

2 Der Testrahmen

In diesem Kapitel werden einige Begriffe erläutert, die mit dem Testen in Zusammenhang stehen. Es wird darauf eingegangen, was Testen eigentlich ist, weshalb es durchgeführt wird und welche Produkte (und Ergebnisse) man von einem Testprozeß erwarten kann. Außerdem wird das Testen im Zusammenhang mit dem Qualitätsmanagement und dem Software-Entwicklungsprozeß betrachtet. Die letzten Abschnitte beschäftigen sich mit den Begriffen Prüfen, Testarten, Teststrategie, White-Box- und Black-Box-Testtechniken.

2.1 Was ist Testen?

Testen ist auf jeden Fall Vergleichen; Testen erfordert ein zu testendes Objekt und einen Bezugsrahmen, dem das Objekt entsprechen muß, die Testbasis. Testen ermittelt den Unterschied zwischen dem tatsächlichen und dem erforderlichen Zustand eines Objekts.

Für den Begriff »Testen« sind viele Definitionen im Umlauf. In TMap [Pol u.a., 1995] wird eine gut handhabbare Definition gegeben, die auch in diesem Buch verwendet wird.

Unter Testen versteht man den Prozeß des Planens, der Vorbereitung und der Messung, mit dem Ziel, die Eigenschaften eines IT-Systems festzustellen und den Unterschied zwischen dem tatsächlichen und dem erforderlichen Zustand aufzuzeigen.

Die spezifische Nennung von Aktivitäten wie Planung und Vorbereitung weist ausdrücklich darauf hin, daß Testen nicht als ein Prozeß betrachtet werden kann, der erst dann beginnt, wenn das zu testende Objekt ausgeliefert wird. Ein Testprozeß erfordert eine gute Planung und Vorbereitung, bevor mit dem eigentlichen »Messen« begonnen werden kann.

2.2 Ziel des Testens

Bei der Entwicklung und Wartung von IT-Systemen ist der Qualität besondere Aufmerksamkeit zu widmen. Qualität bedeutet dabei, daß den Erwartungen der verschiedenen Anwender entsprochen wird. In der Praxis stellt sich das für die Softwarebranche als heikle Angelegenheit heraus, weil sie wie wenige Zweige der Industrie mit dem Image kämpfen muß, schlechte Qualität zu liefern. Die Qualität bezieht sich sowohl auf die Zuverlässigkeit, die Pünktlichkeit, mit der innerhalb des festgelegten finanziellen Rah-

mens geliefert wird, als auch auf die Beantwortung der ursprünglichen Zielsetzungen, nämlich dem Entwickeln einer Lösung für ein gegebenes Problem.

Wenn Qualität also im großen und ganzen definiert werden kann als »den Anforderungen entsprechend«, liefert das Testen demnach Einsicht in das Maß an Qualität. Testen erlaubt somit eine Einschätzung der Risiken, die beim Akzeptieren einer geringeren Qualität eingegangen werden. Damit gelangen wir zur Hauptzielsetzung des Testens.

Testen reduziert das Maß an Unsicherheit in bezug auf die Qualität eines IT-Systems.

Testen kostet Zeit und Geld. Es liegt am Auftraggeber des Testprozesses zu bestimmen, wieviel Zeit und Geld er zur Verfügung stellen will, um Unsicherheiten zu reduzieren oder auszuschließen. Die Menge der Testaufwandes ist von den Risiken abhängig, die man beim Einsatz eines IT-Systems bereit ist einzugehen.

Ohne Risiken ist ein Test unnötig.

Testen gründet sich auf eine möglichst gute Stichprobe aus allen möglichen Situationen, die innerhalb des IT-Systems auftreten können, und ist in der Praxis niemals vollständig. Mit Hilfe des Testens können ausschließlich Symptome festgestellt werden. Die Gefahr dabei ist groß, daß Testen nur zur Symptombekämpfung verwendet wird, so daß also lediglich die gefundenen Fehler korrigiert werden.

