LinkedGeoData.org

— Qualitätssicherung —

Von: Lydia Lotzmann (Dokumentation)

Dennis Konrad (Testkonzept)

Abgabe: 27. Januar 2012

Gruppe: SWP12-11

Inhaltsverzeichnis

1 Dokumentationskonzept

Eine gute Dokumentation trägt wesentlich zur Qualität eines Softwareproduktes bei. Sie verkürzt die Einarbeitungszeit projektfremder Entwickler in den Quellcode und erleichtert damit die Wartung und Weiterentlicklung der Software.

1.1 Kommentare

Es wird direkt während der Programmierung kommentiert, damit der Quelltext verständlich ist und wichtige Details nicht verloren gehen. Zu Beginn jeder Funktion steht eine kurze Beschreibung. Außerdem werden die übergebenen Parameter und die Rückgabewerte festgehalten. Bei Bedarf können weitere Tags angegeben werden.

Bsp:

```
1/**
2 * kurze Beschreibung
3 * @param  uebergebene Werte
4 * @return return Wert
5 * @TAG    weitere Tags bei Bedarf
6 */
```

Algorithmen und sich nicht selbsterklärende Codeblöcke sind durch Kommentare zu erklären.

Bsp:

```
1/** Beschreibung */
2 Code
```

Variablendeklarationen und Kontrollstrukturen sind zu erläutern.

Bsp:

```
1 Variable, Code    //Erklaerung
```

1.2 Einrückung

Um eine gute Lesbarkeit zu sichern, wird jeder Unterbefehl im Rumpf einer Funktion eingerückt. Einrückungen haben eine Länge von vier Leerzeichen. Des Weiteren werden öffnende und schließende geschweifte Klammern des Funktionsrumpfes auf separate Zeilen gesetzt und jeder Befehl steht in einer extra Zeile.

Bsp:

```
1function functionname(varX)
2{
3....if (condition)    // '.' steht fuer ein Leerzeichen
4    {
5    ....then
6    }
7}
```

1.3 Bezeichner

Bei der Implementierung wird die Regel der Verbalisierung eingehalten. Es werden also sprechende Namen für Variablen und Funktionen verwendet. Als einheitliche Bezeichnungssprache ist Englisch festgelegt.

Für die Namensgebung gelten außerdem folgende Regeln:

- Variablen- und Funktionsbezeichner werden klein geschrieben
- Bei zusammengesetzten Bezeichnern wird der Anfangsbuchstabe jedes neuen Wortes groß geschrieben
- Konstantenbezeichner bestehen komplett aus Großbuchstaben

1.4 Dokumentation

Zur Dokumentation wird JSDoc verwendet und die dafür festgeschriebene Syntax eingehalten.

1.5 Externe Dokumentation

Die externe Dokumentation ermöglicht es, das Programm unabhängig vom Quellcode zu verstehen. Daher wird eine Designbeschreibung und ein Benutzerhandbuch erstellt und diese im Verlauf des Projektes aktuell gehalten.

2 Testkonzept

2.1 Allgemeines

Um die Funktionalität der Software sicherstellen zu können, kommt das Test Framework JS-Test-Driver zum Einsatz. Es ermöglicht jedem Programmierer schon während der Implementierung ausgiebig Tests zu schreiben und diese durchzuführen. Dieses Testing unterteilt sich in folgende Phasen:

- Komponententests
- Integrationstests
- Systemtests

Wenn ein Problem identifiziert wird, kann somit sehr schnell zurückverfolgt werden, welche Komponente es verursacht hat. Die Organisatorischen Festlegungen dienen als Anleitung zur Behebung der gefundenen Probleme. In den Gruppeneigenen Projektmanagement-Tools wird festgehalten, wie das Problem entstanden ist und wie es behoben wurde. Dadurch wird vermieden, dass Fehler häufig wiederholt werden und es entsteht eine Dokumentation der durchgeführten Tests und den dabei aufgetretenen Fehlern.

2.2 JS Test Driver

JS-Test-Driver ist ein Betriebssystemunabhängiges Test Framework. Es ermöglicht Tests ohne lange Wartezeiten aus der Sicht des Benutzers sowie des Programmierers. Zusätzlich werden die Testsituationen in allen gebräuchlichen Browsern durchgeführt. Dadurch eignet es sich außerordentlich gut, um während der Implementierung den Qualitätsstandart der entstehenden Software hoch zu halten.

2.3 Komponententests

Die Durchführung von Unit-Tests stellt sicher, dass Programmierfehler frühzeitig behoben werden können. Jeder Programmierer schreibt Tests für die von ihm entworfenen Komponenten selbst und erhält dadurch ein sofortiges Feedback, ob die von ihm angestrebten Funktionalitäten erreicht werden konnten. Nur voll funktionsfähige Komponenten werden der Gruppe zum Peer Review bereitgestellt.

2.4 Integrationstests

Aufbauend auf das Testing der Komponenten finden Integrationstests statt. Diese stellen sicher, dass bereits funktionale Komponenten in der gewünschten Weise miteinander interagieren. Getestet wird nachdem neue Module eingefügt oder verändert werden.

2.5 Systemtests

Regelmäßig wird aus der Anwender- und auch aus Entwicklerperspektive ein Test aller bestehenden Komponenten des Softwareprojekts durchgeführt. Dies gewährleistet ein optimales Zusammenspiel der bestehenden Software und weist auf eventuelle Probleme im weiteren Verlauf des Projektes hin. Die Tests werden automatisiert durch das Test Framework ausgeführt.

3 Organisatorische Festlegungen

Die Gruppenmitglieder machen sich mit den festgeschriebenen Dokumentationsrichtlinien vertraut und setzen diese bei ihrer Arbeit um. Jeder Programmierer überprüft die Einhaltung der Richtlinien selbstständig. Um die Umsetzung zu garantieren kontrolliert der Verantwortliche für Dokumentation und Qualitätssicherung regelmäßig den Code.

Regelmäßige Gruppentreffen, ein Projektmanagement-Tool (redmine) und die Kommunikation über einen Mailverteiler gewährleisten, dass jedes Teammitglied über die aktuelle Situation des Projektes informiert ist.

Treten während der Entwicklung Fehler auf, so werden diese im von der Gruppe genutzen Projektmanagement-Tool beschrieben und mögliche Ursachen dokumentiert, um ähnliche Fehler im weiteren Verlauf zu vermeiden.