Huang and Co.

Boolean Expression Evaluator
Software Development Plan
Version 1.0

Revision History

Date	Version	Description	Author
17/02/2024	1.0	first review of entire document	Andrew, Keaton, Hunter, Elizabeth, Katharine
24/02/2024	1.0	first document rough draft	Andrew, Keaton, Hunter, Elizabeth, Katharine

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

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of the *Software Development Plan* is to house any and all relevant information necessary to control the project. It will describe the approach to the development of the program software and is the top-level plan generated which will be utilized by managers on the project to direct development efforts.

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 for the implementation of the Boolean Expression Evaluator 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

See the Project Glossary.

1.4 References

For the ***Software Development Plan***, the list of referenced artifacts includes:

- *Iteration Plans*
- Development Case
- *Vision*: “To show the world what’s really true and false”
- *Glossary*
- *Project Description [EECS 348 Canvas homepage]*
- *Any other supporting plans or documentation* [all of which may be found on the EECS 348 Canvas page under ‘Project’ in the Files section

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview — contains a description of the project’s purpose, objectives, and scope as well as a list of defined deliverables that the project is expected to deliver and when

Project Organization — describes the organizational structure of the project team

Management Process — outlines the estimated cost and schedule, defines the major phases and milestones for the project based upon the project description, and described how the project will be overseen

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 develop a program that can evaluate complex boolean logic expressions provided by the user and return an output of true or false. The program should be capable of working with NOT, AND, OR, NAND, and XOR functions denoted by various symbols, as well as parenthetical groupings of these functions. The program should also be able to handle any errors. By the end of the project, a working program with these abilities should be delivered, as well as the software engineering artifacts of several planning documents and a README file explaining the program.

2.2 Assumptions and Constraints

The project will be written exclusively in C++, and collaboration will be facilitated via the usage of GitHub. Planning will be conducted at weekly meetings every Saturday, and additional planning may occur before or after EECS 348 lecture on Tuesdays and Thursdays.

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*.

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 Organizational Structure

The project team consists of five members, all who are responsible for developing the Boolean Logic calculator. The team is led by a project manager who is well-versed in the language of the source code, C++, and development practices such as using Github for version control. The project manager provides guidance to the other team members regarding the above and represents the team by submitting all deliverables to the instructor. The project manager also leads the development by coordinating the requirements elicitation and use-case modeling.

The assistant project manager shares the responsibilities of the project manager by guiding the team on determining the requirements elicitation and use-case modeling.

The quality assurance engineer is in charge of conducting tests on each iteration of the program and documenting the results of those tests. The quality assurance engineer performs exploratory tests simulating various errors and modifies the program to handle these errors effectively.

The code quality analyst is in charge of making sure the source code of the program is well-written. Specifically, the code quality analyst provides adequate comments to improve readability, ensures that

naming conventions and code styling guidelines are followed, and breaks larger methods into smaller helper methods if need be.

The project coordinator is in charge of determining each meeting date, taking notes during the meetings, and recording a log of team meetings and each member's contributions.

All team members are responsible for writing the source code to program the Boolean logic calculator. The collaboration will be led by the project manager.

3.2 External Interfaces

N/A

3.3 Roles and Responsibilities *[the more details here, the easier your job; include contact info, availability info, expertise, …]*

Person	Unified Process for EDUcation Role
Andrew Huang	Project manager; provides technical guidance to other team members, conducts communication with stakeholder(s), and coordinates the requirements elicitation and use-case modeling *school email:* a558h089@ku.edu *personal email:* andrewhuang2019@gmail.com
Hunter Long	Assistant project manager; shares responsibilities with the project manager by coordinating the requirements elicitation and use-case modeling *email:* h780l020@ku.edu *personal email:* hwolong@gmail.com
Keaton Xu	Quality assurance engineer; conducts tests on each iteration of the program and logs the outcomes *school email:* keatx@ku.edu *personal email:* keatx05@gmail.com
Elizabeth Miller	Code quality analyst; oversee readability and uniformity of the source code *email:* e071m822@ku.edu *personal email:* archimedes.syracusan@gmail.com
Katharine Swann	Project coordinator; determines the meeting dates, takes notes during the meetings, and records a log of team meetings and team member contributions *email:* k228s253@ku.edu *personal email:* katharinesswann@gmail.com

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

N/A

4.2 Project Plan

Projeet artifact as well as iteration schedules]

The project will follow the UPEDU process.

4.2.1 Phase Plan

N/A

4.2.2 Iteration Objectives

Sprints	Objectives
1	Domain Engineering - learning C++ and boolean logic calculations
2	Requirements Elicitation - Stakeholder driven, research stakeholder requirements Software Requirements Specification (SRS) - Completed detailed report of requirements
3	Requirements Validation - Verify requirements and begin goals that mark quality of product Design - Models created to simulate program flow
4	Implementation - Completed Operator Support, Expression Parsing, Truth Value Input, Error Handling, and Parenthesis Handling Demo Release
5	Testing - Using valid and invalid expressions in file to run test cases and see if the program works as expected
6	Full Release

4.2.3 Releases

Demo - Release Demo version on GitHub, marking code in prototype state.

Full Release - Finished all test cases, and fixed bugs. Checked quality and deliverables. Full Release on discord.

4.2.4 Project Schedule

Project	Status	ETC
Project Plan	Completed	February 25, 2024
Requirements	Not started	March 10, 2024
Design	Not started	March 24, 2024
Implementation	Not started	April 7, 2024
Testing	Not started	April 21, 2024
User Manual	Not started	April 28, 2024

4.2.5 Project Resourcing

N/A

4.3 Project Monitoring and Control:

4.4 Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

4.5 Quality Control

Deliverables will be reviewed by a board of team members on a weekly basis based on the criteria outlined in the Software Development Plan. Inspection procedures will include maintaining readability of and running comprehensive test cases on the product. Defects discovered will be recorded and flagged for correction. Once completed, corrections will undergo additional review to ensure product standards are met.

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 the Inception Phase using the steps identified in the Unified Process for Small Projects activity “Identify and Assess Risks”. Project risk is evaluated at least once per iteration and documented in this table, ordered by magnitude.

Risk Ranking (High, Medium, Low)	Risk Description and Impact	Mitigation Strategy and/or Contingency Plan

4.8 Configuration Management

GitHub will be used to provide a history of Change Requests and a controlled versioned repository of project artifacts.

All source code, test scripts, and data files are to be included in the base branch. Documentation related to the source code is also included in the base branch, such as design documentation. All customer deliverable artifacts are included in the final branch of the iteration, including executables.

All changes will initially be implemented on a sub branch. The Change Requests are reviewed and approved by the Technical Lead and at least one of the two Project Leaders.

5. Annexes

The project will follow the UPEDU process.

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