EECS 348 Group Project

Version 1.0

[Note: The following template is provided for use with the Unified Process for EDUcation. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.] Marked (shaded) areas: items that are OK to leave out.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2/13/2024 | 1.0 | Filled in section 4: Management Process | Katie Nordberg |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 4](#_Toc11132094)

[1.1 Purpose 4](#_Toc11132095)

[1.2 Scope 4](#_Toc11132096)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc11132097)

[1.4 References 4](#_Toc11132098)

[1.5 Overview 5](#_Toc11132099)

[2. Project Overview 5](#_Toc11132100)

[2.1 Project Purpose, Scope, and Objectives 5](#_Toc11132101)

[2.2 Assumptions and Constraints 5](#_Toc11132102)

[2.3 Project Deliverables 5](#_Toc11132103)

[2.4 Evolution of the Software Development Plan 5](#_Toc11132104)

[3. Project Organization 5](#_Toc11132105)

[3.1 Organizational Structure 5](#_Toc11132106)

[3.2 External Interfaces 6](#_Toc11132107)

[3.3 Roles and Responsibilities 6](#_Toc11132108)

[4. Management Process 6](#_Toc11132109)

[4.1 Project Estimates 6](#_Toc11132110)

[4.2 Project Plan 6](#_Toc11132111)

[4.3 Project Monitoring and Control 7](#_Toc11132112)

[4.4 Requirements Management 7](#_Toc11132113)

[4.5 Quality Control 7](#_Toc11132114)

[4.6 Reporting and Measurement 7](#_Toc11132115)

[4.7 Risk Management 8](#_Toc11132116)

[4.8 Configuration Management 8](#_Toc11132117)

[5. Annexes 8](#_Toc11132118)

# 

# Introduction

[The introduction of the **Software Development Plan** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this **Software Development Plan**.]

## Purpose

[Specify the purpose of this **Software Development Plan**. The text below is provided as an example**.** ]

The purpose of the

The purpose of the *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by managers to direct the development effort.

The following people use the *Software Development Plan*:

* The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.
* **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

## Scope

[A brief description of the scope of this **Software Development Plan**; what Project(s) it is associated with and anything else that is affected or influenced by this document. The text below is provided as an example.]

This *Software Development Plan* describes the overall plan to be used by the EECS 348 Group Project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans.  
The plans as outlined in this document are based upon the product requirements as defined in the *Vision Document*.

## Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the **Software Development Plan**. This information may be provided by reference to the project’s Glossary.]

See the Project Glossary.

## References

[This subsection provides a complete list of all documents referenced elsewhere in the **Software Development Plan**. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.

For the **Software Development Plan**, the list of referenced artifacts includes:

* Iteration Plans
* Development Case
* Vision [you may prepare a vision statement of your own: what your vision for the project is]
* Glossary
* Any other supporting plans or documentation.

## Overview

[This subsection describes what the rest of the **Software Development Plan** contains and explains how the document is organized. The text below is provided as an example.]

This *Software Development Plan* contains the following information:

Project Overview  — provides a description of the project's purpose, scope, and objectives.  It also defines the deliverables that the project is expected to deliver.

Project Organization  — describes the organizational structure of the project team.

Management Process  — explains the estimated cost and schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

Applicable Plans and Guidelines — provide an overview of the software development process, including methods, tools and techniques to be followed.

# Project Overview

## Project Purpose, Scope, and Objectives

[A brief description of the purpose and objectives of this project and a brief description of what deliverables the project is expected to deliver.]

## Assumptions and Constraints

[A list of assumptions that this plan is based and any constraints, for example. staff, equipment, schedule, that apply to the project.]

## Project Deliverables

[A list of the artifacts to be created during the project, including target delivery dates. The text below is provided as an example.] Requirements, design specs, test cases, code

Deliverables for each project phase are identified in the Development Case. Deliverables are delivered towards the end of the iteration, as specified in section *4.2.4 Project Schedule*.

## Evolution of the Software Development Plan

[A table of proposed versions of the **Software Development Plan**, and the criteria for the unscheduled revision and reissue of this plan. The text below is provided as an example.]

The *Software Development Plan* will be revised prior to the start of each Iteration phase.

# Project Organization

## Organizational Structure

[Describe the organizational structure of the project team, including management and other review authorities.]