Das Ergebnis des Testens sollte jedoch Anlaß zum Analysieren sein, zur Suche nach den Ursachen hinter den Problemen. Eine Diagnose wird nicht auf einzelne Fehler gegründet, vielmehr sollte auf das zugrundeliegende Fehlermuster geschlossen werden. Auf dieser Grundlage kann anschließend mit dem Korrekturprozeß begonnen werden, und das heißt bei IT-Systemen mehr als das Lösen eines isolierten Problems. *Die strukturelle Erhöhung von Qualität muß top-down, also von oben nach unten, erfolgen.* Qualität muß *implementiert*, nicht »eingetestet« werden! Eines der Fundamente des Qualitätsdenkens besteht darin, daß Vorbeugen besser (und vor allem kostengünstiger) ist als Heilen. Dieses Vorbeugen erfolgt bei den Vorbereitungen, also bei der Systementwicklung. Da die Verbesserung von Qualität durch Testen extrem teuer ist, muß bei einem schlechten System das Testen gestoppt und mit einem Neuentwurf begonnen werden.

Die beobachteten Symptome ermöglichen es, eine Bewertung durchzuführen und die Probleme zu lösen. Mindestens genauso wichtig ist, daß die Symptome es gestatten, eine Aussage zu den Risiken zu machen, die der Einsatz einer bestimmten Software-Version mit sich bringt. Auf der Grundlage der Symptome, die sich während des Testens zeigen, kann das Systemverhalten im Ernstfall extrapoliert werden. Diese Abschätzung ist von wesentlicher Bedeutung für denjenigen, der für die Freigabe oder Annahme eines Systems verantwortlich ist. Viele Probleme können vermieden werden, indem man auf der Grundlage eines guten Testberichts zu dem Schluß gelangt, daß ein System nicht oder nur teilweise eingeführt oder freigegeben werden kann. Der Aufschub der Einführung eines Systems ist

zwar schmerzhaft und häufig mit hohen Kosten verbunden, die Implementierung eines schlechten Systems ist jedoch noch viel teurer und schmerzhafter.

2.3 Produkte

Die wichtigsten Produkte eines Testprozesses sind:

- **Abweichungen**
Die gefundenen Unstimmigkeiten zwischen den Erwartungen und den tatsächlichen Ergebnissen werden Abweichungen genannt. Diese Abweichungen und deren Bearbeitung müssen sorgfältig erfaßt und verwaltet werden.
- **Qualitätsempfehlung**
Auf der Grundlage der gefundenen Abweichungen vermittelt der Testprozeß dem Auftraggeber Informationen über die Qualität des IT-Systems und gibt Empfehlungen zur Freigabe für eine nächste Phase. Es handelt sich hier ausdrücklich um eine Empfehlung, da die Tester keine Entscheidungsbefugnis haben. Der Auftraggeber kann trotz einer negativen Freigabeempfehlung dennoch beschließen, das System einzusetzen. Ein Grund dafür kann sein, daß andernfalls eine Lieferfrist verstreicht und die damit verbundenen Kosten viele Male höher sind als die möglichen Folgen einer unzureichenden Qualität des Systems.
- **Testware**
Während des Testens werden u.a. Testpläne aufgestellt, Testfälle spezifiziert, Testdatenbanken und Dateien gefüllt, Testabläufe und Testresultate erarbeitet. Diese Produkte werden Testware genannt. Wichtig ist, daß diese Testware nicht als Wegwerfprodukt betrachtet wird, sondern als etwas, das aufbewahrt und verwaltet und nach Beendigung des Projektes mit übergeben werden kann. Hiermit erreicht man, daß spätere Tests, auch in der Wartungsphase, nicht jedesmal die zeitraubende Vorbereitungsphase erneut komplett durchlaufen müssen.
- **Übrige Testdokumentation**
Neben der Testware werden auch verschiedene andere Dokumente erstellt, beispielsweise Testberichte.
- **Kennzahlen**
Während des Testens können Informationen über Fortschritt, Umfang und Qualität des Testprozesses gesammelt werden. Diese Daten können als Indikatoren für die Steuerung künftiger Testprozesse nützlich sein.

Das getestete IT-System selbst wird nicht als Testprodukt betrachtet, da der Testprozeß selbst das IT-System nicht anpaßt oder verändert.

2.4 Die Rolle des Testens innerhalb des Qualitätsmanagements

Wie bereits gesagt, ist die Lieferung von Qualität innerhalb der Softwarebranche immer noch ein Problem. Testen ist dabei keine Lösung, sondern nur eines der Instrumente, die dazu beitragen können, die Qualität von IT-Systemen zu verbessern. Testen muß in ein System von Maßnahmen eingebettet werden, mit dessen Hilfe man zu Qualität gelangt: Testen muß gleichsam in das *Qualitätsmanagement* des Unternehmens eingefügt werden.

