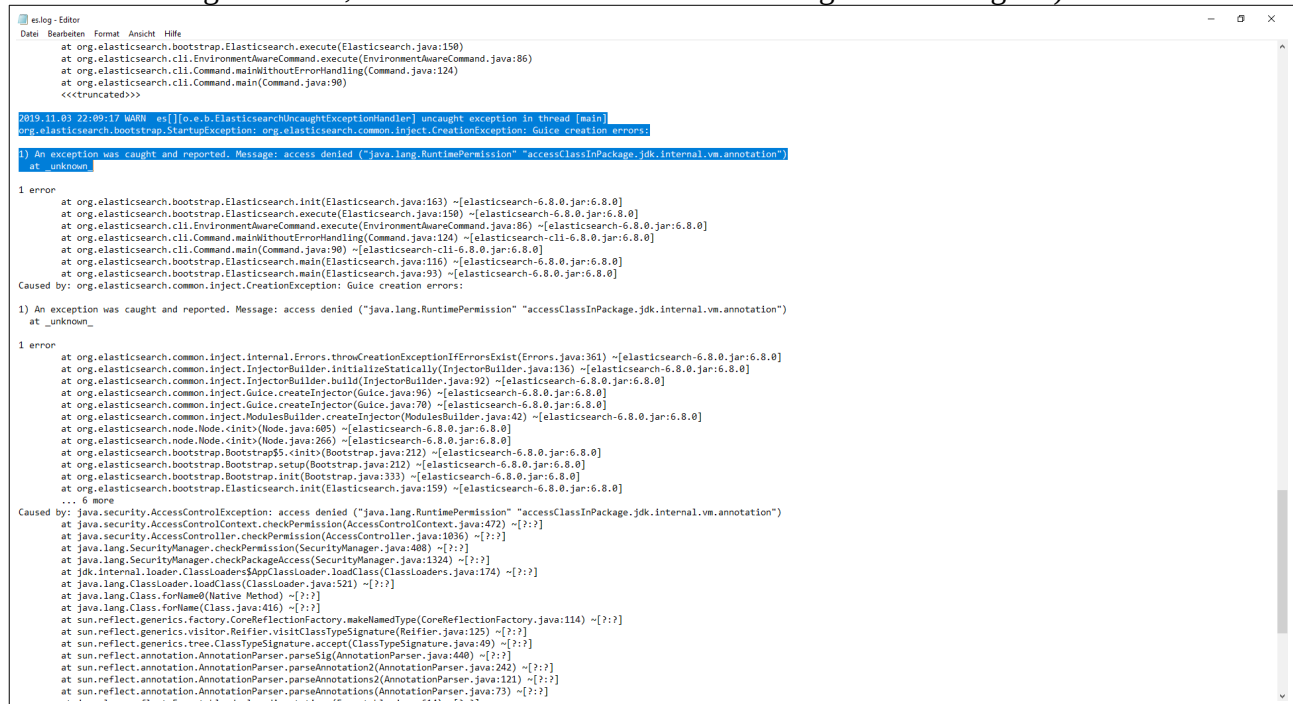


Ich hab mich entschieden, mit Sonarqube zu experimentieren.

Bevor ich loslegen konnte, musste ich mich Java Fehlermeldungen rumschlagen :)



```
2019.11.03 22:09:17 WARN es[[]o.e.b.ElasticsearchUncaughtExceptionHandler] uncaught exception in thread [main]
org.elasticsearch.bootstrap.StartupException: org.elasticsearch.common.inject.CreationException: Guice creation errors:

1) An exception was caught and reported. Message: access denied (\"java.lang.RuntimePermission\" \"accessClassInPackage.jdk.internal.vm.annotation\")
at _unknown_

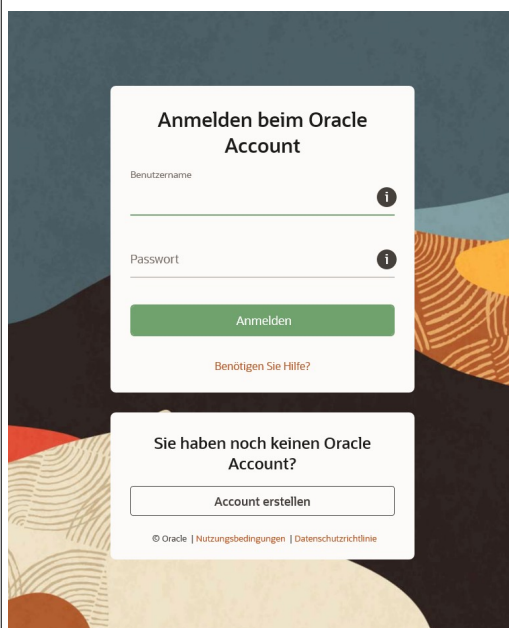
1 error
at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:163) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.bootstrap.Elasticsearch.execute(Elasticsearch.java:150) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.cli.EnvironmentAwareCommand.execute(EnvironmentAwareCommand.java:86) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.cli.Command.mainWithoutErrorHandling(Command.java:124) ~[elasticsearch-cli-6.8.0.jar:6.8.0]
at org.elasticsearch.cli.Command.main(Command.java:90) ~[elasticsearch-cli-6.8.0.jar:6.8.0]
at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:116) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.bootstrap.Elasticsearch.main(Elasticsearch.java:93) ~[elasticsearch-6.8.0.jar:6.8.0]
Caused by: org.elasticsearch.common.inject.CreationException: Guice creation errors:

1) An exception was caught and reported. Message: access denied (\"java.lang.RuntimePermission\" \"accessClassInPackage.jdk.internal.vm.annotation\")
at _unknown_

1 error
at org.elasticsearch.common.inject.internal.Errors.throwCreationExceptionIfErrorsExist(Errors.java:361) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.common.inject.InjectorBuilder.initializeInitially(InjectorBuilder.java:136) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.common.inject.InjectorBuilder.build(InjectorBuilder.java:92) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.common.inject.Guice.createInjector(Guice.java:96) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.common.inject.Guice.createInjector(Guice.java:70) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.common.inject.ModuleBuilder.createInjector(ModuleBuilder.java:42) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.node.Node.<init>(Node.java:605) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.node.Node.<init>(Node.java:266) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.bootstrap.Bootstrap5.<init>(Bootstrap.java:212) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.bootstrap.Bootstrap.setup(Bootstrap.java:212) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.bootstrap.Bootstrap.init(Bootstrap.java:333) ~[elasticsearch-6.8.0.jar:6.8.0]
at org.elasticsearch.bootstrap.Elasticsearch.init(Elasticsearch.java:159) ~[elasticsearch-6.8.0.jar:6.8.0]
... 6 more
Caused by: java.security.AccessControlException: access denied (\"java.lang.RuntimePermission\" \"accessClassInPackage.jdk.internal.vm.annotation\")
at java.security.AccessControlContext.checkPermission(AccessControlContext.java:472) ~[?:?]
at java.security.AccessController.checkPermission(AccessController.java:1036) ~[?:?]
at java.lang.SecurityManager.checkPermission(SecurityManager.java:408) ~[?:?]
at java.lang.SecurityManager.checkPackageAccess(SecurityManager.java:1324) ~[?:?]
at jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:174) ~[?:?]
at java.lang.ClassLoader.loadClass(ClassLoader.java:521) ~[?:?]
at java.lang.Class.forName0(Native Method) ~[?:?]
at java.lang.Class.forName(Class.java:416) ~[?:?]
at sun.reflect.generics.factory.CoreReflectionFactory.makeNamedType(CoreReflectionFactory.java:114) ~[?:?]
at sun.reflect.generics.visitor.Reifier.visitClassTypeSignature(Reifier.java:125) ~[?:?]
at sun.reflect.generics.tree.ClassTypeSignature.accept(ClassTypeSignature.java:49) ~[?:?]
at sun.reflect.annotation.AnnotationParser.parseSig(AnnotationParser.java:440) ~[?:?]
at sun.reflect.annotation.AnnotationParser.parseAnnotation2(AnnotationParser.java:242) ~[?:?]
at sun.reflect.annotation.AnnotationParser.parseAnnotations2(AnnotationParser.java:121) ~[?:?]
at sun.reflect.annotation.AnnotationParser.parseAnnotations(AnnotationParser.java:73) ~[?:?]
```

