Universität des Saarlandes
Lehrstuhl für Softwaretechnik
Prof. Andreas Zeller, Holger Cleve

10. Übung zu Softwaretechnik 2

Qualitätssicherung, Funktionale Tests

1 Qualitätssicherung

1.1 TQM vs. ISO 9000-3 vs. CMM

Welche Rolle spielen die folgenden Konzepte in derISO 9000-Norm, in CMM und in TQM?

a) Qualität (wovon?) ist oberstes Ziel

b) Produkt- und Prozeßmessungen zur Verbesserung der Qualität des Produktes/des Prozesses

c) Internes Kunden-Lieferanten-Verhältnis

d) ganzheitliches Qualitätskonzept

e) Alle Mitarbeiter sind zuständig (wofür?)

f) Das Qualitätssicherungssystem ist integraler Bestandteil der Organisation.

g) Ständige Verbesserung

h) Zielsetzung sind Projekte, weniger Produkte

i) Kundenorientierung

j) Auftraggeber-Lieferanten-Verhältnis

k) Definierte Abnahme von Zwischenprodukten

l) Geschäftsführung ist für die Qualität verantwortlich

m) Prozeßorientierung

n) Festgelegte Verantwortlichkeiten und Befugnisse aller Mitarbeiter in der Qualitätssicherung

2 Funktionale Testmetriken

2.1 Funktionale Testfälle einordnen

Für ein Programm zur Berechnung des grössten gemeinsamen Teilers zweier Zahlen grösser Null werden funktionale Testfälle aufgestellt. Das Programm hat die Signatur `int ggT (int a, int b);` Welche funktionalen Testverfahren wurden in den folgenden vier Beispielen verwendet? Wie kam die Auswahl der Testdaten zustande?

a)

a	b
2	4
-2	4
2	-4
2^{100}	4
2	2^{100}

b)

a	b
0	0

c)

a	b
1	1
`INT_MAX`	`INT_MAX`
2	0
0	4
`INT_MAX`+1	2
2	`INT_MAX`+1

d)

a	b
2	4
5	7
12	32
22	456

2.2 Äquivalenzklassen

Ein Programm zur Berechnung von Preisen soll getestet werden. Das Programm erfasst die Artikelnummer, die Stückzahl und einen Rabatt.

- Die Artikelnummer ist eine 5-stellige Zahl.
- Die Stückzahl ist auf 10.000 begrenzt und muß mindestens gleich 1 sein.
- Der Rabatt liegt zwischen 0 und 100%.

Stellen Sie alle Äquivalenzklassen auf. Bilden Sie Testfälle mit Hilfe der Grenzwertanalyse.

2.3 Mutationstesten

a) Was sind die Ziele, die Vor- und die Nachteile von Mutationstesten?

b) Gegeben sei ein beliebiges grosses Programm. Sie führen 2 Mutationen ein und finden Sie beide mit ihrer Testsuite von 5000 Tests. Diese Testsuite deckt ausserdem 35 weitere Fehler in Ihrem Programm auf. Was können sie daraus schliessen?

c) Sie führen in das gleiche Programm 5000 Mutationen ein, von denen 4735 von der Testsuite aufgedeckt werden. Wieviele Fehler vermuten Sie jetzt in Ihrem Code?

d) Was vermuten Sie, wenn 4999 Mutanten von der Testsuite gefunden werden?

e) Ihr Kollege ist es leid, die Mutationen von Hand in den Code einzufügen. Stattdessen schlägt er vor, einfach die letzte Revision der Software als das mutierte Programm zu verwenden. Warum ist das eine schlechte Idee? Warum sollten Sie ein automatisches Tool zum Mutieren Ihres Programms besorgen oder schreiben?

Fragen?

Bei Fragen wenden sie sich an Holger Cleve `<cleve@cs.uni-sb.de>`.