

Qualitätssicherungskonzept

BETREUER:

Betreuer: Johannes Frey

Betreuer: Kay Müller

Tutor: Marvin Hofer

TEAM:

Paul Reinhardt

Daniel Pohl

Julius Kluge

Michelle Kampfrath

Nicole Timme

Stefan Walter

Tobias Krawetzke

Leipzig, am 14. Januar 2017

Inhaltsverzeichnis

1	Dokumentationskonzepte	3
1.1	Interne Dokumentation	3
1.2	Quelltextnahe strukturierte Dokumentation	3
1.3	Entwurfsbeschreibung	3
2	Coding Standard	3
3	Testkonzepte	4
3.1	Komponententests	4
3.2	Integrationstests	4
3.3	Systemtests	4
3.4	Abnahmetests	4
4	Organisatorisches	5
4.1	Teamtreffen	5
4.2	Protokolle	5
4.3	Kommunikation	5

1 Dokumentationskonzepte

Die Quelltextdokumentation ist ein wichtiger Bestandteil der Qualitätssicherung. Eine gute Dokumentation erleichtert die Zusammenarbeit im Team, da das Einarbeiten in fremden Quelltext wesentlich schneller erfolgen kann.

Um die Dokumentation möglichst aktuell zu halten ist jeder Programmierer selbst für die Dokumentation seines Quelltextes verantwortlich. Die gesamte Quelltextdokumentation ist in Englisch zu verfassen.

1.1 Interne Dokumentation

Die interne Dokumentation erfolgt durch Kommentare der Form: „//Kommentar“ oder „/*mehrzeiliger Kommentar*/“ im Quelltext. Kommentare stehen in einer eigenständigen Codezeile. Sie dienen der näheren Erleuterung des Quelltextes, an komplexen Stellen, die für andere Entwickler nicht sofort einleuchtend sein könnten.

1.2 Quelltextnahe strukturierte Dokumentation

Die quelltextnahe Dokumentation wird durch Kommentare der Form `/** Kommentar */` markiert. Diese werden dann mit dem Javadoc Tool extrahiert und stehen als HTML Dokument zur Verfügung. Es werden Klassen und Methoden mit solchen Kommentaren versehen, um deren Zweck leichtverständlich zu erläutern.

1.3 Entwurfsbeschreibung

Die Entwurfsbeschreibung fasst die wichtigsten Funktionsmerkmale der Software zusammen. Sie gibt Einblick in die zentralen Entwurfsentscheidungen, so dass ein grundlegendes Verständnis des Systems entsteht, mit dem Dritte dieses ergänzen oder ändern können.

2 Coding Standard

Durch einen einheitlichen Coding Standard wird die Verständlichkeit erhöht, was Wartung und Pflege erleichtert und so auch die Softwarequalität verbessert.

Um zusätzlichen Aufwand zu vermeiden, trägt jeder Programmierer selbst die Verantwortung sich an diesen Standard zu halten.

Wir werden uns in diesem Projekt am Coding Standard von Drupal¹ orientieren und für Java spezifische Fälle an den Java Code Conventions². Allerdings soll die Bezeichnung von Methoden und Variablen nach der lower CamelCase Konvention erfolgen. Innerhalb des Quelltextes ist Englisch zu benutzen.

¹<https://www.drupal.org/docs/develop/standards/coding-standards>

²<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

3 Testkonzepte

Um ein gutes Endprodukt abliefern zu können, muss dieses ausgiebig auf Fehler getestet werden. Dazu gehören unter anderem die Tests der einzelnen Klassen und Methoden, sowie die Zusammenarbeit der Komponenten.

Während der Entwicklungsphase ist es wichtig, Fehler oder Probleme zu beseitigen. Dazu werden wir, durch intensives testen, in der Lage sein diese früh zu entdecken. Desweiteren müssen wir die Anforderungen an die Software, sowie die Funktionalität gewährleisten.

