



Konzepte für Agile Qualitätssicherung und -bewertung in Wartungs- und Weiterentwicklungs-Projekten

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von: Dipl. Informationswirt André Janus
geb. am 28. Juli 1979 in Hannover

Gutachter: Prof. Dr. Reiner Dumke
Prof. Dr. Andreas Schmietendorf
Prof. Dr. Martin Gaedke
Prof. Dr. Hans-Knud Arndt

Promotionskolloquium in Magdeburg am 22. Oktober 2012

Magdeburger Schriften zum Empirischen Software Engineering

André Janus

**Konzepte für Agile Qualitätssicherung
und -bewertung in Wartungs- und
Weiterentwicklungs-Projekten**

Shaker Verlag
Aachen 2013

Bibliografische Information der Deutschen Nationalbibliothek

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

Zugl.: Magdeburg, Univ., Diss., 2012

Copyright Shaker Verlag 2013

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-1578-2

ISSN 1618-7946

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: www.shaker.de • E-Mail: info@shaker.de

Zusammenfassung

In dieser Arbeit werden Konzepte für eine Agile Qualitätssicherung und -bewertung in Wartungs- und Weiterentwicklungs-Projekten vorgestellt. Die Agile Software-Entwicklung ist ein Trend, der sich in der Software-Industrie immer mehr durchsetzt. Zu einer professionellen Software-Entwicklung gehört aber auch eine Qualitätssicherung, sowie die Möglichkeit die Qualität von Software und auch der Software-Entwicklung selbst zu bewerten. Die in dieser Arbeit entwickelten Konzepte gehen besonders auf die in der Praxis häufig zu findenden Wartungs- und Weiterentwicklungs-Projekte ein. Da diese Arbeit einem industriellen Kontext entstammt, wurden die Konzepte auch in einem industriellen Wartungs- und Weiterentwicklungs-Projekt evaluiert.

Nach einer kurzen Einführung in den Forschungskontext in Kapitel I werden ebenfalls knapp in Kapitel II die Grundlagen der Agilen Software-Entwicklung sowie der Software-Qualität dargestellt. In Kapitel III werden die Anforderungen an eine Agile Qualitätssicherung und -bewertung gestellt, an denen die vorhandenen Forschungsansätze aus dem Kontext Agile Software-Entwicklung/Qualitätssicherung und -bewertung/Wartung und Weiterentwicklung in Kapitel IV gemessen werden. In Kapitel V werden schließlich die für diese Arbeit neu entwickelten Modelle und Methoden vorgestellt:

- Das **Agile Software Development Model (ASDM)** ist eine Modell zur formalen Charakterisierung von Agilen Vorgehensmodellen und Prozessen.
- Das **Agile Maturity Model Integration (AMMI)** bewertet die Güte Agiler Prozesse anhand der Qualitätssichernden Eigenschaften der einzelnen Agilen Praktiken.
- Der **3C-Ansatz (Continuous Integration, Measurement and Improvement)** ist eine Methode zum Software-Controlling und zur Qualitätssicherung in Agilen Entwicklungsprozessen.

In Kapitel VI werden die Modelle und Methoden in Bezug auf ein industrielles Wartungs- und Weiterentwicklungs-Projekt evaluiert. Im Kapitel VII werden die Ergebnisse zusammengefasst und Schlussfolgerungen gezogen sowie ein Ausblick auf noch zu erforschende Sachverhalte gegeben.

Diese Arbeit zeigt Zusammenhänge von Qualitätseigenschaften der Software und Software-Entwicklung und den Konzepten der Agilen Software-Entwicklung. Obwohl die Agile Software-Entwicklung selbst keine klassische Qualitätssicherung kennt, werden deren Ansätze in den entwickelten Modellen und Methoden mit denen der Agilen Software-Entwicklung kombiniert und entsprechend angepasst, so dass im Ergebnis eine "Agile Qualitätssicherung" entworfen wird. Die Anwendbarkeit wird durch den Einsatz in einem Industrieprojekt empirisch belegt. Agile Qualitätssicherung trägt zu einem ingenieurmäßigen Vorgehen in der Software-Entwicklung bei und ist ein weiterer Schritt in Richtung Industrialisierung der Software-Entwicklung.

Danke!

