

# 1 Einleitung

Unser Alltag ist wie nie zuvor abhängig von Software und softwarebasierten Systemen. Es gibt kaum noch Geräte, Maschinen oder Anlagen, deren Funktion oder Steuerung nicht über Software bzw. Softwareanteile realisiert wird. Aber auch Verwaltungsvorgänge in Industrie und Staat werden durch oft komplexe IT-Systeme getragen. Die Verwaltung von Versicherungspolicen, das Mautsystem »TollCollect«, biometrische Merkmale in Pass und Personalausweis oder die elektronische Gesundheitskarte sind hierfür Beispiele.

*Große Abhängigkeit  
von Software*

Diese starke Abhängigkeit von Software erfordert immer höhere Investitionen in qualitätssichernde Maßnahmen, damit die IT-Systeme möglichst zuverlässig ihre Aufgaben erfüllen. Das Testen von Software entwickelt sich vor diesem Hintergrund zu einer spezialisierten, eigenständigen Fachrichtung und Berufsdisziplin der Informatik.

*Testen von Software  
ist eine eigenständige  
Berufsdisziplin*

Innerhalb der Disziplin Softwaretest hat das Thema »Testmanagement« besondere Bedeutung. Das Testmanagement umfasst klassische Methoden des Projektmanagements und des Risikomanagements sowie das Wissen um den zweckmäßigen Einsatz wohldefinierter Testmethoden. Mit diesem Handwerkszeug ausgerüstet, kann der Testmanager<sup>1</sup> geeignete Maßnahmen zielgerichtet auswählen und umsetzen, die sicherstellen, dass eine bestimmte Mindestproduktqualität erreicht wird. Er verfolgt dabei ein ingenieurmäßiges Vorgehen.

*Testmanagement*

Während die Ausbildung zum Projektmanager etabliert ist und eine Vielzahl von Studiengängen, Ausbildungsprogrammen und Spezialliteratur existiert (s. beispielsweise [Hindel 06]), waren die Ausbildungsinhalte zum »Software Testmanager« bislang kaum definiert oder gar standardisiert. Angesichts der steigenden Verantwortung, die

*Ausbildung für  
Testmanager*

---

1. Wir verwenden im Buch die männliche Form und wollen damit Frauen selbstverständlich nicht ausschließen bzw. ausgrenzen.

Testmanager im Rahmen ihrer Tätigkeit übernehmen, war das ein wenig erfreulicher Zustand.

*ISTQB Certified Tester  
Advanced Level  
– Testmanager*

Mit dem »ISTQB Certified Tester – Advanced Level – Testmanager« steht erstmals ein international anerkanntes Ausbildungsschema zur Verfügung, das auch für den Beruf des Testmanagers Lehrinhalte und Qualifizierungsmodule definiert. Das vorliegende Buch »Praxiswissen Softwaretest – Testmanagement« vermittelt diese Lehrinhalte und kann als Lehrbuch bei der Vorbereitung auf die entsprechende Zertifizierung dienen.

*Foundation Level*

Das »ISTQB Certified Tester«-Qualifizierungsprogramm ist dreistufig aufgebaut. Die Grundlagen des Softwaretests sind im Lehrplan »Foundation Level« beschrieben [URL: GTB CTF]. Dieser Lehrstoff ist im Buch »Basiswissen Softwaretest« [Spillner 05] ausführlich dargestellt.

*Advanced Level*

Der »Advanced Level« Lehrplan [URL: GTB CTA] umfasst weiterführende Kenntnisse im Prüfen und Testen von Software und zeigt Spezialisierungsmöglichkeiten auf:

- die vertiefte Behandlung von verschiedenen Blackbox- und Whitebox-Testverfahren in den »Advanced Level« Modulen »Technical Tester« und »Functional Tester« sowie
- die vertiefte Darstellung von Methoden und Techniken des Testmanagements im Modul »Testmanager«.

Diese Aufteilung entspricht auch der Struktur<sup>2</sup> des Lehrstoffs, wie sie von vielen akkreditierten Weiterbildungsanbietern vorgenommen wird. Da der Lehrstoff des »Advanced Level« sehr umfassend ist, wird dieser im vorliegenden Buch nicht komplett behandelt, sondern ausschließlich das Modul »Advanced Level – Testmanager«, allerdings ohne Berücksichtigung des Themas »Reviews«<sup>3</sup>.

