<Company Name>

Cryptic Coders

Boolean Logic Simulator
Software Development Plan

Version <1.0>

Revision History

Date	Version	Description	Author
02/21/2024	<1.0>	Filled out Software Development Plan with preliminary information for the development stage of the project.	Aiden Burke, Mark Maloney, Ty Farrington, Brett Suhr, Muskan Sharma

Table of Contents

Software Development Plan

1. Introduction

This document is the software development plan for the Boolean Logic Simulator. This document will go through this project's purpose, overview of development, roles and management process. Each section provides insight into a step in the development of this project.

1.1 Purpose

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.
- **Any stakeholder** uses it to understand what to expect out of the process of development.

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

N/A

1.4 References

N/A

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

The purpose of this project is to create a Boolean Logic Simulator by exploring concepts of logic gates, truth tables and expression evaluation. The program will simulate how Boolean operators are interacted with and will return a Boolean calculation. The expected output will be software that takes in a Boolean equation, solves it and returns an output.

2.2 Assumptions and Constraints

- Programming done in C++
- Project to be completed by end of term with smaller, more specific deadlines as per assigned by instructor

2.3 Project Deliverables

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*.

- Project Management Plan
- Documentation of Progress File
- Requirements Document & Design Document
- Test Cases
- C++ Boolean Logic Simulator Program File

2.4 Evolution of the Software Development Plan

The original Sofware Development plan was created on 2/21/2024

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

3. Project Organization

3.1 Organizational Structure

N/A

3.2 External Interfaces

N/A

3.3 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Mark Maloney 773-899-2653 Available M-F 4-8pm	Project Leader: interface with the instructor, group management,
Aiden Burke 913-957-5490 Available M-F 4-8pm	Assistant Project Leader: Assist in management of group; help oversee project technical development.
Ty Farrington 913-291-6059 Available M-F 5pm-8pm	Records Manager: Takes notes of the meetings/recording a log of the team meetings and team contributions
Brett Suhr 630-945-4486 Available M-F 4-8pm	Quality Assurance Engineer: Debug code/ manage and control data
Muskan Sharma 785-550-5822 Available M-F 4-8pm	Technical Manager: Oversees project technical development of the github repository and logs all technical

	development for the group.

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

N/A

4.2 Project Plan

1) Project Plan
2) Software Requirement Specifications
3) Software Architecture
4) Test Cases
5) User's Manual

4.2.1 Phase Plan

N/A

4.2.2 Iteration Objectives

1) Project Plan-Laying out the steps for the development process
2) Software Requirement Specifications-Coming up with how we want our system to work and what requirements we want our software to fulfill.
3) Software Architecture -Designing the software for our boolean logic simulator
4) Test Cases -Testing the software and making changes to handle different cases
5) User's Manual- Coming up with a guide for how our system works and how a user should use it.

4.2.3 Releases

NA

4.2.4 Project Schedule

Iteration	Deadline
Project Plan	2/25/2024
Software Requirement Specifications	
Software Architecture	
Test Cases	
User Manual	

4.2.3 Project Resourcing

N/A

4.3 Project Monitoring and Control

- Requirements Management: Specify the information and control mechanisms which will be collected and used for measuring, reporting, and controlling changes to the product requirements.
- Quality Control: Describe the timing and methods to be used to control the quality of the project deliverables and how to take corrective action when required. Include techniques, metrics, criteria, and procedures used for evaluation— this will include walkthroughs, inspections, and reviews. Note that this is in addition to the Test Plan, which is not enclosed in the Software Development Plan.

- Risk Management: Describe the approach that will be used to identify, analyze, prioritize, monitor and mitigate risks. If available, refer to the **Risk List** document.

  Configuration Management: Describe the process by which problems and changes are submitted, reviewed, and dispositioned. Describe how project or product artifacts are to be named, marked, and numbered, including system software, plans, models, components, test software, results and data, executables, and so on. Describe retention policies, and the back-up, disaster, and recovery plans. **OR** if Available, Refer to the **Configurati**

4.4 Requirements Management

N/A

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.

4.6 Reporting and Measurement

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

The Minimal Set of Metrics, as described in the RUP Guidelines: Metrics will be gathered on a weekly basis. These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

Refer to the Project Measurements Document (AAA-BBB-X.Y.doc) for detailed information.

4.7 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.

4.8 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 references section, including Programming Guidelines.