LU11b - Blackbox Tests

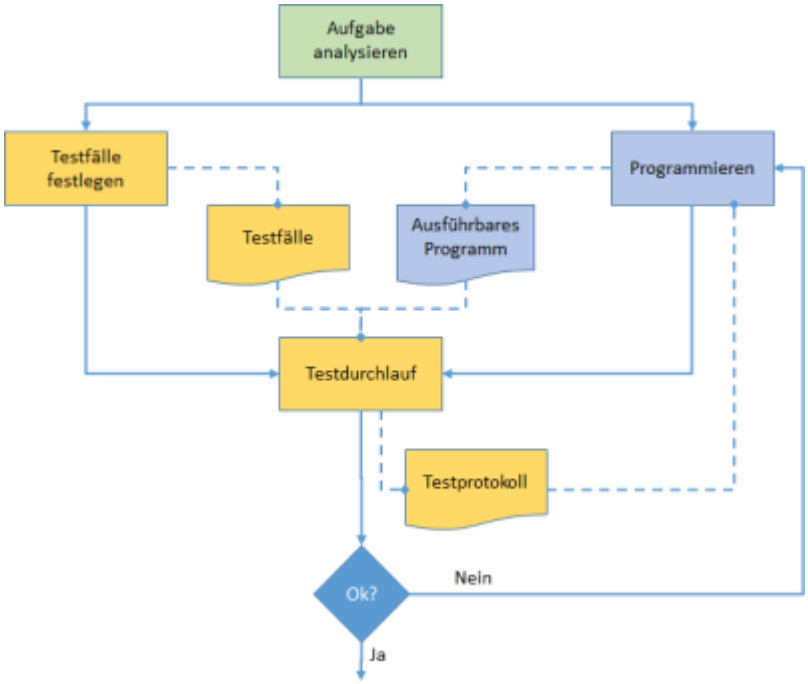
Black-Box-Test bezeichnet eine Methode zum Testen von Software. Die Tests werden anhand der Spezifikation/Anforderung entwickelt.

Bei einem Blackbox-Test werden Testfälle aufgrund der Anforderungen erstellt. Der Aufbau des Sourcecodes wird dabei nicht beachtet. Das Programm wird als Blackbox betrachtet, die auf einen bestimmten Input einen erwarteten Output liefern soll.

Als Basis für die Testfälle dienen uns

- Spezifikation der Applikation
- Zustandsdiagramme

Ablauf



1. In der Analyse werden die Anforderungen an das Programm ermittelt. Diese Anforderungen dienen als Grundlage zum Erstellen der Testfälle und des Programms.
2. Der Tester definiert die Testdaten (Dateien, Datenbanken, ...) mit welchen die Tests durchgeführt werden.
3. Nachdem der Programmierer den Sourcecode erstellt hat, wird das Programm nun systematisch getestet. In kleineren Projekten kann es sein, dass Programmierer und Tester die gleiche Person sind. Bessere Resultate erzielt man aber oft, wenn eine zweite Person die Testfälle durchführt.
4. Dazu benötigt der Tester die schon früher erstellten Testfälle. Beim Testen starten Sie das Programm und geben dem Programm die festgelegten Testdaten ein.
5. Ein Testdurchlauf besteht darin, dass Sie jeden Testfall einmal durchführen. Für jeden Testfall halten Sie das tatsächliche Ergebnis in einem Testprotokoll fest.
6. Sind alle Testfälle durchgeführt, entscheidet der Tester:

    - das Programm muss noch einmal überarbeitet werden
    - das Programm ist in Ordnung
7. Muss das Programm überarbeitet werden, so geht es zurück an den Programmierer. Dieser wird die Verbesserungen vornehmen und die neue Programmversion wieder an den Tester übergeben. Nun führt der Tester erneut **alle** Testfälle durch und hält die Resultate in einem **neuen** Testprotokoll fest.

Testfall

Um eine Software systematisch zu testen, müssen die Testfälle reproduzierbar sein. Ein Testfall legt fest:

- welche Situation wir testen wollen,
- mit welchen konkreten Daten getestet wird,
- welches Resultat wir erwarten.

Zu einem Testfall gehören die Angaben:

- Nr: Eine eindeutige Nummer hilft, den Testfall zu identifizieren.
- Bezeichnung: Was wird getestet.
- Input: Mit welchen Eingaben bzw. Daten wird der Test ausgeführt.
- Erwartetes Resultat: Welche Resultate und Meldungen sollte das Programm ausgeben.

Darstellung

Um den Aufbau der Testfälle zu demonstrieren, zeige ich dies an einem konkreten Beispielprogramm.

Das Programm berechnet eine Seite eines rechtwinkligen Dreiecks.

- Der Benutzer gibt die Länge von 2 der 3 Seiten ein.
- Für die dritte, unbekannte Seite gibt der Benutzer -1 ein.
- Das Programm zeigt dann die Länge der dritten Seite an.

