

Geeignete Organisation von Testprozessen  
und professionelle Qualifikation der Tester

# Für eine bessere Software

Matthias Hamburg, Tilo Linz

Fehlerhafte Software kann große wirtschaftliche Schäden und hohe Unzufriedenheit der Kunden zur Folge haben. Um das zu vermeiden, braucht Software bessere Tester.

In der Softwareentwicklung haben sich in den vergangenen Jahren viele konstruktive Techniken durchgesetzt, die dazu beitragen, die Programmierung zu erleichtern und fehlerärmer zu machen: beispielsweise der systematische Architekturf Entwurf in UML, die damit einhergehende Nutzung bewährter Entwurfsmuster (Design Pattern) und die anschließende Implementierung der Programme in moderne objektorientierte Programmiersprachen. Die Techniken erlauben die systematische Wiederverwendung von Teillösungen und verbessern die Modularisierung der Software. Das erhöht nicht nur deutlich die Produktivität der Entwickler, sondern auch die Qualität der Systeme. Allerdings ist es grundsätzlich unmöglich, allein durch solche Maßnahmen zuverlässig zu vermeiden, dass die Software Programmierfehler, Designfehler oder Fehler durch falsch erhobene, falsch verstandene oder vergessene Anforderungen enthält.

Das Ziel muss daher sein, Fehler effektiver aufzufinden, bevor sie (im produktiven System) Schaden anrichten können. Mit anderen Worten: Unternehmen, die Software erstellen oder in ihre Produkte integrieren, müssen zielgerichteter, wirksamer und produktiver testen als bisher. Die gute Nachricht ist: Es lässt sich mit organisatorischen und ausbildungsorientierten Maßnahmen viel erreichen.

## Wege für eine bessere Organisation des Testens

Der Schlüssel für ein wirksames Testen liegt darin, die Programmierung und das Testen organisatorisch so gut und so früh wie möglich voneinander zu trennen. Fünf Organisationsmodelle sind in dem Zusammenhang denkbar:

1. Die Softwaretests liegen weiter in der Verantwortlichkeit des Entwicklungsteams, die sich jedoch „gegenseitig“ testen. Ein Entwickler testet also die Programme eines Kollegen, aber nicht die eigenen.
2. Einzelne Mitglieder des Entwicklerteams werden ausschließlich für Testarbeiten abgestellt. Diese Tester erledigen alle Testarbeiten ihres Teams.
3. Ein separates Testteam bekommt die Testaufgaben für die Dauer des Entwicklungsprojekts übertragen. Das Team wird von einem Testmanager geleitet. Auch Mitarbeiter aus Fach- und IT-Abteilung arbeiten als unabhängige Tester in diesem Team mit.
4. Für spezielle Aufgaben wie das Testen von Performanz, Benutzbarkeit oder Sicherheit werden zeitweise unabhängige Tester beauftragt.
5. Eine separate Organisationseinheit (interne Testabteilung/Test-Center

oder ein externer Testdienstleister) übernimmt das Testen oder wesentliche Teile davon, etwa den Systemtest. Diese Organisationseinheit ist als Dienstleister für mehrere Projekte oder Produkte zuständig.

Die Maßnahmen ergänzen sich beziehungsweise bauen aufeinander auf. Es ist nicht notwendig, alle Maßnahmen von Beginn an gleichzeitig einzuführen. Man kann die Maßnahmen schrittweise umsetzen und miteinander kombinieren. Das Prinzip ist jedoch immer das Gleiche: Man trennt Test- und Entwicklungsaufgaben personell, um zu erreichen, dass Annahmen, Gedankenmodelle und versteckte Voraussetzungen, die die Programmierer getroffen oder verwendet haben, kritisch hinterfragt werden. Das alleine macht das Testen schon wesentlich wirksamer. Dass die auf das Testen spezialisierten Personen oder Teams Routine bekommen und die relevanten Testmethoden zunehmend besser kennen und beherrschen, erhöht zudem die Produktivität.