## External Interfaces

[Describe how the project interfaces with external groups. For each external group, identify the internal and external contact names. This should include responsibilities related to deployment and acceptance of the product.]

## Roles and Responsibilities *[the more details here, the easier your job; include contact info, availability info, expertise, …]*

[Identify the project organizational units that will be responsible for each of the disciplines, workflow details, and supporting processes. The text below is provided as an example.]

|  |  |
| --- | --- |
| **Person** | **Unified Process for EDUcation Role** |
| Katie Nordberg |  |
| Shravya Matta |  |
| Ibrahim |  |
| Evans |  |

Anyone on the project can perform [Any Role](..\..\..\process\workers\wk_any.htm) activities.

# Management Process

## Project Estimates

We estimate this project will not cost any money. We will use free software to develop and test the code. We will use free resources such as online websites to learn how to code the parts that we do not know. We estimate the project will take the entire semester to complete. While the program itself will not be strenuous to develop, we will break the project up over the entire semester and plan to complete it by the due date in late April or early May.

We will be meeting weekly or biweekly to discuss our progress on the project. Because we will meet often, if a deadline is missed, we will be able to reevaluate and adjust accordingly without delay.

## Project Plan

### Phase Plan

**Major Milestones:**

1. Project Management Plan completed.
2. Software Requirements Specifications completed.
3. Software Architecture Document completed.
4. Program implementation completed.
5. Test Cases Documented
6. User Manual completed.

### Iteration Objectives

Iteration 1 objective: Complete Project Management Plan.

Iteration 2 objective: Complete Software Requirements document.

Iteration 3 objective: Complete Software Architecture Requirements document.

Iteration 4 objective: Implement design.

Iteration 4.1 objective: Implement all simple Boolean logic expressions.

1. AND (&): Returns True if both operands are True
2. OR (|): Returns True if at least one operand is True
3. NOT (!): Inverts the truth value of its operand
4. NAND (@): Returns True only if both operands are False (opposite of AND)
5. XOR ($): Returns True if exactly one operand is True

Iteration 4.2 objective: Implement expression parsing such that operator precedence is taken into account.

1. Parse user-provided Boolean expressions in infix notation, respecting operator precedence and parentheses.

Iteration 4.3 objective: Give users the ability to input truth values for each expression.

Iteration 4.4 objective: Implement the ability to evaluate the expressions and display the results.

Iteration 4.5 objective: Implement errors handling.

a) Handles missing parentheses.

b) Handles unknown operators.

c) Handles other issues as they arise.

d) Provides informative error messages.

Iteration 4.6 objective: Implement parentheses handling within expressions.

Iteration 5 objective: Complete Test Cases documentation.

Iteration 6 objective: Complete User Manual documentation.

### Releases

So far, we have had no releases. This will be updated as the project continues. Our *expected* releases will be:

Version 1.0 with all simple Boolean logic expressions implemented.

Version 1.1 with expression parsing implemented such that operator precedence is taken into account.

