Von den Anforderungen zu den Testfällen
Wie optimiert man Kosten und Nutzen des Tests

Dirk Meyerhoff, Herbert Grün, Ricardo von Spangenberg

SQS Software Quality Systems AG, Stollwerckstr. 11, 51149 Köln
Tel: 02203 9154 0, Fax: 02203 9154 15, Web: www.sqs.de,
Email: dirk.meyerhoff@sqs.de, herbert.gruen@sqs.de, ricardo.von_spangenberg@sqs.de

Einleitung

Kostenbewusstsein und Kostensenkung sind neben Termineinhaltung heutzutage zentrale Aspekte bei der Entwicklung von Anwendungen. Ausgehend von den Anforderungen und dem Fachkonzept wird auf Entwicklungsseite das Design und die Implementierung begonnen. Häufig sind die Anforderungen und das Fachkonzept jedoch nicht stabil genug, um implementiert zu werden.

Gleichzeitig weisen nicht nur Lehrbücher sondern auch Praktiker darauf hin, dass der Test nicht früh genug beginnen kann.

In diesem Artikel wird herausgearbeitet, wie durch geeignete Vorgehensweisen beide Probleme gleichzeitig angegangen werden. Durch eine systematische Testvorbereitung wird gleichzeitig die Qualität der Anforderungen und Fachkonzepte gesteigert, und, sofern man mit dieser Maßnahme früh genug beginnt, gleichzeitig ein Kostenvorteil erschlossen, denn nur die qualitätsgesicherten Vorgaben gehen tatsächlich in den weiteren Entwicklungsprozess ein.

Anforderungen und Fachkonzepte

Die Verkaufszahlen von Werkzeugen wie Doors (Telelogic), Requisite-Pro (Rational) oder Caliber (Borland) belegen eindeutig, dass die Aufgabe der Anforderungsermittlung fest in den Software-Entwicklungsprozessen der Unternehmen verankert wurde.
Dennoch gibt es nach wie vor Probleme mit der Qualität der ermittelten - und nunmehr sauber dokumentierten - Anforderungen. Sie sind nach wie vor in zu vielen Fällen unvollständig, widersprüchlich oder einfach nur unklar formuliert.
Den Systemanalytikern und Entwicklern bleibt die ehrenhafte Aufgabe, diese mangelhaften Anforderungen in weniger mangelhafte Fachkonzepte sowie Analyse- und Designdokumente zu überführen und danach zum Beispiel in Java zu implementieren.

Test

Zu den ersten Aufgaben des Testers nach der Testplanung gehört die Testvorbereitung, d.h. die Spezifikation der Testfälle und Testdaten. Bei der Testvorbereitung hat der Tester im Grunde eine ähnliche Aufgabe wie Systemanalytiker und Entwickler: Aus fehlerhaften Anforderungen und Fachkonzepten korrekte, vollständige und eindeutige Testfälle ermitteln.

Eine wichtige Feststellung in der Praxis ist jedoch, dass es fast immer sehr viel preiswerter ist, einen Testfall zu spezifizieren als ein Programm zu entwerfen und zu implementieren, das den Testfall korrekt umsetzt. Aus dieser Kostenüberlegung heraus macht es wenig Sinn, zunächst den Systemanalytiker und den Entwickler mit der weiteren Umsetzung der Anforderungen und Fachkonzepte zu betrauen, und dann im Anschluss bei der Testfallermittlung festzustellen, dass die Vorgaben noch eine Menge Unzulänglichkeiten und Fehler enthalten.

Dreht man die Reihenfolge um und beginnt mit der Testfallermittlung, so werden die Fehler in den Vorgaben sehr früh und damit preiswert gefunden, denn die fehlerhaften Vorgaben sind noch nicht implementiert.

Ein Beispiel aus der Praxis zeigt die folgende Abbildung:

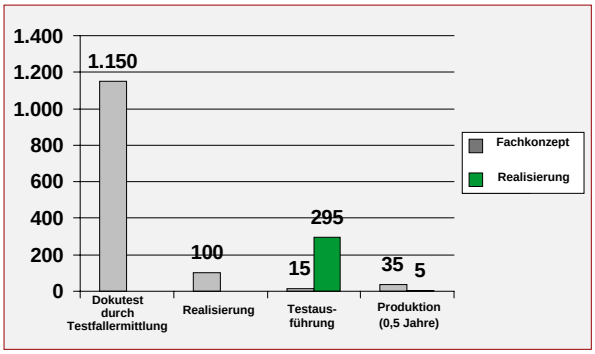

Abb. 1: Testen findet Anforderungsfehler

Die Abbildung zeigt, dass in diesem Projekt von ca. 1.600 Fehlern aus Test und Produktion (6 Monate nach Einführung) insgesamt 1.150 durch die Testfallermittlung gefunden wurden. Diese Fehler waren Fehler in den Anforderungen, Fachkonzepten, Analyse- und Design-Dokumenten und sie wurden vor der Realisierung gefunden.

