

Qualitätssicherungskonzept

1. Dokumentationskonzept

Dieses Konzept ist für die Sicherung eines lesbaren Quellcodes und muss unbedingt eingehalten werden, da es die grundlegenden Bestimmungen bei der Programmierung beschreibt. Nur durch die Einhaltung dieser Grundlagen kann ein übersichtlicher, lesbarer und nachvollziehbarer Code entstehen.

Interne Dokumentation

Als erstes die Bestimmungen für die Bezeichner um Funktionen übersichtlich und selbst erklärend gestalten zu können:

- Die Namen von Methoden, Klassen und Variablen müssen selbsterklärend sein.
- Bei der Namensgebung von Methoden, Klassen und Variablen soll auch darauf geachtet werden bei einer Sprache zu bleiben. Um den Code allgemein lesbar zu machen ist dafür Englisch Pflicht
- Klassennamen beginnen immer mit einem Großbuchstaben, sollte der Name aus einem zusammengesetzten Wort bestehen so wird das zweite Wort direkt dahinter geschrieben und beginnt wieder mit einem Großbuchstaben
- Variablen werden immer klein geschrieben, sollte sie auch aus einem zusammengesetzten Wort bestehen so wird das zweite Wort wie bei den Klassennamen direkt dahinter geschrieben und beginnt mit einem Großbuchstaben
- Methodennamen werden wie Variablennamen immer klein geschrieben und jedes weitere Wort im Namen groß geschrieben
- Konstanten wiederum werden komplett groß geschrieben

Um die Form einzuhalten gibt es auch hier bestimmte Regeln:

- Die umschließenden Klammern einer Funktion beginnen immer nach dem Funktionsnamen und enden auf einer extra Zeile am Ende der Funktion. Der Funktionsinhalt wird immer eingerückt, sodass sofort ersichtlich ist was alles zu der Funktion gehört
- bei geschachtelten Methoden wird genauso vorgegangen
- Funktionsbefehle stehen immer auf einer eigenen Zeile, so kann kein Befehl beim durchschauen übersehen werden
- sollte ein Befehl länger sein als die Breite eines A4 Blattes, sodass die gesamte Zeile nicht auf einen Blick anschaubar ist ohne das man nach rechts scrollen muss, dann muss darauf geachtet werden diesen Befehl über zwei Zeilen zu schreiben und der Übersichtshalber die zweite Zeile eingerückt werden
- zwischen Methoden sollte immer eine Zeile frei gelassen werden um die Übersichtlichkeit zu erhöhen, auch wenn eine Methode in einer anderen drin steht muss darauf geachtet werden das über und unter der Methode eine Zeile frei gelassen wird

Umlaute dürfen nicht verwendet werden um mögliche Fehler dieser Herkunft auszuschließen.

Um den Quellcode nachvollziehbar zu machen muss der Quellcode auch gut kommentiert werden, d.h. , dass jede neue Methode eine kleine Erklärung darüber erhält was sie macht und wofür, wenn notwendig, sie gebraucht wird und auch das nicht leicht nachvollziehbare Befehle kommentiert werden.

Bei PHP-Code ist außerdem noch darauf zu achten das er immer mit „<?php“ eingeleitet wird um die Plattformunabhängigkeit zu gewährleisten. Ein abschließendes „?>“ ist bei der Programmierung für OntoWiki nicht notwendig und kann sogar zu Fehlern führen, somit ist dies wegzulassen.

Ein weiterer wichtiger Bestandteil der Dokumentation ist die PHPDoc. Diese ermöglicht es das alle Klassen in einer HTML-Übersicht dargestellt und kurz erläutert werden.

PHPDoc wird immer mit „/**“ eingeleitet und mit „*/“ abgeschlossen. Jede Zeile dazwischen erhält ein * am Anfang der Zeile. Üblicherweise fängt man mit den Bemerkungen erst in der zweiten Zeile des Blockes an. Dort gehört dann der Name, gefolgt von einer knappen Erklärung die mit in der Übersicht erscheint, der Klasse hin. Danach wird, der Übersichtshalber, eine Zeile frei gelassen. Nun kann man eine längere Erklärung der Methode niederschreiben, welche dann in der Beschreibung erscheint und nach einer weiteren leeren Zeile folgen die Angaben zur Codebeschreibung. Diese werden jeweils von einem Schlüsselwort eingeleitet und danach beschrieben, z.B.: @author Max Müller.

Bei der Bearbeitung des Codes ist es für alle Programmierer wichtig die Änderungen die von anderen Mitgliedern gemacht wurden zu erkennen. Also muss beim dem Abspeichern und hochladen ins SVN darauf geachtet werden das bei der Kommentierung der Änderung darauf geachtet wird auch wirklich alle Änderungen dort festzuhalten.

Externe Dokumentation

Um das Programm, auch ohne den Quellcode kennen zu müssen, verständlich zu machen muss ein Benutzerhandbuch geschrieben werden. Dies gibt eine Einweisung wie das Programm zu verwenden ist und erläutert alle Funktionen des Programmes.

Um nachweisen zu können, dass das Programm auch fehlerfrei nach der Abnahme lief ist ein Testkonzept notwendig, das alle vorgenommenen Tests dokumentiert.

2. Organisatorische Festlegungen

Die Implementierer und Modellierer wurden vom Team und dem Teamleiter aufgeklärt, dass sie sich mit den oben genannten Vorschriften zu beschäftigen haben und diese einhalten müssen. Nur die Einhaltung dieser Regel gewährleistet einen sauberen und lesbaren Quelltext. Die interne Dokumentation liegt somit in den Händen des Implementierer.

Die Verantwortliche für die Qualitätssicherung und Dokumentierung hat die Aufgabe, die erstellten Dokumente und Unterlagen auf die oben angesprochen Vorschriften zu kontrollieren und bei nicht einhalten die entsprechenden Personen/Rollen zu kontaktieren und über die Änderungen aufzuklären. Außerdem muss ein Benutzerhandbuch erstellt werden, welches für die späteren Anwender als Hilfe bei der Benutzung des Programms dient. Die externe Dokumentation liegt somit in den Händen der Verantwortlichen für die Qualitätssicherung und Dokumentierung.

In einer Software-Entwicklung bleiben fehlerhafte Code Segmente nicht aus. Diese fallen oft beim Test der einzelnen Komponente auf. Damit nach Abschluss der Entwicklungsarbeiten belegt werden kann, dass das Programm ausreichend getestet wurde, werden die durchgeführten Tests und deren Ergebnisse dokumentiert. Der Verantwortliche für Tests ist für die Erstellung der Testdokumentation zuständig und wird, wenn benötigt, durch die Verantwortlichen für Dokumentation unterstützt.