Our Awesome Project

Requirements Standard Plan

Author 1, Author 2, Author 3

Version 1, 2025-10-04

Table of Contents

Academic Integrity Disclaimer

We would like to acknowledge that as a dedicated students of McMaster University, we have thoroughly read and comprehended the Academic Integrity Policy published by the university. We are committed to upholding the principles of academic honesty and integrity in all aspects of our educational journey. We understand the importance of acknowledging the work and ideas of others, and we pledge to ensure that all our academic endeavors are conducted with the utmost originality and compliance with the university's policy.

We affirm that the content presented in this document is entirely our own, and any external sources used have been appropriately cited and referenced.

Author 1

As I submit my work, I, **Author 1**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Author 2

As I submit my work, I, **Author 2**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Author 3

As I submit my work, I, **Author 3**, take full responsibility for the integrity of my work and promise to avoid any form of plagiarism, cheating, or dishonest behavior. This acknowledgment serves as a testament to my dedication to academic excellence and the fostering of a trustworthy academic community at McMaster University.

Control Information

Version	Delivery		Feedback	
	Deadline	*Delivered*	*Received*	*Integrated*
V1				
V2				
V3				

Author 1

Here is a quick biography of **Author 1**. You can contact them at **a1@mcmaster.ca**

Author 2

Here is a quick biography of **Author 2**. You can contact them at **a2@mcmaster.ca**

Author 3

Here is a quick biography of **Author 3**. You can contact them at **a3@mcmaster.ca**

Glossary

Vim

Vim is a highly configurable, keyboard-driven text editor built to make creating and changing any kind of text very efficient.[3]

(G) Goals

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 [1]

It must contain enough information to provide — if read just by itself — a general sketch of the entire project. To this effect, chapter G.3 presents a short overview of the system and G.1 will typically include some key properties of the environment. As it addresses a wide readership, it should be clear and minimize the use of specialized technical terms. Together, G.1, G.2 and G.3 describe the rationale for the project. It is important to state these justifications explicitly. Typically, they are well understood at the start of the project, but management and priorities can change [1]

Control Information

Table 1. Our Awesome Project — Versionning Information — Goal Book

Section	Version	Lead	Delivered on	Reviewer	Approved on
G.1					
G.2					
G.3					
G.4					
G.5					
G.6					
G.7					

(G.1) Context and Overall Objectives

High-level view of the project: organizational context and reason for building a system. It explains why the project is needed, recalls the business context, and presents the general business objectives. [1]

Nothing available at this point.

(G.2) Current situation

Current state of processes to be addressed by the project and the resulting system. It describes the current situation, upon which the system is expected to improve [1]

Our Awesome Project arises from the fact that students, especially in technical disciplines such as engineering, often need to create notes containing both structured text and quick sketches of diagrams like state machines, circuits, or tables. Often times these notes also need to be created in a timely manner, as lectures can move quickly.

There are many existing options for taking notes, each with their own trade-offs:

- Pen and paper
  - Not easily editable or shareable
  - Handwriting may be slower than typing
- Tablets with stylus support (e.g. iPad with Apple Pencil)
  - Can be expensive
  - Handwriting may be slower than typing
- Traditional text editors (e.g. Microsoft Word, Google Docs)
  - Limited support for creating diagrams within the app
- Mouse-driven editors (e.g. Draw.io, OneNote)
  - Can be slow to navigate menus for specific geometry
  - Limited keyboard shortcut support for creating and editing geometry
  - May require exporting and importing images into a separate text editor for note-taking (e.g. Draw.io)
- Text-based diagram languages (e.g. Mermaid, PlantUML)
  - Separate definition from rendering, forcing users to manage an inefficient write-compile-insert cycle
  - Requires integration with another text editor for note-taking text
- Document markup languages (e.g. LaTeX, Typst)
  - Limited to a more traditional page format rather than infinite workspace like OneNote for example
  - Editing underlying text rather than geometry itself

On top of inefficiencies to workflow, an important note is that none of these solutions are fully keyboard-driven. This is a key aspect of the Vim ideology that Our Awesome Project aims to embody by providing a unified system for creation and editing of notes involving both text and diagrams exclusively through the keyboard.

(G.3) Expected Benefits

> *New processes, or improvement to existing processes, made possible by the project's results. It presents the business benefits expected from the successful execution of the project.* ***This chapter is the core of the Goals book****, describing what the organization expects from the system. It ensures that the project remains focused: if at some stage it gets pushed in different directions, with "creeping featurism" threatening its integrity, a reminder about the original business goals stated in those chapters will help.* [1]

There are two main expected benefits that will be delivered by Our Awesome Project.