*Expert Level*

Die dritte Stufe »Expert Level« wird von Arbeitsgruppen derzeit definiert und umfasst Inhalte wie beispielsweise die Besonderheiten beim Testen von objektorientierter entwickelter Software, fundiertes Wissen zu TTCN-3 (Testing & Test Control Notation, [URL: TTCN-3]), vertiefte Kenntnisse zu Vorgehensweisen bei der Testprozessverbesserung und andere Spezialgebiete des Softwaretests.

*International Software  
Testing Qualifications  
Board (ISTQB)*

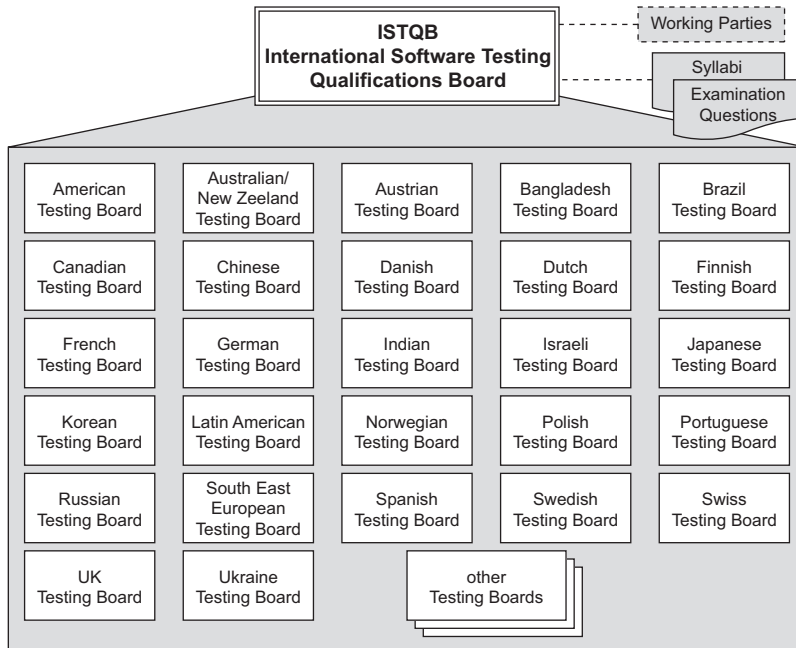
Das »ISTQB« [URL: ISTQB] sorgt für die Einheitlichkeit und Vergleichbarkeit der Lehr- und Prüfungsinhalte unter allen beteiligten Ländern. In ihm sind mittlerweile weltweit mehr als 25 nationale Initi-

2. Die derzeit in Entwicklung befindliche neue Version des »ISTQB Advanced Level Syllabus« wird diese Modulstruktur aller Voraussicht nach übernehmen und fortführen.
3. Zum Thema Reviews siehe beispielsweise [Gill 96].

ativen und Verbände zusammengeschlossen (s. Abb. 1-1, Stand vom Mai 2006). Weitere nationale Boards werden hinzukommen.

Die nationalen Testing Boards sind in ihrem Land als unabhängige Expertengremien dafür zuständig, Ausbildung (Akkreditierung der Weiterbildungsanbieter) und Prüfungen (Zertifizierung durch eine unabhängige Institution) in den jeweiligen Sprachen und Ländern zu ermöglichen und die Einhaltung der ISTQB-Standards zu überwachen.

*Nationale Testing Boards*



**Abb. 1-1**  
ISTQB-Struktur

Die drei ISTQB-Ausbildungsstufen bauen aufeinander auf. Das vorliegende Buch »Praxiswissen Softwaretest – Testmanagement« setzt den Stoff des »Foundation Level« voraus. Lesern, die neu in das Thema Softwaretest einsteigen, wird daher empfohlen, sich den Stoff des »Foundation Level« anzueignen. Dies kann durch den Besuch eines akkreditierten Seminars erfolgen oder durch das Durcharbeiten des Buches »Basiswissen Softwaretest« [Spillner 05]. Im vorliegenden Buch werden lediglich knappe Wiederholungen der wichtigsten Grundlagen geboten.

*Basiswissen wird vorausgesetzt*

## 1.1 Basiswissen – komprimiert

Im Folgenden wird der Inhalt des Lehrplans »Foundation Level« und somit auch das Buch »Basiswissen Softwaretest« kurz zusammengefasst.

