PV260 Software Quality

Assignment 1 - Software Metrics And Disharmonies

Spring 2015

1 General Information

1.1 Dates

- Assignment start: 3.3.2015
- Assignment deadline: 16.3.2015, 23:59

1.2 Submission

Please submit your solution as a pdf report and submit it to the Homework vault (Odevzdávarna) called *Assignment 1: Metrics (Groups 01, 02)*. The name of the pdf file should be as follows: *lastname1-lastname2-assignment1.pdf*. The tasks should be solved in the groups of two (exceptionally in groups of three). The document must include the names of the group members. Only one report per group should be submitted.

1.3 Evaluation

The maximum points for this is assignment is 10. The points will be distributed based on the quality and completeness of the reports. Additional points can be awarded by completing the optional bonus tasks.

1.4 Tools

- Moose 4.3: `https://code.google.com/p/moose-technology/downloads/`
- InFamix: `http://www.intooitus.com/products/infamix`
- iPlasma: `http://loose.upt.ro/reengineering/research/iplasma`
- SonarQube: `http://www.sonarqube.org/`
- CodeCity: (optional) `http://www.inf.usi.ch/phd/wettel/codecity.html`

1.5 Project Selection

There are three projects available for this assignment:

- JFreeChart: `http://www.jfree.org/jfreechart/`
- JEdit: `http://www.jedit.org/`
- FreeMind: `http://freemind.sourceforge.net`

Each group has to select one project and use it to complete the following tasks.

2 Mandatory Tasks

In the report, always include answers/comments and diagrams from the tools that are relevant to the individual tasks.

2.1 Task 1

- Use Moose/iPlasma tool to compute the *Overview pyramid.*
- Interpret the results - what can you infer about the size and complexity, coupling and inheritance hierarchies?

2.2 Task 2

- Generate the *System complexity view.*
- What are the top 5 of the biggest classes in terms of number of attributes, methods and lines of code? What are the classes used for?
- Use *Customized system complexity view* to find the classes with the highest cyclomatic complexity and weighted method count (WMC).
- Which inheritance hierarchy is the largest one (in terms of depth and width)? What is the purpose of the classes involved?

2.3 Task 3

- Use *System complexity view* and *Class blueprints* to identify potential *God classes*, *Brain methods* and *Tradition Breakers* (3-4 examples of each is enough)
- Besides relevant screenshots, explain why do you think these classes suffer from the above mentioned disharmonies.

2.4 Task 4

- Use the iPlasma tool to identify the identity, collaboration and classification disharmonies.
- Select at least three examples from each category. Describe in detail why those disharmonies occurred, what problems they can cause and how they could be remedied (We highly suggest to read through the chapters about disharmonies provided in the study materials).
- For each of the examples, add relevant samples of the source code.

2.5 Task 5

- Compile the source code.
- Analyse it with Sonar and send the results to our SonarQube server (Note that the server can be accessed only from the MU network)
- Examine the Dependency Structure Matrix. Are there any dependency cycles between packages and classes?
- Which packages and classes have the most cycles?
- Are there any popular packages/classes, which are frequently used by other packages/classes? How high/low are they in the dependency hierarchy?

3 Bonus Tasks

- Discuss any other interesting observations about the selected project based on metrics values and visualizations from the available tools. For example, you can also use CodeCity tool to produce and interpret additional findings.
- Select one of the discovered disharmonies and try to remove it by improving the source code. It is not necessary to make the code compilable after the modification. However, you should document the changes and provide the proofs (iPlasma/Moose output, relevant metrics values before and after the change, etc.) that the disharmony has been removed.