

Software-Qualitätssicherung

Testing can show the presence but not the absence of defects
Edsger Wybe Dijkstra (1930-2002)

Hans-Jürgen Stemmer
Herbst 2003

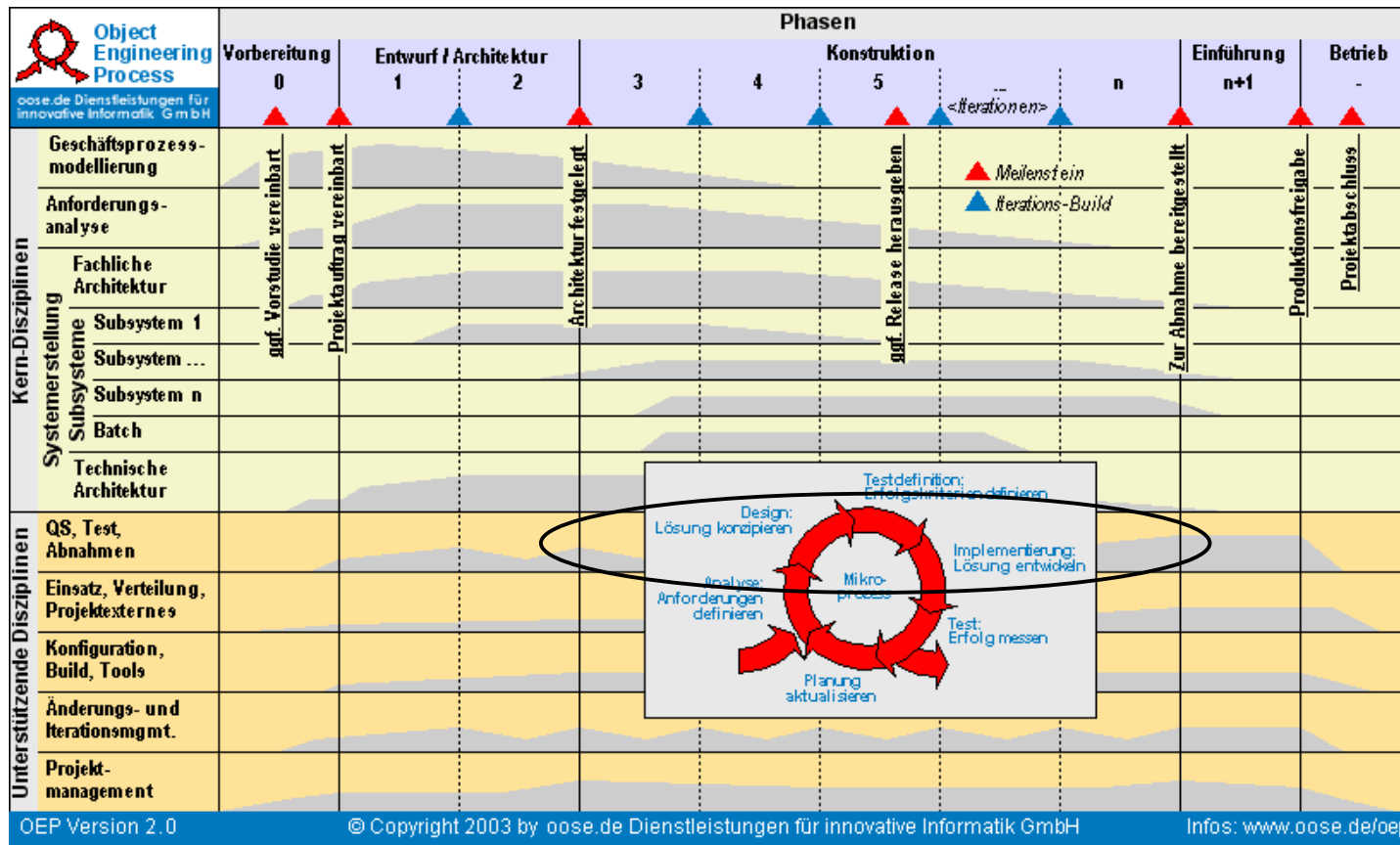
Inhalt

- Abgrenzung
- Funktionsorientiertes Testen
- Kontrollflussorientiertes,
strukturorientiertes Testen
- Werkzeuge für das Testen von Software