*Maßnahmen zur Verbesserung der Softwarequalität*

Es gibt eine Vielzahl von Ansätzen und Vorschlägen, die Qualität der Software durch vorbeugende (konstruktive) Maßnahmen und den Einsatz von prüfenden (analytischen) Verfahren und Methoden zu verbessern. Zu den wichtigsten Maßnahmen gehören:

- Definierte Softwareentwicklungsprozesse, die zu einer strukturierten und nachvollziehbaren Erstellung der Softwaresysteme beitragen.
- Ein wohldefinierter Testprozess und ein geordnetes Änderungs- und Fehlermanagement als Voraussetzungen, um die Testarbeiten wirtschaftlich und wirksam durchzuführen.
- Verwendung von Metriken und Qualitätskennzahlen, die helfen, Softwareprodukte und Entwicklungsprozesse objektiv zu bewerten, Verbesserungspotenziale aufzudecken und die Wirksamkeit von Korrektur- oder Verbesserungsmaßnahmen zu überprüfen.
- Der Einsatz von formalen Methoden, die eine präzise Formulierung der Entwicklungsdokumente und damit deren Überprüfbarkeit bzw. Auswertung durch Werkzeuge ermöglichen.
- Methoden zur systematischen Ermittlung und Durchführung von Testfällen, die für eine effiziente Erkennung von Fehlern und Unstimmigkeiten in den entwickelten Programmen sorgen.
- Methoden zur statischen Prüfung, in erster Linie Reviews, durch die Fehler und Mängel frühzeitig in den erstellten Entwurfsdokumenten aufgedeckt werden.

*Qualitätsziele und Qualitätsmerkmale*

Testmanager müssen diese Methoden, Techniken und Prozesse beherrschen oder zumindest kennen, um im Projektverlauf die der jeweiligen Situation angemessenen Maßnahmen auswählen und anwenden zu können. Die Eignung von qualitätssichernden Maßnahmen ist aber auch abhängig von den jeweils gesetzten Qualitätszielen. Das geforderte Qualitätsniveau kann dabei anhand verschiedener Qualitätsmerkmale definiert werden. Einen Katalog solcher Qualitätsmerkmale (z.B. Funktionalität, Zuverlässigkeit oder Benutzbarkeit) definiert die Norm [ISO 9126].

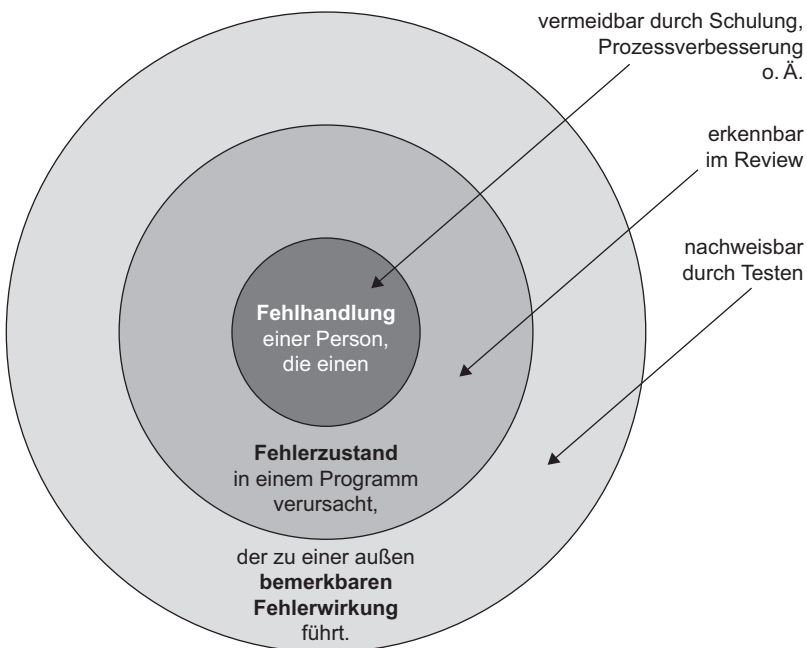
*Testorakel*

Wann liegt ein Defekt oder Fehler vor und was ist unter diesen Begriffen zu verstehen? Eine Situation oder ein Ergebnis kann nur dann als fehlerhaft eingestuft werden, wenn vorab festgelegt wurde, wie die erwartete, korrekte Situation bzw. das erwartete Ergebnis aussieht. Wird eine Abweichung zwischen dem erwarteten Istverhalten

und dem beobachteten Sollverhalten festgestellt, liegt ein Fehler vor. Um Sollwerte bzw. das Sollverhalten zu ermitteln, ist ein so genanntes Testorakel erforderlich, das dem Tester als Informationsquelle dient. Anforderungsdokumente, eine formale Spezifikation oder auch das Benutzungshandbuch sind Beispiele für solche Informationsquellen.

