

Qualitätssicherungs- konzept

swp13-olat

Version 2.0 – 15.04.2013

**Melanie Mendikowski
Boie Thomsen
Klemens Schölhorn**

Inhaltsverzeichnis

Dokumentationskonzept.....	2
Dokumentation des Quelltextes.....	2
Formalismen und Kommentare.....	3
Namensgebung.....	3
Änderungsdokumentation.....	3
Testdokumentation.....	4
Testkonzept.....	4
Klassen- und Modultests.....	4
Oberflächentests.....	4
Organisatorische Festlegungen.....	5

Dokumentationskonzept

Das Dokumentationskonzept beschreibt die Vorgehensweise bei der Dokumentation des Quelltextes und anderer anfallender Aufgaben. Eine gute Dokumentation erhöht die Nachvollziehbarkeit und erleichtert es, die Qualität des Produktes sicherzustellen.

Dokumentation des Quelltextes

Grundsätzlich sollte der Quelltext den *Java Code Conventions*¹ folgen. In diesem Dokument werden jedoch noch einige für unser Projekt relevante Ergänzungen dargelegt.

1 <http://www.oracle.com/technetwork/java/codeconv-138413.html>

Formalisten und Kommentare

- Alle Einrückungen müssen 4 Zeichen breit sein und durch Leerzeichen erfolgen.
- Die Anzahl Zeichen in einer Zeile sollte 120 nicht überschreiten und bei eventuell notwendigen Umbrüchen sollten die zusätzlichen Zeilen sinnvoll eingerückt werden.
- Alle Kommentare sind in Englisch zu verfassen
- Jede Klasse ist mit einer kurzen aber aussagekräftigen Beschreibung ihres Zweckes und eventueller Details zur Verwendung in Form eines JavaDoc Kommentars zu versehen.
- Jede Methode muss mit einem vollständigen JavaDoc Block beschrieben sein
- Falls der Zweck einer Anweisung nicht unmittelbar aus der Syntax klar ist, muss dieser direkt vor oder - wenn genug Platz vorhanden ist – neben der Anweisung durch einen Kommentar beschrieben werden.
- Bei Auswahlanweisungen (if, else und switch) ist jede Verzweigung entsprechend zu kommentieren.

Namensgebung

- Alle Bezeichner sind in Englisch zu verfassen und müssen sprechend sein.
- Alle Bezeichner sind in CamelCase zu verfassen (d.h. es wird jeweils der erste Buchstabe der dem ersten folgenden Wörter groß geschrieben: *meineTolleVariable*), wobei bei Klassennamen auch der erste Buchstabe groß geschrieben wird.
- Konstanten sind ausschließlich aus Großbuchstaben und dem Unterstrich aufzubauen.

Änderungsdokumentation

Jeder Commit für ein Repository muss mit einer aussagekräftigen Beschreibung darüber versehen sein, was hinzugefügt oder geändert wurde. Außerdem sollte

nur funktionierender Quelltext committed werden, d.h. er sollte unbedingt kompilieren und wenn möglich keine UnitTests verletzen.

Testdokumentation

Damit nach Abschluss der Entwicklungsarbeiten belegt werden kann, dass die Anwendung umfassend getestet wurde, werden zum Beispiel die durchgeführten Oberflächentests und deren Ergebnisse ebenfalls in einem entsprechenden Repository dokumentiert.

Testkonzept

Klassen- und Modultests

Für die Tests der einzelnen Klassen kommt wie bei OpenOLAT JUnit² zum Einsatz. Dabei werden für jede neue Klasse, aber auch für aus mehreren Klassen bestehende Module Tests geschrieben, die überprüfen, ob sich die Klasse oder das Modul bei gegebenen Eingabeparametern wie gewünscht verhalten. Diese Tests können anschließend alle automatisch durchgeführt werden

Diese UnitTests sollten für eine Klasse bereits vor oder während der Implementierung derselben auf Basis der Spezifikationen geschrieben werden, da damit bereits während der Entwicklung sichergestellt werden kann, dass die Spezifikationen eingehalten werden. Ein möglichst häufiges Durchführen der Tests stellt zudem sicher, dass vorhandener Code nicht durch Änderungen unbrauchbar wird.

Oberflächentests

Zum Testen der Oberfläche wird Selenium³ eingesetzt, das auch bei OpenOLAT Verwendung findet. Dank des verfügbaren Firefox Addons lassen sich Nutzerinteraktionen mit der Weboberfläche sehr einfach aufnehmen und später automatisch wiedergeben, wobei geprüft werden kann, ob die richtigen Seiten mit den richtigen Inhalten geladen werden.

2 <http://junit.org/>

3 <http://seleniumhq.org/>

Organisatorische Festlegungen

Es finden wöchentliche Teamtreffen statt, bei denen die Aufgaben verteilt und die Ergebnisse der vergangenen Woche vorgestellt und besprochen werden. Außerdem findet bei diesen Treffen auch - wenn nötig - eine Abstimmung mit dem Betreuer statt. Die Kommunikation außerhalb der Treffen erfolgt hauptsächlich über Skype. Alle einzureichenden Dokumenten werden kollaborativ in Pads erstellt, wobei dafür auf dem Projektserver eine Etherpad Lite Instanz allen Teammitgliedern zur Verfügung steht.

Die Dokumentation des Quelltextes ist Aufgabe des jeweiligen Autors, wobei die Einhaltung der in diesem Dokument festgelegten Richtlinien regelmäßig durch den Verantwortlichen für Dokumentation überprüft wird. Sämtliche Quelltexte werden durch das dezentrale Versionsverwaltungssystem git verwaltet, wobei auf dem Projektserver zentrale Repositories für die einfache Synchronisation zwischen den Mitgliedern zur Verfügung stehen.

Jedes Mitglied erstellt wöchentlich einen persönlichen Aufwandsbericht und lässt diesem dem Projektleiter zukommen, der die Berichte zusammenfasst und dem Betreuer übermittelt.