Inhalt

- **Abgrenzung**
- Funktionsorientiertes Testen
- Kontrollflussorientiertes,
strukturorientiertes Testen
- Werkzeuge für das Testen von Software

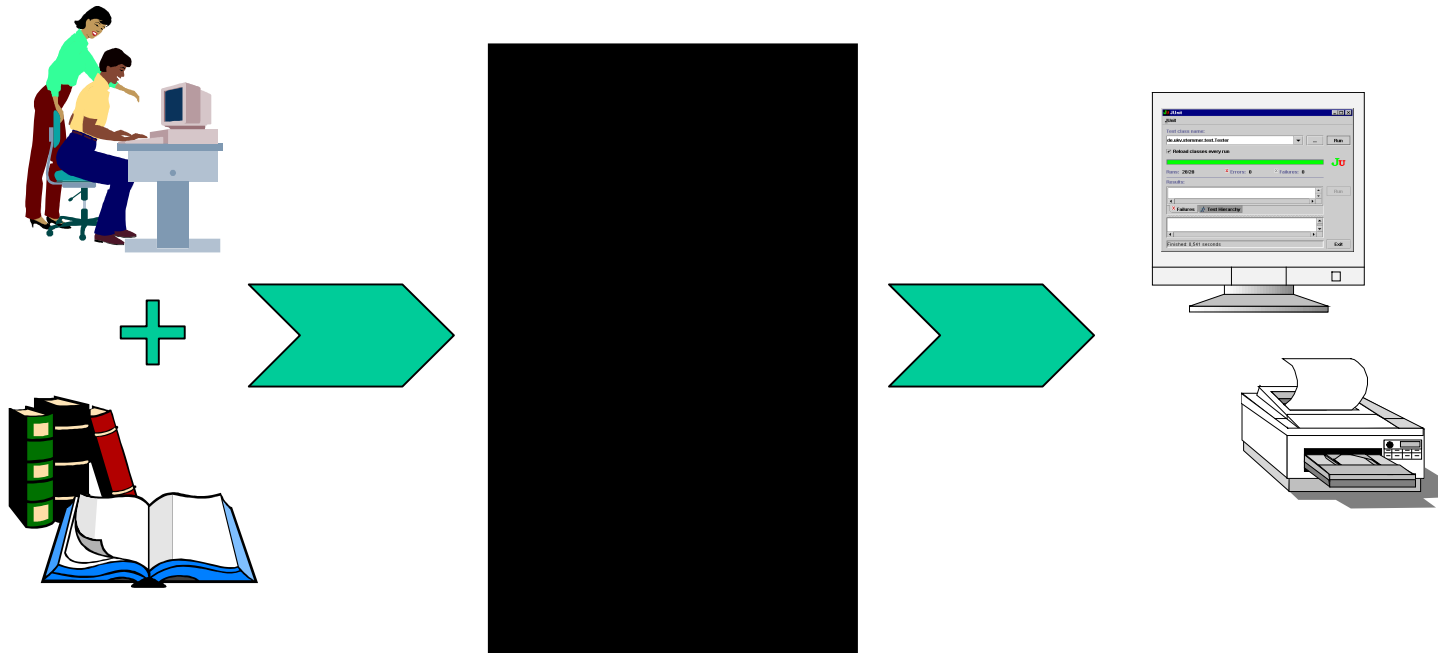
Abgrenzung



Inhalt

- Abgrenzung
- **Funktionsorientiertes Testen**
- Kontrollflussorientiertes,
strukturorientiertes Testen
- Werkzeuge für das Testen von Software

Black Box Test



Äquivalenzklassen

Sei AJ ein Tarif für eine Auslandsreisekrankenversicherung. Der Tarif kann von Personen bis 69 Jahren zu einem Tarifbeitrag von 8€ abgeschlossen werden. Ältere Personen zahlen 25€. Der Tarif gilt für ein Jahr und für Auslandsreisen, die nicht länger als 56 Tage dauern.

