

**Praxiswissen Softwaretest –  
Testmanagement**



**Andreas Spillner** ist Professor für Informatik an der Hochschule Bremen, Fakultät für Elektrotechnik und Informatik. Er war über 10 Jahre Sprecher der Fachgruppe TAV »Test, Analyse und Verifikation von Software« der Gesellschaft für Informatik e.V. (GI) und bis Ende 2009 Mitglied im German Testing Board e.V. 2007 ist er zum Fellow der GI ernannt worden. Seine Arbeitsschwerpunkte liegen im Bereich Softwaretechnik, Qualitätssicherung und Testen.



**Thomas Roßner** ist Mitgründer der imbus AG und in deren Vorstand verantwortlich für Forschung und Technologie des Unternehmens. In dieser Funktion leitete er in den vergangenen Jahren mehrere internationale Forschungsprojekte, u.a. zum Thema Softwarezuverlässigkeit und modellbasiertes Testen. Darüber hinaus arbeitet er aktiv in Testmanagementprojekten und Beratungsprojekten zum Thema Testprozessverbesserung.



**Mario Winter** ist Professor am Institut für Informatik der Fachhochschule Köln und dort Mitglied des Forschungsschwerpunktes »Software-Qualität«. Er ist Mitglied im German Testing Board e.V. und war von 2003 bis Anfang 2011 Sprecher der Fachgruppe »Test, Analyse und Verifikation von Software« im Fachbereich Softwaretechnik der Gesellschaft für Informatik (GI). Seine Lehr- und Forschungsschwerpunkte sind Softwareentwicklung und Projektmanagement, insbesondere die modellbasierte Entwicklung und Qualitätssicherung von Software.



**Tilo Linz** ist Vorstand der imbus AG, eines führenden Dienstleisters für Softwaretest. Er ist Leiter des German Testing Board e.V. und war von 2002 bis 2005 Vorsitzender des ISTQB. Zu seinen Arbeitsschwerpunkten zählen die Themen Berufsbild und Ausbildung im Softwaretest sowie die Optimierung von Softwaretestprozessen.

**Andreas Spillner · Thomas Roßner · Mario Winter · Tilo Linz**

# **Praxiswissen Softwaretest – Testmanagement**

**Aus- und Weiterbildung zum Certified Tester –  
Advanced Level nach ISTQB-Standard**

4., überarbeitete u. erweiterte Auflage



dpunkt.verlag

Andreas Spillner  
Andreas.Spillner@hs-bremen.de

Thomas Roßner  
thomas.rossner@imbus.de

Mario Winter  
mario.winter@fh-koeln.de

Tilo Linz  
tilo.linz@imbus.de

Lektorat: Christa Preisendanz  
Copy-Editing: Ursula Zimpfer, Herrenberg  
Satz & Herstellung: Birgit Bäuerlein  
Umschlaggestaltung: Helmut Kraus, [www.exclam.de](http://www.exclam.de)  
Druck und Bindung: Media-Print Informationstechnologie, Paderborn

Bibliografische Information der Deutschen Nationalbibliothek  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;  
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-86490-052-5

4., überarbeitete und erweiterte Auflage 2014  
Copyright © 2014 dpunkt.verlag GmbH  
Wiebinger Weg 17  
69123 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

# 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 hat sich vor diesem Hintergrund zu einer spezialisierten, eigenständigen Fachrichtung und Berufsdisziplin der Informatik entwickelt. Dies belegen auch die Ergebnisse der 2011 in Deutschland, Österreich und der Schweiz durchgeführten Umfrage zu Entwicklungen und Trends im Bereich Testen und Qualitätssicherung in Unternehmen. Über 80% der Befragten befürworten spezielle Weiterbildungen im Bereich der Qualitätssicherung [URL: Softwaretest-Umfrage].

*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 Testentwurfsverfahren. Mit diesem Handwerkszeug ausgerüstet, kann der Testmanager<sup>1</sup> geeignete Maßnahmen zielgerichtet auswählen und umsetzen, die sicherstellen, dass eine bestimmte Mindestqualität des Produkts erreicht wird. Er verfolgt dabei ein ingenieurmäßiges Vorgehen.

