<The Booleanators>

Boolean Logic Simulator in C++
Software Development Plan

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
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents [keep this; say N/A when inapplicable]

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of this *Software Development Plan* is to gather all information necessary to control the project. The project will be done communally by the group with all members participating in all levels of development.

The following people use the *Software Development Plan*:

- The **Team leader** 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

*[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.]*

Scope: The main point of this project is to build a simplified boolean simulator using C++. When making this simulator we need our program to be able to simulate behavior logic circuits such as AND ,OR, NOT, NAND, and XOR. The program itself should have complex logic circuits with multiple gates and input/output signals.

1.3 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.]*

Definitions:

- **UML Diagram**: Unified Modeling Language (UML) Diagrams are used in software engineering to visually represent various aspects of a system's design, structure, and behavior.

Acronyms/Abbreviations:

- **UML:** Unified Modeling Language

1.4 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.*

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview — Design, construct, and test a program that acts as a simplified Boolean Logic simulator using C++.

Project Organization — The team is composed of a leader and several members. The team will work communally to solve tasks and all members participating in all levels of development.

Management Process — The project will be divided into 5 major parts consisting of the Project Plan, the Software Requirements, the Software Architecture, Test Cases and User Manuals.

Applicable Plans and Guidelines — Using the Unified Process and UML modeling we will create the project using C++.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The aim of this project is to develop a program that simulates the behavior of logic circuits, including operations such as AND, OR, NOT, NAND, and XOR. The program will be able to handle complex logic circuits with multiple gates and input/output signals. The project will feature this project management plan, a requirements document, a design document, and test cases. The C++ program that functions as a Boolean expression evaluator with the specified features and a user manual explaining how to use the program.

2.2 Assumptions and Constraints

This plan is based on the assumption that the requirements for the project will not be changed and are frozen. This includes the boolean operators required as well as the programming language required. Constraints for the project include scheduling meetings among all team members and coordinating completion of deliverables for turn in.

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: Feb 25th

Requirements Document: TBD

Design Document: TBD

Test Cases: TBD

Boolean Expression Evaluator C++ Program: TBD

User Manual: TBD

2.4 Evolution of the Software Development Plan

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

Software Development Plan Ver. 1.0	First Draft - Project Management Plan

3. Project Organization

3.1 Organizational Structure

The team consists of a team leader and team members. All members have equal authority in the creation of the project and will work communally to finish project tasks by the project due date.

3.2 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Zack Corbin	Team Administrator/Programmer/Tester/LogKeeper
Cole Cooper	Programmer/Tester/LogKeeper
Talha Naseer	Programmer/Tester/LogKeeper
Thu Doan	Programmer/Tester/LogKeeper
Chloe Tran	Programmer/Tester/LogKeeper

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Plan

Project Management Plan: Feb 25th

Requirements Document: TBD

Design Document: TBD

Test Cases: TBD

Boolean Expression Evaluator C++ Program: TBD

User Manual: TBD

Each member will provide their own laptop to assist in the project tasks. Each member will use C++ in the creation of the project.

4.1.1 Iteration Objectives

Project Management Plan: Creation of Project Management Plan, detailed layout of project and development plan

Requirements Document: Creation of document listing requirements and constraints of project

Design Document: UML Class Diagram showing the top-level design for the project

Test Cases: Multiple and extensive testing of program to ensure effectiveness

Boolean Expression Evaluator C++ Program: Program that fulfills required functions and operates under given requirements

User Manual: Detailed manual explaining how to use the program, including examples of valid expressions and their expected outputs.

4.1.2 *Project Schedule*

[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.] [Limit to major project milestone, e.g., requirements, design, implementation, and testing]

Project Management Plan	*Target Date: Feb 25th*
Requirements Document	*Target Date:*
Design Document	*Target Date:*
Test Cases	*Target Date:*
Boolean Expression Evaluator C++ Program	*Target Date:*
User Manual	*Target Date:*

4.2 Project Monitoring and Control

[The following is a checklist of items to consider:

- *Requirements Management:*We will be using github to collect. measure, report, and control changes made to the product.
- *Quality Control:* For quality control on top of using git hub when we start to right actual code and work on the project we will give some one the role of being incharge of quality control to make sure we dont fall behind and bugs are found before they become a problem.
- *Risk Management: NA*
- *Configuration Management:* Our plan as a team is to use git hub as instructed this will allow us to manage our code and our project efficiently and will be able to go back to old versions if need be.

4.3 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.

4.4 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.

4.5 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

*[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.