

# 1 Einleitung

*Dieses Kapitel erläutert den Hintergrund, vor dem das Buch geschrieben wurde. Hauptauslöser waren der »Hype« um das Schlagwort »modellbasierter Test« sowie die Vielzahl sehr heterogener existierender Definitionen. Diese wurden mittlerweile auf einen pragmatischen gemeinsamen Nenner gebracht. Darauf aufbauend werden die Ziele des Buches formuliert und der sich daraus ergebende Aufbau sowie der Inhalt der folgenden Kapitel skizziert.*

## 1.1 Model Based Testing – nur ein Hype?

Seit nunmehr über 45 Jahren ist die »Softwarekrise« in mehr oder weniger spürbarer Form immer weiter in den Alltag vorgedrungen. Es vergeht kaum ein Tag, an dem nicht über »Software- oder IT-Fehler« berichtet wird. Vor über 35 Jahren erschien unter dem Titel »The Art of Software Testing« die erste Monografie zum Thema Softwaretest [Myers 1979] die seitdem zigfach fast unverändert nachgedruckt wurde. In jüngerer Zeit ist die Literatur zum Softwaretest förmlich explodiert, und mittlerweile wird bei der überwiegenden Mehrzahl von Softwareprojekten nicht mehr gefragt, ob getestet wird, sondern nur noch was, wie und wie viel. Zur Effizienzsteigerung wird dabei insbesondere auf die automatisierte Durchführung der Tests gesetzt.

Modellbasiertes Testen zielt in erster Linie darauf ab, die Automatisierung auf bislang weniger zugängliche Tätigkeiten in früheren Phasen des Testens auszudehnen. Dort werden nach wie vor aus oft umgangssprachlich formulierten Anforderungen und Systemspezifikationen unter Benutzung mehr oder weniger methodischer Verfahren Testfälle manuell erstellt. Das modellbasierte Testen konzentriert sich auf die Erstellung eines oder mehrerer formaler Modelle, anhand derer z.B. die Testfälle nach vorgegebenen Testkriterien automatisch generiert (und auch ausgeführt) werden können.

*Über 35 Jahre »The Art of Software Testing«*

*Was ist modellbasiertes Testen?*

*Viel testen = gut testen?*

Nun bedeutet »viel testen« bzw. »Testen auf Knopfdruck« im Sinne von vielen automatisch generierten Testfällen nicht automatisch auch »gut testen«. Was aber ist »gutes Testen«? Und wie erreicht man es mit modellbasiertem Testen? Gutes Testen soll einerseits Fehler finden, bevor sie der Kunde findet. Andererseits soll gutes Testen das Vertrauen darin erhöhen, dass das Produkt seinen Einsatzzweck wirklich erfüllt. Für gutes Testen muss also einerseits bekannt sein, wo Fehler am wahrscheinlichsten zu finden sind, und andererseits müssen der Einsatzzweck und die Risiken des Produkts definiert sein.

*Effektiv und effizient  
Testen!*

Die Fragen nach der Effektivität, also den »richtigen« bzw. wichtigen Tests, sowie der Effizienz, also der »richtigen« bzw. Ressourcen sparenden Erstellung und Durchführung dieser Tests, rücken dabei immer mehr in den Vordergrund. Naturgemäß fokussiert das Management eher auf die Effizienz und damit auf die Automatisierung sich oft wiederholender und schematisch ausführbarer Tätigkeiten. Vor diesem Hintergrund ist der teilweise mit völlig überzogenen Erwartungen an die zu erzielende Produktqualität einhergehende »Hype« der automatischen Testausführungswerkzeuge in den späten 90er-Jahren zu verstehen.

*MBT in der Literatur*

Mark Utting und Bruno Legearde setzten in einem der ersten Bücher zu MBT [Utting 2007] den Fokus auf die generativen Aspekte des modellbasierten Testens. Der Titel ihres Buches »Practical model-based Testing« legt die Intention der beiden Autoren nahe, das modellbasierte Testen im praktischen, also vor allem im industriellen Einsatz zu beleuchten. Aber wie reif war das modellbasierte Testen für solche Praxiseinsätze tatsächlich?

Sehr interessante und umfassende Informationen hierzu finden sich in einer der ersten Literaturstudien zu MBT [Dias-Neto 2009]. Die Autoren dieser Studie führten eine systematische Klassifikation von insgesamt 406 bis Mitte 2007 erschienenen Publikationen zum modellbasierten Testen durch. 202 davon erwiesen sich als »gehaltvoll« (d.h., es handelte sich nicht um irrelevante oder duplizierte Informationen). Anschließend wurden 85 Publikationen ausgewählt, die sich entweder auf die Modellierung mit der UML oder auf andere aktuelle (d.h. ab 2004 vorgestellte) oder offenkundig interessante (nämlich mindestens in drei anderen Publikationen referenzierte) Modellierungstechniken bezogen. Diese 85 Veröffentlichungen wurden mit folgendem Ergebnis bezüglich ihrer Praxisrelevanz beurteilt:

Praxisbezug	Anzahl gesichteter Veröffentlichungen			
	UML-basiert	Ohne UML	Gesamt	Anteil in %
<b>Spekulation</b> – Beschreibung des Ansatzes ohne Beispiel oder Referenz auf einen realen Einsatz	17	6	23	27
<b>Beispiel</b> – Erläuterung des Ansatzes mit einem Beispiel, aber ohne Beurteilung der Effektivität und Effizienz	22	16	38	45
<b>Machbarkeitsstudie</b> – Bericht über Anwendung des Ansatzes auf ein »Spieldsystem« mit Nachweis der grundsätzlichen Anwendbarkeit	5	8	13	15
<b>Anwendungsbericht</b> – Bericht über realen Einsatz in einem Industrieprojekt und subjektive Bewertung des Erfolgs	0	4	4	5
<b>Evaluierung</b> – Fallstudie in kontrollierter Praxisumgebung mit quantitativer Analyse von Effektivität und Effizienz	3	4	7	8

**Tab. 1-1**

Ergebnisse der Studie zum modellbasierten Testen [Dias-Neto 2009]

Die beiden letzten Kategorien der Berichte über praktische Anwendungen umfassen lediglich 11 Artikel, also nur 13 % aller vorgestellten Ansätze.

