SOFTWARE REQUIREMENTS SPECIFICATION

Room8

Version 1.0.0

Prepared by:
Mohammed Abed
Maged Armanios
Jinal Kasturiarachchi
Jane Klavir
Harshil Patel

McMaster University

October 11, 2024

Revision History

Name	Date	Reason For Changes	Version
J.-M. Bruel	2021-01-22	First Draft	1.0
J.-M. Bruel	2023-01-28	Check after publication of the Handbook	1.23
J.-M. Bruel	2023-06-12	Add reqs automated numbering	1.23.1
J.-M. Bruel	2023-08-25	Add Minimum Requirements Outcome Principle	1.23.8
J.-M. Bruel	2023-12-22	Remove section numbers	1.23.12
J.-M. Bruel	2024-08-01	Add warning about non empty chapters	1.0.0

This document follows the requirements documentation structure presented in the Handbook of requirements and business analysis, by Bertrand Meyer.

Contents

Goals

Contents

Comment: *Goals are "needs of the target organization, which the system will address". While the development team is the principal user of the other books, the Goals book addresses a wider audience: essentially, all stakeholders.*

G.1 Context and overall objective

Comment: *High-level view of the project: organizational context and reason for building a system.*
Comment: *This chapter should not be empty!*

Goal 1. This is a goal example. If you need explicit (and automatic) numbering, you can use the definitions in the `.tex` template. Is is refined by 3

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

G.2 Current situation

Comment: *Current state of processes to be addressed by the project and the resulting system.*

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

G.3 Expected benefits

Comment: *New processes, or improvement to existing processes, made possible by the projectâ€™s results.* Comment: *This chapter should not be empty!*

G.4 Functionality overview

Comment: *Overview of the functions (behavior) of the system. Principal properties only (details are in the System book).*

G.5 High-level usage scenarios

Comment: *Fundamental usage paths through the system.*

G.6 Limitations and exclusions

Comment: *Aspects that the system need not address.*

G.7 Stakeholders and requirements sources

Comment: *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.* Comment: *This chapter should not be empty!*

Environment

Contents

Comment: *The Environment book describes the application domain and external context, physical or virtual (or a mix), in which the system will operate.*

E.1 Glossary

Comment: *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!*

E.2 Components

Comment: *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.*

E.3 Constraints

Comment: *Obligations and limits imposed on the project and system by the environment.* Comment: *This chapter should not be empty!*

E.4 Assumptions

Comment: *Properties of the environment that may be assumed, with the goal of facilitating the project and simplifying the system.*

E.5 Effects

Elements and properties of the environment that the system will affect.

E.6 Invariants

Properties of the environment that the system's operation must preserve.

System

Contents

Comment: *The System book refines the Goal one by focusing on more detailed requirements about the system under development, mainly its constituents, behaviors and properties.*

S.1 Components

Comment: *Overall structure expressed by the list of major software and, if applicable, hardware parts.*
Comment: *This chapter should not be empty!*

S.2 Functionality

Comment: *One section, S.2.n, for each of the components identified in S.2, describing the corresponding behaviors (functional and non-functional properties).* Comment: *This chapter should not be empty!*

S.3 Interfaces

Comment: *How the system makes the functionality of S.2 available to the rest of the world, particularly user interfaces and program interfaces (APIs).*

S.4 Detailed usage scenarios

Comment: *Examples of interaction between the environment (or human users) and the system: use cases, user stories.*

S.5 Prioritization

Comment: *Classification of the behaviors, interfaces and scenarios (S.2, S.3 and S.4) by their degree of criticality.*

S.6 Verification and acceptance criteria

Comment: *Specification of the conditions under which an implementation will be deemed satisfactory.*

Project

Contents

Comment: *The Project book describes all the constraints and expectations not about the system itself, but about how to develop and produce it.*

P.1 Roles and personnel

Comment: *Main responsibilities in the project; required project staff and their needed qualifications.*

P.2 Imposed technical choices

Comment: *Any a priori choices binding the project to specific tools, hardware, languages or other technical parameters.*

P.3 Schedule and milestones

Comment: *List of tasks to be carried out and their scheduling.* Comment: *This chapter should not be empty!*

P.4 Tasks and deliverables

Comment: *Details of individual tasks listed under P.3 and their expected outcomes.* Comment: *This chapter should not be empty!*

P.5 Required technology elements

Comment: *External systems, hardware and software, expected to be necessary for building the system.*

P.6 Risks and mitigation analysis

Comment: *Potential obstacles to meeting the schedule of P.4, and measures for adapting the plan if they do arise.*

P.7 Requirements process and report

Comment: *Initially, description of what the requirements process will be; later, report on its steps.*

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning.

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?