

Qualitätssicherungskonzept

Inhaltsverzeichnis

Inhaltsverzeichnis.....	1
1 Testkonzept.....	2
2 Dokumentationskonzept.....	3
2.1 Interne Dokumentation.....	3
2.2 Entwurfsdokumentation.....	3
3 Organisatorische Festlegungen.....	4

1 Testkonzept

Tests sollen gewährleisten, dass das Programm auch das macht, was es soll. Wir wollen parallel zum Implementieren von Programmteilen, bzw. nachher Tests schreiben, wobei es bei Rails 3 Gruppen von Tests gibt: die Unit-Tests, die funktionalen Tests und die Integrations-Tests. Unit-Tests sind dabei für die Prüfung der Modelle zuständig, während die funktionalen sich auf den Controller beziehen. Die Integration-Tests beziehen sich zwar auch auf den Controller, legen jedoch mehr Wert auf das Zusammenspiel des ganzen Programms. Rails erstellt auch eigene Tests, die man bei der Programmierung direkt nutzen kann.

Wir versuchen, einen möglichst großen Teil des Projektes durch Tests abzudecken, mindestens jedoch die Hälfte.

2 Dokumentationskonzept

2.1 Interne Dokumentation

Da in diesem Projekt hauptsächlich Ruby on Rails für die Implementierung verwendet wird, unterscheidet sich die quelltextnahe Dokumentation großteils von der anderer Programmiersprachen wie z.B. Java. Dies ist auf die Philosophie mit der Rails entwickelt wurde zurückzuführen, die vor allem die Prinzipien „DRY“ (Don't repeat yourself), „Konvention vor Konfiguration“ und die strikte MVC-Architektur beinhalten. Im Zusammenspiel mit der Natürlichsprachlichkeit der Rubysyntax ermöglichen diese Grundprinzipien, von Menschen intuitiv lesbaren Code zu erzeugen, ohne dass eine Vielzahl von Kommentaren notwendig ist.

Diese Lesbarkeit wird vor allem durch das zweite Prinzip gewährleistet, welches z.B. vorschreibt, dass ein Primärschlüssel einer Tabelle ein Integer ist und ID heißt. Mit diesen Mitteln ist es möglich, eine stabile Projektstruktur zu schaffen, bei der es auch ohne genaue Kenntnis des Projektes möglich ist, sich in das Projekt einzuarbeiten. Zur guten Lesbarkeit des Codes einer Railsapplikation trägt jedoch auch die bereits erwähnte Natürlichsprachlichkeit bei. Eines der Grundprinzipien der Programmiersprache Ruby ist es Code so lesbar wie möglich zu gestalten. So offeriert die Sprache meist elegante Makros und Befehle um gängige Kodieraufgaben elegant zu lösen.

Konventionen sind dabei genau wie in Rails ein großer Bestandteil der Sprache und so haben sich in der Community um Ruby viele De-facto-Standards durchgesetzt, um zu bezeichnen, was guter Code ist und was nicht.

In unserem Projekt wird es also nicht nötig sein, zu jeder Codezeile einzeln einen Kommentar zu verfassen. Dennoch ist es für jedes größere Projekt unumgänglich den Code abschnittsweise zu dokumentieren und so die Funktion und vor allem den Nutzen einzelner Codeblöcke festzuhalten. Es ist dabei, wie in vielen anderen Programmiersprachen auch, gängige Praxis die Funktionen mit in ihrer Form standardisierten, aussagekräftigen Kommentaren zum Nutzen und zur Art der Benutzung zu versehen. Liegen diese Kommentare vor, so ist es mit Hilfe von RDoc möglich, ähnlich wie mit javadoc, fertige Dokumentationen zu erzeugen, welche in ansprechender Form die Funktionalitäten der Module darstellen.

2.2 Entwurfsdokumentation

Das Ziel der Entwurfsdokumentation ist die verständliche und nachvollziehbare Einführung in die Implementierung. Der Entwurf sollte so auskommentiert werden, dass es für alle beteiligten Personen möglich ist, sich gut hinein zu lesen und einen Überblick sowie das Verständnis für die einzelnen Bestandteile des Programms zu bekommen. Dafür ist es wichtig Klassendiagramme sinnvoll zu wählen, das Datenbankschema gut zu dokumentieren sowie wesentliche Abläufe des Programms verständlich darzustellen.

3 Organisatorische Festlegungen

Dem Auftraggeber werden zur Sicherung der künftigen Qualität für Wartung, Weiterentwicklung, etc. der gesamte Quelltext, die Versionshistorie sowie alle im Rahmen des Projektes erstellten Dokumente zugänglich gemacht.

Das Dokumentationskonzept ist in diesem Zuge konsequent einzuhalten, damit sowohl jedem Projektmitglied als auch zukünftig an der Software arbeitenden Programmierern ein schneller Überblick gewährt wird. Bei mehrmaliger Verletzung der oben genannten Dokumentationsregeln muss der jeweilige Projektteilnehmer darauf hingewiesen werden, um weiteren Verletzungen der Regeln entgegenzuwirken.

Der Verantwortliche für Dokumentation stellt unter Mithilfe aller Projektteilnehmer die technische Dokumentation und die Benutzerdokumentation zusammen. Jeder Teilnehmer ist für die Einhaltung der festgelegten Standards für Quellcode und Kommentare im von ihm programmierten Teil verantwortlich.

Dabei stellt der Verantwortliche für Dokumentation die Qualität aller Dokumentationen sicher und passt sie ggf. neuen Anforderungen an. Die Prüfung erfolgt stichprobenartig und wöchentlich, bzw. bei Abgabe in Zusammenarbeit mit dem Projektleiter. Jeder ist angehalten veröffentlichten Code der anderen Gruppenmitglieder auf Dokumentationsfehler zu überprüfen. Der Projektleiter entwirft Zeitpläne während der Entwicklung und kontrolliert die Einhaltung der Fristen in Zusammenarbeit mit dem aktuellen Verantwortlichen für die einzelnen Projektphasen, um so einen reibungslosen Entwicklungsprozess zu gewährleisten. Die Terminabsprachen werden dann über redmine gesteuert, welches eine effektive Ticketverwaltung ermöglicht.

Zudem gibt es wöchentliche Gruppentreffen, um den aktuellen Projektstatus, bzw. den Fortschritt einzelner Aufgaben und die Aufgaben für die folgende(n) Woche(n) zu besprechen und zu verteilen.

Dabei wird zum Treffen durch den Verantwortlichen für Dokumentation ein Protokoll aufgenommen und auf der internen Projektseite (redmine) zugänglich gemacht.

Sollten sich bei der Bearbeitung einer Aufgabe Probleme ergeben, sind diese ggf. In einem außerordentlichen Treffen zu besprechen um schnellstmöglich eine Lösung zu finden. Für eine schnellere Kommunikation wurden Telefonnummern, Skypenamen und Emailadressen im redmine intern hinterlegt.

Das Bugfixing/-tracking erfolgt ebenfalls über redmine. Für jede Teilaufgabe/jeden Fehler wird ein Ticket erstellt. Dieses wird dann direkt einem Gruppenmitglied zugeordnet oder man wählt aus den verfügbaren Tickets aus. Dabei wird auf eine gerechte Lastenverteilung und die individuellen Fähigkeiten der einzelnen Teilnehmer geachtet.