Die personelle Trennung erhöht jedoch den Bedarf an Dokumentation – von den Testfällen bis zu den Testergebnissen. Aber das ist kein Nachteil, im Gegenteil: Denn nur wer aufschreibt, was man testet, kann erkennen, welche Regionen der Software (noch) nicht ausreichend getestet wurden und welche Risiken für unentdeckte Fehler deshalb noch schlummern.

## Trennung und Kooperation

Die Einführung einer organisatorischen Trennung zwischen Entwicklung und Test verläuft erfahrungsgemäß nicht ohne Reibungspunkte. Es lassen sich dabei fünf Phasen unterscheiden – von der Ablehnung bis zur konstruktiven Zusammenarbeit:

### 1. Ausgangsphase

Dass jede Codezeile zu testen ist, weiß jeder Entwickler. Im Projektplan ist vermutlich auch ein Zeitslot für das Testen vorgesehen. Im Zweifel gehen die Entwurfs- und Codierarbeiten aber vor. Die Programme müssen fertig werden, und der Projektplan ist häufig knapp bemessen. Zudem ist der Reiz, Neues zu programmieren, verlockender, als den (fast) fertigen Code systematisch zu testen. Daher wird die Botschaft, Test und Entwicklung zu trennen, grundsätzlich positiv aufge-

nommen. Eine gewisse Skepsis darüber, ob die Trennung in der Praxis funktioniert, ist aber nicht zu übersehen: Wird der (neue) Tester in der Lage sein, „meine“ Programme gut genug zu verstehen? Welche Fehler kann der Tester finden, wenn er nicht derart tief in der Materie steckt wie der Entwickler? Erzeugt das nicht eine Menge unnötige Bürokratie und Overhead durch Testpläne, Fehlerprotokolle und Testberichte? Will die Projektleitung Fehlerstatistiken einführen? Etwa zur Leistungskontrolle?

Das Management sollte der verständlichen Skepsis und Unsicherheit mit früher und offener Information begegnen: Wie wird die Arbeitsteilung genau aussehen? Welche Informationen benötigen die Tester, um wirkungsvoll arbeiten zu können? Wie werden Fehler dokumentiert? Wer wertet das aus? Welche Ergebnisse liefern die Tester? Wie helfen die Tester den Entwicklern, entdeckte Fehler schneller als bisher zu beheben?

## 2. Ablehnungsphase

Viel schneller als erwartet werden die neuen Tester bisher übersehene Fehler aufdecken, und zwar nicht nur Kleinigkeiten, sondern viele ernsthafte Fehler und Lücken oder Ungereimtheiten im Softwaredesign und in den Spezifikationen. Das ist neu und unangenehm für die betroffenen Entwickler. Sie empfinden jede Fehlermeldung schnell als Kritik an der eigenen Leistung. Sie reagieren in der Phase häufig ablehnend und versuchen, sich gegen vermeintliche Angriffe durch die Tester zur Wehr zu setzen. Jeder Tester kennt die empörten Kommentare über Fehlermeldungen nur allzu gut: „Auf meiner Maschine funktioniert das“, „echte Anwender machen das nie so“, „it's not a bug, it's a feature“, „das Problem kenne ich schon lange, ich hatte nur keine Zeit mich darum zu kümmern“.

Tester müssen in dieser Phase Fingerspitzengefühl beweisen: Sie haben Fehler zu finden und schonungslos aufzudecken, müssen dabei allerdings „Kleinigkeiten“ von „relevanten“ Fehlern sicher unterscheiden können – sonst erhalten sie als Sparringpartner fachlich keine Akzeptanz. Aber Tester dürfen nie persönliche Kritik üben, sondern müssen sachlich und objektiv bleiben. Dann verstehen alle Beteiligten schnell, dass ein gemeinsames Ziel verfolgt wird: gemeinsam ein Produkt in hoher Qualität abzuliefern.

