

4 Motivation für das Testen

»Wir bauen Software wie Kathedralen:
Zuerst bauen wir – dann beten wir.«

Gerhard Chroust

Von einem Praxisbuch erwartet der Leser, dass auf jeden Fall genug praxisrelevantes Material enthalten ist. Diese Erwartung finden Sie hier bestätigt. Doch was ist das Ziel und welches theoretische Fundament ist notwendig, um ausreichend informiert zu sein, damit Sie sich als Leser sofort zurechtfinden können?

Übersicht über das Kapitel:

- Die Selbstverständlichkeit des Softwaretestens
- Der Zusammenhang zwischen Testen und Qualität
- Der Vorteil der Risikominimierung

4.1 Tests geben Sicherheit

Ist Testen die Feststellung von Qualität, oder bedeutet es Fehler zu finden?

»Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden.«
Glenford, J. Myers

Möchten Sie in einem Auto sitzen, das nicht fahrtauglich ist? Halten Sie es für wichtig, dass die Sicherheitsprüfungen bei Ihrer Kreditkarte gemacht werden? Für welche Bank würden Sie sich entscheiden? Für die Bank, die mit ihrer langjährigen Zuverlässigkeit in puncto Sicherheit wirbt, oder für die Bank, die als »First Mover« in sehr innovativen IT-Technologien bekannt ist?

Die Entscheidung scheint für Sie persönlich klar zu sein. Aber wieso wird dem Testen häufig in Projekten kaum Gelegenheit gegeben, zur Ausführung zu kommen? Die Antwort liegt klar auf der Hand:

- Software enthält keine Fehler.
- Programmierer sind unfehlbar.

Oder sieht die Antwort eher so aus?

- Das Projekt wurde falsch geplant, für die Testaktivitäten stehen keine Ressourcen und kein Budget zur Verfügung.
- Dem Kunden wird eine Beta-Version¹ ausgeliefert, er wird dies schon akzeptieren und weiteres Budget zur Verfügung stellen, um dann in der Folgephase, als Beta-Tester, mögliche Fehler zu entdecken und entfernen zu lassen.

Diese dürfte schon eher der Praxis entsprechen. Solange die Kunden mitspielen und die Projekte trotz falscher Planung als Erfolg gefeiert werden, ist die Welt noch in Ordnung..Doch diese Einstellung könnte von kurzer Dauer sein. Sollte der Kunde diese Methode entlarven, dann könnte dies einen großen Imageverlust bedeuten – abgesehen vom Ärger beim Kunden und der zukünftigen Zusammenarbeit mit Ihnen als Softwarelieferant.

Test sind wichtige Maßnahmen zur Qualitätssicherung in allen Prozessphasen der Softwareentwicklung. Doch was wird unter Qualität bzw. in unserem Fall unter Softwarequalität verstanden? Schauen wir uns an, was die DIN² 55350 als Qualität definiert:

1. Eine Beta-Version ist eine unfertige Version eines Computerprogramms. Häufig sind Beta-Versionen die ersten Versionen eines Programms, die vom Hersteller zu Testzwecken veröffentlicht werden.

2. Deutsches Institut für Normung e.V., Berlin: nationale Normungsstelle der Bundesrepublik Deutschland

Qualität

»Die Gesamtheit von Eigenschaften und Merkmalen eines Produkts oder einer Tätigkeit, die sich auf deren Eignung zur Erfüllung gegebener Erfordernisse bezieht. Die Erfordernisse ergeben sich aus dem Verwendungszweck des Produktes oder dem Ziel der Tätigkeit, unter Berücksichtigung der Realisierungsmöglichkeiten.«

DIN 55350, Teil 11

Die internationale Norm DIN ISO³ 9126 definiert folgende Qualitätsmerkmale:

Qualitätsmerkmale

- Funktionalität
- Zuverlässigkeit
- Benutzbarkeit
- Effizienz
- Änderbarkeit
- Übertragbarkeit

Zu den Qualitätsmerkmalen ist es wichtig zu verstehen, dass es nicht immer möglich ist, diese gleichzeitig umzusetzen. Beispielsweise können die Punkte Zuverlässigkeit und Effizienz entgegenwirken. Ein sehr robustes, durch viele Sicherheitsprüfungen entwickeltes Softwaresystem wird ein höheres Zeitverhalten (Performance) benötigen. Aus diesem Grund ist eine Priorisierung der Qualitätsmerkmale bei den Qualitätsanforderungen des Projektes festzulegen.

