Boolean Expression Evaluator

Configuration Management Plan

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 25/Feb/2024 | 1.0 | Initial publishing | Stonestreet, B., Schmidt, S. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 4](#_Toc159774863)

[1.1 Purpose 4](#_Toc159774864)

[1.2 Scope 4](#_Toc159774865)

[1.3 Definitions, Acronyms, and Abbreviations 4](#_Toc159774866)

[1.4 References 4](#_Toc159774867)

[1.5 Overview 4](#_Toc159774868)

[2. Code Formatting 6](#_Toc159774869)

[3. GitHub Procedures 7](#_Toc159774870)

[4. Integration Procedures 8](#_Toc159774871)

[5. Change Management Procedures 9](#_Toc159774872)

[6. Points of Contact 10](#_Toc159774873)

Configuration Management Plan

# Introduction

This document defines the procedures that will be followed to complete the design and implementation of the Boolean Expression Evaluator.

## Purpose

The purpose of the *Configuration Management Plan* is to gather all information necessary to design and implement the project. It describes the processes and procedures for the writing of source code, the integration of new code to the project, the administrative instructions for change control, and is the mid-level plan generated and used by the configuration manager to direct the design effort.

The following people use the *Software Development Plan*:

* The **Configuration Manager** uses it to detail design processes and procedures, and to track changes made to source code.
* **Project team members** use it to understand how to format their code, how to define any changes to in-production code, and how to create code that will integrate with the rest of the source code.

## Scope

This *Configuration Management Plan* describes the processes and procedures that will be used during design activities.

The plans as outlined in this document are based upon the directive of the Configuration Manager.

## Definitions, Acronyms, and Abbreviations

Definitions, Acronyms, and Abbreviations included in this document will be defined when written.

## References

This document is reference for and may reference the Boolean Expression Evaluator - Software Development Plan.

## Overview

[This subsection describes what the rest of the **Software Development Plan** contains and explains how the document is organized. The text below is provided as an example.]

This *Configuration Management Plan* contains the following information:

Code Formatting  — provides a description of the formatting for functions and the requirements for commenting of source code.

GitHub Procedures  — provides descriptions for the use of GitHub as the central repository for project source code.

Integration Procedures  — provides a description of the process that will be followed when new source code is integrated into the project.

Change Management  — provide detailed processes and procedures to execute changes to the code or the project.

Points of Contact  — provide contact information for the Configuration Manager and the Design team.

# Code Formatting

* All functions must be commented on with a multiline comment detailing the inputs, outputs, author, date created, and what the function does.
* Function declarations must be as such:

int main()

/\*commented\*/

{

//code

}

* All classes must be commented on with a multiline comment detailing the public methods, variables, and what the class is used for.
* Header files must be used for defining classes.
* Blocks of code must be separated from each other by one line. A block of code is defined as a series of lines of code that acts as a unit with a common goal of completing a task. The definition of a block of code is intentionally vague so that each programmer can define a block of code at their own discretion.
* Each block of code is to be commented on what its purpose is.

# GitHub Procedures

This section will be developed at the transition from the Elaboration phase to the Design phase of this project.

# Integration Procedures

This section will be developed at the transition from the Elaboration phase to the Design phase of this project.

# Change Management Procedures

* Defects will be recorded and tracked as Change Requests.
* Any defects found during review which are not corrected prior to a release date must be captured as Change Requests to be acted on during the next iteration.
* **Defects will identified by analysts and testers. Change Request forms will be submitted to the Change Control Manager for validation. After validation, the Change Control Manager will send the change request form to the Configuration Manager for implementation. The design team will perform corrective action.**

# Points of Contact

* Configuration Manager: Benjamin Stonestreet
* Change Control Manager: Kyle Whitmer