In der Theorie des Qualitätsmanagements wurden ein Reihe von Begriffen und Konzepten entwickelt. Die ISO – die *International Standards Organisation* – spielt eine wesentliche Rolle beim Zustandekommen einheitlicher Definitionen. Die Definition von Qualität gemäß ISO lautet wie folgt:

Qualität ist die Gesamtheit von Merkmalen eines Produktes oder einer Dienstleistung bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen.

Diese Definition von Qualität zeigt sehr deutlich den »unfaßbaren« Aspekt von Qualität. Wenn etwas für den einen selbstverständlich ist, ist es das für den anderen noch lange nicht. Da Selbstverständlichkeit ein subjektiver Begriff *par excellence* ist, besteht ein wichtiger Aspekt von Qualitätsmanagement in der Minimierung als selbstverständlich angesehener Eigenschaften, indem diese in explizite Anforderungen umgesetzt werden. Dadurch kann man anhand von Prüfungen und Tests sichtbar machen, inwieweit diesen Anforderungen entsprochen wird. Zur Festlegung der Anforderungen und Überwachung des Entwicklungsprozesses werden jeweils entsprechende Maßnahmen getroffen. Diese Maßnahmen können mit »Qualitätssicherung« (engl.: *quality assurance*) umschrieben werden.

Qualitätssicherung ist die Gesamtheit aller geplanten und systematischen Aktionen, die erforderlich sind, um in ausreichendem Maße das Vertrauen darin zu vermitteln, daß ein Produkt oder eine Dienstleistung den festgelegten Qualitätsanforderungen entspricht.

Diese Maßnahmen können in drei unterschiedliche Bereiche eingeteilt werden, nämlich in Vorbeuge-, Prüf- und Beurteilungs- sowie in Korrekturmaßnahmen:

- Vorbeugemaßnahmen sollen den Mangel an Qualität *verhindern*. Dabei kann man beispielsweise an Dokumentationsrichtlinien, Methoden und Techniken denken.
- Prüf- und Beurteilungsmaßnahmen sollen den Mangel an Qualität *entdecken*. Beispiele hierfür sind die Durchführung von Überprüfungen, das Durchdenken von Problemen und – natürlich – das Durchführen von Tests.
- Korrekturmaßnahmen sollen den Mangel an Qualität *beheben*. Ein Beispiel dafür ist die Korrektur von Fehlern, die durch das Testen offengelegt wurden.

Die Maßnahmen sollen dazu führen, daß:

- Meßpunkte und -größen entstehen, die einen Hinweis auf die Qualität der Prozesse vermitteln (Normierung);
- für den einzelnen Mitarbeiter klar ist, welchen Ansprüchen seine Arbeit genügen muß und daß er diese anhand der oben genannten Normen überprüfen kann;
- es für eine unabhängige Partei möglich ist, die Produkte/Dienstleistungen anhand der oben genannten Normen zu prüfen;
- das Management bei festgestellten Mängeln an Produkten oder Dienstleistungen nachvollziehen kann, wodurch diese Mängel entstanden sind und wie diese in Zukunft vermieden werden können.

Es ist äußerst wichtig, daß zwischen den verschiedenen Maßnahmen ein Zusammenhang bestehen bleibt. Testen ist keine isolierte Aktivität, sondern nur ein Stein im Gesamtbauwerk des Qualitätsmanagements. Es bildet nur eine der möglichen Formen von Qualitätsprüfung, während Qualitätsprüfung nur eine der Aktivitäten ist, mit der Qualität gesichert werden kann. Qualitätssicherung wiederum ist nur eine der Dimensionen des Qualitätsmanagements.

Die Maßnahmen, die im Rahmen des Qualitätsmanagements getroffen werden müssen, kosten Geld. Der Mangel an Qualität(smanagement) kostet jedoch ebenfalls Geld, nämlich die Fehlerfolgekosten. Der Oberbegriff für diese unterschiedlichen Kosten heißt *Qualitätskosten*. In Abbildung 2.1 von Juran [Juran, 1988] wird der Zusammenhang zwischen den unterschiedlichen Kostenarten dargestellt.

