

Capstone Project Handbook: Masters of Science in Computer Science or Data Science

Contents

[Purpose of the Capstone Project 2](#_bookmark0)

[Project Advisor 2](#_bookmark1)

[Technology Requirements 2](#_bookmark2)

[Project Timeline 3](#_bookmark3)

[Selecting a Project 3](#_bookmark4)

[Milestone 1: Project Proposal and Requirements Analysis 4](#_bookmark5)

[Milestone 2: Architectural Design / Model Pipeline Design 14](#_bookmark6)

[Milestone 3: Implementation 17](#_bookmark7)

[Milestone 4: Performance Analysis and Presentation 18](#_bookmark8)

[Academic Integrity 20](#_bookmark9)

[Plagiarism 21](#_bookmark10)

[Sources of Information and Use of Reference Sources 21](#_bookmark11)

[Intellectual Property (IP) 21](#_bookmark12)

[References 21](#_bookmark13)

[Appendix 22](#_bookmark14)

[Capstone Project Proposal 22](#_bookmark15)

[Requirements Analysis 25](#_bookmark16)

[Final Architectural Plan or Model Pipeline Design 26](#_bookmark17)

[Test Plan 28](#_bookmark18)

# Purpose of the Capstone Project

The vision for the capstone experience is to develop a project that can be used to demonstrate individual knowledge, skills, and abilities within a specified field, as well as communicate to both technical and nontechnical audiences. The capstone project will be a tangible artifact students will refer to in a variety of professional settings.

The capstone project consists of four milestones, following the approach:

Propose

Design

Build

Conclude

and Present

# Project Advisor

In each course in which a capstone milestone is due, the instructor in that course serves as the advisor and mentor for that milestone. The student will discuss the particulars of the milestone with the instructor, as indicated in each course. As students progress from course to course, they will be exposed to different views and receive feedback from different instructors. However, even if an instructor may hold a different opinion regarding a previously completed milestone, the student will never be required to revise the previous milestone.

Students may seek input from other faculty regarding various aspects of the capstone, especially if they are a member of one of the research groups in the department.

Every capstone is unique, varying in scope, resources, and complexity. It is possible that some elements in this document, such as forms and required deliverables, may need to be modified and/or adapted to suit the specific characteristics of a student's capstone. Therefore, for each course in which a milestone is due, students should consult their instructor for guidance regarding design, implementation, and documentation.

An important aspect of the capstone is that it should be a fully functional application that any authorized party can download, install, and run. It is not expected to be the most efficient or elegant, but it must be complete—user interface, data processing, results output, visuals (as applicable), computations, etc.

# Technology Requirements

Each student will independently determine the software and hardware needed for the project, with guidance by faculty. Use of free (open source) software is encouraged. Usual tools such as MS-Office will be used for creating documentation and presentation materials.

Deliverables will be uploaded on a web-based software repository, such as GitHub, where all aspiring and experienced programmers post their accomplishments. An application may possibly be uploaded to an app store, depending upon the nature of the project. For example, students would need access to the Apple Store or Google Play if the project were for a smartphone app.

Capstone projects may be marketable products that promote students' skills and abilities, as well as broaden their career options. Students should consult with faculty to ensure that all created work is published in compliance with GCU regulations, while recognizing the student’s ownership of the intellectual property rights.

# Project Timeline

The project timeline will vary based on the individual nature of the capstone project. Students should initiate discussions with faculty prior to taking CST-560, Research Methods in Computational Sciences. The four milestones will be completed during the last four courses in the program: CST-560, CST/DSC-570, CST/DSC-580, and CST/DSC-590.

|  |  |  |
| --- | --- | --- |
| **Milestone** | **Due** | **Deliverable** |
| **1** | In CST-560 | Proposal Requirements Analysis |
| **2** | In CST-570 or DSC-570 | Architectural Design (CST) ***or***  Model Pipeline Design (DSC) |
| **3** | In CST-580 or DSC-580 | Implementation |
| **4** | In CST-590 or DSC-590 | Performance Analysis Presentation |

# Selecting a Project

### Defining the Project

Clearly state the purpose of the project. Is the project intended to solve a particular problem? Is the project a game? Is it a simulation? Does the project explore a particular concept or technology? Is the project part of a larger solution for a workplace application?

If the project is intended to solve a problem, explain why the project is needed. What is the current process for solving the identified problem, and how will the project improve this process? Is the project's final deliverable a complete product or the implementation of an (original?) algorithm?

### Computer Science Examples

* Create an AI-based tool for efficient allocation of resources in an organization.
* Create a simulation of urban growth over time.

### Data Science Examples

* Create a market behavior predictive model.
* Create a natural language processor for summarizing key ideas in a lecture.

*Key Items to Consider*

* A student already employed may use the capstone project as an opportunity to work on a project for their work. Students must still complete every milestone stated in this handbook.
* The capstone project can be an extension of a project that began in an earlier course in the MS program.

### Project Guidelines

Successful project completion will be measured by the following criteria:

* Submission of all project components within the established deadlines
* Resource utilization within the initial design and plan, if applicable
* Meeting the desired performance/technology level as agreed upon by the student and faculty
* Effective and efficient utilization of the assigned resources, including peer review and instructor feedback
* Final presentation

The project milestones consist of two major components: documentation and implementation. As milestones are completed, they should be appended to the same document; at the end of the capstone, one document will be produced for the entire project. As you advance from course to course, you will need to share the work done in previous milestones with the instructor in order to provide a full context.

# Milestone 1: Project Proposal and Requirements Analysis

Project Proposal

**Objective:** The goal of the project proposal is to identify the purpose of the project, characteristics and requirements, outline the work to be done, and design a timeline for completing the work. The project proposal will contain a management plan, technical approach, and resources needed to complete the project tasks within the allocated time and budget.

**Deliverable:** Capstone Project Proposal and Requirements Document

Note: The project proposal needs to be completed before the end of CST-560. It must be approved by faculty before project development can occur. Once approved, you must complete and submit this milestone before the end of the course and start working on Milestone 2.

