

# M 403

## Software-Test

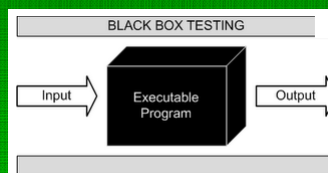
Mit Tests kann nur das Vorhandensein von Fehlern nachgewiesen werden, nicht die Fehlerfreiheit von Programmen!

19.11.2014

# M 403

## Software-Test

### Blackbox Tests



Der Tester kennt den Inhalt (Code, Funktionen) nicht!

19.11.2014

# M 403

## Software-Test

### Blackbox Tests

Basierend auf:

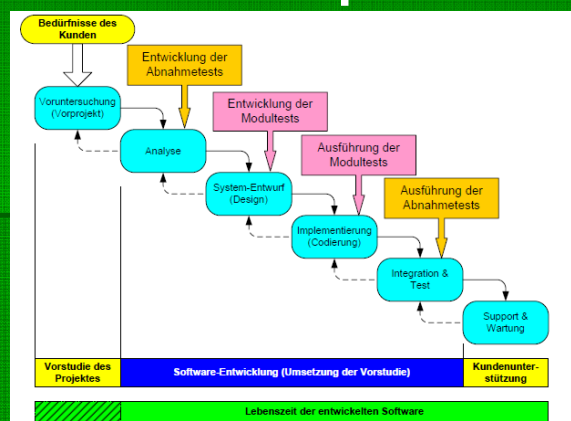
- Anforderungsspezifikation
- Benutzerhandbuch
- Modulspezifikation

19.11.2014

# M 403

## Software-Test

### Blackbox Tests: Zeitpunkte



19.11.2014

## M 403

### Software-Test

#### Blackbox Tests: Grundsätze

1. Grundsätzlich sollte der Tester beim Black Box Test den Standpunkt vertreten, dass ihn die Implementierung nicht zu interessieren hat.
2. Ein Programmierer sollte nie versuchen sein eigenes Programm zu testen.
3. Planen Sie nie Tests in der Annahme, dass keine Fehler gefunden werden.
4. Zu jedem Test gehört die Definition des erwartenden Ergebnisses vor dem Beginn des Tests.

## M 403

### Software-Test

#### Blackbox Tests: Grundsätze

5. Ein guter Testfall hat eine hohe Wahrscheinlichkeit, bisher nicht bekannte Fehler im Code aufzudecken.
6. Die Wahrscheinlichkeit, in der näheren Umgebung eines bereits bekannten Fehlers weitere Fehler zu finden ist überproportional gross
7. Die Ergebnisse von Tests müssen gründlich untersucht werden.
8. Testfälle müssen sowohl für gültige wie auch für ungültige und falsche Eingabewerte geschrieben werden.
9. Testfälle sind eine Investition

# M 403

## Software-Test

### Blackbox Tests: Gute Testfälle

- Geeignete Auswahl von Testfällen
- Halten Sie die einzelnen Testfälle möglichst klein
- Testen Sie die Grenzen (Boundary Analysis)

# M 403

## Software-Test

### Blackbox Tests: Festhalten der Ergebnisse

Nr.	Testsequenz mit Erwartungshal- tung:	Testresultat:	H:	E:	Person:	Datum:
				✓	W.O.	11.10.05
				✓	W.O.	11.10.05
				✓	W.O.	11.10.05

- **Testnummer** (*Nr.*)
- **Beschreibung der Testsequenz** und den **Erwartungen** zu diesem Test. (*Testsequenz*)
- **Beschreibung der Testresultate**. (*Testresultate*)
- **Erfüllungsgrad**: Gibt an in welchem Prozentsatz der Test erfüllt wurde. (*Feld: E*)
- Kürzel der **Testperson** (*Person*)
- **Datum** des Tests (*Datum*)
- **Datum und Unterschrift** der Testperson die bestätigt, diese Tests selber durchgeführt zu haben.