Testen ist ein Maßsystem für die Softwarequalität. [Spillner 03]

Testen ermöglicht es, die Qualität von Software zu messen, ausgedrückt durch die Anzahl der gefundenen Fehler. Wenn wenige oder keine Fehler gefunden werden, kann Testen Vertrauen in das Softwaresystem schaffen. Durch gefundene Fehler und das Beheben dieser Fehler steigt die Qualität. Und höhere Qualität führt zu einer höheren Zuverlässigkeit des Softwaresystems.

3. Internationale Organisation für Normung. Mitglieder sind staatliche und nicht-staatliche Normenorganisationen.

4.2 Fehlerfindung und Kosten

Softwaresysteme werden von Menschen entwickelt. Menschen machen Fehler, sodass jedes Softwaresystem Fehler enthält.

Die Frage, die hier gestellt werden sollte, ist, wann an der Fehlerfindung gearbeitet werden sollte. Je früher Sie einen Fehler entdecken und diesen beseitigen, desto weniger kostet der Fehler! Zu dieser Thematik gibt es diverse Untersuchungen und Studien, die sehr aufschlussreich sind. Hier exemplarisch die Studie der Software Quality Systems Group aus Köln:

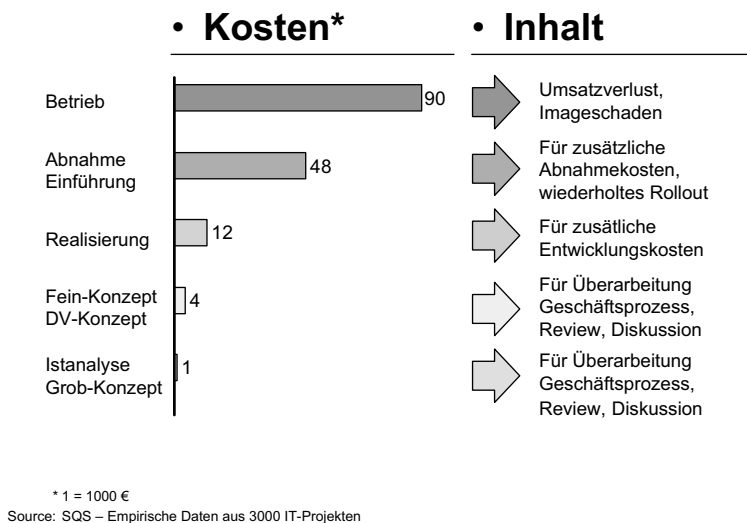


Abb. 4-1 Fehlerkosten in IT-Projekten (Quelle: SQS)

Die Praxis sieht so aus, dass die Fehler in den späten Entwicklungsprozessphasen am teuersten sind. Die Motivation sollte es also sein, die Fehler so früh wie möglich zu finden, denn wer zu spät testet, verschleudert Geld. Doch Unwissenheit, Budget- und Zeitdruck führen dazu, dass die Phase der qualitätssichernden Maßnahmen wie Testen häufig als nicht wichtig empfunden werden. Es wird nicht intensiv genug getestet, und das Finden der Fehler wird dem Kunden überlassen.

Was kosten Softwarefehler⁴?

- Geschätzte Verluste durch Softwarefehler in Mittelstands- und Großunternehmen in Deutschland: ca. 84,4 Mrd. EUR pro Jahr.

4. Untersuchung der LOT Consulting GmbH, Karlsruhe.
Siehe Nachricht des Onlineportals der Zeitschrift Computerwoche vom 15.02.2001:
<http://www.computerwoche.de/nachrichten/519152/>

- Ein Drittel der IT-Budgets müssen Unternehmen zur Beseitigung von Programmfehlern ausgeben.
- Produktivitätsverluste durch Computerausfälle belaufen sich auf durchschnittlich 2,6 Prozent des Umsatzes der Unternehmen und belaufen sich auf 70 Mrd. EUR pro Jahr.

