

Qualitätssicherungskonzept

1 Dokumentationskonzept

Das Dokumentationskonzept beschreibt Vereinbarungen zur Dokumentation. Diese Vereinbarungen sind für die Lesbarkeit sowie Verständlichkeit des Projektes nötig. So wird die Einarbeitungszeit für Teammitglieder, sowie außenstehenden Personen, stark verringert.

Dadurch können Veränderungen schnell von anderen eingearbeitet und Erweiterungen zeitnah implementiert werden.

Das Dokumentationskonzept beinhaltet Vereinbarungen zur internen Dokumentation, als auch zur quelltextnahen strukturierten Dokumentation.

1.1 Interne Dokumentation

Die interne Dokumentation dient zur Erklärung von Variablen und Algorithmen innerhalb des Quelltexts. Sie soll hauptsächlich Entscheidungen verdeutlichen, die man anhand des Quelltexts nicht erschließen kann.

Da dieses Projekt ein Open Source Project ist, welches später auf Github erscheinen soll, wird die Dokumentation in Englisch erfolgen. Es gibt zur internen Dokumentation zum Teil auch Vereinbarungen im Abschnitt des Coding Standards.

1.2 Quelltextnahe strukturierte Dokumentation

Die quelltextnahe strukturierte Dokumentation dient zur einfachen Einarbeitung in Klassen und Funktionen eines komplexen Programms. Sie erklärt in kurzer Form die Funktionalität der Komponenten.

Wir verwenden javadoc zur automatischen Extraktion dieser Dokumentation. So wird die Dokumentation als HTML-Datei ausgegeben. Diese Dateien zeigen dann alle nötigen Informationen.

Wichtig ist, dass die zugehörige Komponente direkt nach den Kommentar kommt.

Die quelltextnahe strukturierte Dokumentation wird, unter gleicher Begründung wie im Teil der internen Dokumentation, auch in Englisch erfolgen.

2 Coding Standard

Der Coding Standard trägt, genau wie die interne Dokumentation, zum wesentlichen Verständnis und Lesbarkeit des Quelltexts bei. Hier ist es vor Allem wichtig, dass sich alle Programmierer, die am Projekt arbeiten, den gleichen Standard benutzen. Dazu gibt es folgende Vereinbarung:

Da wir mit Java eine der gängigsten Programmiersprachen verwenden, benutzen wir einen der bereits vorhandenen Standards, die Java Code Conventions¹. In diesem Standard sind nicht nur Vereinbarungen zur Benennung von Klassen, Variablen, Konstanten, etc., sondern auch Festlegungen zur internen Dokumentation (für bessere Lesbarkeit) und Dateibenennung.

3 Testkonzept

3.1 Allgemein

Da das Programm in Java geschrieben wird, benutzen wir das Framework JUnit. JUnit wurde in Java geschrieben und kann somit alle spezifischen Aspekte der Sprache Java testen. Dazu implementieren wir einen zu testenden Quellcode und entsprechende Tests. Die Antwort ist entweder positiv (grün) oder negativ (rot). Das negative Ergebnis kann in zwei Kategorien unterteilt werden. Entweder es kam beim Testen zu einem Fehler („Error“) oder zu einem falschen Ergebnis („Failure“). Der Unterschied liegt darin, dass ein falsches Ergebnis erwartet werden kann. Beide werden über eine Exception ausgegeben. Errors werden zu dem im Quellcode angegeben. Dabei werden Methoden, Klassen oder das Zusammenspiel von Komponenten betrachtet. Durch JUnit können die zu betrachtenden Objekte auf ein zu erwartendes Ergebnis getestet werden. Zudem sind unter anderem Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit zu berücksichtigen. Der implementierte Code sollte so früh wie möglich und fortlaufend von seinem Programmierer oder einem anderen Gruppenmitglied getestet werden. Somit können Fehler frühzeitig erkannt und behoben werden.

Das Testenkriterium, also das Kriterium was zum erfolgreichen beenden des Tests führt, ist, dass jede Klasse sein erwartetes Ergebnis liefert. Ist ein zu testender Quellcode zu schlecht, so kann dies zu einem Abbruch des Tests führen. Eine Wiederaufnahme erfolgt durch eine entsprechende Verbesserung.

Fehler und ihre Lösungsansätze oder wenn möglich ihre Lösungen, sind zu Dokumentieren. Dieses Wissen sollte den anderen Gruppenmitgliedern zur Verfügung gestellt werden. Treffen die Gruppenmitglieder auf das jetzt bekannte Probleme, so können sie diese schneller beheben. Durch die Tests sollen Fehler ausfindig gemacht und behoben werden, damit ein fehlerarmer,

¹<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

bzw. wenn es möglich ist, ein fehlerfreier Code entsteht. Die Verantwortung liegt dabei bei jedem Gruppenmitglied.

