

Herzlich willkommen

Testverfahren

Testverfahren

- Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden. (Myers 1979)
- Wurde ein Programm sorgfältig getestet (und sind alle gefundenen Fehler korrigiert), so steigt die Wahrscheinlichkeit, dass das Programm sich auch in den nicht getesteten Fällen wunschgemäß verhält
- Die Korrektheit eines Programms kann durch Testen (außer in trivialen Fällen) nicht bewiesen werden.
- Grund: alle Kombinationen aller möglichen Werte der Eingabedaten müssten getestet werden

Testverfahren

Prozess, der sich (sowohl statisch als auch dynamisch) mit der Planung, Vorbereitung und Bewertung einer Software und den hierzu in Beziehung stehenden Arbeitsergebnissen befasst, um die Software mit dem Ziel zu bewerten,

- dass diese allen festgelegten Anforderungen entspricht und
- ihren Zweck erfüllt und
- um etwaige Fehlerzustände zu finden.

Teststufen

Tests werden in der Regel nicht zur gleichen Zeit ausgeführt, sondern immer in Abhängigkeit vom Entwicklungsstand.

| Teststufe | Beschreibung |
|--------------------------------|---|
| Modul-, Unit-, Komponententest | Test einer einzelnen Einheit oder Teile dieser Einheit auf Funktionalität |
| Integrationstest | Überprüfung des fehlerfreien Zusammenwirkens voneinander abhängigen Systemkomponenten, Testschwerpunkt sind die Schnittstellen zwischen den Komponenten |
| Systemtest | Abschließender Test des Gesamtsystems hinsichtlich der funktionalen und nicht funktionalen Anforderungen |
| Abnahmetest | Test des Gesamtsystems in der Kundenumgebung, wird meist von den Kunden selbst durchgeführt, bildet die Grundlage für die Entscheidung zur Abnahme des Projekts |

statisches Testen (ohne Programmausführung)

- **Inspektion** – Stark formalisiert, Quellcode wird nach genau festgelegter Vorgehensweise durch Gutachter untersucht
- **Reviews** – Bei Reviews nutzt man die menschlichen Denk- und Analysefähigkeiten, um durch Lesen und Nachvollziehen das Testobjekt zu prüfen, weniger formal als Inspektion -> dadurch geringerer Aufwand
- **Walkthrough** – unstrukturiert, Entwickler präsentiert, Gutachter stellen spontan Fragen
- **Audit** – strukturierter als Walkthrough, Gutachter stellen Fragen nach vorher festgelegtem Prüfkatalog
- **Statische Analyse** – hat das Ziel, Fehler im Programmcode oder in formal beschriebenen Softwaremodellen zu finden. Die statische Analyse wird mit entsprechender Werkzeugunterstützung durchgeführt. Einsatzgebiete von statischen Analysewerkzeugen sind die Überprüfung gegen Programmierrichtlinien, Datenflussanalyse, Kontrollflussanalyse und Erstellung von Metriken.

dynamisches Testen (mit Programmausführung)

- White-Box Test
- Black-Box Test

White-Box Test:

- bezeichnet eine Methode des Software-Tests, bei der die Tests mit Kenntnissen über die innere Funktionsweise des zu testenden Systems entwickelt werden.
- Der Blick in den Quellcode ist gestattet.
- Es wird am Code geprüft.

White-Box Test

- Mögliche Ausführungspfade durch den Code werden nachverfolgt und es wird herausgefunden, welche Eingabewerte die Ausführung dieser Pfade erzwingt.
- Der Tester, in der Regel der Entwickler, verifiziert den Code entsprechend seinem Design - daher ist die Vertrautheit mit dem Code wichtig für denjenigen, der den Test durchführt.
- Der Code wird getestet, indem Eingabewerte durch den Code laufen, um festzustellen, ob die Ausgabe dem entspricht, was zu erwarten ist.
- Tester können die kleinste Anzahl von Pfaden ausarbeiten, die zum Testen des gesamten Codes erforderlich sind, oder den gesamten Code abdecken.
- Statische Analysewerkzeuge helfen bei dieser Aufgabe.

White-Box Test werden vorwiegend bei Unit-Tests angewendet

Unit-Tests: Beim Unit-Testing wird der Code getestet, um sicherzustellen, dass er wie vorgesehen funktioniert und keine Fehler oder Probleme aufweist. Darüber hinaus werden die Fehler entdeckt, bevor der Code in bereits getesteten Code integriert wird, was wiederum verhindert, dass in späteren Testphasen Fehler auftreten. White-Box-Tests bei Unit-Tests werden isoliert durchgeführt.

White-Box Test-Tools

- JUnit
- Veracode

Black-Box Test

- bezeichnet eine Methode des Softwaretests. Hierbei werden Tests anhand der Spezifikation/Anforderung entwickelt.
- Dies bedeutet, dass Tests ohne Kenntnisse über die innere Funktionsweise/Implementierung des zu testenden Systems entwickelt werden.
- Das zu testende Programm wird also als Black Box behandelt. Nur nach außen sichtbares Verhalten fließt in den Test ein.

