

This section of the Capstone Project Proposal sets an initial high-level cost analysis and project schedule.

It will be required that you create:

1. A spreadsheet of costs related to the scope of the project, with all necessary material and elements required to accomplish it effectively, and the allocated resources. **Note:** If the project being designed will not require any cost calculations, please state that here.
2. A project schedule/timeline (or Gantt chart) with dates for the completion of key components of the project after all project tasks have been defined and prioritized. The timeline should also include Proof of Concept (POC) subprojects that need to be planned to reduce risks within the project. POCs are small tests prior to the project to evaluate processes and roles. POCs do not produce deliverables, but help to identify gaps in processes or unforeseen risks. **Note:** For these subprojects, plan for 1–2 weeks in which to define one problem or issue to resolve, identify who will be completing the project, and describe the measures that will be used to calculate project success.
3. A programming schedule by implementing work breakdown and task time estimates. The programming schedule will be completed in your selected planning tool. Either provide the link so the instructor can access this or provide a screenshot within the proposal. Refer to the instructor for directions on how they would like this submitted.

References

This section of the Capstone Project Proposal includes attribution for any content used by someone else. The reference section should list all sources that were quoted, paraphrased, adapted, modified, etc.

Copyright Compliance

This section of the Capstone Project Proposal ensures copyright compliance demands have been met. **Note: Only a small portion of your project may rely on external code. If working on an open source project, this may be taken into consideration by the instructor.**

Ensure the following:

1. For each external technical tool or code used, provide a reference to its copyright policy, clearly showing your right to use it.
2. For each external technical tool or code used, detail how you used it, how you adapted it, how you modified it (if permitted), and why you used it as opposed to writing your own.
3. When code libraries/packages are used, explain why this was necessary, required, or recommended.

Note: Seek instructor approval for using external resources prior to beginning work on the project.

Milestone 2: Requirements Analysis

Objective: The goal of the requirements analysis is to establish the functionality, requirements, and design of the project. The requirements analysis will contain report definitions and layouts,

user interface (UI)/wireframes designs, a sitemap (if applicable), data element definitions, workflow diagrams, flowcharts, performance metrics, unique features, and security matrices.

Deliverable: Submission of the following in the learning management system:

1. Capstone Project Requirements Specification Document
2. Capstone User Stories
3. Any additional documentation requested

Refer to the "MSSE-MSSD Capstone Project Requirements Specification Template" as well as the "MSSE-MSSD Capstone User Story Template," located on the College of Science, Engineering and Technology page in the Student Success Center.

This milestone will be completed in SWE-540. Once you finish the milestone and submit it to the instructor, you may start working on Milestone 3.

Refer to the information below for additional details on each section of the Projects Requirements Specification.

Special Note: Depending on the project, portions of this milestone may not be applicable. In cases like these, work with the instructor to determine the components of the project that may replace this step. The instructor will determine what documentation is needed and how this documentation will be assessed.

General Information

The Capstone Project Requirements Specification Document will include the following:

1. Cover Page: Project name, author, program (in this case Software Engineering), project organization (GCU), instructor name, document revision number (use the following format v1.0, v2.0), and the date the project proposal form is submitted.
2. Abstract
3. History and Sign-off Sheet
4. Table of Contents
5. Functional Requirements
6. Non-Functional Requirements
7. Technical Requirements
8. Logical System Design
9. User Interface Design
10. Reports Design

Abstract

This section of the Capstone Project Requirements Specification Document is a summary of the entire project. It should be about two to three paragraphs, about 15–20 lines long. It should include the purpose of the project, the tasks involved in completing the project, and what will be accomplished. Details of technology and methodology should be avoided. Additional items to avoid are abbreviations, jargon, and language shortcuts (such as textspeak); repetition of words;

and referencing a table, figure, or any part of the main document, as well as putting in any references. The ideas should be presented in a manner accessible to readers not familiar with the domain.

Note: The abstract will need to be updated when you have completed the project in SWE-590. The updated version should provide a broader context of the major sections of the project (design, development, implementation, testing, and overall functionality).

History and Sign-off Sheet

This section of the Capstone Project Requirements Specification Document is to be used to record decisions made for the project. It documents communication, feedback, and implementation of changes.

Table of Contents

The table of contents should list all the headings (not subheadings) of the Capstone Project Requirements Specification Document, including page numbers. Do not remove any main headings from the "MSSE-MSSD Capstone Project Requirements Specification Template." If necessary, you may revise or add subsections as needed, depending on the topic of your project.

Functional Requirements

In your documentation of requirements, be mindful of the two categories: functional and nonfunctional requirements. Functional requirements refer to certain actions, UI elements, or similar action-driven components that must be implemented. Non-functional requirements refer to constraints imposed on performance metrics or resource utilization. As you gather and present these requirements, specify the nature of each. This will also impact your approach to design (in Milestone 3), implementation (in Milestone 4), and prioritization of development resources.