## 3. Etablierungsphase

Das Testen wird nun immer mehr zu einem abgegrenzten Teilprozess der Softwareentwicklung. Die Tester arbeiten methodisch und können jederzeit begründen, welche Situationen sie getestet haben und welche nicht. Für Management und Entwickler wird sichtbar, wie gut oder schlecht die erreichte Testabdeckung tatsächlich ist. Fehlermeldungen werden in einem Fehlermanagement-Tool zentral dokumentiert, bewertet und verwaltet. Die berichteten Fehler bearbeiten die Entwickler routinemäßig, und die Tester prüfen das Ergebnis nach. Letztere bereiten die Testläufe rechtzeitig vor, damit ihr Feedback möglichst schnell und zeitnah nach der Lieferung korrigierter Softwarestände erfolgen kann. Durch die Kanalisierung und Priorisierung der Fehlermeldungen über das Fehlermanagement-Tool wird aus chaotischem Ad-hoc-Bugfixing ein geordneter, ruhiger Routineprozess. Entwickler und Tester können ihre Arbeit besser planen.

Das Fehlermanagement-Tool darf jedoch nicht das einzige Kommunikationsmittel zwischen den Entwicklern und Testern sein. Direkte Abstimmungen und eine adäquate Moderation durch ein kompetentes Fehler-Bewertungsgremium sind ebenfalls notwendig. Sonst besteht die Gefahr, dass ein Fehler-Pingpong zwischen beiden Seiten einsetzt – mit dem Ergebnis, dass einige Fehlermeldungen endlos zwischen den Parteien hin und her wechseln.

## 4. (Aus)nutzen des Testens zum eigenen Vorteil

Je wirksamer und umfassender die Tester arbeiten, desto mehr Vertrauen entsteht bei den Entwicklern, dass die Tester gute Arbeit leisten. Umgekehrt sehen die Tester, dass die Entwicklung reife und stabile Software an den Test liefert. Das Zusammenspiel kann gegen Projektende unter starkem Zeitdruck jedoch empfindlich gestört werden. In dieser Phase kann es vorkommen, dass sich einige Entwickler unbewusst – in manchen Fällen auch durchaus bewusst – zu sehr auf die nachfolgende Teststufe verlassen. Nach dem Motto: „da schaut ja der Tester ohnehin noch mal drauf“, liefern sie neue Funktionen oder Bugfixes halbgar ab. Der Test wird durch solche nicht testreife Softwarestände unter Umständen stark behindert und gebremst.

Spätestens jetzt ist es an der Zeit, den nächsten Schritt zu gehen und innerhalb des Tests Teststufen abzugrenzen: den entwicklungsnahe Unit-Test, den Integrationstest und den Systemtest. In jeder Teststufe sind gewisse Fehlerarten einfacher und schneller zu finden. Die Testziele und Testaufgaben der Stufen lassen sich aufeinander abstimmen, und in jeder Stufe arbeiten spezialisierte Tester mit spezialisierten Testwerkzeugen und der passenden Testumgebung. Das erfordert gewisse Zusatzinvestitionen. Als Nutzen ergibt sich mittelfristig eine nochmalige Steigerung der Test-Produktivität. Denn die Produktfunktionen und die Fehlerrisiken werden insgesamt zeit- und kostenoptimal abgedeckt.

## 5. Konstruktive Zusammenarbeit

Im Softwareentwicklungsprozess gibt es etliche Themen, die sowohl für Entwickler als auch für Tester von hoher Relevanz sind. Beispiele hierfür sind das Anforderungsmanagement, Schätz- und Planungsverfahren, Metriken und Statistiken sowie Release- und Änderungsmanagement. Bei diesen Themen sitzen Tester und Entwickler im selben Boot. Wenn Tester gegenüber der Unternehmensführung beispielsweise die Konsequenzen unklarer Anforderungen für die Produktqualität verdeutlichen können und eine Verbesserung der Anforderungsqualität durchsetzen, dann profitieren auch die Entwickler davon. Wenn Entwickler im Unternehmen für ein regelmäßiges Konfigurationsmanagement der Entwicklungsdokumente sorgen, dann vereinfacht das auch das Management der Testpläne und Testfälle erheblich. Charakteristisch für diese Phase ist, dass Entwickler und Tester gemeinsam an kontinuierlicher Prozessverbesserung interessiert sind und sich gegenseitig stärken.