Während der Entstehung dieser Arbeit habe ich von vielen Seiten Unterstützung erhalten und möchte mich dafür herzlich bedanken!

Mein Dank geht vor allem an meinen Betreuer Prof. Dr. Reiner R. Dumke, von der Arbeitsgruppe Softwaretechnik am Institut für Verteilte Systeme (IVS). Trotz der räumlichen Distanz betreute er mich vorbildlich, gab mir in Telefonaten und Diskussionen wichtige Hinweise und Tipps und lenkte meine Forschungsarbeit damit in die richtigen Bahnen. Gleichzeitig hatte ich die Freiheit meine eigenen Ansätze zu entwickeln und auszubauen. Mit seiner Hilfe konnte ich meine in der Industrie gewonnenen Erkenntnisse in den richtigen Forschungskontext bringen und der wissenschaftlichen Öffentlichkeit vorstellen. Ich danke auch den anderen Mitarbeitern der Arbeitsgruppe, die ich auf Konferenzen und Aufenthalten in Magdeburg kennenlernen durfte.

Mein Dank geht ebenfalls an meine Gutachter Prof. Dr. Andreas Schmietendorf und Prof. Dr. Martin Gaedke, vor allem für die gute Zusammenarbeit bei gemeinsamen Publikationen und die Begutachtung meiner Dissertation.

Ein großer Dank geht auch an die Mitarbeiter der Firma T-Systems, Bereich Systems Integration, im Projektverbund I2 und besonders dem Projekt IBIS, in dem mir über mehrere Jahre die Möglichkeit gegeben wurde, meine tägliche Projektarbeit mit wissenschaftlicher Forschung zu verknüpfen. Meine Erkenntnisse aus der Arbeit in den Projekten bilden die wesentlichen Grundlagen für meine in dieser Arbeit entwickelten Ansätze. Gleichzeitig hatte ich die Möglichkeit meine Modelle und Methoden mit Daten aus der Industrie und im industriellen Einsatz zu erproben und zu überprüfen. Einige möchte ich hier besonders hervorheben: Jens Jäger und Christoph Nowak für gemeinsame Publikationen und die "Batman Lamp", Ralf Klemmer für das inhaltliche Korrekturlesen dieser Arbeit, Markus Leutner für die "Erfindung" von I2, Annette Lötzbeyer und Christian Syrbe für die angenehme Arbeitsatmosphäre im Büro Karlsruhe und gemeinsame Mittagessen, Tobias Deppe, Lilly Breithaupt, Sascha Janke und Dr. Ingrid Pfeiderer für die organisatorische Einbindung meiner Forschungsarbeit in die tägliche Projektarbeit.

Vor allem danke ich auch meiner Frau Erika für ihre Unterstützung und ihr Verständnis während der Promotionszeit, in der ich mehr als einmal Nächte und Wochenenden am Schreibtisch verbracht habe. Sie hat mich stets motiviert, wenn die Arbeit einmal ins Stocken geriet, sowie diese Arbeit nach Rechtschreib- und Formatierungsfehlern Korrektur gelesen. Ich danke auch meinen Eltern Ingelore und Gérard, sowie meiner Schwester Simone und meinen Großeltern Erna und Otto, die mich während der ganzen Zeit bestärkt haben, meinen Weg (weiter) zu gehen.

