



Software Engineering

Besprechung zur Uebung 6

WS 06/07



Agenda

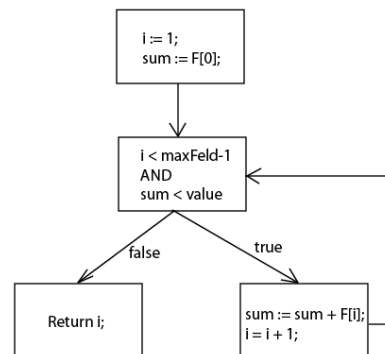
- Besprechung der Uebung 6.
- Besprechung der Musterklausur in Software Engineering (Bachelor).

2.1. Funktionsorientiertes Testen (1)

a.) White Box Test

- 100% Anweisungsüberdeckung
 - Alle Anweisungen müssen einmal durchlaufen werden.
 - Beispiel: [3, 4, 2, 5, 7, 8, 6, 5, 6, 5]
 - Indizes: 0 1 2 3 4 5 6 7 8 9
 3+4+2+5+7+8 = 29
 - Rückgabewert: 6, 7tes Element.
- 100% Zweigüberdeckung
 - --> selbes Beispiel.

2 P je korrekte Antwort + Bsp.



2.1. Funktionsorientiertes Testen (2)

b.) Black Box Test

- i) Äquivalenzklassen
 - Als repräsentative Menge von Eingabedaten.
 - Je Äquivalenzklasse wird ein Repräsentant getestet.
 - Annahme: *Array variabler Länge (1...10)*:
 - Äquivalenzklasse 1: Array mit genau einem Element.
 - Äquivalenzklasse 2: Array mit genau zwei Elementen.
 - Äquivalenzklasse 3: Array mit mehr als zwei Elementen.
 - Annahme: *Array fixer Länge (10)*:
 - Äquivalenzklasse 4: positive Werte der Elemente.
 - Äquivalenzklasse 5: negative Werte der Elemente.
 - Äquivalenzklasse 6: positive und negative Werte der Elemente



2.1. Funktionsorientiertes Testen (3)

b.) Black Box Test

- ii) Grenzfälle
 - Fehler an den Grenzen der Dateneingabe kommen oft vor, daher werden auch hier spezielle Testfälle definiert.
 - Annahme: *Array variabler Länge (1...10)*:
 - Grenzfall 1: Leeres Array mit Länge 0.
 - Grenzfall 2: Array mit Länge 10.
 - Annahme: *Array fixer Länge (10)*:
 - Grenzfälle:
 - Gesuchtes Element bereits an erster Stelle erreicht ($F[0] \geq \text{value}$).
 - Gesuchtes Element an letzter Stelle erreicht.
 - Gesuchtes Element überhaupt nicht vorhanden (Summe aller Werte $< \text{value}$)



2.1. Funktionsorientiertes Testen (4)

b.) Black Box Test

- iii) Den enthaltenen Fehler in SumUpUntilValue finden.
 - Ein beispielhafter Testfall:
 - Beispiel: `[1,1,1,1,1,1,1,1,1,21]`
Indizes: `0 1 2 3 4 5 6 7 8 9`
 `1+1+1+1+1+1+1+1+1=9`
 `9+21=30; 30>28;`
 - Falls die Summe den Wert *value* nicht erreicht soll *maxFeld* als Rückgabe geliefert werden. ($\text{maxFeld} = 10$)
 - Die Abbruchbedingung lautet:
 - `WHILE (i < maxFeld-1) AND (sum < value)`
 - Es wird nach dieser Abbruchbedingung ein Element zu früh abgebrochen.
 - Änderung auf `i <= maxFeld-1` oder auf `i < maxFeld`.



2.1. Funktionsorientiertes Testen (5)

C.) Fehlerarten beim White Box Test

- Leicht erkennbar
 - Falsch implementierte Funktionalitäten
 - Anweisungsüberdeckung: unerreichbarer Programmcode (engl. dead code)
 - Fehlende Zweige, fehlende Anweisungen (Programmierfehler)
- Schwer erkennbar
 - Spezifikationsfehler (durch BB Test leichter erkennbar)
 - 100% Pfadabdeckung (falsch formulierte Bedingungsterme)
 - Falsche Kommentare, ...



2.2. Zielorientiertes Messen, GQM (1)

a.) Ziel: «Problemloses und schnelles Online Check-in für jeden Kunden»

Faktoren:

- (1) Mehrsprachigkeit
- (2) Intuitivität der Benutzerschnittstelle (Benutzerfreundlichkeit)
- (3) Unkompliziertes und ansprechendes Design der Benutzerschnittstelle
- (4) Einfachheit der Integration des Punktesystems (soll die normale Benutzbarkeit nicht negativ beeinflussen)
- (5) Antwortzeiten des PinkAir Servers
- (6) ...

=> Questions:

- (1) Werden für die verschiedenen Nationalitäten unserer Kunden alle Muttersprachen unterstützt?
- (2) Ist die Benutzerschnittstelle so gestaltet, dass der Kunde auf Anhieb findet wonach er sucht?
- (3) Ist das Design so gestaltet, dass sich der Kunde gut aufgehoben fühlt?
- (4) Ist das Punktesystem gut integriert gestaltet und behindert es auch nicht den normalen Check-in?
- (5) Reagiert der Server in zumutbaren Antwortzeiten?
- (6) ...



2.2. Zielorientiertes Messen, GQM (2)

b.) Metrics:

- (1.1) Anzahl verschiedener Nationalitäten und Muttersprachen. *Nominalskala*
- (1.2) Anzahl bisher nicht unterstützter Muttersprachen aus allen Kunden in Prozent. *Verhältnisskala*

- (2.1) Anzahl Anrufe eines Kunden bis er die Information erhält. *Absolutskala*

- (3.1) Übereinstimmung von Vorstellungen des Kunden und Benutzerschnittstelle (empirische Werte). *zB 5-wertige Ordinalskala*
- (3.2) Zufriedenheit des Kunden mit dem Portal (empirische Bewertung, bewertet von sehr zufrieden bis sehr unzufrieden in 5 Gruppen). *Ordinalskala*

- (4.1) Mittlere Anzahl Reklamationen pro Kunde pro Monat, *Verhältnisskala*
- (4.2) Aufwand/Nutzen-Verhältnis des Onlinedienstes gemäß Kundenumfrage auf *fünfwertiger Ordinalskala*

- (5.1) Durchschnittliche Latenzzeiten des Servers. *Absolutskala*
- (5.2) Durchschnittliche Latenzzeit zu Stosszeiten (höchste Zahl der Zugriffe...). *Absolutskala*
- (5.3) Prozentuelle Anzahl an ungewollten Abbrüchen des Online Check-in. *Verhältnisskala*