Software Engineering Methods Q1 2016

Instructor: Dr. Alberto Bacchelli   Assignment 04

Teaching Assistants: Liam Clark, Robin van Heukelum, Bart de Jonge, Jean de Leeuw, Thomas Overklift, Mark van de Ruit, Wouter Smit, Martijn Steenbergen, Gijs Weterings   Week 07

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.

- Complete the assignment by following the SCRUM methodology. Regarding this, you must push the following documents on the main branch of your GitHub repository by the specified deadlines and tag the commit that your TA has to consider for grading:

    - A sprint backlog for this assignment, using the template "Sprint backlog - template" available on Blackboard, by **Oct 17, 2016 @ 10:45AM**.

    - A sprint retrospective for this assignment, using the template "Sprint retrospective - template" available on Blackboard, by **Oct 21, 2016 @ 17:55PM**.

- Provide solutions to the exercises. Each solution will consist of changes to the source code of your project and/or explanations (*e.g.*, of decisions taken):

    - The explanations must be written in a PDF file with the name:
      Group[id on google spreadsheet]-[AssignmentNumber].pdf[1]

    - Changes and explanations must be pushed to the master branch of your GitHub repository by **Oct 21, 2016 @ 17:55**[2] and the commit that you TA has to consider must be tagged.

---

[1] *e.g.*, a correct name would be: *Group0-04.pdf*.
[2] 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.

Exercise 1 - Code improvements (30 pts)

1. Where do you use *if* or *case* statements in your source code? Refactor 10 of these statements used in different scenarios,[3] so that they are not necessary anymore and describe your refactoring.

   Instead of *if* or *case* statements you can use, for example, design patterns, polymorphism, or a technique called double dispatch.[4]

   If you have less than 10 of these *if* or *case* statements, you have less work to do!

   If you find some rare cases of *if* or *case* statements that cannot be refactored you can convincingly explain why (we recommend to consult with your TA before going this way); these cases count toward the 10 you have to refactor (**30 pts**).

Exercise 2 - Teaming up (30 pts)

1. In this exercise, you decide *together* with your TA, during the group meeting on Monday, *new game features* to add to your game.[5]

   After you decide on the tasks, write a requirements document, which will be evaluated in the same way as for the requirements document of the initial version. Afterwards you must implement the requirements. (**22 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) (**8 pts**).

Exercise 3 - Walking in your TA's shoes (30 pts)

Reading, understanding, and evaluating code written by other software engineers is the bread and butter of professional software engineering. Ask your TA for the codebase of another group (which will be provided as an anonymous .zip archive) and perform the following tasks:

1. Grade the *code quality* according to the rubrics. You have to deeply detail the reasons for your choices.[6] (**15 pts**).

2. Propose meaningful enhancements to the other's group codebase. Meaningful enhancements are, for example, based on design patterns and design principles. These enhancements must be worth 30 points for a weekly assignment. (**15 pts**).

Push to the repository the *single* PDF file including the output of the aforementioned two tasks.

---

[3]An *if* statement with multiple *else*s or a *case* statement with multiple cases count as a single scenario

[4]You can read about it on the Internet, for example here: `http://www.javaperformancetuning.com/articles/ddispatch.shtml`

[5]Both you and the TA will have to take into account that this exercise is worth one-third of this assignment

[6]At least one A4-page with 2 cm margins and text in font size 10 is expected.