

AA 2015-2016 Software Engineering 2—Project goal, schedule and rules
READ THIS VERY CAREFULLY—NO EXCUSE FOR IGNORING WHAT WE WRITE HERE

Second project for AA 2015-2016: SmartCityAdvisor

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 different assignments (all assignments except the third one are provided in this document):

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

All assignments will be reviewed during the final presentations that will be scheduled before the end of the project.

The evaluation will aim at assessing the quality of the artifacts you will 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).

If you have questions and doubts on the project, please use the forum of the course for asking them.

2 Project schedule

The code inspection assignment will be provided by 29/05/2016. The deadlines for releasing your documents are every two weeks. This document contains all assignments except one, so within the given timeframe, you can schedule your work as you need, taking into account your obligations for the second semester exams. However, should you see any issue in the schedule, please let us know.

- Group registration deadline 29/04/2016
- RASD submission deadline 15/05/2016
- Design Document submission deadline 29/05/2016
- Inspection document submission deadline 12/06/2016
- Testing document submission deadline 26/06/2016
- Function points document submission deadline 10/07/2016
- 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 not suggested. The assignments will be calibrated based on the size of the group.
- Each group MUST register to the project following the steps indicated in Section 4.
- Each group MUST provide the requested artifacts within the stated deadlines. A delay of a few days will be tolerated but it will 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 on the web.
- The material presented in one artifact is not fixed in stone. You can provide updates after the discussion sessions in the project lab.
- 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 course 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>). Note that in previous projects we have been using GoogleCode, but, unfortunately, Google has decided to dismiss it. Make sure that you give a meaningful name and description for your project. 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://docs.google.com/forms/d/1PaYzfX4Ostqjq4bd3t3XOjl4XdrUM7wQpEnYpaf420/viewform>. Do not forget to include in the form all relevant data!

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

Assume that the city of Milano has installed in its territory a number of sensors to acquire information about the following elements:

- Level of CO₂ in the air.
- Cars that enter in the city center.
- Availability of parking places in all areas in the city center.

It has also installed proper actuators that control the traffic lights at the main intersections in the city. They can allow a manually operated control center or an automatic system to divert the traffic through alternative paths when this is needed.

Moreover, the city has established agreements with the main hospitals in the territory to know in real-time the length of the queues in their emergency rooms, and with ATM (the public transport company) to know in real-time the position and current schedule of all public transports in the city.

Having these data at its disposal, the city is asking your team to develop a system and a web/mobile app for citizens that offers at least the following features:

- If the level of CO₂ is too high or in case of any special situation managed by the control center (e.g., the arrival of some VIP, an accident, ...), limit the traffic that enters in the city center by diverting it through paths that avoid the center. This is done both by controlling the traffic lights and by alerting the citizens through the app and through some large displays that are installed at the main entrances of the city
- When a citizen signals through the app that he/she has to go to an emergency room, provide suggestions on which hospital to choose based on: i) the problem of the citizen and the specializations available in the hospitals, ii) the status of queues in the various emergency rooms, iii) the location of the citizen and iv) the situation of traffic.

Moreover, the city has opened a call for ideas on how to use the data it has available. So, your team is also asked to participate to this call and to identify at least another functionality for your system per each member of your team (for instance, if your team is composed of two members, you have to find at least two new functionalities). Of course, after having identified these functionalities, you are asked to define them in your RASD together with the others and to take them into account in all other relevant assignments of the project.

Hint: you can suppose that the interaction with sensors/actuators is delegated to a specific component that exploits existing platforms and protocols for this. You do not need to investigate such platforms/protocols, but you need to focus on designing the way this component is integrated with the rest of your system.

6 The documents to be created

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, etc. 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.

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. 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 please refer to the following one:

FRONT PAGE: Include at least the project title, the version of the document, your names and the release date

TABLE OF CONTENT: Include the table of content of your document

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 decisions*
3. **ALGORITHM DESIGN:** Focus on the definition of the most relevant algorithmic part of your project
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 into the design elements that you have defined in this document
6. **REFERENCES**

6.3 Assignment 4

The *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), 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):

FRONT PAGE and TABLE OF CONTENT as usual

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 Design document
 - 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/functions 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 of what we expect.

3. Individual Steps and Test Description

For each step of the integration process identified 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 identified 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.

6.4 Assignment 5

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 project 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 April 25 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>