Der Begriff »Fehler« ist unpräzise. Es ist zwischen Fehllhandlung (*error*), Fehlerzustand (*fault*) und Fehlerwirkung (*failure*) zu unterscheiden. Eine Fehllhandlung einer Person führt beispielsweise zu einer fehlerhaften Programmierung. Dadurch enthält das Programm einen Fehlerzustand, der zu einer »außen« sichtbaren Fehlerwirkung führen kann, aber nicht zwangsläufig führen muss. Meist kommt ein Fehlerzustand erst bei nicht alltäglichen Situationen zum Tragen, z.B. kommt eine fehlerhafte Berechnung des Schaltjahrs erst am 29. Februar eines Schaltjahrs zur Wirkung. Abbildung 1–2 soll den Zusammenhang zwischen Fehllhandlung, Fehlerzustand und Fehlerwirkung veranschaulichen und darstellen, welche Gegenmaßnahmen bzw. Methoden zur Aufdeckung angewendet werden können.

Fehlerbegriff



**Abb. 1–2**

Zusammenhang zwischen den Fehlerbegriffen

Ähnlich dem Fehlerbegriff ist auch der Begriff »Testen« mit verschiedenen Bedeutungen belegt. Mit Testen wird oft der gesamte Prozess bezeichnet, ein Programm auf systematische Weise zu prüfen, um Vertrauen in die korrekte Umsetzung der Anforderungen<sup>4</sup> zu gewinnen und um Fehlerwirkungen nachzuweisen. Es ist auch ein Oberbegriff

Testbegriff

für alle Tätigkeiten und (Test-)Stufen im Testprozess. Jede einzelne Ausführung eines Testobjekts unter spezifizierten Bedingungen zum Zwecke der Überprüfung der Einhaltung der erwarteten Ergebnisse wird ebenso als Testen bezeichnet.

*Fundamentaler Testprozess* Testen umfasst eine Vielzahl von Einzelaktivitäten. Folgender fundamentaler Testprozess ist im Lehrplan »Foundation Level« definiert. Zum Prozess gehören folgende Aktivitäten:

- Testplanung und Steuerung,
- Testanalyse und Testdesign,
- Testrealisierung und Testdurchführung,
- Testauswertung und Bericht,
- Abschluss der Testaktivitäten.

*Teststufen* Beim Testen kann das zu testende Produkt (Testobjekt) auf unterschiedlichen Abstraktionsebenen bzw. auf der Basis unterschiedlicher Dokumente und Entwicklungsprodukte betrachtet werden. Die entsprechende Bezeichnung ist Teststufe. Es wird zwischen den Stufen Komponententest, Integrationstest, Systemtest und Abnahmetest unterschieden. Jede Teststufe zeichnet sich durch charakteristische Testziele, Testmethoden und Testwerkzeuge aus.

*Testarten* Daneben werden →Testarten unterschieden, die sich wie folgt abgrenzen lassen: funktionaler Test, nichtfunktionaler Test, strukturbezogener Test und änderungsbezogener Test [Spillner 05, Kap. 3.7].

*Statische und dynamische Prüfung* Beim Testen kann unterschieden werden, ob das Testobjekt zur Prüfung ausgeführt wird oder ob »nur« der zugehörige Programmtext oder die zugrunde liegende Spezifikation oder Dokumentation geprüft wird. Im ersten Fall sind es so genannte dynamische Prüfungen (mit den Vertretern Blackbox- und Whitebox-Testverfahren) [Spillner 05, Kap. 5], im zweiten Fall sind es statische Prüfungen (vertreten u.a. durch verschiedene Reviewarten) [Spillner 05, Kap. 4].