4.3 Qualität des Produkts

Wieso wird der Aspekt der Qualität bei Softwareentwicklungen nicht wahrgenommen?

Häufig gibt es Entscheidungen, die die Randbedingungen von schlechten Programmen verbessern. Aber die eigentlichen Defizite werden nicht angegangen!

- Die Anforderungen sind so dokumentiert. Es wird das umgesetzt, was auf dem Papier steht.
- Bei Performanceproblemen wird Hardware dazugekauft, es wird die Prozessoranzahl verdoppelt. Aber die eigentlichen Ursachen werden ignoriert.
- Bei Anpassungen/Erweiterungen von bestehenden Programmen hört der Projektleiter häufig den Satz »Das Programm kann nicht erweitert werden, ich kopiere die ähnlichen Programmteile und schreibe ein neues Programm«.
- Oder bei der Nachfrage, ob das Programm detailliert getestet wurde, kommen Antworten wie »Die Testumgebung reicht nicht aus« oder »Das Programm ist so geschrieben, dass es nicht einzeln (Unit-Test-Ebene) getestet werden kann«.
- Viele Entwicklerkollegen denken häufig, dass einheitliche Programmierrichtlinien dazu dienen, sie zu ärgern bzw. ihnen den kreativen Spielraum wegzunehmen, der für ihre Arbeit wichtig ist!
- Aber die unglaublichste Aussage ist, dass der aktuelle Projektstand keine Ressourcen für Qualitätsprüfung/Qualitätssicherung zulasse.

Das mag ja richtig sein, aber die Qualität wird so oder so geprüft ... Entweder vor Auslieferung des Softwareproduktes oder durch den Kunden während der Produktivphase ...

Aber zum Glück ist dies den Verantwortlichen, wie dem Auftraggeber, Projektleitung, Teilprojektleitung und anderen Projektbeteiligten, bekannt und auch bewusst!

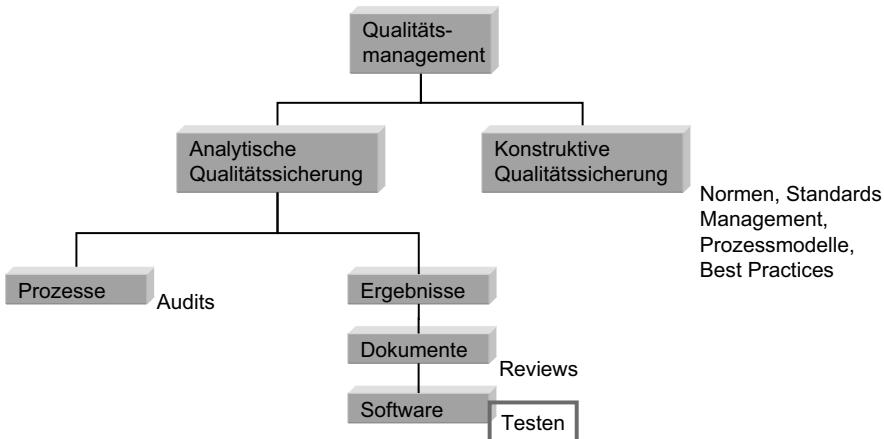


Abb. 4-2 Qualitätssicherung Überblick

Bei der Erstellung von Software sollten qualitätssichernde Maßnahmen angewendet werden. Das Qualitätsmanagement unterscheidet die analytischen und die konstruktiven Maßnahmen. Bei den analytischen Maßnahmen liegt der Fokus auf dem Softwareprodukt. Dabei ist das Ziel, dass Fehler gefunden werden. Die konstruktiven Maßnahmen sind auf den Prozess fokussiert. Das Ziel der konstruktiven Maßnahmen ist, dass Fehler beim Erstellen von Software vermieden werden.

Diverse qualitätssichernde Maßnahmen werden bei der Erstellung von Software angewandt. Das Testen weist die Zuverlässigkeit und damit die Qualität der Software aus. Eine gut getestete Software ist ein Garant für eine positive Außenwirkung beim Auftraggeber.

