Software Engineering Methods 2014

Instructor: Dr. Alberto Bacchelli

Teaching Assistants: Friso Abcouwer, Rob van Bekkum, Moritz Beller, Thijs Boumans, Aimee Ferouge, Jan Giesenberg, Michael de Jong

Assignment 4

Weeks 07–08

The goal of this assignment is to use Software Metrics. To correctly complete this assignment you **must**:

- Carry out the assignment with your team only–unless otherwise stated. You are free to discuss solutions with other teams, but each team should come up its own personal solution. A strict plagiarism policy is going to be applied on all the artifacts submitted for evaluation.
- Your team has to complete the assignment by following the SCRUM methodology:
  - You have to submit a sprint plan for this assignment, using the template "Sprint Plan Template" available on Blackboard, by **Oct 14, 2014 @ 13:00**.
  - You have to submit a sprint reflection for this assignment, using the template "Sprint Reflection Template" available on Blackboard, by **Oct 21, 2014 @ 13:00**.
- Solutions to this assignment will consist (depending on the exercise) in changes to the source code of your project and in explanations (*e.g.*, of decisions taken):
  - All the explanations must be included in a single PDF file with the name: *Group[devhub id]-[AssignmentNumber].pdf* (a correct name would be: *Group1-4.pdf*).
  - Changes and explanations must be pushed to the master branch of your Devhub repository by **Oct 18, 2014 @ 23:55**.[1]

Exercise 1 - 20-Time, reloaded (45 pts)

1. In this exercise you can decide what to do next on your game:[2] It can be an extension/improvement from any perspective, such as improved code quality, or novel features.

   Define your requirements and get them approved by your teaching assistant. The implementation and process will be based on the same criteria used for the working version, plus it will take into account whether you use design patterns and advanced object-oriented programming (**30 pts**).

2. During the analysis and design phases of this extension use responsibility driven design and UML (push to the repository the *single* PDF file including all the documents produced) (**15 pts**).

[1] Solutions sent within the first 24 hours after the deadline will be given 50% of the points they would normally get. Solutions sent after 24 hours from the deadline will not be graded.

[2] Consider that this exercise is worth 58% of the non-optional points of this assignment, so plan its load accordingly.

Exercise 2 - Software Metrics (32 pts)

The *inCode* tool[3] uses software metrics to detect a number of design flaws.[4] In this exercise you will use it to have guidelines to improve your implementation, from a code quality perspective.

1. Use inCode[5] to compute software metrics on your project, then upload the resulting analysis file[6] to your git repository. Write in the explanation PDF file where the analysis file is located (**2 pts**).

2. Consider the 'System summary' view (see Figure 1, Point 1) regarding the analysis of your project. You can see the design flaws that seem affect your system.[7] Pick the first **three** design flaws (in order of severity, see Figure 1, Point 2) that affect your software, and for **each** flaw complete the following points:

   a) Explain the design choices or errors leading to the detected design flaw (**3 pts**).

   b) Fix the design flaw or extensively and precisely explain why this detected flaw is not an error and, thus, should not be fixed (**7 pts**).

   If your project has less than three design flaws, congratulations! In this case, consider other design flaws (to reach a total of three with the previous ones) that inCode could detect, and explain in detail where each of these design flaws could have probably affected your system and how you managed to avoid it (**10 pts** per design flaw).

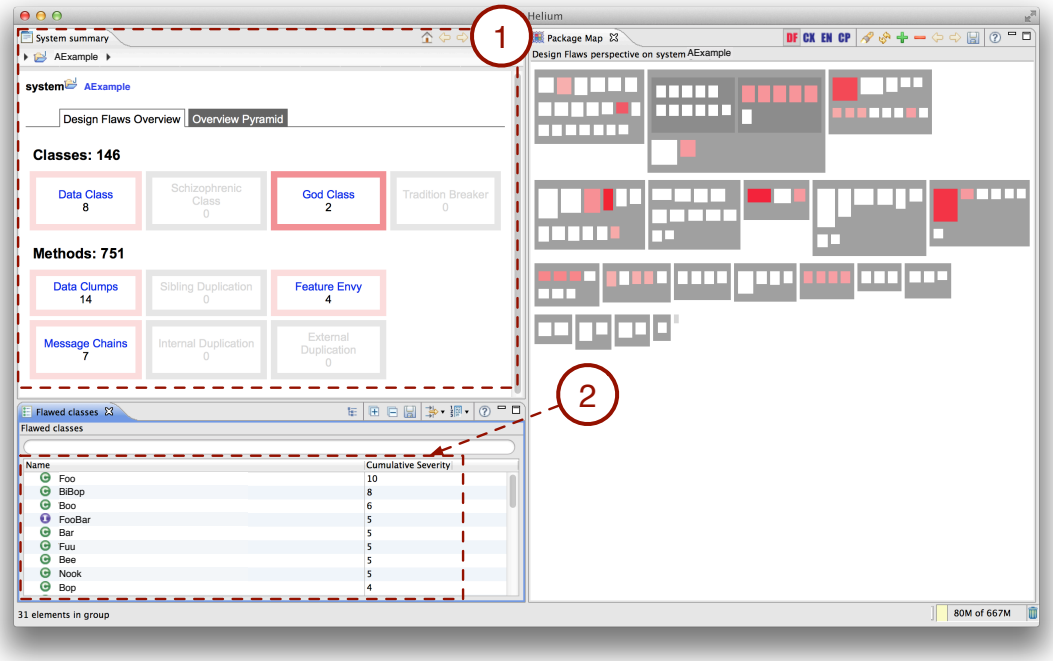

Figure 1: inCode analysis windows

[3] https://www.intooitus.com/products/incode

[4] Described in: `https://www.intooitus.com/products/incode/detected-flaws`

[5] Download the trial from: `https://www.intooitus.com/products/incode`. We applied for the Classroom license (`https://www.intooitus.com/products/incode/buy-it`) which should be granted in the next days. In the meanwhile, the trial has sufficient features to complete this exercise.

[6] It can be found in the folder 'snapshots' in the inCode installation folder

[7] Remember: Software metrics only are half of the truth, so it is possible that detected flaws are not actual flaws.