Karlsruhe im August 2012

André Janus

Inhaltsverzeichnis

Abbildungsverzeichnis.....	viii
Tabellenverzeichnis.....	x
Abkürzungsverzeichnis.....	xi
I Einleitung.....	1
1.1 Hintergrund und Kontext.....	1
1.2 Problematik.....	2
1.3 Fragestellungen.....	2
1.4 Stand der Forschung.....	3
1.5 Ziel der Forschungsarbeit.....	4
1.6 Nutzen und Relevanz.....	4
1.7 Methodologie und Vorgehen.....	4
1.8 Struktur der Arbeit.....	5
1.9 Zusammenfassung.....	6
II Grundlagen.....	7
1 Klassische Software-Entwicklung.....	7
1.1 Software-Produkt.....	7
1.2 Software-Entwicklungsressourcen.....	9
1.3 Software-Entwicklungsprozess.....	11
1.4 Software-Anwendung.....	12
1.5 Software-Lebenszyklus.....	14
1.5.1 Lebenszyklen und Phasen.....	14
1.5.2 Lebenszyklus-Prozesse und Ergebnisse der ISO 12207.....	15
1.5.3 Wartung und Weiterentwicklung im Software-Lebenszyklus.....	15
1.6 Klassischen Vorgehens- und Lebenszyklusmodelle.....	16
1.6.1 Sequentielle Modelle.....	16
1.6.2 Nicht-sequentielle Modelle.....	16
2 Agile Software-Entwicklung.....	17
2.1 Zentrale Begriffe in der Agilen Software-Entwicklung.....	17
2.2 Das Agile Manifest.....	18
2.2.1 Die Werte des Agilen Manifests.....	18
2.2.2 Die Prinzipien des Agilen Manifests.....	18
2.3 Agile Praktiken.....	19
2.4 Agiler Prozess.....	20
2.5 Agile Kernkonzepte.....	21
2.6 Grundannahmen in der Agilen Software-Entwicklung.....	22
2.6.1 Änderung von Anforderungen.....	22
2.6.2 Qualität als fixe Größe.....	23
2.7 Agile Vorgehensmodelle.....	25
2.7.1 eXtreme Programming (XP).....	26
2.7.2 Scrum.....	31
2.7.3 Weitere Agile Vorgehensmodelle.....	34
a Feature Driven Development (FDD).....	34
b Crystal.....	34

3 Qualitätssicherung und -bewertung	35
3.1 Der Qualitätsbegriff in der Software-Entwicklung	35
3.2 Qualitätssichten	36
3.3 Qualitätsmerkmale	36
3.4 Qualitätsdimensionen	37
3.4.1 Produktqualität	37
3.4.2 Prozessqualität	37
3.4.3 Ressourcenqualität	38
3.4.4 Formale Darstellung der Qualitätsdimensionen	38
3.5 Software-Messung	38
3.5.1 Messergebnis	38
3.5.2 Messziele	39
3.5.3 Messprozess	39
3.6 Ausgewählte Metriken und Verfahren	40
3.6.1 Lines of Code (LOC)	40
3.6.2 Zyklomatische Komplexität (McCabe)	40
3.6.3 CK-Metrics Suite	41
3.6.4 Capability Maturity Model Integration (CMMI)	41
3.7 Softwarequalitätsmanagement (SQM)	42
3.7.1 Analytische Qualitätssicherung	43
3.7.2 Konstruktive Qualitätssicherung	43
3.7.3 Organisatorische Qualitätssicherung	43
3.7.4 Total Quality Management (TQM)	43
3.7.5 Verifikation und Validierung	44
3.7.6 Testen	45
3.7.7 Formale Darstellung des Softwarequalitätsmanagement (SQM)	47
4 Wartung und Weiterentwicklung	47
4.1 Software-Wartung	48
4.1.1 Arten der Wartung	48
4.1.2 Durchführung der Wartung	50
4.1.3 Spezielle Wartungsaktivitäten	50
4.1.4 Wartbarkeit	51
4.2 Weiterentwicklung von Software (Evolution)	51
5 Zusammenfassung	52
III Anforderungen an eine Agile Qualitätssicherung und -bewertung	53
1 Anforderungen durch die Agile Software-Entwicklung	53
1.1 Berücksichtigung von Agilen Kernkonzepten	54
1.1.1 Berücksichtigung und Unterstützung der Grundannahmen	54
1.1.2 Berücksichtigung von Werten und Prinzipien	54
1.2 Bezug auf Agile Praktiken	54
1.3 Berücksichtigung des Agilen Prozesses	54
1.4 Formalisierte Zusammenfassung	55
2 Anforderungen aus Sicht der Qualitätssicherung und -bewertung	55
2.1 Berücksichtigung von Qualitätssichten, -merkmalen und -dimensionen	56
2.2 Empirisches Vorgehen – messbare Qualität	56
2.3 Software-Qualitätsmanagement (SQM)	57
2.4 Formalisierte Zusammenfassung	57
3 Anforderungen durch den Kontext Wartung und Weiterentwicklung	57
3.1 Berücksichtigung von Wartungsaktivitäten	58

3.2 Berücksichtigung von Weiterentwicklungsaktivitäten.....	58
3.3 Formalisierte Zusammenfassung.....	58
4 Spezifische Anforderungen an Modelle und Methoden.....	58
4.1 Modelle zur Qualitätsbewertung.....	58
4.1.1 Vergleichbarkeit.....	58
4.1.2 Ganzheitliche Betrachtung.....	59
4.2 Methoden zur Qualitätssicherung.....	59
4.2.1 Operationalisierbarkeit.....	59
4.3 Formalisierte Zusammenfassung.....	59
5 Bewertung anhand der Anforderungen.....	60
6 Zusammenfassung.....	62
IV Gegenwärtige Ansätze zur Agilen Qualitätssicherung und -bewertung.....	63
1 Allgemeine Ansätze aus dem Bereich Agilität und Empirie.....	63
1.1 Empirische Forschungen in der Agilen Entwicklung.....	63
1.2 Empirische Untersuchungsergebnisse von Agilen Methoden.....	65
1.3 Eine vergleichende Analyse von Agilen Methoden.....	66
2 Allgemeine Ansätze aus dem Bereich Agilität und Qualität.....	69
2.1 ISO 9001 und Agile Software-Entwicklung.....	69
2.2 Agile Softwarequalitätssicherung.....	69
2.3 Qualität im Agilen Umfeld.....	70
2.4 Agility Measurement Index.....	71
2.5 Qualitätssicherung in der Agilen Software-Entwicklung.....	72
2.6 Die Rolle der Softwarequalität in XP.....	73
2.7 Die Rolle von Rückmeldungen und Anpassung in der Agilen Entwicklung.....	74
2.8 Bewertung von Agilität.....	75
3 Spezielle Ansätze im Bereich Agile Praktiken und Qualität.....	76
3.1 Erfahrungen mit Pair Programming in einem Agilen Projekt.....	76
3.2 Ein strukturiertes Experiment mit Test-getriebener Entwicklung (TDD).....	77
3.3 Qualitätseffekte von Test-getriebener Entwicklung (TDD).....	77
3.4 Erkennung von und Reaktion auf "Bad Smells" in XP.....	79
4 Spezielle Ansätze im Bereich Agilität und Prozessqualität.....	81
4.1 XP aus CMM-Sicht.....	81
4.2 Reifegradmodell XPMM.....	82
4.3 Einschränkungen Agiler Prozesse.....	83
4.4 Softwareprozessmetriken und Agile Methoden.....	84
4.5 Agile Maturity Model (AMM).....	85
4.6 Ein iterativer Verbesserungsprozess für Agile Software-Entwicklung.....	87
5 Spezielle Ansätze im Bereich Agilität und Produktqualität.....	87
5.1 Agile Qualitätssicherung.....	88
5.2 Messung von Testqualität durch Code-Metriken.....	89
5.3 Die Entwicklung von OO-Qualitätsmetriken in Agilen Projekten.....	90
6 Spezielle Ansätze im Bereich Agilität und Software-Messung.....	90
6.1 Ein Framework zur Bewertung von XP.....	90
6.2 Software-Metriken für Agile Software-Entwicklung.....	92
6.3 Eine empirische Studie zu Stabilitätsmetriken und dem QMOOD Modell.....	92
6.4 Passende Agile Messkonzepte.....	94
6.5 Bedeutung der Softwaremessung bei Agilen Projektparadigmen.....	95

