A.Y. 2025-2026 Software Engineering 2
Requirement Engineering and Design 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 R&DD project. The two assignments are described in the rest of this document.

2 Project schedule

- Group registration deadline: 24/10/2025
- RASD submission deadline: 23/12/2025
- DD submission deadline: 07/01/2026
- Final presentation: to be scheduled

All deadlines are assumed to expire at **23:59** (Rome Time).
Note: You can submit before the deadlines, if you want/need.

3 Rules

- This assignment is optional and replaces part of the written exam of the Software Engineering 2 course.
- The project is 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 interested in taking the project must register itself following the steps listed in Section 4. “Mixed” groups involving students of the three sections are allowed. When registering, these groups must 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 among 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. These artifacts will be presented to the reference professor in a final meeting that will be scheduled later.
- Each group MUST release artifacts by committing and then pushing into the main branch of the git repository created for the project (see the following section).
- Each group MUST use the repository not only to push the final versions of deliverables, but also for intermediate versions. We want to see commits performed by all group members. In the case a group wants to use collaborative writing tools (e.g., Google Docs) that keep track of individual contributions, the group will include in the readme file associated with the git repository the link to the online document, making sure that through that link the reference professor will be able to inspect the contributions to the document.
- The material included in your artifacts is not fixed in stone. You can (and are encouraged to) provide updates at any point before the final submission deadline 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 Webeep website. We will answer as soon as possible.

4 Code of Conduct for the use of Generative AI tools

You are welcome to use any tools that help you prepare high-quality R&DD documents. However, please keep in mind the following key principles:

1. **Responsibility for content**
   You are solely responsible for the entire content of your documents, including all text, UML diagrams, figures, and references. While you may use any software or tool to support your writing and preparation, you must ensure that all material is correct, original, and fully understood by you. During the project discussion, you should be able to explain, justify, and defend every aspect of your work in detail.

2. **Use of Generative AI**
   Generative AI tools—including, but not limited to, Large Language Models (LLMs)—cannot replace your own reasoning or produce high-quality R&DD documents on their own. However, you may use them as support tools for specific, well-defined tasks—such as improving writing style and grammar, generating summaries, or assisting in the creation of UML diagrams.
   If you use Generative AI tools in your work, you must include a dedicated section in your document clearly describing:
   - Which tool(s) you used (e.g., specific chatbots, such as ChatGPT, Gemini);
   - The inputs you provided (e.g., prompts, datasets, source materials, parameters, or constraints);
   - The outputs you obtained;
   - How you verified, refined, and integrated those outputs into your project.

Important notes:

- Some tools—including free or online ones—may store input data for future model training. Use them cautiously and ensure that privacy and confidentiality are maintained.
- Generative AI tools may produce inaccurate or misleading ("hallucinated") outputs, including incorrect diagrams or figures. It is your responsibility to verify the accuracy and integrity of all generated material.

All tools must be used ethically and responsibly. Compliance with this code of conduct will be evaluated, and violations may affect your final grade, up to and including failure of the project in severe cases.

5 Group registration and organization of your repository

Students autonomously form groups and register by following these steps:

1. Create a private repository for your project on GitHub (https://github.com). Please note that, as students, you have the possibility to create a private GitHub repository for free. 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 (write permission). Moreover, invite your reference professor (GitHub accounts *matteocamilli* for Prof. Camilli, *dinitto* for Prof. Di Nitto, and *matteo-g-rossi* for Prof. Rossi) to access your repository (read permission is sufficient). Note: in past years we noticed that GitHub has set some restrictions concerning the creation of repositories from accounts of people from specific countries. If you are experiencing this problem, you can use Bitbucket instead (https://bitbucket.org/product/). We do not suggest this one for all groups because it has other limitations concerning the maximum number of private repositories you can have.
2. Register your group by filling in the following form https://forms.office.com/e/vJvqcdE55a. 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 deadlines, you will commit and push the pdf version of your documents (name it RASDv1.pdf or DDv1.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 submission deadline, should you need to update your document, you can commit in the same folder another pdf file with an increased version number, e.g., RASDv2.pdf. The new file should include a section that describes the performed changes.

