

Universität Leipzig
Softwaretechnik-Praktikum
Sommersemester 2016

Qualitätssicherungskonzept

Editierung von RDF-Datenbanken

Inhaltsverzeichnis

1 Qualitätsanforderungen

2 Dokumentationskonzept

2.1 Programmierstandards

2.2 Quelltextdokumentation

3 Testkonzept

3.1 Komponententest

3.2 Integrationstest

3.3 Systemtest

4 Organisatorische Festlegungen

5 Quellen

1 Qualitätsanforderungen

Qualitätsmerkmale	Definition
Funktionalität <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Interoperabilität</div>	Interoperabilität ist die Fähigkeit unterschiedlicher Systeme, möglichst nahtlos zusammenzuarbeiten.[2]
Zuverlässigkeit <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Reife</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Fehlertoleranz</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Wiederherstellbarkeit</div>	"Software-Zuverlässigkeit ist definiert als „Wahrscheinlichkeit der fehlerfreien Funktion eines Computerprogramms in einer spezifizierten Umgebung in einer spezifizierten Zeit.“ [3]
Benutzbarkeit <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Verständlichkeit</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Erlernbarkeit</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Bedienbarkeit</div>	"Benutzbarkeit bedeutet, dass eine interaktive Software ihre Benutzer/innen gut dabei unterstützt, die gewünschten Arbeitsgänge zu erledigen." [4]
Wartbarkeit <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Änderbarkeit</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Stabilität</div> <div style="border: 1px solid black; padding: 2px; margin-top: 5px;">Testbarkeit</div>	"Eigenschaft der Software, die eine Modifizierung mit geringem Aufwand möglich macht." [5]

Tabelle 1: Benötigte Software Qualitätsmerkmale [1]

2 Dokumentationskonzept

Eine gute Dokumentation bewirkt sowohl eine bessere Verständlichkeit für die Projektmitglieder als auch eine optimale Einarbeitungszeit von projektfremden Entwicklern in den Quellcode und erleichtert damit die Wartung und die Weiterentwicklung der Software. Außerdem wird dadurch die Qualität eines Software Produktes gesteigert.

Eine Dokumentation gibt einen Überblick über mögliche auftretende Fehler und den Arbeitsaufwand. Der Vorteil, der sich daraus ergibt, ist eine bessere Krisenbewältigung und somit wird das Risiko des Scheiterns minimiert.

2.1 Programmierstandards

Um zur besseren Lesbarkeit und schnelleren Verständlichkeit des Quellcodes zu gelangen, ist es sinnvoll, einen Coding Standard verbindlich zu vereinbaren. Unter folgendem Link ist der Coding-Standard zu finden:

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

2.2 Quelltextdokumentation

Es existieren in den Programmierstandards auch Standards zur Dokumentation des Quelltextes. So müssen Funktionen dokumentiert sein, und es sollten auch einzelne Schritte innerhalb der Funktionen beschrieben sein. Da man grundsätzlich immer sprechende Variablennamen in der sogenannten Camel-Case Schreibweise nutzen sollte, sollte der Programmcode auch für externe Programmierer gut lesbar sein.

3 Testkonzept

Bei der Entwicklung aufwendiger Software ist es relevant zu testen. Die meisten Softwareprodukte, die in Teams programmiert werden, werden in Teilprojekte aufgeteilt und jedem Mitglied zugeteilt. Um Probleme beim zusammenfügen der Softwarekomponenten zu vermeiden, muss jedes Teilprojekt bereits einzeln getestet werden. Dennoch können Fehler im Gesamtprodukt nicht ausgeschlossen werden, deshalb wird auch das gesamte Projekt überprüft. Es existieren viele vorgefertigte und automatische Tests und Testumgebungen, um eine Software zu testen. Im Allgemeinen sind Komponententests, Integrationstests, der Systemtest und der Abnahmetest zu berücksichtigen.

3.1 Komponententest

Der Komponententest (oder auch Unit-Test) ist ein Test der einzelnen Komponenten der Anwendung. Es werden also Klassen bzw. Module der Software betrachtet.