## Bessere Ausbildung der Tester

Die bessere Organisation des Testens ist nur ein Baustein, der notwendig ist. Die Aus- und Weiterbildung der Tester gehört ebenfalls dazu. Unabhängig davon, nach welchem Organisationsmodell man die Aufgaben trennt und wer als Software-Tester Verantwortung übernimmt, muss jeder Einzelne sein Handwerkszeug gut beherrschen. Je besser die einschlägigen Testmethoden und Techniken trainiert sind, desto schneller und effizienter kann ein

Tester arbeiten. Die wichtigsten dieser Techniken sind die Testentwurfsverfahren, die eine systematische Abdeckung von Fehlerrisiken ermöglichen. Sie bilden die Kernkompetenz eines Testers. Heute sind mehr als ein Dutzend grundlegende Testentwurfsverfahren mit vielen Varianten im Einsatz, die ein professioneller Tester beherrschen muss.

Um es auf den Punkt zu bringen: Ausgebildete Tester sind bessere Tester. Die Ausbildung zum Certified-Tester stellt sich dieser Aufgabe. Das International Software Testing Qualifications Board (ISTQB) definiert die Lehrpläne und Qualifizierungsverfahren für die Certified-Tester-Ausbildung, nationale Organisationen wie das German Testing Board (GTB) setzen sie in ihrem Gebiet um, akkreditieren Trainingsanbieter und überwachen die Prüfungen. Die strikte Trennung von Trainings- und Prüfungswesen ist dadurch gewährleistet. Zwei unabhängige Zertifizierungsinstitute, die Dienstleistungsgesellschaft für Informatik mbh und die iSQI GmbH, wickeln derzeit die Prüfungen in Deutschland ab. Mittlerweile wurden hierzulande über 17.000 Studenten und Arbeitnehmer zum Certified-Tester ausgebildet, weltweit durchliefen schon mehr als 155.000 Personen diese Zusatzausbildung.

Der Certified-Tester hat gerade in Branchen mit Systemen, die viel Software enthalten und Softwarefehler potenziell hohen Schaden anrichten können, hohes Renommee. Dazu gehören die Automobilindustrie, die Medizintechnik, die Luftfahrt und die Finanzbranche. Auch in öffentlichen Ausschreibungen wird zunehmend nach der ISTQB-Certified-Tester-Qualifikation gefragt.

## Guter Arbeitsmarkt für Certified-Tester

Das German Testing Board richtet sein Angebot aber nicht nur an Industrieunternehmen und Behörden, auch Hochschulen brauchen den Certified-Tester – dazu arbeitet das GTB mit derzeit 40 Universitäten, Fachhochschulen und Berufsakademien zusammen. Der Stellenmarkt für Certified-Tester ist erfreulich, wie der Blick auf Online-Jobbörsen wie monster.de und stepstone.de zeigt: Rund die Hälfte setzen eine Zertifizierung nach ISTQB voraus oder beschreiben sie als hilfreich. Je nach Position wird eine Ausbildung im sogenannten Foundation- oder Advanced Level verlangt. Ersterer vermittelt die Grundlagen des Softwaretestens, der Advanced Level vertieft die Inhalte.

Auch jeder Entwickler profitiert von einer Foundation-Level-Ausbildung zum ISTQB Certified Tester. Im Testen geschulte Entwickler können die Programme ihrer Kollegen effizienter testen. Zudem machen Entwickler viele Fehler erst gar nicht, wenn sie schon beim Programmieren an die Testfälle denken und so die potenziellen (Denk-)Fehler im Ansatz erkennen. Tester mit höherer Qualifikation spezialisieren sich als Testmanager, als funktionale Tester oder auf technischen Testgebieten wie Testautomatisierung, Last- und Performancetests oder Sicherheits-Tests. Spätestens an der Stelle gilt: Tester benötigen ebenfalls Entwickler-Background.

## Fazit

Mit fortschreitenden Entwicklungstechniken wagt sich die Softwareindus-

trie an immer komplexere Aufgaben in zunehmend kritischeren Anwendungen. Große Fortschritte bei Softwareentwurf und -realisierung ermöglichen das. Aber auch wenn dadurch die Fehlerdichte im Code geringer wird, werden IT-Systeme auf der anderen Seite immer intensiver und in immer kritischeren Anwendungsfeldern genutzt. Deshalb zeigt das Risiko von Fehlerwirkungen im produktiven Einsatz eine insgesamt steigende Tendenz.

