

Analyse von Testprozessen in der Industrie

Rainer Schmidberger
rainer.schmidberger@informatik.uni-stuttgart.de
Universität Stuttgart, Institut für Softwaretechnologie
Universitätsstr. 38
70569 Stuttgart

Abstract: Studien zeigen, dass in der Industrie großer Verbesserungsbedarf im Bereich der Testprozesse besteht. Um aber die Prozesse gezielt und wirksam verbessern zu können, ist eine Erhebung des Ist-Prozesses erforderlich. In diesem Artikel wird ein Vorgehen bei einer solchen Ist-Prozess-Erhebung anhand eines Kriterienkatalogs vorgestellt.

1 Einführung

In der industriellen Praxis spielt der Programmtest eine große Rolle. Auch wenn die Zahlen von Unternehmen zu Unternehmen schwanken, kann man annehmen, dass bis zu 50 % [Har00, We05] der gesamten Projektaufwände für den Programmtest verwendet werden. Trotz dieser gewichtigen Rolle des Programmtests findet dieser oft sehr „hemdsärmelig“, d. h. unsystematisch und lückenhaft statt, und die heute bekannten Methoden und Werkzeuge zum Programmtest sind in den Unternehmen sehr unterschiedlich verbreitet [Wi98, We05, Mü98]. Die Studien zeigen, dass sich in vielen Unternehmen der Prozess des Tests deutlich verbessern lässt. Allgemein gültige Verbesserungsmaßnahmen gibt es aber nicht: Um systematisch eine angemessene Prozessverbesserung vorzunehmen, muss zunächst der Ist-Zustand festgestellt werden.

In dieser Arbeit werden ein Vorgehen und ein Kriterienkatalog zur Bewertung von Testprozessen vorgestellt. Die Kriterien sind aus der Literatur und den „Best practices“ abgeleitet und werden zur Bewertung eines konkreten Testprozesses in der Industrie angewendet und evaluiert. Ausgangspunkt dieser Arbeit war der Auftrag eines Unternehmens, seinen Testprozess zu begutachten um daraus Verbesserungsvorschläge abzuleiten. Weder in der industriellen Praxis noch in der Literatur wurde ein geeignetes Vorgehen oder ein Kriterienkatalog zur Strukturierung einer solchen Untersuchung gefunden.

1.1 Verwandte Arbeiten

Um den Reifegrad von Prozessen der Software-Entwicklung zu erheben, haben sich in der Industrie Prozessreifegradmodelle wie z. B. CMM, CMMI oder Spice durchgesetzt. In diesen Reifegradmodellen wird der Programmtest allerdings nicht speziell adressiert; so gibt es z.B. in CMM keine speziellen „Key Process Areas“ für Programmtest.

Burnstein et al. [Bu96] definieren erweiternd zu CMM ein Reifegradmodell für Testprozesse (TMM), das ähnlich dem CMM in fünf Reifestufen eingeteilt ist. TMM setzt dabei die Schwerpunkte sehr stark auf die Organisationsstruktur sowie Testmanagement-Aktivitäten. Zudem werden im TMM Test-relevante Metriken sowie der Werkzeugeinsatz betrachtet. Die Struktur des TMM ist der des CMM recht ähnlich: Ein bestimmter Reifegrad bedeutet, dass ein Unternehmen eine bestimmte Test-Kompetenz besitzt. Der Reifegrad selbst wird durch Ziele und Unterziele beschrieben, die Unterziele werden durch Aktivitäten oder Zuständigkeiten erreicht. Wie im CMM wird im TMM die Stufe 1 ohne weitere Voraussetzungen erreicht. Ab Stufe 2 „Phase Definition“ besteht im Unternehmen ein geplanter und kontrollierter Testablauf, und grundlegende Test-Ausprägungen wie Unit-, Integrations-, System- und Akzeptanztests werden angewendet. Auch der Glassbox-Test wird in Stufe 2 angewendet. Stufe 3 „Integration“ betrachtet den Test nicht mehr als abgegrenzte Tätigkeit nach der Implementierung, sondern als kontinuierliche Tätigkeit im gesamten Projektverlauf. Auch wird ab Stufe 3 eine eigenständige Test-Organisationseinheit gefordert. Stufe 4 und 5 umfassen dann noch neben anderem die Erhebung von Fehlerdaten, Ausbildungsprogramme für Mitarbeiter, Aufwandsmetriken und Maßnahmen zur kontinuierlichen Prozessverbesserung. Zwar beschreibt das TMM die Testaktivitäten nur sehr grob, als Grundlage für den zu erstellenden Kriterienkatalog ist es aber gut geeignet.