Sei AR ein Tarif für eine Auslandsreisekrankenversicherung. Der Tarifbeitrag wird nach der Dauer der Auslandsreise berechnet. Personen bis 69 Jahre können Reisen bis zu 90 Tagen zu einem Beitrag von 30 Cent pro Tage versichern. Jeder Tag, der darüber hinaus versichert wird, kostet 1,50€. Ältere Personen können Reisen bis zu 90 Tagen für 1€ pro Tag versichern. Längere Reisen werden für diese Personengruppe nicht versichert. Der Mindestbeitrag für alle Personen beträgt 3€.

Wenn der Tarif AJ vor mehr als einem Jahr abgeschlossen wurde, können AJ und AR kombiniert werden, so dass Reisen bis zu 56 Tagen durch den Tarif AJ versichert sind und für weitere Tage ein Tarif AR abgeschlossen werden kann.

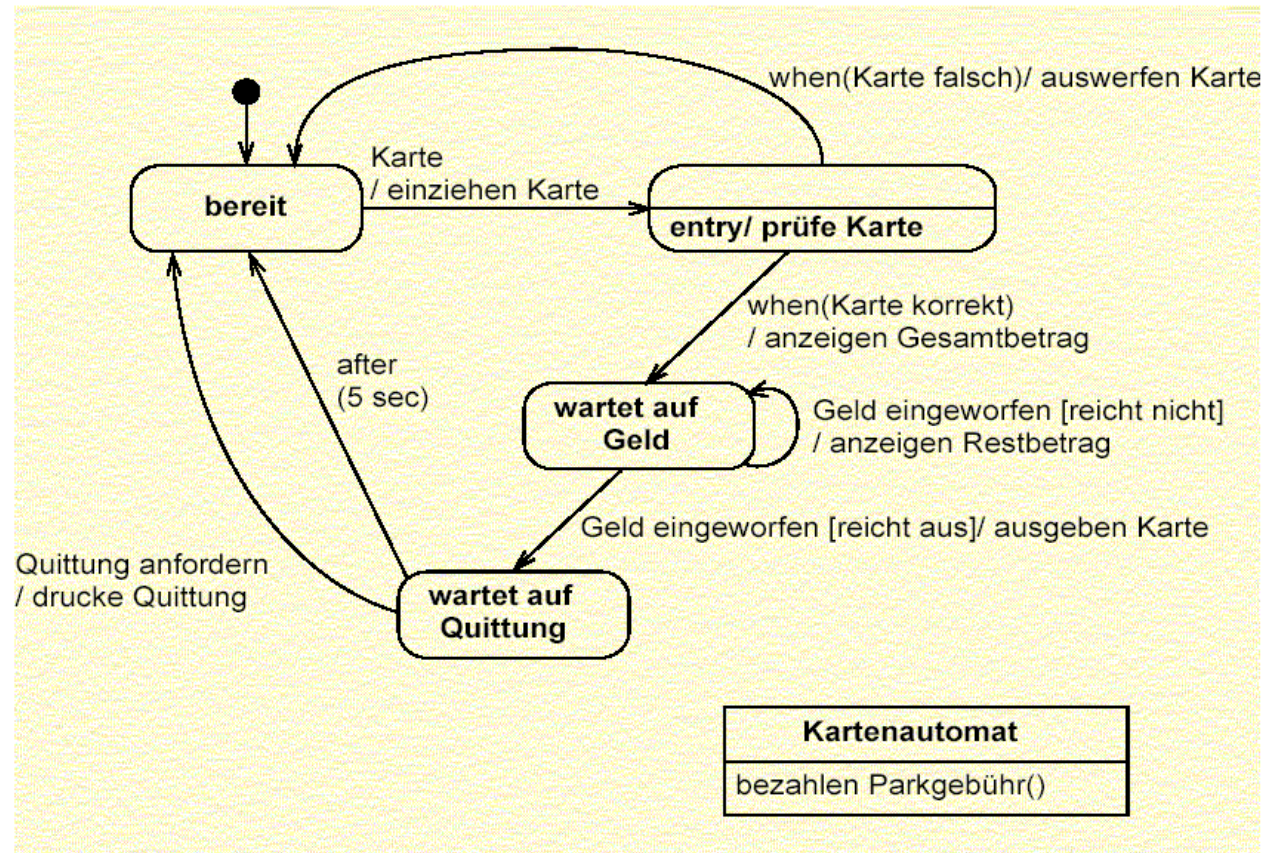
Äquivalenzklassen

- Falls die Eingabebedingung einen zusammenhängenden Wertebereich oder eine Anzahl von Werten spezifiziert, so werden eine gültige und zwei ungültige Äquivalenzklassen gebildet.
- Falls eine Eingabebedingung eine Menge von Werten spezifiziert, die unterschiedlich behandelt werden, so ist für jeden Wert eine eigene gültige Äquivalenzklasse zu bilden. Für alle Werte mit Ausnahme der gültigen Werte ist eine ungültige Äquivalenzklasse zu bilden.
- Falls eine Eingabebedingung eine Situation festlegt, die zwingend erfüllt sein muß (z.B. Buchstabe als erstes Zeichen), so sind eine gültige und eine ungültige Äquivalenzklasse zu bilden.
