

Softwarequalitäts-Konzept

Bei der Implementierung unseres Spiels möchten wir unsere Softwarequalität sicherstellen, indem wir uns am konstruktiven Qualitäts-Konzept „Logging“ orientieren. Sowohl für den Server als auch für den Client haben wir auf das in der Vorlesung vorgeschlagene Tool „self4j“ zurückgegriffen. Nachfolgend ein paar Beispiele aus den Client Klassen:

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

Abb.1: Einbindung der beiden benötigten Klassen, um das Logger-Objekt zu erzeugen (GameClient.java)

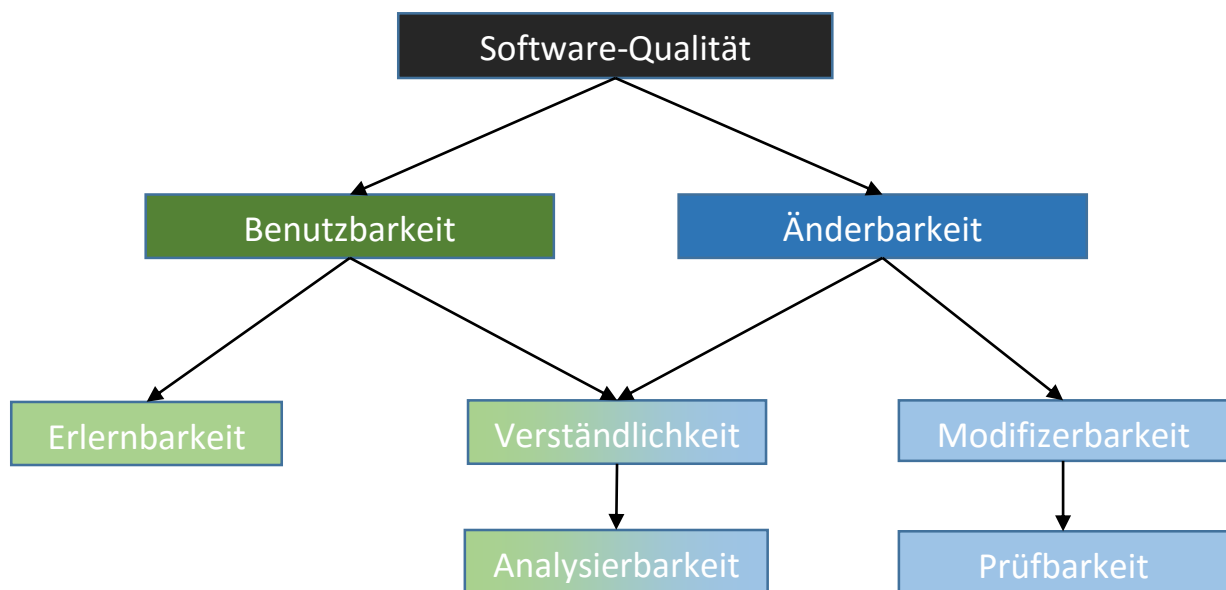
Abb.2: Für die Klasse GameClient wird eine Instanz des Logger-Objekts erzeugt

```
log = LoggerFactory.getLogger(GameClient.class);
```

```
} catch (IOException e) {
    log.error(e.getMessage());
}
```

Abb.3: Das Logger-Objekt erzeugt eine Error-Message, mit der Beschreibung der Exception (Natürlich nur wenn der catch-Block durchlaufen und eine Exception geworfen wird)

FCM-Modell (Qualitäts-Merkmale/-Teilmerkmale)



Zu den Qualitätsmerkmalen (Indikatoren)

Benutzbarkeit
Wir möchten unser Spiel so Benutzerfreundlich wie nur möglich gestalten. Der Spielspass sollte nicht durch Fehlersuche im Code beeinträchtigt werden
Unser Qualitätsmanagement ist der perfekte Grundstein, um dieses Ziel auch erreichen zu können

Änderbarkeit
Hexagon (insbesondere das GUI) ist nach dem „Lego“-Prinzip aufgebaut. Diese Vorgehensweise macht es nicht nur einfacher, Veränderungen an unserem Spiel vorzunehmen, sondern macht den Code auch Verständlicher
Jede geschriebene Klasse wird ausführlich beschrieben und allfällige Errors werden (durch den Logger) dem Benutzer angezeigt

Messung der Qualitätsindikatoren

Es ist sehr schwierig, für die ausgewählten zwei Qualitätsmerkmale geeignete Skalen (Qualitätsmasse) zu erstellen, um diese zu Messen. Jedoch können wir Standards angeben (d.h. die Anforderungen, die wir an unser Spiel haben), welche durch einen konkreten Wert beschrieben werden können:

- Wir wollen, dass zum Schluss unseres Projektes mindestens 95% unserer geschriebenen Methoden in Javadoc erfasst sind!
- Wir wollen, dass unsere geschriebenen Methoden im Durchschnitt maximal dreissig Befehle (Kommentare und Leerzeilen sind damit nicht gemeint) umfassen. Dies steigert die Lesbarkeit unseres Codes!
- Wir wollen, dass am Ende unseres Projekts mindestens 95% der (vorher) definierten Testfällen(test cases) erfolgreich verlaufen!

Benutzbarkeit:

- Unser Spiel sollte ohne Anleitung spielbar sein
- Die Lobby sollte eher einfach gehalten werden, damit sich jeder Mitspieler darin zurechtfindet
- Die Steuerung im Spiel soll mit nur vier Richtungstasten erfolgen (Wie im MS1 vorgestellt)

Änderbarkeit:

- Von jeder Klasse sollen alle Methode so dokumentiert sein, dass jedermann Sinn und Zweck der Methode erkennen kann
- Soll z.B. ein Eckpunkt des Spielfeldes verschoben werden, soll dies mittels Änderung zweier Koordinaten fehlerfrei bewerkstelligt werden können