*Testmanagement*

---

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

*Ausbildung für  
Testmanager*

Während die Ausbildung zum Projektmanager seit Langem etabliert ist und eine Vielzahl von Studiengängen, Ausbildungsprogrammen und Spezialliteratur existiert (s. beispielsweise [Hindel 09], [Spitzok von Brisinski 2010], [Pichler 07] oder [Pichler 11]), waren die Ausbildungsinhalte zum »Softwaretestmanager« lange Zeit kaum definiert oder gar standardisiert. Angesichts der steigenden Verantwortung, die Testmanager im Rahmen ihrer Tätigkeit übernehmen, war das ein unerfreulicher Zustand.

*ISTQB Certified Tester  
– Advanced Level  
– Testmanager*

Mit dem »ISTQB Certified Tester – Advanced Level – Testmanager« steht mittlerweile 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« (s. [Spillner 12]) ausführlich dargestellt.

*Advanced Level*

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

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

Diese Aufteilung entspricht auch der Struktur 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«.

*Expert Level*

Die dritte Stufe, »Expert Level«, richtet sich an erfahrene, professionelle Softwaretester und besteht aus einer Reihe von Modulen zu unterschiedlichen Spezialthemen. Seit 2011 sind die Lehrpläne zu den Modulen »Improving the Testing Process – Implementing Improvement and Change« [Bath 14] und »Test Management – Managing Testing, Testers, and Test Stakeholders« veröffentlicht [URL: GTB CTE]. Geplant sind weitere Themen wie Test Automation, Security Testing, TTCN-3 [URL: TTCN-3] u. a.

Es ist geplant, aktuelle Themen oder branchenorientierte Spezialgebiete im »Certified-Tester«-Schema bedarfsorientiert und kurzfristig im Rahmen sogenannter »Extensions« der Foundation- und Advanced-Level-Lehrpläne zu berücksichtigen. Als erster Extension-Baustein wird in 2014 »Foundation Level Extension Syllabus Agile Tester« veröffentlicht.

*Add-on-Erweiterungen*

Das »ISTQB« [URL: ISTQB] sorgt weltweit für die Einheitlichkeit und Vergleichbarkeit der Lehr- und Prüfungsinhalte unter allen beteiligten Ländern. In ihm sind mittlerweile knapp 50 nationale Initiativen und Verbände aus über 70 Ländern zusammengeschlossen. Weitere nationale Boards werden hinzukommen.

*International Software  
Testing Qualifications  
Board (ISTQB)*

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

*Nationale Testing Boards*

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« (s. [Spillner 12]). 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.

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:

*Maßnahmen zur  
Verbesserung der  
Softwarequalität*

- Definierte Softwareentwicklungsprozesse (inkl. agiler Vorgehensweisen), 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 Entwicklungsdokumenten 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]<sup>2</sup> (s.a. [ISO 25010]).

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  $\rightarrow^3$  Abweichung zwischen dem beobachteten Istverhalten und dem erwarteten Sollverhalten festgestellt, liegt ein Fehler vor. Um Sollwerte bzw. das Sollverhalten zu ermitteln, ist eine Testbasis bzw. ein sogenanntes Testorakel als Informationsquelle erforderlich. Anforderungsdokumente, eine formale Spezifikation oder auch das Benutzungshandbuch sind Beispiele für solche Informationsquellen.