Ist das modellbasierte Testen auch heute nur ein »Hype« oder ist es mittlerweile eine ernst zu nehmende Technologie? Die Analysten und Berater der Gartner Group [URL: Gartner] veröffentlichen jährlich Berichte zu verschiedenen technologiebezogenen Themen, die sogenannten »Hype Cycles« [URL: WikipediaHypeCycle]. Im Hype Cycle des Jahres 2007 zum Thema »Application Development« fand sich das modellbasierte Testen unter dem Begriff »Scriptless Testing« im Bereich des »peak of inflated expectations«, also auf dem »Gipfel der überzogenen Erwartungen«.

Im Jahr 2009 ordnete Gartner das »Scriptless Testing« sogar noch weiter vorne im Hype Cycle an. 2011 war es dann wieder auf dem Gipfel der (überzogenen) Erwartungen, 2011 und 2012 fand es sich auf dem Weg in das Tal der Desillusionierung. Ab 2013 taucht »Scriptless Testing« dann gar nicht mehr im Hype Cycle auf. Das lässt nach der Systematik von Gartner zwei unterschiedliche Schlussfolgerungen zu: Entweder ist die Technologie in Vergessenheit geraten oder sie ist auf dem Weg zur produktiven Nutzung.

Erst wenige Veröffentlichungen über praktische Anwendungen von MBT

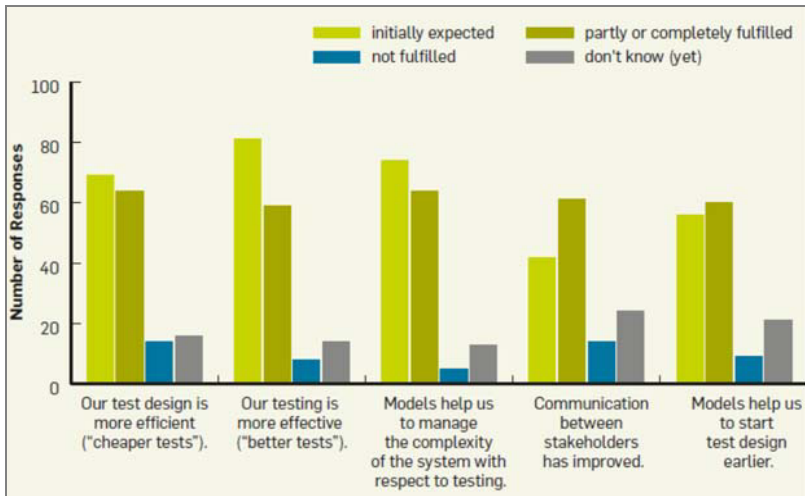
MBT – jahrelang eine Hype-Technologie

Letzteres sehen Mlynarski et al. in ihrem Übersichtsbeitrag zu MBT und folgern:

*Model-based testing has been and is applied in many industry projects. [...] there exist a strong literature and empirical body of knowledge. Based on our industry experience, an even wider application of MBT in the industry will be the case. The methods and tools used for MBT gain a great momentum, which makes this technique promising for many companies. [Mlynarski 2012]*

**Abb. 1-1**

Erfüllte Erwartungen in der MBT-Umfrage 2015  
(aus: [Binder 2015])



MBT heute

Im Jahr 2014 wollten es Robert Binder et al. genau wissen und führten erneut eine groß angelegte Umfrage zu MBT durch [Binder 2015]. Über hundert MBT-Experten aus aller Welt gaben Auskunft. Binder et al. fanden heraus, dass MBT durchaus die Effektivität und Effizienz des Testens steigert, aber die teilweise überzogenen Erwartungen der Anwender immer noch nicht ganz erfüllen kann (s. Abb. 1-1). Die größten Nutzen werden bei der Testüberdeckung, der Komplexitätsbeherrschung, der automatischen Testfallgenerierung und der Wiederverwendung von Modellen und Modellelementen gesehen, die höchsten Hürden bei der Werkzeugunterstützung, den erforderlichen Kompetenzen und dem Widerstand gegen Veränderungen.

Interessante Einzelaussagen sind, dass MBT-Modelle teilweise lediglich zur Dokumentation genutzt werden, vorhandene Werkzeuge als nutzungsunfreundlich gelten, MBT in einem globalen Unternehmen keine Akzeptanz findet, in einem anderen Unternehmen aber als die einzige praktikable Möglichkeit zum Zuverlässigkeits-Assessment angesehen wird ([URL: MBTSurvey1], [URL: MBTSurvey2], [Binder 2015]).

Welche Erwartungen darf man bei diesen Ergebnissen also tatsächlich bezüglich Effektivität und Effizienz an modellbasierte Vorgehensweisen und Testwerkzeuge haben und wo sind die Grenzen? Wie motiviert man den Einsatz und wie trifft man eine fundierte Auswahl?

Bevor dieses Buch Antworten auf diese und weitere Fragen geben kann, gilt es zunächst zu klären, was genau unter dem Begriff »modellbasierter Test« zu verstehen ist und auf welche Arten von Modellen er sich bezieht.

## 1.2 Viele Definitionen für modellbasiertes Testen

Bereits 1990 gab Boris Beizer folgende Charakterisierung des Softwaretests:

*Testing is a process in which we create mental models of the environment, the program, human nature, and the tests themselves. [...] The art of testing consists of creating, selecting, exploring, and revising models.*

[Beizer 1990]

Ausgehend von dieser allgemeinen Definition hat der Testspezialist Robert Binder wohl recht, wenn er fast 10 Jahre später schreibt, Testen wäre immer modellbasiert [Binder 1999]. Tester nutzen nämlich seit jeher vereinfachende Bilder der zu testenden Systeme, um Testfälle und Testszenarien zu definieren. Oftmals sind diese Bilder aber nur als mentale Modelle implizit in ihren Köpfen vorhanden.