The first is that the system will provide a new interface for creating multimedia (i.e. text and diagrams) notes using only a keyboard. This will include both the creation of note files from the user interface perspective and the content in a note. This benefits the user as it provides a keyboard only interface for creating and managing multimedia notes, which does not currently exist for this type of system.

By extension of the first, the second expected benefit is to provide the user a potential boost to their note taking speed due to the keyboard-driven nature of the system and the support for shortcuts. This assumes that at full proficiency, a user can expect to be quicker taking simple notes with Our Awesome Project than with other existing solutions outlined in G.1. For our purposes, proficiency can be defined as completion of the user manual and 10+ hours of usage. An example of a simple note can be described as a finite state machine with 5 states, at least one arrow flowing in and out of each state, labels for states and state transitions, and some corresponding description paragraph in an arbitrary location around the state machine.

(G.4) Functionality overview

Overview of the functions (behavior) of the system. Principal properties only (details are in the System book). It is a short overview of the functions of the future system, a kind of capsule version of book S, skipping details but enabling readers to get a quick grasp of what the system will do. [1]

🔄 Nothing available at this point.

(G.5) High-level usage scenarios

*Fundamental usage paths through the system. It presents the main scenarios (use cases) that the system should cover. The scenarios chosen for appearing here, in the Goals book, should only be the **main usage patterns**, without details such as special and erroneous cases; they should be stated in user terms only, independently of the system's structure. Detailed usage scenarios, taking into account system details and special cases, will appear in the System book (S.4).* [1]

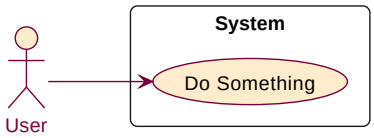

Figure 1. High Level use cases diagram

🔄 Nothing available at this point.

(G.6) Limitations and Exclusions

Aspects that the system need not address. It states what the system will not do. This chapter addresses a key quality attribute of good requirements: the requirements must be delimited (or "scoped"). G.6 is not, however, the place for an analysis of risks and

> *obstacles, which pertain to the project rather than the goals and correspondingly appears in chapter P.6.* [1]

🔄 Nothing available at this point.

(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. It lists stakeholders and other requirements sources. It should define stakeholders as categories of people, not individuals, even if such individuals are known at the time of writing. The main goal of chapter G.7 is to avoid forgetting any category of people whose input is relevant to the project. It also lists documents and other information that the project, aside from soliciting input from stakeholders, can consult for requirements information.* [1]

🔄 Nothing available at this point.

(E) Environment

The Environment book describes the application domain and external context, physical or virtual (or a mix), in which the system will operate. [1]

Control Information

Table 2. Our Awesome Project — Versionning Information — Environment Book

Section	Version	Lead	Delivered	Reviewer	Approved
E.1					
E.2					
E.3					
E.4					
E.5					
E.6					

(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. It introduces the terminology of the project; not just of the environment in the strict sense, but of all its parts. [1]

🔄 Nothing available at this point.

(E.2) Components

List of elements of the environment that may affect or be affected by the system and project. It includes other systems to which the system must be interfaced. These components may include existing systems, particularly software systems, with which the system will interact — by using their APIs (program interfaces), or by providing APIs to them, or both. These are interfaces provided to the system from the outside world. They are distinct from both: interfaces provided by the system to the outside world (S.3); and technology elements that the system's development will require (P.5). [1]

🔄 Nothing available at this point.

(E.3) Constraints

Obligations and limits imposed on the project and system by the environment. This chapter defines non-negotiable restrictions coming from the environment (business rules, physical laws, engineering decisions), which the development will have to take into account. [1]

Nothing available at this point.

(E.4) Assumptions

Properties of the environment that may be assumed, with the goal of facilitating the project and simplifying the system. It defines properties that are not imposed by the environment (like those in E.3) but assumed to hold, as an explicit decision meant to facilitate the system's construction. [1]

Nothing available at this point.

(E.5) Effects

Elements and properties of the environment that the system will affect. It defines effects of the system's operations on properties of the environment. Where the previous two categories (E.3, E.4) defined influences of the environment on the system, effects are influences in the reverse direction. [1]

Nothing available at this point.

(E.6) Invariants

Properties of the environment that the system's operation must preserve, i.e., properties of the environment that operations of the system may assume to hold when they start, and must maintain. [1]

Nothing available at this point.