Fehlerbegriff

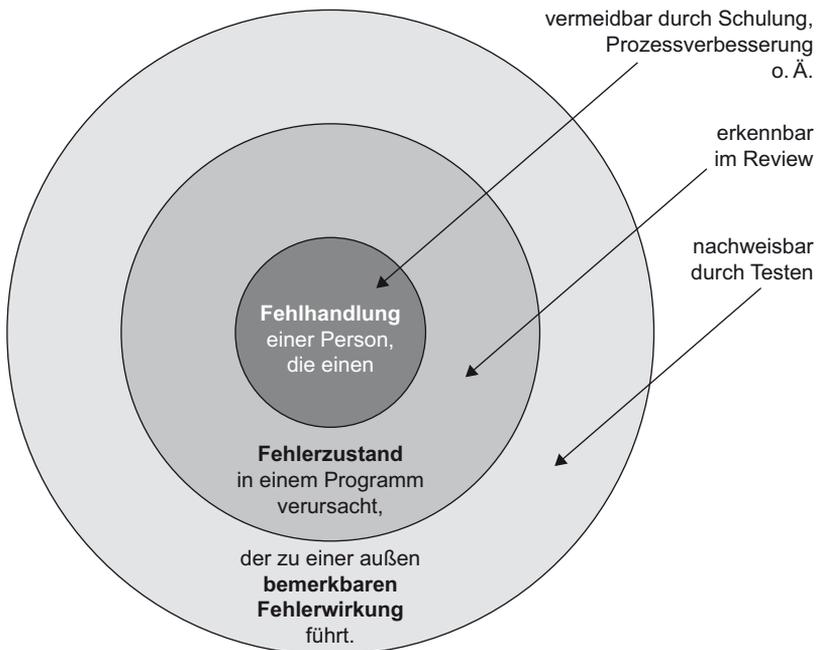
Der Begriff »Fehler« ist unpräzise. Es ist zwischen Fehlhandlung (engl. *error*), Fehlerzustand (engl. *fault*) und Fehlerwirkung (engl. *failure*) zu unterscheiden. Eine Fehlhandlung einer Person führt beispielsweise zu einer fehlerhaften Programmierung. Dadurch enthält das Programm einen Fehlerzustand, der zu einer »von außen« sichtbaren

2. Die ISO-Norm 9126 ist durch die neue ISO-Norm 25010 abgelöst worden, wird aber zurzeit noch im »Foundation Level«-Lehrplan referenziert.
3. Mit dem Pfeilsymbol werden Begriffe gekennzeichnet, die im Glossar im Anhang des Buches erläutert werden.

Fehlerwirkung führen kann, aber nicht zwangsläufig führen muss. Meist kommt ein Fehlerzustand erst bei nicht alltäglichen Situationen zum Tragen, z.B. wirkt sich eine fehlerhafte Berechnung des Schaltjahrs erst am 29. Februar eines Schaltjahrs aus. Abbildung 1–1 soll den Zusammenhang zwischen Fehlhandlung, Fehlerzustand und Fehlerwirkung veranschaulichen und darstellen, welche Gegenmaßnahmen bzw. Methoden zur Aufdeckung angewendet werden können.

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 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.

*Testbegriff*



**Abb. 1–1**  
Zusammenhang zwischen den Fehlerbegriffen

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

*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 Testentwurf,
- Testrealisierung und Testdurchführung,
- Bewertung von Endkriterien 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, Testentwurfsverfahren und Testwerkzeuge aus.

*Testarten* Daneben werden →Testarten unterschieden, die sich wie folgt abgrenzen lassen: funktionaler Test, nicht funktionaler Test, strukturbasierter Test und änderungsbezogener Test (s. [Spillner 12, Abschnitt 3.7]).

*Statische und dynamische Prüfung* Beim Testen kann unterschieden werden, ob das Testobjekt zur Prüfung auf dem Rechner ausgeführt wird oder ob »nur« der zugehörige Programmtext, die zugrunde liegende Spezifikation oder Dokumentation geprüft wird. Im ersten Fall handelt es sich um sogenannte dynamische Prüfungen (mit den Vertretern Blackbox- und Whitebox-Testentwurfsverfahren, s. [Spillner 12, Kap. 5]), im zweiten Fall um statische Prüfungen (vertreten u.a. durch verschiedene Reviewarten und werkzeuggestützte statische Analysen, s. [Spillner 12, Kap. 4]).

*Unabhängigkeit zwischen Test und Entwicklung* Unabhängig davon, welche Methoden zum Testen eingesetzt werden, sollen Entwicklung/Programmierung und Test organisatorisch möglichst getrennt bzw. unabhängig voneinander ablaufen. 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 von Tests, Werkzeuge zur Testspezifikation, zum statischen und dynamischen Test und für nicht funktionale Tests (s. [Spillner 12, Kap. 7]).

