

# Test (SW)

© OptiOrg Ruten www.102.ch

**Ziel**

- Tests sollen viele Fehler finden - mehr als bisher - denn Fehler gibt es immer
- Tests sollen Fehler finden - schon während der Testentwicklung - den das ist billiger
  - frühe Testentwicklung = Diskussionen und Reviews mit den Entwicklern
  - frühe Testentwicklung = zuerst Tests spezifizieren, dann Software implementieren
- Tests sollen die schwerwiegendsten Fehler finden, d.h. Fehler, die den grössten Schaden anrichten würden, wenn sie unentdeckt blieben - ö für den Rest fehlt das Geld (Nutzen/Aufwand)

**White/Glass-Box-Test, Strukturtest**

White-Box-Tests benötigen Kenntnisse über die interne Programmstruktur (wegen dieser Transparenz auch Glass-Box-Test genannt). Die Testfälle werden von der Design-Spezifikation oder direkt vom Code abgeleitet. Als ein Kriterium für das Test-Ende wird häufig vorgegeben, welche Testabdeckung (in%) erreicht werden soll. Allerdings können mit dieser Testart fehlende Funktionen nicht gefunden werden.

**Verantwortung:**  
Diese Testart wird wegen der notwendigen Detailkenntnisse und der notwendigen Code-Instrumentierung typisch von Entwickler selber angewendet (wird bei sicherheitskritischen Anwendungen wegen der fehlenden Unabhängigkeit aber nicht toleriert).

Funktionale Tests dienen der Überprüfung der funktionalen Anforderungen und werden fast ausschliesslich als Black-Box-Test ausgeführt, d.h. ohne Kenntnisse der inneren Struktur der Software (darum werden diese Begriffe hier gleichgestellt). Die Testfälle werden aus den funktionalen Spezifikationen gewonnen und prüfen die Software aus Anwendersicht.

**Black-Box-Test, Funktions-Test**

**Verantwortung:**  
Dieses Testart wird vor allem von unabhängigen Testpersonen angewendet. Diese müssen aber nicht zwingend einer unabhängigen Organisationseinheit angehören (z.B. eigene Testabteilung)

**Hinweise:**  
Bei spärlich vorhandenen oder ungenauen funktionalen Anforderungen sind Kenntnisse über die zu testende Funktion unerlässlich. Kenntnisse der Anwendung helfen beim Design von guten Testfällen. Software-Komponenten werden vor der Systemintegration typisch einem Funktionstest unterworfen.

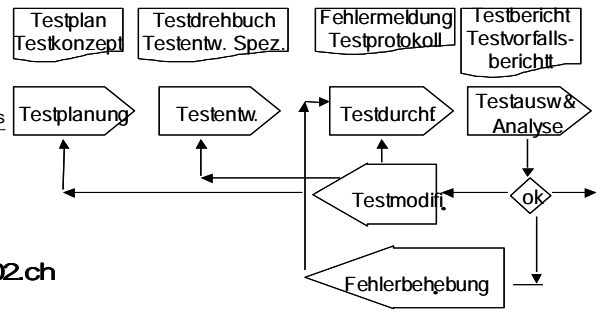
**Probleme:**  
Es werden exakt definierte Spezifikationen benötigt. Fehlen diese für die Testentwicklung, so ist anzunehmen, dass die Vorgaben auf für den Programmwurf ungenügend sind. Diese Tests decken in der Regel nur einen kleinen Teil aller Testbedingungen ab. Sie sind deshalb als Stichprobe zu verstehen. Beim Auftauchen von Fehlern ist eine Untersuchung angesagt.

**Regressions-Test**

Testen ist sowohl progressiv als auch regressiv. Die progressive Phase beinhaltet die Test von neuen Funktionen oder Applikationen. Die regressiv Phase berücksichtigt Änderungen in bereits getestetem Code, das bedeutet hauptsächlich Wiederholung von Tests. Regressionstests sind vor allem für den Produkteunterhalt (Wartung) wichtig. Die Schwierigkeit liegt darin, welche Tests ann wiederholt werden.

