

# Dokumentationskonzept

## 1. Zielbestimmung

Ziel des Dokumentationskonzept ist einerseits, Bugfixing, Wartbarkeit und Änderbarkeit zu unterstützen. Die Einarbeitung für Projektfremde soll mit Hilfe der Dokumentation ermöglicht bzw. vereinfacht werden. Andererseits legt es Qualitätsstandard fest und soll deren Einhaltung organisieren und kontrollieren.

Es wird später von einem Testkonzept<sup>1</sup> ergänzt, das hier vorbereitet wird. Beide Teile werden in den „Qualitätssicherungs - Plan“ einfließen.

Für die Dokumentation sind nach [Balzert] von Bedeutung:

- die gute Strukturierung des Quellcodes (gute Lesbarkeit),
- eine „gute“ Dokumentation des Designs und des Quellcodes,
- Testmaterial zu einzelnen Funktionen, das alle wesentlichen Funktionen und Zweige abdeckt.

## 2. Qualitätsstandards

Es müssen Prioritäten bezüglich der Produktqualität gesetzt werden. Diese legt man in dieser Darstellung fest.

Produktqualität	Sehr gut	Gut	Normal	Nicht relevant
Funktionalität			x	
Zuverlässigkeit			x	
Benutzbarkeit	x			
Effizienz			x	
Änderbarkeit		x		
Übertragbarkeit				x

## 3. Vorgehensweisen

Vorschauprozess: Im Team sind verschiedene Personen für die Programmierung/Verfassen von Texten, für Vorschau und Tests verantwortlich. Vorbereitete Dokumente werden allen zugänglich gemacht (Mailingliste) und in einem Treffen gemeinsam bearbeitet.

Testen: Testfälle werden auf Basis von den Geschäftsprozessen erstellt, müssen von den jeweiligen Testern auch dokumentiert werden (ob, wann der Test durchgeführt wurde, ob erfolgreich oder nicht).

<sup>1</sup> JUnit Framework ist zu verwenden: <http://www.junit.org>, Komponententestfälle – Testfälle idR vor dem Programmieren (dienen während dieser Phase als Testfälle, typische abzufangende Fehlerquellen)

## 4. Verantwortlichkeiten für verschiedene Teile der Dokumentation

Quelltext und javadoc: Die interne, quelltextnahe Dokumentation liegt in den Händen der jeweiligen Programmierer, ebenso die javadocs. Die Programmierer werden von Vorschau unterstützt, die auf die Verständlichkeit des Quelltextes und der javadocs achten. Ziel ist dabei insbesondere, die Qualität der Dokumentation zu erhöhen und gleichzeitig das Anwendungswissen innerhalb des Teams auf eine breitere Basis zu stellen. Auf dieser Basis ist es auch möglich, in Diskussionen voneinander zu lernen und das Produkt voranzutreiben. Die javadoc-Dokumentation wird jeweils mit der Ergänzung neuer Funktionen/Methoden neu kompiliert, um dem Team eine zügige Weiterarbeit zu ermöglichen.

Die Verantwortliche für Dokumentation muss den Quellcode in Bezug auf die Einhaltung der festgelegten Richtlinien für das Kommentieren überprüfen. Sie ist verantwortlich für das Sammeln von Informationen um eine begleitende Dokumentation erstellen zu können. Beispielsweise ist für den Benutzer ein Handbuch zum besseren Verständnis im Umgang mit der neuen Software zu schreiben, welche nur ausführlich und nützlich sein kann, wenn es alle wichtige Informationen im Rahmen des Projektes enthält.

Schließlich hat die Verantwortliche für Dokumentation alle entstehenden Dokumente in eine einheitliche Form zu bringen. Getreu dem Motto: „Ein Bild sagt mehr als tausend Worte“ werden Grafiken genutzt, um bestimmte Sachverhalte zu veranschaulichen – z.B. Schichten- Architektur, USE CASE- Diagramme u.a..

Dem Verantwortlichen für Tests obliegt die Kontrolle des Quellcodes bezüglich richtigem und nützlichem Kommentieren und auch die Begutachtung in Hinblick auf ausreichende Schnittstellendokumentation. Dies kann ihm später großen Aufwand ersparen, da ein automatisches Testen erleichtert wird.

Der Implementierer ist dafür zuständig, dass die Richtlinien zum internen Kommentieren von Quellcode bereits während des Programmierens eingehalten werden. Ihm obliegt weiterhin die Schnittstellenbeschreibung und das Sammeln von implementierungsspezifische Daten um die Funktionsprinzipien der Software transparent zu gestalten.

So wird eine spätere Benutzerhandbucherstellung erleichtert. Diese Vorarbeit wird der Implementierer an die Verantwortliche für Dokumentation übergeben.

Die javadocs liegt in den Händen der jeweiligen Programmierer. Sie werden von Vorschauen unterstützt, die auf die Verständlichkeit des Quelltextes, ihrer Kommentierung und der damit verbundenen javadocs achten.