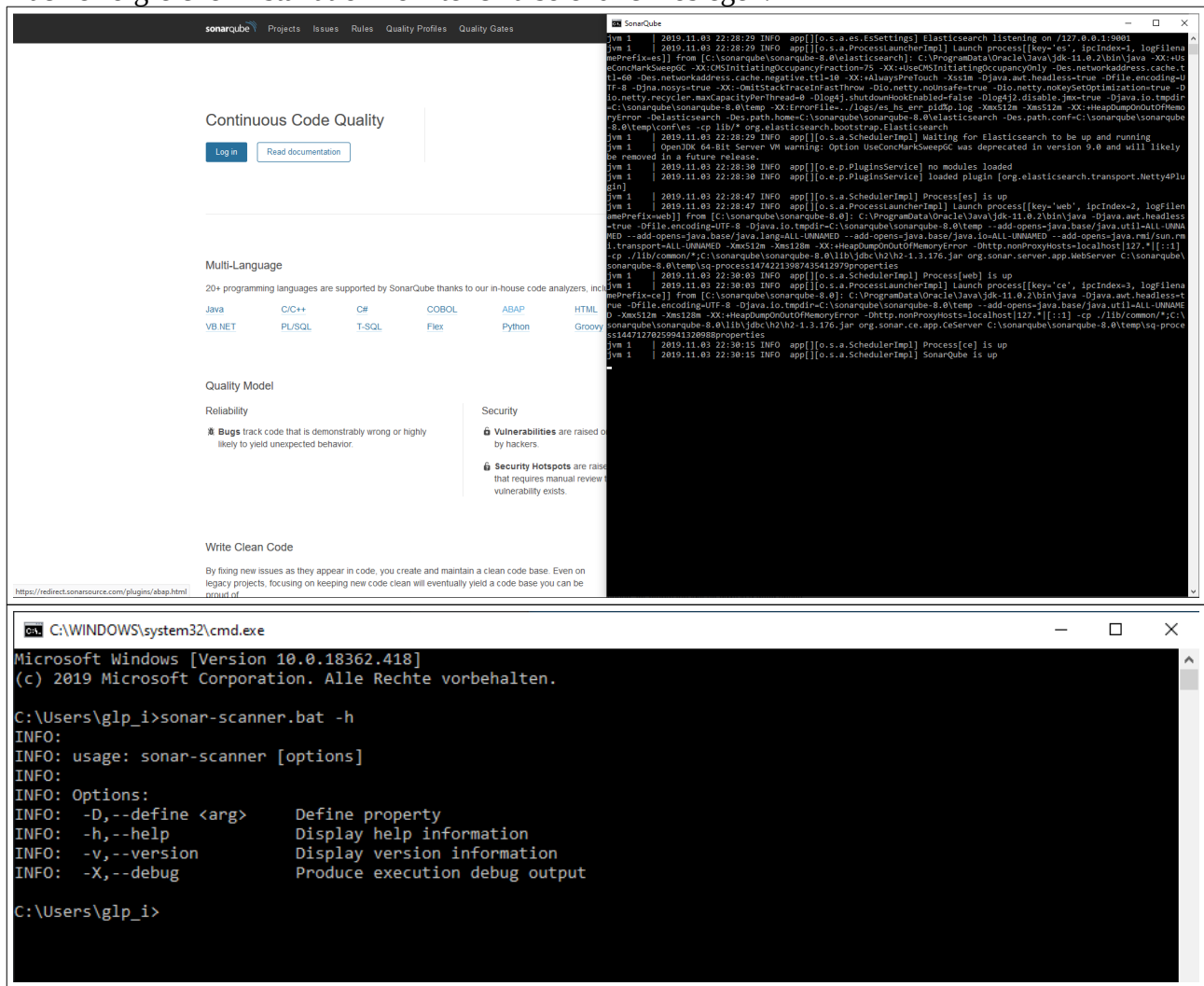
Ich dachte, das liege an meiner veralteten Java-Version (laut Sonarqube benötigt man Version 11+) und wollte die aktuellste Version herunterladen – wurde aber vom Login abgeschreckt.

Glücklicherweise stolperte ich irgendwo über die Info, dass man nicht „11+“ benötigt, sondern „genau 11“.

