Boolean Logic Simulator in C++

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 02/20/24 | 1.0 | Initial Template Filled out | Kevinh Nguyen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

This Software Development Plan is for a C++ program that can parse and evaluate Boolean expressions. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this Software Development Plan.

## 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.

## Scope

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

Change Request: Means the same as Pull Request at <https://github.com/KNEternity/348HL/pulls>

Issue: <https://github.com/KNEternity/348HL/issues>

Ticket: Synonymous with Story. Tickets are to be tracked via Issues, which will automatically link to associated Project Board at [https://github.com/KNEternity/348HL/projects](https://github.com/KNEternity/348HL/projects?query=is%3Aopen)

## 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

## 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.

# Project Overview

## Project Purpose, Scope, and Objectives

This project delves into the world of digital logic. You will develop a C++ program acting as a  
simplified Boolean logic simulator. 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 should be able to handle complex logic circuits with multiple gates and  
input/output signals.

This is a software engineering project, and as such the emphasis extends beyond the final product; it encompasses the development process. Project deliverables will include a meticulously crafted project plan, a requirements document, a design document that seamlessly aligns with the specified requirements, a set of rigorous test cases derived from the requirements and the design, and ultimately, the fully realized product.

## Assumptions and Constraints

* All team members are students (limited working and meeting times)
* Deadline of ~late April

## Project Deliverables

* Engineering Artifacts/Documentation
  + Project Management Plan
  + Requirements Document
  + Design Specification
* C++ Program to simulate behavior of logical circuits
  + User Manual / README

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

The *Software Development Plan* will be revised prior to the start of each Iteration phase. Unplanned revisions to this document can be done at any time for any reason, but common reasons may include fixing typographical errors, improving clarity of document, et. cetera.

# Project Organization

## Organizational Structure

## All members of the team have equal responsibility and there is no organizational hierarchy, except for the Project Leader who generally dictates Project Direction as well. Organizational roles are also flexible, there will be expected variance in role assignment through project iterations (except for Project Leader). All available roles are described now:

* Team Leader: Overall project leadership; Point of Contact with the instructor/client; handles meetings and coordination
* Technical Leader: Make decisions about how to implement requirements, primary approver of code changes
* Quality Assurance Engineer (QA): Automated Q/A Engineer or Manual Q/A Engineer; Write tests and suggests corrections to meet requirements; Secondary approver of code changes.
* Project Leader: Handles dividing up requirements into issues; tracks progress of work; Secondary Team Leaders

|  |  |
| --- | --- |
| **Person** | **Unified Process for EDUcation Role** |
| Kevinh Nguyen | Team Leader |
| Changwen Gong | Technical Leader |
| Rajkunwar Kaura | Product Owner |
| Riley England | Quality Assurance Engineer |
| Kemar Wilson | Project Leader 1 |
| Abdulahi Mohamed | Project Leader 2 |

# Management Process

## Project Plan

This section contains the schedule and resources for the project

### Iteration Objectives

[Briefly list the objectives to be accomplished for each of the iterations and Refer to the related **Iteration Plan Documents** for more details.]

### Releases

[A brief description of each software release and whether it’s demo, beta, and so on.]

### Project Schedule

[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.]

## Project Monitoring and Control

 [The following is a checklist of items to consider:

* 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 **Configuration Management Plan** document

The text that follows is provided as an example.]

## **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.

## **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.*

## **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.*

# 

# 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.