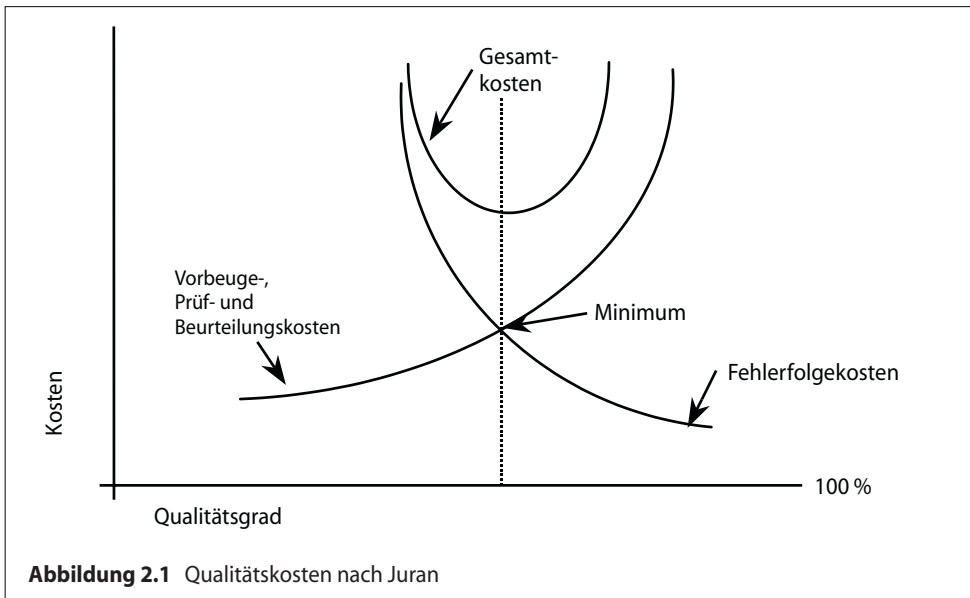


Abbildung 2.1 zeigt, daß ein Minimum bei den Qualitätskosten existiert. Unter Qualitätskosten werden folgende Kosten verstanden:

- Fehlvorbeugekosten: Kosten für die Durchführung von Vorbeugemaßnahmen
- Prüf- und Beurteilungskosten: Kosten für das Treffen von Prüf- und Beurteilungsmaßnahmen
- Fehlerfolgekosten: Kosten für die Durchführung von Korrekturmaßnahmen oder Kosten, die infolge unzureichender Qualität entstehen (Erlöseinbußen, zusätzliche Servicekosten, Garantieansprüche, Schadensersatzansprüche)

Eine wichtige Aussage von Juran ist, daß das Streben nach perfekter Qualität wirtschaftlich nicht sinnvoll ist, da irgendwann jeder kleine Zuwachs an Qualität immer größere Investitionen erfordert.

Testen ist ökonomisch sinnvoll, solange die Kosten für das Finden und die Beseitigung eines Fehlers im Test niedriger sind als die Kosten, die mit dem Auftreten eines Fehlers bei der Nutzung des Produktes verbunden sind.

2.5 Testen und Prüfen

Testen gehört zu den Prüf- und Beurteilungsmaßnahmen eines Qualitätssystems. Diese beinhalten u.a. Revision, Simulation, Inspektion, Auditing, Bewertung, Schreibtischtest, Nachvollziehung eines Problems (sog. Walkthrough). Die unterschiedlichen Prüf- und Beurteilungsinstrumente werden zwei Gruppen zugeordnet: Prüfen und Testen¹.

Unter Prüfen versteht man die Inspektion von Zwischenprodukten und Entwicklungsprozessen. Dabei handelt es sich um sämtliche Aktivitäten, die die Frage beantworten sollen, ob optimal entwickelt wird.

Unter Testen versteht man die Aktivitäten, die nötig sind um die Qualität eines Endprodukts herauszufinden. Entspricht das Produkt den Anforderungen? Wird das richtige Produkt entwickelt?

Natürlich kann diese Trennung nicht scharf gezogen werden. Ein wesentlicher Aspekt bei der Vorbereitung des Testens ist beispielsweise die Evaluierung der Testbasis, um zu prüfen, ob sie ausreichend testfähig ist (siehe auch Abb. 2.2).

1. In der Theorie wird auch Verifikation und Validierung benutzt. Verifikation beinhaltet die Evaluierung eines (Teil-)Systems, um zu prüfen, ob die Produkte der Entwicklungsphase den vorab festgelegten Anforderungen entsprechen. Validierung ist die Prüfung, ob Produkte der Systementwicklung den Wünschen und Anforderungen der Benutzer entsprechen [IEEE-Standard 1012, 1994].

