



Projekt: JBombberman
Qualitätsmassnahmen

Pascal Kistler
Silvan Adrian
Fabian Binna

1 Änderungshistorie

Datum	Version	Änderung	Autor
13.04.15	1.00	Erstellung des Dokuments	Gruppe
13.04.15	1.01	Qualitätsmassnahmen	Pascal Kistler
28.04.15	1.02	Massnahmen angepasst	Silvan Adrian
18.05.15	1.03	GitLab hinzugefügt + Codreview angepasst	Silvan Adrian
27.05.15	1.04	Vorbereitung Abgabe	Silvan Adrian

Inhaltsverzeichnis

1	Änderungshistorie	2
2	Massnahmen zur Qualitätssicherung	4
2.1	Git Repository	4
2.2	JUnit Tests	4
2.3	Findbugs	4
2.4	Jacoco	4
2.5	Jenkins Buildserver	4
2.6	GitLab	4
2.7	CodeReview	4

2 Massnahmen zur Qualitätssicherung

2.1 Git Repository

Über das Git Repository haben alle Zugriff auf den aktuellen Code. Jedes Teammitglied erstellt eigene branches für die Features. Sobald ein Feature fertig ist kann dies in den Masterbranch gemerged werden, sodass im Master immer eine voll funktionsfähige Version ist.

Die Dokumente der Dokumentation werden in einem separaten Repository versioniert. In diesem Repository werden keine Branches erstellt, sondern alles in den Master gepusht.

2.2 JUnit Tests

JUnit-Tests werden verwendet, um kleinere Einheiten der Software automatisiert auf Fehler zu überprüfen. Diese werden vor jedem Commit ausgeführt, so dass kein Code mit fehlgeschlagenen Tests eingecheckt wird.

2.3 Findbugs

Findbugs wird eingesetzt, um durch statische Codeanalyse Bugs oder mögliche Probleme im Code zu finden. FindBugs liefert dem Entwickler Hinweise auf mögliche oder tatsächliche Probleme im Code.

2.4 Jacoco

Jacoco wird eingesetzt um die Test Code Coverage zu sehen und zu verbessern.

2.5 Jenkins Buildserver

Nach jedem Commit startet der Jenkins Server einen Build vom Masterbranch. Dabei führt Jenkins alle JUnit-Tests aus und sucht mittels FindBugs nach Fehlern im Code. Sollten nicht alle Tests erfolgreich sein muss der Fehler sofort behoben werden. Ein Merge mit dem Master wird nur durchgeführt, wenn alle Tests grün sind. Zusätzlich zum Findbugs wird mit Jacoco eine Test Code Coverage Analyse durchgeführt.

2.6 GitLab

Bei jedem Push oder merge in das Code repository kriegt jedes Teammitglied eine Notification über Slack, welche es jedem Teammitglied erlaubt sofort die neusten Änderungen anzusehen und womöglich einen Kommentar dazu abzugeben.

2.7 CodeReview

Zum Schluss des Projekts hin wird ein Codereview im Team durchgeführt, um nochmals den ganzen Code durchzugehen.