

## **Dokumentationskonzept**

### **Grundlagen:**

Als erste wesentliche Grundlagen sind die wöchentlichen Treffen und deren Problemlösungsbesprechungen anzusehen. In diesen werden Unklarheiten beziehungsweise Fragen zum Projekt geklärt. Die Treffen werden mittels Protokolle, die jeweils sonntags fertig ausgearbeitet dem Projektleiter übersandt werden, festgehalten. Diese Protokollierung ermöglicht eine Rückverfolgung des Planungsverlaufes und schlüsselt eventuelle Fehlplanungen in der Zeit auf. Des Weiteren werden wöchentlich Aufgaben an jedes Team-Mitglied verteilt, die er entweder dem Projektleiter zusendet oder, wenn es relevant für die Gruppe ist, im nächsten Gruppen-Meeting präsentiert, zum Beispiel bei Rechercheberichten oder Anleitungen. Um nachzuvollziehen, inwiefern er seine Aufgaben erledigt hat, welchen Schwierigkeitsgrad er darin bemessen sieht und welche Zeit er für diese aufgewendet hat, ist ein Aufwandsprotokoll anzufertigen, das sonntags abzugeben ist.

Sämtliche Schriftstücke unterliegen einem von uns entworfenem Layout, welches eine Vereinheitlichung und Strukturierung gewährleisten soll. Dies geschieht durch die Ausgabe von Schablonen im tex-, doc-, und sxw- Format. Jedem dieser einheitlichen Schriftstücke ist der Name, die Gruppe und das Erstellungsdatum hinzuzufügen.

### **Dokumentationen:**

Als Grundlage für die Entwicklerdokumentation und die Endbenutzerdokumentation dient das Lastenheft und das daraus später resultierende Pflichtenheft in Zusammenhang mit dem Glossar. Diese entstehen durch Besprechungen in Gruppentreffen und Kundengesprächen und weisen alle Funktionen und deren Zusammenhänge vor.

## **Entwicklerdokumentation:**

### **1) Design und Modellierung**

Zur Dokumentation in der Entwurfsphase gehören nicht nur UML-Diagramme, sondern auch Erklärungen dieser. Somit ist es für die anderen Mitglieder des Teams sowie den späteren Entwicklern leichter die Diagramme nachzuvollziehen.

### **2) Implementierung**

Ein sehr wichtiger Faktor zur Vermeidung von Unübersichtlichkeiten liegt beim regelkonformen Schreiben des JAVA-Quelltextes. Diese Normen sind dem Dokument für guten JAVA-Code (ausgearbeitet von Enrico Vierig) von der Webseite zu entnehmen. Hinzu kommen grundlegende Prinzipien der Implementierung, die ebenfalls dem gleichnamigen Dokument entnommen werden können.

Der Quelltext sollte wichtige und umfangreiche Funktionen beziehungsweise Teilabschnitte erläutern und ihre Funktionalität offen legen. Der Name des jeweiligen Programmierers hat immer über der entsprechenden Klasse zu stehen. Falls weitere Programmierer involviert sind oder später etwas an schon geschriebenem Quellcode ändern, dann sind ihre Namen ebenfalls an den entsprechenden Stellen zu vermerken.

Ein weiteres Prinzip der Quellcodedokumentation ist Javadoc, welches aus Java-Programmen HTML-Dokumentationen, die mit jedem Browser angeschaut werden können, generiert. Somit werden Klassen und deren Methoden beschrieben und deren Autoren sowie andere wichtige Details zu diesen vermerkt. Die genaue Anleitung ist dem Dokument Javadoc (von Xuan Zheng) zu entnehmen.

Ein letztes wichtiges Prinzip der Dokumentierung stellt die Arbeit mit CVS. Dieses protokolliert den Verlauf des geschriebenen Quelltextes und ermöglicht das Zusammenfügen unterschiedler Fragmente der einzelnen Entwickler. Dabei werden automatisch unterschiedliche Versionen des Programms erstellt, die dann rückwirkend nachvollzogen werden können. Des Weiteren ist es möglich Kommentare und beabsichtigte Ziele zur besseren Nachvollziehbarkeit hinzuzufügen. Diese Funktionen ermöglichen das passiv-kommunikative Zusammenspiel der Entwickler.

## Endbenutzerdokumentation:

Dem eigentlichen User des Programms soll ein Handbuch in zweifacher Ausführung dargeboten werden. Zum Einen soll das Handbuch als Online-Hilfe, in Protege, zur Verfügung stehen und zum Anderen als richtiges Benutzerhandbuch. Beide Varianten sind vom Inhalt gesehen identisch sollen aber praxisorientiert sein. Das heißt, dass die Erklärungen einfach gehalten und mit Screenshots unterlegt werden. Die Entstehung der ersten Version des Benutzerhandbuches erfolgt mit der Entstehung des Pflichtenheftes und wird im weiteren Verlauf der Entwicklung verfeinert und ausgearbeitet.

## Anhang:

```
/**
 * Ausgabe von "Hello World" auf dem Bildschirm <br>
 *
 * @author Enrico Vierig
 */
public class HelloWorld {

    /**
     * Gibt den String "Hello World" zurück <br>
     * @return String "Hello World"
     */
    public String hello (){
        return "Hello World";
    }

    /**
     * Standart main zum starten der Anwendung <br>
     * @param args
     */
    public static void main(String[] args) {
        HelloWorld ausgabe=new HelloWorld(); /* erstellen eines neuen Objekts der Klasse HelloWorld */
        System.out.println(ausgabe.hello()); /*ausgabe des Rückgabewertes der funktion hello */
    }
}
```

JAVAdoc

Quellcodedokumentation