7 Spezielle Ansätze im Bereich Agilität in Wartung und Weiterentwicklung.....	96
7.1 Der Beitrag der Agilen Software-Entwicklung zur Weiterentwicklung.....	96
7.2 Agile Software-Entwicklung und Software-Wartung.....	97
7.3 Agile Software-Entwicklung für große fragile Bestands-Anwendungen.....	97
7.4 Agile Weiterentwicklung zur Verbesserung von Bestandssoftware.....	99
7.5 Agile Ansätze für die Software-Wartung.....	100
7.6 eXtreme Programming (XP) in der Software-Wartung.....	100
7.7 Ein Agiler Prozess in Wartung und Weiterentwicklung.....	101
8 Zusammenfassung.....	102
V Neue Ansätze zur Agilen Qualitätssicherung und -bewertung.....	105
1 Agile Software Development Model (ASDM).....	105
1.1 Zielsetzung des Modells.....	107
1.2 Vorgehen zur Entwicklung des Modells.....	109
1.3 Die Basis-Konzepte des ASDM.....	112
1.4 Beziehungen zwischen den Basiskonzepten.....	113
1.5 Vollständigkeit der Modell-Instantiierung.....	113
1.6 eXtreme Programming (XP) im ASDM.....	115
1.7 Scrum im ASDM.....	116
1.8 Herausforderungen.....	118
1.9 Zusammenfassung.....	118
2 Agile Maturity Model Integration (AMMI).....	119
2.1 Qualitätssichten und -dimensionen Agiler Praktiken.....	119
2.2 Der Reifegrad Agiler Entwicklungsmethoden und Vorgehensmodelle.....	122
2.3 Reifegrad (Level) 1 - iterative & incremental.....	123
2.4 Reifegrad (Level) 2 - 3-Dim. Practices.....	124
2.5 Reifegrad (Level) 3 - 2-Dim. Practices.....	124
2.6 Reifegrad (Level) 4 - 1-Dim. Practices.....	125
2.7 Reifegrad (Level) 5 - Adapting Practices.....	125
2.8 Herausforderungen.....	126
2.9 Zusammenfassung.....	127
3 Continuous Integration, Cont. Measurement, Cont. Improvement (3C).....	127
3.1 Geeignete Metriken für (interne) Software-Qualität im 3C.....	127
3.1.1 Geeignete klassische Metriken für den 3C-Ansatz.....	128
3.1.2 Geeignete Agile Metriken für den 3C-Ansatz.....	129
a Tests.....	129
b Test-Growth-Ratio.....	129
c Test-Coverage.....	130
d Broken Builds.....	130
3.2 Der 3C-Ansatz.....	133
3.2.1 Schritt 1 - Continuous Integration.....	133
3.2.2 Schritt 2 - Continuous Measurement.....	134
3.2.3 Schritt 3 - Continuous Improvement.....	137
3.3 Herausforderungen.....	139
3.4 Zusammenfassung.....	139
VI Validation der neuen Ansätze zur Agilen Qualitätssicherung und -bewertung.....	140
1 Das Agile Vorgehensmodell I2.....	140
1.1 Entwicklungs- und Releasezyklen.....	141
1.2 Konfigurationsmanagement.....	142

1.3 Multi Stage Environment.....	143
1.4 Anforderungs-, Planungs- und Fehlermanagement.....	143
1.5 Zusammenfassung.....	144
2 Anwendung der Modelle und Methoden in I2.....	144
2.1 I2 und ASDM.....	144
2.1.1 I2-Abbildung im ASDM.....	144
a Werte und Prinzipien.....	144
b Praktiken.....	145
c Rollen.....	145
d Artefakte.....	145
e Prozess.....	145
f Substitute.....	145
g Beziehungen.....	147
2.1.2 Ergebnisse.....	147
2.2 I2 und AMMI.....	148
2.2.1 Umfrage zu Agilen Praktiken.....	148
2.2.2 Ergebnisse.....	149
2.3 I2 und 3C.....	152
2.3.1 Wartung und Weiterentwicklung in I2.....	152
2.3.2 Typisierung von Java-Generics.....	153
2.3.3 Ergebnisse.....	153
2.4 Zusammenfassung.....	153
3 Prüfung der Modelle und Methoden gegen die Anforderungen.....	153
3.1 ASDM.....	154
3.2 AMMI.....	157
3.3 3C.....	160
3.4 Zusammenfassung.....	163
VII Zusammenfassung und Ausblick.....	164
1.1 Zusammenfassung.....	164
1.1.1 Fazit ASDM.....	164
1.1.2 Fazit AMMI.....	165
1.1.3 Fazit 3C.....	165
1.2 Ergebnisse.....	165
1.2.1 Agile Qualitätsbewertung.....	166
1.2.2 Agile Qualitätssicherung.....	166
1.2.3 Kontext Wartung und Weiterentwicklung.....	166
1.2.4 Qualitätseigenschaften Agiler Software-Entwicklung.....	167
1.2.5 Schlussfolgerungen.....	167
1.3 Ausblick.....	167
1.3.1 Weiterentwicklung der Modelle und Methoden.....	167
1.3.2 Weitere Forschungsfragen.....	168
Literaturverzeichnis.....	169
Anhang 1 - Extreme Feedback Device: "The Batman Lamp".....	176