Ideas for the project may come from a variety of sources (e.g., personal ideation or an opportunity offered by your current employer). Some projects are created out of pure academic curiosity, while others are intended to create an open-source tool as a contribution to the programming community. As you overview the project and its objectives, describe its origin, intended use, and other important characteristics.

General Information

The general information section of the Capstone Project Proposal should list the following:

* Project name
* Author
* Project organization
* Project manager
* Date project proposal document is submitted Project Overview and Project Objectives

This section of the Capstone Project Proposal states the business case or the problem that this project will be solving.

Example: The purpose of this project is to develop an online voter registration application that allows citizens to register to vote from any location where web access is available. The goal is to develop a secure application that is easy to access and use so that a wider population will have the ability to register to vote.

Background is one of the key characteristics of a project, to explain why the project should be initiated, what prerequisites are, and what results are supposed to be obtained at the successful completion.

This section should contain a few sentences about the high-level business need or justification for the project, as well as how it originated and other background information. While this information may be obvious to the author and the project sponsor, it is imperative to be aware that a scope of work is often distributed well beyond the immediate project audience. This background clarifies the usefulness of the project to someone who is not familiar with the project.

Example: Fabrikam, Inc. is migrating its current fabrication design system to new hardware and software platforms. Fabrikam, Inc. employees designed and developed the current Fabrikam-D system, which does not use a machine learning algorithm. The Fabrikam-D system produces schematics and bills of material for fabrication jobs.

This project will add an artificial intelligence layer to efficiently produce schematics and bills of material. After individually reviewing this version of the project, Fabrikam, Inc. stakeholders will meet to approve the approach before the project begins.

Project Scope

This section of the Capstone Project Proposal states the scope of the project and the requirements, as well as the characteristics of the final deliverable. Include details about the project, such as:

* Is it a real time system?
* Is it all screen-based?
* What reports can be produced?
* Where does the information come from?
* What manipulation is required for the data?
* Does it store data?
* How will the system manage security?
* What problem(s) (if any) does the project address/solve?
* Is this an innovative project or an improvement to an existing one?
* Is it an implementation of a published method or algorithm? Project Completion

This section of the Capstone Project Proposal describes what measures will be used to calculate project success, taking into account completion criteria, assumptions, and constraints.

Completion criteria are explicit goals that must be attained to call an element of a project, or the entire project, “complete.” The example below shows criteria specific to the listed project. You will work with your instructor to set criteria appropriate to your project.

Example: Successful Project Measurements for the Online Voting System

1. Developed within the time allocated

1. Developed within cost
2. Meets the desired performance/technology level
3. Utilized the assigned resource effectively and efficiently
4. Customer approval and acceptance
5. No disruption to the main workflow of the organization
6. Coincides with the corporate culture

**Important Notes:** There can be projects in which performance is irrelevant because the main accomplishment is a product's completion (i.e., a prototype or proof of concept). Therefore, the issue of measuring performance versus product completion must be decided, and expectations set at this time. In other cases, project work will be done solely for the author, meaning there will be no approving party for the acceptance test final report. The instructor should help address and document this by providing continuous feedback before final approval of the Milestone.

Project assumptions are events and circumstances that are expected to occur during the project life cycle for successful implementation and completion. Although assumptions are the driving force that determines project success, typically they are outside the total control of the project. Project assumptions are accepted as true, often without any proof or demonstration.

Example: The online voting system is a new website; it is anyone’s guess what the demand will be. What should the performance goal be in terms of concurrent users? The server must be able to handle the load for the number of concurrent users while maintaining performance.

Constraints are limitations affecting the project. Examples include timelines, resource availability, funding, and other important factors that may impact project success.

Project assumptions and constraints tend to exist around project resource availability or competence. That is why, when developing a description of assumptions and constraints, there should be an attempt to give extra consideration to project cost, timing, and human-related issues.

Example: There is an unknown delivery timeframe for a key project component that would directly impact the other modules. Consider an alternative approach with a lesser ripple effect.

Project Controls

This section of the Capstone Project Proposal defines the potential risks, and lists steps to prevent or minimize the chances of the risk occurring.

Project Schedule

This section of the Capstone Project Proposal sets a schedule by implementing work breakdown and task time estimates. It will be essential to create a timeline (or Gantt chart) with dates for completion of key components of the project.

Requirements Analysis (Computer Science)

**Special Instructions:** These instructions are to be completed by MS Computer Science Program students. Instructions for the MS Data Science Program students are provided below.

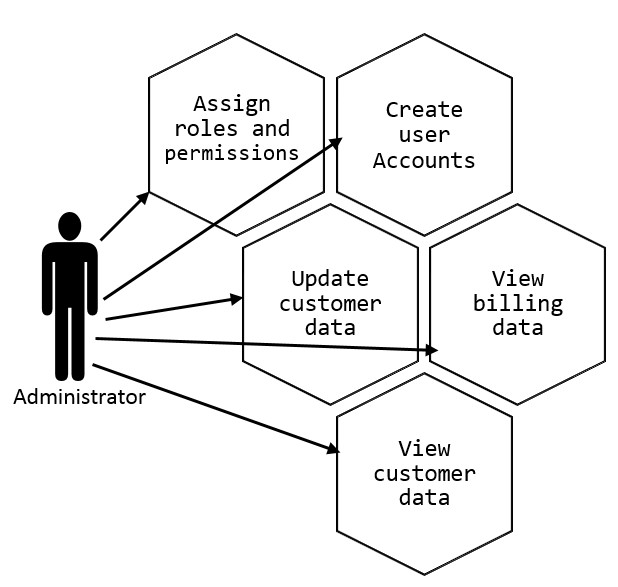
**Objective:** In the Requirements Analysis, the report definitions and layouts, screen definitions and layouts, data element definitions, workflow diagrams, performance metrics, unique features, and security matrices are gathered.

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. Nonfunctional 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 architectural design (in Milestone 2), implementation (in Milestone 3), and prioritization of development resources.