*Unabhängigkeit zwischen Test und Entwicklung* Unabhängig davon, welche Methoden zum Testen eingesetzt werden, gilt, dass Entwicklung/Programmierung und Test organisatorisch möglichst getrennt bzw. unabhängig voneinander ablaufen sollen. Denn ein Entwickler, der sein eigenes Programm testet, ist »blind« gegenüber eigenen Fehlhandlungen. Wer weist sich schon gerne seine eigenen Fehler nach?

*Testwerkzeuge* Für das Testen von Software gibt es eine Vielzahl unterstützender Werkzeuge. Je nach Einsatzzweck werden verschiedene Werkzeugklassen unterschieden: u.a. Werkzeuge für Management und Steuerung

---

4. Mit Testen kann nicht nachgewiesen werden, dass die Anforderungen zu 100 % erfüllt sind, da Testen nur stichprobenartige Überprüfungen vornimmt.

von Tests, Werkzeuge zur Testspezifikation, zum statischen und dynamischen Test und für nichtfunktionalen Test [Spillner 05, Kap. 7].

Im »Foundation Level« werden auch schon die grundlegenden Aspekte des Testmanagements behandelt. Neben Testplanung, Teststeuerung und Berichtswesen gehören hierzu auch die Themen Fehler-, Änderungs- und Konfigurationsmanagement sowie das Thema Wirtschaftlichkeit des Testens [Spillner 05, Kap. 6]. Das vorliegende Buch vertieft diese Aufgaben des Testmanagements.

Zur Veranschaulichung des Stoffs wird in diesem Buch das Fallbeispiel aus dem »Basiswissen«-Buch fortgesetzt:

Testmanagement

Ein Automobilkonzern entwickelt ein neues elektronisches Verkaufssystem, genannt *VirtualShowRoom* (VSR). Das Softwaresystem soll in der Endausbaustufe weltweit bei allen Händlern installiert sein. Jeder Kunde, der ein Fahrzeug erwerben möchte, kann dann unterstützt durch einen Verkäufer oder vollkommen selbstständig sein Wunschfahrzeug am Bildschirm konfigurieren (Modellauswahl, Farbe, Ausstattung usw.).

Das System zeigt mögliche Modelle und Ausstattungsvarianten an und ermittelt zu jeder Auswahl des Kunden sofort den jeweiligen Listenpreis. Diese Funktionalität wird vom Teilsystem *DreamCar* realisiert.

Hat sich der Kunde für ein Fahrzeug entschieden, kann er am Bildschirm die für ihn optimale Finanzierung kalkulieren (*EasyFinance*), das Fahrzeug online bestellen (*JustInTime*) und bei Bedarf auch die passende Versicherung (*NoRisk*) abschließen. Das Teilsystem *ContractBase* verwaltet sämtliche Kundeninformationen und Vertragsdaten. Abbildung 1–3 zeigt eine schematische Darstellung des Systems.

Fallbeispiel

»VirtualShow-Room« –  
VSR

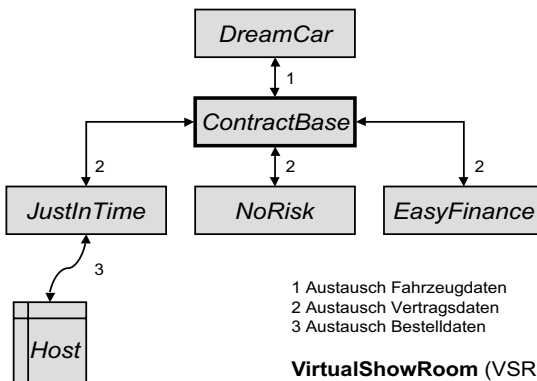


Abb. 1–3

Architektur des  
VSR-Systems

Jedes Teilsystem wird von einem eigenen Entwicklungsteam separat entworfen und entwickelt. Insgesamt sind ca. 50 Entwickler und weitere Mitarbeiter aus den jeweils betroffenen konzerninternen Fachabteilungen an dem Projekt beteiligt sowie externe Softwarefirmen.

Im »Basiswissen«-Buch wurden die verschiedenen →Testtechniken und Vorgehensweisen beschrieben, um das System gründlich zu testen, bevor das VSR-System in Betrieb geht.