*Testmanagement* 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 Wirt-

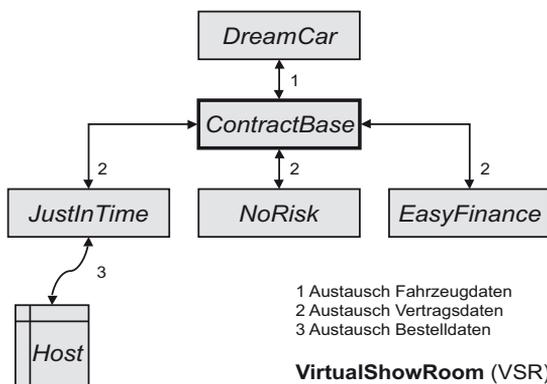
schaftlichkeit des Testens (s. [Spillner 12, Kap. 6]). Das vorliegende Buch vertieft diese Aufgaben des Testmanagements.

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

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 1–2 zeigt eine schematische Darstellung des Systems.



### Fallbeispiel

#### »VirtualShow-Room« – VSR

### Abb. 1–2

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 →Testentwurfsv Verfahren 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 aufeinanderfolgenden 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 der 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. Die Marginalien zitieren die im Übersichtsdokument zu den Lehrplänen des Certified Tester – Advanced Level angegebenen Punkte zum geschäftlichen Nutzen von Testmanagern:

*Ein erfolgreicher  
Testmanager kann  
[CTAL 12] ...*

*... ein Testprojekt leiten und die für die Testorganisation festgelegten Aufgaben, Ziele und Testprozesse umsetzen;*

- In Kapitel 2 wird der grundlegende Testprozess erörtert. Die wesentlichen Aktivitäten des Testmanagements im Testprozess werden ausführlich beschrieben. Das Kapitel geht insbesondere näher auf die Testplanung ein, eine wichtige, wenn nicht sogar die wichtigste Aufgabe des Testmanagers. Die Planung muss während des Projekts angepasst werden.
- 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.

*... Risikoidentifizierung und -analysesitzungen organisieren und leiten, und deren Ergebnisse für die Planung, Aufwandsschätzung, Überwachung und Steuerung der Testaktivitäten verwenden;*

- Identifikation und Analyse der Risiken sowie risikoorientierte Tests sind für das Testmanagement wichtige Instrumente zur Verteilung der beschränkten Testkapazitäten und dienen zur risikomindernden Steuerung des →Testprojekts. In Kapitel 4 sind ent-

sprechende Hinweise zum Vorgehen sowie zu anderen Testansätzen zu finden.

- Kapitel 5 erläutert das grundlegende Vorgehen sowie einige Techniken zur Aufwandsschätzung, die die zielgenaue Zeit- und Ressourcenplanung unterstützen.

*... Testkonzepte erstellen und umsetzen, die der Richtlinie und der Teststrategie der Organisation entsprechen;*

- Dokumente sind ein zentraler Bestandteil des Testprozesses. Planung und Status der Tests werden in zentralen Dokumenten festgehalten und aktualisiert. Kapitel 6 stellt einen Überblick über Arten und Zusammenhänge der wichtigsten Testdokumente dar und erläutert die für das Testmanagement relevanten Dokumente im Detail.

*... die Testaktivitäten zur Erreichung der Projektziele kontinuierlich überwachen und steuern;*

- Testmetriken erlauben quantitative Aussagen bezüglich der Produktqualität, des aktuellen Projektstands und der Reife des Entwicklungs- und Testprozesses und helfen, Kriterien für die Beendigung des Testens festzulegen. Maßtheoretische Grundlagen und konkrete Beispiele hierfür werden in Kapitel 7 gegeben.

*... den relevanten Teststatus bewerten und den Projektbeteiligten zeitgerecht darüber berichten;*

- Die Steuerung des Testprozesses auf Grundlage der Messwerte in den Berichten über den Testfortschritt ist für den Testmanager eine entscheidende Maßnahme, um den Testprozess erfolgreich durchführen zu können. Kapitel 8 geht auf diesen Aspekt ein.

*... wirtschaftliche Argumente für Testaktivitäten vorbringen und darlegen, welche Kosten und Nutzen zu erwarten sind;*

- Da Testen bei vielen Stakeholdern nur mit Kosten assoziiert wird, zeigt Kapitel 9 den Mehrwert des Testens auf, der aus dem investierten Testaufwand gezogen werden kann.

