Dokumentationskonzept

(überarbeitete Version)

Die Qualität einer Software ist von einscheidender Bedeutung. Das Dokumentationskonzept dient der einheitlichen Festlegung von Qualitätssicherungskriterien. Das Erstellen der Dokumentation parallel zur Programmentwicklung minimiert den zeitlichen und somit auch den finanziellen Aufwand eines Softwareprojektes. Eine gut dokumentierte Software erlaubt es neuen Mitarbeitern und Programmierern eines Projektes sich schnell einarbeiten zu können, weil der Quellcode leicht lesbar und verständlich ist und zusätzliche Informationsquellen verfügbar sind. Weitere Ziele sind ein hoher Grad an Widerverwendbarkeit und leichtere Änderbarkeit.

1. Interne Dokumentation - Quelltextnahe Dokumentation

Eine gute Dokumentation erleichtert eine Einarbeitung in den Quellcode für Projektfremde oder nach längerem Projektstillstand, aufgrund besserer Lesbarkeit. Zu beachten sind dabei in unserem Projekt grundsätzlich die unter http://java.sun.com/docs/codeconv/CodeConventions.pdf zu findenden "Java Code Conventions".
Unter anderem sind folgende Prinzipien umzusetzen:

Javadoc:
Der Quellcode enthält javadoc-tags aus denen später die Dokumentation mittels javadoc generiert wird. Insbesondere sind Klassen und Methoden zu beschreiben.

Die folgende Tabelle gibt einen Überblich über wesentliche Tags:

Tag & Parameter	Ausgabe	Verwendung in
@author name	Beschreibt den Autor.	Klasse, Interface
@version version	Erzeugt einen Versionseintrag. Maximal einmal pro Klasse oder Interface.	Klasse, Interface
@since jdk-version	Seit wann das Feature existiert.	Klasse, Interface
@see reference	Erzeugt einen Link auf ein anderes Element der Dokumentation.	Klasse, Interface, Instanzvariable, Methode
@param name description	Parameterbeschreibung einer Methode.	Methode
@return description	Beschreibung des Returnwerts einer Methode.	Methode
@exception classname description @throws classname description	Beschreibung einer Exception, die von dieser Methode geworfen werden kann.	Methode
{@inheritDoc}	Kopiert die Beschreibung aus der überschriebenen Methode	Überschreibende Methode

Jeder Klasse ist folgende Information voranzustellen:

```
/**
 * Diese Klasse erzeugt eine Kundenliste
 * @author Anja Krabbes
 * @version 1.0 / 12.05.2006
 */
```

Namensgebung:
Namenskonventionen für Klassen, Methoden, Variablen etc. garantieren die verbesserte Lesbarkeit aufgrund "sprechender" Bezeichnungen.

-Klassen:	NameDerKlasse		z.B.	TestPortlet
-Attribute:	nameDesAttributes	(Substantiv)		anzahlSpalten
-Methoden:	nameDerMethode	(Verb & ggf. Object)		openTable

Kommentare:
Das Einfügen von zusätzlichen Kommentaren an geeigneten Stellen unterstützt die Verständlichkeit.

Gliederung des Quellcodes:
Durch Einrückungen und Strukturierungen insbesondere von Klammern und dazugehöriger Strukturen wird der Quellcode überschaubar. Beispielhaft sieht dies wie folgt aus:

```
if (Bedingung)
        Anweisung1;
else
        Anweisung2;
```

Testfälle:
Erstellung und Kommentierung der Testfälle (Eine genauere Beschreibung des Testkonzeptes erfolgt in der Testdokumentation.)

2. Externe Dokumentation - Entwurfsdokumentation

Die Dokumentation für die Endanwender (User, Kunden) basiert auf der Designbeschreibung und dem Pflichtenheft, in der die Funktionalität des zu entwickelnden Softwareproduktes spezifiziert wurden.

Designbeschreibung:
Diese dokumentiert und begründet die getroffenen Designentscheidungen und bezieht sich vor allem auf die verwendeten OOD Konzepte und die sich daraus ergebende Paket- und Klassenstruktur, erläutert unter anderem durch UML-Diagramme. Vorangestellt ist eine Produktübersicht.

Benutzerhandbuch/Online-Hilfe:
Diese Dokumentation ist für den Endbenutzer bzw. Anwender des Software-Produktes vorgesehen. Das Benutzerhandbuch – ggf. in Kombination mit oder in Form einer Online-Hilfe – entsteht parallel zur Entwicklung der Software selbst.
Alle Funktionen der Anwendung werden ausreichend beschrieben und erklärt. Außerdem sollte es detaillierte Informationen über Systemvoraussetzungen, Installation und Handhabung des Programms beinhalten. Ebenfalls ist das Vorhandensein einer Begriffsübersicht mit zugehörigen Erläuterungen zu beachten, für die Begriffe des Glossars herangezogen werden können.

3. Verantwortlichkeiten:

Die Verantwortung für die interne Dokumentation liegt in den Händen der Implementierer. Die Dokumentation entsteht zeitlich parallel zum Programm.
Verantwortlich für die Entwurfsdokumentation sind alle mit den Merkmalen (Struktur- und Entwurfs-Prinzipien und -Entscheidungen) der entstehenden Software vertrauten Teammitglieder.

Die Überwachung der Einhaltung der Qualitätsstandards obliegt dem Verantwortlichen für Qualitätssicherung und Dokumentation.