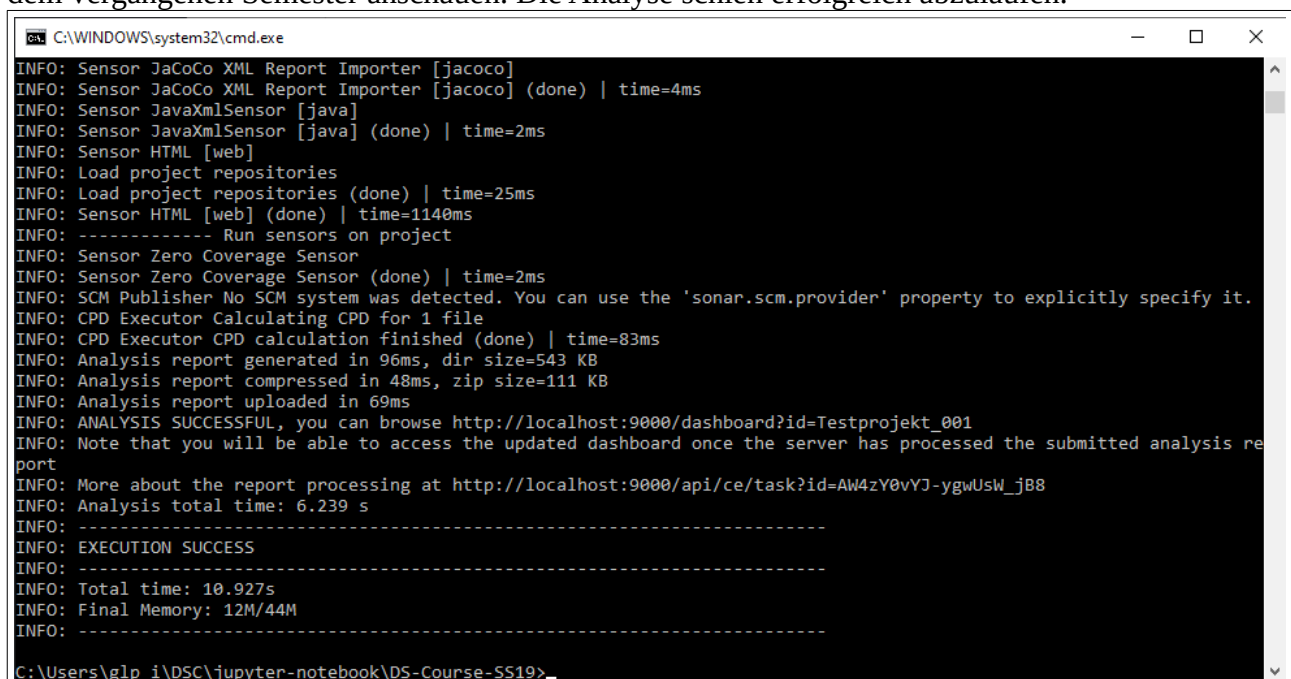


		Source	Tag is jdk-12.0.2+ga
12.0.1 (build 12.0.1+12)			
Windows	64-bit	zip (sha256) 188M	
Mac	64-bit	tar.gz (sha256) 181M	
Linux	64-bit	tar.gz (sha256) 189M	
	Source	Tag is jdk-12.0.1+ga	
12 GA (build 12+33)			
Windows	64-bit	zip (sha256) 188M	
Mac	64-bit	tar.gz (sha256) 182M	
Linux	64-bit	tar.gz (sha256) 189M	
	Source	Tags are jdk-12+33, jdk-12-ga	
11.0.2 (build 11.0.2+9)			
Windows	64-bit	zip (sha256) 179 MB	
Mac	64-bit	tar.gz (sha256) 174 MB	
Linux	64-bit	tar.gz (sha256) 179 MB	
	Source	Tags are jdk-11.0.2+9, jdk-11.0.2-ga	
11.0.1 (build 11.0.1+13)			
Windows	64-bit	zip (sha256) 179 MB	
Mac	64-bit	tar.gz (sha256) 174 MB	
Linux	64-bit	tar.gz (sha256) 179 MB	
	Source	Tag is jdk-11.0.1+13	
11 GA (build 11+28)			
Windows	64-bit	zip (sha256) 179 MB	
Mac	64-bit	tar.gz (sha256) 175 MB	
Linux	64-bit	tar.gz (sha256) 179 MB	
	Source	Tag is jdk-11+28	
10.0.2 (build 10.0.2+13)			
Windows	64-bit	tar.gz (sha256) 190 MB	
Mac	64-bit	tar.gz (sha256) 192 MB	
Linux	64-bit	tar.gz (sha256) 196 MB	
	Source	Tag is jdk-10.0.2+13	
10.0.1 (build 10.0.1+10)			
Windows	64-bit	tar.gz (sha256) 199 MB	
Mac	64-bit	tar.gz (sha256) 201 MB	
Linux	64-bit	tar.gz (sha256) 205 MB	
	Source	Tag is jdk-10.0.1+10	
10 GA (build 10+46)			
Windows	64-bit	tar.gz (sha256) 190 MB	
Mac	64-bit	tar.gz (sha256) 192 MB	
Linux	64-bit	tar.gz (sha256) 195 MB	
	Source	Tag is jdk-10+46	