Dustin [Du03] beschreibt 50 bewährte Bestandteile eines effektiven Testprozesses, sogenannte „Best practices“, ohne dabei einen Soll-Prozess vollständig zu definieren. Ihre Empfehlungen zu Testorganisation, Testmanagement, Testautomation oder Testausführung sind für die detaillierte Prozess-Verbesserung hilfreich. Gelperin und Hetzel [Ge88] beschreiben einen grundlegenden Testprozess „Systematic Test and Evaluation Process, STEP“ über Planung, Testentwurf, Ausführung und Auswertung. Sie unterteilen auch die Testplanung in eine Master- und eine Detailplanung, da es projektweite Testplanung, aber auch eine Planung für einzelne Testaktivitäten gibt. Perry [Pe99] beschreibt detailliert einen elfstufigen Testprozess. Neben Testmanagement, Entwurf, Ausführung und Auswertung beschreibt er auch noch den Akzeptanztest, den Test der Installation, die Wartung von Tests und die Bewertung der Testgüte. Spillner et al. [Sp05] beschreiben einen fundamentalen Testprozess, der die Testplanung und -steuerung, den Entwurf von Testfällen mit Soll-Resultaten, die Testdurchführung und die Testauswertung enthält. Koomen und Pol [Ko99] beschreiben ein Modell zur Verbesserung von Testprozessen (Test Process Improvement, TPI). Sie unterscheiden dazu 20 Schlüsselbereiche. Neben Bereichen wie „Test-Strategie“ oder „Planung und Schätzung“ behandeln sie auch statische Prüfungen und Metriken. Je Bereich nennen sie „Checkpoints“, die den Erfüllungsgrad prüfen.

1.2 Definitionen

Eine der großen Schwierigkeiten bei Befragungen zum Testprozess ist, dass keinerlei einheitliches Vokabular für den Programmtest besteht. Dies beginnt bereits beim Begriff des Tests an sich. Es gibt durchaus Auffassungen, die statische Prüfungen, Begutachtungen wie z. B. Reviews oder undokumentierte Prüfungen der Entwickler als Test betrachten. In diesem Artikel wird Test im Sinne der Definition von Myers [My79] verstanden als „Programmausführung in der Absicht, Fehler zu finden“. Statische Programmanalysen und Begutachtungen fallen damit nicht unter den Begriff Test. Zusätzlich wird für einen Test gefordert, dass die genauen Ausführungsbedingungen, insbesondere die Testfälle und Resultate, dokumentiert werden. Auch muss sich der Test auf einen definierten Versionsstand der Software beziehen.

Die Aktivitäten im Programmtest bestehen im Wesentlichen aus drei aufeinander folgenden Teilaktivitäten: Testfallermittlung, Testdurchführung und Testauswertung. Begleitet werden diese Aktivitäten von einem Testmanagement und der Testdokumentation [Lu06]. Die Testvorbereitung umfasst die Testfallermittlung, Soll-Resultatsbestimmung und Testdatengenerierung. Der Testfallermittlung kommt dabei eine entscheidende Bedeutung zu: Bei der Festlegung der Testfälle wird über den Umfang und die Chancen zur Fehleraufdeckung entschieden. Das Testmanagement definiert und überwacht die Testziele und Test-Abbruchkriterien. Auch Aufwandsschätzung und -überwachung sowie die Festlegung geeigneter Organisationsstrukturen (Personal, Verantwortlichkeiten) zählen zum Testmanagement.