*Testen ist immer modellbasiert!*

Ein wesentliches Ziel von MBT liegt darin, diese Modelle explizit und formal (also eindeutig interpretierbar und maschinenlesbar) zu machen. Dadurch werden komplexe Sachverhalte verständlich, eindeutig und nutzbar für die Automatisierung von Testaktivitäten. Robert Binder folgert:

*Ziel von MBT: informelle (mentale) Modelle explizit und formal machen*

*Test design must be based on both general and specific models.*

[Binder 1999]

Im Eintrag »Model-Based Software Testing« der »Encyclopedia of Software Engineering« geben Whittacker und El-Far folgende allgemeine Definition für MBT:

*MBT-Definitionen in der Literatur*

*Model-Based Testing (MBT) is [...] an approach that bases common testing tasks such as test case generation and test result evaluation on a model of the application under test.*

[Marciniak 2002]

Die allgegenwärtige Wikipedia enthält folgende Erläuterung:

*Modellbasiertes Testen (MBT) ist ein Oberbegriff für die Nutzung von Modellen zur Automatisierung von Testaktivitäten [bzw. zur] Generierung von Testartefakten im Testprozess. Darunter fällt insbesondere die Generierung von Testfällen aus Modellen (z.B. unter Verwendung der UML), die das Sollverhalten des zu testenden Systems beschreiben.*

[URL: Wikipedia]

Dort finden sich auch Erläuterungen für die ähnlich gelagerten Begriffe »modellzentriertes Testen« und »modellgetriebene Testentwicklung«.

In der oben skizzierten Literaturstudie zu MBT wird modellbasiertes Testen wie folgt abgegrenzt:

*MBT approaches help automatically generate test cases using models extracted from software artifacts.*

[Dias-Neto 2009]

Conrad interpretiert den Begriff dagegen in Richtung des Testens von Modellen und verknüpft ihn eng mit der modellbasierten Entwicklung:

*Im Hinblick auf den Softwaretest eröffnet die Modellbasierte Entwicklung [...] die Möglichkeit, schon das ausführbare Modell des Systems einer analytischen Qualitätssicherung zu unterziehen (Modelltest) [...] Ein derartiger den Modellbasierten Entwicklungsprozess begleitender Test wird als Modellbasierter Test bezeichnet.*

[Conrad 2004]

In einem Artikel von Holger Schlingloff liest sich das so:

*Beim modellbasierten Testen geht es dabei sowohl um den Test von Dokumenten einer modellbasierten Entwicklung wie zum Beispiel Systemmodellen, Implementierungsmodellen oder generiertem Code als auch um die Ableitung von Testsuiten aus Modellen (Testfallgenerierung).*

[Schlingloff 2007]

Mark Utting und Bruno Legeard legen in ihrem oben bereits erwähnten Buch »Practical Model-Based Testing« den Fokus auf die generativen Aspekte des modellbasierten Testens und schreiben:

*The following are the four main approaches known as model-based testing:*

1. *Generation of test input data from a domain model.*
2. *Generation of test cases from an environment model.*
3. *Generation of test cases with oracles from a behavior model.*
4. *Generation of test scripts from Abstract tests.*

[Utting 2007]

Die Erfinder des UML2 Testing Profile (UTP, s. Abschnitt 9.3.1) fügen diesen Definitionen als weitere Fassetten die formale Beschreibung bzw. Modellierung der Tests selbst zu:

*Model-based testing requires the systematic and possibly automatic derivation of tests from models. In our case, UML is the language for specifying models and UTP the formalism to describe the derived tests.*

[Baker 2008]

Im Juli 2011 erschien mit der ETSI-Spezifikation ES 202 951 [ETSI 2011] erstmals eine international gültige Spezifikation u.a. zum Thema MBT, die MBT sehr allgemein definiert als

*An umbrella of approaches that generate tests from models.*

[ETSI 2011]

Welche grundsätzlichen gemeinsamen Merkmale von MBT lassen sich nun in all diesen Definitionen bzw. den sie enthaltenden Publikationen finden?

Beim modellbasierten Testen wird die zu testende Software, ihre Umgebung oder der Test selbst über Modelle betrachtet, die sich auf besondere Eigenschaften des Testobjekts bzw. der Tests konzentrieren, meist auf deren Verhalten. Diese Modelle können eigenständig, parallel zu Entwicklungsmodellen erstellt oder aber aus diesen abgeleitet werden. Wie alle softwaretechnischen Artefakte sind sie Gegenstand der Qualitätssicherung. Aus den Modellen werden – mit teilweise seit vielen Jahren bekannten Verfahren – u.a. Testfälle automatisch generiert, hinsichtlich der Testdaten sowie technologisch konkretisiert und über entsprechende Testrahmen oder -adapter ausgeführt.

*Merkmale von MBT*

Diese Erkenntnisse führen zu einer vorläufigen pragmatischen Definition von MBT, die in Kapitel 4 im Detail betrachtet wird:

**Modellbasiertes Testen** umfasst mindestens einen der beiden folgenden Aspekte:

- Tests modellieren
- Tests aus Modellen generieren

*(Vorläufige)*

**Definition**

*von MBT*

Nach diesen grundlegenden Begriffsklärungen können nun die Ziele des Buches formuliert werden.

### 1.3 Ziele des Buches

Das globale Ziel des Buches ist die Beantwortung der Frage, ob sich der Einsatz von MBT in Ihrem Hause lohnt und wenn ja, welche der vielen in der Literatur vorgeschlagenen Methoden und Techniken sowie welche der verfügbaren MBT-Werkzeuge die passenden sein könnten. Hieraus ergeben sich die folgenden Teilziele.

#### **Rahmen und Definitionen zu MBT schaffen**

Wichtigstes Ziel des Buches ist – mit Blick auf die oben aufgeführten und durchaus unterschiedlichen Definitionen modellbasierten Testens –, einen belastbaren und möglichst allgemeingültigen Rahmen für das Thema zu schaffen. Dazu gehört zwingend, einige Definitionen einzuführen. Diese werden mit einigen der bereits existierenden Quellen teilweise im Widerspruch stehen oder Unterschiede aufweisen.