6 The problem: Best Bike Paths

A biker's association has decided to develop *Best Bike Paths (BBP)*, a new software system supporting users in creating and browsing through an inventory of bike paths.
Registered users can use the application to record their trips and store them to keep track of their cycling activities. The system provides various statistics about each trip, such as the total distance covered, average speed, and other performance metrics. When available, the application can enrich trip data with meteorological information—such as weather conditions, temperature, and wind speed—retrieved from an external service.
Moreover, registered users can insert and make publishable information about bike paths, including their status (e.g., optimal, medium, sufficient, requires maintenance, …) and the presence of relevant obstacles, for instance, potholes. A bike path is intended to be one where a proper bike track exists or where cars are rare and speed limits are compatible with the average speed of a bike.
Insertion of information in the system can happen in two ways:

- *Manual mode*: users insert the data manually, specifying the name of the streets in the path and their status.
- *Automated mode*: users let BBP acquire data from their mobile devices while they bike. BBP should guess the user is biking given their speed; it should collect GPS information to reconstruct the followed path, and, at the same time, it should acquire data from the mobile device's accelerometer and gyroscope to keep track of any significant movement of the device itself that suggests the presence of potholes or other problems. Since there is the possibility of having false positives (e.g., non-existent potholes), the user will have to confirm or correct the information acquired by BBP before this is made available to the community.

As mentioned, information provided to BBP either manually or automatically can be made publishable by the owner.
Any user, either registered or not, can take advantage of the information collected by BBP. Users can specify an origin and a destination and ask the system to visualize on a map the bike path(s) between these two points. If multiple paths exist, BBP visualizes them based on their score. The path score is computed based on the status of the path and on how effective the path is to let the user reach the destination from the given origin.
If BBP collects publishable information about some paths from multiple users, it merges the different pieces of information considering their freshness and the number of confirming data obtained. For example, if a path is tagged as "optimal" by two users and as "requiring maintenance" by another and the three users introduce the information in the system in the same timeframe, BBP will consider the path as optimal, unless further information arrives.

7 Project Scope

The goal of the project is to define the RASD and DD for BBP. The task is different for groups of different sizes. More specifically:

- **Groups composed of a single student** will focus on the aspects of BPP **concerning**:
    - The recording of personal trips.
    - The insertion of publishable information in manual mode.
    - The visualization of the information about possible paths between an origin and a destination specified by the user.
- **Groups composed of two students** focus on the same aspects as groups of one student **plus**:
    - The insertion of publishable information in automated mode and its confirmation by the user.
- **Groups of three students** will focus on all aspects of BBP, so those tackled by groups of two students **plus**:
    - The merging of information incoming from different users (last paragraph in the description above).

8 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.
- **A TABLE OF CONTENTS** that includes the headers of the first three levels of 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 headings (e.g., 3.1.1), they are not part of the table of contents.

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

8.1 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 the requirements and specification for the problem under consideration. You are to use a suitable mix of natural language, UML, and Alloy. Any Alloy model should be validated through the tool, by reporting the models obtained by using it and/or by showing the results of assertion checks. 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 scenarios and further details on the shared phenomena and a domain model (class diagrams and state diagrams)
   - 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:** This section should include a brief presentation of the main objectives driving the formal modeling activity, as well as a description of the model itself, what can be proved with it, and why what is proved is important given the problem at hand. To show the soundness and correctness of the model, this section can show some worlds obtained by running it, and/or the results of the checks performed on meaningful assertions.
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**

8.2 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, but not mandatory. 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
   - B. *Component view*
   - C. *Deployment view*
   - D. *Runtime view*: You can use sequence diagrams to describe the way components interact to accomplish specific tasks typically related to your use cases
   - E. *Component interfaces*
   - F. *Selected architectural styles and patterns*: Please explain which styles/patterns you used, why, and how
   - G. *Other design deci*sions
3. **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.
4. **REQUIREMENTS TRACEABILITY**: Explain how the requirements you have defined in the RASD map to the design elements that you have defined in this document.
5. **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.
6. **EFFORT SPENT:** In this section you will include information about the number of hours each group member has worked for this document.
7. **REFERENCES**