:</i> gut (3); mittel (2); schlecht (1); nicht erfüllt (0)	2	2	3

70	Berichte werden automatisch, z. B. zu bestimmten Zeiten oder Ereignissen, an konfigurierbare Benutzer und -gruppen verteilt werden (z. B. per Email).	3	9	erfüllt (3); nicht erfüllt (0)	0	0	3
			300		132	213	174

Quelle: Eigene Darstellung

Tab. B.4: Vollständig bewerteter Anforderungskatalog

Literaturverzeichnis

- Andler, Nicolai: *Tools für Projektmanagement, Workshops und Consulting: Kompendium der wichtigsten Techniken und Methoden*, 4. Auflage, Erlangen 2012.
- Bath, Graham; McKay, Judy: *Praxiswissen Softwaretest - Test Analyst und Technical Test Analyst: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard*. 2.Auflage, Heidelberg 2011.
- Brügge, Bernd; Dutoit, Allen H.: *Objektorientierte Softwaretechnik: mit UML, Entwurfsmustern und Java*, 2. Auflage, München [u.a.] 2004.
- Buxmann, Peter; Diefenbach, Heiner; Hess, Thomas: *Die Softwareindustrie: Ökonomische Prinzipien, Strategien, Perspektiven*, Berlin / Heidelberg 2008.
- Colditz, Marei, et al.: *Zur Qualität von IT-Systemen – Methoden und Verfahren für den Aktuar*, Berlin / Heidelberg 2010. In: Aschenbrenner, Michael: *Informationsverarbeitung in Versicherungsunternehmen*, S. 389-416.
- Ewijk, Alexander van: *TPI® Next – geschäftsbasierte Verbesserung des Testprozesses*, Heidelberg 2011.
- Jackson, Recardo: *Testmanagement: Professionelles Testen*. In: Informatik-Spektrum, 32. Jg. 2009, Nummer 1, S. 37-41.
- Koomen, Tim; et al.: *TMap Next*, Heidelberg 2008.
- Ludewig, Jochen; Lichter, Horst: *Software Engineering: Grundlagen, Menschen, Prozesse, Techniken*, Heidelberg 2007.
- Pezzè, Mauro; Young, Michal: *Software testen und analysieren: Prozesse, Prinzipien und Techniken*, München 2009.
- Pohl, Klaus: *Requirements engineering: Grundlagen, Prinzipien, Techniken*, 2. Auflage, Heidelberg 2008.
- Pol, Martin; Koomen, Tim; Spillner, Andreas: *Management und Optimierung des Testprozesses: ein praktischer Leitfaden für erfolgreiches Testen mit TPI und TMap*, 2. Auflage, Heidelberg 2002.
- Rupp, Chris: *Requirements Engineering und –Management: professionelle, iterative Anforderungsanalyse für die Praxis*, 5. Auflage, München / Wien 2009.

Sarodnick, Florian; Brau, Henning: *Methoden der Usability Evaluation: wissenschaftliche Grundlagen und praktische Anwendung*, 2. Auflage, Bern 2011.

Schill, Alexander; Springer, Thomas: *Verteilte Systeme: Grundlagen und Basistechnologien*, 2. Auflage, Berlin / Heidelberg 2012.

Spillner, Andreas; Linz, Tilo: *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester - Foundation Level nach ISTQB-Standard*, 4. Auflage, Heidelberg 2010.

Spillner, Andreas; et al.: *Praxiswissen Softwaretest – Testmanagement: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard*, 3. Auflage, Heidelberg 2011.

Stockmann, Reinhard: *Handbuch zur Evaluation: Eine praktische Handlungsanleitung*, Münster [u.a.] 2007.

Tabeling, Peter: *Softwaresysteme und ihre Modellierung: Grundlagen, Methoden und Techniken*, Berlin / Heidelberg 2006.

Thaller, Georg Erwin: *Software-Test: Verifikation und Validation*, 2. Auflage, Hannover 2002.

Internetquellen

German Testing Board e. V.: *ISTQB® Certified Tester*.
http://german-testing-board.info/de/certified_tester.shtml
 (letzter Zugriff: 24.05.2012 - 19:45Uhr).

Hamborg, K.-C; Gediga, Günther: *IsoMetrics: Development of software usability instrument*.
<http://www.isometrics.uni-osnabrueck.de>
 (letzter Zugriff 30.07.2012 - 18:15Uhr).

Hamburg, Matthias; Hehn Uwe (Hrsg.): *ISTQB / GTB Standardglossar der Testbegriffe*, Version 2.1, 30.09.2010.
http://www.german-testing-board.info/downloads/pdf/CT_Glossar_EN_DE_V21.pdf
 (letzter Zugriff: 15.06.2012 - 17:25Uhr).

International Software Testing Qualifications Board: *About ISTQB: ISTQB® - Advancing the software testing profession*.
<http://www.istqb.org/about-istqb.html>
 (letzter Zugriff: 24.05.2012 - 18:55Uhr).

ISTQB - International Software Testing Qualifications Board: *Certified Tester: Advanced Level Syllabus*, Version 2007.
http://www.software-tester.ch/PDF-Files/ISTQB_CTAL_Lehrplan_2007_german_approved.pdf
(letzter Zugriff: 15.06.2012 - 19:10Uhr).

Mieg, Harald A.; Näf, Matthias: *Experteninterviews*, 2. Auflage, Institut für Mensch-Umwelt-Systeme (HES), ETH Zürich 2005.
http://www.mieg.ethz.ch/education/Skript_Experteninterviews.pdf
(letzter Zugriff: 13.07.2012 - 16:45Uhr).

imbus AG.
http://www.imbus.de/tool-liste/#cat_11
(letzter Zugriff am 17.07.2012 - 15:30Uhr).

Aberdour, Mark: *opensourcetesting.org: test management tools*.
<http://opensourcetesting.org/testmgt.php>
(letzter Zugriff am 17.07.2012 - 18:00Uhr).

XQual: *Support: Open-Source tools*.
<http://www.xqual.com/qa/tools.html>
(letzter Zugriff am 18.07.2012 - 12:30Uhr).

Interne und geschützte Quellen

Gediga, Günther; Hamborg, Kai-Christoph; Willumeit, Heinz: *Das IsoMetrics-Manual*, Version 1.15b, Osnabrücker Schriftenreihe Software-Ergonomie – Universität Osnabrück 2000.

Lufthansa Technik Logistik Services GmbH: *Interne Ergebnispräsentation: Lessons Learned Net^{WG}*, 14.10. 2010.

Lufthansa Technik Logistik Services GmbH: *Testhandbuch – UE/I: Für IT-Realisierungen der Lufthansa Technik Logistik Services GmbH*, Version 1.0, 29.02.2012.

Selbstständigkeitserklärung

Hiermit versichere ich, Philipp Ernst, Matrikel-Nummer: 182045, dass ich diese Bachelorarbeit mit dem Thema: „Evaluierung und Auswahl eines Testmanagementsystems“ selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht habe.

Ort, Datum

Unterschrift