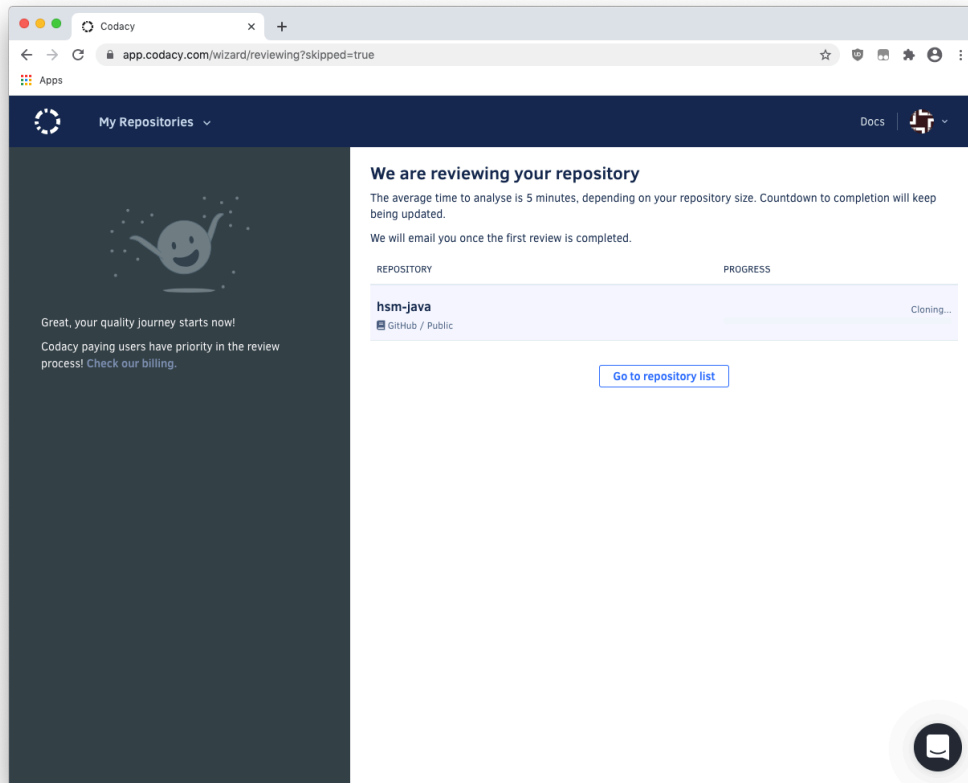


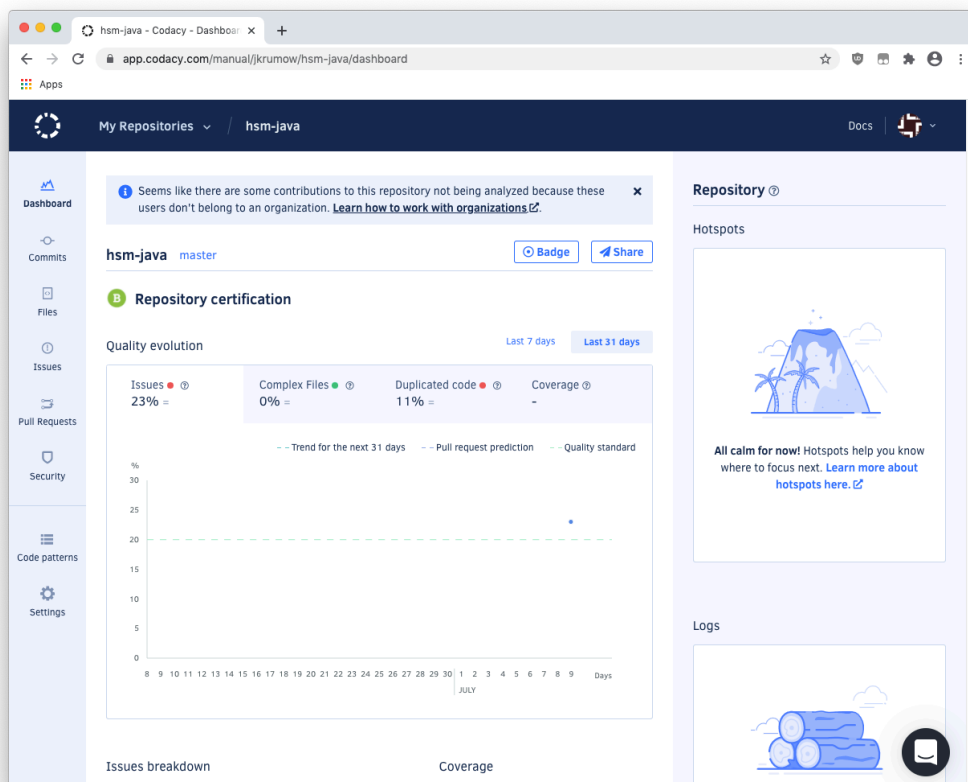
# Analyse mit Codacy

Ich habe ein Github-Repository, an dem ich mitgewirkt habe, analysiert:  
<https://github.com/jkrumow/hsm-java>

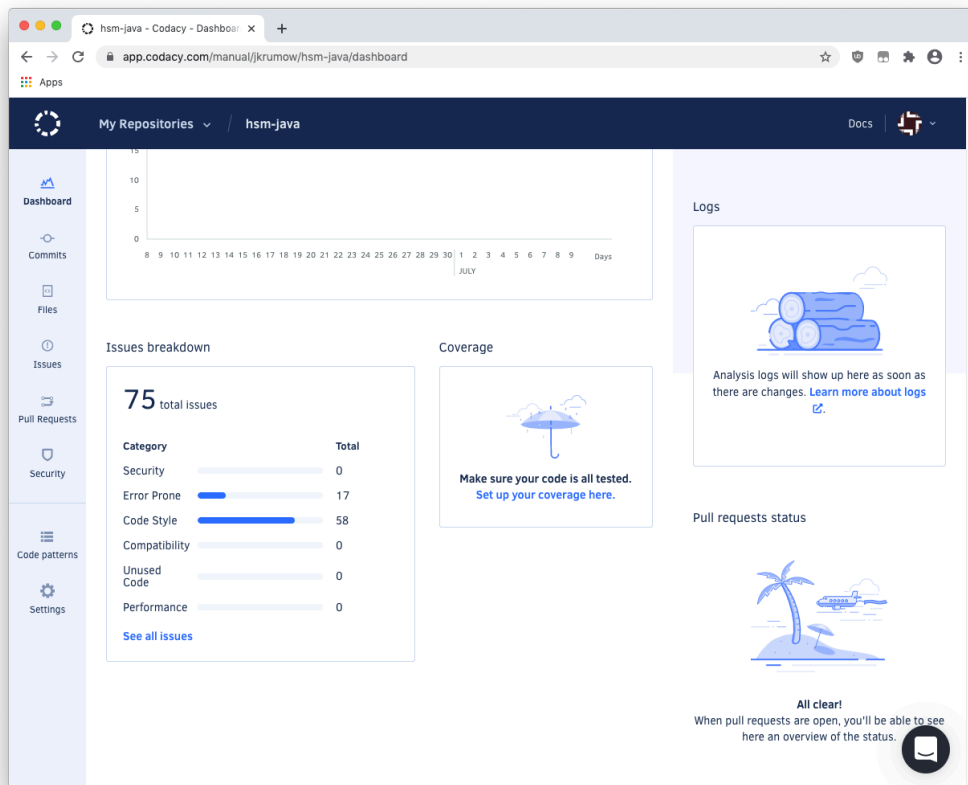
Login via Codacy und klonen des Repositories:



Übersicht über das Projekt auf dem Dashboard. Das Projekt hat die Schulnote B bekommen:



Zum Großteil wurden nur Probleme mit dem Code Style gefunden (58). Potenzielle Fehlerquellen wurden wenige ausgemacht (17):

