

Qualitätssicherungskonzept – Gruppe 3

Die Qualität einer Software ist ausschlaggebend für ihren Wert und die Benutzerzufriedenheit. Um die gewünschte Qualität zu erreichen und aufrechtzuerhalten sind gewisse Massnahmen nötig. Für unser Projekt *Vector Racer* führen wir die folgenden Massnahmen durch zur Sicherung der Softwarequalität.

Javadoc

Javadoc ist ein Dokumentationswerkzeug, dass aus kommentiertem Code eine Dokumentation des Codes anfertigt.

Den Code im Javadoc-Stil zu kommentieren bringt zwei Vorteile: Einerseits ist der Code verständlicher, da die Erläuterung dessen unmittelbar darübersteht und andererseits wird die Dokumentation des Codes parallel zur Entwicklung geschrieben, da Javadoc diese aus den Kommentarzeilen im Code selbstständig generiert.

Um die Einhaltung dieser Massnahme zu sichern, wird regelmässig überprüft, dass für alle Klassen stets vollständige Einträge in der Dokumentation vorhanden sind. Sollte das nicht der Fall, wird schnellstmöglich der Code mit den nötigen Kommentaren ergänzt.

Unit Tests

Mit Unit Tests lassen sich einzelne Funktionen und Klassen auf ihre Funktionalität prüfen, indem Tests geschrieben werden, die auf die jeweilige Klasse oder Funktion angewendet wird.

Diese Massnahme gewährleistet, dass der geschriebene Code so funktioniert wie er soll. Dabei werden diese mit unterschiedlichen Argumenten und Aufrufparameter getestet. Bei Fehlschlägen wird der Quellcode überarbeitet.

Nach der Implementation einer grösseren Funktion oder Klasse wird jeweils ein Test geschrieben, der diese testet. Ist noch keiner vorhanden, so wird dieser geschrieben.

Profiler

Ein Profiler gibt Information über die Zeit der Methodenausführung sowie Speicherverteilung und Auslastung aus.

Falls das Projekt schleppend ausgeführt wird oder gar regelmässig abstürzt, kann ein Profiler Aufschluss darüber geben, was das Problem ist. Damit lassen sich Engpässe des Programms feststellen.

Diese Massnahme wird ohnehin eingehalten, da wir in der Gruppe ohnehin das Starten des Profilers in der run-configuration innerhalb von IntelliJ haben.

Pair Programming

Das Pair Programming beschreibt das Programmieren in einer Zweiergruppe, wobei jemand den Code schreibt und ein anderer diesen sofort evaluiert und Feedback gibt.

Pair Programming hat zur Folge, dass der lesbarer und qualitativ besser wird, da der Code vom Informationsaustausch der Gruppe profitiert.

Um die Massnahme einzuhalten, wird auf der Wiki-Seite des Repository ein Protokoll über die Aktivität im Pair Programming geführt und regelmässig überprüft.

Code Review

Ähnlich wie im Pair Programming wird im Code Review ein geschriebener Code von einem anderem Teammitglied evaluiert. Das resultierende Feedback wird besprochen und ggf. auf den Code angewendet.

Auch hier ist die Konsequenz ein lesbarer und besserer Code, der einfach nachzuvollziehen ist.

Der gesamte Code ist jederzeit für alle Teammitglieder zugänglich auf Gitlab. Dies ermöglicht, dass der Quellcode regelmässig von Teammitgliedern durchgelesen und gewertet wird. Es liegt in der Verantwortung jedes Teammitglieds, zumindest Teile des Codes im Repository regelmässig zu überprüfen.

Innerhalb des Teams wurde folgende Abmachung getroffen: Falls ein git pull Befehl eine grössere Änderung im lokalen Repository nach sich zieht, müssen diese Änderungen - sofern man sie nicht selbst geschrieben hat - vom «Puller» inspiziert werden.