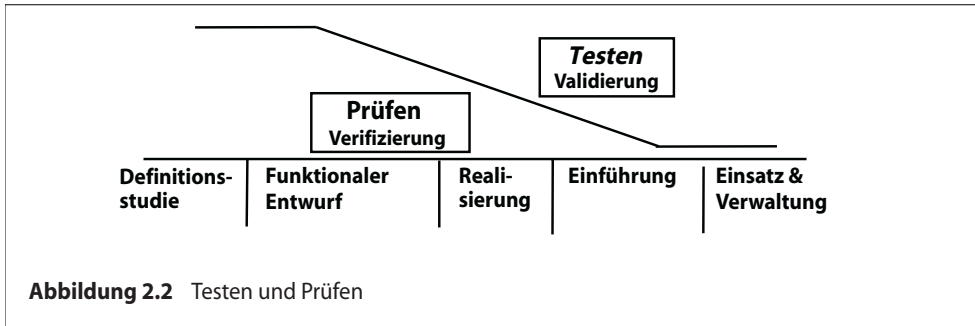


Abbildung 2.2 Testen und Prüfen

2.6 Teststufen und -techniken

Um Einsicht in den aktuellen und den geforderten Zustand des IT-Systems zu erhalten, ist es wichtig, sich der Tatsache bewußt zu sein, daß der geforderte Zustand in mehr als nur einem Dokument beschrieben ist. Beispielsweise entstehen zunächst technische und anwenderorientierte Anforderungen, nach denen dann ein funktionaler Entwurf spezifiziert wird, auf den anschließend der technische Entwurf und die Programmierung folgen. Um die Qualität des getesteten Systems einschätzen zu können, reicht es nicht aus, dieses mit dem funktionalen oder dem technischen Entwurf zu vergleichen, da diese Entwürfe ebenfalls Fehler enthalten können. Ebenso reicht es nicht aus, nur zu testen, ob das System den Anwenderanforderungen entspricht, da diese häufig auf so hohem Abstraktionsniveau formuliert sind, daß die sich darauf gründenden Testfälle nicht ausreichend viel vom Programm oder sogar von der Systemfunktionalität erfassen, um eine gute Beurteilungsgrundlage zu erhalten. Die Antwort ist natürlich, daß unterschiedliche Tests erforderlich sind, um herauszufinden, inwiefern Programme gemäß dem technischen Entwurf arbeiten, inwiefern die Anwendung gemäß dem funktionalen Entwurf arbeitet und inwiefern das gesamte System den Anforderungen des Kunden entspricht.

Damit all diese Tests effizient organisiert werden können, kommen verschiedene Testtechniken auf verschiedenen Teststufen zum Einsatz. Jede Stufe deckt dabei eine bestimmte Gruppe von Anforderungen oder funktionellen oder technischen Spezifikationen ab.

Eine Teststufe ist eine Gruppe von Testaktivitäten, die gemeinsam organisiert und gelenkt werden.

Beispiele für Teststufen sind der Modultest, der Integrationstest, der Systemtest und der Abnahmetest.

Auf der Grundlage der vorliegenden Anforderungen oder Spezifikationen und der Risiken, die eintreten, falls diese Anforderungen nicht erfüllt werden, wird bei jeder Teststufe eine Strategie definiert, um die wichtigsten Fehler so früh und so effizient wie möglich herauszufinden.

Ein wichtiger Aspekt bei der Bestimmung dieser Strategie ist die Wahl der einzusetzenden Spezifikationsverfahren. Diese Verfahren bieten eine strukturierte Vorgehensweise zur Ableitung von Testfällen von der Testbasis (beispielsweise Anforderungen, funktionale oder technische Spezifikationen) und zielen darauf ab, bestimmte Fehlertypen zu finden. Der Einsatz von gut ausgewählten Techniken ergibt eine sehr viel effizientere Fehlersuche als eine »zufällige« Spezifikation von Testfällen.

Zur Messung der Gründlichkeit, mit der ein Produkt getestet wird, wird der Begriff »Coverage« oder »Deckungsgrad« verwendet, womit man das Verhältnis bezeichnet zwischen dem, was getestet werden kann (Anzahl der möglichen Testziele), und dem, was getestet wurde. Häufig wird der Deckungsgrad im Zusammenhang mit dem Programmcode verwendet (»mit den verfügbaren Testfällen werden x % aller Programmteile oder Zustände abgedeckt«), er kann jedoch auch auf funktionale Spezifikationen bezogen werden (Bedingungen, Zweige oder Schnittstellen (Interfaces)).

