

Automatische Kwaliteits Controle

Guus Hamm en Rick Rongen

6 april 2017

Inhoudsopgave

1	Code Review	3
2	SonarQube	3
3	Eisen	3
4	Kotlin	4

1 Code Review

Doordat al onze code door GerritHub gaat, wordt er al een code kwaliteits controle gedaan voordat deze op GitHub komt te staan. Iedere deelnemer van het project kan de code controleren en afkeuren als deze niet goed genoeg is.

2 SonarQube

SonarQube voert een statische code kwaliteits controle uit op de code die op GerritHub wordt gezet. Er wordt aan de hand van regels vastgesteld of de code structureel goed in elkaar zit en geen grote fouten bevat als oneindige loops etc.

Uit deze scan wordt er een rapport gegereerd dat in dit geval door iedereen ingezien kan worden. In dit rapport staat of en welke problemen de code bevat. Dit kan gaan om oneindige loops, vage code of onbeschreven methoden (geen comentaar).

3 Eisen

Vanuit de resultaten van SonarQube hebben wij enkele eisen opgesteld waar onze code aan moet voldoen. Als dit niet zo is zullen wij de code zelf afkeuren op GerritHub.

Het is belangrijk dat we minimaal op een B uitkomen in code kwaliteit. Dit is wel degelijk haalbaar omdat de meeste problemen die SonarQube zal vinden op stijl en vergeten commentaar neer zal komen.

4 Kotlin

Een probleem met SonarQube is dat het (nog) geen Kotlin ondersteund. Ons project bestaat voor een groot deel uit Kotlin omdat dit gemakkelijker te schrijven is. Echter zien wij dit niet echt als een groot probleem, omdat de belangrijkste onderdelen die vatbaar zijn voor problemen wel nog in Java zijn geschreven.



Figuur 4.1: SonarQube