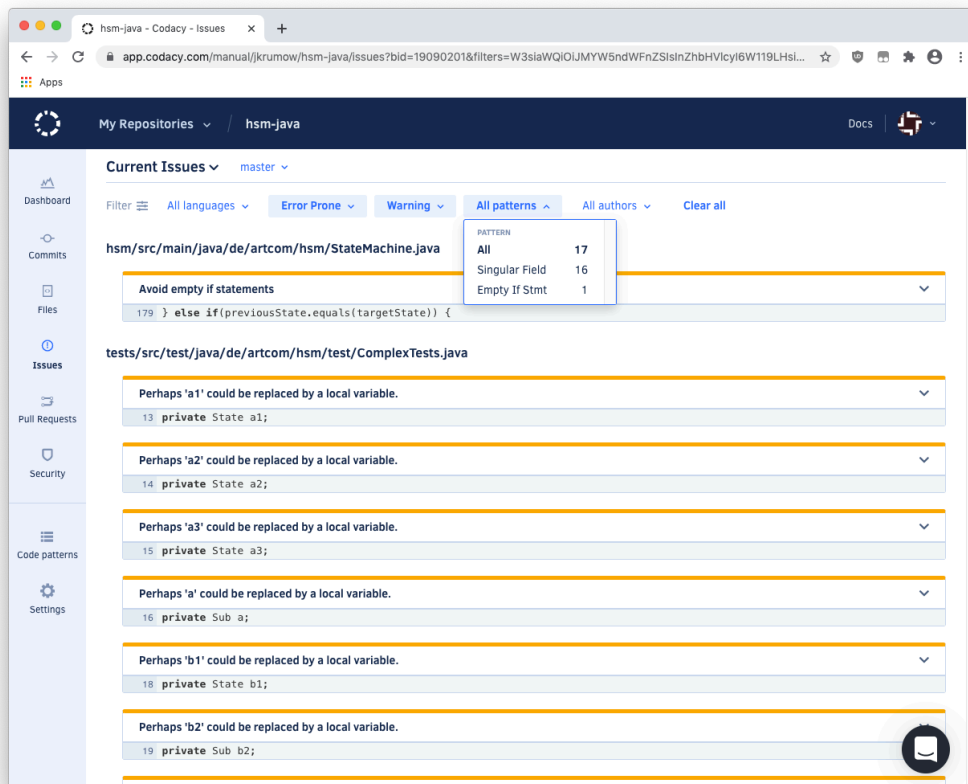


In der Dateiliste sieht man, dass sich die Benotung der Dateien vorwiegend zwischen A und B bewegt:

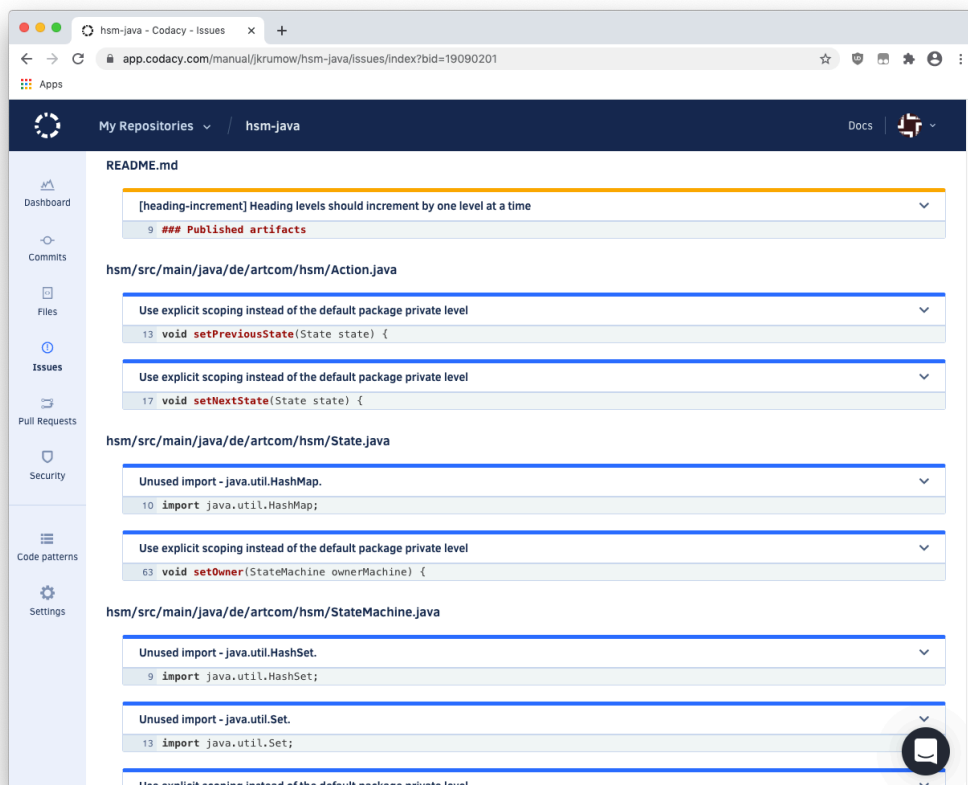
The screenshot shows the 'Files' page for the 'hsm-java' repository. The table lists files with their quality grades, issue counts, duplication, and complexity. The columns are: GRADE, FILENAME, ISSUES, DUPLICATION, and COMPLEXITY.