Ein sinnvoller Weg kann darin bestehen, ein minimales Set von Tests bei jeder Änderung auszuführen, während das gesamte Testpaket nur in bestimmten Zeitabständen ausgeführt wird. Eine andere Möglichkeit ist es den Umfang des Tests von der Tragweite der Änderung abhängig zu machen

Für dies Tests ist der Einsatz eines Testtools angebracht. Dies kann wirtschaftlich aber fragwürdig sein, wenn das Tool nicht bereits in der progressiven Phase eingesetzt wurde.



**Testprozess**

Testplan  
Testentw.  
Testdurchf.  
Testausw. & Analyse

**Rollen im Test**

- Testmanager
- Testverantwortlicher
- Testdesigner
- Tester
- Betreuer Testumgebung

By M. Ruten www.102.ch

© OptiOrg Ruten Organisation & Projektleitung Morgenholzstr. 27a CH-8912 Obfelden www.102.ch info@102.ch

**Kunde**

aber es entspricht nicht meinen Erwartungen" = image leidet Auch wenn der Kunde bleibt, so wird er misstrauisch und bildet sich eine Meinung, die für die Unternehmen selten von Vorteil ist.

**Konkurrenz**

"ich habe ein besseres Produkt" = Marktanteil leidet  
Die Konkurrenz wird bei seinen Produktdemonstrationen die Vorteile gegenüber den Mitbewerbern hervorheben - auch die Qualität

**Entwickler**

"acht Noch mehr Feldprobleme" = Time to Money ist in Frage gestellt! Motivation  
Ausgelieferte Produkte in unzulänglicher Qualität und Zuverlässigkeit führen zu einem Druck zur Fehlerkorrektur - der entsprechende Stress ist kaum ein optimales Umfeld für eine sorgfältige Fehleranalyse und -behebung

**Verkäufer**

sucht Umgebung, work arounds = Problem beliebt Bestehen  
Der gleiche Fehler erscheint immer wieder und verursacht auch immer wieder Kosten

**Teststrategie**

- wo ist die Testverantwortung
  - Entwicklungsteam
  - gemischtes Testteam
  - unabhängiges Testteam
- wie ist die zeitliche Abfolge
  - separate Testphasen
  - paralleles Testen
- wie gross sind Testumfang und Testtiefe
  - minimal Testware
  - quick & dirty-Testen
  - risikobasiertes Testen

**Teststufen**

1. Akzeptanztest
2. Feldtest
3. Integrationstest (der Systeme)
4. Systemtest
5. Integrationstest (der Einheiten)
6. Modultest
  - 6.1 Einheiten-Test
  - 6.2 Inegrationstest (der Elemente)
  - Element-Test

**Testmanager**

- Festlegung der Rahmenbedingungen/Regelungen für die Testaktivitäten eines Projektes
- Gesamtplanung aller Testaktivitäten (Testplan)
- Beauftragung und Koordination des Testverantwortlichen
- Bereitstellung der nötigen Ressourcen (Personen, Budget)
- Beratung bei der Festlegung der Testkonzepte (Methoden, Tools) der einzelnen Teststufen
- Beurteilung der Risiken und Kennzahlen
- Berichterstattung an den Projektleiter und den QS-Verantwortlichen

**Testverantwortlicher**

- Detailplanung, Steuerung und Kontrolle der Testaktivitäten einer Teststufe
- Beauftragung aller Testbeteiligten einer Teststufe
- Definition und Beurteilung von Testprioritäten
- Zusammenfassung der Testergebnisse
- Beurteilung der Testergebnisse entsprechend den Test-Endekriterien der Teststufe
- Berichterstattung gegenüber dem Testmanager und Projektleiter
- Sicherstellung der Ausbildung der Testteams (Testtechnik und Produkt)
- Sicherstellung der Verwaltung der Testware
- Entscheid in kritischen Testphasen (Testabbruch)

**Testdesigner**

- Design von Testfällen
- Erstellung der Testspezifikation
- Definition der Testumgebung
- Erstellen der Testprozeduren
- Erstellen der Testprogramme/scripts
- Testdatengenerierung
- Analyse der Testergebnisse
- Anpassung der Testware

**Tester**

- Testdurchführung
- Erstellung von Prüfprotokollen
- Meldung der gefundenen Fehler

**Betreuer Testumgebung**

- Beschaffung und Aufbau der Testumgebung (Hardware, Software)
- Support beim Betrieb der Testanlagen
- Support bei Installation und Anwendung von Testtools und Testdaten(banken)
- Unterhalt und Anpassungen der Testumgebung