Da das Thema MBT selbst noch vergleichsweise jung ist, lässt sich das nicht vermeiden – es existieren eben noch unterschiedliche Ansichten darüber, was unter »modellbasiertem Testen«, »modellgetriebenem Testen« oder aber »modellzentrischem Testen« zu verstehen ist. Wir haben uns im Zuge der Buchentstehung auf ein Begriffsmodell und sogar auf ein MBT-Reifegradmodell geeinigt, das wir für tragfähig halten. Hier mögen Sie selbst entscheiden, ob Sie einige der hier getroffenen Festlegungen für ungeeignet halten – in solchen Fällen würden wir uns sehr freuen, wenn Sie uns davon in Kenntnis setzen.

#### **Rüstzeug für weiterführende Fachliteratur vermitteln**

Der so geschaffene Rahmen soll Sie insbesondere in die Lage versetzen, die stetig anwachsende Fachliteratur zum Thema MBT (siehe z.B. das Literaturverzeichnis für eine aktuelle Auswahl der wichtigsten Beiträge) einerseits zu verstehen, andererseits einordnen und bewerten zu können. Immer noch ist zuweilen die eine oder andere unreife oder vielleicht auch vom Marketing getriebene Aussage anzutreffen. Das hier vermittelte Wissen soll es ermöglichen, die Spreu vom Weizen zu trennen.

#### **MBT in der Breite, aber nicht immer in der Tiefe präsentieren**

Trotz allem trägt das Buch nicht umsonst den Titel »Basiswissen«. Es hat ausdrücklich nicht das Ziel, jedes hier genannte Thema und jeden möglichen Übergang von Modellen zu Testartefakten in gleicher Tiefe und Gründlichkeit zu präsentieren. Dies liegt u.a. daran, dass die UML (Unified Modeling Language) der OMG in der Welt der vielen in der IT verwendeten Modellierungstechniken und -sprachen unbestreitbar eine führende Rolle einnimmt.



Folglich steht der Übergang von UML-Modellen hin zu Testfällen deutlich im Vordergrund. Es würde den Rahmen eines »Basiswissen«-Buches sprengen, in gleichem Umfang etwa BPMN, EPK-, SA/SD-Modelle oder Petri-Netze sowie die Techniken zum Model Checking im Hinblick auf modellbasierte Testmethoden zu beleuchten. Hier verweisen wir jeweils auf weiterführende Fachliteratur.

### **Unterschiede zum konventionellen Testprozess klären**

Aus mehreren guten Gründen setzen wir in unserer Darstellung auf den Lehrplänen des ISTQB® und damit auch auf den in derselben Buchreihe wie das vorliegende Buch erschienenen Begleitbüchern zum Certified-Tester-Ausbildungsschema auf. Darin wird der sogenannte fundamentale Testprozess präsentiert, der aus insgesamt fünf Phasen mit zugehörigen Rollen und typischen (Test-)Artefakten besteht.

Eine der Kernfragen, denen sich dieses Buch widmet, lautet, ob sich dieser Testprozess beim modellbasierten Testen verändert und wenn ja, wie ein MBT-Testprozess im Vergleich dazu aussieht – wozu dann auch entsprechende Rollen und Artefakte gehören.

### **Unrealistische Erwartungen an MBT beseitigen**

MDA- und MDSD-Ansätze (*Model-Driven Architecture, Model-Driven Software Development*) haben in den letzten Jahren das Bild gefestigt, dass es unter geeigneten Voraussetzungen möglich ist, aus Modellen Code oder wenigstens Codeteile zu generieren und damit einen Effizienz- und Qualitätsgewinn zu erzielen. Vor diesem Hintergrund verwundert es nicht, wenn regelmäßig die Frage gestellt wird: »Wenn wir aus unseren Modellen so gut Code generieren können – können wir dann nicht auch gleich noch unsere Tests aus ihnen generieren?«

Das Buch wird klarmachen, dass dies zu kurz gedacht ist. Nicht jedes Modell eignet sich gleich gut für Generierungen in Richtung Test, und die Formulierung »Tests generieren« ist zu undifferenziert: Was genau wird generiert? Testdaten? Testfälle – abstrakte oder konkrete? Testskripte? Testergebnisse? Testrahmen? Insbesondere wird klar werden, dass es »das MBT«, d.h. modellbasiertes Testen »von der Stange«, nicht gibt.

### **Nutzen und Potenziale von MBT identifizieren**

Stattdessen ist es Aufgabe des Buches, deutlich herauszustellen, was MBT leisten kann und was dazu benötigt wird. Es wird sich zeigen, dass MBT eine sinnvolle Ergänzung zu den bekannten »konventionellen« Testansätzen darstellt. Die konkrete Ausgestaltung und Einführung sind aber in jedem Projekt individuell vorzunehmen. Insbeson-

dere soll deutlich werden, in welchen Situationen MBT helfen kann, Probleme im Testvorgehen zu lösen, und für welche Probleme dies möglich ist.

### **Eignungsprüfung für eigene Testprojekte ermöglichen**

Ebenfalls wird klar werden, dass modellbasiertes Testen – wie jede andere Test- bzw. Qualitätssicherungsmaßnahme auch – mit Aufwand und somit mit Investitionen verbunden ist. Außerdem existiert eine Reihe von Faktoren, die für einen erfolgreichen MBT-Einsatz erfüllt sein müssen. Beide Punkte müssen im Vorfeld geprüft werden, wenn man sich über den Einsatz vom MBT im eigenen Projekt Gedanken macht: Sind die Rahmenbedingungen dafür gegeben? Und ist auf einen messbaren Mehrwert bzw. ROI (Return on Investment) zu hoffen, wenn MBT im Projekt eingeführt wird?

Auch zu diesen beiden wichtigen Fragen will das Buch Hilfestellung und Hinweise geben – bis hin zu einem Fragenkatalog zur Klärung der Voraussetzungen für den erfolgreichen MBT-Einsatz.