Use Cases

A use case describes a sequence of actions a project performs that provide an observable result of a value to a particular actor. The narrative text of a use case tells the story of how a user and the project interact in a simple, structured, and complete way.

Use cases may be provided as diagrams or text narratives. A generic example follows:



Top-Down Approach

The top-down design approach begins with a listing of the primary functions of the system. For example, if the software has three primary functions, then each function is treated as a major task. The major tasks are then analyzed and further decomposed into subtasks as needed.

Use the top-down approach in the analysis of the system requirements. This will then be used in the design of the system. Represent the top-down design with a diagram similar to a flow chart. A generic example follows for a system that calculates a utility bill:

generate

billing

get customer

meter readings

calculate

bill

print and

mail bill

record

customer usage data

validate

readings

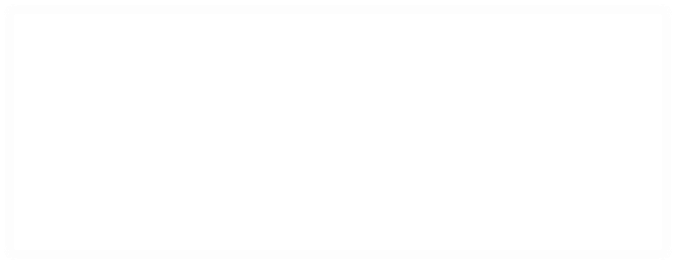
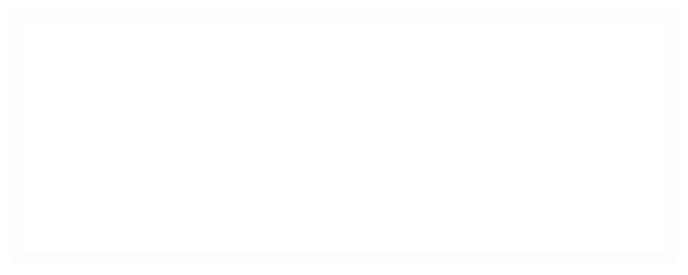
determine

rate category

record amount

due

There are many more examples online. Try to use a verb as the first word in the task description(s) to signify the action to be taken.

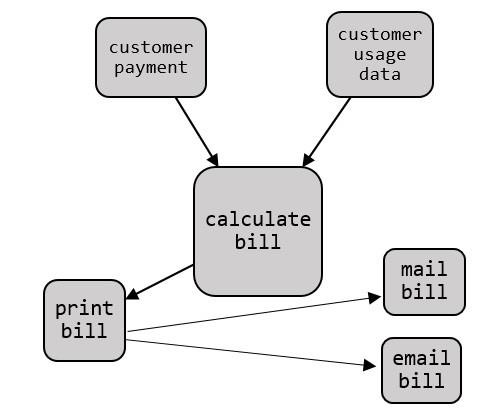
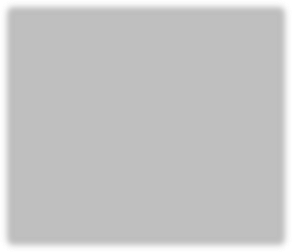


Technical Requirements

Provide a listing of the technical requirements that the system must fulfill. 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 requirement.

System Logical Model

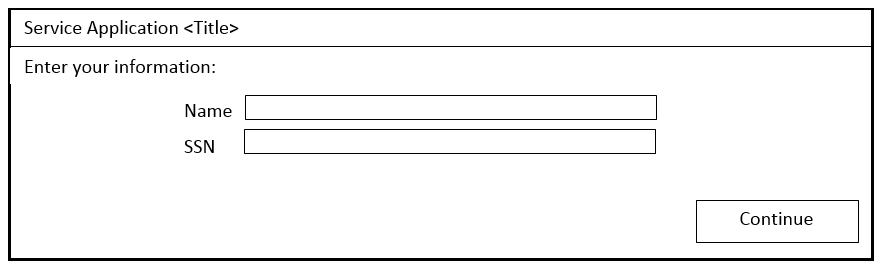
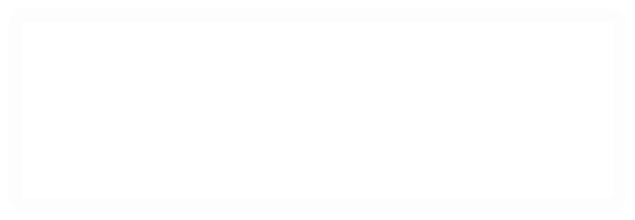
Provide a diagram of the logical architecture of the system. This diagram should reflect the functional requirements of the application, and illustrate the flow of information through the system. The logical design should be abstract; it should not include any implementation details. The purpose of the 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. An example of a logical architecture diagram of the utility bill system follows:



Screen Definitions and Layouts

Provide a draft of each user interface screen required by the system. These interface “sketches” will be refined in Milestone 2 (the design phase). Diagrams should include components such as, but are not limited to:

1. Title
2. Toolbars
3. Date entry fields
4. Content, such as text, charts, tables, etc. Example:

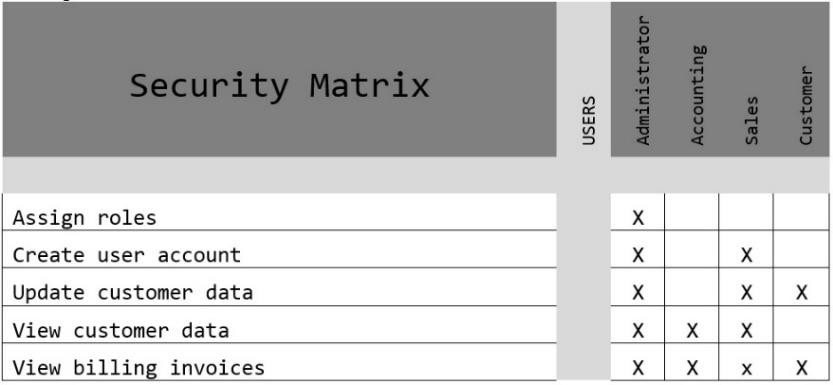
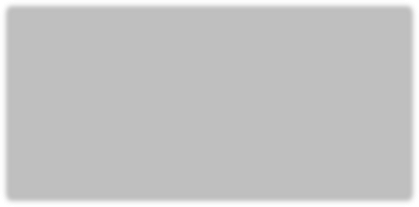


Work with the instructor to define the components of the project, such as screen definitions and layouts. The project may be software development, an implementation of a concept, a mobile application, or other. Flowcharts and technical diagrams are essential components in all projects.

Security

An often-overlooked security issue is badly designed software, which can be exploited by hackers to penetrate a system even if the hacking objective is a different component of that system. Therefore, discuss the security issues relevant to the system as well as the nature and magnitude. In addition, provide a matrix that illustrates the permissions/access rights of different types of users on the system.

Example:



Requirements Analysis (Data Science)

**Special Instructions:** These instructions are to be completed by MS Data Science Program students. Instructions for the MS Computer Science Program students are provided above.

**Objective:** In the Requirements Analysis, the report definitions and layouts, screen definitions and layouts, data element definitions, workflow diagrams, performance metrics, unique features, and security matrices are gathered.

In your documentation of requirements, be mindful of the two categories: Functional and non- functional 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 architectural design (in Milestone 2), implementation (in Milestone 3), and prioritization of development resources.

Use Cases

A use case describes a sequence of actions a project performs that provide an observable result of a value to a particular actor. More specifically, it is a sequence of actions initiated by the user and how the system reacts as a result. The narrative text of a use case tells the story of how a user and the project interact in a simple, structured, and complete way.

Use cases may be provided as diagrams or text narratives. A generic, abbreviate example of two use cases in an *Inventory Management* project follows:

### Use Case 1: Inventories RAM Sticks for IT Departments

The IT Department in a company must manage IT equipment ranging from full computers, such as laptops, desktops, and phones, to small components like RAM sticks and CPUs. All of the IT equipment the company has had to purchase in most cases. Therefore, if the hardware gets stolen, broken, lost, or forgotten then the company loses money. Some IT departments combat this by periodically requiring an employee to take inventory and sort a storage room. This product attempts to automate inventory and allows users to easily search locations for equipment.

The idea is that the robot will take inventory and sort items, then an IT Technician would log in to a web interface and have the ability to view the inventory in a table that is searchable. The following list breaks down things expected to happen in this use case.

* The system (an application server, robot, and database) must be able to locate RAM sticks in a RAM container and remember the location.
* An IT Technician should be able to access a website that gives details on inventory, such as:
  + Current locations of items
  + An item location history
  + Information about the hardware:
    - Serial number
    - Manufacturer if applicable
    - Brand
    - RAM speed (ex. 800MHz, etc.)
    - RAM size (ex. 2 GB, 4 GB, 8 GB, etc.)
    - RAM data transfer rate (DDR3, DDR4, etc.)
    - RAM rank (ex. 1Rx4, 2Rx8, etc.)
    - RAM in-line memory module type (SIMM, DIMM, etc.)
* The website should have an authentication system to ensure confidentiality, integrity, and availability (CIA).
  + Login system

### Use Case 2: Third-Party Support

The application created through this project should allow other developers to utilize an API to build their own applications. The idea is that other developers could then use the inventory management system as a service to enable their application. For example, this system could expand to inventorying food products in a fridge. An external application could then use the API to pull data about the food products that the user has available to produce a grocery list or a recipe. Another option is identifying items through an image and having a robot perform something else.

The third-party developer would expect the following:

* Access to inventory details through an API server.
* The ability to send an image to an application server and receive a response containing machine learning predictions of items within the image, such as:
  + Text detection
  + QR/Barcode detection
  + Object position
* Application access to authenticated APIs

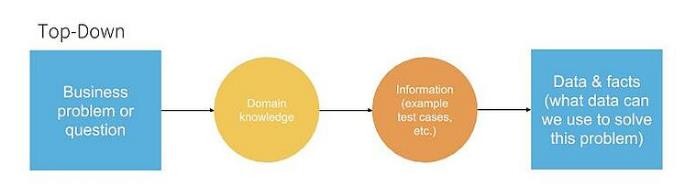
These two use cases are the foundation for the application’s success. Without inventorying RAM sticks, the proof of concept would not be strong enough to ensure a viable product, and developing the application for third-party support shows scalability.

Top-Down Approach

The top-down design approach begins with a listing of the primary functions of the system. For example, if the software has three primary functions, then each function is treated as a major task. The major tasks are then analyzed and further decomposed into subtasks as needed.

Use the top-down approach in the analysis of the system requirements. This will then be used in the design of the system. Represent the top-down design with a diagram similar to a flow chart. A typical top-down approach in data science is:

1. Formulate a hypothesis about the problem
2. Collect data
3. Build a model
4. Deploy the model
5. Test the hypothesis
6. Inform decision makers about the characteristics of an action plan A generic high-level example follows:



There are many more examples online. Try to use a verb as the first word in the task descriptions to signify the action to be taken.

Technical Requirements

Provide a listing of the technical requirements that the system must fulfill. 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.

