

Dokumentationskonzept

Verantwortlicher: Peter Scholz

Gruppe: ueb13

5. Juni 2003

1 Einleitung

Im Kapitel zur Produktentwicklung werden in der Norm ISO 9001 die Rahmenbedingungen geschaffen, wie ein Produkt (in unseren Fall ein Softwareprodukt) normgerecht zu entwickeln ist.

In der Produktentwicklung wird dabei auf zwei Hauptaspekte besonderer Wert gelegt, daß ist zum einen *die Planung* und zum anderen *die Entwicklung*. Im Rahmen des Dokumentationskonzeptes wird nur kurz auf beide Aspekte eingegangen um den Zusammenhang zur Dokumentation herzustellen.

Planung Zu Beginn des Projektes muß definiert werden, was überhaupt zu erledigen ist, dies geschieht in Form eines Lastenheftes, woraus dann das Pflichtenheft resultiert. Dabei sind im *Dialog mit dem Kunden* die Spezifikationen genau festzuhalten.

Weiterhin muß die Entwicklung des Produktes geplant und bspw. Deadlines für Meilensteine (Wochenscheiben, etc.) gesetzt werden.

Entwicklung Die eigentliche Entwicklung des Produktes beginnt mit dem Entwurf anhand der Spezifikation, dabei müssen alle Funktionen aus der Spezifikation abgedeckt sein. Weiterhin muß bereits der Entwurf dokumentiert und vor der Implementierung verifiziert werden.

Während der Implementierung ist darauf zu achten, daß der Code kommentiert wird und die Kommunikation mit dem Kunden geregelt ist. Die Kommunikation kann dabei mittels Reviews erfolgen, diese müssen protokolliert werden und der Entwicklungspro-

zess kann erst dann fortschreiten, wenn alle Fehler beseitigt sind.

Weiterhin müssen „Mittel zur Kennzeichnung und Rückverfolgbarkeit“ bereitgestellt werden, in der ISO 9000-3 wird dies als Konfigurationsmanagement bezeichnet. Dies gilt insbesondere nicht nur für den Programmcode, sondern für alle Dokumente des Projektes (Verträge, Entwicklungsplan, Designbeschreibungen, etc.).

Es muß darauf geachtet werden, daß alle Versionen jeder Softwarekomponenteeindeutig zu identifizieren sind, Änderungen zu verfolgen und die Arbeit zu koordinieren.

Ein weiterer Aspekt ist die Betrachtung der Dokumentation nicht nur als Hilfsmittel zur Einarbeitung des Softwareentwicklers in den Quellcode, sondern auch als Hilfsmittel für den Anwender/Kunden um korrekt – im Sinne der Spezifikation – mit dem Produkt arbeiten zu können.

2 Umsetzung

Allgemein sollte mit der Dokumentation, sei es für den User, sei es für den SW-Entwickler, bereits sehr früh begonnen werden, d.h. bereits in der Definitions- und Entwurfsphase (Designentwurf), um zum einen den Arbeitsaufwand gering zu halten, und zum anderen auch einen Überblick zu haben, in welcher Phase der Entwicklung man sich befindet und ob man im gesetzten Zeitrahmen ist, daraus ergibt sich eine integrierte Dokumentation, die parallel zum Entwurf und zur Implementierung entsteht, sie wächst dynamisch mit dem Projekt mit.

Dabei werden die entsprechenden Dokumentationen dynamisch dem jeweiligen Entwicklungsstand angepasst. Dies kann automatisch geschehen oder auch manuell, wobei hier im konkreten Fall ein Mischung aus beiden zum Einsatz kommen wird.

Ein hervorragendes Beispiel für eine vollständig automatisch generierte Dokumentation ist CLS, ein Optimierungswerkzeug in der Abteilung Technische Informatik. Hier wurde pro Modul eine Readme-Datei geschrieben, als Funktionsbeschreibung für den Anwender und entsprechende Quelldokumentation. Diese werden zur Compilationszeit ausgewertet und eine entsprechende Dokumentation, angepaßt an die complierten Module erstellt, so daß sich der User nicht mit allen Funktionen konfrontiert sieht, sondern nur mit denen die ihm auch zur Verfügung stehen. Dieser Ansatz basiert auf der Benutzung des L^AT_EX-Textsatzsystemes, welches für solche Zwecke sehr gut geeignet ist, ähnlich anderen Ansätzen wie XML, HTML, etc. bieten diese durch ihren strukturierten Aufbau die Möglichkeit des leichten generierens von Dokumenten (siehe hier auch entsprechende Papers zum Thema Workflow-Management mit XML).

Konkret soll sich, da es ein Projekt in Java wird, an die „Code Conventions for the JavaTM Programming Language“ gehalten werden, die unter der URL <http://java.sun.com/docs/codeconv/CodeConvTOC.doc.html> zu finden sind.

Jedem Team-Mitglied wird ein Leitfaden und die entsprechenden Schablonen bereitgestellt (Team intern), in dem geregelt ist, wie der Quellcode zu dokumentieren ist und wie die Doku-Datei zu erstellen ist. Näheres dazu in den entsprechenden Abschnitten.

2.1 User Dokumentation

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

Hierbei soll darauf Wert gelegt werden, daß die Dokumentation nicht eine Auflistung der Funktionen oder Features ist, sondern vielmehr sollen diese Funktionen/Features DAU-kompatibel erklärt werden, so

daß es dem Anwender ermöglicht wird, nur mit der Softwaredokumentation sich in das Produkt einzuarbeiten und sich die volle Funktionalität zu erarbeiten.

Dabei sollte sich eine solche Dokumentation nicht unbedingt nur auf Papierbasis beschränken, sondern um einen guten Arbeitsablauf zu ermöglichen sollte auch eine Online-Hilfe, so wie sie in jedem UNIX-System über die man-Pages vorhanden ist, existieren, wobei die man-Pages nur die einfachste Form, dafür aber garantiert auch nutzbar ist.

Umsetzung der Anforderungen:

- Erstellung eines ASCII-8bit-Dokumentes namens `modulname.txt` in dem jede implementierte Funktion dokumentiert wird (Aufruf - passiert), es werden entsprechende Schablonen bereitgestellt
- Generierung der Doku aus den entsprechenden ASCII-Dateien

2.2 Entwicklerdokumentation

Die Dokumentation für Entwickler basiert zum überwiegenden Teil auf den Quellcodekommentaren, d.h. für diese, daß sie an Stellen im Quellcode einzusetzen sind an denen dieser nicht selbsterklärend ist, bzw. schwierig nachzuvollziehende Zusammenhänge zu respektive zwischen anderen Funktionen, Modulen, etc. bestehen.

Weiterhin werden Kommentare (Tags) im javadoc-Style gesetzt um dieses Werkzeug effektiv nutzen zu können, hier ist es sehr wichtig, bereits während der Implementierungsphase entsprechende Tags einzubinden, bzw. Schablonen für die Quelldateien zu erstellen mit den üblichen Informationen (Autor, Datum, Funktion, Klasse, etc.).

Umsetzung der Anforderungen:

- Bereitstellung entsprechender Schablonen für Quellcode-Dateien
- Generierung der Doku mittels javadoc
- Verwendung der Designbeschreibung zur Einführung in das Design, die Paket- und Klassenstruktur und die statischen und dynamischen Aspekte des Modells