Testkonzept

1. Einführung

Das Testen von Software ist einer der wichtigsten Bestandteile des Softwareentwicklungsprozesses. Der Testprozess soll die Korrektheit der Software nachweisen und sicherstellen. Durch genaue Testfälle werden Abweichungen des lauffähigen Produkts von der Spezifikation systematisch aufgedeckt und Fehler in der Implementierung, aber auch in anderen Dokumenten lokalisiert. Deshalb ist die Durchführung einer gut durchdachten Testphase für den Erfolg eines Projeckts entscheidend.

2. Testkonzept

J-Unit-Test bietet Werkzeuge, um Tests für Java-Quellcode zu implementieren. Die wichtigste Klasse des J-Unit-Frameworks ist die Klasse TestCase, Welche die Funktionalität zur Implementierung und zur Ausführung der Tests anbietet. Durch Implementierung von TestCase und Hinzufügen der Testmethoden, kann die Software getestet werden. Um einen Test auszuführen, wird der TestCase einer TestSuite hinzugefügt und die Methode TestSuite.run() aufgerufen. Die TestSuite enthält mehrere Testfälle, welche alle ausgeführt werden.Die Umgebungsbedingungen für die Funktionalität sind dabei so gewählt, dass die Normalfälle und Randfälle enthalten sind, außerdem müssen alle sonstigen (ungültigen) Fälle abgefangen und nicht weiter verarbeitet werden. Dabei unterscheidet man zwischen

- Komponenten-,
- Integrations-,
- System-,
- Komponententests

Komponententests bezeichnen den isolierten Test einer Programmeinheit. Hierzu wird ein eigens für die Einheit bereitgestelltes Testgeschirr benötigt, das die Einheit über ihre echten Schnittstellen mit Eingaben versorgt, die Ausgaben entgegennimmt und für die Bewertung des Tests eventuell ausgibt. Bei einem Unit-Test handelt es sich um ein Stückchen Quellcode, das der Programmierer zum Testen eines sehr kleinen und klar umrissenen Teiles der Funktionalität der Software geschrieben hat. Mit der erfolgreichen Ausführung des Unit-Tests weist der Entwickler dessen korrekte Funktionalität nach. Die Testklassen werden vor der Implementierung der eigentlichen Klasse erstellt, die Funktionalität der Klasse prädefiniert. Jede Methode der Klasse muss in der Testklasse durch einen vorangestellten Test im Methodennamen implementiert werden.
Ein typischer Testfall besteht aus drei Schritten:

- Testobjekt erzeugen
- Methode vom Testobjekt ausführen

- Ergebnisse überprüfen

Integrationstest

Der Integrationstest ist die Erweitung des Komponententest. Es wird nach Fehlern in der Kommunikation und Zusammenspiel der Komponenten gesucht. Aus zwei getesteten Komponenten wird eine Gruppe gebildet – nun wird also das Interface der Komponenten getestet.

Systemtest

Das Ziel des Systemtestes ist die Überprüfung des Gesamtsystemes auf Erfüllung sowohl von funktionalen wie auch nicht-funktionalen Anforderungen geprüft. Die wichtigste funktinale Anforderung ist sicherlich die Richtigkeit der errechneten Ergebnisse.

Abnahmetest

Der Abnahmetest ist der Test für das Endprodukt, er wird vom Auftraggeber durchgeführt. Im Abnahmetest prüft der Auftraggeber das erstelllte Produkt auf vertragliche Akzeptanz und Benutzerakzeptanz.

3. Testfälle

Test 1: Anzeigen einer Grafischen Oberfläche

Verlauf: Ein Benutzer erstellt eine neue grafische Arbeitsfläche und benennt oder umbenennt die grafische Oberfläche.

Ergebnis: Es wird eine grafische Oberfläche erstellt und ein Name zugewiesen oder geändert.

Test 2: Anzeigen grafischer Objekte

Verlauf: Ein Benutzer erstellt grafische Objekte, benennt oder umbenennt die grafischen Objekte, ändert die Größe oder die Farbe oder löscht die Objekte.

Ergebnis: Grafische Objekte werden erstellt, ein Name für jedes Objekt zugewiesen oder geändert, die Größe oder die Farbe geändert, oder grafische Objekte gelöscht.

Test 3: Hinzufügen oder entfernen bei Objekt

Verlauf: Ein Benutzer fügt ein Attribut oder eine Methode hinzu oder entfernt diese von einem Objekt.

Ergebnis: Ein Attribut oder eine Methode von einem Objekt wird hinzufügt oder entfernt.

Test 4: Anzeigen der Verknüpfung grafischer Objekte

Verlauf: Ein Benutzer verknüpft vorhandene Objekte, benennt oder umbenennt die Verknüpfung, ändert die Verknüpfungsart oder die Linienfarbe von einer Verbindung, löscht eine Verknüpfung.

Ergebnis: Vorhandene Objekte werden verknüpft. Die Verknüpfungen werden benannt oder umbenannt. Die Verknüpfungsart oder die Linienfarbe von einer Verbindung wird geändert. Eine Verknüpfung wird gelöscht.

Test 5: Hinzufügen oder entfernen bei Verbindung

Verlauf: Ein Benutzer fügt ein Attribut zu einer Verknüpfung hinzu.

Ergebnis: Ein Attribut wird von einer Verbindung hinzufügt oder entfernt.

Test 6: Speichern, Löschen oder Laden eines Modells

Verlauf: Ein Benutzer speichert oder löscht ein erstelltes Modell, oder lädt ein vorhandenes, abgespeichertes Modell.

Ergebnis: Ein Modell wird gespeichert, gelöscht oder geladen.

Test 7: Übersetzung eines Modells in GMF-Daten

Verlauf: Ein Benutzer erstellt ein rohes Modell und übersetzt das Modell in GMF-Daten.

Ergebnis: Ein rohes Modell auf der grafischen Oberfläche wird in ein für GMF lesbares Datenformat übersetzt.