Bei den Testtechniken unterscheidet man zwischen White-Box- und Black-Box-Verfahren. Die White-Box-Techniken basieren auf dem Programmcode, der Programmbeschreibung oder dem technischen Entwurf. Hierbei werden also ausdrücklich Kenntnisse über den internen Aufbau des Systems verwendet. Die Black-Box-Techniken gründen sich auf die funktionalen Spezifikationen und die Qualitätsanforderungen. Bei diesen Tests betrachtet man das System in der Erscheinungsform, die es auch beim späteren Einsatz annehmen wird. Theoretisch werden hier die Informationen darüber, wie das System aufgebaut ist, nicht verwendet, sondern es wird ausschließlich auf der Grundlage dessen beurteilt, was das System können muß. Nur die Schnittstelle nach außen ist dabei wichtig: Leistet »der schwarze Kasten« bei Input A tatsächlich Output B im Zeitraum C und Umgebung D usw.?

Tests können in zwei Kategorien eingeteilt werden, die im Kontext dieses Buches von Bedeutung sind: Low-Level- und High-Level-Tests.

Low-Level-Tests (Modul- und Integrationstest)

Diese Teststufen beziehen sich auf das Testen einzelner Systemkomponenten, beispielsweise von Prozeduren oder Funktionen, individuell oder in Kombination [Kit, 1995]. Die Low-Level-Tests werden nahezu ausschließlich von den Entwicklern durchgeführt. Schon mit der Entstehung der ersten Bausteine eines Systems werden sogenannte *Unit-, Programm- oder Modultests* ausgeführt. In welchem Maße einzelne Tests für Units oder Module eingesetzt werden, ist von der angewandten Infrastruktur und Programmiersprache abhängig. Es muß sichergestellt werden, daß die elementarsten Programmteile oder entsprechende Kombinationen davon gemäß den technischen Spezifikationen codiert sind. In diesem Buch wird hierfür ausschließlich die Bezeichnung *Modultest* verwendet.

Ein Modultest ist ein von einem Entwickler unter Laborbedingungen ausgeführter Test, der nachweisen soll, daß ein Programm(-segment) den technischen Spezifikationen entspricht.

Nachdem festgestellt ist, daß die elementarsten Teile eines Systems eine gute Qualität aufweisen, werden hieraus gebildete komplexere Teile des Systems in sogenannten Integra-

tionstests geprüft. Dabei liegt der Schwerpunkt auf dem Datenfluß und den Schnittstellen zwischen den Programmteilen bis auf Teilsystemebene. Der Integrationstest ist gleichsam ein Montagetest; abhängig von der Entwicklungsstrategie wird das System schrittweise freigegeben und in immer umfangreicheren Teilen integral getestet.

Ein Integrationstest ist ein von einem Entwickler unter Laborbedingungen ausgeführter Test, der nachweisen soll, daß ein logischer Block von Programmteilen den in den technischen Spezifikationen festgestellten Anforderungen entspricht.

Da diese Low-Level-Teststufen eine gute Kenntnis der internen Struktur eines Systems erfordern, werden hier vornehmlich White-Box-Techniken angewandt.

High-Level-Tests (System- und Abnahmetest)

Bei diesen Teststufen handelt es sich um das Testen von kompletten Produkten [Kit, 1995]. Die Tests bieten dem Entwickler Einsicht in die Qualität des Systems, das zur Abnahme angeboten wird. Ferner informieren die Tests den Auftraggeber, Anwender und Systemadministrator über den Umfang, in dem der Auftrag erfüllt worden ist und ob das System (erneut) eingesetzt werden kann. Nach der Durchführung der Low-Level-Tests und der Korrektur der entdeckten Fehler führt der Entwickler einen *Systemtest* aus, um herauszufinden, ob das System den funktionalen und den technischen Bedingungen entspricht. In der Praxis läuft es häufig darauf hinaus, daß das System zunächst »schichtweise« getestet wird, bis das gesamte System integral getestet werden kann. Der Systemtest erfordert eine in Hinsicht auf die Software und die Testdaten kontrollierbare Umgebung.

