

Dokumentationskonzept

geoX

Gruppe: geo10

Dokumentationsverantwortlicher:
Hannes Niederhausen

9. Juni 2003

1 Einleitung

Wie wir ja alle wissen, ist die Dokumentation eines Softwareproduktes nicht zu vernachlässigen. Aus diesen Grund sollen wir eine Designbeschreibung, sowie gut dokumentierten Quellcode produzieren. Die Designbeschreibung sollte *eigentlich* so bestehen bleiben, wie sie jetzt ist, doch ist uns allen bekannt, das es im Laufe einer Programmentwicklung zu kleineren, oder größeren Veränderungen kommt. Und wie wir die Dokumentation dieser behandeln, beschreibe ich in den folgenden kurzen Abschnitten.

2 Veränderung der Designbeschreibung

Wie oben beschrieben ist die Designbeschreibung in der Version 1 vorhanden. Doch kann es im Laufe der Entwicklung doch dazu kommen, daß Schwachpunkte im derzeitigen Design gefunden werden, die eine Veränderung bedürfen.

Wie sollt ihr nun vorgehen, falls ihr meint das Design sollte geändert werden?

1. Schreibt eine Mail an alle Mitglieder der Gruppe. Haltet euch an folgenden Aufbau:
 - (a) Benennung des Problems
 - (b) Ort der Änderung (Paket und Klasse(n))
 - (c) Begründung der Änderung, d.h. Beschreibung eines Beispiels warum das vorhergehende Konzept nicht funktioniert
2. Danach besprechen wir die Änderung, entweder per ICQ oder persönlich in einem Treffen. Sollte jemand schon beim Lesen der Änderung auf Probleme stoßen oder einen anderen (besseren) Lösungsweg finden, sollte er am besten gleich eine Mail schreiben.
3. Ist die Änderung abgesegnet wird sie in das UML-Diagramm eingetragen und die Designbeschreibung aktualisiert.

3 Veränderungen bei einzelnen Klassen

Das Programm ist in drei Pakete geteilt, Model, Controller und View. Im Design wird das unter anderen dadurch deutlich, daß wir für jedes Paket eine mdl-Datei angelegt haben. Mit dem Projektleiter (Heiko Kern) ist abgesprochen, das jeweils ein Paar an einem Paket arbeitet. Sollte es also dazu kommen, daß eine Klasse noch eine weitere Methode oder ein weiteres Attribut benötigt, so kann das Paar, die Änderungen vornehmen und danach die aktualisierte Datei auf den CVS Server stellen. Dafür wird extra ein „doc“ Modul angelegt. Durch eine Mail an alle Gruppenmitglieder werden die anderen informiert, das die mdl-Datei verändert wurde und man sie vom CVS Server ziehen kann. Da pro Paarzyklus jeweils ein Paar an einem Paket areitet sollte es danke der Aufteilung der Dateien zu keinen Überschneidungen kommen.

4 Die JavaDoc und die Quellcodedokumentation

Nun noch ein paar Worte zur Quellcodedokumentation:

Zu erst sei gleich gesagt: Lieber einhundert Kommentare zu viel als einer zu wenig. Also dokumentiert selbst die selbstverständlichsten Sachen. Es wird nicht schaden.

Einige Sachen werden schon allein wegen der JavaDoc kommentiert müssen. Das sind dir Klassen,

ihre Methoden und ihre Attribute. Bei jeder Methode und Klasse möchte ich neben der Beschreibung eine Version und die Autoren. Sollte etwas an einer Funktion geändert werden (Fehlerkorrekturen sind dabei ausgeschlossen) möchte ich das die Versionsnummer erhöht wird (um 0.1) und eine kurze Beschreibung, was geändert wurde. Diese Beschreibung beginnt auf einer neuen Zeile mit einem fettgedruckten „Änderung:“ am Ende der ursprünglichen Dokumentation.

Wird der aktuelle Quellcode auf den CVS-Server „committed“, wird die aktuelle javadoc ebenfalls mitgeschickt. Damit ersparen wir uns nach jedem „checkout“, die Erstellung der aktuellen Dokumentation.