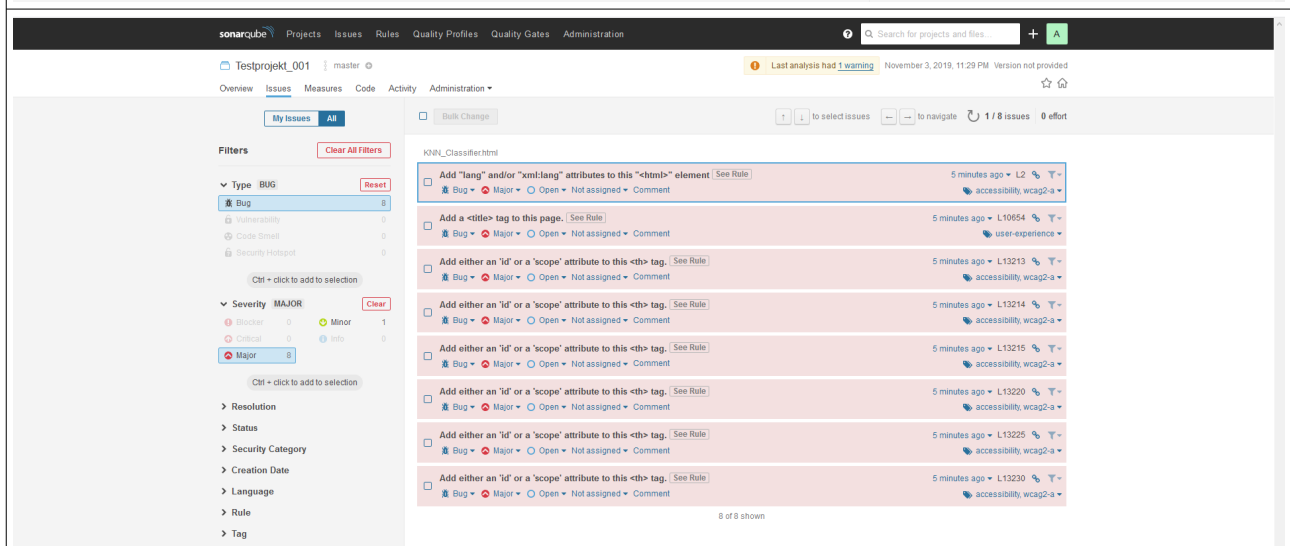
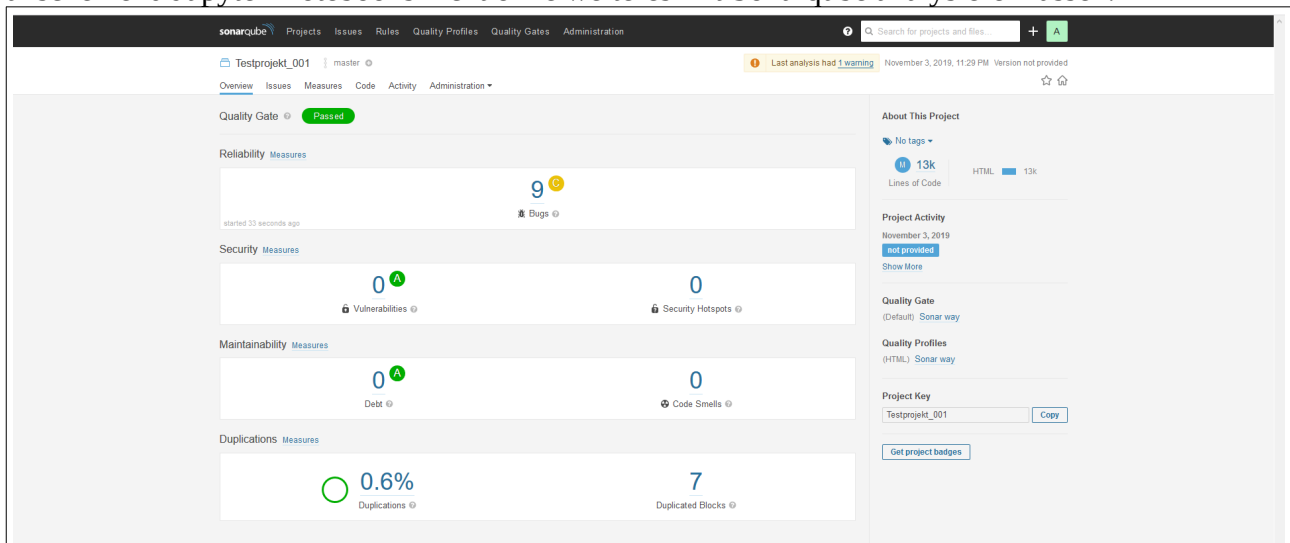
Nach erfolgreicher Installation konnte ich also endlich loslegen.



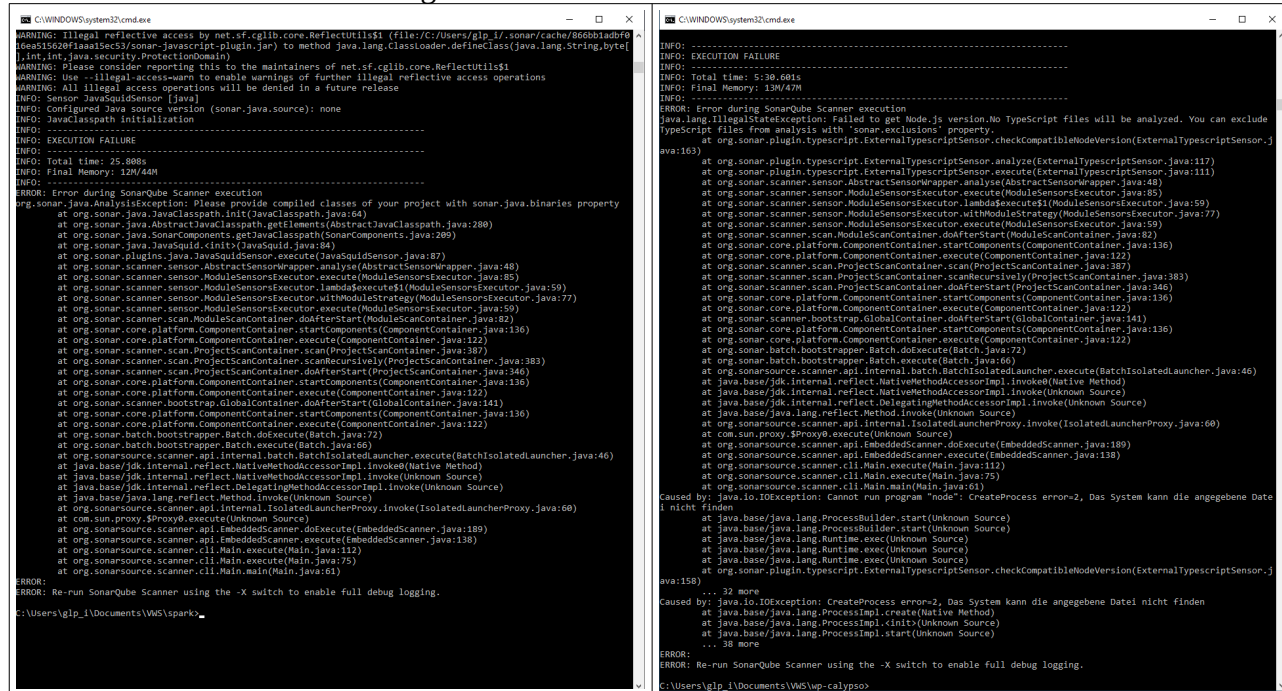
Als erstes Testprojekt wollte ich mir einige als Jupyter-Notebooks vorliegende Python-Übungen aus dem vergangenen Semester anschauen. Die Analyse schien erfolgreich abzulaufen.



