

Qualitätssicherung

Universität Leipzig – Softwarepraktikum 2014/2015

Semantic Chess

Jonas Herrig

Hanno Krümpelmann

Nathalie Bargenda

Duc Hieu Nguyen

Stefan Süsmeier

Lisa Höncke

Inhaltsverzeichnis

1. Dokumentationskonzept.....	2
1.1 Sprache.....	2
1.2 Coding Standard.....	2
1.3 Code-Dokumentation.....	2
1.4 Sonstige Dokumentation.....	2
2. Testkonzept.....	3
2.1 Komponententest.....	3
2.2 Integrationstest.....	3
2.3 Systemtest.....	4
2.4 Abnahmetest.....	4
3. Organisatorisches.....	4

1. Dokumentationskonzept

Wie auch bereits in der Risikoanalyse festgestellt ist eine durchdachte und vollständige Dokumentation des Projekts, sowohl intern auf Codeebene als auch extern, in der Form von Protokollen entscheidend für eine erfolgreiche Bewältigung der gesetzten Ziele.

Durch die Dokumentation ist es möglich älteren Code ohne längere Einarbeitung wieder aufzusuchen, Fehler zu finden etc. Außerdem erlaubt eine ausführliche Dokumentation es auch nach Fertigstellung des Projektes anderen Personen sich einzuarbeiten und evtl. Anregungen für eigene Arbeiten zu finden bzw. Teile des Projektes nach ihren Vorlieben zu erweitern oder zu ändern.

1.1 Sprache

Als Sprachen werden sowohl Englisch als auch Deutsch verwendet. Die interne Dokumentation wird in Englisch verfasst, die externen Dokumenten hingegen in Deutsch. Die Fachbegriffe, Variablen- und Klassennamen etc. werden hingegen aus dem Englischen übernommen.

1.2 Coding Standard

In diesem Projekt wird als Coding Standard der in den Java Code Conventions definierte festgelegt. Dies trägt wesentlich zu Übersichtlichkeit des Quellcodes bei.

1.3 Code-Dokumentation

Sämtlicher Java Quellcode, also sowohl Klassen als auch Methoden und Attribute werden mit entsprechenden selbsterklärenden javadoc-Kommentaren versehen. Ebenso sollen eventuelle Unklarheiten bzw. Dinge welche nicht auf den ersten Blick ersichtlich sind im Quellcode durch Kommentare an den jeweiligen Stellen erläutert werden. So sollen beispielsweise Schleifen in denen die Laufvariablen nicht eindeutig sind bzw. Verzweigungen immer näher kommentiert werden.

1.4 Sonstige Dokumentation

Neben den Dokumentationen in und um den Quellcode ist es wichtig auch die gruppeninterne Treffen sowie die Treffen mit dem Auftraggeber zu protokollieren. Diese Protokolle werden in das Wiki innerhalb des Olats gestellt, so dass die Gruppenmitglieder auch im Nachhinein noch in der Lage sind den Prozess nachzuverfolgen bzw. verpasste Treffen aufzuarbeiten.

2. Testkonzept

Für ein Projekt dieser Größe, sowohl im Umfang als auch in der Komplexität des Inhalts bzw. der Anzahl an beteiligten Personen ist es unerlässlich von Beginn der Implementation an regelmäßige Test durchzuführen. Dies ermöglicht es frühzeitig Fehler zu erkennen und zu beheben, was wiederum ein effizientes Arbeiten ermöglicht und Zeitengpässen, grade im späteren Verlauf des Projektes vorbeugt.

Bei den Tests setzen wir auf wenn möglich auf automatische Vorgehensweisen. So sollen die Komponententest mithilfe des JUnit Frameworks durchgeführt werden. Für kurzfristig benötigte Resultate können die Test manuell ausgeführt werden, wobei jedoch offensichtlicherweise eine Einbindung in JUnit zu bevorzugen ist.

Die Ergebnisse der Test sollen klar und eindeutig dokumentiert werden, etwa durch eine dem gesamten Team zugängliche Exeltabelle oder eine ähnliche Methode. So können Fehler in der Programmhistorie zurückverfolgt werden und die Problembehebung wird wesentlich vereinfacht.

2.1 Komponententest

Komponententests werden genutzt um innerhalb des Gesamtprojekts einzelne Klassen oder Funktionen auf ihre Korrektheit zu testen. Mithilfe des Frameworks JUnit ist es so möglich mit relativ geringem Aufwand Neuerungen zu testen und zu sehen ob ihre jeweilige Funktionalität gewährleistet ist, sogenannte Regressionstests. Die Verantwortlichkeit für den jeweiligen Teil des Codes liegt dabei bei denjenigen welche ihn geschrieben bzw. zuletzt angepasst haben. Falls ein Test nicht durch Junit nicht möglich ist, werden anstelle dessen Testklassen implementiert um die Funktionalität zu überprüfen.

2.2 Integrationstest

Beim Integrationstest werden einzelne Komponenten des Projektes zusammengeführt um die Funktionalität im Verbund zu testen. Hierbei wird vor allem auf die Leistung und Zuverlässigkeit des Programms geachtet, da diese teilweise erst ab diesem Schritt mit Sicherheit festzustellen sind. Sollten Fehler oder andere Probleme auftreten, so müssen diese zuerst gelöst werden bevor weiter fortgefahren werden kann.

2.3 Systemtest

Beim Systemtest werden alle erstellten Komponenten zusammengeführt und gemeinsam getestet. Dabei wird darauf geachtet ob die vorher definierten Anforderungen als erfüllt angesehen werden können und ansonsten ob es möglich ist diese doch noch zu erfüllen. Sollten Fehler auftreten und behoben werden, so müssen Komponenten- und Integrationstest natürlich wiederholt werden.

2.4 Abnahmetest

In diesem letzten Test wird das Produkt mit den Vorstellungen und Wünschen des Auftraggebers abgeglichen und diesem vorgestellt. Ist dieser zufrieden wird das Projekt abgenommen und der Auftrag gilt als erledigt.

3. Organisatorisches

Die wöchentlichen Treffen der Gruppe haben sich in diesem Rahmen als äußerst effektiv herausgestellt. Durch die Treffen war es möglich Fragen direkt vor Ort zu beantworten, sich zu koordinieren und den gemeinsamen Stand zu bestimmen. Dieses Konzept werden wir auch weiterhin fortführen und außerdem eine lose Verbindung per Skype und E-mail halten. Gerade in den Semesterferien wenn vermutlich nie alle in Leipzig vor Ort sind werden diese Wege die lokalen Treffen ersetzen.