Dabei werden Testreihen und Beispiele für die einzelnen Klassen und Methoden entwickelt. Bei jedem Durchlauf eines Tests wird das Ergebnis über eventuelle Fehler und Funktionsdefizite sorgfältig dokumentiert. Die Fehlerquelle sollte dann gefunden und wenn möglich behoben werden. Voraussichtlich eignet sich für unserer Projekt das Framework JUnit oder TestNG.

3.2 Integrationstest

Der Integrationstest testet die Zusammenarbeit voneinander abhängiger Komponenten. Der Testschwerpunkt liegt auf den Schnittstellen der beteiligten Komponenten und soll korrekte Ergebnisse über komplette Abläufe hinweg nachweisen. Ein solcher Test ist deshalb besonders dann sinnvoll, wenn fertige Komponenten existieren und es auf deren Einbettung in bereits vorhandene Anwendungsstrukturen ankommt.

Somit sollten fertige Komponenten stets auf reibungslose Integration geprüft werden. Voraussichtlich eignet sich für unserer Projekt das Framework JUnit oder TestNG.

3.3 Systemtest

Beim Systemtest wird das gesamte System gegen die Anforderungen getestet. Der Test findet auf einer Testumgebung statt und wird mit Testdaten durchgeführt. Dabei möchte man so gut wie möglich die Produktivumgebung des Kunden simulieren. Voraussichtlich eignet sich für unserer Projekt das Framework FitNesse oder Selenium.

4 Organisatorische Festlegungen

Es findet jede Woche am Montag um 11:30 Uhr ein Teamtreffen mit dem Auftraggeber und Betreuer statt. Dabei wird die Aufgabenstellung analysiert, Aufgaben auf die Teammitglieder verteilt und es besteht die Möglichkeit mit dem Auftraggeber und/oder dem Betreuer Rücksprache zu halten. Die Kommunikation untereinander läuft per Telegram-Chat oder per Mail. Dokumente an denen wir arbeiten, speichern und teilen wir mittels Google Docs.

Das Dokumentations- und Testkonzept ist für alle Mitglieder des Projektes verbindlich und einzuhalten. Der Qualitätsverantwortliche achtet auf eventuelle Verletzungen von Regeln und weist auf diese hin. Jedes Projektmitglied ist verpflichtet, den organisatorischen Treffen nachzukommen oder sich über den besprochenen Inhalt im Nachhinein zu informieren. Sollte kurzfristig ein Gruppenmitglied den Termin nicht wahrnehmen können, so kann er ggf. per Skype-Konferenz an der Besprechung teilnehmen. Jeder Programmierer trägt selber die Verantwortung für die Dokumentation des Quellcodes.

5 Quellen

Internet:

[1]

<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Management-der-Systementwicklung/Software-Qualitätsmanagement/Qualitätsmerkmale-von-Software/index.html> aufgerufen: 16.01.2016

[2] <http://www.duden.de/rechtschreibung/Interoperabilitaet> aufgerufen: 16.01.2016

[3] <https://de.wikipedia.org/wiki/Software-Zuverl%C3%A4ssigkeit> aufgerufen: 16.01.2016

[4] http://www.inf.fu-berlin.de/inst/ag-se/teaching/V-AWS-2011/42_Benutzbarkeit.pdf
aufgerufen: 16.01.2016

[5]

<https://books.google.de/books?id=jq0Jhnl-TwkC&pg=PA606&lpg=PA606&dq=definition+wartbarkeit&source=bl&ots=b1gYImtLQC&sig=I9-5tsM8mGaVATuncQw8wth5Tyl&hl=de&sa=X&ved=0ahUKEwjVtp6N7q7KAhWHWBQKHRNhChYQ6AEIQzAF#v=onepage&q=definition%20wartbarkeit&f=false> aufgerufen: 16.01.2016

[6] <https://de.wikipedia.org/wiki/Softwaretest> aufgerufen: 16.01.2016