Data Modeling and Analysis Process

Provide diagrams illustrating the type of database or data types and format. The data should reflect the requirements of the problem/challenge being addressed. Include naming conventions and data integrity; data archiving/retention; privacy and security information; data replication; data flow diagrams (DFD) and pipelines; statistical models; analytics models; data mining models; machine learning models. An example of a data science modeling process follows:



Monitoring

Deployment

Evaluation

Modeling

Exploratory Data Analysis

Data

Collection

Data

Requirement

Objective

The main steps in building a data science model are:

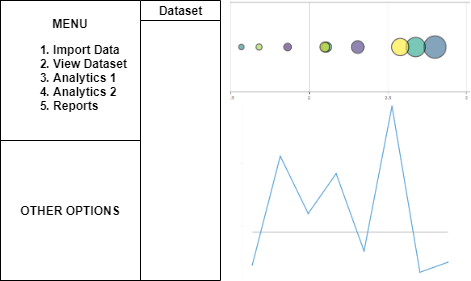
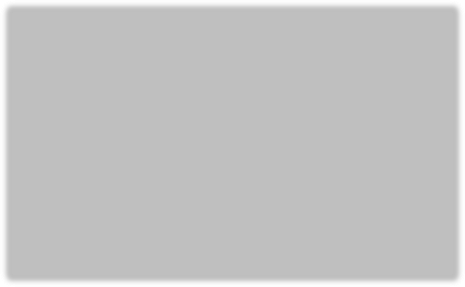
* Define objectives
* Collaborate with stakeholders
* Collect data for exploratory analysis
* Determine the functional form of the model
* Split the data into training and validation
* Assess the model performance
* Deploy the model
* Improve the model

If your project is in the area of Data Science, you may find the following resource useful, [Data](https://towardsdatascience.com/data-science-modeling-process-fa6e8e45bf02) [Science Modeling Process & Six Consultative Roles.](https://towardsdatascience.com/data-science-modeling-process-fa6e8e45bf02)

Screen Definitions and Layouts

Provide a draft of each user interface screen required by the system. These interface “sketches” will be refined in Milestone 2. Diagrams should include components such as, but are not limited to:

1. Title
2. Toolbars
3. Date entry fields
4. Content, such as text, charts, tables, etc. Example:

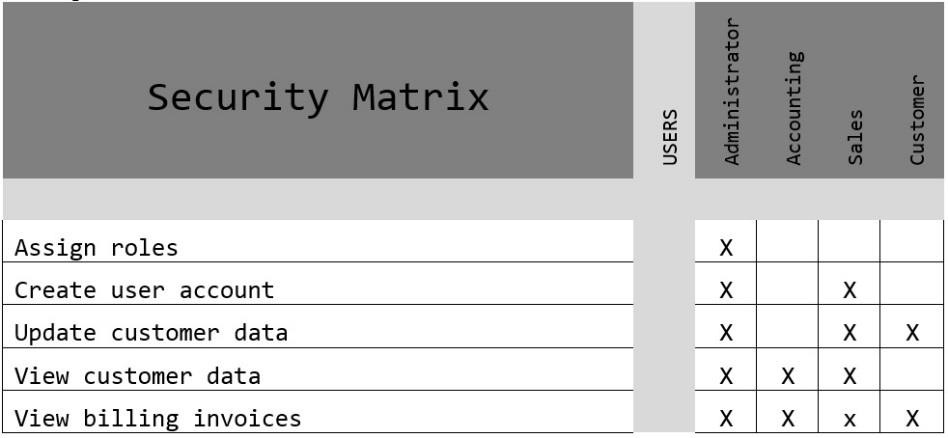
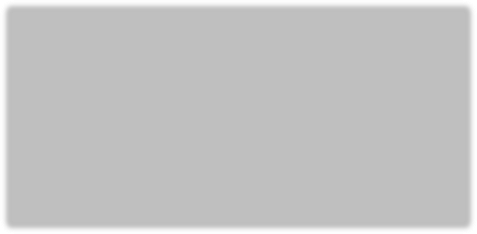


Work with the instructor to define the components of the project, such as screen definitions and layouts. The project may be software development, an implementation of a concept, a mobile application, or other. Flowcharts and technical diagrams are essential components in all projects.

Security

An often-overlooked security issue is badly designed software, which can be exploited by hackers to penetrate a system even if the hacking objective is a different component of that system. Therefore, discuss the security issues relevant to the system as well as the nature and magnitude. In addition, provide a matrix that illustrates the permissions/access rights of different types of users on the system.

Example:



# Milestone 2: Architectural Design / Model Pipeline Design

**Special Instructions:** Computer Science Capstone Projects must include the Architectural Design component, whereas Data Science Capstone Projects must include a Model Pipeline Design.

Architectural Design (Computer Science)

**Objective:** The purpose of the Architectural Design is to transform the requirements into complete and detailed system design specifications. The logical model created from the requirements analysis is used to develop a physical model of the application. Once the Architectural Design is approved, coding/development begins.

**Deliverable:** Architectural Design

Note: This milestone will be completed in CST-570. Once you finish the milestone and submit it to the instructor, you may start working on Milestone 3.

Architectural Design Title Page

List the stakeholders' names (including student and instructor), the project name, and document contributors.

Design Planning Summary

The design planning summary is an overview of the specific development project, a synopsis of the situation that led to the need, and a short description of the issues that the development project is going to solve, as well as a general description of the proposed solution and the rationale for the solution.

Overview of Design Concepts

Provide the high-level design of the proposed solution with supporting narrative text. This document should include mock-up screenshots for the proposed user interface screens, pseudocode, and/or flowcharts that show the logic for the program, as well as the anticipated process flow. The purpose of this document is to allow the stakeholder to approve the concepts before committing resources to the technical design and implementation.

Detailed Solution Architecture

Provide a detailed overview of the system architecture. **Keep in mind, the purpose of the detailed solution architecture is to provide sufficient information for a developer to produce the system.**

The architectural overview should include:

1. An overview of how the proposed design fits into the overall project structure.
2. Object and data element definitions, including a description of all objects with UML diagrams. Describe the purpose and characteristics of all data elements, data sources, and collection methods. The following site provides useful examples of UML diagrams: <https://www.uml-diagrams.org/index-examples.html>
3. Additional diagrams as needed by your particular system, such as entity-relationship diagrams, workflow diagrams, database schema, detailed flowcharts, algorithms, and so on.
4. Collaboration diagrams and/or sequence diagrams to show the workflows of components/packages/classes inside the component.
5. Algorithm descriptions, including detailed performance analysis and metrics.
6. Detailed specifications for all screens, interfaces and integration points, processes, conversion, and reports.
7. Describe any packages, software libraries, hardware, and software technologies, and their use.
8. Description of the approach and resources required to assure system security.
9. Use the template to list the hardware and software technologies.

Model Pipeline Design (Data Science)

**Special Instructions:** The Model Pipeline Design instructions are to be completed by MS Data Science Program students. Instructions for the MS Computer Science Program students are provided above.

**Objective:** The purpose of the Model Pipeline Design is to transform the requirements into complete and detailed system design specifications. The general process for building models is to set up the structure of the model (e.g., predictive, classifier, recommender, etc.), compile the model, fit the model, and evaluate the model.

**Deliverable:** Model Pipeline Design

Note: This milestone will be completed in the DSC -570 course. Once you finish the milestone and submit it to the instructor, you may start working on Milestone 3.

Model Pipeline Design Title Page

List the stakeholders' names (including student and instructor), the project name, and document contributors.

Design Planning Summary

The design planning summary is an overview of the specific development project, a synopsis of the situation that led to the need, and a short description of the issues that the development project is going to solve, as well as a general description of the proposed solution and the rationale for the solution.

Overview of Model Pipeline Design

The design concepts must address the following:

1. How will the data be obtained?
2. How will the data be scrubbed or cleaned?
3. How the data will be explored and visualized (e.g., to detect patterns and trends)?
4. What data model will be used (e.g., how will you set up a predictive model)?
5. What methods will be used to interpret the results of analysis? Detailed Model Pipeline Design

Provide a detailed overview of how the proposed design fits into the overall solution/business case structure**. Keep in mind, the purpose of the detailed model pipeline design is to provide sufficient information for a developer to implement the steps listed in the pipeline.**

The design overview should include:

1. The data sources
2. The dataset types and formatting
3. The data cleaning procedure
4. Method of initial data exploration and visualization
5. The data model used and its nature (e.g., predictive)
6. The methodology for interpreting the analysis results
7. Any configuration changes that will be required to develop and implement the proposed solution.
8. Describe the approach and resources required to assure system security, if applicable; otherwise, explain why security is not relevant.
9. Use the template to list the hardware and software technologies.

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: Implementation

**Objective:** The main objective of this milestone is to build the project by implementing in code everything planned in Milestones 1 and 2, as well as create the documentation based on the requirements and analysis decisions. This transforms the design documents into an executable program. System entities, such as objects, data tables, classes, and the like are developed. These entities are then integrated to create a working application. The application is prepared for testing. Code reviews are part of implementation.

**Deliverables**: Functional Requirements, Source Code Listing, Implementation Plan, and Complete Working Project and a User Guide.

Note: This milestone will be completed in CST/DSC-580. Once you finish the milestone and submit it to the instructor, you may start working on Milestone 4.

**Special Note:** Consult the instructor regarding documentation, assessment, expectations for mathematical proofs, analysis of algorithm correctness, analytics models, or other mathematical treatment of the subject.

Mapping of Functional Requirements

Ensure that all functional requirements have been satisfied by creating a mapping from each requirement to the module(s) or function(s) that satisfies it (diagrams and descriptions).

Example:

|  |  |
| --- | --- |
| (FR1)  Monthly bill sent by email | EBill.sendBill() Billing.generateBill() |

Source Code Listing

Provide a listing of the source code for the project. Include a brief description of all classes and files.

Example:

Billing.java

//defines a customer bill and calculates amount due

EBill.java

//generates, distributes, and archives paperless bill

Implementation Plan

Implementing the project is the act of actually making the software available for use.

Identify software that may need to be integrated with the libraries, databases, and other programs. Describe how the software will be deployed as an operational system. Describe the strategy, potential impacts, and activities that will be addressed in the implementation.

The implementation plan provides an outline of activities necessary to ensure that the system is available for use by its end users as originally planned.

Projects Requirements Review

Look back at what has been done so far in the project and make changes/modifications to previous content as necessary. It is important to continually review the project plan to address issues and oversights that are discovered.

# Milestone 4: Performance Analysis and Presentation

### Important Instructions: If the project has not been mostly completed in the previous milestones, consult with the instructor to either a) construct a timeline for completion or b) devise a plan to scale back on the project.

This plan might include:

* Scaling down certain functions and features of the project
* Modifying the approach to implementation
* Replacing certain modules
* Revising the plan for completing Milestone 4
* Communicating with the instructor frequently to ensure you receive advice in a timely manner

Most importantly, ensure that your plan for revising milestones is realistic and will still allow you to complete the project within the time allocated in the course.

Testing

**Objective:** Testing ensures the application functions as requested. During testing, verify the functionality of each module, revise and improve as needed.

**Deliverable:** Testing Process Document, including the test results, emphasizing execution of correctness and performance metrics.

**Special Note:** Consult the instructor regarding the documentation, assessment, expectations for mathematical proofs, analysis of algorithm correctness and performance, analytics models, or other mathematical treatment of the subject.

Module Test Cases

Develop test cases for all modules. Test modules as part of the development process, and continually revise code as necessary.

Example of Test Case Listing:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case Name:** | | | | |
| **Priority:** | | | | |
| **Module:** | | | | |
| **Test Objective:** | | | | |
|  | | | | |
| **Step** |  | **Test Detail** | **Expected Results** | **Problem/Issue** |
| **1** |  |  |  |  |
| **2** |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3** |  |  |  |  |

Requirements Testing

Once the components are tested and the system operates as designed, the application is tested against the requirements gathered in the requirements phase of development. Ensure that all requirements map to test scenarios. An example for documenting requirements testing (can be adapted to other tests), follows.

Example of Performance Verification:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Component:**  Ability to Initiate Robot Vehicle Controls |
| **Name of Developer:**  Student 1 |
| **Name of Reviewer:**  Student 2 |
| **Checklist**  **Type Pass Comments**  Functionality **Passed** Can perform commands given to it with full control.  Performance **Passed** Was able to perform with no missed commands, however more time would have allowed for more sophisticated movement. |

System Testing

System testing verifies that all components of the functional business requirements, business processes, data flows, and other system criteria are met. The developer(s) test specific end-to-end relevant processes until the complete application environment mimics the intended use upon release.