4.4 Potenzial

»Unsere Hauptaufgabe ist nicht, zu erkennen, was unklar in weiter Entfernung liegt, sondern zu tun, was klar vor uns liegt.«

Thomas Carlyle

Das Testen ist kein Garant dafür, dass in Softwaresystemen keine Fehler enthalten sind. Außerdem ist es unmöglich, Software komplett auszutesten. Der Punkt beim Testen ist, dass Softwaresysteme nicht ungetestet in Betrieb genommen werden sollten.

Das Testen von Softwaresystemen ist immens wichtig. Doch Wichtigkeit genügt dem Management nicht immer, das Testen wird häufig noch als der Flaschenhals in Projekten gesehen. Dies ist verwunderlich, wenn Sie sich das Potenzial⁵, das im Testen steckt, ansehen:

- 88 Prozent – Besseres Testen senkt die allgemeinen Entwicklungskosten.
- 78 Prozent – Mit mehr Zeit für das Testen können neue Services die Kunden- und Marktanforderungen erfüllen.
- 74 Prozent – Besser strukturiertes und auf dem Geschäftsrisiko basiertes Testen senkt die Entwicklungszeit.
- 68 Prozent – Zu viele Softwaresysteme gehen ohne ausreichendes Testen in Betrieb.
- 32 Prozent – Nicht ausreichendes Testen gefährdet die rechtzeitige Produktivsetzung.

Das mögliche Potenzial des Testens ist klar zu erkennen, und aus meiner Praxis stelle ich fest, dass dem Testen eine immer höhere Bedeutung zukommt. Auf allen Ebenen wird der Einsatz von qualitätssichernden Maßnahmen nicht mehr nur als Kostenpunkt, sondern als notwendige Investition empfunden, um die Qualität zu gewährleisten.

Sehen Sie das Testen einfach als Ihren Schutz an. Nehmen Sie einmal folgendes Zukunftsszenario an: Sie möchten Ihr Softwaresystem bei einer Versicherung gegen Fehler versichern! Im Grunde genommen ist dies nichts anderes, als einen Vertrag gegen einen möglichen Schaden zu erwerben – also das Prinzip einer jeden Versicherung. Die würde das Risiko eines jeden Fehlers ermitteln.

Risiko = möglicher Schaden x Wahrscheinlichkeit des Auftretens

Sie können sich vorstellen, dass sich je nach Geschäftsumfeld, in dem das Softwaresystem eingesetzt werden soll, astronomische Versicherungssummen ergeben könnten.

Naja, dies ist schon ein finanztechnisches Horrorszenario. Aber im Grunde genommen doch ein mögliches neues Versicherungsgebiet für die Versicherungsbranche.

Aus diesem Grunde ist das Testen Ihr Versicherungsschein, der Ihnen dazu verhilft, ein qualitativ hochwertiges Softwaresystem, da ja genügend getestet, mit gutem Gewissen Ihrem Kunden auszuliefern.

5. LogicaCMG Studie »Testing Times for Board Rooms«, die im September 2005 veröffentlicht wurde. Es wurden 255 Unternehmen aus England, den Niederlanden und Schweden befragt.

4.5 Kennzahlen und Metriken

»You cannot manage what you cannot measure.«

Bekannter Managementspruch

Dies ist ein etwas überstrapazierter Managementspruch. Doch der Grundgedanke ist aktueller denn je zuvor. Um Aussagen über den aktuellen Teststand in Ihrem Projekt zu geben oder um vernünftige Planungen für das Testen aufzustellen, benötigen Sie Informationen. Als Informationsquelle sollte nicht ausschließlich Ihr Bauchgefühl⁶ dienen.

Für jede Teststufe sind Testmetriken zu definieren und zu erheben. Dadurch erhalten Sie quantitative Aussagen zur Softwaresystemqualität und der Qualität des Entwicklungs- und Testprozesses. Diese Kennzahlen (siehe auch systematisches Testen auf Seite 51) sind die Basis für eine transparente und nachvollziehbare Planung und Steuerung Ihres Projektes.

Folgende Informationsquellen können Sie benutzen, um Ihre projektspezifischen Kennzahlen zu erhalten:

- Interne Richtlinien
- Best-Practices voriger Projekte zum Thema Testen
- Studien
- Aktuelle Projekt-Kennzahlen des Testcontrolling
 - Testfälle je Entwicklungsprozessphase
 - Testfälle zu Entwicklungskomponente
 - Testfälle zu Geschäftsprozess
 - Testfälle zu Schnittstelle
 - Testabdeckung
 - ...

