

## **Software Quality Concept:**

### **Entwicklungs- und Teamkultur:**

Grundsätzlich halten sich alle Teammitglieder an folgende Fragen:

- Verstehen meine Teammitglieder meinen Code mithilfe von JavaDoc?
- Entspricht mein Code der vereinbarten Programmiersprache?
- Gibt es bereits Code mit ähnlicher Funktionalität an anderer Stelle?

Wir arbeiten bevorzugt im Pair Programming mit zwei definierten Rollen: Eine Person schreibt den Code, die andere observiert und kommentiert ihn. Die Rollen werden regelmäßig getauscht, um Know-how gleichmäßig zu verteilen und die Codequalität konstant hoch zu halten.

Wir wollen zusätzlich mit Checklisten arbeiten und uns so oft wie möglich persönlich treffen. Zwischen den physischen Sitzungen wird über WhatsApp, Discord und Zoom kommuniziert.

### **Dokumentation mit JavaDoc:**

Wir setzen auf JavaDoc-basierte Dokumentation. Jede `public`-Klasse und -Methode muss mit JavaDoc beschrieben sein (Zweck, Parameter, Rückgabewert, Ausnahmen). Nur dokumentierter Code sollte gepusht werden – dies versuchen wir teamweit als Regel durchzusetzen.

Für die Dokumentation der Methoden und Klassen sind die Personen verantwortlich, die sie geschrieben haben. Zur Vereinfachung wird das JavaDoc-Plugin in IntelliJ verwendet.

### **Logging:**

Um die Fehlersuche zu vereinfachen, werden Logger eingesetzt. Die Idee dahinter ist, dass möglichst vor und nach jeder Aktion eine Lognachricht erzeugt wird. So können Fehler leichter erkannt und eingegrenzt werden.

### **Code Coverage:**

Unser späteres Ziel ist es, JUnit-Tests durchzuführen und mehr als die Hälfte des Codes durch Tests abzudecken. Im Fokus stehen dabei die Spiel-Logik und die Client-Server-Interaktion, was auch das Encoding/Decoding des Protokolls sowie Lobby- und Spieler-Management einschließt. Als Tool verwenden wir das Jacoco-Plugin.

### **Bug-Report:**

Wenn ein Bug auftritt, erstellen wir einen spezifischen Bug-Report, sodass alle Teammitglieder informiert sind und nachvollziehen können, was passiert ist. Bug-Reports können direkt in GitLab angelegt werden.

### **Metrik-Messung:**

Da wir uns noch in der Anfangsphase des Projekts befinden und dies unser erstes QA-Konzept ist, haben wir noch keine endgültige Entscheidung zu den Metriken getroffen.

Eine Regel haben wir jedoch bereits festgelegt: Funktionen sollen nicht länger als 150 Zeilen Code umfassen. Code Coverage durch Tests soll regelmäßig überprüft werden.