Abbildungsverzeichnis

Abb. 1 - Struktur der Arbeit.....	6
Abb. 2 - Software-Produkt (aus [Dumke2005]).....	9
Abb. 3 - Software-Ressourcen (aus [Dumke2005]).....	10
Abb. 4 - Software-Prozess (aus [Dumke2005]).....	12
Abb. 5 - Software-Anwendung (aus [Dumke2005]).....	13
Abb. 6 - Software-Lebenszyklus-Phasen (angelehnt an [Dumke2003]).....	14
Abb. 7 - Lebenszyklus-Prozesse und Ergebnisse der ISO 12207.....	15
Abb. 8 - Werte, Prinzipien und Praktiken (basierend auf [Abrahamsson2005]).....	20
Abb. 9 - Typischer Agiler Prozess (mit Agilen Praktiken).....	21
Abb. 10 - Änderung der Anforderungen (aus [iteratec2012]).....	22
Abb. 11 - Cost of Change (aus [Beck1999_2]).....	23
Abb. 12 - Das Magische Quadrat - klassische Sicht.....	24
Abb. 13 - Das Magische Quadrat - Agile Sicht.....	25
Abb. 14 - Werte, Prinzipien und Praktiken in Agilen Methoden.....	26
Abb. 15 - Funktionsumfang und Technische Schulden (basierend auf [Fowler2000]).....	28
Abb. 16 - eXtreme Programming (XP) in der Übersicht (aus [Beck1999_2]).....	29
Abb. 17 - eXtreme Programming (XP) Lebenszyklus (aus [Wells2006]).....	30
Abb. 18 - eXtreme Programming (XP) - Praktiken, Rollen und Artefakte	30
Abb. 19 - Burndown-Chart des I2-Projekts IUCCA [Koto2012].....	33
Abb. 20 - Scrum - Praktiken, Rollen und Artefakte	33
Abb. 21 - Phasenmodell im FDD (aus [ITAgile2012]).....	34
Abb. 22 - Qualitätsmerkmale des SW-Produkts nach ISO 25000 (aus [ISO25000_2012]).....	37
Abb. 23 - Goal-Question-Metric (aus [GQM2012]).....	39
Abb. 24 - Messprozess nach ISO 15939 (aus [Dumke2003]).....	40
Abb. 25 - QS-Varianten.....	43
Abb. 26 - Anwendung der QS-Varianten.....	44
Abb. 27 - Verifikation und Validierung in den Entwicklungs-Phasen (aus [Dumke2003]).....	45
Abb. 28 - Software-Wartung (aus [Dumke2005]).....	50
Abb. 29 - Software-Qualität im Kontext dieser Arbeit.....	52
Abb. 30 - Anforderungen durch die Agile Software-Entwicklung.....	53
Abb. 31 - Anforderungen aus Sicht der Software-Qualität.....	56
Abb. 32 - Anforderungen durch Wartung und Weiterentwicklung.....	57
Abb. 33 - Anforderungen an Modelle und Methoden.....	59
Abb. 34 - Zusammenfassung der Anforderungen.....	62
Abb. 35 - Evolutionäre Entwicklung Agiler Methoden (aus [Abrahamsson2003]).....	67
Abb. 36 - Mapping von XP Praktiken auf CMM Prozessgebiete.....	82
Abb. 37 - Zuordnung von XP Praktiken zu XPMM Reifegraden.....	83
Abb. 38 - Agile Maturity Model (AMM) (aus [Patel2009]).....	86
Abb. 39 - Begriffe von XP und Scrum gegenübergestellt.....	106
Abb. 40 - Vergleich von Begriffen in XP und Scrum.....	107
Abb. 41 - Charakterisierung und Vergleichbarkeit von XP und Scrum.....	108
Abb. 42 - Gemeinsam Eigenschaften und Basis-Konzepte.....	108
Abb. 43 - Basiskonzepte des Agilen Manifests	112
Abb. 44 - Basiskonzepte des ASDM	113
Abb. 45 - Basiskonzepte des XP	116
Abb. 46 - Basiskonzepte von Scrum.....	118
Abb. 47 - Explizite und implizite Basiskonzepte des ASDM.....	119
Abb. 48 - Agile Maturity Model Integration (AMMI).....	123
Abb. 49 - Level 1.....	123

