Ein Kosten-Nutzen-Modell für die Softwareprüfung

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Tilmann Hampp

aus Ludwigsburg

Hauptberichter:	Prof. Dr. rer.nat. J. Ludewig
Mitberichter:	O. Univ.-Prof. Dr. R. Mittermeir
Tag der mündlichen Prüfung:	19. Juli 2010

Institut für Softwaretechnologie der Universität Stuttgart

2010

Berichte aus der Softwaretechnik

Tilmann Hampp

Ein Kosten-Nutzen-Modell für die Softwareprüfung

D 93 (Diss. Universität Stuttgart)

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.: Stuttgart, Univ., Diss., 2010

Printed in Germany.

ISBN 978-3-8322-9567-7
ISSN 1433-9986

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

Danksagung

Zu dieser Arbeit haben viele Personen beigetragen, denen ich an dieser Stelle danken möchte. Mein ganz besonderer Dank gilt meinem Doktorvater Prof. Jochen Ludewig, der diese Arbeit ermöglicht hat. Seine Sicht auf die Software-Qualitätssicherung, seine Anregungen und seine Kritik haben diese Arbeit geprägt. Mein besonderer Dank geht auch an Prof. Roland Mittermeir für sein Interesse an der Arbeit, die freundliche Übernahme des Zweitgutachtens und die wertvollen Hinweise. Prof. Erhard Plödereder danke ich für seinen Einsatz als Zweitprüfer.

Herzlich danken möchte ich auch allen Kollegen und Kolleginnen der Abteilung Software Engineering für die angenehme Zusammenarbeit, die ermutigenden Gespräche und die anregenden Diskussionen.

Die Partner in der Industrie haben mir in den Gesprächen und mit den Projektdaten einen tiefen Einblick in ihre Projekte gewährt. Vielen Dank! Ebenso danke ich den Teilnehmern und Hilfskräften des Software-Praktikums 2007 für die Datenerhebung und die Unterstützung bei der Analyse. Mein Dank geht auch an die Studenten, die durch ihre Diplom- und Studienarbeiten und Fachstudien zu dieser Arbeit beigetragen haben.

Nicht zuletzt danke ich allen, die mich während der Arbeit begleitet haben, für ihr Vertrauen, ihre Geduld und ihre Unterstützung.

Zusammenfassung

Prüfungen können große Teile des Budgets eines Software-Projekts aufzehren, erlauben aber, die Produktqualität zu beurteilen und zu verbessern. Sie dürfen nicht vernachlässigt werden, da Defizite der Produktqualität nach Projektende teuer werden können. Projektleiter und Verantwortliche für die Qualität müssen bereits in der Planung über Prüfungen entscheiden. Sie sind in einer schwierigen Situation, weil sie dabei viele komplexe und langfristig wirkende Entscheidungen über Prüfungen und über einzelne Parameter der Prüfungen treffen müssen. Die Kosten der Prüfungen sind früh sichtbar und messbar. Im Gegensatz dazu wird der Nutzen durch schwierig zu messende Qualitätsverbesserungen erreicht, die zu langfristigen Einsparungen führen. Zusätzlich hängen Kosten und Nutzen von der Projektsituation ab. Für jedes Projekt ist darum ein individueller Kompromiss zwischen den Kosten für Prüfungen und ihrem Nutzen nötig, so dass minimale Gesamtkosten erreicht werden.