Tabellarische Darstellung

Nr	Testfall	Input			Erwartetes Resultat
		Seite a	Seite b	Seite c	
1	Berechnung der Seite c	3	4	-1	Seite c: 5
2	Berechnung der Seite a	-1	4	5	Seite a: 3
3	Seitenlänge a ist 0	0	15	7	Eine der Seiten hat die Länge 0: Berechnung nicht möglich
4	Alle Seiten angegeben	3	4	5	Alle drei Seiten wurden angegeben: Berechnung nicht möglich

Jede Zeile enthält die Angaben zu einem Testfall. Diese Form ist sehr kompakt und für einfache Programme gut geeignet.

Testfall einzeln darstellen

Nr		1
Testfall		Berechnung der Seite c
Input	**Seite a**	3
	Seite b	4
	Seite c	-1
Erwartetes Resultat		Seite c: 5
Nr		2
Testfall		Berechnung der Seite a
Input	**Seite a**	-1
	Seite b	4
	Seite c	5
Erwartetes Resultat		Seite a: 3
Nr		3
Testfall		Seitenlänge a ist 0
Input	**Seite a**	0
	Seite b	15
	Seite c	7
Erwartetes Resultat		Eine der Seiten hat die Länge 0: Berechnung nicht möglich
Nr		4
Testfall		Alle Seiten angegeben
Input	**Seite a**	3
	Seite b	4
	Seite c	5
Erwartetes Resultat		Alle drei Seiten wurden angegeben: Berechnung nicht möglich

Diese Form ist weniger kompakt und eher für Programme mit komplexem Input/Output geeignet.

Testprotkoll

Sobald das Programm fertig ist, führen wir einen Testdurchlauf mit allen Testfälle durch. Das Testprotokoll hält die Ergebnisse aus einem Testdurchlauf fest.

In der Regel werden wir beim Testen noch Fehler entdecken. Diese Fehler kann der Programmierer **nach** dem Testdurchlauf anhand des Testprotokolls korrigieren. Anschliessend führen wir einen neuen Testdurchlauf mit allen Testfällen durch und erstellen ein **neues** Testprotokoll.

Ein Testprotokoll enthält die Angaben:

- Name und Version des getesteten Programmes.
- Name der Person welche die Testfälle durchgeführt hat.
- Datum und Uhrzeit des Testdurchlaufs.
- Pro durchgeführtem Testfall eine Zeile mit
    - Nr des Testfalls.
    - Tatsächliches Resultat.
    - Bewertung des Resultats (Korrekt, Hauptfehler oder Nebenfehler)

    - Bemerkungen
- Fazit: Wie soll weiter vorgegangen werden. Zum Beispiel: Das Programm muss überarbeitet und erneut getestet werden.

Aufbau

Programm	Dreieckseite berechnen	**Version**	1.2
Datum	15.10.2014	**Uhrzeit**	15:00
Tester	Marcel Suter		
Nr	**Effektives Resultat**	**Bewertung**	**Bemerkung**
1	Seite c: 5.00	N	Nachkommastellen überflüssig
2	Seite a: 2.9998	H	Resultat ungenau
3	Absturz	H	Jede Seite mit Länge 0 führt zu einem Absturz.
4	Alle drei Seiten wurden angegeben: Berechnung nicht möglich	OK	
Fazit	Das Programm funktioniert mit bestimmten Eingaben nicht korrekt. Es muss überarbeitet und neu getestet werden.		

Beschreibung

Bewertung

OK	Das effektive Resultat ist identisch mit dem erwarteten Resultat.
H	Hauptfehler: Die Funktion des Programms ist nicht korrekt. Der Fehler muss korrigiert werden.
N	Nebenfehler: Das Programm arbeitet korrekt. Die Anzeige enthält Schönheitsfehler. Das Programm sollte bei Gelegenheit überarbeitet werden.

Bemerkung

Im Feld „Bemerkung“ können zusätzliche Angaben zu einem Fehler gemacht werden. Diese Bemerkungen können dem Programmierer helfen, den Fehler zu lokalisieren.

Vielleicht erkennen Sie beim Testen, dass ein Fehler nur bei bestimmten Werten auftritt.

Beispiel: „Rundungsfehler treten nur bei Seitenlängen > 50 auf.“

Das Bemerkungsfeld soll nur Beobachtungen des Testers enthalten. Vermeiden Sie es, dem Programmierer die Lösung auf's Auge zu drücken.

Fazit

Das Fazit fasst die Resultate zusammen und enthält die Empfehlung des Testers zum weiteren Vorgehen.

- Das Programm kann freigegeben werden
- Das Programm muss korrigiert werden. Ein erneuter Test ist nicht notwendig.
- Das Programm muss überarbeitet und erneut getestet werden.

M431-LU11

Marcel Suter, Kevin Maurizi

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/m431/learningunits/lu11/blackbox

Last update: **2025/12/01 06:24**