

Metrik-basierte Auswertung von Software-Entwicklungsarchiven zur Prozessbewertung

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften
der RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Dipl.-Inform. Holger Schackmann
aus Bitburg

Berichter: Prof. Dr. rer. nat. Horst Lichter
Prof. Dr. rer. nat. Kurt Schneider

Tag der mündlichen Prüfung: 6. Mai 2010

Diese Dissertation ist auf den Internetseiten der
Hochschulbibliothek online verfügbar.

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Software Engineering
RWTH Aachen University

Band 7

Holger Schackmann

Metrik-basierte Auswertung von Software-Entwicklungsarchiven zur Prozessbewertung

Shaker Verlag
Aachen 2010

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.: D 82 (Diss. RWTH Aachen University, 2010)

Das Bildmotiv/Umschlag basiert auf einer Vorlage von Joaquim Alves Gaspar.
Quelle: http://commons.wikimedia.org/wiki/File:Using_the_caliper_new.gif

Copyright Shaker Verlag 2010

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

Printed in Germany.

ISBN 978-3-8322-9405-2

ISSN 1869-9170

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

Sonja gewidmet

Kurzdarstellung

Die Entwicklung und Wartung von komplexen Softwareprodukten erfordert Transparenz bei Prozessen und Kosten. Messungen im Entwicklungsprozess sind ein Mittel um diese Transparenz zu schaffen. Sie ermöglichen die Bewertung und kontinuierliche Überwachung von Prozessen. Dem versprochenen Nutzen stehen die Kosten der Messungen gegenüber. Systematisches Messen im Prozess verlangt eine an das Unternehmen angepasste Infrastruktur zur Erfassung und Aufbereitung der Messergebnisse. Der dazu nötige Aufwand stellt gerade für kleine und mittelgroße Organisationen eine Hürde dar.

Viele Daten aus Entwicklungsprozessen werden heute schon routinemäßig erfasst, beispielsweise in Issue-Tracking-Systemen und Konfigurationsmanagement-Systemen. Die in solchen Software-Entwicklungsarchiven ohnehin gesammelten Daten spiegeln den Ablauf von Teilen der Entwicklungsprozesse wieder. Ihre Auswertung stellt also potentiell eine kostengünstige Alternative gegenüber einer manuellen Erfassung von Statusinformationen dar. In der industriellen Praxis wird diese Datenbasis allerdings nur unzureichend für Messungen zum Prozess genutzt. Gründe dafür sind zum einen fehlende methodische Unterstützung für die Entwicklung von Metriken, zum anderen unflexible Werkzeugunterstützung.

Die in dieser Arbeit entwickelten Methoden und Werkzeuge zielen darauf ab, die in Software-Entwicklungsarchiven gesammelten Daten besser für die Bewertung von Prozessen nutzbar zu machen. Kern der entwickelten Lösung ist die deklarative Sprache ITMS zur Spezifikation von Metriken auf Issue-Tracking-Systemen. Diese Sprache ermöglicht eine kompakte und präzise Beschreibung von Metriken auf einem hohen Abstraktionsniveau. Die vorgestellte Referenzimplementierung der Sprache kann flexibel an unterschiedliche Software-Entwicklungsarchive angebunden werden. Weiterhin sind die in ITMS spezifizierten Metriken leicht anpassbar. Dies ermöglicht ein iteratives Verfahren zur Entwicklung und Validierung von Metriken, welches in dieser Arbeit beschrieben wird.

Um eine systematische Interpretation von Messergebnissen zu erleichtern, wird ein Metamodell für Qualitätsmodelle vorgestellt. Ein solches Qualitätsmodell stellt den Bezug zwischen subjektiven Qualitätsmerkmalen und den Messungen dar, und ist operativ zur Prozessbewertung einsetzbar. Diese Konzepte wurden in einem Qualitätsmodell-Editor und Auswertungs-Werkzeug umgesetzt. Das Auswertungswerkzeug unterstützt die Klassifikation von Messergebnissen auf Basis der Werteverteilung empirischer Vergleichsdaten. Dies ermöglicht eine pragmatische und realistische Einordnung der Messergebnisse.