Operation and Maintenance

Prepare documentation that explains how to operate the software (user guide), and how to keep the software operational (system admin guide). The system admin guide details effective system maintenance, including maintenance procedures, standards, or other essential information on areas such as backup, networking and connectivity, access and authentication, cabling, and critical services.

Project Completion Phase

**Objective:** Complete the course project.

**Deliverable:** Completed final project, including source code, executable, comprehensive documentation, ready for presentation and demonstration via screencast video.

**Special Note:** Consult with the instructor regarding the finalized documentation (scope, level of detail, rigor) and the way it will be assessed.

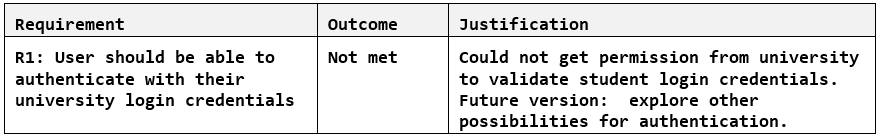
### Evaluation

Part of the completion and delivery step of a project is the importance of performing a comprehensive evaluation of all the milestones.

Review each of the milestone assignments and:

1. Determine whether modifications are necessary.
2. Update milestone documents wherever possible.

Prepare a document that lists each of the project requirements. For each requirement, state whether or not that requirement was adequately met. If the requirement was not adequately met, explain why. Include a section for planned improvements to your project. A sample excerpt from such a document follows:



Project Presentation Phase

Create a short, screencast presentation (approx. 7 minutes) in which the working project is demonstrated. The video of the presentation can be created with a variety of software tools, such as Loom, Movavi, QuickTime, or similar applications. You can upload the video to your favorite free video streaming site, like Loom or YouTube, and include the link to it in your documentation. In addition to the working project demonstration, present:

* An overview of the approach to implementation
* GUI elements pointing out key functionality
* An interesting function or feature in the code that was particularly challenging to implement
* How any unexpected challenges were overcome

Final Project Submission

In the project folder and the software repository (e.g., GitHub), upload the updated project containing all source code. Include comments in the code as needed. Adhere to efficient naming conventions (e.g., self-documenting variable and method names).

Include an updated version of the ReadMe file of the project, detailing the background of the project, approach to implementation, and what the user needs to do in order to run it. Upload the project presentation.

# Academic Integrity

For detailed information, refer to the [GCU Academic Integrity webpage](http://students.gcu.edu/academics/academic-integrity.php).

# Plagiarism

Students are encouraged to use code created by others**, if explicit written permission is given**. It is common practice for programmers to use each other’s code. Refer to Creative Commons

license. GitHub is a place where everyone uploads and everyone borrows others’ work. It is acceptable as long as these are only small portions of the code or essential libraries upon which students are allowed to build. The student must consult with the instructor, and obtain written approval for the use of borrowed code.

For detailed information regarding the University's Plagiarism Policies, refer to the [GCU](http://www.gcu.edu/academics/academic-policies.php) [University Policy Handbook](http://www.gcu.edu/academics/academic-policies.php).

# Sources of Information and Use of Reference Sources

The GCU Library is dedicated to supporting the University’s educational mission with library services that support both course content and research assignments. Be sure to look at the library resources at library.gcu.edu.

# Intellectual Property (IP)

For detailed information regarding the University's Intellectual Property Rights, refer to the [GCU](http://www.gcu.edu/academics/academic-policies.php) [University Policy Handbook](http://www.gcu.edu/academics/academic-policies.php).

# References

Grand Canyon University. (2015). Academic Integrity. <http://students.gcu.edu/academics/academic-integrity.php>

# Appendix

## Capstone Project Proposal

Retain this document throughout the development of the project. This form should be submitted as needed for instructor feedback. Make sure all instructions/prompts following each subtopic are removed before submission.

### General Information

Project name Author

Project organization, if applicable Project manager, if applicable

Date project proposal form is submitted

### Project Overview and Project Objectives

**State the Problem Background**

Explain why the project is being undertaken. Include pertinent background information for the project.

**Project Objectives**

List objectives that will be used to measure project success.

**Challenges**

List the known challenges that will be used to measure project success.

**Benefits and Opportunities**

Describe the benefits or opportunities resulting from project implementation.

### Project Scope

1. Give a clear, concise statement that states the scope of the project.
2. List the work breakdown required to satisfy the project objectives. Identify teams and other resources that may be required to successfully complete the project.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |

### Project Completion

1. Describe what measures will be used to calculate project success.
2. Use the template to list the project completion criteria.

|  |
| --- |
| Project Completion Criteria |
| 1 - |
| 2 - |

1. Use the template to list the project assumptions and constraints, if applicable. An assumption is an educated guess that a likely condition or circumstance is presumed to be true. A constraint is a limiting condition or circumstance that defines the project boundaries. Assumptions allow the project to succeed. Constraints restrict or limit the project execution.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |

### Project Controls

1. Use the template to define the risk(s), and list the steps to prevent or minimize the chances of the risk(s) occurring. The contingency plan describes alternative solutions to reduce the impact of the risk(s). An example of a contingency plan is to provide the customer a temporary web server if there are delays in delivery/completion.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
| **Event Risk** | **Risk Probability (high, medium, low)** | **Risk Impact** | **Risk Mitigation** | **Contingency Plan** |
| What is the risk? | What is the probability? | What is the impact if the risk occurs? | What can be done to minimize the risk? | What can be done to minimize the impact of the  risk? |
|  |  |  |  |  |
|  |  |  |  |  |

1. All projects have either anticipated and planned or unexpected changes. Describe any issues in management or management changes due to the anticipated and planned or unexpected changes. Use the template to list anticipated and planned or unexpected changes.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change Control Log | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |

1. Use the template to describe how the end user is involved in the software development, if applicable. Include relevant information about meetings, reviews, presentations, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Roles and Responsibilities | | | |
| Name | Team | Project Role | Responsibility |
|  |  |  |  |
|  |  |  |  |

### Project Schedule

1. Create a project schedule after all project tasks have been defined and prioritized. The schedule is driven by the sequence of courses in the MS program and the milestones due dates.
2. Set a programming schedule by implementing work breakdown and task time estimates. Create a timeline with dates for completion of key components of the project.