- Falls Grund zu der Annahme besteht, dass Element einer Äquivalenzklasse unterschiedlich behandelt werden, so ist die Klasse entsprechend aufzutrennen.

Testfälle

Tarif	Alter	Reisedauer	Komb. AJ	Tarifbeitrag
AJ	Alt (>=70)	-	-	25€
AJ	Jung (<=69)	-	-	8€
AR	Alt	Kurz (<=56)	Nein	56€
AR	Alt	Kurz (<=56)	Ja	sinnlos
AR	Alt	Kurz (>=57, <=90)	Nein	57€
AR	Alt	Kurz (>=57, <=90)	Ja	28€ (Mindestb.)
AR	Alt	Mittel (>=91)	-	ungültig
AR	Jung	<=0	-	ungültig
AR	Jung	Kurz (1)	Nein	3€ (Mindestb.)
AR	Jung	Kurz (<=56)	Nein	16,80€
AR	Jung	Kurz (<=56)	Ja	sinnlos
AR	Jung	Kurz (>=57, <=90)	Nein	17,10€
AR	Jung	Kurz (>=57, <=90)	Ja	11€ (Mindestb.)
AR	Jung	Mittel (>=91; <=365)	Nein	28,50€
AR	Jung	Mittel (>=91; <=365)	Ja	19,70€
AR	Jung	>=366	-	ungültig

Zustandsautomaten



Zustandsübergangstabelle

Zustand/ Ereignis	bereit	entry	wartet auf Geld	wartet auf Quittung
Karte	entry			
Karte falsch		bereit		
Karte korrekt		wartet auf Geld		
Geld [reicht]			wartet auf Qu.	
G. [reicht nicht]			wartet auf Geld	
Quittung				bereit
"Timeout"				bereit

Inhalt

- Abgrenzung
- Funktionsorientiertes Testen
- **Kontrollflussorientiertes,
strukturorientiertes Testen**
- Werkzeuge für das Testen von Software

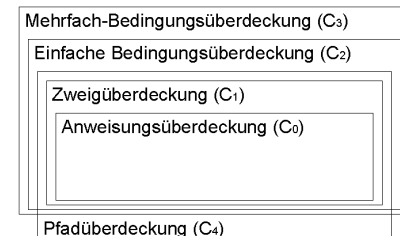
Anweisungsüberdeckung (C_0)

$$C_0 = \frac{\text{Anzahl der ausgeführten Anweisungen}}{\text{Gesamtzahl der Anweisungen}}$$

do

something()

while (a < 1)