Anwendbarkeit und Skalierbarkeit der entwickelten Methoden und Werkzeuge wird am Beispiel der Prozessbewertung in der industriellen Softwareentwicklung als auch bei der Bewertung von Prozessen im Open Source Bereich demonstriert.

Le vrai voyage ce n'est pas de chercher des
nouveaux paysages mais un nouveau regard.

(Marcel Proust, 1871-1922)

Danksagung

Softwareentwicklung als Expedition ist eine gebräuchliche Metapher. Dieses Bildnis lässt sich auch auf eine Forschungsarbeit übertragen. Genauso wie eine Expedition verfolgt eine Forschungsarbeit ein Ziel. Sie verläuft in verschiedenen Etappen, und gelegentlich ist es notwendig bei Problemen zu verweilen. Die Reiseroute lässt sich nicht vollständig planen, da ein neues Territorium betreten wird. Die Planung muss an die vor Ort anzutreffenden Verhältnisse angepasst werden. An dieser Stelle möchte ich allen danken, die zum Gelingen dieser Expedition beigetragen haben.

Mein herzlicher Dank gilt Prof. Horst Lichter für die konstruktive inhaltliche Begleitung und auch dafür, dass er schon im Studium das richtige Training für eine solche Expedition vermittelt hat. Prof. Kurt Schneider danke ich für die bereitwillige Übernahme des Zweitgutachtens.

Viele Ideen aus dieser Arbeit sowie deren praktische Evaluierung wurden durch die langfristige Zusammenarbeit mit der KISTERS AG möglich. Mein Dank gilt Dr. Heinz-Josef Schlebusch und Dr. Juliane Ruhland, stellvertretend für alle Mitarbeiter der KISTERS AG, mit denen ich zusammenarbeiten konnte. Klaus Kisters danke ich für die Unterstützung des Kooperationsprojekts.

In einer Reihe von Diplom- und Masterarbeiten wurden nicht nur wichtige Beiträge zu verschiedenen Forschungsvorhaben geleistet. Bei der Begleitung einer solchen Abschlussarbeit findet auch immer ein voneinander Lernen statt. Mein Dank gilt Georg Richlofsky, Henning Schäfer, Martin Jansen, Christoph Lischkowitz, Netiya Ounjai, Sofia Adamanova, Andreas Hermanns und nicht zuletzt Lars Grammel, der die grundlegende Arbeit zu ITMS geleistet und den Anstoß für das Open Source Projekt *BugzillaMetrics* gegeben hat. Eine besondere Freude ist es, dass mich Veit Hoffmann, Malek Obaid, Andreas Ganser und Matthias Vianden auch als Kollegen am Lehr- und Forschungsgebiet Informatik 3 weiter begleiteten. Desgleichen möchte ich mich bei Bärbel Kronewetter, Dr. Thomas von der Maßen, Thomas Weiler und Dr. Alexander Nyßen für die vielfache Unterstützung und die freundschaftliche Atmosphäre bedanken.

Ausdrücklich danke ich meinen Eltern, sowie meinen Geschwistern Heike, Frank und Jens, die in vieler Hinsicht zu der nötigen Ausrüstung für diese Expedition beigetragen haben. Besonders danke ich meiner Freundin Sonja für ihre Liebe und ihre Unterstützung.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Beitrag dieser Arbeit	3
1.2. Überblick über die Arbeit	3
2. Grundlagen	5
2.1. Grundbegriffe	5
2.1.1. Metriken und Messung	5
2.1.2. Messprozess	7
2.1.3. Zielgerichtete Definition von Metriken	7
2.1.4. Software-Entwicklungsarchive	10
2.2. Qualitätsmodelle	11
2.2.1. Historische Entwicklung	13
2.2.2. Werkzeugunterstützung für Qualitätsmodelle	15
2.3. Überwachung und Bewertung von Prozessen	16
2.3.1. Projekt- und Produktmanagement	16
2.3.2. Prozessbewertung	18
2.3.3. IT-Controlling	20
2.3.3.1. IT Service Management	21
2.3.3.2. IT Governance	24
2.3.4. Bewertung von Open Source Softwareentwicklung	25
2.4. Auswertung von Software-Entwicklungsarchiven	26
2.4.1. Kategorisierung von Auswertungsansätzen	26
2.4.2. Auswertungen zur Analyse von Entwicklungsprozessen	28
2.4.3. Werkzeuginfrastrukturen zur Metrikauswertung	29
3. Ziele	31
3.1. Motivation	31
3.2. Fragestellungen	33
3.3. Überblick über den entwickelten Lösungsansatz	34
4. Deklarative Spezifikation von Metriken	37
4.1. Motivation	37
4.2. Verwandte Arbeiten	39
4.2.1. Attributgrammatiken	39
4.2.2. Abfragesprachen für relationale Datenbanken	39
4.2.3. Abfragesprachen für OLAP-Datenbanken	40
4.2.4. Weitere Abfragesprachen	41
4.2.5. Modell-zu-Modell Transformationsprachen	42
4.2.6. Domänenspezifische Sprachen	42

