

Arbeitskonferenz  
„Software-Qualität und Test“  
Klagenfurt, 22.-23. September 2003

Testabdeckung  
ist keine weiße Magie



Karol Frühauf, *INFOGEM AG*, CH-5401 Baden  
Karol.Fruehauf@INFOGEM.CH

# Schlussfolgerung

---

## Irrglauben

- man kann nicht vollständig Testen
- Testabdeckung hat nur mit White-Box (Glass-Box) Test zu tun
- beim Black-Box Test gibt es kein Kriterium um die Vollständigkeit des Satzes von ausgewählten Testfällen zu bestimmen
- systematische Auswahl der Testfälle ist teuer

## Tatsachen

- + man kann nicht vollständig Testen – wenn mit vollständig "alle real möglichen Ausführungen des Programms" gemeint ist
- + man kann vollständig Testen – wenn man definiert, was man mit dem Testen erreichen will
- + Testabdeckung nennt man das Kriterium, das die Vollständigkeit (und die Teststrenge) definiert
- + teuer ist nicht zu wissen, was wie getestet wurde

# Ein typisches Gespräch ...

---



# Beispiel 1: Black-Box Test der Windows Uhr



# Beispiel 1: Vollständiger Satz von Testfällen (1)

---

Ausgabe	Testfälle		
	1	2	3
analoge Anzeige		X	
digitale Anzeige			X
Schriftart (28 Arten)		Arial	TnR
Anzeige der Greenwich-Zeit		X	
Anzeige der System-Zeit			X
Anzeige der Titelleiste		X	
Anzeige ohne Titelleiste			X
Anzeige von Sekunden		X	
keine Anzeige von Sekunden			X
Anzeige des Datums			X
keine Anzeige des Datums		X	
Anzeige von Info	X		

---

## Beispiel 1: Vollständiger Satz von Testfällen (2)

---

analoge Anzeige:

8 Testfälle

	1	2	3	4	5	6	7	8
Anzeige von Zeit	gw	gw	gw	gw	sys	sys	sys	sys
Anzeige der Titelleiste	ja	ja	nein	nein	ja	ja	nein	nein
Anzeige von Sekunden	ja	nein	ja	nein	ja	nein	ja	nein
Anzeige des Datums	nein	nein	nein	nein	nein	nein	nein	nein

digitale Anzeige:

448 Testfälle

Datum möglich:

plus 8 = 16

28 Schriftarten möglich:

28 x 16 = 448

insgesamt:

analoge Anzeige + digitale Anzeige + Info =  
 $8 + 448 + 1 = 457$  Testfälle

# Beispiel 1: Abdeckung

---

im ersten Fall (3 Testfälle)

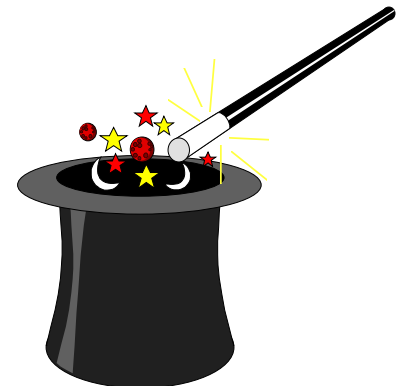
⇒ von jeder Art der Anzeige ist mindestens eine der möglichen Ausgaben in mindestens einem Testfall zur Anzeige gebracht

im zweiten Fall (457 Testfälle)

⇒ jede mögliche Kombination der Ausgaben ist in mindestens einem Testfall zur Anzeige gebracht

eine mögliche Zwischenlösung (30 Testfälle)