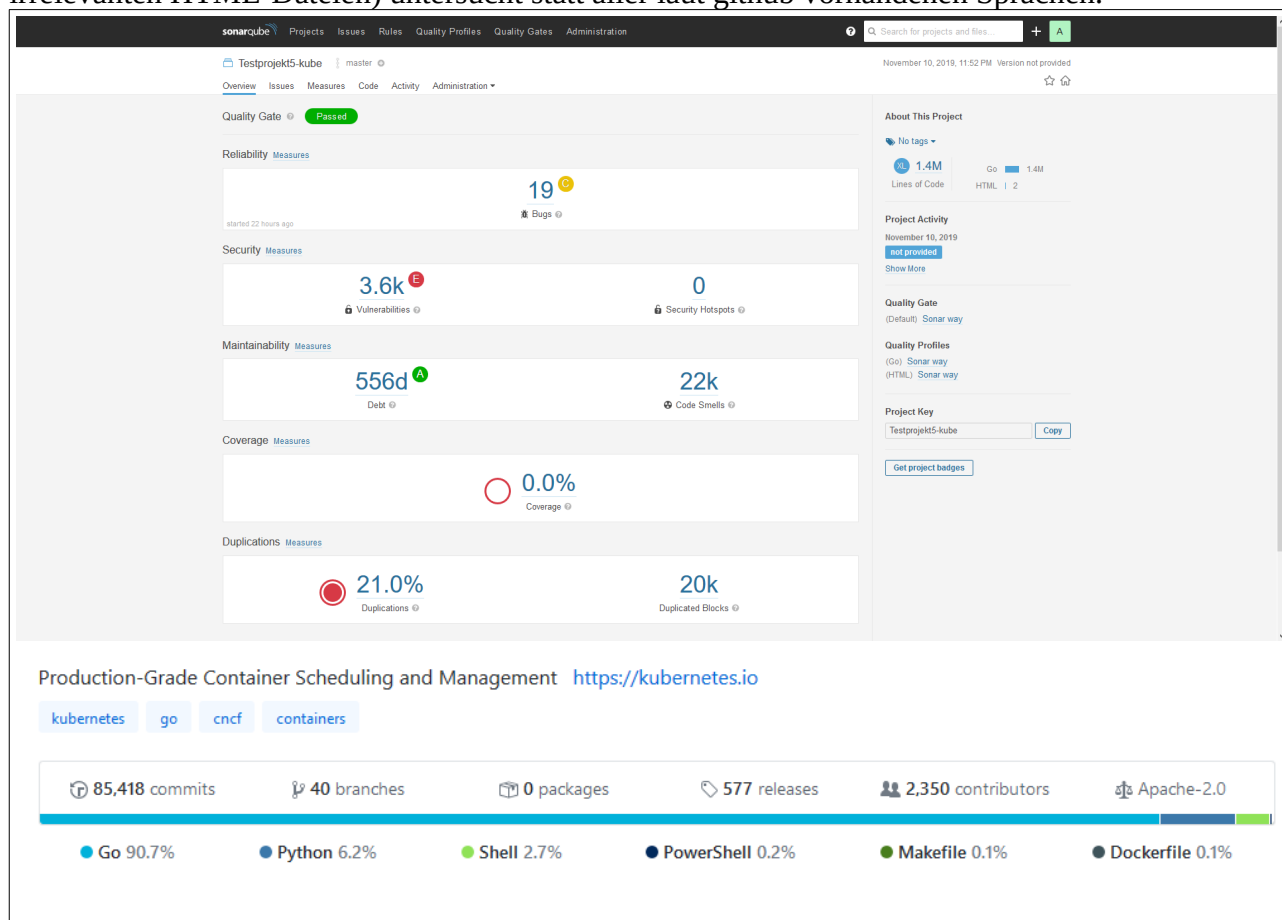
Die Analyse-Ergebnisse erwiesen sich allerdings als recht dünn. Die wenigen auf den ersten Blick sichtbaren Bugs etc bezogen sich sämtlich auf den HTML-Code. Später fand ich heraus, dass sich anscheinend Jupyter-Notebooks nicht ohne weiteres mit Sonarcube analysieren lassen.



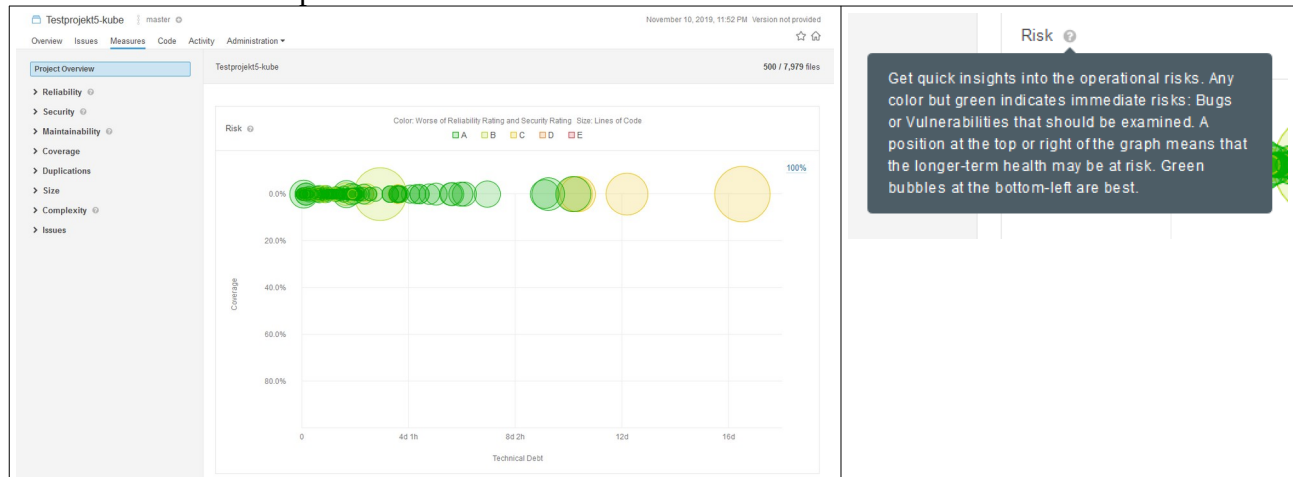
Da ich aktuell keine weiteren eigenen Projekte habe, die ich mir anschauen konnte, habe ich mir von github die populärsten Projekte gecloned, um diese unter die Lupe zu nehmen. Spark und WP-Calypto konnte ich allerdings nicht untersuchen – bei beiden endete der SonarQube-Testlauf mit einer Fehlermeldung.