Inhaltsverzeichnis

4.2.7.	Diskussion	43
4.3.	ITMS: Eine Sprache zur Spezifikation von Metriken	44
4.4.	Anforderungen an die Sprache ITMS	44
4.4.1.	Grundbegriffe	45
4.4.1.1.	Attribute	45
4.4.1.2.	Zustandsfilter	45
4.4.1.3.	Ereignisfilter	46
4.4.1.4.	Zeitreihen	46
4.4.2.	Elemente einer Metrik-Spezifikation	47
4.4.2.1.	Berechnungszeitraum und Zeitgranularität	47
4.4.2.2.	Basisfilter	47
4.4.2.3.	Gruppierung	47
4.4.2.4.	Bewertungsverfahren	48
4.4.2.5.	Gruppenwertberechnung	49
4.4.2.6.	Fixierte Attribute	49
4.4.3.	Beispiel	50
4.4.4.	Überblick über den Ablauf einer Metrikberechnung	51
4.4.5.	Bewertungsverfahren im Detail	52
4.4.5.1.	Das Bewertungsverfahren „Zählen von Ereignissen in einer Zeitperiode“	52
4.4.5.2.	Das Bewertungsverfahren „Zählen von Ereignissen bis zu einem Ereignis“	53
4.4.5.3.	Das Bewertungsverfahren „Intervalllänge“	53
4.4.5.4.	Das Bewertungsverfahren „Verweilzeit“	54
4.4.6.	Zusammenfassung der Anforderungen	55
4.5.	Aufbau von ITMS	55
4.5.1.	Zustandsfilter	58
4.5.2.	Gruppenwertberechnung	60
4.5.3.	Bewertungsverfahren	60
4.5.3.1.	Gewichtung	64
4.5.3.2.	Ereignisfilter	65
4.5.4.	Beispiel	67
4.6.	Formale Semantik von ITMS	69
4.6.1.	Überblick	70
4.6.2.	Grundlegende Definitionen	70
4.6.2.1.	Zeit und Zeitreihen	70
4.6.2.2.	Änderungsanträge	72
4.6.3.	Filter	73
4.6.3.1.	Zustandsfilter	73
4.6.3.2.	Ereignisfilter	74
4.6.4.	Metrik	75
4.6.4.1.	Gruppenwertberechnungen	76
4.6.4.2.	Gewichtungsfunktion	76
4.6.5.	Bewertungsverfahren	77
4.6.5.1.	Zählen von Ereignissen in einer Zeitperiode	77
4.6.5.2.	Zählen von Ereignissen bis zu einem Ereignis	78
4.6.5.3.	Intervalllänge	78