(S) System

the System book refines the Goal one by focusing on more detailed requirements.

Control Information

Section	Version	Lead	Delivered	Reviewer	Approved
S.1					
S.2					
S.3					
S.4					
S.5					
S.6					

(S.1) Components

Overall structure expressed by the list of major software and, if applicable, hardware parts. [1]

Nothing available at this point.

(S.2) Functionality

***This is the bulk of the System book, describing elements of functionality (behaviors)**. This chapter corresponds to the traditional view of requirements as defining "**what the system does**". It is organized as one section, S.2.n, for each of the components identified in S.1, describing the corresponding behaviors (functional and non-functional properties).* [1]

Nothing available at this point.

(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). It specifies how that functionality will be made available to the rest of the world, including people (users) and other systems. These are interfaces provided by the system to the outside; the other way around, interfaces from other systems, which the system may use, are specified in E.2. [1]

Nothing available at this point.

(S.4) Detailed usage scenarios

Examples of interaction between the environment (or human users) and the system, expressed as user stories. Such scenarios are not by themselves a substitute for precise descriptions of functionality (S.3), but provide an important complement by specifying cases that these behavior descriptions must support; they also serve as a basis for developing test cases. The scenarios most relevant for stakeholders are given in chapter G.5 in the Goals book, at a general level, as use cases; in contrast, S.4 can refer to system components and functionality (from other chapters of the System book) as well as special and erroneous cases, and introduce more specific scenarios. [1]

Nothing available at this point.

(S.5) Prioritization

Classification of the behaviors, interfaces and scenarios (S.2, S.3 and S.4) by their degree of criticality. It is useful in particular if during the course of the project various pressures force the team to drop certain functions. [1]

Nothing available at this point.

(S.6) Verification and acceptance criteria

Specification of the conditions under which an implementation will be deemed satisfactory. Here, "verification" as shorthand for what is more explicitly called "Verification & Validation" (V&V), covering several levels of testing — module testing, integration testing, system testing, user acceptance testing — as well as other techniques such as static analysis and, when applicable, program proving. [1]

Nothing available at this point.

(P) Project

Control Information

Section	Version	Lead	Delivered	Reviewer	Approved
P.1					
P.2					
P.3					
P.4					
P.5					
P.6					
P.7					

(P.1) Roles and personnel

Main responsibilities in the project; required project staff and their needed qualifications. It defines the roles (as a human responsibility) involved in the project. [1]

Nothing available at this point.

(P.2) Imposed technical choices

Any a priori choices binding the project to specific tools, hardware, languages or other technical parameters. Not all technical choices in projects derive from a pure technical analysis; some result from company policies. While some project members may dislike non-strictly-technical decisions, they are a fact of project life and must be documented, in particular for the benefit of one of the quality factors for requirements: "requirements must be justified". [1]

Nothing available at this point.

(P.3) Schedule and milestones

List of tasks to be carried out and their scheduling. It defines the project's key dates. [1]

Nothing available at this point.

(P.4) Tasks and deliverables

***This is the core of the Project book**. It details the individual tasks listed under P.3 and their expected outcomes. It define the project's main activities and the results they must produce, associated with the milestone dates defined in P.3.* [1]

🔄 Nothing available at this point.

(P.5) Required technology elements

External systems, hardware and software, expected to be necessary for building the system. It lists external technology elements, such as program libraries and hardware devices, that the project is expected to require. Although the actual use of such products belongs to design and implementation rather than requirements, it is part of the requirements task to identify elements whose availability is critical to the success of the project — an important element of risk analysis (P.6). [1]

🔄 Nothing available at this point.

(P.6) Risk and mitigation analysis

Potential obstacles to meeting the schedule of P.4, and measures for adapting the plan if they do arise. It is essential to be on the lookout for events that could derail the project, and devise mitigation strategies. It can include a SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) for the project. [1]

🔄 Nothing available at this point.

(P.7) Requirements process and report

Initially, description of what the requirements process will be; later, report on its steps. It starts out as a plan for conducting the requirements elicitation process, but is meant to be updated as part of that process so that it includes the key lessons of elicitation. [1]

🔄 Nothing available at this point.

References

- [1] Bertrand Meyer. *Handbook of Requirements and Business Analysis*. Springer. 2022.
- [2] Ian Sommerville and Peter Sawyer. *Requirements Engineering: A good Practice Guide*. Wiley. 1997.
- [3] Vim - the ubiquitous text editor. 2025. https://www.vim.org/