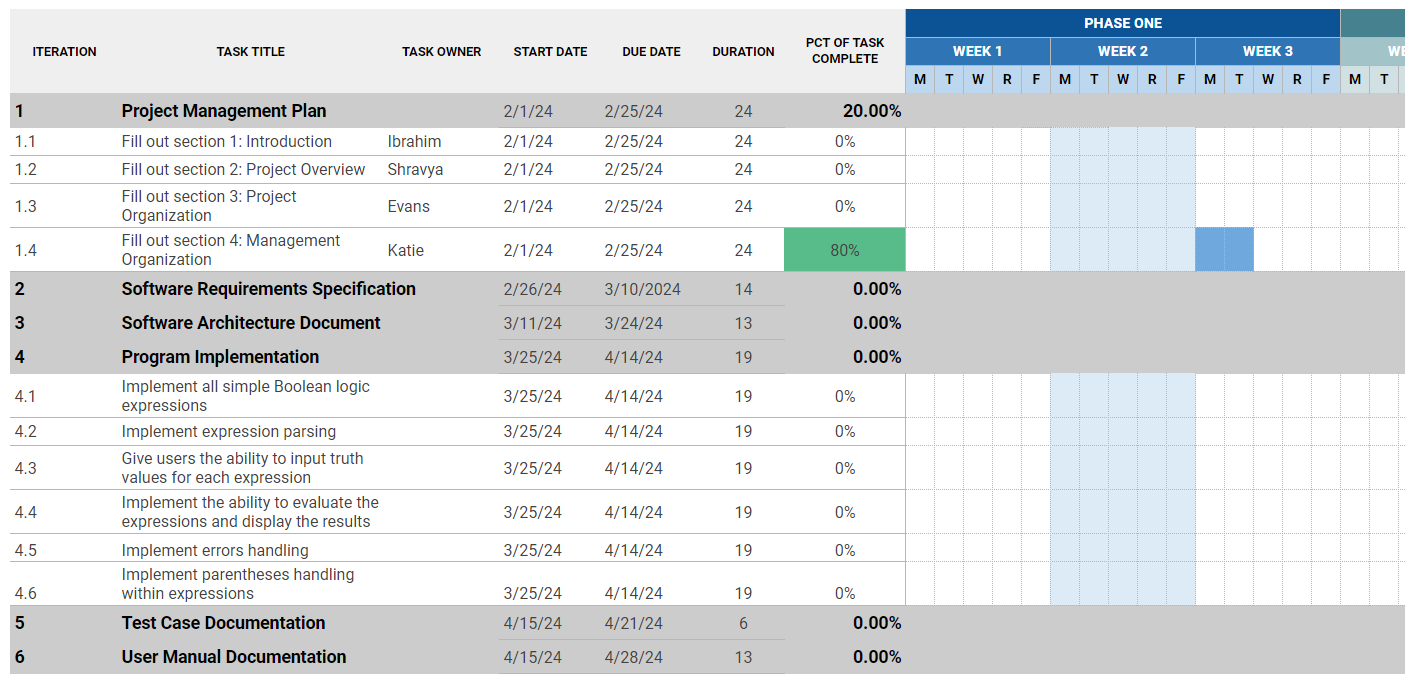
Version 1.2 with the ability for users to input truth values for each expression.

Version 1.3 with the ability to make the calculations and display the results.

Version 1.4 with the ability to handle errors.

Version 1.5 with the ability to respect parentheses in an expression.

### Project Schedule



### Project Resourcing

The team will be required to know and use C++ to complete this project. The team will also need to know UML to develop the requirements and design.

2 main team members, Katie and Evans, will be assigned to the programming portion of this project.

2 main team members, Ibrahim and Shravya, will be assigned to the writing portion of this project.

However, at certain times, all 4 team members will be expected to work in both programming and writing. So it is important that all team members are familiar with all aspects of the project.

No special training will be required. It is expected that all of the team will attend EECS 348 lectures.

## Project Monitoring and Control

## **Requirements Management**

The requirements for this system are captured in the project description document given on Canvas. Requested changes to requirements **or feature requests may be added to the GitHub issues page with the tag “feature.” This will allow us to keep track of the requirements, who is responsible for implementing them, and what progress has been made on them.**

## **Quality Control**

Defects will be recorded and tracked as Change Requests, and defect metrics will be gathered (see Reporting and Measurement below).

All deliverables are required to go through the appropriate review process, as described in the Development Case. The review is required to ensure that each deliverable is of acceptable quality, using guidelines and checklists.

Any defects found during review which are not corrected prior to releasing for integration must be captured as Change Requests so that they are not forgotten. They must also be entered as issues on GitHub.

## **Reporting and Measurement**

Updated schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

Metrics will be gathered on a weekly basis. They will include:

1. Tasks completed
2. Issues run into
3. Any test cases that now pass

## **Risk Management**

Risks will be identified in the design and implementation phase through identifying potential problems based on similar projects. Project risk is evaluated at least once per iteration and documented on GitHub issues.

## **Configuration Management**

GitHub issues will contain Change Requests and the controlled versioned repository of project artifacts.

All source code, test scripts, and data files are included in baselines. Documentation related to the source code is also included in the baseline, such as design documentation. All customer deliverable artifacts are included in the final baseline of the iteration, including executables.

The Change Requests are reviewed and approved by one member of the project, the team leader.

# 

# Annexes

[Additional material of use to the reader of the **Software Development Plan**. Reference or include any project technical standards and plans which apply to this project. This typically includes the Programming Guidelines, Design Guidelines, and other process guidelines. The text that follows is provided as an example.]

The project will follow the UPEDU process.

Other applicable process plans are listed in the references section, including Programming Guidelines.