KI-69 Calculator

Arithmetic Expression Evaluator in C++

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 15/09/23 | 1.0 | Changed name/project name, assigned tasks | Ginny Ke |
| 22/09/23 | 1.1 | Filled out each section of document | Harrison Wendt |
|  |  |  |  |
|  |  |  |  |

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)

Software Development Plan

# 

# Introduction

The software development plan is to gather all information necessary to create a program that can parse and evaluate arithmetic expressions. The purpose, scope, definitions, references, and overview are all detailed below.

## Purpose

The purpose of the software development plan is to gather all information necessary to create a program that can parse and evaluate arithmetic expressions. It describes the approach to the development of the software and is created and used by managers to direct the development effort.

## Scope

This software development plan describes the overall plan to be used by the **KI-69 Calculator** project. The details of the individual iterations will be described in the Iteration Plans.

## Definitions, Acronyms, and Abbreviations

See the Project Glossary.

## References

* + Iteration Plans
  + Glossary

## 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 and the roles and responsibilities of the project team.

Management Process  — defines the project plan 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

The purpose of this project is to create a C++ program that can evaluate the arithmetic expressions containing PEMDAS operators as well as numeric constants.

This software development plan describes the overall plan to be used by the **KI-69 Calculator** project. The details of the individual iterations will be described in the Iteration Plans.

Our objective for the project is to figure out all our project plan details and assign roles. After we will start the design and construction aspect of the calculator.

The deliverables include the project management plan, design document, construction code, test cases, as well as user manual/ README file explaining our program.

## Assumptions and Constraints

Assumptions for this project include that there will be six people working on project, we all meet every week on Fridays, we can use our own laptops as well as school desktops, after program will be tested and the program will have all necessary documents when submitted.

Constraints for this project include the program must be in the C++ and the program must support all the key features mentioned in the project description PDF.

## Project Deliverables

The deliverables include the project management plan, design document, construction code, test cases, and user manual/README file explaining our program. To see specific dates refer to 4.2.4 Project Schedule.

## Evolution of the Software Development Plan

The evolution of the software development plan will be recorded at every Iteration phase.

# Project Organization

## Organizational Structure

The project team consists of 5 roles (Team Administrator, Assistant Team Administrator, Technical Leader, Assistant Project Leader, and Data Administrator/Quality Assurance Engineer) spaced across 6 individuals. This may be subject to change as the team decides what structure is most suitable. All communication between the stakeholders (TA, Professor, and Clients) will be handled by the Team Administrator. Additionally, updates will be submitted by the team administrator or assistant team administrator if need. Updates to the software will be tested by the quality assurance engineer and technical lead before submitting to the project leaders for review.

## External Interfaces (N/A)

## Roles and Responsibilities

|  |  |
| --- | --- |
| **Person** | **Unified Process for EDUcation Role** |
| Dylan Sailors | Team Administrator |
| Ginny Ke | Assistant Team Administrator |
| Harrison Wendt | Project Leader |
| Hayden Roy | Assistant Project Leader |
| Jenna Luong | Technical Lead |
| Nikka Vuong | Data Administrator/Quality Assurance Engineer |

Anyone on the project can perform [Any Role](..\..\..\process\workers\wk_any.htm) activities.

# Management Process

## Project Estimate (N/A)

## Project Plan

|  |  |  |
| --- | --- | --- |
| **Project Part** | **Start** | **Due** |
| Part 1: Project management plan | Week 3 | Week 5 |
| Part 2: Project requirements | Week 5 | Week 10 |
| Part 3: Project architecture and design | Week 10 | Week 11 |
| Part 4: Project Implementation | Week 11 | Week 13 |
| Part 5: Project test cases | Week 13 | Week 15 |
| Part 6: Project user manual | Week 15 | Week 16 |
| Final Implementation due | n/a | Week 16 |

### Iteration Objectives

* Expression Parsing
  + Implement a function to tokenize the input expression
  + Create a data structure, such as a stack, to represent the expression’s structure
* Operator Precedence
  + Define the precedence of the operators according to PEMDAS rules
  + Implement the logic to evaluate the expression while considering operator precedence
* Parenthesis Handling
  + Develop a mechanism to identify and evaluate expressions within parentheses
* Numeric Constants
  + Recognize numeric constants in the input.
* User Interface
  + Create user-friendly and legible command-line interface that allows users to enter expressions and display the calculated results
* Error Handling
  + Implement robust error handling to manage invalid expressions or mathematical operations

### Releases

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

|  |  |  |
| --- | --- | --- |
| Release | Date | Description |
|  |  |  |
|  |  |  |

### Project Schedule

|  |  |  |
| --- | --- | --- |
| **Project Part** | **Start** | **Due** |
| Part 1: Project management plan | Week 3 | Week 5 |
| Part 2: Project requirements | Week 5 | Week 10 |
| Part 3: Project architecture and design | Week 10 | Week 11 |
| Part 4: Project Implementation | Week 11 | Week 13 |
| Part 5: Project test cases | Week 13 | Week 15 |
| Part 6: Project user manual | Week 15 | Week 16 |
| Final Implementation due | n/a | Week 16 |

### Project Resourcing (N/A)

## Project Monitoring and Control

* Requirements Management: The project description document will be used to collect the information and control mechanisms used for measuring, reporting, and controlling changes to the product requirements. This document includes key features, project tasks, project guidelines, required deliverables, and test cases for valid and invalid expressions.
* Quality Control: Outputs will be compared to sample outputs given in the project description document to control the quality of the calculator. Team will walk through each deliverable to review and inspect it before final submission.
* Risk Management: Communication among team members and with TA will be used to identify, analyze, prioritize, monitor, and mitigate risks.
* Configuration Management: Problems and changes are to be submitted to project leader who will review and bring up problem or change with the team. The team will discuss and decide the best course of action. Artifacts will be marked in order starting at 00 and counting up based on start date. Retention policies include reverting back to an older version and storing documents on Google Drive or Github.

## **Requirements Management (N/A)**

## **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.

## **Reporting and Measurement (N/A)**

## **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

The project will follow the UPEDU process.

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