1.3 Charakteristische Merkmale von Programmtests

In der Praxis haben sich zahlreiche Test-Ausprägungen entwickelt, die dem beschriebenen Aktivitäten-Muster folgen. Beispiele für solche Test-Ausprägungen sind der Unit-Test, der Systemtest oder der Abnahmetest. Die folgenden charakteristischen Merkmale variieren je nach Test-Ausprägung:

- Testziel (Modulprüfung, vertragliche Abnahme, Erprobung, Prüfung von Performance oder Robustheit, wiederholte Prüfung unveränderter Systemfunktionen, ...)
- Methode zur Herleitung der Testfälle
- Zeitraum bzw. Phase im Entwicklungsprozess
- Test-Endekriterium
- Systemeigenschaft, die getestet wird
- Beteiligte Rollen (z. B. Entwickler, Tester, Kunde, Anwender, ...)
- Werkzeugeinsatz

Neben den Test-Ausprägungen hat sich noch eine Reihe von Methoden entwickelt, die die praktische Durchführung der Tests ermöglichen oder erleichtern. Besonders viele Methoden sind im Bereich der Testfallermittlung bekannt: Spezifikationsbasiert, programmcodebasiert (Glassbox-Test), Zufallswerte, modellbasiert, „ad hoc“, „Error-guessing“ usw. Weitere Methoden gibt es beim Testaufbau, z. B. Mock-Erstellung, Soll-Resultatsberechnung, Test-Automatisierung oder Testtreiber.

2 Erhebung von Ist-Testprozessen

Um nach Burnstein et al. [Bu96] im TMM die Stufe 2 zu erreichen, werden ein Test-Management sowie Unit-, Integrations-, System-, Akzeptanz- und Glassboxtest gefordert. Dieser Forderung schließen sich Perry und Spillner im Wesentlichen an. Auch zeigen empirische Untersuchungen, dass zumindest einzelne dieser Test-Aktivitäten in der Praxis auch verbreitet sind. Es liegt somit nahe, diese Test-Aktivitäten bei einer Prozessanalyse vorrangig zu betrachten. Die Erhebung der Ist-Testprozesse wäre sehr einfach, wenn es unter den Beteiligten darüber Einvernehmen gäbe, was unter den einzelnen Test-Aktivitäten zu verstehen ist und wenn zudem die einzelnen Test-Aktivitäten auch „lehrbuchkonform“ in den Unternehmen durchgeführt werden würden. Beides ist in der Praxis aber nicht der Fall.

2.1 Struktur der Befragung

Wegen der zu erwartenden Verständigungsschwierigkeiten wird das Interview als Untersuchungsmethode gewählt. Interviews sind zwar aufwändiger als Befragungen per Fragebogen, lassen aber eine unmittelbare Begriffsklärung zu. Zudem ist der Kreis der Personen, die Auskünfte zum Programmtest geben können, überschaubar. Um dem Problem des nicht einheitlichen Vokabulars in den Interviews aus dem Wege zu gehen, werden weniger die in Testerkreisen etablierten Begriffe [ISTBQ] als die jeweiligen charakteristischen Merkmale der einzelnen Test-Aktivitäten verwendet. So sind die Fragen für die Befragten verständlicher und die Befragten können präziser antworten.