### **Bei der MBT-Werkzeugauswahl unterstützen**

Prozesse und Werkzeuge gehen Hand in Hand, und so verwundert es nicht, dass modellbasiertes Testen teilweise auch den Einsatz neuer bzw. neuartiger Testwerkzeuge mit sich bringt. Auch hier gilt, dass es einerseits große Unterschiede zwischen den aktuell vorhandenen Werkzeugen am Markt gibt und dass andererseits nur mit der Anschaffung eines Werkzeugs nicht bereits alles getan ist. Das Buch führt verschiedene MBT-Werkzeugkategorien ein, stellt einige Vertreter kurz vor und hilft bei der Auswahl passend zur gewählten MBT-Strategie.

### **An Beispielen veranschaulichen**

Wo möglich, werden praxisnahe Beispiele herangezogen, um Aussagen zu MBT zu illustrieren. Dabei haben wir versucht, mit zwei durchgängigen, leicht nachvollziehbaren Beispielen möglichst viele Aspekte von MBT zu erläutern.

## 1.4 Aufbau des Buches

Nach dieser kurzen Einführung in das modellbasierte Testen möchten wir Ihnen noch einen Überblick darüber geben, was Sie in den folgenden Kapiteln erwartet.

Teil I (MBT – Einstieg und Grundlagen) ist optional je nach Vorkenntnissen zu überspringen und besteht aus dieser Einleitung sowie den Kapiteln 2 bis 5. Nach Lektüre dieses Teils sollten Sie in der Lage sein, einer Diskussion, einem Fachartikel oder -buch zu MBT (und insbesondere den restlichen Kapiteln dieses Buches) folgen zu können.

- Kapitel 2 erläutert in knapper Form die wichtigsten in diesem Buch verwendeten Grundbegriffe des Testens und den fundamentalen Testprozess nach ISTQB®. Am Ende des Kapitels werden grundlegende Herausforderungen genannt, denen sich das Testen heute gegenüber sieht und die durch moderne Technologien wie MBT adressiert werden können.
- Kapitel 3 führt Grundlagen der Modellierung ein. Der Schwerpunkt liegt hierbei auf der Syntax und Semantik der UML.
- In Kapitel 4 wird die Definition für »modellbasiertes Testen« erläutert, die wir dem Buch zugrunde gelegt haben. Davon abgeleitet ist eine Darstellung der wesentlichen Modellkategorien und Einsatzszenarien für MBT.
- Kapitel 5 ist ein frühes Intermezzo zwischen den methodischen Teilen des Buches. Es detailliert anhand des Fallbeispiels zum modellbasierten Test einer elektronischen Türsteuerung ein typisches Einsatzszenario von MBT.

Teil II (MBT im Testprozess), ist das »Herz« dieses Buches, besteht aus den Kapiteln 6 bis 11 und stellt MBT als Prozess schrittweise dar.

- In Kapitel 6 wird zunächst der Rahmen gebildet, indem erläutert wird, wie der Testprozess unter Einbeziehung von MBT zu gestalten ist. Des Weiteren stellt das Kapitel dar, wie dieser Prozess für unterschiedliche Teststufen anzupassen ist und wie er mit unterschiedlichen Vorgehensmodellen zur Softwareentwicklung zusammenwirkt. Die Betrachtungen dieses Kapitels sind unabhängig von der konkreten Modellierungsmethode.
- Kapitel 7 konkretisiert die Aktivitäten zur Planung und Steuerung des Testprozesses mit Hinblick auf die Nutzung von MBT. Dabei wird zwar eine iterativ-inkrementelle Vorgehensweise angenommen, aber – wie schon im vorangegangenen Kapitel – keine spezifische Modellierungsmethode vorausgesetzt.
- Kapitel 8 ist als weiteres Intermezzo und als Klammer für die beiden darauf folgenden Kapitel zu betrachten. Anhand der hypothe-

tischen Anwendung »CarKonfigurator«, die als Testbeispiel weithin bekannt ist, werden erste Zusammenhänge zwischen Testmodellen und Generierungsverfahren vermittelt.

- Kapitel 9 greift dieses Beispiel auf und beschreibt die Testanalyse und den Testentwurf auf Basis von MBT. Es diskutiert einige industriell einsetzbare Modellierungsmethoden auf der Basis von System- und Testmodellen und deren Zusammenspiel mit Testfallgeneratoren. Darüber hinaus erläutert es einige »Fallstricke« der Modellierung sowie die wesentlichen Maßnahmen zur Qualitätssicherung von Modellen.
- Im Fokus des Kapitels 10 steht die Testrealisierung aus Sicht von MBT. Es wird erläutert, wie Testfallgeneratoren funktionieren und was das Ergebnis des Generiervorgangs ist. Generierungsalgorithmen, Abdeckungskriterien und Methoden zur Steuerung von Generatoren werden diskutiert.
- Kapitel 11 schließt die Betrachtung der Phasen des fundamentalen Testprozesses ab und beleuchtet die Testdurchführung und die Bewertung des Tests. Dabei wird die Brücke sowohl zur manuellen als auch zur automatisierten Durchführung generierter Testfälle geschlagen. Die Bewertung der Endkriterien und die Berichterstattung der Testergebnisse auf Basis von Modellinformationen stellen den zweiten Schwerpunkt des Kapitels dar.

In Teil III (MBT im Praxiseinsatz) werden aus den Kernaussagen des zweiten Teils weiterführende Themen entwickelt, um den Leserinnen und Lesern dabei zu helfen, MBT im eigenen Projekt einzusetzen. Dieser Teil umfasst die Kapitel 12 bis 15.