Um diese Entscheidungen zu unterstützen, wird in dieser Arbeit ein Kosten-Nutzen-Modell für Softwareprüfungen, CoBe, entwickelt und validiert. Mit diesem Modell kann untersucht und prognostiziert werden, wie sich Entscheidungen über Prüfungen und über einzelne Parameter der Prüfungen auswirken. Dazu werden die Entscheidungen und die Projektsituation durch Modelleingaben dargestellt. Die Modellresultate sind die Wirkungen dieser Entscheidungen: die Kosten, die durch die Prüfung entstehen, und der daraufhin erreichte Nutzen durch eingesparte Kosten. Kosten und Nutzen zeigen sich während des Projekts, während der Wartung des Produkts und beim Einsatz des Produkts. Damit Kosten und Nutzen abgewogen und Gesamtkosten minimiert werden können, werden die Modellresultate als Geldwerte berechnet. Zur Projektplanung werden Kosten und Nutzen durch Aufwand, Dauer und Personalbedarf einzelner Aktivitäten dargestellt. Dazu enthält CoBe feingranulare Prüfungsmodelle aus einzelnen, quantitativen Wirkungszusammenhängen.

Die Validierung des Modells erfolgte mit Daten aus Software-Projekten. Dabei wurden einzelne Zusammenhänge und das gesamte Modell mit Daten aus über 20 studentischen Projekten geprüft. CoBe ist mit Daten aus zwei iterativen Industrieprojekten mit umfangreicher, paralleler Entwicklung validiert. Das Modellverhalten wird durch Sensitivitätsanalyse untersucht, zusätzlich wird das Kosten-Optimum analysiert. Die Validierung zeigt, dass CoBe ausreichend genau beschreibt, wie sich Entscheidungen über Prüfungen auswirken. Da die Resultate der studentischen Projekte deutlich streuen, ergibt sich eine gewisse Abweichung zwischen den Projektresultaten und den Modellresultaten. Die Resultate sind für die beiden Industrieprojekte genauer. Deutlich wird, dass CoBe für eine bestimmte Umgebung kalibriert werden muss, damit die Resultate ausreichend genau sind. Dazu sind wenige Daten aus abgeschlossenen Software-Projekten notwendig. Die Daten sind oft verfügbar, da sie häufiger als andere Daten erhoben werden. Die Validierung zeigt, dass CoBe gut verallgemeinerbar ist. Die Daten, die für den Einsatz von CoBe notwendig sind, sind in Projekten verfügbar, können gemessen oder erfragt werden.

Abstract

Software quality assurance can consume large parts of a software project's budget. On the other hand, quality assurance permits product quality to be assessed and improved. Cutting quality assurance investments may lead to increased costs after delivery.

Project managers and quality managers have to decide on quality assurance while planning and running a project. They have to make many complex and far-reaching decisions on reviews, tests, and their parameters without having the necessary information available. In particular, quality-related information is hard to get because quality improvements appear as long-term savings, whereas the costs of reviews and tests can be measured early on. As every project has its own special characteristics, a tailored trade-off between costs and benefits of quality-assurance activities is needed to minimise total costs.

In this work CoBe, a quantitative cost-benefit model to support these decisions, is developed and validated. CoBe is able to analyse and predict the effects that decisions on quality assurance activities and on their parameters will have. To do so, decisions on quality assurance and project characteristics are modelled as inputs. The output of the model consists of the costs and benefits resulting from these decisions. Costs and benefits occur during the project, in maintenance, and during product usage. For the purpose of comparing costs and benefits and minimising overall costs, the model results are expressed in monetary values. For project planning, costs and benefits are expressed as effort, duration, and staff of single activities. For calculating the results, CoBe uses fine-grained models built on single quantitative relationships.

CoBe was validated against real-world software project data. Single relationships and the entire model were examined using data from more than 20 student projects. CoBe was validated using data from two iterative industry projects that used extensive parallel development. Model behaviour is subjected to sensitivity analysis, and the optimum cost is analysed. The validation shows that CoBe describes the effects of decisions on quality assurance sufficiently accurate. As student projects scatter, model results differ from project results up to a certain extent. Results are more accurate for industry projects. It is evident that CoBe needs to be calibrated for a certain environment. However, only few data from past projects is required for this. The necessary data is readily available in most projects. Results indicate that CoBe is generalisable, and that the necessary data is either available in the projects or can be measured or obtained by enquiry.