Software Requirements Specification for Software Engineering: subtitle describing software

Team #6, Six Sense
Omar Alam
Sathurshan Arulmohan
Nirmal Chaudhari
Kalp Shah
Jay Sharma

September 25, 2025

Contents

Revision History

Date	Version	Notes
2025-10-06	1.0	Initial Write-up

1 Goal

1.1 G.1 Context and overall objective

[High-level view of the project: organizational context and reason for building a system. —SS]

Goal 1. This is a goal example. If you need explicit (and automatic) numbering, you can use the definitions in the `.tex` template.

Requirement 2. *This is a requirement example. It illustrates how numbering is continuous and cross-types (if this is what you need).*

1.2 G.2 Current situation

[Current state of processes to be addressed by the project and the resulting system. —SS]

Requirement 3. *This is a requirement example. It refines 1*

1.3 G.3 Expected benefits

[New processes, or improvement to existing processes, made possible by the projectâ€™s results. —SS]

1.4 G.4 Functionality overview

[Overview of the functions (behavior) of the system. Principal properties only (details are in the System book). —SS]

1.5 G.5 High-level usage scenarios

[Fundamental usage paths through the system. —SS]

1.6 G.6 Limitations and exclusions

[Aspects that the system need not address. —SS]

1.7 G.7 Stakeholders and requirements sources

[Groups of people who can affect the project or be affected by it, and other places to consider for information about the project and system. —SS]

2 Environment

2.1 E.1 Glossary

[Clear and precise definitions of all the vocabulary specific to the application domain, including technical terms, words from ordinary language used in a special meaning, and acronyms. This chapter should not be empty! —SS]

2.2 E.2 Components

[List of elements of the environment that may affect or be affected by the system and project. Includes other systems to which the system must be interfaced. —SS]

2.3 E.3 Constraints

[Obligations and limits imposed on the project and system by the environment. —SS]

2.4 E.4 Assumptions

[Properties of the environment that may be assumed, with the goal of facilitating the project and simplifying the system. —SS]

2.5 E.5 Effects

[Elements and properties of the environment that the system will affect. —SS]

2.6 E.6 Invariants

[Properties of the environment that the systemâ€™s operation must preserve. —SS]

3 System

3.1 S.1 Components

[Overall structure expressed by the list of major software and, if applicable, hardware parts. —SS]

3.2 S.2 Functionality

[One section, S.2.n, for each of the components identified in S.2, describing the corresponding behaviors (functional and non-functional properties). —SS]

3.3 S.3 Interfaces

[How the system makes the functionality of S.2 available to the rest of the world, particularly user interfaces and program interfaces (APIs). —SS]

3.4 S.4 Detailed usage scenarios

[Examples of interaction between the environment (or human users) and the system: use cases, user stories. —SS]

3.5 S.5 Prioritization

[Classification of the behaviors, interfaces and scenarios (S.2, S.3 and S.4) by their degree of criticality. —SS]

3.6 S.6 Verification and acceptance criteria

[Specification of the conditions under which an implementation will be deemed satisfactory. —SS]

4 Project

4.1 P.1 Roles and personnel

[Main responsibilities in the project; required project staff and their needed qualifications. —SS]

4.2 P.2 Imposed technical choices

[Any a priori choices binding the project to specific tools, hardware, languages or other technical parameters. —SS]

4.3 P.3 Schedule and milestones

[List of tasks to be carried out and their scheduling. —SS]

4.4 P.4 Tasks and deliverables

[Details of individual tasks listed under P.3 and their expected outcomes. —SS]

4.5 P.5 Required technology elements

[External systems, hardware and software, expected to be necessary for building the system. —SS]

4.6 P.6 Risks and mitigation analysis

[Potential obstacles to meeting the schedule of P.4, and measures for adapting the plan if they do arise. —SS]

4.7 P.7 Requirements process and report

[Initially, description of what the requirements process will be; later, report on its steps. —SS]

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)?

4. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

5. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

6. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?