Ein Systemtest ist ein von einem Entwickler unter (gut kontrollierten) Laborbedingungen ausgeführter Test, der nachweisen soll, daß das entwickelte System oder Teile davon den in den funktionalen und technischen Spezifikationen (Fachkonzept und DV-Konzept) festgelegten Anforderungen entspricht.

Nachdem der Entwickler den Systemtest durchgeführt und die damit verbundenen Fehler korrigiert hat, wird das System dem Auftraggeber zur Abnahme angeboten. Die spezifizierten Abnahmetests können nun ausgeführt werden. Der *Abnahmetest* (auch: *Akzeptanztest*) muß Fragen beantworten wie »Kann das System (erneut) eingesetzt werden?«, »Welche Risiken bestehen dabei?« und »Hat der Lieferant seine Verpflichtungen erfüllt?«. Die Durchführung des Abnahmetests erfordert eine Umgebung, die für die meisten Aspekte repräsentativ für die Produktions- oder Praxisumgebung ist (»produktionsnahe Umgebung«).

Ein Abnahmetest ist ein von dem/den Anwender(n) und Systemadministrator(en) in einer möglichst »produktionsnahen« Umgebung ausgeführter Test, der nachweisen soll, daß das entwickelte System den funktionalen und qualitativen Anforderungen entspricht.

Innerhalb des Abnahmetests sind zwei gesonderte Teststufen zu unterscheiden, die aufgrund ihres besonderen Charakters meistens separat vorbereitet und ausgeführt werden: der funktionale Abnahmetest und der Produktions-Abnahmetest.

Der *funktionale Abnahmetest* richtet sich insbesondere auf die Funktionalität, während der *Produktions-Abnahmetest* den Nachweis erbringt, ob das System den verwaltungs- und betriebsmäßigen Anforderungen des Praxisbetriebs entspricht. Der von den Anwendern und dem Funktionsverwalter ausgeführte funktionale Abnahmetest schließt in bezug auf die Planung weitestgehend eng an den Systemtest an und ist in vielen Fällen zu Anfang schichtweise organisiert. Der Produktions-Abnahmetest wird in den meisten Fällen erst kurz vor Produktionsanlauf von den Systemadministratoren ausgeführt.

Insbesondere die High-Level-Tests können als individuelle Prozesse betrachtet (und somit auch organisiert) werden. Es handelt sich dabei um Prozesse, die parallel zum Entwicklungsprozeß verlaufen und in der Phase des funktionalen Entwurfs beginnen. Ein gutes Management dieser Prozesse und ihre Abstimmung auf den Rest des Projekts in Form von Testberichten und Rücksprachen ist sehr wichtig. Die Erfahrungen aus der Praxis zeigen, daß man sich bei High-Level-Testarten sehr viel bewußter ist, welche Bedeutung ein guter Testprozeß hat als bei Low-Level-Tests.

Da die High-Level-Tests sich häufig auf die extern sichtbaren Eigenschaften eines Systems richten, werden hier vornehmlich Black-Box-Techniken eingesetzt. Übrigens verläuft die Grenze zwischen High-Level-Tests/Black-Box-Techniken und Low-Level-Tests/White-Box-Techniken nicht so scharf, wie man annehmen könnte. Bei ersteren stellt sich immer häufiger heraus, daß man ebenfalls Kenntnisse über die internen Funktionen und die Struktur des Systems braucht. Auf der anderen Seite werden in typischen Low-Level-Tests wie dem Modultest immer häufiger funktionale Spezifikationen als Testgrundlage verwendet.