*... die angemessene Kommunikation zwischen den Mitgliedern des Testteams untereinander sowie zwischen Testteam und anderen Projektbeteiligten sicherstellen;*

- Die Einbindung des Testteams in die Aufbauorganisation des Unternehmens – von einzelnen Testern im Projekt bis hin zum verteilten Testen – sowie die damit verknüpften Koordinations- und Kommunikationsaufgaben sind Gegenstand von Kapitel 10.
- In Kapitel 11 werden für das Testmanagement relevante Normen und Standards vorgestellt und diskutiert.

- Reviews zur Qualitätssicherung von Dokumenten werden in vielen Unternehmen mit sehr gutem Erfolg angewendet. Die unterschiedlichen Vorgehensweisen werden ausführlich in Kapitel 12 beschrieben.
- Wie ist mit den beim Testen gefundenen Abweichungen und Fehlerwirkungen umzugehen? Antworten hierzu gibt Kapitel 13.

*... sich an Initiativen zur Testprozessverbesserung beteiligen und diese leiten;*

- Auch der Entwicklungs- und Testprozess selbst kann und soll regelmäßig bewertet und verbessert werden. Welche Verfahren und Vorgehensweisen dazu anzuwenden sind, wird in Kapitel 14 beschrieben.
- Mit entsprechender Werkzeugunterstützung lässt sich der Testprozess meist effizienter durchführen. Welche Werkzeugtypen generell sinnvoll im Testprozess eingesetzt werden können und wie der Testmanager passende Werkzeuge auswählt und einführt, wird in Kapitel 15 beschrieben.

*... Qualifikationen und unzureichende Ressourcen im Testteam identifizieren und bei der Beschaffung angemessener Ressourcen mitwirken und die für das Testteam benötigte Entwicklung von Qualifikationen identifizieren und planen;*

- Ohne Mitarbeiter mit den erforderlichen Fähigkeiten und Qualifikationen – ohne Berücksichtigung des »Faktors Mensch« – kann der Testmanager die Testaufgaben nicht erfolgreich durchführen. In Kapitel 16 wird beschrieben, was bei der Zusammenstellung des Testteams zu berücksichtigen ist.

Selbstverständlich muss das Buch nicht in dieser linearen Reihenfolge gelesen werden, sondern kann auch punktuell als Nachschlagewerk oder anhand der Querverweise als Hypertext »explorative« Verwendung finden.

Eine weitere Reihenfolge, in der die Kapitel gelesen werden können, ergibt sich aus der folgenden, oft in der Praxis zu beobachtenden Handlungskette:

- Worum geht es im Kern? Um den fundamentalen Testprozess (Kap. 2).
- Wie sehen die Randbedingungen dazu aus? Der Kontext des Testmanagements (Kap. 3).
- Was bringt das Testen denn? (Eine sehr oft gestellte Frage, mit der Testmanager oft konfrontiert werden!) Der Mehrwert des Testens (Kap. 9).
- Wie organisiere ich als Testmanager das Testen? Testorganisation (Kap. 10).

- ... und welche Leute brauche ich dazu? Kompetenzen und Teamzusammensetzung (Kap. 16).
- Natürlich mache ich mir Gedanken zur Dokumentation, bevor es mit dem Test losgeht! Testdokumentation (Kap. 6).
- ... und selbstverständlich müssen jegliche Dokumentation und alle anderen Ergebnisse geprüft werden. Reviews, Audits, Assessments (Kap. 12).
- ... und wie aufwendig das Testen ist, muss vorab klar sein. Testaufwandsschätzung (Kap. 5).
- Na, dann kann es ja losgehen mit dem Test. Risikoorientierte und andere Testverfahren (Kap. 4).
- ... und natürlich finden wir Fehler! Fehlermanagement (Kap. 13).
- Wie können Testmanager managen? Testmetriken definieren (Kap. 7) und Testmetriken anwenden (Kap. 8).
- Worauf müssen Testmanager noch achten? Normen und Standards (Kap. 11).
- Was kann langfristig verbessert werden? Bewertung und Verbesserung des Testprozesses (Kap. 14).
- Und wie sieht es mit Werkzeugen aus? Werkzeuge zur Unterstützung des Testprozesses (Kap. 15).

Das Glossar enthält alle hier im Buch verwendeten Begriffe. Weitere Glossareinträge aus dem Buch »Basiswissen Softwaretest« (s. [Spillner 12]) finden Sie auch unter [URL: Glossar GTB] oder als mehrsprachiges Glossar unter [URL: Glossar imbus].