This section of the Capstone Project Requirements Specification Document describes the sequence of functional actions a project performs by providing a link to the User Stories.

A User Story describes a sequence of actions a project performs that provide an observable result of a value to a particular actor. The User Stories will be formatted using standard Agile Scrum format.

Example

As a [USER/ROLE], I want [FEATURE] so that [RATIONAL FOR FEATURE].

The User Stories should completely and clearly define all of the features, functionalities, and capabilities of the system or application that is being defined.

NOTE: Once the functional requirements have been completed, there may be situations where User Stories may need to be taken out of scope, possibly due to technical or timeline challenges. Any User Stories that are taken out of scope once the project development has started must be approved by the instructor with justification as to why the functionality is being removed from the project.

The following table must be updated and approved by the instructor if any User Stories are taken out of scope.

Table <Insert number>. Functional Requirements - User Stories Taken Out of Scope

User Story ID	User Story	Approval Date	Justification

Non-Functional Requirements

This section of the Capstone Project Requirements Specification Document describes the sequence of non-functional actions a project performs by providing a link to the User Stories.

A user stories should also be defined for non-functional requirements (NFRs). A list of common NFR's can be found on Wikipedia when searching for Non-Functional Requirements. This list should be reviewed with the appropriate NFRs applicable to the project defined as use cases or as a narrative using user stories. If necessary, the use cases can be documented in an external Excel spreadsheet with a reference to the document provided in the Capstone Project Requirements Specification Document.

NOTE: Once the non-functional requirements have been completed, there may be situations where User Stories may need to be taken out of scope, possibly due to technical or timeline challenges. Any User Stories that are taken out of scope once the project development has started must be approved by the instructor with justification as to why the functionality is being removed from the project.

The following table must be updated and approved by the instructor if any User Stories are taken out of scope.

Table <Insert number>. Non-Functional Requirements - User Stories Taken Out of Scope

User Story ID	User Story	Approval Date	Justification

Technical Requirements

This section of the Capstone Project Requirements Specification Document describes the tools, technologies, and their respective versions used in the project. List these requirements and include brief descriptions. Follow the listing of technical requirements with a listing and explanation of the software and/or hardware that is necessary to meet the requirements.

NOTE: Once the technical requirements have been completed there may be situations where technologies or tools may need to be taken out of scope or changed, possibly due to technical or timeline challenges. Any technologies or tools that are taken out of scope or changed once the project development has started must be approved by the instructor with justification as to why the functionality is being removed from the project.

The following table must be updated and approved by the instructor if any technologies or tools are taken out of scope or changed.

Table <Insert number>. Technical Requirements Taken Out of Scope

Technology or Tool	Approval Date	Justification

Logical System Design

This section of the Capstone Project Requirements Specification Document diagrams the logical architecture of the system. **Keep in mind, the purpose of the detailed logical architecture is to provide sufficient information for a developer to produce the system.**

The logical architectural diagram should reflect the functional requirements of the application and illustrate the flow of information through the system. The logical design should be abstract; it should not include any implementation details. The purpose of the logical model is to map data from its source to its appropriate destination, following the flow of the decision-making process in the system. The logical model describes how the system works in terms of function and logical information.

User Interface Design

This section of the Capstone Project Requirements Specification Document presents the sitemap and user interface design diagram for each user interface screen in the application.

Wireframe diagrams should include components such as, but are not limited to:

1. Title
2. Toolbars
3. Date entry fields
4. Text fields
5. Content, such as text, charts, tables, etc.
6. Application/webpages look and feel (one or the other)

Work with the instructor to define the components of the project, such as screen definitions and layouts. The project may be software development, an implementation of a concept, a mobile application, or something else. Flowcharts and technical diagrams are essential components in all projects.

Projects Requirements Review

Prior to submitting the milestone deliverable, review the prior milestone and ensure consistency throughout. The project may have evolved since the first proposal; therefore, some revisions may be required to maintain coherence and stay true to the original proposal. At this point, you are committing to the foundation of the project. While changes are possible (and likely) in future milestones, a good, well-planned, and well-presented foundation will minimize the need for future changes.

Milestone 3: Design

Objective: The purpose of the design phase is to transform the requirements into complete and detailed technical requirements/design specifications. The design phase will contain object-oriented analysis, classical design patterns, and UML notation such as UML structural diagrams (component, class, and deployment diagrams) and UML behavioral diagrams (case, sequence, and activity diagrams). It will also include report definitions and layouts, screen definitions and layouts, data element definitions, workflow diagrams, and security matrices. Once the design is approved, development begins.

Deliverable: Submission of the following in the learning management system: