Das ultimative Flaggenspiel

Risiken-Liste

Version 1.2

Überarbeitungshistorie

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Autor** |
| 21.08.2018 | 1.0 | Formatvorlage | Timo Krauskopf |
| 21.08.2018 | 1.1 | Inhalt | Patrick Stein,  Kenny O’Kagu Niklas Warmuth |
| 21.08.2018 | 1.2 | Ergänzung Risiken | Patrick Stein,  Kenny O’Kagu Niklas Warmuth |
|  |  |  |  |

Inhaltsverzeichnis

1. Vorwort 4

1.1 Ziel 4

1.2 Umfang 4

1.3 Definitionen, Akronyme und Abkürzungen 4

1.4 Referenzen 4

1.5 Übersicht 4

2. Risiken 4

2.1 Entwicklung eines verteilten Systems 4

2.1.1 Risikohöhe oder Einstufung 4

2.1.2 Beschreibung 4

2.1.3 Auswirkungen 4

2.1.4 Indikatoren 4

2.1.5 Schadensregulierung 4

2.1.6 Lösungsmöglichkeiten für Eventualfall 4

2.2 Datenbankzugriff aus Java 5

2.2.1 Risikohöhe oder Einstufung 5

2.2.2 Beschreibung 5

2.2.3 Auswirkungen 5

2.2.4 Indikatoren 5

2.2.5 Schadensregulierung 5

2.2.6 Lösungsmöglichkeiten für Eventualfall 5

2.3 Implementierung eines Timers 5

2.3.1 Risikohöhe oder Einstufung 5

2.3.2 Beschreibung 5

2.3.3 Auswirkungen 5

2.3.4 Indikatoren 5

2.3.5 Schadensregulierung 5

2.3.6 Lösungsmöglichkeiten für Eventualfall 5

Risiken-Liste

# Vorwort

## Ziel

Das Ziel dieses Dokumentes ist die Erstellung einer Risiken-Liste, damit alle möglichen Gefahren und Gefährdungen des Projektes weitestgehend im Voraus bekannt sind und in dem weiteren Projektablauf berücksichtigt werden können.

## Umfang

Dieses Dokument enthält alle Risiken, die für die Entwicklung der Quiz-Software als problematisch identifiziert wurden.

## Definitionen, Akronyme und Abkürzungen

Siehe Glossar.

## Referenzen

* Glossar

## Übersicht

Dieses Dokument enthält eine Übersicht über alle Risiken, die zur Zeit der Publikation dieses Dokumentes bekannt waren. Zusätzlich werden mögliche Lösungsvorschläge und Alternativen präsentiert.

# Risiken

## Implementierung eines Timers

### Risikohöhe oder Einstufung

Die Risikohöhe ist niedrig.

### Beschreibung

Die Software benötigt eine Information über die verstrichene Zeit und muss die verbleibende Zeit für den Spieler darstellen, die für eine Frage übrig ist.

### Auswirkungen

Es kann eines der Kernelemente des Spiels nicht korrekt umgesetzt werden - das Spiel verliert an Umfang.

### Indikatoren

Keine Erfolgreiche Implementierung des Timers.

### Schadensregulierung

In der Java-API nach einem Timer schauen.

### Lösungsmöglichkeiten für Eventualfall

Eigenen Timer programmtechnisch realisieren, der die Anforderungen für unser Spiel erfüllt.

## Flüssige Wechsel der Bilder

### Risikohöhe oder Einstufung

Die Risikohöhe ist hoch.

### Beschreibung

Das Spiel basiert darauf, dass der Benutzer Eingaben schnell macht und schnell eine Eingabe auf eine Ausgabe folgt. Ein nicht flüssiger Wechsel oder Ladezeiten beim Wechsel der Fragen könnte den Spielspaß beeinflussen.

### Auswirkungen

Spielspaß des Benutzers stark beeinträchtigt.

### Indikatoren

Darstellung nach Fertigstellung auf Ruckeln überprüfen.

### Schadensregulierung

Nach Strategien zur Ruckelfreien Darstellung suchen.

Die Java-API nach Techniken zum flüssigen Wechsel von Bildern durchsuchen.

Prototyp erstellen.

### Lösungsmöglichkeiten für Eventualfall

Mehr Schadensregulierung betreiben.

Mit dem Timer das Ruckeln ausgleichen.

## Datenbankzugriff aus Java

### Risikohöhe oder Einstufung

Die Risikohöhe ist mittel.

### Beschreibung

Wie kann der Zugriff auf eine Datenbank aus Java heraus erfolgen?

### Auswirkungen

Der Zugriff auf eine Datenbank wird nicht umgesetzt.

### Indikatoren

Keine Erzeugung einer korrekten Datenbankanbindung.

### Schadensregulierung

Prototyp entwickeln und testen der JDBC-API.

### Lösungsmöglichkeiten für Eventualfall

Nutzen von Serialisierungsmechanismen für Persistenz.

## 

## 

### 