GRADE	FILENAME	ISSUES	DUPLICATION	COMPLEXITY
D	tests/src/test/java/de/artcom/hsm/test/ComplexTests.java	17	0	1
C	tests/src/test/java/de/artcom/hsm/test/LcaTest.java	2	4	1
C	README.md	1	-	-
C	hsm/src/main/java/de/artcom/hsm/Action.java	3	0	1
B	tests/src/test/java/de/artcom/hsm/test/LocalTransitionTest.java	5	0	1
B	tests/src/test/java/de/artcom/hsm/test/GuardTest.java	0	2	1
B	hsm/src/main/java/de/artcom/hsm/StateMachine.java	20	0	6
B	hsm/src/main/java/de/artcom/hsm/State.java	11	0	3
B	tests/src/test/java/de/artcom/hsm/test/EventHandlingTest.java	4	2	2
B	tests/src/test/java/de/artcom/hsm/test/SubStateMachineTest.java	2	0	1
B	hsm/src/main/java/de/artcom/hsm/Sub.java	3	0	2
A	tests/src/test/java/de/artcom/hsm/test/BasicStateMachineTest.java	4	0	2

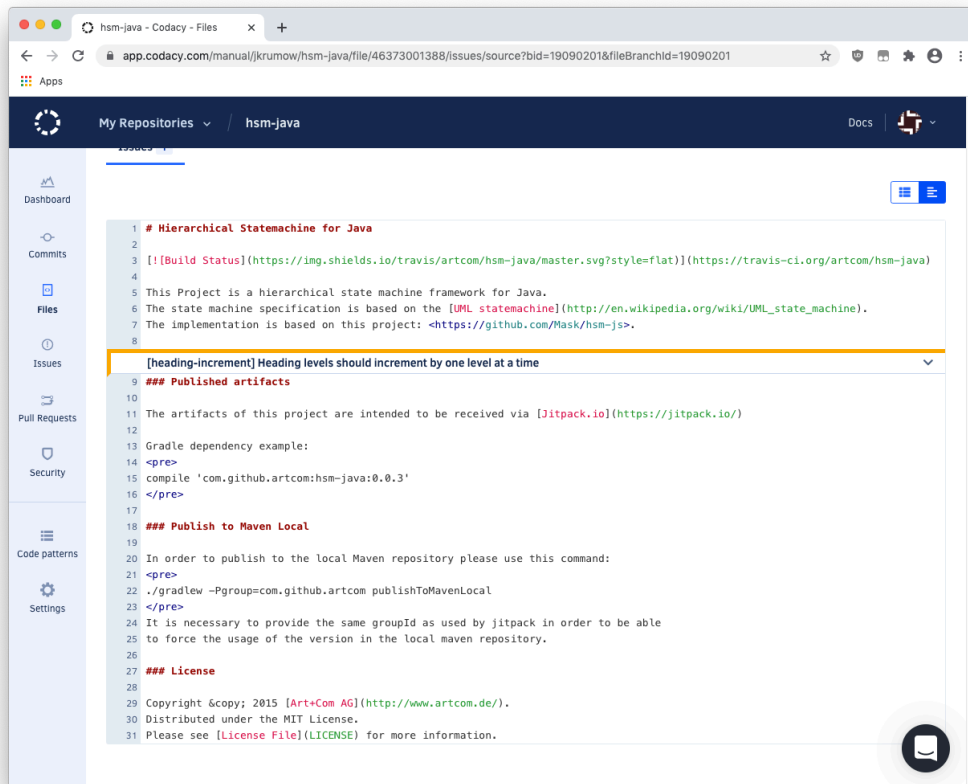
Lässt man sich die schwersten Mängel anzeigen (Error Prone, Warning) sieht man unvollständige Verzweigungen und unnötige Attribut-Deklarationen:



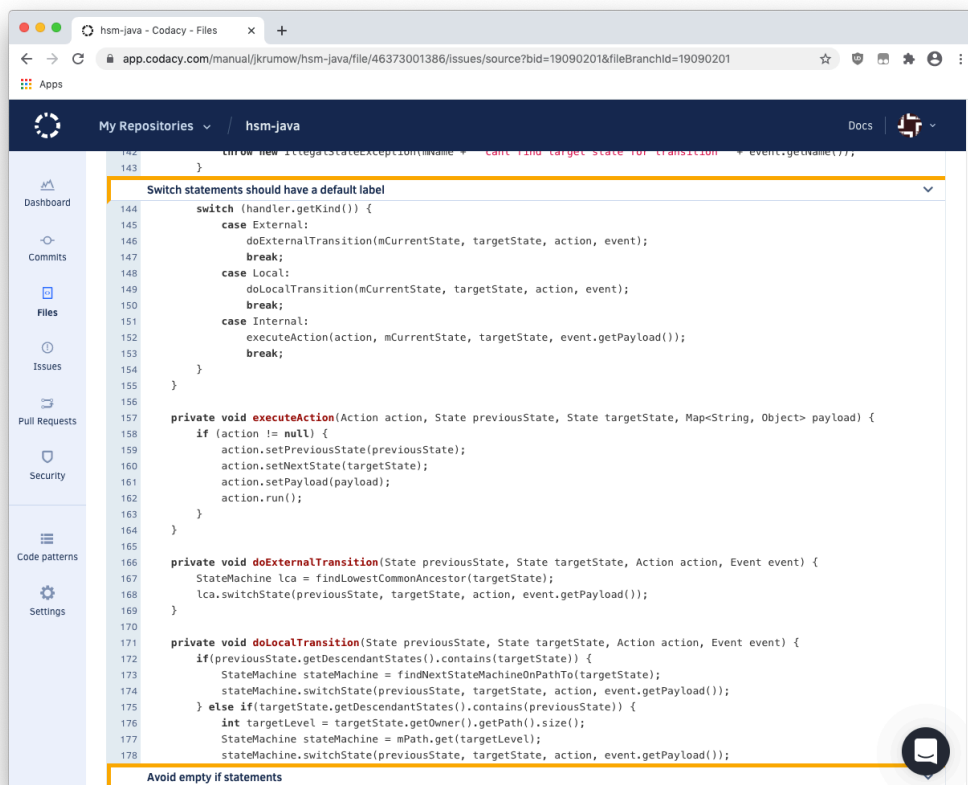
Unter den Coding Style Issues sind Sichtbarkeit der Attribute und unnötige Importe:



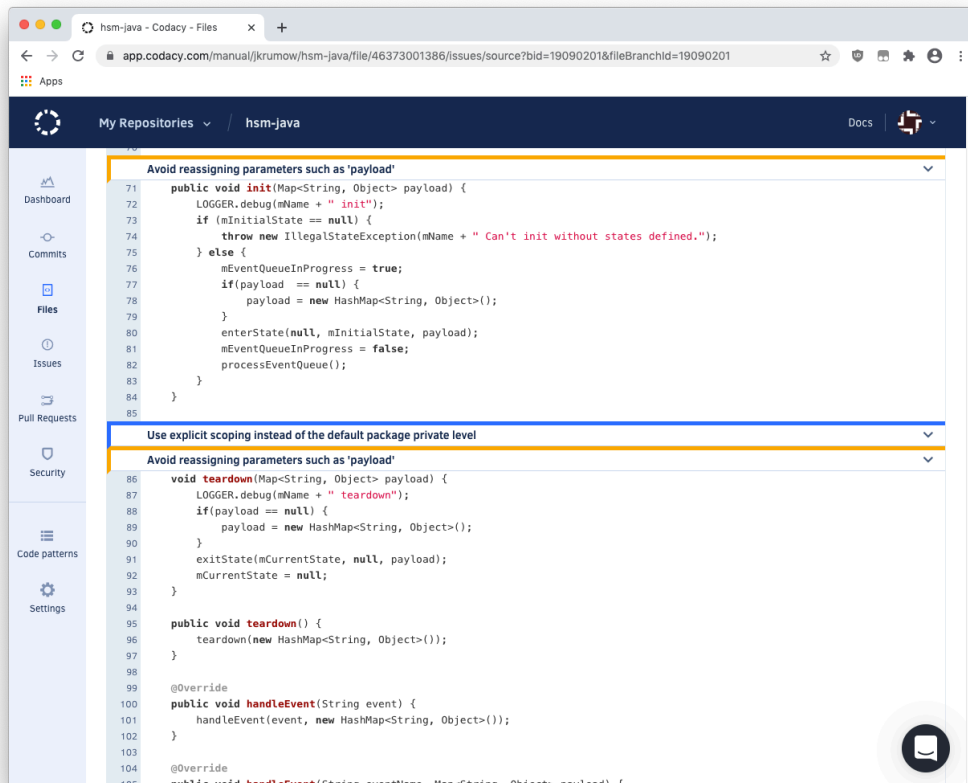
Auch die REAME-Datei wird geprüft. Hier wird bemängelt, dass die Ebenen der Überschriften nicht schrittweise durchlaufen werden, sondern sprunghaft:



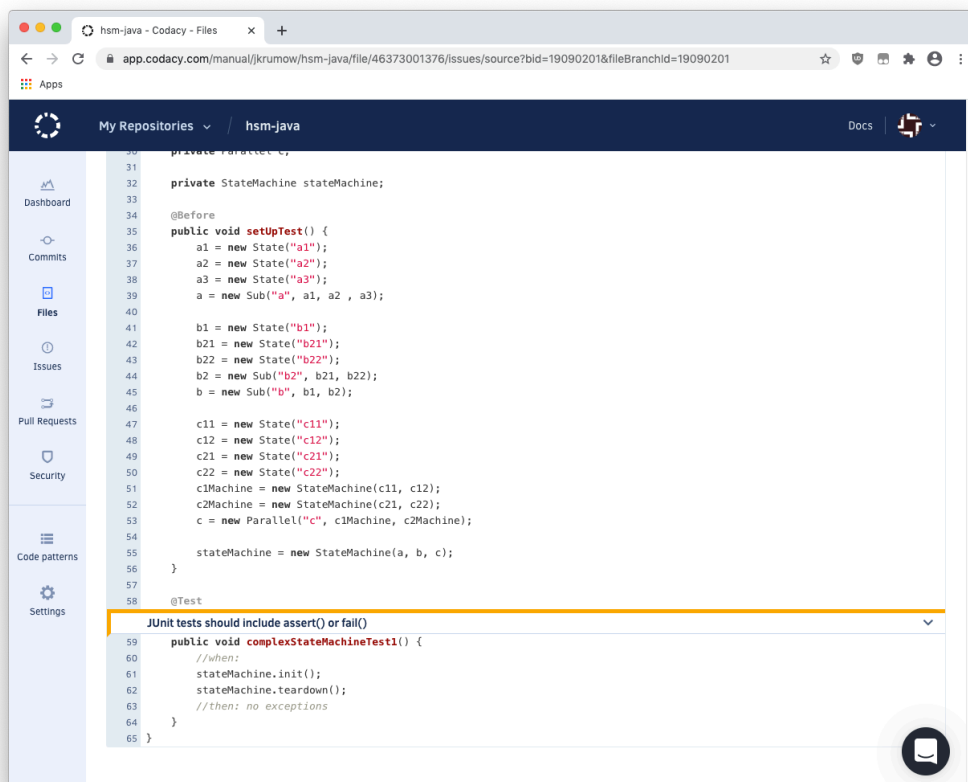
Hier die Details zu einem leeren if-Zweig:



Hier wird ein Überschreiben eines Parameters bemängelt:



