***IMPLEMENTATION FOR JUnit***

JUnit is a unit testing framework for the Java programming language. JUnit has been important in the development of test-driven development and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit.

In Junit testing, we used a test fixture a state code which is tested used as input for a test. For example, a test fixture is a string of Hadoop class, which is used as input for a method. The test would validate if the method which gives the cyclomatic complexity behaves correctly with this input. A JUnit test consists of a method contained in a class which is only used for testing. We are using an assert method, provided by JUnit or another assert framework, to check an expected result versus the actual result. These method calls are typically called asserts or assert state. The assertEquals() method compares two objects for equality, using their equals(). Here it checks the expected value that we pass the results provided by SONARQUBE and the value that comes out of the JUnit method. If the two objects are equal according to the implementation of their equals () method, the assertEquals() method will return normally. Otherwise, the assertEquals() method will throw an exception, and the test will stop there showing the red bar on the console.

For the conditions to calculate the Cyclomatic complexity, we are using the definitions given by SONARQUBE.

We have taken two classes from Release 3.0.0 named as AHSproxy and ClientRMProxy done the Unit testing for these two classes. The logs and TestClientRMProxyCyclomaticComplexity.java(test class) are placed in the respective folder of Cyclomatic complexity.