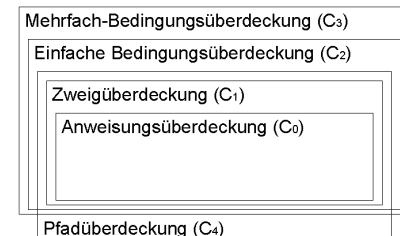


Zweigüberdeckung (C₁)

$$C_1 = \frac{\text{Anzahl besuchter Zweige}}{\text{Gesamtzahl der Zweige}}$$

- *Minimales Testkriterium*

```
if (a > 1)
    doThis()
else
    doThat()
```



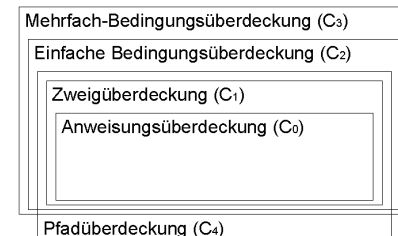
Bedingungsüberdeckung (C_2)

einfacher Bedingungsüberdeckungstest

$$C_2 = \frac{\text{Anzahl atomare Prädikate}_{\text{TRUE}} + \text{Anzahl atomarer Prädikate}_{\text{FALSE}}}{2 * \text{Gesamtzahl aller atomaren Prädikate}}$$

A	B	A B	
1	1	1	
1	0	1	x
0	1	1	x
0	0	0	

A	B	A B	
1	-	1	x
0	1	1	x
0	0	0	x



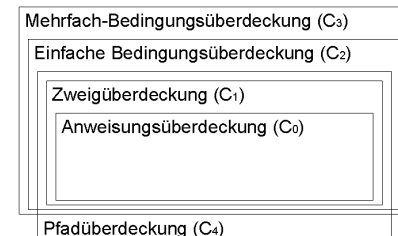
Bedingungsüberdeckung (C_3)

(minimaler) Mehrfach-Bedingungsüberdeckungstest

$$C_3 = \frac{\text{Anzahl möglicher Wertekombinationen}_{\text{TRUE}} + \text{Anzahl mögl. Kombinationen}_{\text{FALSE}}}{2 * \text{Gesamtzahl aller möglichen Wertekombinationen}}$$

$$C_{\text{MinMehrfach}} = \frac{\text{Anzahl Prädikate}_{\text{TRUE}} + \text{Anzahl Prädikate}_{\text{FALSE}}}{2 * \text{Anzahl aller Prädikate}}$$

A	B	A&&B	C	(A&&B) C	1.	2a	2b
1	1	1	(1)	1	X	X	X
1	1	1	(0)	1			
1	0	0	1	1		X	
1	0	0	0	0			X
0	(1)	0	1	1			X
0	(1)	0	0	0			
0	(0)	0	1	1			
0	(0)	0	0	0	X	X	

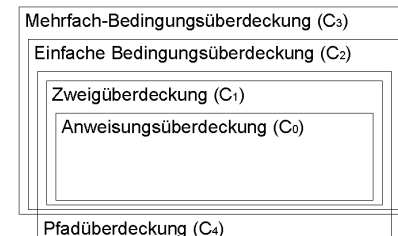


Bedingungsüberdeckung

modifizierter Mehrfach-Bedingungsüberdeckungstest

- *Kombinationen*

A	B	C	(A&&B) C	1.	2.	3.
1	1	(1)	1			
1	1	(0)	1	X	X	
1	0	1	1			
1	0	0	0		X	
0	(1)	1	1			X
0	(1)	0	0	X		X
0	(0)	1	1			
0	(0)	0	0			

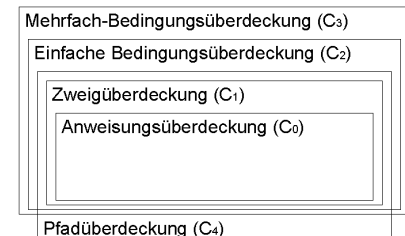


Pfadüberdeckung (C_4)

