

Qualitätssicherungskonzept Link Discovery

Projektverantwortlicher: Sascha Hahne

13. April 2015

Inhaltsverzeichnis

1	Dokumentationskonzept	3
1.1	Allgemein	3
1.2	Code	3
1.2.1	Quellcode	3
1.2.2	Code Dokumentation	3
1.3	Handbuch	4
1.4	Weiteres	4
2	Testkonzept	4
2.1	Allgemein	4
2.2	Oberfläche	5
2.3	Evaluationskonzept	5
2.4	Dokumentation der Tests	5
2.5	Sicherung	5
3	Organisatorische Festlegungen	5
3.1	Treffen	5
3.2	Kommunikation	6
3.3	Dokumente	6
3.4	Teamwork	6

Die in diesem Dokument genannten und festgelegten Richtlinien sind konsequent und durchgehend einzuhalten, um die eigenen Ansprüche an ein effizientes Arbeiten einzuhalten.

1 Dokumentationskonzept

1.1 Allgemein

Die Dokumentation ist die Basis jeder Form von gemeinsamer Arbeit im Bereich der Softwareentwicklung. Sie ermöglicht es schnell den Code anderer zu verstehen, sodass Klassen und Funktionen einfach genutzt werden können. Zudem erleichtert die Dokumentation ein späteres Verständnis des Codes, sodass der Zeitraum der Wiedereinarbeitung stark verkürzt wird. Desweiteren ist eine gute Dokumentation wichtig, um den Überblick über den Verlauf des Projekts zu behalten. Dies dient sowohl der Kontrolle, welche Aufgaben noch erfüllt werden müssen und welche bereits fertig gestellt wurden, als auch der Fehlererkennung und der Vermeidung von Streitsituationen. Daher ist es essentiell, dass eine einheitliche und saubere Dokumentation eingehalten wird.

1.2 Code

1.2.1 Quellcode

Bei der Form des Quellcodes halten wir uns an den Standard von Eclipse. Dies dient zum einen der Lesbarkeit und der Übersicht und ist zum anderen ein gut bekanntes und vertrautes Format für die meisten Java-Programmierer. Die wichtigsten Punkte werden hier nochmal genannt: Die öffnende geschweifte Klammer ist auf der Höhe des Statements und ein Zeichenabstand in Form eines Leerzeichen von diesem entfernt. Unterstatements werden eingerückt. Dabei wird der eingerückte Abstand eine Länge von 4 Satzzeichen in Form von Leerzeichen betragen. Die schließende geschweifte Klammer bekommt eine Extrazeile. Die Position ist dabei auf Höhe des ersten Satzzeichens des Statements, welches sie schließt. Zwischen verschiedenen Methoden befinden sich Leerzeilen, um sie optisch gut voneinander abzugrenzen. Weiterhin befinden sich Leerzeilen vor und nach Kommentaren. Methoden und Klassennamen sowie Variablen werden im CamelCase verfasst. Bei Variablen und Methoden ist der erste Buchstabe klein, bei Klassennamen groß. Sämtliche Namen werden in Englisch gehalten und haben sprechende Namen.

1.2.2 Code Dokumentation

Um sich im Code schnell zurechtzufinden und die Funktion eines Abschnitts oder einer Methode schnell zu erkennen, ohne zuerst den ganzen Code nach-

zuvollziehen, ist es sinnvoll, diesen zu kommentieren. Auch hierbei sollten einige Richtlinien eingehalten werden. Zum einen sollten Kommentare stets auf Deutsch geschrieben werden. Desweiteren stehen Kommentare immer oberhalb des Abschnitts, welchem sie zugehören. Einzige Ausnahme davon bilden einzeilige Kommentare für die Beschreibung von Codezeilen. Dabei ist der Javadoc-Standard einzuhalten. Vor jedeweglassene Methode gehört ein mehrzeiliger Kommentar. Dieser hat folgende Punkte zu enthalten: Die Funktion der Methode Die Eingabewerte für den Aufruf Die Rückgabewerte Der Autor Notiz zur Vollendung Letzte Änderung Selbiges gilt auch für die Kommentare am Anfang von Klassen. Bei langen Methoden werden kurze Kommentare zur Funktionalität der einzelnen Abschnitte eingefügt werden.

