Phoenix Star Dawgs

EECS 348 Group Project
Software Development Plan

Version 1.4

Revision History

Date	Version	Description	Author
2/13/2024	1.0	Filled in section 4: Management Process	Katie Nordberg
2/22/2024	1.1	Filled out introduction	Ibrahim Muhammad
2/22/2024	1.2	Filled out Organization	Evans Chigweshe
2/22/2024	1.3	Filled out Project Overview	Shravya Matta
2/24/2024	1.4	Compiled all into one final document	Katie, Ibrahim, Evans, Shravya

Table of Contents

Software Development Plan

1. Introduction

1.1 Purpose

This Software Development Plan describes the overall plan to be used by the EECS 348 Group Project to outline the approach for developing the Boolean Logic Simulator. This plan acts as a blueprint for effectively managing the development process, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans.

1.2 Scope

The project will focus on implementing a simplified Boolean logic simulator in the language C++, with basic operators such as AND, OR, NOT, NAND, XOR. The program will accept a Boolean expression and it will output True or False. The program will implement error handling. The purpose is to learn more about the software development cycle with a team and to create a Boolean logic simulator. The outline for this project comes from EECS 348 project description.

1.3 Definitions, Acronyms, and Abbreviations

This Software Development Plan sees the development of a C++ program acting that is a Boolean logic simulator. The scope includes implementing logical operations such as AND, OR, NOT, NAND, NOR and XOR, along with features like evaluation, and error handling.

AND - AND operation takes two inputs and produces an output that is true only when both inputs are true.

OR - OR operation takes two inputs and produces an output that is true when one input is true, or both are true.

NOT - This operation takes something and gives the opposite.

NAND - It combines AND and NOT. Does the same thing as AND but gives the opposite answer.

NOR - It combines OR and NOT. Does the same thing as OR but gives the opposite answer.

XOR - XOR operation takes two inputs and produces an output that is true if exactly one of the inputs is true.

1.4 References

- Iteration Plans
- Lecture Notes
- EECS 348 Project Description

1.5 Overview

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.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The purpose of the project is to develop a Boolean logic stimulator in C++ that can serve as a Boolean Expression Evaluator. The important aspect of this stimulator is to stimulate the behavior of logic circuits, which includes fundamental operations such as AND, OR, NOT, NAND and XOR. This project will help provide hands on experience in software engineering, emphasizing the development process from project planning to a fully realized project to a fully realized product. The deliverables from this project include code in C++ which will act as a Boolean expression evaluator, project management artifacts such as this project management plan, the requirements document, design document, test cases document, a user manual, and README file explaining how to use the program including examples of valid expressions and their expected outputs.

2.2 Assumptions and Constraints

The list of assumptions is

I. That everyone has knowledge about the Boolean logic.

II. The members of the team have enough knowledge to code in C++.

III. There is enough time to iterate and test each case.

The list of constraints is

I. Only the team members can work on this project.

II. There is also a due date where each phase should be delivered.

III. We must code the project in C++.

2.3 Project Deliverables

I. Project Management plan
II. Requirements Document
III. Design Document
IV. Test cases.
V. C++ program that works as a Boolean Expression evaluator
VI. User Manual

2.4 Evolution of the Software Development Plan

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

Version	Date	Revision Criteria
1.0	Date: <dd/mmm/yy>	Initial release
1.1	Date: <dd/mmm/yy>	Add features based on input from outside sources/feedback
1.2	Date: <dd/mmm/yy>	Adapt to changes from testing
1.3	Date: <dd/mmm/yy>	Addition of advanced features
1.4	Date: <dd/mmm/yy>	Improve performance
1.5	Date: <dd/mmm/yy>	Fix bugs and final testing
1.6	Date: <dd/mmm/yy>	Final copy

3. Project Organization

3.1 Organizational Structure

The project team is structured to efficiently manage two primary tasks: documentation and programming. Within each task, roles are delineated into primary and secondary responsibilities to ensure a smooth workflow. In the documentation task, Katie Nordberg serves as both the project manager and proofreader, overseeing the overall progress and ensuring quality control. Supporting Katie in this role is Shravya Matta, who also serves as a proofreader to ensure accuracy and consistency in the documentation. For the programming task, Ibrahim takes on the role of assistant project manager while also contributing as a code writer and code tester. Evans Chigweshe focuses primarily on writing code and conducting program testing to ensure functionality and reliability. This structure ensures that each aspect of the project is managed effectively, with a clear delineation of roles and responsibilities. Additionally, the presence of review authorities such as proofreaders and testers ensure that quality standards are upheld throughout the project lifecycle.

3.2 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Katie Nordberg	-Oversees the entire project, coordinating tasks, managing resources, communicating with stakeholders, and ensuring that the project stays on schedule and within budget. -Contributes as a software engineer and proofreader when needed. -Will support Shravya in writing documentation as needed, ensuring that all aspects of the project are well-documented and easily accessible.
Shravya Matta	-Serves as proofreaders, ensuring that all project documentation and written materials are free from grammatical errors, typos, and inconsistencies. Her role is essential for maintaining the professionalism and clarity of the project deliverables. -Responsible for creating project documentation such as user manuals, technical specifications, and system documentation. This documentation is crucial for ensuring that users and other stakeholders understand how to use the software effectively.
Ibrahim	-Will support Katie as the assistant project manager. He will assist with task coordination, resource management, and stakeholder communication. -Will also be involved in writing code and testing it, ensuring the quality and functionality of the software deliverables.
Evans Chigweshe	-Responsible for developing the software according to the project requirements and specifications. We will write code, design software architecture, implement features, and ensure that the software meets quality standards. -Responsible for testing the software to identify bugs, errors, and usability issues. This includes both manual testing and automated testing, ensuring that the software

	functions correctly and meets user expectations.

4. Management Process

4.1 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.

4.2 Project Plan

4.2.1 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.

4.2.2 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.

a) AND (&): Returns True if both operands are True

b) OR (|): Returns True if at least one operand is True

c) NOT (!): Inverts the truth value of its operand

d) NAND (@): Returns True only if both operands are False (opposite of AND)

e) XOR ($): Returns True if exactly one operand is True

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

a) 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.

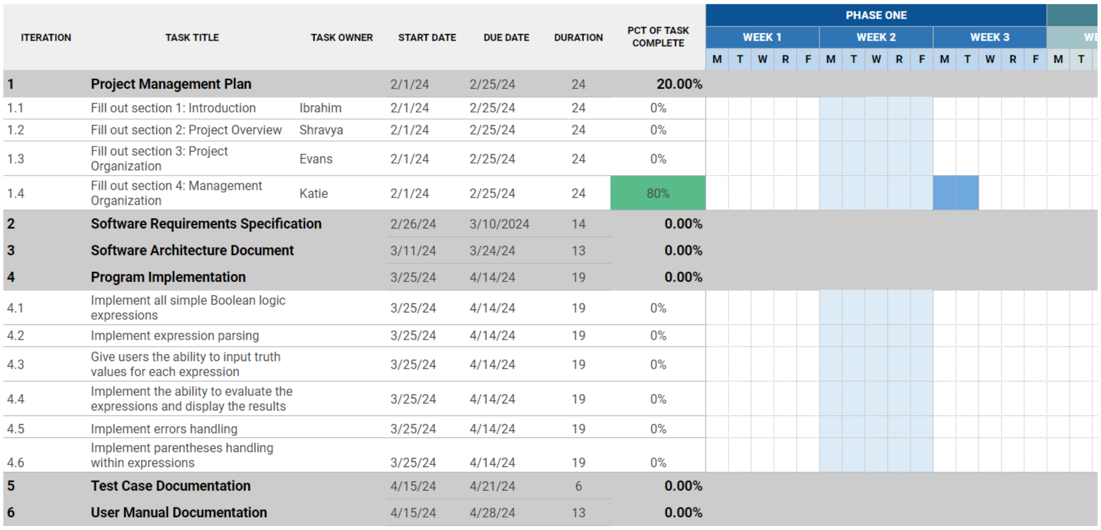
Iteration 5 objective: Complete Test Cases documentation.

Iteration 6 objective: Complete User Manual documentation.

4.2.3 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.

4.2.4 Project Schedule



PHASE ONE
WEEK 1
WEEK 2
WEEK 3
WE
M T W R F M T W R F M T W R F M T

ITERATION	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Management Plan**		2/1/24	2/25/24	24	**20.00%**
1.1	Fill out section 1: Introduction	Ibrahim	2/1/24	2/25/24	24	0%
1.2	Fill out section 2: Project Overview	Shravya	2/1/24	2/25/24	24	0%
1.3	Fill out section 3: Project Organization	Evans	2/1/24	2/25/24	24	0%
1.4	Fill out section 4: Management Organization	Katie	2/1/24	2/25/24	24	80%
2	**Software Requirements Specification**		2/26/24	3/10/2024	14	**0.00%**
3	**Software Architecture Document**		3/11/24	3/24/24	13	**0.00%**
4	**Program Implementation**		3/25/24	4/14/24	19	**0.00%**
4.1	Implement all simple Boolean logic expressions		3/25/24	4/14/24	19	0%
4.2	Implement expression parsing		3/25/24	4/14/24	19	0%
4.3	Give users the ability to input truth values for each expression		3/25/24	4/14/24	19	0%
4.4	Implement the ability to evaluate the expressions and display the results		3/25/24	4/14/24	19	0%
4.5	Implement errors handling		3/25/24	4/14/24	19	0%
4.6	Implement parentheses handling within expressions		3/25/24	4/14/24	19	0%
5	**Test Case Documentation**		4/15/24	4/21/24	6	**0.00%**
6	**User Manual Documentation**		4/15/24	4/28/24	13	**0.00%**

4.2.5 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.

4.3 Project Monitoring and Control

4.4 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.

4.5 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.

4.6 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:

a) Tasks completed

b) Issues run into

c) Any test cases that now pass

4.7 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.

4.8 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.

5. Annexes

The project will follow the UPEDU process.

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