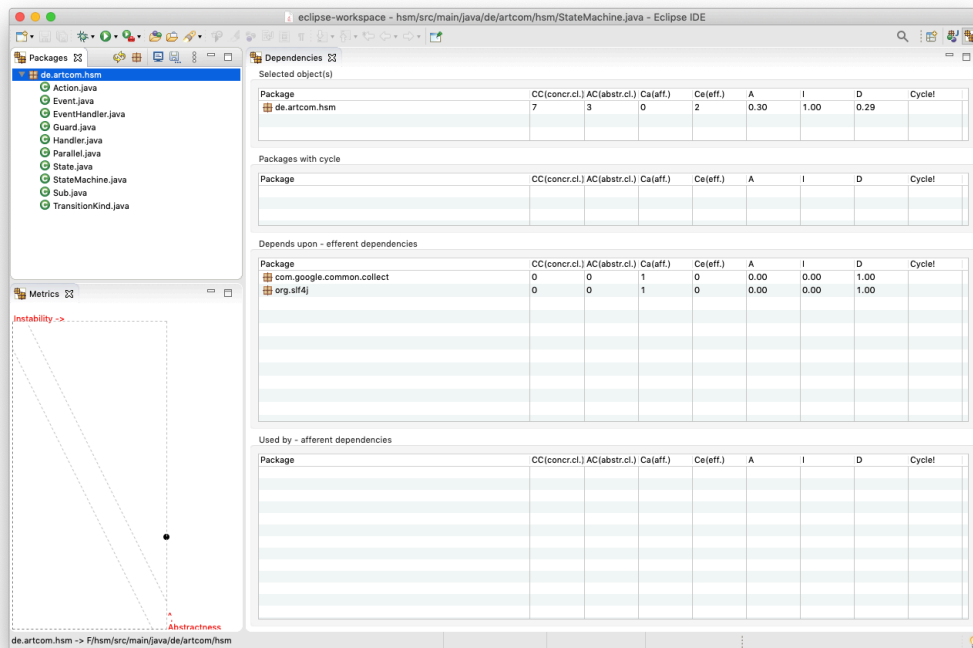
Unit-Tests werden ebenfalls geprüft. Hier fehlt die Assertion:



# Analyse mit JDepend

Das selbe Projekt wurde jetzt mit dem JDepend-Plugin für Eclipse analysiert.

Zuerst das Package `de.artcom.hsm` in dem die HSM implementiert ist:



**CC:** Das Package enthält 7 Klassen

**AC:** Das Package enthält 1 abstrakte Klasse und 2 Interfaces

**Ca:** kein abhängendes Package

**Ce:** 2 Packages werden benötigt: `com.google.common.collect` und `org.slf4j`. Diese haben wiederum ein Ca von 1 da sie vom analysierten Package anhängen

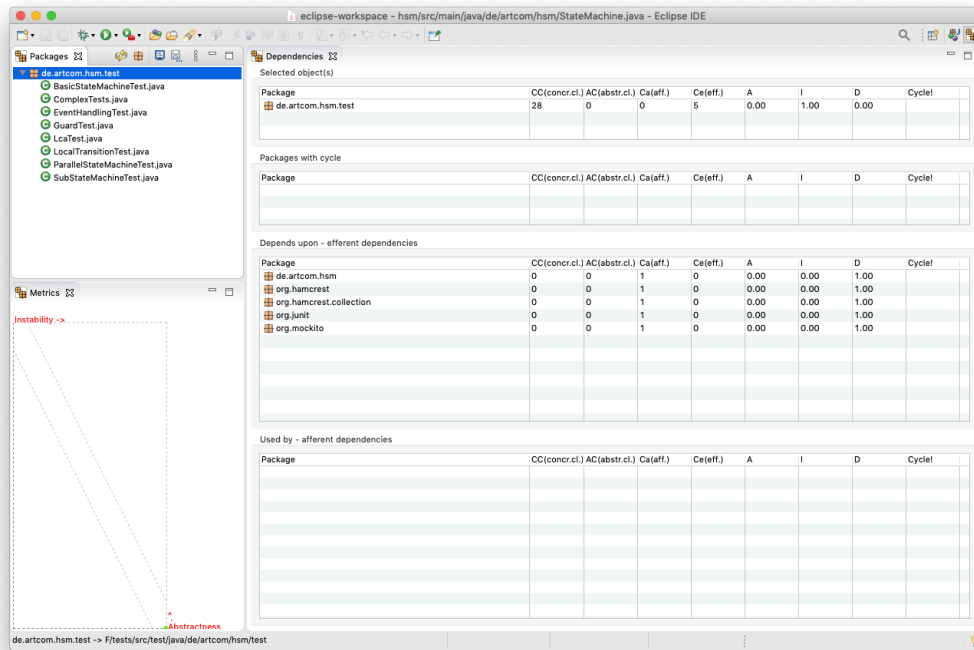
**A:** Ein Drittel sind abstrakte Klassen und Interfaces: 0.3