⇒ jede mögliche Ausgabe ist in mindestens einem Testfall zur Anzeige gebracht



## Beispiel 2: Black-Box Test von Kreditgesuch ablehnen

---

### Benutzer

### System

- |  |  |
|--|--|
| 1. Kreditgesuch wählen, o.k.               | 2. alle Ausgabefelder ausser „abgelehnt am“ gefüllt anzeigen |
| 3. Ablehnungsgrund auswählen, o.k.         | 4. Frage nach Löschen der Sicherheiten                       |
| 5. Sicherheiten Löschen: Ja eingeben, o.k. | 6. „abgelehnt am ..“ und Status „abgelehnt“ anzeigen         |
- 

### Alternativer Pfad a:

5a: Sicherheiten löschen: Nein eingeben

### Alternativer Pfad b:

4b: keine Sicherheit vorhanden; keine Frage nach Löschen der Sicherheiten

5b: keine Antwort nötig



## Beispiel 2: Angestrebte Abdeckung

---

1. Positive Fälle: Mindestens in einem Testfall ist berücksichtigt
  - a) jede Kreditart und jeder mögliche Ablehnungsgrund
  - b) das Auslösen der Aktion aus allen Workflow-Positionen, in der sie möglich ist
  - c) das Auslösen der Aktion von Benutzern jeder Benutzerklasse, die sie durchführen darf
  - d) Kredit mit und ohne Sicherheiten
  - e) das Abbrechen jeder Aktion in jedem Stand seiner Bearbeitung
2. Negative Fälle: Mindestens in einem Testfall ist berücksichtigt
  - a) das Auslösen der Aktion aus allen Workflow-Positionen, in der sie verboten ist
  - b) das Auslösen der Aktion von Benutzern jeder Benutzerklasse, die sie nicht durchführen darf

## Beispiel 2: Testdimensionen für Kreditgesuch ablehnen

---

Dimension	gültige & <i>ungültige</i> Werte
1. Kreditart (5, 1)	H, D, K, KK, O, <i>andere</i>
2. Stand der Bearbeitung (m, n)	Workflow-Position (m mögliche, <i>n verbotene</i> )
3. Benutzerklasse (2, 1)	gleicher, <i>anderer</i> , Boss
4. Sicherheiten (4)	0, 1, > 1, max
5. Sicherheit löschen (2)	ja, nein
6. Ablehnungsgrund (10, 1)	aus der Liste (10 möglich), <i>keiner</i>

---

*kursiv: "ungültige" Eingabe*

Annahme: die Art der Sicherheit hat keinen Einfluss darauf, ob sie gelöscht werden darf oder muss

## Beispiel 2: Anzahl Testfälle

---

für die angestrebte Abdeckung:

= grösste Anzahl gültiger Werte in einer Dimension (positiv) plus Summe der Anzahl ungültiger Werte, die negative Bearbeitung verursachen

=  $\max(5, m, 2, 2, 2, 10) + (1 + n + 1 + 1) = 10 + 20$

= 30 Testfälle für  $m = 5$  und  $n = 17$

für eine strengere Abdeckung

"alle mögliche Kombinationen der Ausprägungen in den Dimensionen"

= Produkt der Anzahl gültiger Werte für die einzelnen Dimensionen (positiv) plus die Summe der Anzahl ungültiger Werte, die negative Bearbeitung verursachen

=  $(5 * m * 2 * 4 * 2 * 10) + (n + 1 + 1 + 1) = 4000 + 20$

= 4020 Testfälle für  $m = 5$  und  $n = 17$

*Welcher Teil der angestrebten Abdeckung ist hier nicht abgedeckt?*

# Sinn und Zweck der Abdeckung beim Black-Box Test

---

- Testen ist ein Stichprobeverfahren
  - die Stichprobe hängt von den Risiken ab
    - der Umfang der Stichprobe bestimmt das "Vertrauensintervall" des Tests
- ⇒ die Abdeckung definiert die Stichprobe
  - ⇒ die Abdeckung ist eine Vorgabe für den Tester
    - ⇒ die Abdeckung definiert die Strenge und damit die Kosten des Tests
    - ⇒ die Abdeckung ermöglicht dem Projektleiter / Manager den Stand der Dinge (besser) zu bewerten