Wie in jeder Ingenieursbranche kann man bei der Software die Qualität nicht allein durch konstruktive Maßnahmen sicherstellen. Das unabhängige Testen muss mit dem Fortschritt in der Softwareentwicklung Schritt halten. Erst dadurch gewinnt das Unternehmen die benötigte Transparenz über die Qualität der Software und die Sicherheit vor unliebsamen Überraschungen. Zwei Aspekte sind für den Erfolg wichtig: die geeignete Organisation des Testprozesses und die professionelle Qualifikation der Tester. Letzteres wird in Deutschland durch Trainings und Zertifizierungen nach dem Certified-Tester-Lehrplan unterstützt. (ane)

### TILO LINZ

ist Vorsitzender des German Testing Board e.V. und Vorstand der imbus AG.

### DR. MATTHIAS HAMBURG

ist Mitglied im German Testing Board e.V. und Managing Consultant bei der Sogeti Deutschland GmbH.

## German Testing Board

[www.german-testing-board.info](http://www.german-testing-board.info)

### Über den German Testing Board e.V. (GTB)

Das GTB wurde 2005 als ehrenamtlicher Verein gegründet und hat sich zum Ziel gesetzt, das Testen von Software zu professionalisieren. Durch die Arbeit des GTB hat sich das Berufsbild „Softwaretester“ erst herausgebildet. Mittlerweile wurden in Deutschland bereits über 17.000 Studenten und Arbeitnehmer zum Certified-Tester ausgebildet, weltweit durchliefen bereits mehr als 155.000 Personen diese Zusatzausbildung. Das Ausbildungsprogramm wird derzeit in 47 Ländern angeboten. Nach Großbritannien ist Deutschland bei den Absolventenzahlen führend, dahinter folgen Indien und die USA.

Das International Software Testing Qualifications Board (ISTQB) definiert die Lehrpläne für die Certified-Tester-Ausbildung, nationale Organisationen wie das German Testing Board (GTB) setzen diese in die jeweilige Landessprache um und akkreditieren Provider, die Trainings anbieten. Die strikte Trennung von Trainings- und Prüfungswesen ist dabei gewährleistet: Zwei unabhängige Zertifizierungsinstitute, die Dienstleistungsgesellschaft für Informatik mbh und die iSQI GmbH, wickeln die Prüfungen ab.



### Über die Sogeti Deutschland GmbH

Als größter Dienstleister für Testlösungen in Europa und den USA mit über 20.000 IT-Professionals in 15 Ländern hilft Sogeti Unternehmen, deren Qualitätssicherungs- und Testleistungen zu verbessern. Wir unterstützen unsere Kunden, aus ihren IT-Systemen, den bestmöglichen geschäftlichen Nutzen zu ziehen. Die umfassenden Dienstleistungen der Sogeti – von Outsourcing in Test Centern bis zu Beratung in der Verbesserung von Testprozessen – helfen unseren Kunden, die finanzielle Kontrolle zu behalten und das Risiko zu reduzieren.

Unsere zentralen Leistungen sind:

#### **Managed Testing Services**

Sogetis erprobter Leistungsansatz, bei dem wir volle Verantwortung für Test und Qualitätssicherung übernehmen, jeweils individuell angepasst auf die Anforderungen unseres Kunden.

#### **TMap®-basierter Projekt- und Programmtest**

Ein weltweit anerkannter, erfolgreicher und pragmatischer Ansatz für den anforderungsorientierten, risikobasierten, strukturierten Softwaretest.

#### **TPI® - Geschäftsbasierte Testprozessverbesserung**

Das weltweit führende Modell zur Testprozessverbesserung berücksichtigt aktuelle Entwicklungen am Markt und in der Methodik, zudem wurde das Einsatzspektrum erweitert.

Zur Vorbereitung auf Zertifizierungen bietet Sogeti sowohl standardisierte als auch maßgeschneiderte Trainings an.