

Testkonzept

1. Einleitung:

Ein Softwaretest dient der Qualitätssicherung einer neu erstellten oder geänderten Software. Dabei geht es darum, das tatsächliche Verhalten mittels Testfall und gemäß eines Testplanes zu untersuchen und die Ergebnisse mit den erwarteten zu vergleichen.

Es gibt entweder automatisierte oder manuelle Tests. In diesem Projekt werden automatisierte Tests auf solche beschränkt, die sich mit dem JUnit – Framework umsetzen lassen. Die View wird manuell geprüft.

2. Vorgehensweise

2.1 Komponententests

Zum Testen wird das JUnit Framework 4.0 genutzt. Dabei sollen die von JUnit vorgeschlagenen Vorgaben eingehalten werden, insbesondere wird festgelegt:

- zu jeder Klasse wird ein TestCase mit dem Namen `Test<NameDerZuTestendenKlasse>` im selben Paket erstellt
- zu jeder zu prüfenden Methode wird eine Methode mit dem Namen `test<NameDerZuTestendenMethode>` erstellt
- zu überprüfende Werte werden mit `assertTrue()` getestet. Es sollen auch mutwillig Fehler erzeugt werden, um das Werfen von Exceptions zu prüfen
- für häufig verwendete Variablen, die mit bestimmten Startparametern versehen werden müssen, werden mithilfe von `setUp` und `tearDown` Methoden Text-fixtures eingerichtet.
- in den einzelnen TestCases werden nur klassenabhängige Funktionen untersucht, für integrative Test werden separate TestCases geschrieben

2.2 TestSuites

- Klassen werden erst in das TestSuite des Paketes übernommen, sofern sie ihr `TestCase` erfolgreich absolviert haben
- die TestSuite für ein Paket wird mit `AllTest` bezeichnet, somit greift man mittels `package.AllTest` auf den Test eines gesamten Paketes zu.
- die TestSuite für ein Paket wird in regelmäßigen Abständen, vor allem für integrative Tests oder bei Änderungen, die Auswirkungen auf andere Klassen haben könnten, benutzt

2.3 Systemtests

- alle Pakete zusammen werden im Systemtest noch einmal überprüft. Dabei wird das komplett integrierte System unter funktionalen und nicht-funktionalen Gesichtspunkten getestet.
 - der funktionale Systemtest beinhaltet die Überprüfung der funktionalen Qualitätsmerkmale Korrektheit und Vollständigkeit
 - der nicht-funktionale Systemtest beinhaltet Qualitätsmerkmale, wie z.B. die Sicherheit, die Dokumentation oder die Zuverlässigkeit eines Systems

3. Dokumentation

Die Dokumentation der einzelnen Tests übernehmen die jeweiligen Testgruppen. Es werden zunächst alle nötigen Tests notiert. Alle abgearbeiteten Punkte werden festgehalten. Wenn Fehler aufgetreten sind, werden die notwendige Änderungen bestimmt. Noch offene Tests werden auch markiert. Wenn aufgrund von Fehlern Änderungen am Fachkonzept nötig sein sollten, muss der Projektleiter hinzugezogen werden.

4. Testverantwortliche

Für den Test einzelner Klassen ist grundsätzlich der Implementierer der jeweiligen Klasse selbst zuständig. Für die einzelnen TestSuites gibt es jeweils einen Verantwortlichen:

Package editor
Heiko Fischer

Package modul
Mohamed Bary

Package komponent
Hai Long

Systemtest
Mohamed Bary