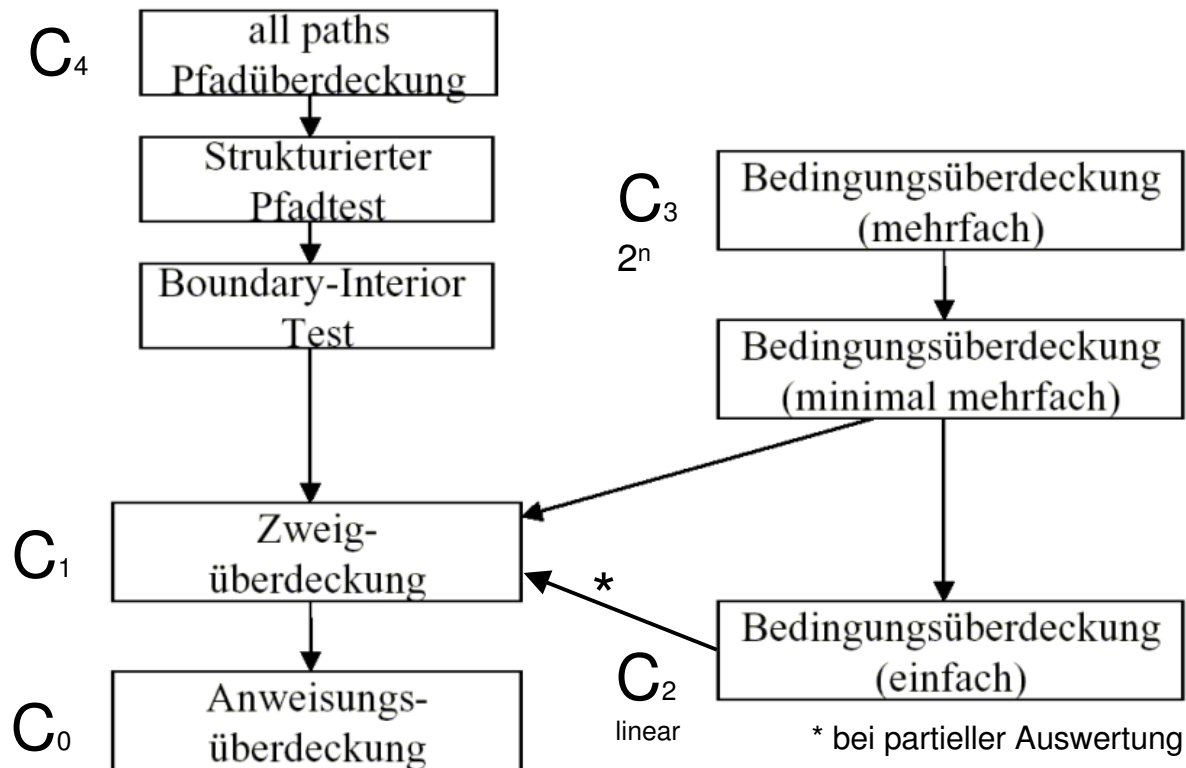
$$C_4 = \frac{\text{Anzahl besuchter Pfade}}{\text{Gesamtzahl der Pfade}}$$