# M 403

## Software-Test

### Blackbox Tests: Testabdeckung

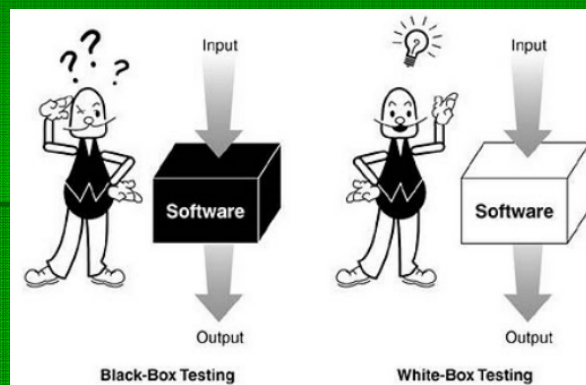
**Sinnvoll und machbar**

Mass der Testabdeckung	Beschreibung der Systemtests
S0	Jedes Modul und jede Komponente wird zumindest einmal aufgerufen.
S1	Alle Aufrufe jedes Moduls und jeder Komponente innerhalb der Programmstruktur werden durchexerziert.
S2	Dieses Mass verlangt über S1 hinaus den Aufruf der Module oder Komponente mit den möglichen und sinnvollen Fällen bei den übergebenen Parametern.
S2p	Dieses Mass der Testabdeckung verlangt über S2 hinaus das Testen aller möglichen Arten zur Beendigung eines Moduls, z.B. durch normalen Rücksprung (return) und Sonderfälle und Ausnahmen (exceptions). Das Mass S2 kann sprachabhängig sein.
Sd	Hier wird versucht, ein Programm bis zu einer gewissen Tiefe der Programmstruktur und ihrer Verästelungen auszutesten.
St	Es wird – ausgehend vom Hauptprogramm – versucht, alle möglichen Aufrufe der Komponenten und Module zu testen.
S3	Dieses Mass versucht, für alle unterscheidbaren Klassen der Programmausführung Testfälle auf der Ebene des Hauptprogramms zu kreieren.

# M 403

## Software-Test

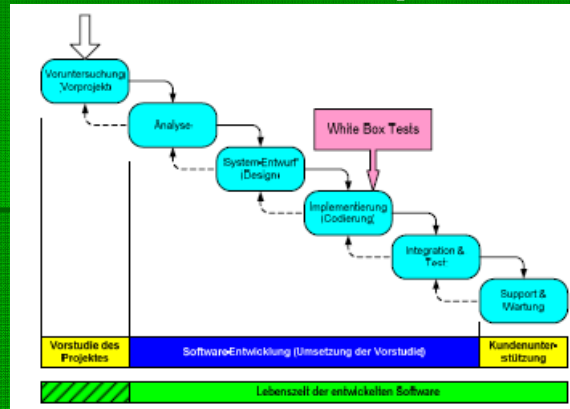
### Whitebox Tests:



## M 403

### Software-Test

#### Whitebox Tests: Zeitpunkt



## M 403

### Software-Test

#### Whitebox Tests: Ziel

*Wir wollen erreichen, dass jeder Teil des Programms im Test mindestens einmal ausgeführt wird und somit der ganze Code durch Tests abgedeckt wird.*

# M 403

## Software-Test

### Whitebox Tests: Testabdeckung

Sinnvoll und machbar

Mess der Testabdeckung	Beschreibung der White Box Tests
C0	Alle Anweisungen in einem Modul oder Programm werden ausgeführt. Bei einer Falluntersuchung (if, switch) wird aber nur immer einer der möglichen Wege abgearbeitet.
C1	Alle Segmente eines Moduls oder Programms werden mindestens einmal ausgeführt. Es werden also alle Pfade durch das Programm wenigstens einmal durchlaufen.
C1+	Alle Programmsegmente werden wie bei C1 mindestens einmal ausgeführt. Zusätzlich wird bei Schleifen mit den Extremwerten getestet.
C1p	Alle Programmsegmente werden mindestens einmal ausgeführt. Darüber hinaus wird bei jedem logischen Ausdruck so getestet, dass der Code bei jeder logischen Bedingung zumindest einmal ausgeführt wird.
C2	Alle Programmsegmente werden wenigstens einmal ausgeführt. Bei Schleifen muss so getestet werden, dass die Schleife: <ol style="list-style-type: none"> <li>1. nicht ausgeführt wird,</li> <li>2. mit einem niederen Wert des Schleifenzählers getestet wird und</li> <li>3. mit einem hohen Wert des Schleifenzählers getestet wird.</li> </ol>
Cik	Alle Programmsegmente werden mindestens einmal ausgeführt. Darüber hinaus werden alle Schleifen in dem Modul oder Programm für die Werte i bis k ausgeführt, wobei i = 1,2,3 ... k ist.
Ct	Die Kombination aller möglichen Pfade wird durch White Box Test abgedeckt.

# M 403

## Software-Test

### Whitebox Tests: Checkliste

- Wurden alle nötigen Variablen initialisiert?
- Ist eine Ganzzahldivision wirklich gewollt?
- Ist es möglich, dass eine Division durch 0 auftreten kann?
- Denken Sie an die Grenzen!
- Verzweigungen
- Schleifen
- Arrays

# M 403

## Software-Test

### Whitebox Tests: Checkliste

- Ein gefundener Fehler erspart viel höhere Folgekosten -> Erfolg
- Testen erfordert Hartnäckigkeit, ein gesundes Misstrauen, Spürsinn und Kreativität