2.7 Zusammenhang zwischen Entwicklungs- und Testprozeß

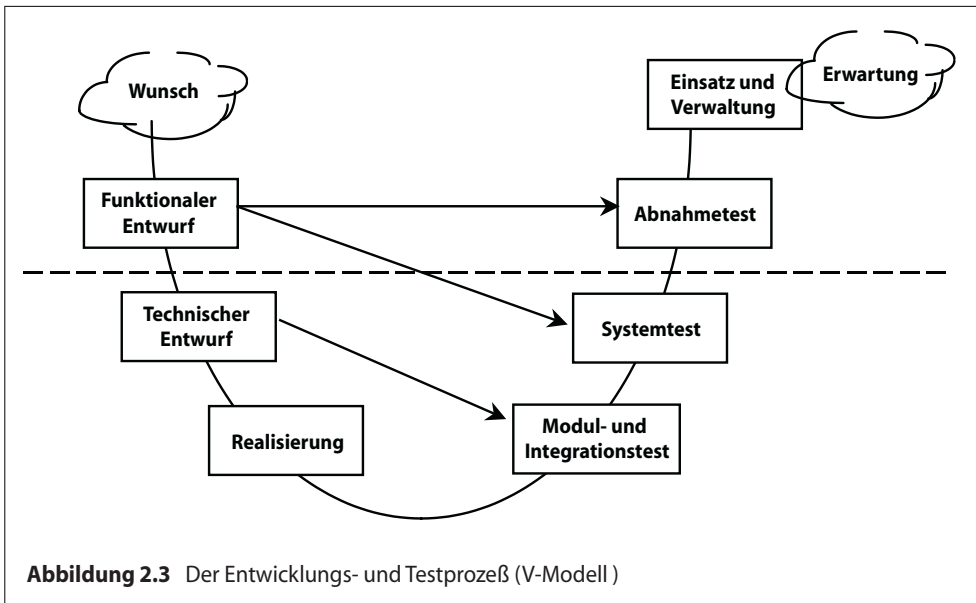
Die Entwicklung von IT-Systemen erfolgt größtenteils immer noch nach dem üblichen Phasenmodell, der »Wasserfallmethode«. Diese Methode verläuft wie folgt:

1. Man beginnt betriebsumfassend mit der Definition der *Möglichkeiten*, die die Informationstechnologie für die Lösung von Problemen oder für die Optimierung von Betriebsprozessen bietet, und weist hierbei entsprechende Prioritäten (Informationspolitik, Informationsplanung) zu.
2. Man stellt grob fest, welchen funktionalen und sonstigen *Anforderungen* das System zu entsprechen hat (Informationsanalyse, Definitionsstudie).
3. Anschließend wird festgelegt, *welche* Funktionalität entwickelt werden soll, um die Probleme zu lösen (funktionaler Entwurf oder Fachkonzept).
4. Man legt fest, *wie* diese Lösung zu erfolgen hat (technischer Entwurf oder DV-Konzept).

5. Man entwickelt das System; anschließend testet man es, führt es ein und wendet es an.

Die hier benutzte Terminologie stammt von *einem* Phasenmodell. Es bestehen zahlreiche Phasenmodelle, die alle in irgendeiner Weise ähnliche Vorgehensweisen beinhalten. Das gilt auch für moderne Varianten des klassischen Modells, wie Rapid Application Development (RAD) und Evolutionäres Entwickeln. Die Vielfalt an Teststufen kann gut mit einem solchen Phasenmodell in Beziehung gebracht werden.

Eine häufig angewandte Präsentation des Phasenmodells für Systementwicklung und Testen ist das sogenannte V-Modell [V-Modell, 1997] (siehe Abb. 2.3).



Auf der linken Seite des Modells befinden sich die Phasen, in denen das System implementiert oder modifiziert wird, ausgehend von Wunsch, Idee, Bedürfnis, Gesetz, Politik, Änderung bzw. Anforderung hin zur Lösung. Auf der rechten Seite befindet sich der entsprechende Testvorgang. Die Strichellinie deutet (formal) eine Trennung zwischen den Zuständigkeiten an: Auftraggeber, Anwender, Manager und Rechenzentrum oberhalb der Linie, Systementwickler, Zulieferer und Entwickler unterhalb. Für das Testen ist es gewiß von Bedeutung festzustellen, wo die Zuständigkeiten liegen. Wer ist der Auftraggeber für das Testen, wer fordert einen Qualitätsbericht? Jeder Phase stehen eine oder mehrere Testarten gegenüber.

Die Kästen des V-Modells stellen nur die Durchführungsphasen der unterschiedlichen Teststufen dar. Die Durchführungsphase bildet lediglich 40% des gesamten Aufwands einer solchen Testart. Die Pfeile deuten den Pfad von der Ausgangsdokumentation (der

Testgrundlage) zur Durchführung des Tests an. Die restlichen 60% sind Planungs- und Vorbereitungsaktivitäten. Das V-Modell verdeutlicht, daß die Durchführungsphasen der verschiedenen Testarten sich auf dem kritischen Pfad des Entwicklungsprozesses befinden. Es liegt auf der Hand, daß es äußerst wichtig ist, die Planungs- und Vorbereitungsaktivitäten abzuschließen, bevor mit der tatsächlichen Testdurchführung begonnen wird.