

# Qualitätssicherungs-Konzept

## 1 Dokumentationskonzept

Guten Quellcode zeichnet neben der Funktionalität ebenso eine optimale Lesbarkeit aus. Durch festgelegte Vorgehensweisen wird die Struktur und Dokumentation unseres Codes gesichert. Der produzierte Code bleibt daher zu jedem Zeitpunkt leicht verständlich, wodurch sich alle Mitarbeiter gut in fremden Quellcode einarbeiten können, und die Wartbarkeit auch auf lange Sicht gewährleistet ist. Im Folgenden werden wichtige Standards festgelegt.

**Verbalisierung:** Für Variablen, Funktionen, Klassen etc. sind aussagekräftige, problembezogene und leicht zu merkende Bezeichnungen zu wählen. Dadurch sind die Bezeichner selbsterklärend und das Programm besser lesbar. Hierzu werden englische Namen/Begriffe herangezogen.

**Kopfkommentare** Am Anfang einer jeden Quellcode-Datei sind vorgeschriebene Tags zur automatisierten Dokumentation (phpDocumentor) zu setzen. Diese sind umgehend auszufüllen, bzw. werden vervollständigt, sobald die benötigten Informationen zur Verfügung stehen. Weiterhin werden komplexe Funktionen und Methoden mit zusätzlichen Kommentaren ergänzt, um deren Funktionsweise zu erläutern.

**Einrückung und Zeilenlänge** Der Quellcode ist mittels vier Leerzeichen einzurücken. Die Verwendung eines Tabulators ist nicht gestattet, somit werden Probleme mit dem SVN-Repository (z.B. mit der Diff-Funktion) vermieden. Bei maximal 80 Zeichen wird ein Zeilenumbruch vorgenommen.

**Kontrollstrukturen** Bei der Verwendung von Kontrollstrukturen (if, for, while, ect.) sind immer geschweifte Klammern zu verwenden, auch wenn es sich um kurze (einzeilige) Aufrufe handelt. So wird logischen Fehlern vorgebeugt und die Lesbarkeit erhöht. Leerzeichen zwischen dem Kontrollausdruck und der öffnenden Klammer werden zur besseren Unterscheidbarkeit von Funktionsaufrufen verwendet.

**Funktionsaufrufe** Funktionen werden ohne Leerzeichen zwischen dem Funktionsnamen, der öffnenden Klammer und dem ersten Parameter aufgerufen. Leerzeichen werden allerdings zwischen einem Komma und jedem folgendem Parameter gesetzt. Zwischen dem letzten Parameter, der schliessenden Klammer und dem Semikolon werden keine Leerzeichen gesetzt.

**PHP-Code-Tags** PHP-Code wird immer mit „<?php“ eingeleitet und mit „?>“ beendet, um eine korrekte Funktionsweise unabhängig von Betriebssystem und PHP-Konfiguration zu gewährleisten.

**Dateiformat(Zeichensatz)** Alle Dateien werden im ASCII Format gespeichert und mit dem Zeichensatz UTF-8 kodiert.

### **phpDocumentor**

Der phpDocumentor ist das derzeitige Standardwerkzeug für die automatische Dokumentation von php-Dateien, und kann von der Kommandozeile oder über ein Webinterface ausgeführt werden, wodurch eine professionelle Dokumentation aus dem Quellcode erzeugt wird. Der so generierte Überblick fasst alle relevanten Eckdaten (Autor, Funktionen, Rückgabewerte, usw.) zusammen, wobei die Ausgabe für viele Formate (HTML, PDF, CHM, XML DocBook) unterstützt wird. Lesbar wird der Quellcode für das Werkzeug durch eine große Auswahl verschiedener Tags (zB „@author“). Dadurch das sich der Programmierer an die gängigen Tags hält, dient diese Art der Dokumentation als Hilfestellung um Vollständigkeit zu gewährleisten.

## **2 Organisatorische Festlegungen**

### **2.1 Verantwortlichkeit**

Verantwortlich für die Qualitätssicherung ist jeder Mitarbeiter des Projektes, denn bereits während der Codeerstellung sind die Implementierer dazu angehalten, ihren Code nach den oben genannten Standards zu produzieren. Der Qualitätssicherungsbeauftragte kontrolliert die Einhaltung der Vorgaben an zwei Terminen pro Woche, indem sämtlicher entstandener Quellcode überprüft wird. Darüber hinaus werden große Commits bzw. bedeutende Programmfunktionen sofort nach der Veröffentlichung geprüft.

### **2.2 Umgang mit Fehlern**

Werden in unserem Softwareprodukt Mängel entdeckt, behandeln wir diese in unserem Bugtracker. Reporter kann jeder aus dem Team sein, der einen Fehler entdeckt. Nach einem selbstständig durchgeführten Reproduktionstest (falls möglich), wird der Fehler eingetragen. Die Qualitätssicherung prüft den Report, und weist das Problem einem Entwickler beziehungsweise dem zuständigen Bereich zu. Automatisierte Zuteilungen über Voreinstellungen im Bugtracker beschleunigen die Bearbeitung zusätzlich. Nach der Bearbeitung durch den Entwickler wird das Problem von der QS erneut geprüft, und dann als behoben markiert bzw. wieder zugewiesen (wenn der Fehler noch existiert). Wir stellen somit sicher, dass keine Problem in Vergessenheit geraten, und mögliche Qualitätsmängel stets erfasst sind.