

# Systematische Testfallableitung und Tests durchführen

**Bereich**  
Kontrolle

**Aktivität**  
Interne Qualitätssicherung durchführen  
(Verifikation)

## Ziele

- Tests werden systematisch und zielgerichtet erstellt
- Vertrauen in die Qualität der Software erhöhen
- Informationen über das Qualitätsniveau gewinnen
- Fehler finden

**schnell durchführbar** ★★☆☆

**einfach durchführbar** ★★☆☆

**agil einsetzbar** ★★☆☆

## Motivation/Problemstellung

Bei der Entwicklung von Software stellt sich immer die Frage, was wie getestet werden soll. Werden Testfälle unsystematisch abgeleitet, so können Teile beim Testen vergessen werden, so dass eine Aussage über die Qualität nur schwer möglich ist. Andererseits können durch unsystematische Testfallerstellung Teile doppelt getestet werden, was wiederum die Effizienz des Testens reduziert.

## Kurzbeschreibung

Die systematische Testfallableitung sorgt für eine zielgerichtete Erstellung von aufeinander abgestimmten Testfällen. Es wird klar, welche Teile der Software wie gut durch Tests abgedeckt sind. Für unterschiedliche Qualitätsanforderungen können unterschiedliche Testfallableitungsmethoden verwendet werden.

Als Basis für die Testfallableitung dient entweder eine Spezifikation, Modelle (für Black Box-Tests) oder der Code selbst (für White Box-Tests). An Black Box-Methoden werden unter anderem die Äquivalenzklassen und Grenzwertanalyse, der zustandsbezogene Test, Ursache-Wirkungs-Graphen und Entscheidungstabellen oder der anwendungsfallbezogene Test empfohlen. Die Wahl der Methodik hängt zum Beispiel ab von den Eigenschaften des Systems, der verfügbaren Testbasis, der Teststrategie, den Testressourcen oder dem Wissen über die Methodiken. An White Box-Tests werden die Anweisungsüberdeckung, die Entscheidungsüberdeckung sowie die Bedingungsüberdeckung empfohlen, wobei die genaue Wahl zum Beispiel von den Qualitätsanforderungen, der Systemstruktur, oder dem zu erreichenden Qualitätslevel abhängt.

Die systematisch abgeleiteten Tests werden im Anschluss durchgeführt.

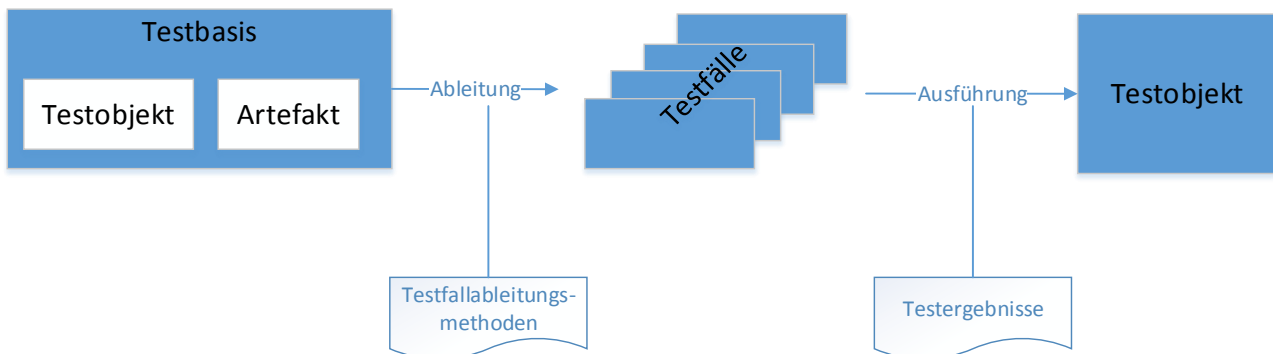


Abbildung 1: Systematische Testfallableitung und Durchführung

## Input

- Testbasis (z.B. Anforderungen, Design, Code)

## Output

- Testfälle, Verfolgbarkeitsmatrix, Aussagen über Abdeckungsgrade mit Tests

## Rahmenbedingungen

### Ausführender

Tester

### Werkzeuge, Hilfsmittel

Entsprechende Software (z.B. Excel oder fortgeschrittenere Werkzeuge), Papier, Stift

### Vorkenntnisse/Erfahrungen

Vorkenntnisse in QS und dem Ableiten von Testfällen sind hilfreich

### Ort/Umgebung

Büro

### Weitere Teilnehmer

-

### Voraussichtliche Dauer

Je nach Umfang der Testbasis und den Qualitätsanforderungen

## Vorgehensweise

### Vorbereitung

Die Testbasis sollte in einem Zustand sein, der das Ableiten von Testfällen erlaubt (z.B. ausreichend detailliert, fehlerfrei). Dies kann durch die Best Practice „Reviews von Entwicklungsartefakten“ geprüft werden. Es hängt von unterschiedlichen Faktoren ab, welche Technik zum Einsatz kommt – es muss also basierend auf den konkreten Umgebungsfaktoren wie zu Verfügung stehender Testbasis, Art des Systems, Testziele, Teststrategie oder Testressourcen ausgewählt werden, welche Testfallableitungsmethoden zur Ableitung von Testfällen genutzt werden soll. Zum Beispiel können neben der Äquivalenzklassenbildung auch die Grenzwertanalyse eingesetzt werden, wenn es sich um eine besonders kritische Funktionalität handelt; oder die Grenzwertanalyse wird immer dann eingesetzt, wenn noch Aufwand zur Verfügung steht bzw. der Tester dies als relevant ansieht. Eine allgemeine Empfehlung ist aufgrund der Menge der Einflussfaktoren und der verschiedensten Umgebungen nicht möglich.

Zudem sollte die Testbasis einen ausreichend stabilen Zustand erreicht haben, damit die erstellten Testfälle nicht fortlaufend angepasst werden müssen.

## Durchführung

Hinsichtlich Black-Box Tests ist eine der am häufigsten angewendeten Methoden zur Ableitung von Testfällen die Äquivalenzklassenbildung und Grenzwertanalyse, bei der Gruppen von Werten gesucht werden, für die das System das gleiche Verhalten zeigt. Ist das Verhalten des Systems stark von Zuständen abhängig, sollte der zustandsbezogene Test als Methode für das Ableiten von Werten für die Tests gewählt werden. Hierbei werden oft Zustandsmodelle für die Spezifikation genutzt. Für das Prüfen von Ursachen und damit verbundenen Wirkungen werden oft Entscheidungstabellen genutzt, mit deren Hilfe die Anzahl der nötigen Testfälle leicht sehr stark reduziert werden kann. Testfälle können ebenfalls aus Use Cases oder User Stories abgeleitet werden, indem für jeden möglichen Pfad eines Use Cases ein Testfall gebildet wird. Entsprechend der Granularität der Use Cases ist der abgeleitete Test in dieser Form nicht direkt automatisierbar, er kann aber mit Hilfe von Capture Replay-Werkzeugen bei der Ausführung aufgezeichnet und damit automatisiert wiederholbar gemacht werden. Bei der Ableitung aus funktionalen Anforderungen werden für jede Anforderung Testfälle gebildet. Je nach der Formulierung können auch mehrere Testfälle notwendig sein. Besonderes Augenmerk ist wie in allen Fällen auf negative Tests zu legen, d.h. solche, die unerwartete Eingaben vornehmen. Für User Stories müssen die Szenarien im Test abgebildet werden, die durch die Story beschrieben werden. Da diese Stories im Allgemeinen sehr viel knapper formuliert sind als Use Cases, müssen die einzelnen Schritte im Vorfeld verfeinert werden, um die Schritte, die ein Benutzer über der Benutzeroberfläche ebenfalls vornehmen würde, zu ermitteln. Bei den Techniken sollte mit einer Matrix die Abdeckung von Anforderungen mit Tests dokumentiert werden. Zudem ist Verfolgbarkeit sicherzustellen, d.h. festzuhalten welcher Test welche Anforderung abdeckt. Die folgende Tabelle fasst die genannten Techniken noch einmal übersichtlich zusammen.

Nr.	Testtechnik	Erklärung
1.	Äquivalenzklassenbildung	Nutzung bei Wertebereichen, Auswahl für gültige und ungültige Repräsentanten
2.	Grenzwertanalyse	Ergänzung der Repräsentanten aus Äquivalenzklassen an den Grenzwerten
3.	Zustandsbezogener Test	Berücksichtigung des Ablaufs des Systems durch Zustandsdiagramme als Basis für Testfälle
4.	Entscheidungstabellen	Aufführung einer Menge von Bedingungen und Aktionen in einer Entscheidungstabelle, so dass jede Spalte dann einem Test entspricht
5.	Use Case / User story-basierter Test	Ableitung von Tests aus Anwendungsfallbeschreibungen

