

Test und Verlässlichkeit Grosse Übung zu Foliensatz 4

Prof. G. Kemnitz

18. November 2016

1 Testen allgemein

Aufgabe 4.1: Statische und dynamische Tests

Einer Programmieraufgabe in ein Praktikum wird folgenden Test unterzogen:

1. Erstellen der Aufgabenstellung: Korrekturlesen durch den Lehrenden.
2. Beispielimplementierung durch den Lehrenden, Syntaxtest.
3. Beispiele ausprobieren.
4. Lösungssuche durch den Studierenden und Diskussion über die Lösung mit dem Praktikumpartner.
5. Programmieren und Syntaxtest,
6. Ausprobieren.
7. Praktikumsbetreuer schaut sich den Code an.
8. Praktikumsbetreuer lässt sich Beispiel vorführen.

Welche der Tests sind statisch und welche dynamisch?

Zur Kontrolle

1. Erstellen der Aufgabenstellung: Korrekturlesen durch den Lehrenden (S).
2. Beispielimplementierung durch den Lehrenden, Syntaxtest (S), Beispiele ausprobieren (D).
3. Lösungssuche durch den Studierenden und Diskussion über die Lösung mit dem Praktikumpartner (S).
4. Programmieren und Syntaxtest (S), Ausprobieren (D).
5. Praktikumsbetreuer schaut sich den Code an (S).
6. Praktikumsbetreuer lässt sich Beispiel vorführen (D).

(S – statisch, D – dynamisch).

Aufgabe 4.2: Produkthaftung und Testauswahl

1. Wie wirkt sich die Produkthaftung auf die Testauswahl aus?
2. Was ist für einen Testfall nach IEEE-Standard 829 zusätzlich zu den Eingaben und Sollausgaben zu dokumentieren?

Zur Kontrolle

1. Außer der Sicherung einer ausreichenden Verlässlichkeit muss der Hersteller bei einem durch Fehler verursachten Schadensfall nachweisen, dass er entsprechend Stand der Technik ausreichend getestet hat. Er muss also dokumentieren, was er getestet hat und dass das entsprechend geltender Standards ausreicht.
2. Zusätzlich zur Eingabe und Sollausgaben ist nach geltenden Standards für jeden Testfall zu dokumentieren:
 - Testfall-Identifikation: eindeutiger Bezeichner.
 - Testgegenstand: Referenz auf die Beschreibung, aus der Anforderungen überprüft werden.
 - Zweck: Anforderung, deren Erfüllung der Test bestätigt.
 - Testfallstatus: spezifiziert, durchgeführt, ...

2 Statische SW-Tests**Aufgabe 4.3: Inspektionsfehlerüberdeckung**

Inspektionsergebnisse für ein Programm aus 1000 Codezeilen:

- Inspekteur 1: 85 gefundene Fehler
- Inspekteur 2: 76 gefundene Fehler
- Schnittmenge: 56 übereinstimmende gefundene Fehler.

Schätzen Sie nach dem Verfahren »Capture-Recapture«

1. die Gesamtanzahl der Fehler,
2. die Anzahl der nicht gefundenen Fehler und
3. die Inspektionsfehlerüberdeckung.

Zur Kontrolle

1. Geschätzte Gesamtfehleranzahl:

$$\varphi = |\mathbf{M}| \approx \frac{|\mathbf{M}_1| \cdot |\mathbf{M}_2|}{|\mathbf{M}_1 \cap \mathbf{M}_2|} = \frac{85 \cdot 76}{58} \approx 115$$

2. Geschätzte Anzahl der nicht gefundenen Fehler:

$$\varphi_{\text{NErk}} \approx |\mathbf{M}| - |\mathbf{M}_1 \cup \mathbf{M}_2| \approx 115 - (85 + 76 - 58) \approx 10$$

3. Inspektionsfehlerüberdeckung:

$$IFC = 1 - \frac{\varphi_{\text{NErk}}}{\varphi} \approx 1 - \frac{10}{115} \approx 91\%$$

Aufgabe 4.4: Effizienz und Effektivität

In der Aufgabe zuvor hat der erste Inspekteur zehn Stunden für das Aufspüren seiner 85 gefundenen Fehler und der zweite Inspekteur 12 Stunden für das Aufspüren seiner 76 Fehler benötigt. Wie groß waren Effizienz¹ und Effektivität² beider Inspektoren einzeln und wie groß waren Effizienz und Effektivität der gesamten Inspektion?

¹Gefundene Fehler pro Stunde.

²Gefundene Fehler auf 1000 Nettocodezeilen.

