

Testkonzept

JUnit-Tests

Tests dienen der Überprüfung der implementierten Funktionen und sind Grundlage der Qualitätssicherung dieses Projektes. Die Umsetzung einer Klasse beginnt grundsätzlich mit der Implementierung von Tests mit JUnit 4. Im folgenden gibt es einen ständigen Wechsel von Test- und eigentlicher Programmierarbeit. Folgende Regeln sind dabei zu befolgen:

- Die Tests sind jeweils ca. eine Woche vor der entsprechenden Story zu implementieren, dabei sollten Implementierer der Test und der Story unterschiedlich sein.
- Zu jeder Story ist dabei eine eigene Test-Klasse anzulegen.
- Auch die Test-Klassen und Methoden sind dem Dokumentationskonzept folgend mit Kommentaren zu versehen. Darin erwähnt wird was konkret getestet wird.
- In den Tests soll die zu testende Funktion nicht selbst wieder implementiert werden, sondern anhand von geeigneter Beispielen und insbesondere Randfällen geprüft werden.
- Das Werfen von Exceptions ist mit passenden Fehleingaben zu überprüfen. Der Test gilt als erfolgreich, wenn die richtige Exception geworfen wurde.
- Sollte während der Implementierung die Notwendigkeit vorher nicht geplanter Klassenfunktionen festgestellt werden, sind für diese entsprechende Tests in der passenden Testklasse zu ergänzen und durch einen Kommentar entsprechend zu markieren.
- Eine Klasse gilt erst als fertig implementiert, wenn die entsprechenden Tests fehlerfrei durchgeführt wurden.
- Für jedes Release ist die zugehörigen TestSuite zu veröffentlichen, die alle Tests für die implementierten Funktionen und Klassen beinhaltet.
- Auch die TestSuite ist gemäß dem Dokumentationskonzept richtig zu versionieren.
- Ein Release muss alle Tests seiner TestSuite fehlerfrei durchlaufen, bevor es freigegeben wird.

Die TestSuite wird nach folgendem Schema erstellt:

```
import junit.framework.*;

public class AllTests
{
    public static Test suite()
    {
        TestSuite suite = new TestSuite();
        suite.addTestSuite(Test1.class);
        suite.addTestSuite(Test2.class);
        //... weitere Klassen
        return suite;
    }
}
```

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

Endbenutzertests

Die JUnit-Tests sichern, dass die Funktionen der Klassen korrekt arbeiten, können aber nur bedingt Auskunft darüber geben, ob das Projekt in seiner Portletumgebung für den Endbenutzer tatsächlich wie gewünscht arbeitet. Einige Fehler können nicht durch automatische Tools festgestellt werden, sondern bedürfen manueller Überprüfung. Dazu begeben sich die Tester in die Rolle des Endbenutzers und testen vor der Freigabe eines Releases alle Funktionen des fertigen Portlets. Dabei sind sowohl richtige als auch Eingaben, die zu Fehlern führen zu testen. Insbesondere ist auf Randbedingungen zu achten. Bsp.: Bei Eingabe des Namen mit auf 12 Zeichen begrenzter Länge sind sowohl Eingaben mit weniger als 12 Zeichen, wie auch einige mit mehr als 12 Zeichen zu überprüfen. Besonders wichtig sind hier die Randbedingungen 0 Zeichen, 11, 12 und 13 Zeichen. Außerdem ist auf die Stabilität des Systems zu achten. Für jede Story ist neben der TestSuite auch ein Testprotokoll im Voraus anzulegen. Darin sollen die zu testenden Funktionen tabellarisch vorgefertigt werden. Die Testergebnisse sind darin zu protokollieren und eventuelle Fehlermeldungen zu vermerken. Das Release wird erst freigegeben, wenn das gesamte Testprotokoll fehlerfrei abgearbeitet wurde.

Das Testprotokoll wird nach folgendem Schema erstellt:

<i>Durchzuführender Test</i>	<i>Erwartetes Ergebnis</i>	<i>OK</i>
<i>Eingabe von Namen bei Anmeldung mit 0, 3 und 12 Zeichen</i>	<i>Eingabe akzeptiert</i>	
<i>Kommentar</i>		
<i>Eingabe von Namen bei Anmeldung mit 13, 24 und 40 Zeichen</i>	<i>Fehler: Name zu lang</i>	
<i>Kommentar</i>		
...	...	
<i>Kommentar</i>		