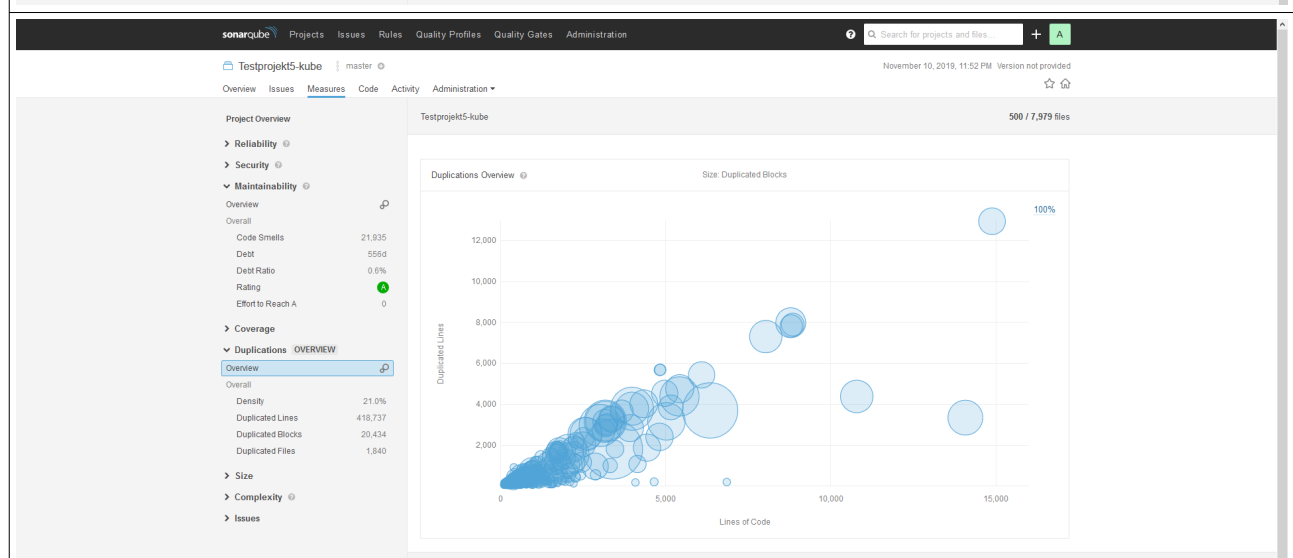
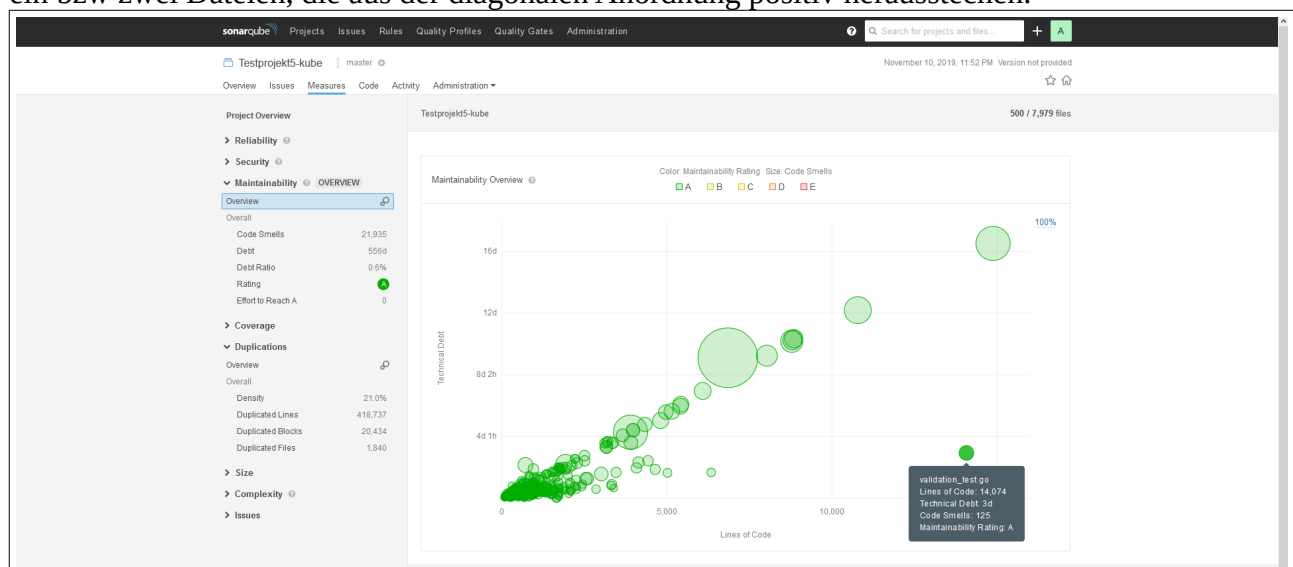
Kubernetes brachte schließlich den ersten Erfolg. Allerdings wurden hier nur die Go-Files (und die irrelevanten HTML-Dateien) untersucht statt aller laut github vorhandenen Sprachen.



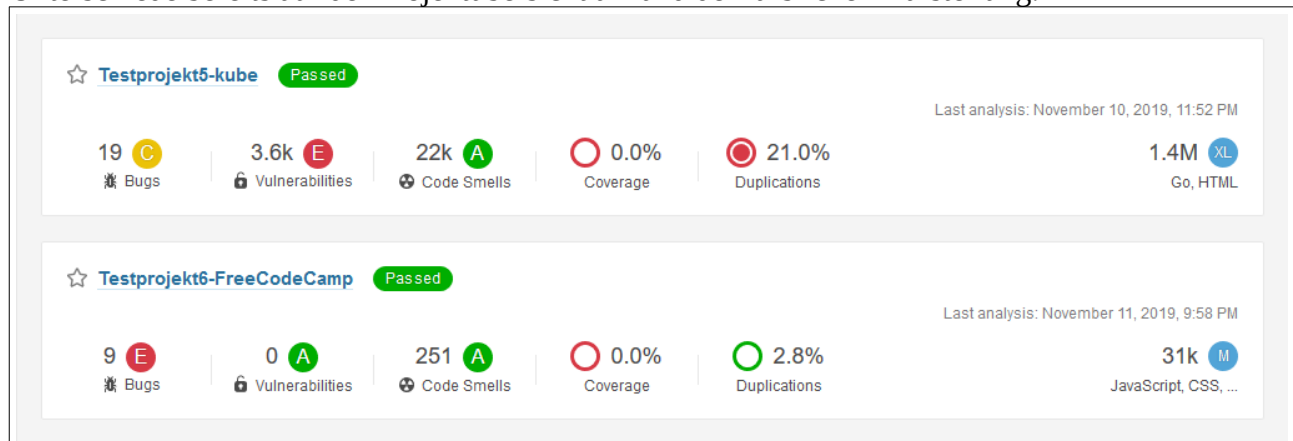
Einen guten ersten Überblick bietet die visuelle Darstellung. Per Rollover erhält man grobe Infos, wie die Kreise zu interpretieren sind.



