AA 2016-2017 Software Engineering 2—Project goal, schedule, and rules

READ THIS VERY CAREFULLY
NO EXCUSE FOR IGNORING WHAT WE WRITE HERE

Table of contents

1 Software Engineering 2 project: goal and approach

The objective of this project is to apply in practice what you learn during lectures with the purpose of becoming confident with software engineering practices and able to address new software engineering issues in a rigorous way. The project includes five assignments:

1. The preparation of a Requirement Analysis and Specification Document (RASD) for a problem we provide you.
2. The definition of the Design Document (DD) for the system considered in point 1 above.
3. A testing-related activity focusing on the same project considered in point 1 and 2 above.
4. An assessment of the effort and cost required for the development of the project considered in the points above.
5. A code inspection and bug identification activity. This activity will focus on the analysis of an existing well-known open source project.

All assignments will be reviewed during the final discussions that will be scheduled in January and will be held in February. The evaluation will assess the quality of the artifacts you prepare (accurateness, completeness, soundness) and the quality of your presentation (if you are able to explain your point in an appropriate way and if your presentation fits in the allowed time). Please check the introduction to the course for more information on the evaluation criteria for the project. All assignments except the fifth one are described in the following of this document

2 Project schedule

- Group registration deadline 16/10/2016

- RASD submission deadline 13/11/2016
- DD submission deadline 11/12/2016
- Testing document submission deadline 15/01/2017
- Project management deadline 22/01/2017
- Code inspection deadline 05/02/2017
- Final presentation (to be scheduled)

All deadlines are assumed to expire at **23:59** of the days listed above.

3 Rules

- The project has to be developed in groups of two or three persons. Groups composed of a single student are allowed even if strongly discouraged. The assignments and the corresponding expectations of the professor will be calibrated based on the size of the group.
- Each group MUST register to the project following the steps indicated in Section 4. "Mixed" groups involving students of the two sections are allowed. When registering, each such group will need to indicate a single "reference" professor. This will be the one holding the discussion at the end of the course and deciding the grade. The choice is up to the students.
- Each group MUST provide the requested artifacts within the stated deadlines. A delay of a few days, if notified in advance to the reference professor, will be tolerated but it will also result in a penalty in the final score. It is mandatory to provide these artifacts and to present them to the reference professor in a final meeting that will be scheduled later.
- The material presented in one artifact is not fixed in stone. You can (and are encouraged to) provide updates after the discussion sessions in the project lab or at any point before the discussion at the end of the course, if you think these are needed.
- During the development of the project each group will keep track of the number of hours each group member works toward the fulfillment of each deadline.
- For any question related to the project that could be interesting also for the other groups, please use the forum available on the Beep website. We will answer as promptly as possible.

4 Group registration

You should form your group and register it by going through the following steps:

1. Create a repository for your project on github (https://github.com). Make sure that all group members have a github account and have access to the repository. We would like to see commits performed by all group members.
2. Register your group by filling in the following form https://goo.gl/forms/yin2E1b0GxPifTfr2. Do not forget to include in the form all relevant data!

5 The problem for assignments 1, 2, 4, and 5 – *PowerEnJoy*

You are to develop a digital management system for a car-sharing service that exclusively employs electric cars. First, the system should provide the functionality normally provided by car-sharing services. These include:

- Users must be able to register to the system by providing their credentials and payment information. They receive back a password that can be used to access the system.
- Registered users must be able to find the locations of available cars within a certain distance from their current location or from a specified address.
- Among the available cars in a certain geographical region, users must be able to reserve a single car for up to one hour before they pick it up.
- If a car is not picked-up within one hour from the reservation, the system tags the car as available again, and the reservation expires; the user pays a fee of 1 EUR.
- A user that reaches a reserved car must be able to tell the system she's nearby, so the system unlocks the car and the user may enter.
- As soon as the engine ignites, the system starts charging the user for a given amount of money per minute; the user is notified of the current charges through a screen on the car.
- The system stops charging the user as soon as the car is parked in a safe area and the user exits the car; at this point, the system locks the car automatically.
- The set of safe areas for parking cars is pre-defined by the management system.