Für die Erstellung von White Box Tests muss die konkrete Codestruktur in die Erstellung einbezogen werden. Als Hilfsmittel werden häufig Kontrollflussdiagramme aus dem Code abgeleitet, um die Überdeckung zu prüfen. Auf einfachen Abdeckungsstufen muss dabei nur jede Anweisung (dargestellt als Knoten in einem Kontrollflussdiagramm) einmal von einem Testfall durchlaufen werden. Für die Entscheidungsüberdeckung müssen die Testfälle so gestaltet sein, dass jede Entscheidung im Code sowohl positiv als auch negativ ausfällt. In diesem Fall reicht die Gesamtentscheidung. Bei komplexen Systemen kann es dagegen erforderlich sein, die atomaren Teilentscheidungen im Test positiv und negativ ausfallen zu lassen. Letztlich liefern alle genannten White-Box Testfalldesignmethoden eine Aussage

darüber, wie gut die zugrundeliegende Codestruktur mit Testfällen überprüft wird. Die Abdeckung des Codes kann dabei mit entsprechenden Werkzeugen gemessen werden. Eine weitere Technik für White Box Tests ist es, Test-Driven Development oder Test First zu verwenden. Dabei werden Testfälle geschrieben, ohne dass Code zur Verfügung steht. Dabei wechselt sich die Implementierung von Tests und von Code ab. Nach jedem Test wird eine geringe Menge an Code geschrieben, die es erlaubt genau diesen Testfall zum erfolgreichen Durchlaufen zu bringen.

Zur effizienten Erreichung einer hohen Codeabdeckung kann zunächst mit Black Box-Tests getestet werden. Dabei wird die Abdeckung gemessen, und bei Bedarf werden ergänzend White Box Tests erstellt. Die abgeleiteten Testfälle sind zu dokumentieren, bzw. zu implementieren.

### **Nachbereitung/Auswertung**

Nach der Ableitung sollten die Testfälle ausgeführt werden.

## Gütekriterien/Empfehlungen

Es ist keine allgemeingültige Empfehlung für eine Testfallableitungstechnik zu geben, da diese Auswahl von unterschiedlichen Faktoren abhängt, eine Hilfestellung bei der konkreten Auswahl bieten aber die zuvor genannten Heuristiken. Alle Techniken sind im agilen Umfeld einsetzbar. Zudem ist es unbedingt empfehlenswert, systematische Testfallableitung zu nutzen, um die Qualität des Systems einschätzen zu können, sowie die Tests planbar zu machen; ergänzend können erfahrungsbasierte Testfallableitungstechniken genutzt werden. Die Testfälle sollen nach den Grundsätzen der genannten Techniken für die Testableitung erstellt werden. Ebenso sollte sich die Intensität der Testfälle, d.h. die zu erreichende Abdeckung nach der Kritikalität richten. Überdeckungsmaße sollten jedoch mit Bedacht eingesetzt werden, da eine hohe Überdeckung noch keine hohe Qualität gewährleistet. Die abgeleiteten Testfälle sind zu dokumentieren.

## Risiken

Die Best Practice ist bei sich ständig ändernden Testbasen nur schwer umzusetzen, weil sich damit auch die Testfälle ändern müssen. Entgegen der Empfehlung können die Testfälle somit erst spät entworfen werden. Bei sich änderndem Code müssen auch die Testfälle häufig angepasst werden.

## Einordnung in den agilen Referenzprozess

### **Mögliche Vorgänger**

- Testen\Teststrategie festlegen und Teststufen aufeinander abstimmen
- Architektur\Architekturlösungen im Team entwickeln
- Architektur\Architekturentscheidungen dokumentieren

### **Mögliche Nachfolger**

- Testen\Fehlermanagement einsetzen
- Testen\Reviews von Entwicklungsartefakten durchführen

### **Mögliche Alternativen, verwandte Praktiken**

- Unterstützung\Checklisten verwenden
- Unterstützung\Standardisierte Dokumentstrukturen verwenden

- Testen\Erfahrungsbasiertes Testen durchführen

## Einordnung in das PQ4Agile-Qualitätsmodell

Produktqualität: Funktionale Tauglichkeit, Zuverlässigkeit, Sicherheit, Wartbarkeit, Übertragbarkeit. Strukturqualität: Fähigkeit und Schulung des Personals, Bewusstsein des Personals

## Schlagworte

Qualitätssicherung, Softwaretest, Testen, Testableitung

## Weiterführende Informationen

### Literatur

A. Spillner, T. Linz: Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB-Standard, dpunkt.verlag GmbH, 2012.

P. Liggesmeyer: Software-Qualität. Testen, Analysieren und Verifizieren von Software, Spektrum Akademischer Verlag, 2009.

I. Burnstein: Practical Software Testing. A Process-Oriented Approach. Springer-Verlag, 2003.

G. Bath, J. McKay: Praxiswissen Softwaretest – Test Analyst und Technical Test Analyst, Aus- und Weiterbildung zum Certified Tester – Advanced Level nach ISTQB-Standard, dpunkt.Verlag GmbH, 2015.

Best Practice „Systematische Testfallableitung und Tests durchführen“  
Version 1.0 – 15.06.2015 – Autor: Frank Elberzhager, Alexander Klaus, Fraunhofer IESE  
Das Projekt PQ4Agile wird vom Bundesministerium für Bildung und Forschung im  
Rahmen der Maßnahme KMU-innovativ: IKT (01 | S13032) gefördert.