Zur Strukturierung des Interviews werden zunächst die Test-Aktivitäten der TMM-Stufe 2 untersucht. Da zumindest einzelne dieser Test-Aktivitäten in der Regel auch durchgeführt werden, wird so der Einstieg in das Gespräch erleichtert. Konkret werden Fragen zu den charakteristischen Merkmale dieser Test-Aktivitäten gestellt. Die Fragen werden vor den Interviews in einem Kriterienkatalog, wie er im folgenden Abschnitt beschrieben ist, zusammengestellt. Aus der konkreten Befragung geht das Interview dann in ein offenes Gespräch zu weiteren Test-Aktivitäten über. Werden in diesem Verlauf charakteristische Merkmale einer weiteren Test-Ausprägung genannt oder gibt es Hinweise darauf, werden zu dieser Test-Ausprägung weitere Merkmale konkret erhoben. Ist beispielsweise im offenen Interview von einer automatisierten Durchführung von Testfällen die Rede, könnte dies ein Hinweis auf einen Regressionstest sein. Um diese Frage zu klären, wird nun nach den charakteristischen Merkmale von Regressionstests weitergefragt.

2.2 Kriterienkatalog am Beispiel des Unit-Tests

Nachfolgend wird in Tabelle 1 als Beispiel der Kriterienkatalog für die Test-Ausprägung des Unit-Tests vorgestellt. Es werden Fragen formuliert, die die charakteristischen Merkmale des Unit-Tests abprüfen. In der durchgeführten Befragung wurden neben der Tabelle für den Unit-Test auch entsprechende Tabellen für Integrations- und Systemtest sowie für das Test- und Fehlermanagement erstellt.

Unit-Test	
Testziel	Findet eine Prüfung einzelner Module oder einzelner Funktionen statt?
Methode zur Herleitung der Testfälle	Findet ein Test gegen eine Modulspezifikation statt?
Zeitraum bzw. Phase im Entwicklungsprozess	Findet ein Test kurz nach der Entwicklung statt?
Systemeigenschaft, die getestet wird	Findet eine Prüfung der Modul-Funktion statt?
Beteiligte Rollen	Findet ein Test durch die Entwickler statt?
Werkzeugeinsatz	Werden Testtreiber verwendet, die in der Sprache des Prüflings geschrieben sind?

Tabelle 1: Kriterienkatalog für Unit-Tests

3 Durchgeführte Befragung

Bei der Befragung hat es sich als günstig herausgestellt, die Interviews möglichst offen zu führen, d. h. die Interviewpartner werden angeregt, zu den Aktivitäten, Rollen und Dokumenten, die sie dem Test zurechnen, zu berichten. Um den Einstieg ins Gespräch zu erleichtern, wurden wie beschrieben zu Beginn die „Pflicht“-Test-Ausprägungen, also Unit-Test, Integrations-, System- und Glassboxtest abgefragt. Insgesamt wurden Personen zu fünf Rollen befragt. Die Befragungen dauerten 1 - 2 Stunden. Nach Ende des Interviews wurden die Antworten in den Kriterienkatalog übertragen. Neben dem QA-Beauftragten, dem Projektleiter und Mitgliedern des Entwicklerteams wurde auch das Management einbezogen. Im Unterschied zu den anderen Gesprächspartnern war beim Gespräch mit dem Management der Gesprächsanteil zu den Ist-Prozessen relativ gering. Statt dessen wurden Veränderungswünsche, z. B. im Bereich der Testgüte oder des Automatisierungsgrads, genannt. Neben den Interviews fanden auch Auswertungen der existierenden Dokumente statt. Insgesamt wurden mit Berichtserstellung etwa zwölf Aufwandstage benötigt.

4 Bewertung der Vorgehensweise

Das beschriebene Vorgehen hat sich bewährt. Die Interviews waren durchweg produktiv. Fast alle Gespräche dauerten länger als vorab festgelegt. Auch hat sich das Interview als die ergiebigste Methode zur Erhebung der Ist-Prozesse gezeigt. Nach den gesammelten Erfahrungen ist von einer schriftlichen Befragung zum Programmtest eher abzuraten; zu oft war es in den Gesprächen nötig, Begriffe zu klären. Zudem ergab sich, dass eine Untersuchung der Dokumente allein nicht ausreicht, um ein Bild von einem (gelebten) Testprozess zu gewinnen.