In addition to the functionality above, the system should incentivize the virtuous behaviors of the users. Specifically:

a) If the system detects the user took at least two other passengers onto the car, the system applies a discount of 10% on the last ride.

b) If a car is left with no more than 50% of the battery empty, the system applies a discount of 20% on the last ride.

c) If a car is left at special parking areas where they can be recharged and the user takes care of plugging the car into the power grid, the system applies a discount of 30% on the last ride.

d) If a car is left at more than 3 KM from the nearest power grid station or with more than 80% of the battery empty, the system charges 30% more on the last ride to compensate for the cost required to re-charge the car on-site.

e) If the user enables the money saving option, he/she can input his/her final destination and the system provides information about the station where to leave the car to get a discount. This station is determined to ensure a uniform distribution of cars in the city and depends both on the destination of the user and on the availability of power plugs at the selected station.

Single person groups will focus, through all 4 assignments, on the main functionality plus points a and b from the list above.

Groups composed of two people will focus, through all 4 assignments, on the main functionality plus points a, b, c and d from the list above.

Groups composed of three people will focus, through all 4 assignments, on the main functionality plus all points from the list above.

6 The documents to be created

Each document you produce will include the following element:

- **A FRONT PAGE** that includes the project title, the version of the document, your names and the release date.
- **The TABLE OF CONTENT** that includes the headers of all the first three levels headings in your document with the corresponding page number. At the beginning of this document you find a table of content that you can use as an example. Since in this document there are no level three heading (e.g., 3.1.1), they are not part of the table of content as well.

The specific characteristics each document should have are described in the next subsections.

6.1 Assignment 1

The *Requirements analysis and specification document (RASD)* contains the description of the scenarios, the use cases that describe them, and the models describing requirements and specification for the problem under consideration. You are to use a suitable mix of natural language, UML, and Alloy. UML and Alloy MUST be part of the documentation. You must also show that you used the Alloy tool for analysis, by reporting the models you obtained by using it. Of course, the initial written problem statement we provide suffers from the typical drawbacks of natural language descriptions: it is informal, incomplete, uses different terms for the same concepts, and the like. You may choose to solve the incompleteness and ambiguity as you wish, but be careful to clearly document the choices you make and the corresponding rationale. You will also include in the document information on the number of hours each group member has worked towards the fulfillment of this deadline. As a reference structure for your document you can select the one suggested by IEEE or you can refer to one of the project examples published on the course website.
Please include in the document information about the effort spent by each group member for completing this document.

6.2 Assignment 2

The *Design document (DD)* must contain a functional description of the system, and any other view you find useful to provide. You should use all the UML diagrams you need to provide a full description of the system. Alloy may also be useful, although its use is not mandatory here. You will also include information on the number of hours each group member has worked towards the fulfillment of this deadline. As a reference structure for your document please refer to the following one:

1. **INTRODUCTION**
   - A. *Purpose*
   - B. *Scope*
   - C. *Definitions, Acronyms, Abbreviations*
   - D. *Reference Documents*
   - E. *Document Structure*
2. **ARCHITECTURAL DESIGN**

A. *Overview*: High level components and their interaction
C. *Component view*
D. *Deployment view*
E. *Runtime view*: You can use sequence diagrams to describe the way components interact to accomplish specific tasks typically related to your use cases
F. *Component interfaces*
G. *Selected architectural styles and patterns*: Please explain which styles/patterns you used, why, and how
H. *Other design deci*sions

