CS/SE 6356 Software Maintenance, Evolution & Re-engineering
Spring 2024
Assignment 3: Coupling and Cohesion

Assignment date: **March 18**

1. Goals of the assignment

In this assignment, you will learn to interpret coupling and cohesion measurements. Each team must complete the following instructions. You will measure and analyze coupling and cohesion of class in the **two** systems you made changes to in the previous assignments: MangoDB and JEdit.

Note: this is a team assignment. Communication with other teams is allowed, but only to address technical issues that you may encounter with the setup processes. You must explicitly document in the deliverable what (if any) helped you received from other students/teams in the class.

2. Investigate software analysis tools

Below is a list of several tools that can be used to measure software properties. Familiarize yourself with them (or at least a subset). Then pick one or multiple tool(s) that you can use to measure software properties. The total number of measures provided by the tool(s) should include at least two coupling and two cohesion measure.

Also pick a tool that can detect code smells, which we will use in the next assignment.

NOTE: Feel free to look beyond these tools.

SourceMeter - Tool built for the precise static source code analysis of Java projects.
https://www.sourcemeter.com/

SonarQube - Open platform to manage code quality.
http://www.sonarqube.org/

MetrcisReloaded - Automated code metrics plugin for IntelliJ IDEA
https://plugins.jetbrains.com/plugin/93-metricsreloaded

CodeMR – an architectural software quality and static code analysis tool
https://www.codemr.co.uk/

CK - class-level and method-level code metrics in Java projects
https://github.com/mauricioaniche/ck

Metrics 3 – an Eclipse plug-in for computing software metrics
https://github.com/qxo/eclipse-metrics-plugin

Decor (Ptidej) - Design smell detector
http://www.ptidej.net/tools/designsmells/

InsRefactor - Eclipse-plugin helping developers to identify code smells instantly and pushing developers to resolve code smells
https://liuhuigmail.github.io/tools/InsRefactor.htm

JDepend - Generates design quality metrics for Java packages
http://clarkware.com/software/JDepend.html

Lint - A code scanning tool that can help you to easily identify and correct problems with the structural quality of your code
http://developer.android.com/tools/debugging/improving-w-lint.html

RefactorIT - Automated refactorings, source-code metrics, audits and corrective actions
http://refactorit.sourceforge.net/

Stench Blossom - A code smell detector that provides an interactive ambient visualization
https://github.com/DeveloperLiberationFront/refactoring-tools

3. Measuring and interpreting coupling and cohesion

Use any of the tool(s) you prefer from the list above or another one you may have found online.

1. Select two coupling and two cohesion metrics for measuring the coupling/cohesion of MangoDB and JEdit. Carefully read and understand the description of the selected metrics.
2. Compute the selected metrics for each class of the two systems. Exclude test classes.
3. For each metric you chose, select the top three non-trivial classes with the highest **cohesion** and the three classes with the lowest cohesion (according to the selected metrics). Explain why the selected classes have high/low cohesion and what type of cohesion is being measured (according to the types discussed in class). Analyze and explain the differences between the classes with the highest cohesion versus the classes with the lowest cohesion, and also among the classes with the highest/lowest cohesion, produced by the different metrics. Explain what type of cohesion you observe, and contrast why it is not another type (from the five types we discussed in class).
4. For each coupling measure, select the top three non-trivial classes with the highest **coupling** (this can be derived by adding coupling between classes) and the three classes with the lowest coupling (according to the selected metrics). Explain why the selected classes have high/low coupling and what type of coupling is being measured (according to the types discuss in class). Analyze and explain the differences between the classes with the highest coupling versus the classes with the lowest coupling, and also among the classes with the highest/lowest coupling, produced by the different metrics. Explain what type of coupling you observe, and contrast why it is not another type (from the seven types we discussed in class).
5. Save the output of the analysis tools in your repository where you made the changes for the previous two assignments.

4. Deliverables

This is a team assignment, so only one submission per team is necessary.

Submit a PDF document on eLearning containing:

1. List of team members
2. Indicate the tool(s) you used or computing the metrics.
3. A brief description of the metrics selected (use your own description).
4. Your answers for step 3.3
5. Your answers for step 3.4
6. Indicate in which file/folder you saved the metrics in the repositories.
7. Describe briefly the tasks performed by each team member.
8. Contribution information. Include the name of each team member and how much they contributed to this assignment (in percentages). For example: John Doe 40%; Jane Doe 60%. The percentage should add up to 100%.

5. Rubric

Item	Points
Metrics description	10
Analysis of the classes with the highest/lowest **cohesion** for each metrics	45
Analysis of the classes with the highest/lowest **coupling** for each metric	45
Total	**100**

You will receive points off if:

- You are not specific enough when describing your analysis.
- Your document contains typos and writing errors.