6. Obwohl die menschliche Intuition ein sehr sicheres und auch durchaus erprobtes Orakel darstellt.

Unter Zuhilfenahme der Kennzahlen würde auf die Frage der Projektleitung:

»Wie ist der Stand beim Testen Ihres Softwaresystems?«

aus dieser Antwort:

»Ja, wir haben vieles ausprobiert, und es läuft.«

folgende Antwort werden:

»Anhand unserer internen Richtlinien wurden alle Einzelkomponenten einem Test mit funktionalen Eingabe-Äquivalenzklassen und Äquivalenzklassen für ungültige Werte getestet. Wir haben eine Zweigüberdeckung von 75 Prozent erreicht. Die Geschäftsprozesse inklusive Schnittstellen, die abgebildet werden sollten, haben wir mit 50 Testfällen je Geschäftsprozess erfolgreich getestet. Wir haben eine Fehlerrate von 5,3 Fehlern pro 10.000 Zeilen Coding. Im Vergleich zu den letzten drei Projekten in einem ähnlich gelagerten Umfeld haben wir eine qualitative Verbesserung von 20 Prozent erreicht!«

4.6 Der Auftraggeber ist zufrieden

Die Situation in der Softwarebranche ist einfach. Wenn der Auftraggeber unzufrieden ist, bezahlt er auch nicht für das in Auftrag gegebene Softwaresystem. Abgesehen von den finanziellen Einbußen ist auch ein Imageschaden vorprogrammiert, und ob sich das ein Softwarelieferant auf Dauer leisten kann, ist fraglich.

Sollte der Auftraggeber jedoch zufrieden sein, dann spiegelt sich diese Zufriedenheit auch möglicherweise in vollen Auftragsbüchern beim Softwarelieferanten wider, was daraufhin Zufriedenheit beim Auftragnehmer erzeugt.

Was erwartet der Auftraggeber?

- Umsetzung des Softwaresystems
 - gemäß den Anforderungen
 - termingetreu
 - innerhalb des Budgets
- qualitativ hochwertiges System
 - robust und richer
 - performant
 - wartbar und
 - flexibel für künftige Erweiterungen

Welche Möglichkeiten bieten sich dem Softwarelieferanten in puncto Testen, um diese Ziele zu erreichen?

- Testszenarien für alle Prozessphasen der Softwareentwicklung
- Qualitätskontrollen der Entwicklungskomponenten durch
 - ⇨ Reviews
 - ⇨ Design Pattern für Entwicklung und das Testen
 - ⇨ Refactoring unter Nutzung von automatisiert ablaufenden Tests der Entwicklungskomponenten und der dazugehörigen Geschäftsprozesse

Qualitativ hochwertige, da getestete Software ist einer der wesentlichsten Punkte, damit der Auftraggeber zufrieden ist.

4.7 Zusammenfassung

- Software enthält Fehler. Aus diesem Grunde sollte Software getestet werden, um eine qualitative Sicherung der Software zu gewährleisten.
- Das Testen von Software ist genauso wichtig wie das eigentliche Programmieren der Software.
- Je früher die Phase des Testens beginnt, desto günstiger ist die Beseitigung von Fehlern. Am teuersten ist die Fehlerbeseitigung, wenn das Softwaresystem schon in Betrieb genommen wurde.
- Softwarequalität wird vom Auftraggeber wahrgenommen. Qualitätssichernde Maßnahmen dienen dazu, die Qualität des Softwaresystems zu erhöhen.
- Enormes Potenzial liegt beim Testen. Durch Einsatz des Testens ist Einsparpotenzial bei Zeit, Kosten und Ressourcenaufwand in Projekten möglich.
- Kennzahlen und Metriken sollten insbesondere für das Testen aufgestellt werden. Eine transparente und auf Fakten gesicherte Planung und Steuerung Ihres Projektes ist dadurch gewährleistet.
- Erfüllen Sie die Erwartungen des Auftraggebers in das zu erstellende Softwaresystem. Ein zufriedener Auftraggeber dankt es Ihnen mit einer guten Geschäftsbeziehung zu Ihnen.