**I:** hohe Instabilität, da abhängig von anderen Packages, jedoch keine gleichzeitigen eingehenden Abhängigkeiten: 1

**D:** Mit 0.29 relativ nahe. Hohe Tendenz zu Instabilität, mit niedriger Abstraktion

**Cycle:** zirkuläre Abhängigkeiten kommen nicht vor

Auch die Unit-Tests wurden analysiert:



**CC:** Das Package enthält 28 Klassen

**AC:** Das Package enthält keine abstrakte Klasse Interfaces

**Ca:** kein abhängendes Package

**Ce:** 5 Packages werden benötigt. Darunter die zu testende HSM und unsere Testumgebung: de.artcom.hsm, org.hamcrest, org.hamcrest.collection, org.junit, org.mockito. Diese Packages haben wiederum ein Ca von 1 da sie vom analysierten Package anhängen.

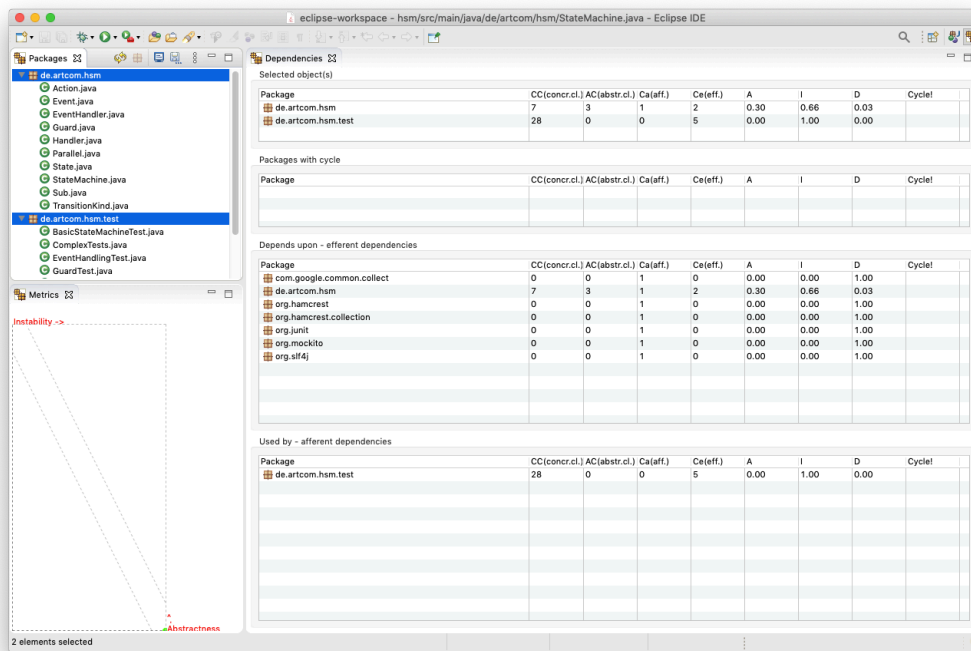
**A:** es gibt nur konkrete Klassen

**I:** hohe Instabilität, da abhängig von anderen Packages, jedoch keine gleichzeitigen eingehenden Abhängigkeiten: 1

**D:** Mit 0 sehr eindeutig. Hohe Instabilität ohne Abstraktion

**Cycle:** zirkuläre Abhängigkeiten kommen nicht vor

Analysiert man beide Packages gemeinsam gibt es folgende Werte:



Ca für die HSM steigt an, da das Test-Package diese nutzt. Dies senkt die Instabilität unter 0.7, was schlecht ist, da das HSM-Package nun eingehende und ausgehende Abhängigkeiten aufweist, was Veränderungen am Code erschwert. Es muss sowohl auf Änderungen in den importierten Packages geachtet werden und gleichzeitig die eigene API stabil gehalten werden. Dies ist insofern wichtig, da die HSM als Bibliothek für andere Projekte bereitgestellt werden soll.

Für diesem konkreten Fall ist es natürlich unerheblich, da im Test-Driven-Development beide Packages ständig aufeinander abgestimmt werden müssen. Die Werte für unser Test-Package bleiben unverändert.