

# Qualitätssicherungskonzept

swp-helios

15. April 2014

## Inhaltsverzeichnis

<b>1</b>	<b>Dokumentationskonzept</b>	<b>2</b>
1.1	Code . . . . .	2
1.2	Handbuch . . . . .	2
<b>2</b>	<b>Testkonzept</b>	<b>3</b>
2.1	Allgemeines . . . . .	3
2.2	Komponententest . . . . .	3
2.3	Integrationstest . . . . .	3
2.4	Systemtest . . . . .	3
2.5	Abnahmetest . . . . .	3
2.6	Regressionstests . . . . .	3
2.7	Oberflächentests . . . . .	4
<b>3</b>	<b>Organisatorische Festlegungen</b>	<b>4</b>

# 1 Dokumentationskonzept

## 1.1 Code

### **Klammern**

Geschweifte Klammern erhalten jeweils eine neue Zeile und stehen unter der dazugehörigen Funktion. Ihr Inhalt wird eingerückt. So ist der Code weniger kompakt, aber lesbarer.

### **Tabulatoren und Leerzeichen**

Statt Tabulatoren verwenden wir jeweils 4 Leerzeichen, damit Einrückungen in jedem Editor gleich groß sind. Um Gleichheitszeichen, Doppelpunkte oder ähnliches gleichmäßig untereinander zu schreiben, werden ebenfalls Leerzeichen verwendet.

### **Kommentare**

Zur allgemeinen Verständlichkeit, müssen Kommentare ausreichend ausführlich beschreiben, was Codezeilen oder Codeblöcke tun und wie eigene Funktionen verwendet werden müssen.

### **Zeilenlänge**

Ein Zeilenumbruch wird spätestens bei 80 Zeichen durchgeführt. Sollte das Limit nur um bis zu 2 Zeichen überschritten sein, kann darauf verzichtet werden. Dies gilt auch für Kommentare.

### **Bezeichner**

Für neu definierte Funktionen, Klassen, usw., deren gewählter Bezeichner aus mehr als einem Begriff besteht, benutzen wir Großschreibung als Worttrenner. Bsp. Klasse: FooBar ; Bsp. andere: fooBar

## 1.2 Handbuch

Der entstehende Javacode soll ausführlich mittels javadoc dokumentiert werden. Für JavaScript werden ausführliche Kommentare geliefert. Ein echtes Handbuch zur Benutzung der Anwendung wird es nicht geben, jedoch werden wir eine Onlineanleitung zur richtigen Bedienung schreiben.

## **2 Testkonzept**

### **2.1 Allgemeines**

Um einen erfolgreichen Abschluss des Projektes zu garantieren, ist es essentiell notwendig ein gut durchdachtes Testkonzept zu erstellen. Primär dient es dazu frühzeitig Fehler aufzudecken und diese zu beseitigen bzw. die Fehlerquote möglichst gering zu halten.

### **2.2 Komponententest**

Der Komponententest ist für die Qualität der Software bedeutsam. Unabhängig vom Gesamtsystem wird jede Klasse und deren Funktionen/Methoden hinsichtlich auf ihre Funktionalität, Robustheit und Vollständigkeit geprüft. Dies kann in Form von Automatisierung (z.B. durch Test-Frameworks) oder von den Entwicklern selbst gewählten Testkriterien (z.B. zufällig gewählter Parameter) geschehen. Ziel ist es im Quellcode der Komponenten Anomalien rasch zu erkennen.

### **2.3 Integrationstest**

Der Integrationstest prüft das Zusammenspiel der einzelnen, voneinander abhängigen Komponenten. Schwerpunkt sind insbesondere die Schnittstellen der beteiligten Module. Die Ergebnisse geben den Entwicklern eine Übersicht über die Prozesse der Komponenten.

### **2.4 Systemtest**

In diesem Test wird ein Probedurchlauf der vollständigen Software mit allen dazugehörigen Komponenten in einer Testumgebung durchgeführt. Hierbei wird, im Gegensatz zu den anderen Tests, die Anforderung und Produktivumgebung des Kunden und des Anwenders geprüft bzw. simuliert.

### **2.5 Abnahmetest**

Der Abnahmetest ist der letzte Testlauf der gesamten entwickelten Software durch den Kunden bzw. den Auftraggeber in einer Benutzerumgebung. Das Produkt wird auf alle vorher festgelegten Anforderungen chronologisch geprüft und stets auf ihre Vollständigkeit, Nutzbarkeit, Wartbarkeit und Dokumentationsqualität untersucht. Auf diese Weise können Kompatibilitätsprobleme größtmöglich ausgeschlossen werden.

### **2.6 Regressionstests**

Bei dem Regressionstest werden bereits bestandene Testverfahren wiederholt angewendet. Aufgrund ständiger Änderung, Pflege oder Korrektur der Software

(Modifikation) müssen bereits getestete Komponenten wiederholt geprüft werden.

## **2.7 Oberflächentests**

Oberflächentests werden manuell mit verschiedenen Browsern ausgeführt um die Kompatibilität mit möglichst vielen Browsern zu bewahren.

## **3 Organisatorische Festlegungen**

Es werden wöchentliche Gruppentreffen stattfinden, in denen der Fortschritt sowie das weitere Vorgehen besprochen werden. Bei diesen Treffen werden neu anfallende Aufgaben identifiziert und auf die Gruppenmitglieder aufgeteilt. Auch werden dabei Termine für die Erledigung dieser Aufgaben festgelegt.

Die Dokumentation des Quelltextes ist die Aufgabe des jeweiligen Autors. Die Verantwortliche für Dokumentation überwacht dabei, dass der Code ausreichend Dokumentiert wird.