

Dokumentationskonzept

1. Einleitung

Die Dokumentation ist ein wichtiger Bestandteil jeder Software und in der Einarbeitungsphase in ein Softwareprodukt nicht wegzudenken. Dabei werden alle Schritte und Maßnahmen detailliert dokumentiert.

- Das Prinzip der integrierten Dokumentation
 - reduziert den Aufwand zur Dokumentenerstellung
 - stellt sicher, dass keine Informationen verloren gehen
 - garantiert die rechtzeitige Verfügbarkeit der Dokumentation
 - erfordert die entwicklungsbegleitende Dokumentation

- Eine gute Dokumentation bildet die Voraussetzung für
 - leichte Einarbeitung in ein Produkt bei Personalwechsel oder durch neue Mitarbeiter
 - gute Wartbarkeit des Produkts

2. Bestandteile

Das zu entwickelnde Softwareprodukt unterliegt einer Dokumentation, die sich aus folgenden Aspekten zusammensetzt:

2.1 Externe Dokumentation:

Die Externe Projektdokumentation subsumiert alle Dokumente, die in gedruckter oder elektronischer Form für den Kunden zugänglich gemacht werden. Einzelne Bereiche der internen Dokumentation können für die Erstellung der externen Dokumentation verwendet bzw. auf Anfrage dem Kunden bereitgestellt werden.

2.1.1 Design-Beschreibung

Die Software-Design-Beschreibung ist eine Repräsentation eines Software-Systems, die verwendet wird, um Software-Design-Informationen zu kommunizieren. Es umfasst wesentliche Teile der Development-Requirements und beschreibt die Architektur der Software, beziehungsweise des Gesamtsystems und der einzelnen Komponenten. Zusammenhänge im Gesamtdesign sollten in der Designbeschreibung näher erklärt werden und regelmäßig vom Projektleiter aktualisiert werden.

2.1.2 Handbuch

Zu jedem Soft-Produkt gehört eine adäquate, vollständige und fehlerfreie Handbuch-Dokumentation. Im Bereich von Software wird die Benutzung eines Programms beschrieben, ohne auf die Details der Implementierung einzugehen, insofern ist es nur ein Teil der technischen Dokumentation. Es richtet sich als Hilfestellung vor allem an den Benutzer. Es sollte deshalb leicht zugänglich und verständlich

sein, den Benutzer mit der Software vertraut machen und ihm sowohl erste Schritte ermöglichen, als auch einen Überblick über alle verfügbaren Funktionen liefern.

2.2 Interne Dokumentation:

In der internen Dokumentation sollen sich neben nützlichen Informationen über den Quellcode vor allem die Prinzipien der Implementierung (Verbalisierung, problemadäquate Datentypen, Verfeinerung) erkennen lassen.

2.2.1 Javadoc

Javadoc ist ein automatisiertes Software-Dokumentationswerkzeug, das aus Java-Code automatisch HTML-Dokumentationsdateien erstellt. Mit Hilfe von Javadoc-Kommentaren, beginnend mit `/**` und endend mit `*/`, können Beschreibungen für Interfaces, Klassen, Methoden, und Felder über spezielle Javadoc-Tags (beginnend mit `@`) definiert werden. Es stehen verschiedene tags zur Verfügung, womit z.B. Parameter und Rückgabewerte von Funktionen entsprechend behandelt werden können.

Zur Dokumentation verwenden wir unteranderen folgende Tags:

Tag	Beschreibung	Ort
<code>@author</code>	der Autor der Klasse, des Interfaces	vor jeder Klasse/Interface
<code>@version</code>	die aktuelle Versionsnummer und das Datum der letzten Änderung	vor jeder Klasse/Interface
<code>@param</code>	bezeichnet die Parameter und ihre Beschreibung	vor einer Methode
<code>@return</code>	bezeichnet den Rückgabewert und seine Beschreibung	vor einer Methode
<code>@exception</code>	Beschreibung der Exception	vor einer Methode
<code>@see</code>	Verweis auf ein anderes Dokument	vor einer Methode

Neben der externen Dokumentation mit Javadoc verwenden wir `/**` für Zeilen. Damit die Dokumentation gewährleistet wird, wird der Verantwortliche für Dokumentation in regelmäßigen Abschnitten den Quellcode auf ausreichende Dokumentation prüfen.

2.2.2 QuellCode Convention

Als Code Conventions bezeichnet man einen Satz von Regeln, nach welchen der Quellcode einer Programmiersprache formatiert wird. Die Quellcodedokumentation ist ein unabdingbarer Bestandteil des Programmierprozesses. Der Programmierer kommentiert den von ihm geschriebenen bzw. modifizierten Quellcode, so dass dessen Änderung zu einem späteren Zeitpunkt ohne großen Zeitaufwand geschehen kann.

Es müssen dokumentiert werden:

- Klassen
- Attribute bzw. –Methoden
- Übergabe- und Rückgabeparameter
- Variable, Konstanten, Typen

Es müssen stets sinnvolle, aussagekräftige und selbsterklärende Bezeichnungen verwendet werden, falls dies die Lesart des Quellcodes erleichtern würde. Dies beinhaltet auch die Einführung von

Konstanten für unverständliche bzw. mehrdeutige Werte oder Parameter. Bei der Verwendung der „If“, „Switch“ oder äquivalenter Befehle muss jede der möglichen Varianten stets mit einem Kommentar versehen werden der die zutreffende Bedingung verständlich beschreibt. Bei längeren Sichtbarkeitsblöcken muss dessen Ende durch Angabe des zugehörigen Anfangs bekannt gegeben werden.

2.2.3 Kommentierung der Testfälle

Die Kommentierung der Testfälle erfolgt intern durch die Quellcode und die Javadoc – Dokumentationen. Alle Ergebnisse der Komponenten-, Integrations-, System- und Abnahmetests werden dokumentiert. Alle Klassen und Methoden werden getestet. Das vorliegende Testkonzept wird in die laufenden Projektarbeiten einbezogen. Der Kunde kann alle Aspekte im Bezug auf Tests im Testkonzept nachlesen.