6. Aufgabenserie

Dokumentations- und Testkonzept

Dokumentationskonzept

Die Dokumentation des Programms ist ein wichtiger Aspekt der Qualitätssicherung. Sie erfolgt in zwei Etappen: Der internen Dokumentation und der JavaDoc. Im OLAT ist die Dokumentation relativ nachlässig vorgenommen wurden, hierauf muss im Projekt mehr Wert gelegt werden. Die interne Dokumentation erfolgt an den Stellen, wo Erklärungen erforderlich sind (unnötige oder nichtssagende Kommentare sind zu vermeiden). So können Kommentare an Variablen, vor Schleifen, Bedingungen und neuen Programmabschnitten stehen. *Wichtig ist, dass nicht nur dokumentiert wird, was getan wird, sondern vor allem, warum es getan wird.* Die eigentliche Dokumentation der Klassen und Methoden erfolgt in JavaDoc, wobei jede Klasse, jede Methode und jede Klassenvariable in JavaDoc zu dokumentieren ist, einschließlich der Verwendung der zur Verfügungen stehenden Tags. Bei Klassen wird der Tag @author verwendet, der den Autor/die Autoren angibt. Bei jeder Methode werden, sofern möglich, die Tags @param und @return verwendet, außerdem wird zuvor in einer Zeile die Aufgabe/Funktion der Methode angegeben.

Die Dokumente (JavaDoc-Dateien) müssen stets auf dem aktuellsten Stand und allen zugänglich sein. Nur somit kann gewährleistet werden, dass sich jeder, der an der Implementierung beteiligt ist, möglichst schnell einen Überblick über eine entsprechende Klasse, einschließlich ihrer Funktionalitäten, verschaffen kann, und damit die Voraussetzungen hat, diese zu erweitern oder zu verändern. *Die interne Dokumentierung erfolgt sofort, das heißt während des Programmierens und nicht erst danach.*

Wir verwenden die allgemeinen Regeln für die Implementierung, wie auch im Balzert festgelegt, also u.a.: *Verwendung einheitlicher, sprechender Namen (einheitlich heißt: Variablen in Kleinbuchstaben, Konstanten in Großbuchstaben, zusammengesetzte Wörter durch Unterstriche zusammenfügen etc.), Angabe zu jeder Variable, ob sie allgemein, statisch oder final ist, Einrücken des Codes (3 Zeichen pro Ebene), ausgiebige Verwendung von Leerzeilen zur Abtrennung zusammengehöriger Teile im Code.* Die Zeilenlänge sollte 80 Zeichen nicht überschreiten.
Nach jeder abgeschlossenen Programmierarbeit ist der Code im SVN-Repository hochzuladen; hierbei sind eine ausführliche Beschreibung der Neuerungen und Änderungen mit zu übertragen. Treten bei dem Commit Differenzen auf, so sind diese und deren Lösung explizit in den SVN-Kommentaren anzugeben.

Testkonzept

Testzyklus: Der Testablauf in unserem Projekt wird sich in 4 Phasen untergliedern.

- Der Komponententest
- Der Integrationstest

- Der Systemtest
- Der Abnahmetest

Der Komponententest:

Wie schon angesprochen wird das Hauptwerkzeug des Komponententests Junit sein. Durch das MVC Konzept von OLAT ist der Fokus der Testobjekte vor allem auf den Model - Klassen. Jede Klasse wird vorher auf ihre Spezifikation überprüft. Durch die Analyse wird sichergestellt, was ein erfolgreicher und was ein nicht erfolgreicher Testlauf ist. Diese Information hilft, den Junit-Test zu programmieren. Dies ist sehr hilfreich um z.B. nicht definierte Werte in Grenzbereichen oder Exceptions zu finden. Einfache Klassen, deren Funktionalitäten nicht so komplex sind, können mittels eines Testprogrammes in der Konsole überprüft werden.

Der Integrationstest:

Den Integrationstest kann man wieder in 2 Kategorien unterteilen. Zum einen wird getestet, ob Model, View und Controller wie geplant miteinander kommunizieren. Zum anderen kann überprüft werden, ob die Module auch im OLAT integriert funktionieren. Der OLAT-Debugger gibt dann Aufschluss darüber, welche Komponenten nicht richtig funktionieren. Werden die Komponenten nicht richtig eingebunden, gibt die Fehlernummer in OLAT Aufschluss darüber, wo die Ursache liegen könnte.

Der Systemtest

Es ist wichtig, zu überprüfen, ob die Anforderungen an das System erfüllt sind. Hierbei könnten verschiedene Testtechniken zum Zuge kommen - zum Beispiel Tests durch die Entwickler eines Moduls selbst oder Tests durch andere Teammitglieder, die das Modul nicht programmiert haben. Auch andere Black- und Whiteboxtests wären denkbar. Des Weiteren wäre die Kontrolle durch Projektunabhängiger Personen ein guter Hinweis auf die Qualität. Dies gibt nicht nur Aufschluss über die Richtigkeit der Funktionalität und Kommunikation der einzelnen Bestandteile, sondern auch ein Feedback über die Softwareergonomie und ob das System für einen Normalverbraucher selbsterklärend genug ist.

Der Abnahmetest

Der Abnahmetest schließlich bildet den Produktabschluss, hier wird der Kunde dem neuen System bekannt gemacht. Es wird ersichtlich, ob das Produkt seinen Vorstellungen entspricht und ob er mit der Umgebung zurecht kommt.

Fehler Dokumentation:

Alle Fehler, die auftreten, müssen dokumentiert werden. Erwartete Fehlerquellen müssen mittels Exceptions abgefangen werden, damit beim Testen die Ursache sofort ersichtlich wird. Ist der bekannte Fehler nicht dokumentiert, so muss man dies nachholen. Nachdem man die Fehler von mehreren Testläufen aufgelistet hat, muss man diese systematisch beseitigen. Wie schon erwähnt, gibt es OLAT-intern auch eine solche Fehlerdokumentation, der erzeugte Fehler wird durch eine Nummer referenziert.