CSSE 376 – Lab8 – Caio Balthazar – CM 1478

[Question 1 (5 pts)] Please identify the metrics whose values are outside the optimal range. Also identify the methods that are responsible for this.

Number of Parameters (avg/max per method) is out of the optimal range due to the TwitterClient constructor of the TwitterClient.java class having 7 parameters.

[Question 2 (5 pts)] Please provide a strategy or solution (without writing code) that will bring the value of the given metrics into an optimal range.

We could use the refactoring technique called Introduce parameter object. This object would contain the necessary parameters for the method and reduce the number of objects passed to the constructor drastically.

[Question 3 (5 pts)] A good heuristic for cyclomatic complexity is to keep it below 15. Does this code meet that heuristic?

Yes. The code has a mean of 2.478 McCabe Cyclomatic complexity and a standard dev. of 2.124. The maximum value for that code is 10.

[Question 4 (5 pts)] Cyclomatic complexity can be used to identify the number of independent paths that need to be tested in a method. Please identify the number of independent paths in the method backOff in the inner class BackOff in TwitterClient.java. Identify conditions that would lead to each of these paths. (These conditions establish test cases for the method.)

There are 3 independent paths. The conditions are:

- backOffMillis == 0
- backOffMillis != 0 && backOffMillis <= capMillis
- backOffMillis != 0 &&backOffMillis > capMillis

[Question 5 (5 pts)] Explain, in your own words, the “afferent coupling" and “efferent coupling" methods. Describe how they can be used in project analysis.

Afferent couple refers to the number of external/different packages that depend on classes of this particular package. It indicates the package’s responsibility.

Efferent couple refers to number of external/different packages that this particular packages depends on. It indicates the package independence.

They both can be used to analyze the level of coupling and dependency between packages and decide upon better code abstractions if this dependency is higher than desired.

[Question 6 (5 pts)] Compute the effectiveness of Top Level Design inspection activities.

806/(154 + 928)*100 = 74.4%

[Question 7 (5 pts)] Compute the effectiveness of Low Level Design inspection activities.

761/(154+928-806+948) * 100 = 62.1%

[Question 8 (5 pts)] Compute the overall defect removal effectiveness of the development process.

(1 – 126/3526) * 100 = 96.4%