- Kapitel 12 stellt eine Systematik zur Kategorisierung von Softwarewerkzeugen im MBT-Umfeld bereit. Ausgehend von diesen Grundlagen können MBT-Werkzeuge evaluiert und auf die Einsetzbarkeit im eigenen Umfeld hin bewertet werden.
- Kapitel 13 widmet sich der Frage, wie MBT in einem konkreten Projekt oder einer Organisation systematisch und erfolgreich eingeführt wird. Die Grundlage hierfür bildet ein Reifegradmodell für modellbasiertes Testen.
- Kapitel 14 gibt dem Leser mittels einer Fallstudie einen Leitfaden an die Hand, mit dem die Wirtschaftlichkeit von MBT bewertet werden kann.
- Kapitel 15 schließlich zieht ein Fazit aus Sicht der Autoren, welches Potenzial MBT heute im industriellen Einsatz erschließen kann. Es werden aber auch die Grenzen der im Buch aufgegriffenen Ansätze zusammengefasst. Zum Abschluss wagen wir einen Blick in die (nähere) Zukunft von MBT.

Jedes Kapitel (mit Ausnahme dieser Einleitung) endet mit einer Zusammenfassung, in der die wichtigsten Punkte kurz und prägnant dargestellt werden. Einsteigern in die Thematik MBT, die sich nur einen Erstüberblick verschaffen wollen und die kein entsprechendes Training absolvieren möchten, empfehlen wir mit den Kapiteln bzw. Abschnitten 2, 3.1/3.2 sowie 4 zu starten. Die Abschnitte 6.1 und 6.2 würden als prozessuales Grundwissen diesen Erstüberblick bereits schon wieder abschließen. Für den »gestressten Manager« bzw. Teamleiter oder für andere Entscheider sind die Kapitel 13 bzw. 14 zu empfehlen, da darin sowohl die wirtschaftlichen ROI-Faktoren als auch die Faktoren zur erfolgreichen Einführung von MBT behandelt werden. Für diejenigen, die sich gezielt auf die Prüfung zum ISTQB® Certified Model-Based Tester vorbereiten wollen, ist als Direkteinstieg Abschnitt 1.5 erstellt worden. Ansonsten darf man unser Buch auch ganz normal von vorne bis hinten durchlesen.

## 1.5 Leseanleitung für die Qualifikation zum ISTQB® Certified Model-Based Tester

Der Hauptteil des Buches orientiert sich grob an den Phasen bzw. Aktivitäten des ISTQB®-Testprozesses. Dies wurde im Lehrplan zum ISTQB® Certified Model-Based Tester nur teilweise übernommen. Die folgende Leseanleitung dient daher zur zielgerichteten Vorbereitung auf die Zertifizierungsprüfung zum ISTQB® Certified Model-Based Tester, für die wir Ihnen viel Erfolg wünschen!

*MBT – seit 2015 im ISTQB®  
Certified-Tester-Schema*

Einsteigern in die Materie legen wir ein entsprechendes Training bei einem vom GTB e.V. akkreditierten Trainingsunternehmen ans Herz [URL: GTB]. Vielleicht haben Sie dieses Buch ja sogar als Begleitmaterial zu einem solchen Training erhalten! In der untenstehenden Tabelle finden Sie Einstiegspunkte in die Kapitel bzw. Abschnitte des Buches bezüglich der Inhalte des Lehrplans.

Einsteigern, die kein entsprechendes Training absolvieren möchten, empfehlen wir, das Buch zunächst von vorne bis hinten durchzulesen. Studieren Sie danach den Lehrplan, und konsultieren Sie dabei noch einmal vertiefend die Kapitel bzw. Abschnitte des Buches gemäß der Tabelle.

Fortgeschrittene MBT-Anwender können nach einem ersten »Querlesen« des Buches beim Studium des Lehrplans direkt die Kapitel bzw. Abschnitte des Buches gemäß Tabelle 1–2 konsultieren.

Essenzielle Buchkapitel für die Zertifizierungsprüfung

Insgesamt decken folgende Buchkapitel den ISTQB®-Lehrplan ab:

- Kapitel 3, Abschnitte 3.1, 3.2, 3.5, 3.6
- Kapitel 4, Abschnitte 4.3, 4.4
- Kapitel 6, Abschnitte 6.1, 6.2, 6.3, 6.5
- Kapitel 7, Abschnitte 7.2, 7.3, 7.6
- Kapitel 9, Abschnitte 9.2, 9.5, 9.6, 9.7, 9.8, 9.9
- Kapitel 10, Abschnitte 10.2, 10.3, 10.4, 10.5, 10.8
- Kapitel 11, Abschnitte 11.1, 11.2, 11.3
- Kapitel 12
- Kapitel 13, Abschnitte 13.2, 13.3, 13.4, 13.5
- Kapitel 14, Abschnitt 14.1
- Kapitel 15, Abschnitte 15.1, 15.2, 15.3, 15.4
- sowie die beiden MBT-Beispiele in den Kapiteln 5 und 8

**Tab. 1-2**  
Einstiegspunkte gemäß  
ISTQB®-MBT-Lehrplan

Lehrplan		Buch		Seiten
<b>1</b>	<b>Introduction to Model-Based Testing</b>	1	Einleitung	3
1.1	Objectives and Motivations for MBT	2.3	Herausforderungen beim Testen heute	38
1.1.1	Main Motivations for MBT	15.1	Was modellbasiertes Testen leisten kann	403
		15.2	Was modellbasiertes Testen unter Umständen leisten kann	408
1.1.2	Misleading Expectations and Pitfalls of MBT	15.3	Was modellbasiertes Testen nicht leisten kann	410
		15.4	Was modellbasiertes Testen nicht ersetzen kann	411
1.2	MBT Activities and Artifacts in the Fundamental Test Process	6	MBT und der Testprozess	135
1.2.1	MBT Specific Activities	6.1	Ausprägungen von MBT im Testprozess	136
1.2.2	Essential MBT Artifacts (Inputs and Outputs)	6.2	Allgemeine Integration von MBT in den Testprozess	140
		6.3	Abhängigkeit des MBT-Prozesses von den verwendeten Modellkategorien	144
		7.3	Aktivitätenplanung und Aufwandsschätzung	174