Nur ca. 300 Fehler - also insgesamt weniger als 20% der Fehler - waren auf fehlerhafte Realisierung zurückzuführen. Dies bedeutet, dass in diesem Projekt mehr als 80% der Fehler prinzipiell ohne Ausführung der Programme aufdeckbar gewesen sind. Immerhin wurden mehr als 70% der Fehler auch vor der Implementierung gefunden.

Kostenreduktion durch frühen Testbeginn

Aufgrund des Beispieles, das repräsentativ ist für die meisten der Projekte, an denen SQS beteiligt ist, wird deutlich, dass der frühe Beginn des Tests sehr viel Geld einspart, denn fehlerhafte Vorgaben werden nicht zunächst umgesetzt und dann korrigiert. Vielmehr ist die Reihenfolge umgekehrt. Die Anwender werden

durch dieses Vorgehen ebenfalls entlastet, denn sie müssen weniger Zeit investieren. Sie korrigieren Anforderungen in einem Stadium, wo die Korrektur noch preiswert ist, weil noch keine Realisierung erfolgt ist.

Systematische Testvorbereitung

Nachdem wir den geeigneten Zeitpunkt für die Testvorbereitung diskutiert haben, kommen wir nun zur Frage der geeigneten Mittel. Auch hier geht es wieder um Kosten. Ziel ist, mit möglichst wenig Testfällen eine möglichst hohe fachliche Abdeckung zu erzielen. Durch die hohe Abdeckung wird das kommerzielle Risiko, das mit der Freigabe der Anwendung verbunden ist, reduziert.

In Workshops mit mehr oder weniger erfahrenen Testern stellen wir immer wieder fest, dass Tester mit sehr unterschiedlichen Strategien Testfälle ermitteln. Das reicht von "Testen bis zur Erschöpfung" (des Testers) über Checklisten-gesteuertes Testen bis zur individuellen, jedoch meist undokumentierten Testmethodik.

Dadurch ist die Anzahl der Testfälle, die jeder Tester für eine vorgegebene GUI-Maske erwartet, sehr unterschiedlich. Abbildung 2 stellt dies dar.

In diesem Beispiel haben wir unterschiedliche Testteams nach ihrer Einschätzung gefragt: Wie viele Testfälle werden für eine vorgegebene GUI-Maske benötigt? Jeder Tester hat nach seiner individuellen Methode die Testfälle ermittelt und dann das Ergebnis dokumentiert. So erzielten die Teams dann jeweils Einzelaussagen von zwischen 4 und 90 Testfällen. Vergleicht man diese Anzahl der Testfälle mit der Anzahl der Testfälle, die sich nach systematischer Vorgehensweise - in diesem Fall mit der Methode TCS (Test Case Specification) - ergeben, so wird das Einsparpotenzial deutlich. Die mit TCS ermittelte Anzahl Testfälle unterscheidet sich von Tester zu Tester nicht signifikant, d.h. man darf hier generell von der Anzahl 4 ausgehen. Methodisch werden es nicht weniger Testfälle, weil die Methode dem Tester in diesem Beispiel bei weniger als 4 Testfällen eine mangelhafte fachliche Abdeckung anzeigt.

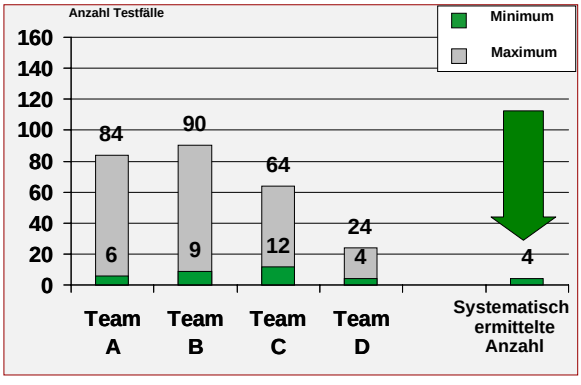

Abb. 2: Anzahl erwarteter Testfälle für eine GUI-Maske

Nach unserer Erfahrung benötigt ein Tester für die Ermittlung von 4 systematischen Testfällen ähnlich viel Zeit wie für 4 bis 8 ad hoc spezifizierte Testfälle.

