

UNIVERSITÄT LEIPZIG

SOFTWARETECHNIK-PRAKTIKUM

# Qualitätssicherungsplan Gruppe cz17a - Gamification

*Lisa Vogelsberg, Felix Fink, Michael Fritz, Thomas Gerbert, Steven  
Lehmann, Fabian Ziegner, Willy Steinbach, Christian Schlecht*

supervised by

Dr. Christian ZINKE, Julia FRIEDRICH, Christian FROMMERT

5. Januar 2018

# Inhaltsverzeichnis

<b>1</b>	<b>Dokumentationskonzept</b>	<b>2</b>
1.1	Entwurfsbeschreibung . . . . .	2
1.2	strukturelle Dokumentation . . . . .	2
1.3	Code-Kommentierung . . . . .	3
1.4	Sprachgebrauch . . . . .	3
<b>2</b>	<b>Testkonzept</b>	<b>3</b>
2.1	Unit-Tests . . . . .	3
2.2	manuelle Tests . . . . .	3
<b>3</b>	<b>Organisatorische Festlegungen</b>	<b>3</b>
3.1	Fertigstellung vor Releasetermin . . . . .	4

## 1 Dokumentationskonzept

### 1.1 Entwurfsbeschreibung

In der Entwurfsbeschreibung, welche bei Release mit ausgeliefert wird, ist der aktuelle Entwurfsfortschritt dokumentiert und begründet. Dies bezieht sich vor allem auf Datenmodell sowie verwendete Softwarearchitekturen. Die Entwurfsbeschreibung bietet einen Gesamtüberblick über das ausgelieferte Release und dessen Umsetzung. Sie ist damit das zentrale Dokument, mit der sich Dritte als Erstes auseinandersetzen. Dementsprechend muss von Details abstrahiert werden, da sonst wichtigere („größere“) Grundkonzepte verloren gehen könnten.

### 1.2 strukturelle Dokumentation

Die strukturelle Dokumentation des Programms erfolgt mittels JavaDoc. Somit wird eine detaillierte Beschreibung automatisch als Web-Dienst zur Verfügung gestellt. Es wird vereinbart, dass jede existierende Funktion mittels JavaDoc kommentiert sein muss. Dabei muss so dokumentiert werden, dass auch Dritte verstehen, was die Funktion ausführt. Es darf nicht auf projektinterne Konventionen zurückgegriffen werden (falls doch nötig,

müssen diese mit ausgeführt werden). Es ist speziell auf Parameter und Return-Werte zu achten.

### **1.3 Code-Kommentierung**

Neben der Dokumentation mittels JavaDoc müssen zusätzliche Inline-Kommentare existieren, um spezielle Vorgänge/Funktionen zu erläutern. Solche Kommentare sollen ebenfalls möglichst nicht auf projektinterne Konventionen zurückgreifen und möglichst einfach sein bzw. möglichst wenig Vorkenntnisse voraussetzen. Verweise auf andere Kommentare sind zu vermeiden.

### **1.4 Sprachgebrauch**

TO DISCUSS: Deutsch oder Englisch oder Denglisch.

## **2 Testkonzept**

### **2.1 Unit-Tests**

Es werden für einzelne Funktionen Unit-Test mittels JUnit durchgeführt. Diese sind von jedem zu seinem individuellen Teil anzufertigen und ins Git zu pushen. Diese sollten möglichst eine große Breite der Funktionalitäten abdecken, um Fehleranfälligkeit zu vermeiden.

Im Rahmen eines Continuous Integration Prozesses werden die Unit-Test mittels Jenkins automatisch im Gitlab eingebunden.

### **2.2 manuelle Tests**

Der Projektleiter führt in regelmäßigen Abständen manuelle Test durch, um Funktionalität "im Standardgebrauch" zu garantieren. Besonders soll hierbei auf das User Interface geachtet werden.

## 3 Organisatorische Festlegungen

### 3.1 Fertigstellung vor Releasetermin

Die eigentliche Programmierung soll mindestens vier Tage vor Abgabezeitpunkt des Releasebündels abgeschlossen sein, um anschließend umfangreiche Test durchführen zu können, sowie die Dokumentation fertigzustellen.