Qualitätssicherungskonzept

Dokumentationskonzept

Der Quelltext der Software und möglicher automatisierter Tests soll in angemessenem Umfang vollständig dokumentiert werden. Dies umfasst die Dokumentation externer Schnittstellen (API-Dokumentation) mit Hilfe eines Dokumentationsgenerator, Darstellung für das Verständnis der externen Schnittstelle notwendigen Konzepte und die Dokumentation nicht sofort verständlicher Programmteile. Die Dokumentation der für das Verständnis der externen Schnittstelle notwendigen Konzepte wird je nach Eigenschaften des Dokumentationsgenerators direkt in die Schnittstellendokumentation eingebettet oder in die technische Dokumentation ausgelagert.

Bei der Implementierung der Software sind allgemein übliche Qualitätsregeln zu beachten und die innerhalb der Entwicklergemeinschaft der jeweiligen Programmiersprache verbreitetsten Konventionen zur Formatierung (Einrückung, Schreibweise der Bezeichner, Kommentarzeichen) und Quelltextdokumentation (Docstrings, Annotationen, Dokumentationssyntax) zu benutzen, um bestmögliche Annahme der Software zu gewährleisten und die Einstiegshürde zu senken. Sofern möglich wird die Einhaltung dieser Konventionen automatisch bei Erfassung und Übertragung von Änderungen im Versionskontrollsystem überprüft und bei Nichteinhaltung abgelehnt.

Für die in Vala zu implementierenden Teile der Software wird Valadoc[1], für die in Java zu implementierenden Teile wird Javadoc[2] als Dokumentationsgenerator für die Dokumentation externer Schnittstellen (API-Dokumentation) verwendet. Als Richtline für die Quelltextformatierung dienen für Vala in geschriebenen Quelltext die für den Vala-Compiler geltenden Formatierungskonventionen[3] und die *Code Conventions for the Java Programming Language*[4] für in Java geschriebenen Quelltext.

Neben der Dokumentation des Quelltexts wird eine technische Dokumentation der Software erstellt, die in einem Wiki, das in einem Versionskontrollsystem verwaltet wird, abgelegt wird. Die technische Dokumentation beinhaltet eine genaue Dokumentation der Konzepte der Software aus technischer Sicht und vermittelt einen Überblick über die Software, so dass Entwickler sich schnell orientieren und Änderungen durchführen können. Da die technische Dokumentation zusammen mit der Schnitstellendokumentation sehr wichtig ist, um die Einstiegshürde zum Mitwirken an dem Projekt zu senken, aber im Gegensatz zur Schnitstellendokumentation, die zumeist in den Quelltext eingebettet ist, nicht Teil des Quelltexts ist und eher nicht mehr mit dem Quelltext übereinstimmt, muss die technische Dokumentation sorgfältig gepflegt und aktualisiert werden, damit sie einen Nutzen hat.

[1] `https://live.gnome.org/Valadoc`

[2] `http://www.oracle.com/technetwork/java/javase/documentation/index-jsp-135444.html`

[3] `https://live.gnome.org/Vala/Hacking#Coding_Style`

[4] `http://www.oracle.com/technetwork/java/codeconv-138413.html`

Spätestens nach Abschluss der Implementierungsphase im Projekt wird eine Benutzerdokumentation im Form eines Wiki angelegt, die Anleitungen zur Installation und Benutzung wichtiger Funktionen der Software beinhalten soll. Es ist vorstellbar diese wie bei vielen Projekten üblich in dem gleichen Wiki mit der technischen Dokumentation abzulegen, da so die Einstiegshürde gesenkt wird und erfahrene Benutzer schnell zu Entwicklern werden können.

Die Änderungsbeschreibungen (*commit messages*) müssen detailierte Änderungsbeschreibungen enthalten, die ein Verständnis der durchgeführten Änderungen ohne Betrachtung der Änderungen selbst ermöglicht. Änderungsbeschreibungen, die nur die obligatorische einzeilige Zusammenfassung enthalten, werden abgesehen von trivialen Änderungen nicht erlaubt.

Testkonzept

Sämtliche Teile des Quelltexts werden automatisch getestet, wobei eine vollständige Abdeckung aller Programmzeilen des Quelltexts angestrebt wird. Die Tests und die gestesteten Quelltextteile werden dabei von separaten Personen erstellt, so dass keine impliziten Annahmen aus der Implementierung über die zu testenden Quelltextteile getroffen werden können. Es werden sowohl Modultests, Integrationstests, Systemstests und Regressionstests durchgeführt.

Die Automatisierung wird in zwei Formen angestrebt: Automatischen zufallsgesteuerte Tests, bei denen die Eingabedaten anhand eines Generators erzeugt werden und die Ausgabedaten auf bestimmte Eigenschaften überprüft werden, Tests mit festen Ein- und Ausgaben.

Für den Test des in Java geschriebenen Quelltexts wird JUnit[5] und eine QuickCheck Variante für Java verwendet und für die in Vala geschriebenen Teile wird Valadate[6] und dogtail[7] eingesetzt.

Organisatorische Festlegungen

- Sämtlicher Quelltext wird als Freie Software veröffentlicht und die Versionshistorie öffentlich einsehbar sein. Alle im Rahmen des Praktikums erstellten Dokumente und Dokumentationen werden veröffentlicht.
- Das aufgeführte Dokumentationskonzept ist umzusetzen. Absichtliche oder mehrmalige Verletzung der aufgestellten Regeln führen zu einem Gespräch der Gruppe mit dem Betreffenden.
- Als Verantwortlicher für die Dokumentation ist Robert Rößling mit der Koordination, Erstellung und dem Zusammentragen der technischen Dokumentation und der Benutzerdokumentation unter Mithilfe aller Beteiligten beauftragt. Er auch die Qualität der von ihm und anderen Projektmitgliedern erstellten Dokumentation überwachen und den Forderungen von Herrn Prof. Gräbe anpassen.
- Matthias-Christian Ott ist bis auf weiteres verantwortlich für die Einhaltung der den Quelltext, LaTeX-Dokumente und die Änderungesbeschreibungen betreffende Konventionen.

[5] `http://www.junit.org/`
[6] `https://gitorious.org/valadate`
[7] `https://fedorahosted.org/dogtail/`

- Bei der Entwicklung grafischer Benutzeroberflächen sind die *GNOME Human Interface Guidelines* einzuhalten.
- Als Build-System wird vorerst unter Vorbehalt waf[8] verwendet. Bei größeren Problem ist jedoch ein Umstieg auf Autotools[9] und Apache Ant[10] vorgesehen.
- Matthias-Christian Ott wird die Infrastruktur einrichten und zusammen mit Simon Chill die Einhaltung der Tests überwachen.

[8] `https://code.google.com/p/waf/`

[9] `https://secure.wikimedia.org/wikipedia/en/w/index.php?title=GNU_build_system&oldid=422097679`

[10] `https://ant.apache.org/`