3. **ALGORITHM DESIGN**: Focus on the definition of the most relevant algorithmic part
4. **USER INTERFACE DESIGN**: Provide an overview on how the user interface(s) of your system will look like; if you have included this part in the RASD, you can simply refer to what you have already done, possibly, providing here some extensions if applicable.
5. **REQUIREMENTS TRACEABILITY**: Explain how the requirements you have defined in the RASD map to the design elements that you have defined in this document.
6. **EFFORT SPENT:** In this section you will include information about the number of hours each group member has worked towards the fulfillment of this deadline.
7. **REFERENCES**

6.3 Assignment 3

The *Integration Test Plan Document (ITPD)* aims at describing how you plan to accomplish the integration test. This document is supposed to be written before the integration test really happens. Often it is written in parallel to the Design Document and takes the architectural description of the software system as a starting point. This document needs to explain to the development team what to test, in which sequence, which tools are needed for testing (if any), and which stubs/drivers/oracles need to be developed. The structure we suggest for this document is the following (if you introduce changes to this structure, please provide a justification for this):

1. Introduction

1.1 Revision History: Record all revisions to the document
1.2 Purpose and Scope: State the purpose and scope of the document
1.3 List of Definitions and Abbreviations
1.4 List of Reference Documents: List all reference documents, for instance:

- The project description
- The RASD
- The DD
- The documentation of any tool you plan to use for testing

2. Integration Strategy

2.1 Entry Criteria

Specify the criteria that must be met before integration testing of specific elements may begin (e.g., functions must have been unit tested).

2.2 Elements to be Integrated

Identify the components to be integrated, refer to your design document to identify such components in a way that is consistent with your design.

2.3 Integration Testing Strategy

Describe the integration testing approach (top-down, bottom-up, functional groupings, etc.) and the rationale for the choosing that approach.

2.4 Sequence of Component/Function Integration

NOTE: The structure of this section may vary depending on the integration strategy you select in Section 2.3; use the structure proposed below as a non mandatory guide

2.4.1 Software Integration Sequence. For each subsystem, identify the sequence in which the software components will be integrated within the subsystem; relate this sequence to any product features that are being built up.

2.4.2 Subsystem Integration Sequence. Identify the order in which subsystems will be integrated; if you have a single subsystem, 2.4.1 and 2.4.2 are to be merged in a single section.

You can refer to Section 2.2 of the test plan example [1] as an example.

3. Individual Steps and Test Description

For each step of the integration process above, describe the type of tests that will be used to verify that the elements integrated in this step perform as expected. Describe in general the expected results of the test set. You may refer to Chapter 3 and Chapter 4 of the test plan example [1] as an example of what we expect.

(NOTE: This is not a detailed description of test protocols. Think of this as the test design phase. Specific protocols will be written to fulfill the goals of the tests in this section.)

4. Tools and Test Equipment Required

Identify all tools and test equipment needed to accomplish the integration. Refer to the tools presented during the lectures. Explain why and how you are going to use them.

Note that you may also use manual testing for some part. Consider manual testing as one of the possible tools you have available.

5. Program Stubs and Test Data Required

Based on the testing strategy and test design, identify any program stubs or special test data required for each integration step.

6. Effort Spent

In this section you will include information about the number of hours each group member has worked towards the fulfillment of this deadline.

6.4 Assignment 4

The *Project Plan (PP)* aims at defining a planning for your project. Your project team will be composed of you and your colleagues in the group. Should you be alone, you can imagine to have one or two partners helping you during the project. The steps you should document in the PP are the following:

- Apply Function Points to estimate the size and then COCOMO to estimate effort and cost.

- Identify the tasks for your project and their schedule. Do so retrospectively, assuming that the project has started in October 16, 2016 as it really happened.
- Allocate the resources (all members of your group) to the various tasks. In defining the allocation, take into account your actual availability for the project.
- Define the risks for the project, their relevance and the associated recovery actions.

You will also include in the document information on the number of hours each group member has worked towards the fulfillment of this deadline.

7 Bibliography

[1] Integration Test Plan Example https://beep.metid.polimi.it/documents/3343933/5b3768d0-d949-4369-87e1-7a31b6943726