A.Y. 2018-2019 Software Engineering 2
Mandatory Project: goal, schedule, and rules

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

Table of contents

1   Goal and approach

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

The two assignments will be reviewed during the final discussions that will take place during the winter exam sessions according to a schedule that will be proposed in the forthcoming months. 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 mandatory project. The two assignments are described in the rest of this document.

2   Project schedule

- Group registration deadline 14/10/2018
- RASD submission deadline 11/11/2018
- DD submission deadline 10/12/2018
- Final presentation (to be scheduled)

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

3 Rules

- The project MUST 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, but if we will realize that there is an unbalance between the groups under the responsibility of each of us, we may change your reference professor. In this case, we will inform you a few days after the group registration deadline.
- 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.
- Each group MUST release artifacts by committing them into a specific folder of the github repository created for the project (see the following section).
- Each group MUST use the repository not only to upload the final versions of deliverables but also to commit intermediate versions. We want to see commits performed by all group members.
- The material included in your artifacts is not fixed in stone. You can (and are encouraged to) provide updates at any point before 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 and organization of your repository

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

1. Create a private repository for your project on Github (https://github.com). Your repository should be named by combining the names of all group members. For instance, BianchiRossiVerdi will be the name of the repository of the group composed of the students Tommaso Bianchi, Maria Rossi e Veronica Verdi. Make sure that all group members have a Github account and have access to the repository. Moreover, invite your reference professor (Github accounts dinitto and matteo-g-rossi) to access your repository (reading access is sufficient).
2. Register your group by filling in the following form https://goo.gl/forms/Vdl3iO1Zj2oDKs1g2. Do not forget to include in the form all relevant data!
3. Create a directory for each of the documents you will be working on.
4. Moreover, create a directory called DeliveryFolder where, by the due deadline, you will commit the pdf version of your document (name it RASD1.pdf or DD1.pdf, depending on the document you are releasing) plus any additional file you may want to include (e.g., the Alloy

model and/or any UML model).

5. After the deadline for submission, should you need to update your document, you can commit in the same folder another pdf file with an increased version number, e.g., RASD2.pdf

5 The problem: TrackMe

TrackMe is a company that wants to develop a software-based service allowing third parties to monitor the location and health status of individuals. This service is called **Data4Help**. The service supports the registration of individuals who, by registering, agree that TrackMe acquires their data (data acquisition can happen through smartwatches or similar devices). Also, it supports the registration of third parties. After registration, these third parties can request:

- Access to the data of some specific individuals (we can assume, for instance, that they know an individual by his/her social security number or fiscal code in Italy). In this case, TrackMe passes the request to the specific individuals who can accept or refuse it.

- Access to anonymized data of groups of individuals (for instance, all those living in a certain geographical area, all those of a specific age range, etc.). These requests are handled directly by TrackMe that approves them if it is able to properly anonymize the requested data. For instance, if the third party is asking for data about 10-year-old children living in a certain street in Milano and the number of these children is two, then the third party could be able to derive their identity simply having people monitoring the residents of the street between 8.00 and 9.00 when kids go to school. Then, to avoid this risk and the possibility of a misuse of data, TrackMe will not accept the request. For simplicity, we assume that TrackMe will accept any request for which the number of individuals whose data satisfy the request is higher than 1000.

As soon as a request for data is approved, TrackMe makes the previously saved data available to the third party. Also, it allows the third party to subscribe to new data and to receive them as soon as they are produced.

Imagine now that, after some time, TrackMe realizes that a good part of its third-party customers wants to use the data acquired through Data4Help to offer a personalized and non-intrusive SOS service to elderly people. Therefore, TrackMe decides to build a new service, called **AutomatedSOS**, on top of Data4Help. AutomatedSOS monitors the health status of the subscribed customers and, when such parameters are below certain thresholds, sends to the location of the customer an ambulance, guaranteeing a reaction time of less than 5 seconds from the time the parameters are below the threshold.

Finally, TrackMe realizes that another great source of revenues could be the development of a service to track athletes participating in a run. In this case, the service, called **Track4Run**, should allow organizers to define the path for the run, participants to enroll to the run, and spectators to see on a map the position of all runners during the run. Of course, also in this case, Track4Run will exploit the features offered by Data4Help.

You are required to explore the above problem and define the corresponding RASD and DD. In particular:
- **Single-person groups** are required to focus only on Data4Help.

- **Two-people groups** are required to focus both on Data4Help and on AutomatedSOS

- **Three-people groups** are required to consider all the three services described above.

6   The documents to be created

Each document you produce will include the following elements:

- **A FRONT PAGE** that includes the project title, the version of the document, your names and the release date.

- **The TABLE OF CONTENTS** 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 contents 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 contents as well.

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

6.1   Assignment 1 - RASD

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 should refer to the one reported below that is derived from the one suggested by IEEE.

Please include in the document information about the effort spent by each group member for completing this document.

1. **INTRODUCTION**
   A.   *Purpose*: here we include the goals of the project
   B.   *Scope*: here we include an analysis of the world and of the shared phenomena
   C.   *Definitions, Acronyms, Abbreviations*
   D.   *Revision history*
   E.   *Reference Documents*
   F.   *Document Structure*
2. **OVERALL DESCRIPTION**
   A.   *Product perspective*: here we include further details on the shared phenomena and a domain model (class diagrams and statecharts)
   B.   *Product functions*: here we include the most important requirements
   C.   *User characteristics*: here we include anything that is relevant to clarify their needs
   D.   *Assumptions, dependencies and constraints*: here we include domain assumptions

3. **SPECIFIC REQUIREMENTS:** Here we include more details on all aspects in Section 2 if they can be useful for the development team.
   - A. *External Interface Requirements*
     - A.1 *User Interfaces*
     - A.2 *Hardware Interfaces*
     - A.3 *Software Interfaces*
     - A.4 *Communication Interfaces*
   - B. *Functional Requirements*: Definition of use case diagrams, use cases and associated sequence/activity diagrams, and mapping on requirements
   - C. *Performance Requirements*
   - D. *Design Constraints*
     - D.1 *Standards compliance*
     - D.2 *Hardware limitations*
     - D.3 *Any other constraint*
   - E. *Software System Attributes*
     - E.1 *Reliability*
     - E.2 *Availability*
     - E.3 *Security*
     - E.4 *Maintainability*
     - E.5 *Portability*
4. **FORMAL ANALYSIS USING ALLOY:** in this section you will include your Alloy model. We require you to comment on the model by discussing the purpose of the model, what you can prove with it and why what you prove is important given the problem at hand. You are also required to show one or more worlds obtained by running your model.
5. **EFFORT SPENT:** In this section you will include information about the number of hours each group member has worked for this document.
6. **REFERENCES**

6.2 Assignment 2 - DD

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. *Revision history*
   - E. *Reference Documents*
   - F. *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

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.

7. **IMPLEMENTATION, INTEGRATION AND TEST PLAN**: Identify here the order in which you plan to implement the subcomponents of your system and the order in which you plan to integrate such subcomponents and test the integration.

8. **EFFORT SPENT:** In this section you will include information about the number of hours each group member has worked for this document.

9. **REFERENCES**