3.1 Komponententests

Der Unittest ist zweifelsohne einer der wichtigsten Tests bei der Softwareentwicklung. Durch ihn wird sichergestellt, dass Methoden und Klassen korrekt funktionieren. Darunter versteht sich, dass die Komponente zum einen fehlerfrei ist und zum anderen das gewünschte Ergebnis liefert. Dabei ist es bei unserem Projekt essenziell auf Wertebereiche, sowie Effizienz zu achten.

Als Testumgebung verwenden wir JUnit, da es speziell zum Testen der Units, sprich Klassen und Methoden, entwickelt wurde. JUnit ist in Java geschrieben und kann deshalb sogar sprachspezifische Aspekte unseres Quellcodes berücksichtigen.

3.2 Integrationstests

Hierbei wird die Zusammenarbeit der Komponenten geprüft, das heißt ob sie ordnungsgemäß miteinander kommunizieren und keine unerwarteten Schwierigkeiten auftreten. Da mit erhöhter Anzahl der Komponenten, die Menge an Tests unverhältnismäßig steigt, werden Integrationstests für einzelne Subsysteme durchgeführt und diese werden für die weitere Entwicklung der Software als gesonderte Komponente betrachtet. Dies wird Bottom-Up-Methode genannt und führt zwangsläufig dazu, dass die Integrationstests erst in einer späteren Phase des Projektes abgeschlossen werden.

3.3 Systemtests

An diesem Punkt ist es substanziell den Test aus Anwendersicht zu betrachten. Vorerst werden hier alle Komponenten der Software zusammengeführt und auf ihre Funktionalität als Gesamtheit geprüft. Außerdem wird sichergestellt, dass die zuvor festgelegten Anforderungen an das Projekt erfüllt wurden.

3.4 Abnahmetests

In dieser Testphase wird das Projekt dem Auftragsgeber präsentiert, dabei prüft dieser, ob seine Anforderungen und eventuelle Wünsche zu seiner Zufriedenheit erfüllt worden sind. Ist dies der Fall so wird das Projekt von der Gruppe übergeben und der Auftrag als erledigt abgeschrieben.

4 Organisatorisches

4.1 Teamtreffen

Das Team trifft sich einmal die Woche, meist am Montag um 15:00 Uhr, oder an einem anderen vereinbarten Termin. Diese Treffen dienen dazu, sich mit den Teammitgliedern über den aktuellen Stand des Projektes auszutauschen, sich über die nächste Abgabe zu verständigen, Unklarheiten zu klären und Aufgaben zu verteilen.

Treffen mit den Betreuern finden, wenn vereinbart, Mittwochs 15:15 Uhr in der Hainstraße 11 statt.

In der Prüfungszeit werden die Treffen an gesonderten Terminen stattfinden, ebenso in der vorlesungsfreien Zeit. Im neuen Semester wird ein neuer regelmäßiger Termin festgelegt.

4.2 Protokolle

Zu oben genannten Treffen wird jeweils von einem Teammitglied Protokoll geführt. Diese dienen besonders den Mitgliedern, welchen es nicht möglich war am Treffen teilzunehmen, um nachzuvollziehen, was zu den Treffen besprochen wurde. Die Protokollführer werden in alphabetischer Reihenfolge bestimmt. Protokolle werden ins Git hochgeladen.

4.3 Kommunikation

Das Team kommuniziert über Slack. Für jeden Themenbereich gibt es einen gesonderten Channel, damit alle Absprachen übersichtlich sind und sich nichts vermischt.

Außerdem hat das Team einen Gitbot eingerichtet. Dieser informiert sofort über jegliche Commits und Issues.

Im GitLab werden alle relevanten Dokumente und Dateien für das Team hochgeladen, sowie der Quellcode. Commits erfolgen mit ausführlicher Beschreibung, z.B:

Betreff

Änderungsort₁ : Änderungsbeschreibung₁

...

Änderungsort_n : Änderungsort_n