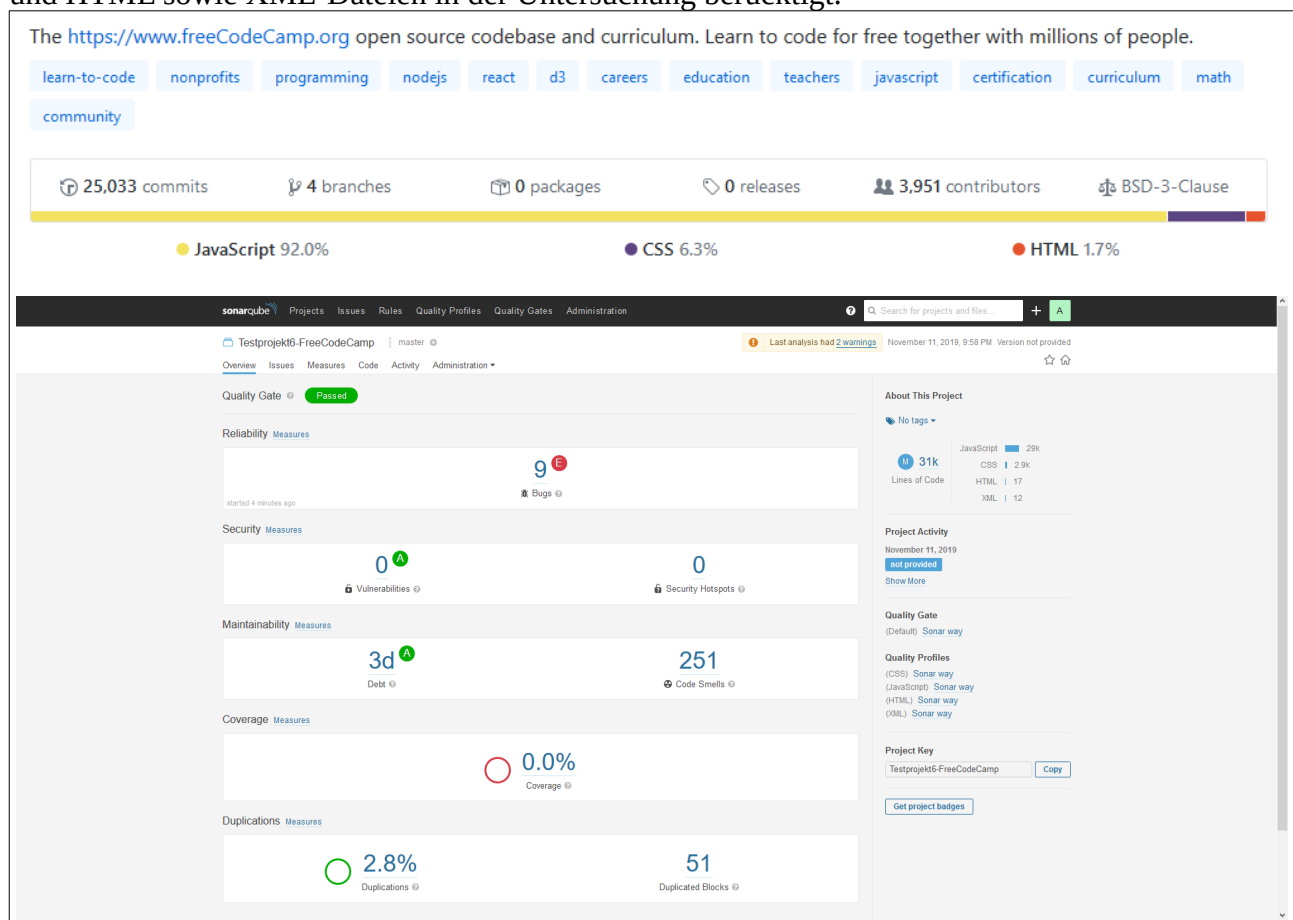
In diesem Projekt erscheinen die Probleme direkt proportional zu Anzahl der Codezeilen – bis auf ein bzw zwei Dateien, die aus der diagonalen Anordnung positiv herausstechen.



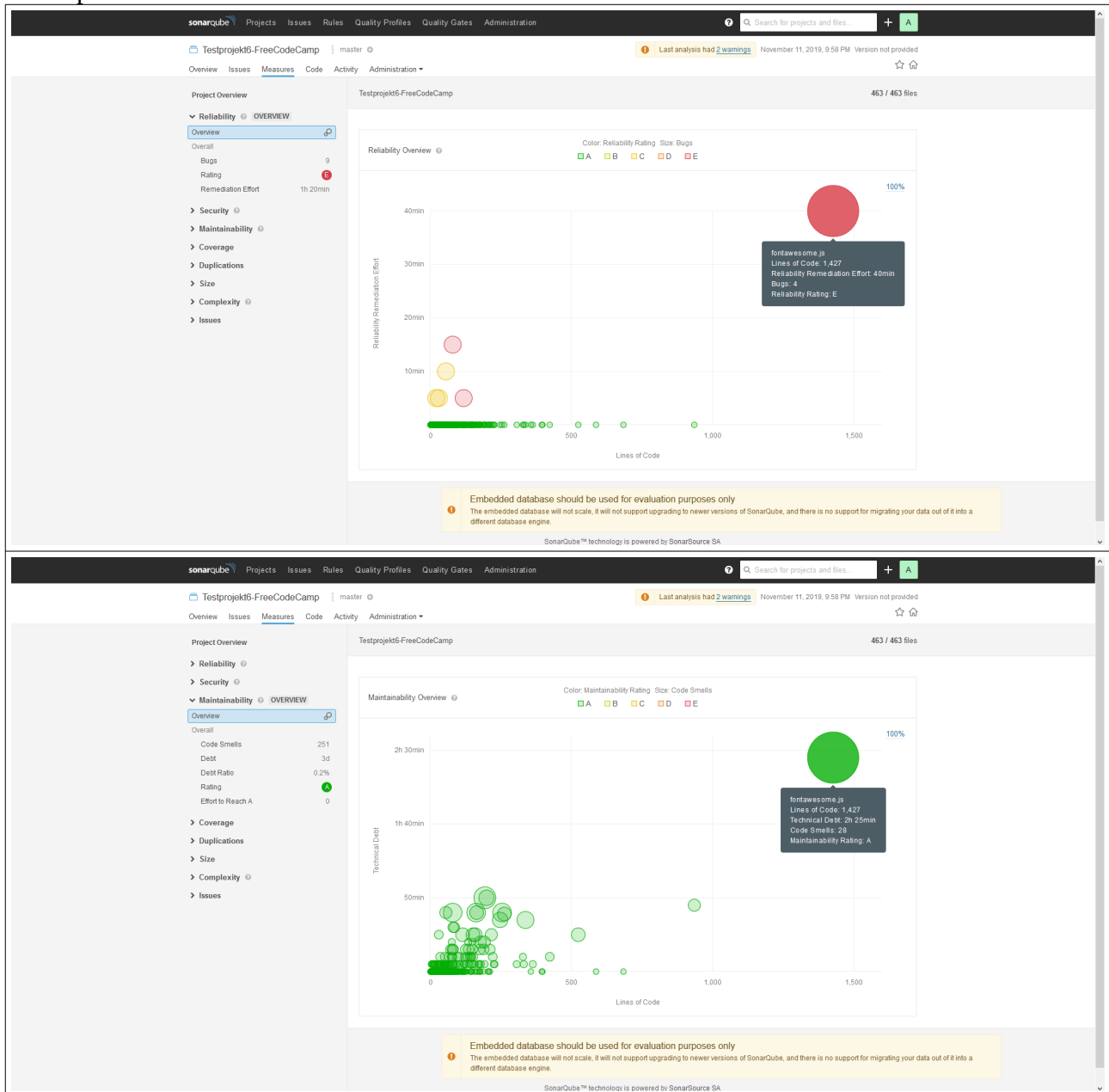
Da ich das Ergebnis von Kubernetes fast schon langweilig fand, habe ich mir aus Interesse die Dateien vom „freeCodeCamp“ angeschaut. Zu meiner Freude und Überraschung erkennt man erste Unterschiede bereits auf der Projektübersicht anhand der farblichen Darstellung.