4.6.5.4.	Verweilzeit	79
4.6.6.	Weitere Elemente der Metrik-Spezifikation	80
4.6.7.	Beispiel	81
4.6.8.	Zusammenfassung	82
5.	Entwicklung von Metriken	83
5.1.	Validierung von Metriken	84
5.2.	Ein Vorgehen zur Entwicklung von Metriken	87
5.2.1.	Identifikation von Qualitätsmerkmalen	87
5.2.2.	Identifikation von Qualitätsattributen	88
5.2.3.	Definition von Qualitätsindikatoren	88
5.2.4.	Interpretation basierend auf empirischen Daten	90
6.	Ein Metamodell für Qualitätsmodelle	91
6.1.	Wiederkehrende Konzepte beim Aufbau von Qualitätsmodellen	92
6.2.	Anforderungen	94
6.3.	Metamodell	95
6.4.	Fazit	99
7.	Werkzeugunterstützung	101
7.1.	Anforderungen	101
7.1.1.	Metrikberechnungskomponente	105
7.1.2.	Metrik-Abfrage-Werkzeug	106
7.1.3.	Qualitätsmodell-Werkzeuge	107
7.2.	Architektur	108
7.2.1.	Überblick	109
7.2.2.	Metrikberechnungskomponente	110
7.2.2.1.	Anbindung eines konkreten Issue-Tracking-Systems	112
7.2.2.2.	Berechnungsalgorithmus	112
7.2.3.	Anbindung von Konfigurationsmanagement-Systemen	113
7.2.4.	Web-Anwendung zur Abfrage von Metriken	114
7.2.5.	Editor und Auswertungswerkzeug für Qualitätsmodelle	118
7.2.5.1.	Qualitätsmodell-Editor	119
7.2.5.2.	Auswertungswerkzeug	121
7.3.	Testumgebung	123
8.	Evaluierung	125
8.1.	Praktische Anwendung	125
8.1.1.	Bewertung von Open Source Entwicklungsprozessen	125
8.1.2.	Analyse von industriellen Softwareentwicklungsprozessen	131
8.2.	Bewertung der Werkzeugunterstützung	133
8.2.1.	Funktionalität	135
8.2.2.	Zuverlässigkeit	136
8.2.3.	Verwendbarkeit	137
8.2.4.	Effizienz	138
8.2.5.	Wartbarkeit	139
8.2.5.1.	Analysierbarkeit und Änderbarkeit	139

Inhaltsverzeichnis

8.2.5.2. Testbarkeit und Stabilität	142
8.2.6. Portabilität	142
8.3. Bewertung der Sprache ITMS	143
8.3.1. Erlernbarkeit	143
8.3.2. Benutzbarkeit	146
8.3.3. Ausdrucksstärke	146
8.3.4. Wiederverwendbarkeit	148
8.3.5. Entwicklungskosten	148
8.3.6. Zuverlässigkeit	148
8.4. Bewertung des Ansatzes zur Qualitätsmodellierung	149
8.4.1. Erlernbarkeit	149
8.4.2. Benutzbarkeit	150
8.4.3. Ausdrucksstärke	151
8.4.4. Wiederverwendbarkeit	152
8.4.5. Entwicklungskosten	152
8.4.6. Zuverlässigkeit	153
8.5. Datenqualität in Software-Entwicklungsarchiven	153
8.5.1. Verwandte Arbeiten	154
8.5.2. Erfahrungen aus den Fallstudien	155
8.5.3. Umgang mit mangelnder Datenqualität	155
8.6. Resümee	156
8.6.1. Offene Fragen	158
9. Zusammenfassung und Ausblick	159
A. Anhang	163
A.1. Beispiel-Metriken zur Prozessqualität in Eclipse	163
A.2. Beispiel eines Qualitätsmodells in der industriellen Softwareentwicklung	166
Literaturverzeichnis	169
Stichwortverzeichnis	190