Abb. 50 - Level 2.....	124
Abb. 51 - Level 3.....	124
Abb. 52 - Level 4.....	125
Abb. 53 - Level 5.....	125
Abb. 54 - Zuordnung Agiler Praktiken zu Qualitäts-Dimensionen.....	126
Abb. 55 - Metriken im Tool Sonar.....	128
Abb. 56 - Test-Coverage im Tool Cobertura	130
Abb. 57 - Broken Builds im Tool CruiseControl.....	132
Abb. 58 - Continuous Integration (blau umrahmt).....	134
Abb. 59 - Continuous Measurement (blau umrahmt).....	135
Abb. 60 - Cockpit - Darstellung der Messergebnisse.....	136
Abb. 61 - Vorgehen bei GQM (aus [GQM2012]).....	138
Abb. 62 - Continuous Improvement (blau umrahmt).....	138
Abb. 63 - I2-Praktiken (aus [Leutner2001]).....	141
Abb. 64 - I2-Entwicklungs- und Releasezyklen (aus [Leutner2001]).....	142
Abb. 65 - I2-Konfigurationsmanagement (aus [Leutner2001]).....	142
Abb. 66 - I2-Multi Stage Environment (aus [Leutner2001]).....	143
Abb. 67 - Basiskonzepte von I2.....	147
Abb. 68 - Rollen der Befragten im Projekt.....	149
Abb. 69 - Testing.....	150
Abb. 70 - Continuous Integration.....	150
Abb. 71 - Retrospektive.....	151
Abb. 72 - Zusammenfassung der Arbeit.....	164
Abb. 73 - Kontext der entwickelten Modelle und Methoden.....	166
Abb. 74 - IP-fähige Steckerleiste.....	176
Abb. 75 - Lampe 1.....	176
Abb. 76 - Lampe 2.....	177
Abb. 77 - Symbol des Quality Managers an der Büro-Decke.....	177
Abb. 78 - Piktogramm des Quality Managers.....	177

Tabellenverzeichnis

Tab. 1 - Tabelle der Bewertungssymbole.....	60
Tab. 2 - Tabelle der Anforderungen.....	61
Tab. 3 - Research Methods.....	63
Tab. 4 - Type of Agile Method and Studies.....	64
Tab. 5 - Topics and Paper.....	64
Tab. 6 - Maturity of Research.....	65
Tab. 7 - Goals for Research.....	65
Tab. 8 - Zusammenfassung der Ergebnisse.....	68
Tab. 9 - Industrielle Fallstudien.....	78
Tab. 10 - Fallstudien aus der akademischen Forschung.....	79
Tab. 11 - "Bad Smells".....	81
Tab. 12 - Zusammenfassende Bewertung der Ansätze 1-2.....	103
Tab. 13 - Zusammenfassende Bewertung der Ansätze 2-2.....	104
Tab. 14 - Die Werte des Agilen Manifests.....	109
Tab. 15 - Agile Praktiken und Qualitätssichten und -dimensionen	121
Tab. 16 - Vergleich Agiler Reifegrad- und Bewertungsmodelle.....	152
Tab. 17 - Anforderungserfüllung durch ASDM.....	156
Tab. 18 - Anforderungserfüllung durch AMMI.....	159
Tab. 19 - Anforderungserfüllung durch 3C.....	162

Abkürzungsverzeichnis

3C	Continuous Integration, Continuous Measurement, Continuous Improvement
AMMI	Agile Maturity Model Integration
ASDM	Agile Software Development Model
CI	Continuous Integration
CK	Chidamber & Kemere (Metriken/Metricsuite)
CMM(I)	Capability Maturity Model (Integration)
DeTo	Deployment Tool
EAI	Enterprise Application Integration
FDD	Feature-Driven-Development
GQM	Goal-Question-Metric
GUI	Graphical User Interface
I2	Industrialisierte Iteration
IBIS	Internet-Based Buyback Information System
IUCCA	Intranet-Based Used Car Center Application
INT	Integration
JSP	Java Server Pages
KoTo	KontierungsTool
LOC	Lines of Code
PROD	Produktion
Q-Gate	Quality-Gate
QA	Quality Assurance
QS	Qualitätssicherung
ShaRM	SharedRequirementsManagement
SQM	Softwarequalitätsmanagement
SVN	Subversion
SW	Software
TDD	Test-Driven-Development
TQM	Total Quality Management
VCS	Version-Control-System
XP	eXtreme Programming