1.3 Handbuch

Bei der Umsetzung unseres Projekts legen wir großen Wert auf eine intuitive Benutzbarkeit. Es sollte für jeden Anwender verständlich sein, ohne sich lange mit der Materie oder dem Programm auseinander zu setzen. Trotzdem werden wir ein Handbuch schreiben, um die Funktionalität des Programms zu dokumentieren. Darin werden die Userstories geschildert und Anleitungen gegeben um diese zu erfüllen. Zusätzlich werden darin Hinweise zur Installation von Limes und unserem Programm gegeben und erläutert, wie die beiden Programme zur Interaktion gebracht werden.

1.4 Weiteres

Nicht nur die Elemente des Projekts müssen dokumentiert werden, sondern auch dessen Verlauf und die Arbeit der einzelnen Mitglieder. Die Aufwandsfassung ist automatisch enthalten in den Projektbedingungen, in denen festgelegt ist, dass ein jedes Mitglied seinen Workload aufzuschreiben hat. Zusätzlich wird bei jedem Treffen ein Protokoll geführt. Dieses beinhaltet die folgenden Punkte: bereits erledigte Aufgaben den Fortschritt bei fortlaufenden Aufgaben die noch zu erledigenden Aufgaben die Teilnehmer Die Verteilung der weiteren bzw neuen Aufgaben

2 Testkonzept

2.1 Allgemein

Für den reibungslosen Ablauf und effektives Arbeiten ist es wichtig, Fehlerquellen schnell zu lokalisieren. Das gilt sowohl für die einzelnen Komponenten als auch für das gesamte Projekt. Daher müssen alle Komponenten einzeln getestet werden. Gleiches gilt auch für ihre Zusammenarbeit. Zusätzlich muss sichergestellt werden, dass das Projekt seine Funktionen erfüllt und seinen Anforderungen gerecht wird.

2.2 Oberfläche

Die Funktionalität der Methoden und Klassen während der Entstehung werden manuell getestet durch daran angepasste Main Funktionen. Sie ermöglichen es in jeder Klasse Testdatensätze zu erzeugen und die Methoden mit den Daten dahingehend zu überprüfen, ob sie die gewünschten Ergebnisse zurückgeben. Mit Hilfe von JUnit lässt sich in der fertigen GUI testen, ob alle Funktionen richtig arbeiten, daher ist es für die Tests der Feinjustierung gut geeignet. Für den Test des Programms selbst wird Selenium verwendet.

2.3 Evaluationskonzept

Da es bei unserem Projekt auf intuitive Bedienung ankommt, werden wir das Programm von Außenstehenden testen lassen. Diese Testprobanden werden die von uns entwickelten Userstories ausprobieren. Probanden mit Fachwissen werden dabei auch die Möglichkeit haben, eigene Userstories zu schreiben und mit diesen die Software zu testen. Testdatensätze für Entwicklungstests lassen sich bei W3C unter “Limes” “examplefiles” finden.

2.4 Dokumentation der Tests

Bei allen Erweiterungen der GUI wird der Teil getestet, ob er seiner Funktion nachkommt, ohne die anderen Funktionen der GUI zu stören. Dabei muss dokumentiert werden: welche Teile des Programms getestet wurden welche Ergebnisse erbracht wurden wie getestet wurde durch wen getestet wurde wann der Test stattgefunden hat welche Fehler aufgetreten sind ob das Modul fertig gestellt wurde

Dabei ist darauf zu achten, dass Fehler vollständig genannt und rundum beschrieben werden.

2.5 Sicherung

Da das Programm nach und nach entsteht und gerade die GUI öfters geändert werden muss, ist es notwendig frühere Versionen zu sichern. Immerhin ist es nicht sicher, dass alle Erweiterungen auf Anhieb gelingen, wodurch manche eventuell rückgängig gemacht werden müssen. Daher müssen die wichtigen Programmteile auch in früheren Versionen gesichert werden.