Black-Box Test

- Zu Beginn werden die Anforderungen und Spezifikationen des Systems untersucht.
- Der Tester wählt gültige Eingaben aus (positives Testszenario), um zu überprüfen, ob die Software diese korrekt verarbeitet.
- Außerdem werden einige ungültige Eingaben (negatives Testszenario) ausgewählt, um zu überprüfen, ob die Software in der Lage ist, diese zu erkennen.
- Der Tester bestimmt die erwarteten Ausgaben für all diese Eingaben.
- Der Software-Tester konstruiert Testfälle mit den ausgewählten Eingaben.
- Die Testfälle werden ausgeführt.
- Der Softwaretester vergleicht die tatsächlichen Ausgaben mit den erwarteten Ausgaben.
- Etwaige Mängel werden behoben und erneut getestet.

Black-Box Test

- Gültiges Geburtsdatum z.B. im Intervall von heute bis heute vor 120 Jahren.
- Ungültiges Geburtsdatum, weil es in der Zukunft liegt.
- Ungültiges Geburtsdatum, weil es mehr als 120 Jahre zurückliegt

Black-Box kommt in allen Teststufen zum Einsatz

1. Unit-Tests
2. Integrationstests
3. Systemtests
4. Abnahmetests

Black-Box Test-Tools

- Selenium
- Appium

White-Box Test

Vorteile von White-Box-Tests gegenüber Black-Box-Tests

- Testen von Teilkomponenten und der internen Funktionsweise
- Geringerer organisatorischer Aufwand
- Automatisierung durch gute Tool-Unterstützung

Nachteile von White-Box-Tests gegenüber Black-Box-Tests

- Erfüllung der Spezifikation nicht überprüft
- Eventuell Testen "um Fehler herum"

Black-Box Test

Vorteile von Black-Box-Tests gegenüber White-Box-Tests:

- bessere Verifikation des Gesamtsystems
- Testen von bedeutungsmäßigen Eigenschaften bei geeigneter Spezifikation
- Portabilität von systematisch erstellten Testsequenzen auf plattformunabhängige Implementierungen

Nachteile von Black-Box-Tests gegenüber White-Box-Tests:

- zusätzlich eingefügte Funktionen bei der Implementierung werden nur durch Zufall getestet
- Testsequenzen einer unzureichenden Spezifikation sind unbrauchbar

Test Übersicht

| White Box | Black Box | Grey Box |
|---|---|--|
| 1. Die Prüfer kennen die interne Struktur der Software genau. | 1. Black-Box-Tests befassen sich nicht mit der internen Funktionsweise der Software und haben daher keine Kenntnis davon. | 1. Begrenztes Wissen über die interne Struktur sowie die Funktionsweise der Software ist vorhanden. |
| 2. Auch bekannt als Glass Box, Open Box oder codebasiertes Testen. | 2. Bekannt als Closed Box oder datengesteuertes Testen. | 2. Es ist auch als transparentes Testen bekannt. |
| 3. Es prüft die Sicherheitslücken im Code und der internen Struktur der Software. | 3. Er befasst sich hauptsächlich mit der externen Funktionalität und dem Verhalten der Software. | 3. Hier werden sowohl White- als auch Blackbox-Tests vom Testerteam durchgeführt. |
| 4. Sie werden von einem Team aus Software-Testern und Entwicklern durchgeführt. | 4. Diese Art des Testens wird entweder von den Testern/Entwicklern oder den Endbenutzern durchgeführt. | 4. Wie die Black-Box-Tests werden auch die Gray-Box-Tests von Endbenutzern und Testern durchgeführt. |
| 5. Es ist eine der umfassendsten und zeitaufwändigsten Testtechniken. | 5. Es ist eine zeit- und kosteneffektive Testtechnik. | 5. Sie ist teilweise zeitaufwendig und umfassend. |

Schreibtischtest

Der Schreibtischtest ist ein Verfahren, das im Bereich der Softwareentwicklung verwendet wird, um Algorithmen oder Routinen auf Richtigkeit zu prüfen.

Der Schreibtischtest wird nicht mit Hilfe eines Rechners durchgeführt, sondern vielmehr im Kopf des Entwicklers und mit Hilfe von Zettel und Stift.

Testdokumentation

Die einzelnen Testschritte und deren Ergebnisse müssen protokolliert werden. Dafür gibt es einen Standard: **ISO/IEC/IEEE 29119**

| Kategorie | Dokumente |
|-------------------|--|
| Übersicht | Testkonzept |
| Testspezifikation | Testentwurfsspezifikation Testfallspezifikation Testablaufspezifikation |
| Testbericht | Testobjektübergabebericht Testprotokoll Testabweichungsbericht Testabschlussbericht |

Es müssen nicht alle Dokumente in der Testdokumentation verwendet werden!