Die Entwicklung des VSR-2 folgt einem iterativen Entwicklungsprozess. Aus dem vorhandenen VSR-1 soll mit vier aufeinander folgenden Iterationen der VSR-2 entstehen. Dafür ist eine Entwicklungsdauer von einem Jahr vorgesehen. Es wird also etwa quartalsweise eine Zwischenversion geben.

Jede neue Version soll die Funktionalität der Vorgängerversion weiterhin korrekt bereitstellen. Allerdings kann dem eine andere, vielleicht bessere oder effizientere Implementierung zugrunde liegen. Zusätzlich implementiert jede Version erstmalig einen Satz neuer Funktionen.

Der Produktmanager erwartet vom Testmanager daher zweierlei:

- Zum einen muss das Testteam sicherstellen, dass jede VSR-2-Version die bisherige Altfunktionalität korrekt enthält.
- Zum anderen soll das Testteam möglichst schnell eine objektive Beurteilung abgeben, ob bzw. wie gut ein neues *Feature* umgesetzt ist.

Die Aufgaben, die bei einer solchen Problemstellung vom Testmanager zu erfüllen sind, werden in den folgenden Kapiteln behandelt und anhand obigen Beispiels jeweils verdeutlicht.

---

## 1.2 Praxiswissen Testmanagement – Übersicht

*Praxiswissen-  
Kapitelübersicht*

Die Themen des Buches und die Inhalte der einzelnen Kapitel sind im Folgenden kurz beschrieben.

- In Kapitel 2 werden der grundlegende Testprozess und die Arten von Werkzeugen, die im Testprozess eingesetzt werden können, erörtert. Beides ist im Buch »Basiswissen Softwaretest« bereits behandelt worden.
- Wie das Testen in Verbindung zum Softwarelebenszyklus steht, wird in Kapitel 3 dargestellt. Unterschiedliche Vorgehensmodelle der Softwareentwicklung werden diskutiert und die jeweilige Bedeutung des Testens im Modell bewertet.
- Welchen Stellenwert das Testen in einer Organisation einnimmt, ist von großer Bedeutung für den Testmanager. Qualitäts- und Testpolitik der Organisation müssen von Management festgelegt werden. Kapitel 4 behandelt diesen Themenkomplex.
- Kapitel 5 geht näher auf die Testplanung ein, eine der wichtigen, wenn nicht sogar die wichtigste Aufgabe des Testmanagers.
- Die Planung muss während des Projekts angepasst werden. Eine Steuerung des Testprozesses auf Grundlage von Berichten über den Testfortschritt ist für den Testmanager eine entscheidende Maß-



nahme, um den Testprozess erfolgreich durchführen zu können. Kapitel 6 geht auf diesen Aspekt ein.

- Der Entwicklungs- und Testprozess selbst kann bewertet und verbessert werden. Welche Verfahren und Vorgehensweisen anzuwenden sind, wird in Kapitel 7 beschrieben.
- Wie ist mit den beim Testen gefundenen Abweichungen und Fehlerwirkungen umzugehen? Antworten hierzu gibt Kapitel 8.
- Risikobewertung und risikobasierte Tests sind für den Testmanager wichtige Instrumente zur Verteilung der beschränkten Testkapazitäten und dienen zur risikomindernden Steuerung des Testprojekts. In Kapitel 9 sind entsprechende Hinweise zum Vorgehen zu finden.
- Ohne Mitarbeiter mit den erforderlichen Fähigkeiten und Qualifikationen – ohne Berücksichtigung des »Faktors Mensch« – kann der Testmanager eine Aufgabe nicht erfolgreich durchführen. In Kapitel 10 wird beschrieben, was bei der Zusammenstellung des Testteams zu berücksichtigen ist.
- Testmetriken helfen, Kriterien für die Beendigung des Testens festzulegen und Aussagen über die Qualität des Testobjekts zu ermitteln. Beispiele hierfür werden in Kapitel 11 gegeben.
- Mit entsprechender Werkzeugunterstützung lässt sich der Testprozess meist effizienter durchführen. Wie der Testmanager die Werkzeuge auswählt und einführt, wird in Kapitel 12 beschrieben.
- Im letzten Kapitel des Buches werden die relevanten Normen und Standards vorgestellt.

Das Glossar enthält alle hier im Buch neu verwendeten Begriffe. Die Glossareinträge aus dem Buch »Basiswissen Softwaretest« [Spillner 05] finden Sie auch unter [URL: GTB Glossar].