3 Organisatorische Festlegungen

3.1 Treffen

Wie durch den Scrumprozess vorgesehen finden wöchentliche Treffen mit dem Scrum Master statt. Diese dienen der Kontrolle des Fortschritts und

der Sicherstellung, dass der Arbeitsplan eingehalten wird. Desweiteren werden dort Probleme geäußert und gelöst. Zudem hält der Scrum Master den Kontakt zum Project Owner setzt mit ihm Termine fest und kann uns seine Wünsche mitteilen, da Treffen mit dem Project Owner nicht frequent vorkommen. Zu diesen Treffen haben alle Mitglieder zu erscheinen. Weiterhin findet nach allen Gesprächen mit dem Scrum Master ein Meeting der Gruppenmitglieder statt. Dieses dient zur Absprache, welches Mitglied welche Aufgabe bis zum nächsten Treffen zu erledigen hat. Zudem werden Absprachen gehalten und Entscheidungen zu möglichen Design- und Funktionalitätsfragen getroffen. Desweiteren finden nach Bedarf weitere Spontantreffen statt.

3.2 Kommunikation

Kommunikation ist für jede Gruppenarbeit essenziellen. Sie sichert die Produktivität der Zusammenarbeit und verhindert Streit und böses Blut. Daher wurden primäre E-mail Adressen und Handynummern ausgetauscht. Der Hauptteil der Gruppenkommunikation verläuft über Rundmails. Dies hat den Vorteil, dass die Themen geordnet bleiben und durch den formalen Aspekt einer Mail bleiben Spammnachrichten aus, ebenso wie humoristische Anmerkungen, welche andere für unangebracht halten würden. Durch Kurznachrichten und Telefonate können Absprachen zwischen nur zwei Leuten getroffen werden. Dies ist notwendig, da nicht alle Arbeiten einzeln gemacht werden, sondern manchmal Teilgruppen sich einer Aufgabe annehmen. Daher wendet sich nicht jede Frage an jeden.

3.3 Dokumente

Das Erstellen von Dokumenten ist eine wichtige Aufgabe im Prozess der Softwareentwicklung. Da diese auch von mehr als nur einer Person verfasst werden und damit auch von weiteren Mitgliedern notfalls gegengelesen werden können, verwenden wir gemeinsame Online-Dokumente. Dazu nutzen wir den kostenlosen Dienst Google Docs. Die Dokumente konnten dort von allen gelesen und bearbeitet werden, die den Link erhalten haben. Im Laufe des Projektes hat sich erwiesen, dass ein Git-Repository zur Verwaltung und Bearbeitung der Dokumente sich besser eignet, da wir als Gruppe die Dokument nun mit Latex verfassen. Diese können direkt als pdf auf die Webseite hochgeladen werden.

3.4 Teamwork

Ein gutes Klima im Team führt zu einer höheren Effizienz der Zusammenarbeit. Daher sollten wir ein freundliches Miteinander einhalten. Jedes Mitglied hat Stärken und Schwächen, diese sind bereits im Voraus bekannt gewesen und darauf angepasst werden auch die Arbeiten verteilt. Somit fühlt

sich keiner über- oder unterfordert. Auch eine Ungleichverteilung der Arbeit soll so vermieden werden, da im Vorfeld bekannt ist, wer an welcher Aufgabe wie viel Anteil haben wird. Somit wird das Aufwandsverhältnis für jedes Mitglied entsprechend ausbalanciert. Aus diesen Gründen gibt es keinen Grund für Diffamierung oder Beleidigung untereinander. Jedes Mitglied hat den Ansporn, dieses Projekt positiv zu beeinflussen und sollte daher mit dem nötigen Respekt behandelt werden. Außerdem ist bekannt, dass jedes Mitglied ein eigenes Leben hat und weitere Pflichten innerhalb und außerhalb der Universität. Darauf muss Rücksicht genommen werden, wenn auch jeder verpflichtet ist, ausreichend Zeit und Konzentration auf dieses Praktikum zu verwenden. Kritik an der Arbeit eines einzelnen oder allen darf genannt werden. Diese sollte trotz aller Bestimmtheit immer noch höflich und respektvoll geäußert werden. Je nach Fall reicht es, diese der Person selbst zu nennen. In manchen kritischen Fällen ist es aber durchaus richtig, sie in den gemeinsamen Treffen zu nennen, sodass alle Mitglieder an einer Lösung arbeiten können.