Zur Kontrolle

	Insp. 1	Insp. 2	zusammen
gefundene Fehler	85	76	85+76-56=105
Zeit	10 h	12 h	22 h
Effizienz	$8,5 \frac{\text{Fehler}}{\text{h}}$	$8,3 \frac{\text{Fehler}}{\text{h}}$	$4,8 \frac{\text{Fehler}}{\text{h}}$
Effektivität	$85 \frac{\text{Fehler}}{1000 \text{ NLOC}}$	$76 \frac{\text{Fehler}}{1000 \text{ NLOC}}$	$105 \frac{\text{Fehler}}{1000 \text{ NLOC}}$

Aufgabe 4.5: Typ und Wertebereichkontrollen

In VHDL seien folgende Typen und Variablen definiert:

```

type tWahrsch is range 0.0 to 1.0;
type tEX is 0.0 to 10.0;
variable w, w1, w2, w3, w4: tWahrsch;
variable EX: tEX;

```

Welche der nachfolgenden Zuweisungen sind

- typentechnisch erlaubt und
- welche weisen bei der Abarbeitung immer zulässige Werte zu?

```

n1: w3 := 0.5 + w1 * w2;
n2: w4 := (0.1*w1) + (0.9*w2);
n3: w := 1 - (1-w1)*(1-w2);
n4: Ex := 2.0 * (w1+w2+w3+w4);

```

Ergänzen Sie fehlende Typumwandlungen bei Typunverträglichkeit und Assert-Anweisungen vor möglichen Wertebereichsüberläufen.

Zur Kontrolle

```

type tWahrsch is range 0.0 to 1.0;
type tEX is 0.0 to 10.0;
variable w, w1, w2, w3, w4: tWahrsch;
variable EX: tEX;
...
n1: assert w1*w2<=0.5;
    w3 := 0.5 + w1 * w2;
n2: w4 := (0.1*w1) + (0.9*w2);
n3: w := 1.0 - (1.0-w1)*(1.0-w2);
n4: Ex := 2.0 * (tEx(w1)+tEx(w2)+tEx(w3)+tEx(w4));

```

n1:	Assert-Anweisung ergänzt
n2:	alle Typ- und WB-Zuordnungen o.k.
n3:	ganzzahlige »1« durch »1.0« ersetzt
n4:	Konvertierungen von »tWahrsch« nach »tEx«

Aufgabe 4.6: Statische Code-Analyse

Nennen Sie drei Kontrollmöglichkeiten für Software, die der statischen Code-Analyse zuzuordnen sind.

Zur Kontrolle

- Kontrolle, dass alle Variablen vor ihrer ersten Nutzung initialisiert werden.
- Kontrolle der Einhaltung von API-Benutzerregeln durch Treiber.
- Kontrolle auf Nichtverwendung von Code-Bausteinen, die als problematisch gelten, z.B. in C »strcpy()«.

3 Statische Test für BG und CPS

Aufgabe 4.7: MDA und ICT

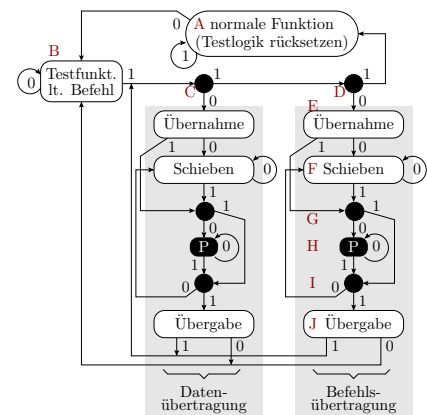
1. Wodurch unterscheidet sich der analoge In-Circuit-Test von einer Zweipunktmessung zur Kontrolle auf Fertigungsfehler?
2. Ersetzt ein digitaler In-Circuit-Test Zweipunktmessungen zur Kontrolle auf Fertigungsfehler vollständig?
3. Was bedeutet bei Boundry-Scan »Ersatz der Nadelbettadapters durch Silicon Nails«?

Zur Kontrolle

1. Die Strom-Spannungs-Beziehung einer Zweipunktmessung zur Kontrolle auf Fertigungsfehler hängt außer vom Bauteil zwischen den Punkten auch von der umgebenden Schaltung ab. Beim analogen In-Circuit-Test werden die wegfließenden Ströme von einem der Punkte unterdrückt und so die Abhängigkeit der gemessenen Strom-Spannungs-Beziehung von anderen Bauteilen unterbunden. Vereinfacht die Testerstellung.
2. Kein vollständiger Ersatz. Digitaler ICT ist ein Test unter Spannung. Mindestens zur Kontrolle und Beseitigung von Kurzschlüssen vor Anlegen der Spannung sind Zweipunktmessung zwischen den Leitungen erforderlich.
3. Bei Boundary-Scan erfolgt der Lese- und Schreibzugriff der logischen Pegel der Leitungen auf einer Baugruppe mit integrierten Teststrukturen (Schieberegisterringen) statt der Kontaktierung mit einem Nadeladapter.

Aufgabe 4.8: Boundary-Scan

1. Mit welcher Bitfolge an TMS lässt sich der TAP-Controller aus einem beliebigen Zustand in den Zustand »normale Funktion« versetzen?
2. Mit welcher Zustands- und Bitfolge an TMS und TDI lässt sich ausgehend vom Zustand »normale Funktion« das 8-Bit-Befehlswort 0x4D und anschließend der Zustand »Testfunktion laut Befehl« einstellen?



Zur Kontrolle

1. TMS-Folge, um den TAP-Controller aus einem beliebigen Zustand in den Zustand »normale Funktion« zu versetzen: 5-mal '1'.
2. Zustands- und Bitfolge an TMS und TDI zur Einstellung des Befehlswort 0x4D und des Zustand »Testfunktion lt. Befehl«:

Takt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zustand	A	B	C	D	E	F	F	F	F	F	F	F	G	I	J	B
TMS	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0
TDI	-	-	-	-	-	1	0	1	1	0	0	1	0	-	-	-