Methodik zur Systematischen Testfallermittlung

Für den Test kommerzieller Software und in vielen Fällen auch von technischer Software hat sich die Methode TCS (Test Case Specification) bewährt. Diese Methode kombiniert die klassische Äquivalenzklassenanalyse mit einigen wichtigen Erweiterungen. Der Tester analysiert zunächst die Ein- und Ausgabeelemente der Anwendung. Eingabeelemente sind zum Beispiel GUI-Eingabefelder oder Schaltflächen. Ausgabeelemente sind Reports, DB-Tabellen oder Ergebnisanzeigen. Für jedes Eingabeelement (Ausgabeelemente analog) werden seine Äquivalenzklassen ermittelt, d.h. die Eingabemöglichkeiten, die jeweils zur gleichen Wirkung führen.

Einen Testfall erstellt der Tester manuell, indem er zu jedem Eingabeelement eine Äquivalenzklasse auswählt. Die Menge der ausgewählten Eingaben (genauer der ausgewählten Äquivalenzklassen) mit zugeordneten Wirkungen definiert dann den Testfall. Abbildung 3 zeigt das Testfallermittlungswerkzeug SQS-TEST/Professional TCS. In der Ansicht hat der Tester bereits zwei Testfälle erstellt.

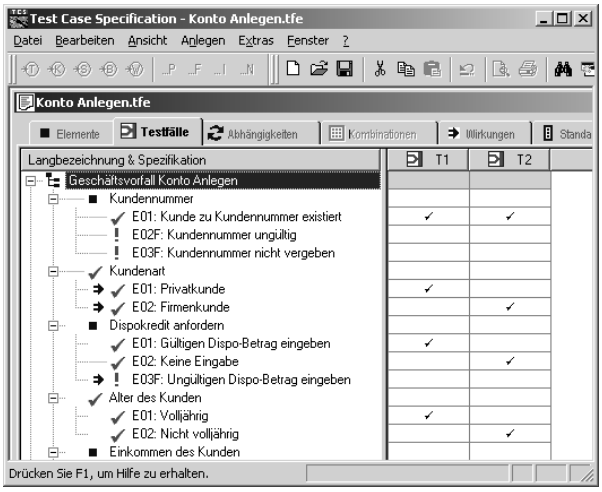

Abb. 3: Werkzeugunterstützte Testfallermittlung

Die Testfälle sind als Spalten dargestellt. Noch nicht abgedeckte Äquivalenzklassen (Zeilen) sind nun für den Tester direkt erkennbar. Sie sind durch ein rotes Ausrufungszeichen gekennzeichnet. Weitere Testfälle erstellt der Tester dann typischerweise mit dem Ziel, die noch nicht abgedeckten Äquivalenzklassen abzudecken.

Das Werkzeug kann die Testfälle für den Tester generieren, sobald die Äquivalenzklassen definiert sind. Zusätzlich ermöglicht SQS-TEST die Generierung und Verwendung von Entscheidungstabellen, um nachvollziehbar auch Kombinationen von Äquivalenzklassen vollständig oder mit gezielter Auswahl zu testen. Dazu wählt der Tester die für die Entscheidungstabelle relevanten Äquivalenzklassen aus und lässt die Entscheidungstabelle generieren. Im Anschluss geht er die Kombinationen durch und fasst sie bei Bedarf zusammen oder schließt sie als irrelevant vom Test aus. Diese Betrachtungen des Testers werden im Werkzeug dokumentiert und der Tester kommt so zu nachvollziehbaren, wartbaren und weitestgehend vollständigen

Tests. Abbildung 4 zeigt eine generierte Kombinationstabelle.

Die erste Spalte gibt an, welche Äquivalenzklassen miteinander kombiniert werden sollen. In diesem Fall werden 1x2x2 Äquivalenzklassen kombiniert, was zu vier Kombinationen führt. Der Tester hat dann die letzten beiden Kombinationen, die zur gleichen Wirkung führen, zusammengefasst. Dies ist im Werkzeug jederzeit nachvollziehbar und änderbar.