Vorschläge zur Prozessverbesserung konnten aus den Untersuchungsergebnissen gut abgeleitet werden. So wurden Mängel bei den Unit-Tests gefunden, die in Teilen durch starke Kopplung unter den Modulen behindert waren. Zudem fehlten weitestgehend die Testtreiber zur Automatisierung der Unit-Tests. Auch wurden die Aufwände zur Testdurchführung nicht erfasst. Beim Integrations- und Systemtest erfolgte die Auswahl der Testdaten nicht systematisch. Ein weiteres unmittelbares Resultat war das Fehlen eines Glassboxtests. Auch hierzu wurden Maßnahmen eingeleitet. So wurden beispielsweise Ziele festgelegt (spezielle Überdeckungsmaße wie z. B. Zweigüberdeckung), und Werkzeuge zur Überdeckungserhebung wurden untersucht. Ein Pilotprojekt zur Einführung eines Glassboxtests ist geplant.

5 Zusammenfassung

In diesem Artikel wurde ein Vorgehen für die Untersuchung von Testprozessen und der Aufbau eines Kriterienkataloges vorgestellt. In einer konkreten Untersuchung eines Testprozesses in der Industrie wurde dieses Vorgehen angewendet und hat sich bewährt. Ein besonderer Vorteil des Verfahrens ist, dass aus dem Kriterienkatalog bereits Vorschläge zur Prozessverbesserung abgeleitet werden können.

Literatur

- [Bu96] Burnstein, I., Suwannasart, T., and Carlson, R.: "Developing a Testing Maturity Model for Software Test Process Evaluation and Improvement". In Proceedings of the IEEE international Test Conference on Test and Design Validity (October 20 - 25, 1996). IEEE Computer Society, Washington, DC, 581-589.
- [Du03] Dustin, E.: "Effective Software Testing: 50 Specific Ways to Improve Your Testing". Addison-Wesley Professional, 2003
- [ISTQB] "Standard glossary of terms used in Software Testing", Version 1.2 (dd. June, 4th 2006), www.istqb.org
- [Ge88] Gelperin, D. and Hetzel, B.: "The growth of software testing". Commun. ACM 31, 6 (Jun. 1988), 687-695.
- [Ha00] Harrold, M. J.: "Testing: a roadmap". In Proceedings of the Conference on the Future of Software Engineering (Limerick, Ireland, June 04 - 11, 2000). ICSE '00.
- [Ko99] Koomen, T., Pol, M.: "Test Process Improvement". Addison-Wesley, 1999
- [Lu07] Ludewig, J., Lichter, H.: "Software Engineering". dpunkt Verlag 2007
- [Mü98] Uwe Müller, Thomas Wiegmann: "State of the practice" der Prüf- und Testprozesse in der Softwareentwicklung — Ergebnisse einer empirischen Untersuchung bei Softwareunternehmen in Deutschland" Technischer Bericht, Universität Köln, März 1998
- [My79] Myers, G. J.: "Art of Software Testing", John Wiley & Sons, Inc., New York, 1979
- [Pe99] Perry, W.: "Effective Methods for Software Testing". 2. Auflage, John Wiley & Sons Inc., 1999
- [Sp05] Spillner, A., Linz, T.: "Basiswissen Softwaretest". 3. Auflage, dpunkt Verlag 2005
- [We05] Wetzel, M; Siegwart, K.: "Empirische Untersuchung der analytischen Qualitätssicherung in der Industrie". Softwaretechnik-Trends, Band 25 Heft 3, Gesellschaft für Informatik, ISSN 0720-8928, August 2005.
- [Wi98] Karl E. Wiegers: "Read my Lips: No New Models!". IEEE Software, Sep./Okt. 1998, S. 10-13