Abbildungsverzeichnis

1.1. Kosten von Software-Messungen [NAS95]	2
2.1. Messung nach Kriz ([ED07], siehe auch [Zus97, Kri88])	6
2.2. Messprozess nach ISO/IEC 15939 [ISO07b]	8
2.3. Beispiel für mögliche Zustandswechsel von Änderungsanträgen [Bug09a]	12
2.4. Projektmanagement als Regelkreis (nach [Buh04])	17
2.5. Prozesse im IT Service Management [ISO05a]	22
3.1. Überblick über den entwickelten Lösungsansatz	34
4.1. Die Begriffe Zeitperiode und Berechnungszeitraum	47
4.2. Beispiel einer Metrikberechnung	51
4.3. Aufbau einer Metrik-Spezifikation (Legende siehe Tabelle 4.2)	57
4.4. Aufbau eines Zustandsfilters in einer Metrik-Spezifikation (Le- gende siehe Tabelle 4.2)	59
4.5. Aufbau einer Gruppenwertberechnung (Legende siehe Tabelle 4.2)	61
4.6. Aufbau der Bewertungsverfahren (Legende siehe Tabelle 4.2)	62
4.7. Aufbau eines Ereignisfilters (Legende siehe Tabelle 4.2)	66
4.8. Beispiel einer ITMS Metrik-Spezifikation: Durchschnittliche An- zahl der Bearbeiterwechsel eines Änderungsantrags	68
4.9. Überblick der verwendeten Funktionen zur Beschreibung der Semantik	71
5.1. Zusammenhang zwischen Messmethode und Messprozedur	86
5.2. Vorgehen zur Entwicklung von Metriken mit ITMS	87
6.1. Metamodell für Qualitätsmodelle	97
6.2. Kombinationsfunktion und Umformfunktion	98
7.1. Anwendungsfalldiagramm der QMetric-Werkzeug-Suite	102
7.2. Feature-Diagramm für die Metrikberechnungskomponente (Le- gende siehe Tabelle 7.1)	104
7.3. Feature-Diagramm für das Metrik-Abfrage-Werkzeug (Legende siehe Tabelle 7.1)	106
7.4. Feature-Diagramm für die Qualitätsmodell-Werkzeuge (Legende siehe Tabelle 7.1)	108
7.5. Überblick über die Architektur der QMetric-Werkzeug-Suite	109
7.6. Paketdiagramm: Metrikberechnungskomponente und Adapter für Bugzilla	110

Abbildungsverzeichnis

7.7. Komponentendiagramm: Beteiligte Systeme im Kontext der Auswertung von Bugzilla und Subversion	115
7.8. Dialog zur Auswahl häufig genutzter Metriken	115
7.9. Dialog zur Konfiguration des Bewertungsverfahrens Zählen von Ereignissen (<i>CountEvents</i>)	116
7.10. Paketdiagramm der Web-Anwendung zur Abfrage von Metriken	117
7.11. Komponentendiagramm: Editor und Auswertungswerkzeug für Qualitätsmodelle	119
7.12. Der Qualitätsmodell-Editor	119
7.13. Paketdiagramm zum Qualitätsmodell-Editor	120
7.14. Anzeige von Ergebniszeitreihen im Auswertungswerkzeug	121
7.15. Paketdiagramm zum Auswertungswerkzeug	122
8.1. Qualitätsmodell zu Eclipse strukturiert nach den Schritten im Change Request Prozess [SSL09]	127
8.2. Qualität der Klassifikation von Änderungsanträgen bewertet auf einer Ordinalskala von 1 bis 4	130
8.3. Streudiagramm zur Visualisierung unterschiedlicher Planungsansätze [Ric09]	134
8.4. Beispiel einer Flusskarte [Ric09]	134
8.5. Distanzgraph nach Martin [Mar94]	141
A.1. Qualitätsmodell zur Bewertung eines industriellen Software-Entwicklungsprozesses (Teil 1) [Ric09]	167
A.2. Qualitätsmodell zur Bewertung eines industriellen Software-Entwicklungsprozesses (Teil 2) [Ric09]	168

Tabellenverzeichnis

2.1. Beispiele für Metriken aus ITIL	23
4.1. Metrik-Spezifikation als Pseudocode	50
4.2. Diagrammelemente zur Darstellung eines XML Schemas	56
7.1. Verwendete Modellierungselemente im Feature-Diagramm	105
7.2. Erreichte Überdeckung bei dem Paket <i>org.qmetric.core.algorithm</i> [Cov]	124
8.1. Metriken zum Qualitätsmodell aus Abbildung 8.1 [SSL09]	128
8.2. Codemetriken für QMetric (Stand 01.09.2009)	140
8.3. Zusammenfassende Bewertung der entwickelten Konzepte und Werkzeuge	157