Das Werkzeug hat eine Reihe weiterer Features, die dem Tester helfen, effizient und effektiv Testfälle zu erstellen. So identifiziert das Werkzeug redundante Testfälle und unterstützt den Tester dabei, die Testfälle zu finden, die von Änderungen der Vorgaben betroffen sind und diese bei Bedarf zu pflegen und zu warten.

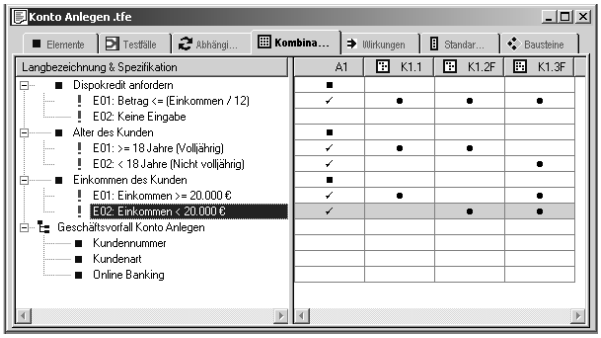

Abb. 4: Kombinationstabelle zur Testfallermittlung

Kostenreduktion durch Systematische Testfallermittlung

Die mit der Systematischen Testfallermittlung verbundenen Kosteneinsparungen ergeben sich durch eine hohe Abdeckung - das heißt vorhandene Fehler werden mit hoher Wahrscheinlichkeit tatsächlich gefunden - sowie durch den reduzierten Ressourcenverbrauch bei der Ausführung, Auswertung und ggf. auch bei der Automatisierung der Testfälle. Wenn die Anzahl der Testfälle durch systematische Testfallermittlung beispielsweise um 50% reduziert wird, so sinken die Aufwände für jede manuelle Testausführung und Testauswertung ebenfalls um 50%. Die Aufwände für die Automatisierung sinken ebenfalls um 50%, sofern man generelle Basiskosten etwa für den Aufbau eines Testsystems oder eines Testarchivs hier nicht berücksichtigt.

Fazit

Anforderungen und Fachkonzepte für die Software-Entwicklung werden heute mehr und mehr in geeigneten Werkzeugen dokumentiert. So ist der Weg von der Anforderung über das Fachkonzept bis zur Implementierung in aktuellen Projekten oft sehr gut nachvollziehbar. Es hapert jedoch immer wieder an der Qualität der Anforderungen und Fachkonzepte. Dies stellt man meist zu spät, nämlich nach der Realisierung, im Abnahmetest fest.

Kosten können reduziert werden, wenn man die Phase der Testfallermittlung beginnt, bevor die Realisierung startet. So werden Fehler in Anforderungen und Fachkonzepten gefunden, bevor sie größere Kosten verursacht haben. Geht man zusätzlich systematisch vor, so können die Testaufwände reduziert werden, ohne dass die fachliche Abdeckung der Tests leidet. So bedeuten weniger Testfälle generell einen geringeren Testaufwand, jedoch ohne dass das Risiko bei der Abnahme steigt.

Literatur

[1] *Methoden und Verfahren der Software-Qualitätssicherung*, Deutsche Gesellschaft für Qualität e.V. Ffm, Berlin, Beuth Verlag GmbH, 1992

[2] Beizer, B., *Software Testing Techniques*, Second ed., Van Nostrand Reinhold, New Y-ork, 1990

[3] Fewster, M., Graham, D., *Software Test Automation*, Addison Wesley, New York, 1999

[4] Keese, P., Scalable Acceptance Testing, in Wieczorek, M. and Meyerhoff, D. *Software Quality, State of the Art in Management, Testing, and Tools*, Springer, Berlin, 2001

[5] Meyerhoff, D, *Von den Anforderungen zu den Testfällen*, Telelogic Anwenderkonferenz, Bonn, 2003

[6] Myers, G.J., *Methodisches Testen von Programmen*, 7. Auflage, Oldenbourg Verlag, München, 2001

[7] http://www.borland.com liefert Hinweise zum Requirements-Management-Werkzeug Caliber

[8] http://www.rational.com liefert Hinweise zum Requirements-Management-Werkzeug Requisite-Pro

[9] http://www.sqs.de liefert Hinweise auf Methoden, Werkzeuge und Schulungen zum Thema Test

[10] http://www.telelogic.de liefert Hinweise zum Requirements-Management-Werkzeug Doors