Einige Risiken wurden bereits in der Risikoanalyse angegeben. Für die Tests heben sich vor allem folgende Risiken heraus: Ein Problem wird nicht gelöst, Ein Problem wird nicht schnell genug gelöst, das Testen beansprucht zu viel Zeit.

3.2 Komponententest

Alle Gruppenmitglieder nutzen Eclipse, was das Implementieren von JUnit Test Cases erleichtert. Dazu wird File»»New»»"JUnit Test Case eine neue Klasse im Package eingerichtet, die ausgewählte Methoden und Klassen getestet. Das Ziel des Komponententest ist es, einzelne Komponenten zu testen, daher, den von der Gruppe implementierten Quellcode, bevor auf diesen andere Komponenten aufgebaut werden.

Ein weiterer Punkt, der unter den Komponententest fällt, ist die Evaluierung der Oberfläche mittels der SUS(System Usability Scale). Hierfür bekommen die Teilnehmer einen Fragebogen mit 10 angepassten Aussagen, welche sie mit einer Zahl von 1(ich stimme vollständig zu) bis 5 (ich stimme überhaupt nicht zu) bewerten können. Aus einem Fragebogen werden die vergebenen Punkte addiert (hierbei gilt eine 1 als 0 und eine 5 als 4) und mit 2,5 multipliziert. Das erhaltene Ergebnis ist der SUS Score, welcher zwischen 0 und 100 liegt. Ein System gilt als Perfekt, wenn es die 100 erreicht. Diese wird an 20 bis 30 Testpersonen durchgeführt.

3.3 Integrationstest

Im Integrationstest wird das Miteinander arbeiten der Komponenten getestet. Bei uns betrifft das die einzelnen Komponenten die vom Team implementiert werden (z.b.Internetseite) und die vom Project Owner vorgegebenen Programme.

3.4 Systemtest

In diesem Test wird die Gesamtfunktionalität geprüft. Dies geschieht nach dem Komponenten- und Integrationstests. Hierbei wird jede Komponente an ihre vorgesehene Stelle eingebunden und das somit entstehende Gesamtprogramm geprüft. Außerdem wird getestet, ob es im geplanten Umfeld zurechtkommt. In diesem Fall die Verknüpfung mit dem Web.

3.5 Abnahmetest

Das Produkt wird dem Auftraggeber vorgeführt und auf Vollständigkeit geprüft. Trifft dies zu wird das Produkt übergeben.

3.6 Regressionstest

Unter diesen Test versteht man das wiederholte Testen von Quellcode. Wir können mehrere Juni-Tests in ein JUnit Suite ablegen und ausführen lassen, was uns das einzelne Ausführen jedes Tests erspart. Die Regressionstests werden ausgeführt, wenn jemand den Code modifizieren (Pflege, Korrektur oder Änderung) will. Will also jemand am Code etwas ändern, so muss er erst einen Test durchführen, um sicher zu gehen, dass er vor seiner Änderung fehlerfrei ist.

4 Organisatorische Festlegungen

Teamtreffen

Das Team trifft sich vorerst jeden Mittwoch um 15:30 Uhr. Diese Treffen dienen zur Erfassung des aktuellen Stands der zu bearbeitenden Aufgabe, sowie zur Spezifizierung von neuen Aufgaben. Da diese Aufgaben gerecht auf alle Mitglieder, je nach ihren Stärken, verteilt werden sollen, ist die Anwesenheit aller Pflicht.

Wichtig: Diese Regelung tritt zur Prüfungszeit WS 2015/16 außer Kraft! Zur vorlesungsfreien Zeit werden gesondert Termine ausgemacht. Im neuen Semester wird dann wieder ein regelmäßiger Termin festgelegt.

Kommunikation

Also Kommunikationskanäle werden weiterhin der Telegram Messenger und das gruppeninterne Forum genutzt.

Die Teamtreffen stellen jedoch den wichtigsten Kommunikationskanal dar, da bei persönlichen Gesprächen wesentlich weniger Missverständnisse auftreten.

Ergebnisse von Aufgaben

Alle Dokumente werden an den Projektleiter geschickt, um diese noch mal grob zu überfliegen und danach zur Abgabe freizugeben.

Alle Ergebnisse werden in das Repo geladen, damit diese für jedes Teammitglied zur Verfügung stehen.

Sonstiges

Die in diesem Dokument festgelegten Regeln müssen von allen Mitgliedern beachtet werden. Jeder ist selbst dafür verantwortliche.

Das Qualitätssicherungsmanagement und Testmanagement wird in regelmäßigen Abständen die Einhaltung dieser Regeln kontrollieren.