Lehrplan		Buch		Seiten
1.3	Integrating MBT into the Software Development Lifecycles	6.5	MBT und der Entwicklungsprozess	159
1.3.1	MBT in Sequential and Iterative Software Development Lifecycles	6.5.2	Betrachtung verschiedener Vorgehensmodelle im Zusammenhang mit MBT	160
1.3.2	Supporting Requirements Engineering	3.2	Allgemeiner Vergleich von Texten und Modellen	50
		4.3	MBT im Projektkontext	107
<b>2</b>	<b>MBT Modeling</b>	3	Einführung in die Modellierung	47
2.1	MBT Modeling			
2.1.1	MBT Modeling Activities	4	MBT – ein Einstieg	105
2.1.2	Subject and Focus of MBT Models	3.1	Wat is'n Modell?	47
		4.4	MBT-Modellkategorien	109
2.1.3	MBT Models Depend on Test Objectives	7.2	Bestimmung der Teststrategie	172
2.2	Languages for MBT Models	3.1	Wat is'n Modell?	47
2.2.1	Main Categories of Modeling Languages for MBT	3.5	Modellierung statischer Strukturen	58
2.2.2	Language Categories Relevant for Different Systems and Project Objectives	3.6	Modellierung dynamischen Verhaltens	69
		4.3	MBT im Projektkontext	107
2.3	Good Practices for MBT Modeling Activities	9	Modellierung und Modellprüfung	213
2.3.1	Quality Characteristics for MBT Models	9.8	Qualitätsmerkmale und Ziele von Modellen	253
2.3.2	Typical Mistakes and Pitfalls in MBT Model Design	9.6	Achtung: Fallstricke!	241
2.3.3	Linking Requirements and Process Related Information to the MBT Model	9.2.7	Verknüpfung von Testmodellen und zu testenden Anforderungen	223
2.3.4	Modeling Guidelines for MBT	9.9.1	Modellierungsrichtlinien als Grundlage	258
2.3.5	Reuse of Existing System Design or Requirements Models	6.3	Abhängigkeit des MBT-Prozesses von den verwendeten Modellkategorien	144
		9.5	Nutzung vorhandener Systemmodelle	240
2.3.6	Tool Support for Modeling Activities	9.8.11	Werkzeugunterstützung	257
		12	Werkzeuge für MBT	343
2.3.7	Iterative Model Development, Review and Validation	9.9	Prüfung von Modellen	258



Lehrplan	Buch	Seiten
<b>3 Selection Criteria for Test Case Generation</b>	10 Testauswahlkriterien und Testfallgenerierung	265
3.1 Classification of MBT Test Selection Criteria	10.2 Einführung in die Generierung	266
3.1.1 Test Selection Criteria	10.3 Generierungsverfahren im Einzelnen	272
3.1.2 Test Case Selection in Practice	10.4 Steuerung der Testfallgenerierung	293
3.1.3 Examples of Test Selection Criteria	10.3 Generierungsverfahren im Einzelnen	272
	5 Intermezzo 1: Türsteuerung	119
	8 Intermezzo 2: CarKonfigurator	197
3.1.4 Relation to Foundation Level Test Design Techniques	8.2.2 Auswahl der Testdaten	200
	10.3.1 Anforderungsbasierte Testgenerierung	272
	10.3.5 Datenüberdeckungskriterien	280
3.2 Applying Test Selection Criteria	6.1 Ausprägungen von MBT im Testprozess	136
3.2.1 Degree of Automation in Test Generation		
3.2.2 Pros and Cons of Specific Test Selection Criteria	10.8 Bewertung der Testauswahlkriterien	308
3.2.3 Good Practices of MBT Test Selection	10.4.1 Verknüpfung von Testauswahlkriterien	293
<b>4 MBT Test Implementation and Execution</b>	11 Realisierung, Durchführung und Auswertung	311
4.1 Specifics of MBT Test Implementation and Execution	9.7 Übergang von Systemmodellen zu Testmodellen	245
4.1.1 Abstract and Concrete Test Cases in the MBT Context	10.2.1 Was wird generiert – abstrakte oder konkrete Testfälle?	266
4.1.2 Different Kinds of Test Execution	11.1 Manuelle Testdurchführung	311
	11.2 Automatisierte Testdurchführung	314
4.1.3 The Impact of Changes on the MBT Artifacts	10.5 Testneugenerierung nach Änderungen	300
4.2 Activities of Test Adaption in MBT	11.2 Automatisierte Testdurchführung	314



Lehrplan	Buch	Seiten
<b>5 Evaluating and Deploying an MBT Approach</b>	13 MBT erfolgreich einführen	353
5.1 Evaluate an MBT Deployment	13.5 MBT und Prozessreife	367
5.1.1 ROI Factors for MBT Introduction	14 Wirtschaftlichkeitsbetrachtung	383
5.1.2 Organizational Objectives and their Relationship to the Characteristics of the MBT Approach	13.5.2 MBTPI – Prozessverbesserung für modellbasiertes Testen	370
5.1.3 Metrics and Key Performance Indicators	11.3 Auswertung und Berichterstattung	327
	10.3 Generierungsverfahren im Einzelnen	272
	13.2.1 Ziele klar definieren und messbar machen	356
	7.6 Steuerung mit Metriken und Testendekriterien	185
5.2 Manage and Monitor the Deployment of an MBT Approach	13 MBT erfolgreich einführen	353
5.2.1 Good Practices when Deploying MBT	13.3 Die Einführung als Projekt planen und durchführen	358
5.2.2 Cost Factors of MBT	14.1 Übersicht über Kostenfaktoren von MBT	383
5.2.3 Integration of the MBT Tool	13.4 Werkzeuge richtig auswählen	363
<b>9 Appendix A – Simple Modeling Language</b>	3.6 Modellierung dynamischen Verhaltens	69
9.1 A Simple Graphical Modeling Language for Workflows	3.6.1 Aktivitätsmodellierung	70
9.2 A Simple Graphical Modeling Language for State Transition Diagrams	3.6.3 Zustandsmodellierung	76