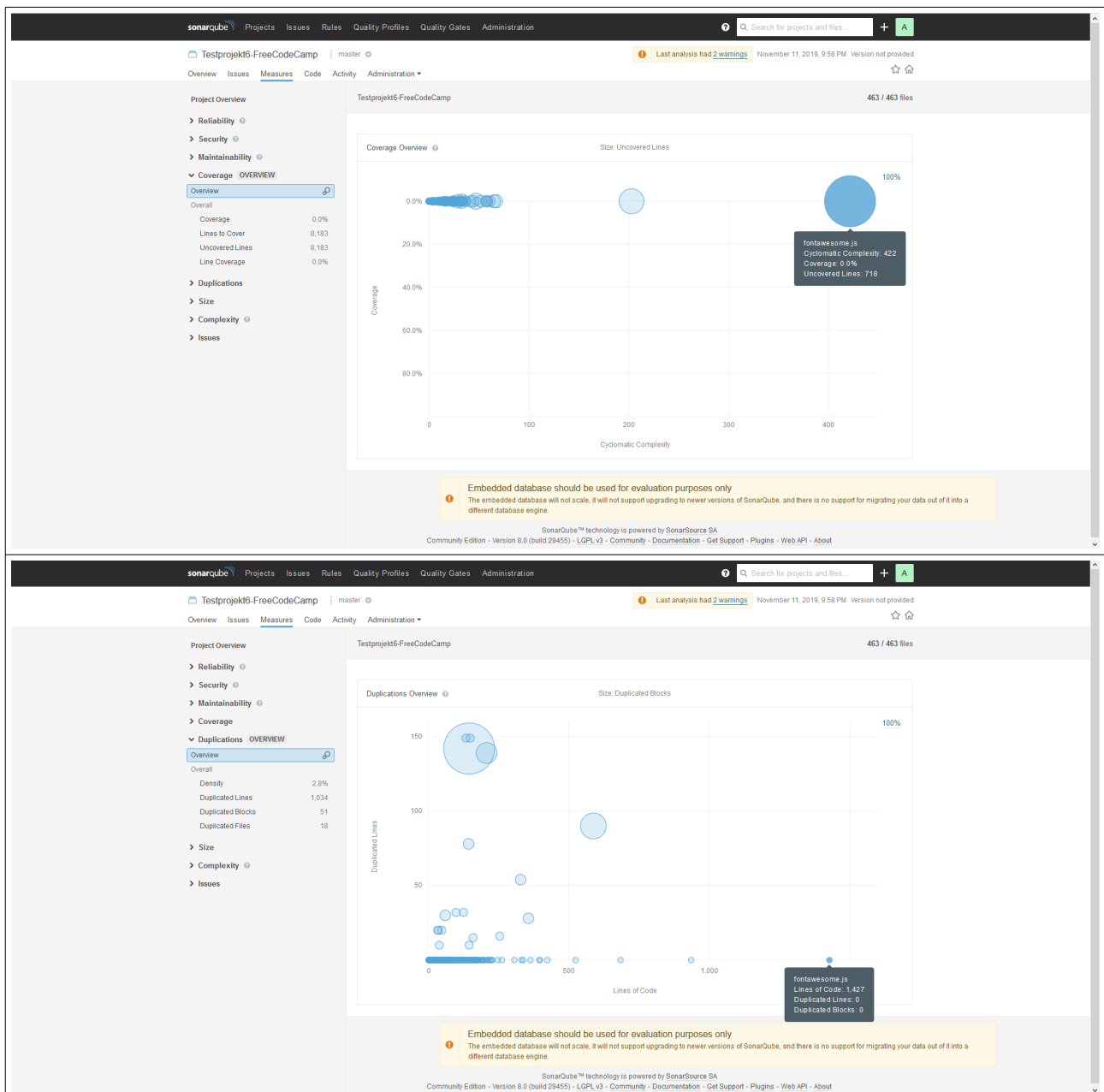
Und diesmal werden sogar zusätzlich zu den dominanten JavaScript-Dateien auch die CSS-Dateien und HTML sowie XML-Dateien in der Untersuchung berücksichtigt.



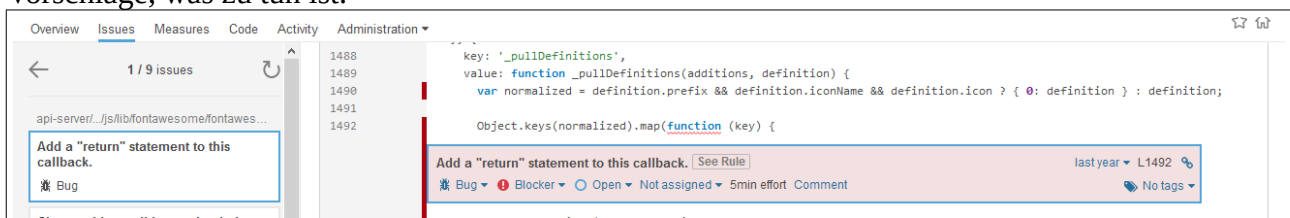
Die Verteilung der Kreise in jeder einzelnen Ansicht unterscheidet sich erkennbar von den Ergebnissen bei Kubernetes. Ich kann jedoch auf den ersten Blick nicht sagen, ob auf mich alles weniger aufgeräumt wirkt oder ob hier bereits einige Optimierung mit unterschiedlichem Ergebnis stattgefunden hat. Ganz deutlich hebt sich aber aus der Masse immer dieselbe Datei ab – und zwar in mehreren Ansichten. Wäre ich an dem Projekt beteiligt, würde ich mir als erstes diese Datei unter die Lupe nehmen.







Diese visuellen Darstellungen sind selbstverständlich nur ein Einstieg. Im nächsten Schritt würde man sich genau ansehen, welchen Teil im Code man anpacken muss. Sonarqube liefert konkrete Vorschläge, was zu tun ist.



Besonders Spannend klingt der Reiter „Activity“, der die Entwicklung über die Zeit hinweg verfolgt. Leider war nicht genug Zeit, hier tiefer einzutauchen.