### Cost Estimate (if applicable)

1. Create 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 and will not require any cost calculations, please state that here.

### Issue Log

1. Use the template to identify and monitor project issues.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues Log | | | | | | | | |
| **ID** | **Issue Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | What is the issue? | How will this impact scope, schedule &  cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

* Yes ☐ No

**Project Approval**

* Instructor <Insert Name and Title>

## Requirements Analysis

### <Project Title>

**Note:** Make sure all instructions/prompts are removed before submission.

### Use Cases

Describe the sequence of actions a project performs with either a listing or graphic.

### System Design

Provide a top-down design of the system with a diagram similar to a flowchart.

### Technical Requirements

Provide a listing of the technical requirements for the system.

### System Logical Model or Data Science Model

If this is for a Computer Science Capstone project, provide a diagram of the logical architecture of the system. For a Data Science Capstone project, provide a data science modeling pipeline diagram (illustrating the source of data and the data types and format).

### Reports

Provide a listing of the reports that the application will generate, if applicable. If not, state that the application does not produce any reports, and provide additional documentation as described in the handbook.

### Screen Definitions and Layouts

Provide a layout diagram for each user interface screen in the application; if not applicable, define the components of the project as described in the handbook.

### Security

Provide a security matrix and a statement of security issues that the system must address. If there are no security issues for the system, state why.

### Other (as dictated by the context and scope of the project)

## Final Architectural Plan or Model Pipeline Design

Prepared for [Stakeholder Name] (if applicable) [Project Name]

Prepared by [Name]

Contributors [Document contributors] (if applicable)

**Note:** Make sure all instructions/prompts are removed before submission.

Design Planning Summary

1. Write an overview of this specific development project, a synopsis of the situation that led to the need (if

applicable), and a short description of the issues that the development project is going to solve, as well as a general description of the proposed solution and the rationale for the solution.

Overview of Design Concepts

1. Provide the high-level design of the proposed solution or business case with supporting narrative text.
2. Use the template to list the project deliverables. Include all components, features, and tasks your finished project is expected to perform.

|  |  |  |
| --- | --- | --- |
| Deliverable Acceptance Log | | |
| ID | Deliverable Description | Comments |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |

(For Computer Science Students Only) Detailed Solution Architecture

The purpose of the detail solution architecture is to provide sufficient information for a developer to produce the system. Write a detailed overview of how the proposed design fits into the overall solution/business case structure. Make sure to:

1. Create object model and use cases to depict the system.
2. Use collaboration diagrams and/or sequence diagrams to show the workflows of components/packages/classes inside the component.
3. Describe algorithms.
4. Include detailed specifications for all screens, interfaces and integration points, processes, conversion, reports, and any required modification to existing systems.
5. Include any solution configuration changes that will be required to develop and implement the proposed solution.
6. Describe the approach and resources required to assure system security, if applicable; otherwise, explain why security is not relevant.
7. Use the template (below) to list the hardware and software technologies.

(For Data Science Students Only) Detailed Model Pipeline Design

The purpose of the detailed model pipeline design is to provide sufficient information for a developer to implement the steps listed in the pipeline. Write a detailed overview of how the proposed design fits into the overall solution/business case structure, to include:

1. The data sources
2. The dataset types and formatting
3. The data cleaning procedure
4. Method of initial data exploration and visualization
5. The data model used and its nature (e.g., predictive)
6. The methodology for interpreting the analysis results
7. This section should also include any configuration changes that will be required to develop and implement the proposed solution.
8. Describe the approach and resources required to assure system security, if applicable; otherwise, explain why security is not relevant.
9. Use the template to list the hardware and software technologies.

|  |
| --- |
| Hardware and Software Technologies |
| 1 - |
| 2 - |
| 3 - |
| 4 - |
| 5 - |

|  |  |  |
| --- | --- | --- |
| Revision and Signoff Sheet | | |
| **Change Record** | | |
| **Date** | **Editor** | **Revision Notes** |
|  |  | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

## Test Plan

<Project Title>

The test approach is the overall test strategy that underpins the whole test plan. A test approach asks, “How are you going to test the software?” If this Test Plan is part of a larger parent project and there are other Test Plans for other parts of the overall system, then the test approach should dovetail with the other test approaches. Also, consider if you are going to use established, documented test processes or procedures, or if you will need to tailor tests specifically for this project. If so, then name these documents and include them in the reference section. As projects vary, consult the instructor regarding the **interpretation** and **adaptation** of certain elements in this module to better match the specific characteristics of your project.

In your test, quantify performance metrics and resource utilization such as speed of computation, memory usage, processing power, accuracy of results, size of data, and more, as pertinent to your project.

Use the template below, based on the IEEE 829 standard, [https://standards.ieee.org/standard/829-2008.html.](https://standards.ieee.org/standard/829-2008.html) **Note:**

Make sure all instructions/prompts are removed before submission.

1. Analyze the Product
   1. Who will use the product/system/tool?
   2. What is the intended use?
   3. How will it work?
   4. What software and hardware does it use?
2. Design the Test Strategy
   1. The scope of testing
   2. Testing type
   3. Document risks and issues
   4. Create test logistics
3. Define the Test Objectives
   1. List all software features you need to test
   2. Define performance goals for each feature tested
4. Define Test Criteria
   1. List specific criteria that measure successful completion of each task
   2. Define an overall *pass* criteria for the project
5. Resource Planning
   1. List all human resources needed to complete the test
   2. List all system requirements needed to complete the test
6. Plan Test Environment
   1. Describe the hardware and software required to test the project
7. Determine Test Deliverables *(as applicable to your project)*
   1. Test plan document (this document)
   2. Test cases
   3. Test scripts
   4. Error logs
   5. Etc.