

## Dokumentationskonzept

Das Dokumentationskonzept dient der Spezifikation von Dokumentationsstandards und damit der Qualitätssicherung des gesamten Projektes. Vorteile durch die Festlegung solcher Kriterien ergeben sich sowohl für das Softwareteam (intern) als auch natürlich für den Kunden (extern).

Für den Kunden wird für die auf ihn zurechtgeschnittene Dokumentation ersichtlich, ob in erster Linie das entstehende bzw. fertige Produkt seinen Wünschen gerecht wird. Natürlich gibt es ihm und den zukünftigen Nutzern auch die entscheidenden Hinweise zur Benutzung der Software bzw. Oberfläche.

Für das Team wird in erster Linie der Entwicklungsprozess durch eine gescheite Dokumentation beschleunigt. Bereits gemachte Erfahrungen können außerdem besser ausgewertet werden. Ein weiterer Aspekt ist die Wartung und Pflege des Softwareproduktes, die mit einer guten Dokumentation einfacher und übersichtlicher gemacht wird.

### 1. Interne Projektdokumentation

#### a) phpDoc

Beim Programmieren, unabhängig von der verwendeten Sprache, ist nicht nur die Funktionalität und Effizienz (Qualität) des Codes wichtig, sondern meist auch die Verständlichkeit und Wiederverwendbarkeit. Das liegt darin begründet, daß teilweise mehrere Programmierer an einem Projekt arbeiten, zu dem der jeweilige Quelltext gehört; auch wird oft der Code stetig erweitert und verbessert oder es müssen Fehler gefunden und behoben werden.

In jedem Fall ist es also notwendig, daß der Code übersichtlich ist. Darüber hinaus ist es jedoch unerlässlich, den Quelltext ausführlich zu kommentieren.

Akteur: Implementierer

Verantwortlich: Implementierer

#### b) einheitliche Quellcodedokumentation

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 die (sprachenabhängigen) Konventionen für das Schreiben von Quellcode befolgt werden.

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.

Akteur: Implementierer  
Verantwortlich: Implementierer

### **c) Testfalldokumentation**

Eine Beschreibung aller Tests der Komponenten und des gesamten Systems mit Angabe des Zwecks, Ablaufs, und der Ergebnisse. Dafür wird ein spezielles Testkonzept entwickelt und eingesetzt.

Akteur: Tester, Dokumentierer, Projektleiter  
Verantwortlich: Tester

## **2. Externe Projektdokumentation**

### **a) Handbuch**

Als Anwendungsdokumentation wird für den Administrator ein Administratorhandbuch und für den Nutzer ein Benutzerhandbuch erstellt. Das Benutzerhandbuch informiert über die Produktbestandteile, Arbeitsobjekte, Produktfunktionen, Produktstruktur und die Arbeitsabläufe der Software. Das Administratorhandbuch beinhaltet zusätzlich technische Details, wie z.B. Informationen über die erfolgreiche Installation der Software u.a.. Weiterhin steht dem Benutzer nach erfolgreicher Entwicklung der lauffähigen Software eine Online Dokumentation zur Verfügung, welche schwierige oder sich nicht selbst beschreibende Bereiche der Software erläutert. Beispielsweise: wie kann dem Buchbestand eine neues Buch hinzugefügt werden.

Akteur: Projektleiter, Dokumentierer, Implementierer  
Verantwortlicher: Dokumentierer

### **b) Onlinehilfe**

Das fertige Softwareprodukt beinhaltet eine Online-Dokumentation, die seine Bedienung durch die Bereitstellung von Erklärungen oder „Tooltips“ erleichtern soll. Sie ist selbsterklärend, dient der Benutzerfreundlichkeit und macht die Produkthanwendung effizienter. Für die tiefer gehenden Informationen wird das Studieren des Handbuchs empfohlen.

Akteur: Projektleiter, Ergonom, Implementierer, Dokumentierer  
Verantwortlich: Implementierer

### **c) Designbeschreibung**

Hier werden im Wesentlichen die Architektur und das Oberflächendesign beschrieben. Durch die Schilderung von Zusammenhängen von Modulen und Objekten können sich andere Personen besser in das Projekt einarbeiten. Dies ist wichtig, wenn der Kunde anderen Entwicklern zur Erweiterung sein Projekt weitergibt. Außerdem bekommt der Kunde schon früh einen Eindruck über die Eigenschaften seines Systems

Akteur: Projektleiter, Implementierer, Modellierer, Dokumentierer  
Verantwortlich: Dokumentierer