

## **Measuring Software Quality**

### **Aggregating from a Security-Based Quality model**

#### **Requirements:**

Uses java 8 (should automatically use Google guava because imported)

#### **Introduction:**

The program "runs" in two separate parts. The first part adds security vulnerability information to an extension of the Quamoco Quality Meta-Model. The part that returns an actual aggregation value is the second part, which reads in a CSV file in a specific format. With that information, a MutableNetwork (using Google's guava library) is created to show the security quality model representing the data in the CSV file. There's a variety of methods that could be called on this created MutableNetwork to return an aggregation value, but ideally one would use the evaluate() method because it returns a value between 0 and 1 for scale.

#### **Instructions:**

A CSV file (GRAPH.csv) containing info gathered from the Parasoft static analysis test and CWE.mitre.org will be read in when you run QMTest.java.

#### **FORMAT OF CSV FILE:**

- The first line will contain 1) the number of factors, 2) the number of impacts, 3) the number of product factors, and 4) the number of quality aspects.
- The impacts will then be listed - each on its own line - in the following format: Name of Impact (ex. OEdge1), Effect of Impact (FALSE for negative, TRUE for positive), Severity of Impact (ex. 3), Name of Factor the impact is assigned to (ex. CWE-397...), Name of the Factor that is the impact's target (ex. Integrity)
- Next, the product factors are listed - each on its own line - in the following format: Name of Product Factor (ex. CWE-397...), Multiplicity of Product Factor (ex. 25), Name of FIRST Impact assigned to the Product Factor (ex. OEdge1), Name of SECOND Impact assigned to the Product Factor IF APPLIES (ex. OEdge2)
- The quality aspects are then listed - each on its own line - in the same format as the product factors.

As the CSV file is read into QMTest.java, the Quamoco Quality Meta-Model Extension is populated. QMTest.java then exports another CSV file called GRAPH1.csv for GraphGenerator.java to read in.

The CSV file outputted by QMTest will be in the format required by GraphGenerator.java.

Save GRAPH1 somewhere and copy the file path as a string into the constructor of the GraphGenerator object called "sample" in the main class, Graph.java.

Now, you can run Graph.java and it will output information about the graph and the various aggregation values you can get with it. There are other public methods in GraphGenerator.java, QualityElement.java, and Impact.java that will allow for printing out properties of the graph created.