<Tech Titans>

<Boolean Logic Simulator>
Software Development Plan

Version <1.0>

Revision History

Date	Version	Description	Author
<25/02/2024>	<0.1>	<Finalizing documentation>	<Alexander Phibbs>

Table of Contents

Software Development Plan

1. Introduction

I this document the Boolean Logic simulator will be outlined, the purpose of this project is to complete the 348 project description to its fullest and further pursue our knowledge and understand of what professional group orientated work looks like.

1.1 Purpose

The purpose of our software development plan is to outline what we will be needing to do for the project at hand. This will be an extensive list of differing to do's that we may all look back on later to make sure we have accomplished all of our deliverables.

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.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the Boolean Logic Simulator 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*.

1.3 Definitions, Acronyms, and Abbreviations

C++ : This is the coding language our project will be done in and will be referenced several times.

AND = &

OR = l

NOT = !

NAND = @

XOR = $

See the Project Glossary.

1.4 References

- *Iteration Plans*
- *Vision*
- *Glossary*

- *Any other supporting plans or documentation.*

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.

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

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

This project will be a Boolean Expression Evaluator, this means that a user will had our program several differing logical expressions and the program will then process it and give the logical output. Some of our deliverables include, a well documented C++ program with differing specified features, A user manual on how to use the program and what examples we have of this working, it will also include all of the project documentation that is specified in our canvas page.

2.2 Assumptions and Constraints

This project is to be completed in one semester's time and turned in before the stop day of the spring semester. There are six people that can work on the documentation and coding of this project.

2.3 Project Deliverables

Deliverables will be updated biweekly as we progress through this project. The documentation will be turned in on time and completed by the group, however everything the group completes must be turned in by the group leader.

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*.

2.4 Evolution of the Software Development Plan

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

3. Project Organization

3.1 External Interfaces

This project will be handed in to our lab TA Agraj. He will then run our code and grade the documentation according to the corresponding rubrics

3.2 Roles and Responsibilities

]

Person	Unified Process for EDUcation Role
Alexander	Team Lead
Theodora	Meeting Secretary
Cole	Division of Tasks
Justin	Debugger
Caden	Product Owner
Ceres	Second Team lead

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

This project will cost nothing in terms of purchasing for completion. The project will be completed by the agreed upon deadline.

4.2 Project Plan

4.2.1 Iteration Objectives

This document will be the first iteration objective, followed by the software Requirements -spec, then the Software Architecture, with Test Cases and then the user manual with our final code.

4.2.2 Releases

Our demo version of the project will be ready to use by the end of next month with the beta following two weeks after that and then the final version in mid april.

4.2.3 Project Schedule

The Requirements software will be done in two Weeks from Sunday the 25th of February, then the Software Architecture a month from that, following test Cases the first sunday of April and then lastly our user manual and final code coming mid to end April.

4.3 Project Monitoring and Control

- Requirements Management:We will be using github to store all of our project documentation and all of the coding will be done in C++ .
- Quality Control: This will be done by our debugger and then will follow through the chain of command with our second team lead and then demoed by the team lead before handing anything in. If there does happen to be something that slips through the team will come together to figure out what went wrong and then correct it to make sure the problem doesn't persist.
- Risk Management:There are very few risks involved with this project that are not getting a bad grade, the best way to avoid this is with thorough planning and following through with regular team meetings..

4.4 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.

4.5 Risk Management

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table.

Refer to the Risk List Document (CCC-DDD-X.Y.doc) for detailed information.

4.6 Configuration Management

Appropriate tools will be selected which provide a database of Change Requests and a 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 Change Control Manager role.

Refer to the Configuration Management Plan (EEE-FFF-X.Y.doc) for detailed information.

5. Annexes

The project will follow the UPEDU process.

Other applicable process plans are listed in the Spring 2024 term project description PDF..