Boundary-interior-Test

Strukturierter Pfadtest



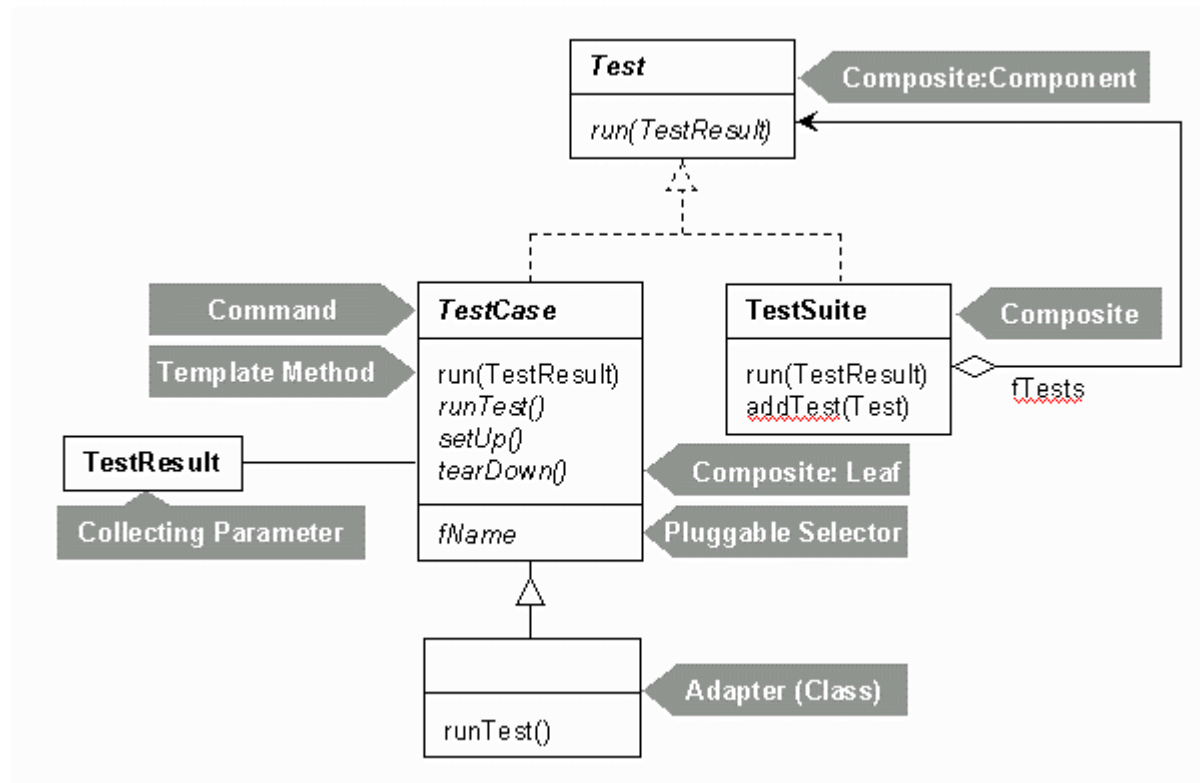
Kontrollflussorientiertes, strukturorientiertes Testen



Inhalt

- Abgrenzung
- Funktionsorientiertes Testen
- Kontrollflussorientiertes,
strukturorientiertes Testen
- **Werkzeuge für das Testen von Software**

JUnit



JProbe

Coverage Browser: snapshot_2

File Edit Tools Display Window Help

Show Only Methods With Coverage: <= 100%

Name	Calls	Hit Methods	Total Methods	Hit Lines	Total Lines
de.ukv.stemmer.test.Tester	2.137	45 (58,4%)	77	186 (68,9%)	270
de.ukv.stemmer.test	2.137	45 (58,4%)	77	186 (68,9%)	270
TarifrechnerException	16	1 (50,0%)	2	1 (50,0%)	2
TarifPropertyException	22	3 (75,0%)	4	11 (78,6%)	14
TarifCrosscheckException	14	3 (75,0%)	4	9 (75,0%)	12
TarifFactory	250	4 (40,0%)	10	29 (50,9%)	57
Tarif	1.081	9 (64,3%)	14	24 (82,8%)	29
AJ	98	5 (83,3%)	6	20 (90,9%)	22
AR	218	5 (83,3%)	6	43 (93,5%)	46
Person	382	7 (41,2%)	17	29 (56,9%)	51
Person\$Sex	3	2 (100,0%)	2	4 (100,0%)	4
Property	5	2 (100,0%)	2	7 (100,0%)	7
Tarifrechner	48	4 (40,0%)	10	9 (34,6%)	26

java - de.ukv.stemmer.test.AJ

Tools Display Window Help

Missed Block Previous Missed Block Methods: AJ.validate(Person)

Source Code

```
1      if (beginn.before(Calendar.getInstance()))
1          newTarif.setProperty(Property.AJ, new Object()
    )
1  }
    public String toString()
    {
54      Calendar beginn = (Calendar)getProperty(Property.BEGINN);
55      0      return getName() + " ab " + Utilities.format(beginn);
56      0
57  }
58  /**
```

Legend

Hit Lines: Missed Lines:

de.ukv.stemmer.test.AJ.validate(de.ukv.stemmer.test.Person)

The END

Kontrollflussorientiertes, strukturorientiertes Testen