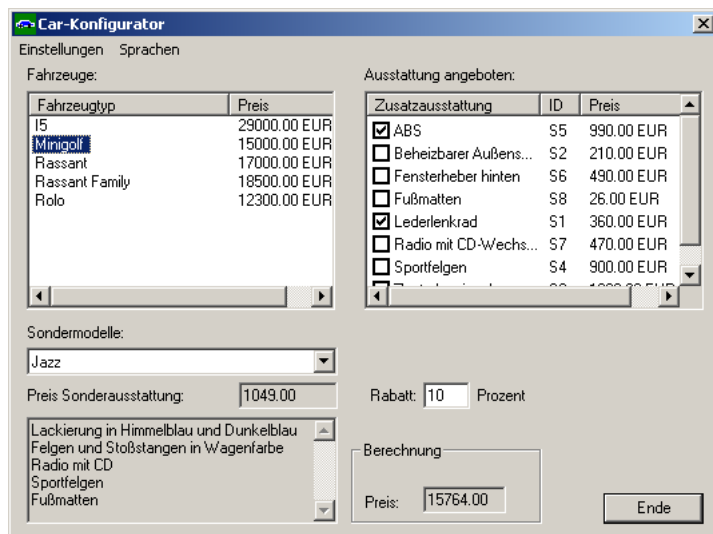
## 1.6 Zwei Fallbeispiele

Die folgenden beiden einfachen hypothetischen Systeme dienen im gesamten Buch als durchgängige Beispiele für Modellierungs- und Testfallgenerierungstechniken.

### 1.6.1 Produktskizze CarKonfigurator

Bei dem ersten Fallbeispiel handelt es sich um den aus [Spillner 2012] bekannten CarKonfigurator, der für ein neues elektronisches Verkaufssystem, genannt VirtualShowRoom (VSR), eines Automobilkonzerns entwickelt wird. Diese kleine Applikation ist eine Art Taschenrechner, mit dem Autohändler bzw. deren Kunden den Preis des Wunschautos berechnen können (s. Abb. 1–2). Die Anwendung soll in der Endausbaustufe weltweit bei allen Händlern eines Automobilkonzerns installiert werden.

**Abb. 1–2**  
Benutzerschnittstelle des  
CarKonfigurators



Folgende Anforderungen an den CarKonfigurator sind relevant:

#### Funktionale Anforderungen

*Funktionale  
Anforderungen*

Das System ermöglicht die interaktive Konfiguration von Fahrzeugen am Bildschirm (Modellauswahl, Farbe, Ausstattung usw.). Nach der Auswahl eines Fahrzeugtyps zeigt das System mögliche Sondermodelle und zusätzlich angebotene Ausstattungsvarianten an. Zu jeder Auswahl ermittelt das System sofort den jeweiligen Listenpreis.

Während der Auswahl von Fahrzeugtyp sowie – optional – Sondermodell, Zusatzausstattungen und Händlerrabatt wird der Gesamtpreis des so konfigurierten Autos immer sofort aktualisiert und angezeigt.

Das System bietet mehrere Sprachen zur Benutzerführung an, aus der eine ausgewählt werden kann. In der ersten Version sollen dies Deutsch und Englisch sein.

### **Benutzbarkeitsanforderungen**

Benutzer des Systems sind Kunden, die entweder unterstützt durch einen Verkäufer oder aber vollkommen selbstständig ihr Wunschfahrzeug am Bildschirm konfigurieren möchten. Daher muss das System intuitiv und ohne jegliche Einarbeitungszeit benutzbar sein.

*Benutzbarkeitsanforderungen*

### **1.6.2 Produktskizze Türsteuerung**

Im zweiten Fallbeispiel geht es um die Steuersoftware für einen elektronischen Schiebetürantrieb, der für den Einsatz an Außentüren öffentlich zugänglicher Gebäude (z.B. Krankenhäuser, Ämter) konzipiert ist. Während im Fallbeispiel CarKonfigurator neben den funktionalen Anforderungen vor allem die Benutzbarkeit als nicht funktionale Anforderung im Mittelpunkt steht, sind bei der Türsteuerung neben der Funktion hauptsächlich Sicherheitsanforderungen zu beachten.

*Ein elektronischer Schiebetürantrieb*

Folgende Anforderungen an die Türsteuerung sind dokumentiert:

### **Funktionale Anforderungen**

Der Antrieb öffnet die Tür bei Annäherung von Personen automatisch. Hierzu wird er mit jeweils einem Annäherungssensor auf der Innen- und der Außenseite der Tür verbunden.

Ob die Schiebetür vollständig geöffnet oder geschlossen ist, wird dem Antrieb über zwei Anschlagssensoren mitgeteilt. Außerdem gibt es einen Einklemmsensor, der die Verletzung von Personen vermeiden soll, die sich während des Schließvorgangs im Türrahmen befinden. Spricht während des Schließvorgangs der Annäherungssensor oder der Einklemmsensor an, so stoppt der Schließvorgang und der Motor beginnt die Tür wieder zu öffnen.

*Funktionale Anforderungen:  
automatisches Öffnen  
und Schließen der Tür*

Ist die Tür geöffnet und nähert sich keine weitere Person, so soll die Tür sich nach einer Wartezeit von 10 Sekunden wieder schließen. Falls während dieser 10 Sekunden eine weitere Person die Tür durchschreitet, beginnt die Wartezeit erneut.

Über eine Netzwerkleitung kann der Antrieb verriegelt werden, wenn sich die Tür im geschlossenen Zustand befindet. Durch diese Leitung wird die zentrale Verriegelung des Gebäudes durch die Nachtwache unterstützt.

### **Sicherheitsanforderungen**

*Sicherheitsanforderungen:  
Verhalten der  
Türsteuerung im  
Gefahrenfall*

Da es sich um die Eingangstür eines öffentlichen Gebäudes handelt, muss die Tür im Gefahrenfall jederzeit manuell von innen entriegelt und dauerhaft geöffnet werden können. In diesem Fall kann eine Person die manuelle Entriegelung betätigen und anschließend die Tür aufschieben; falls zu diesem Zeitpunkt der Antrieb noch funktioniert, soll die Steuerung das Aufschieben durch Anstoßen eines automatischen Öffnungsvorgangs unterstützen und, wenn die Tür vollständig geöffnet ist, den Antrieb bei geöffneter Türposition verriegeln.