<Git Goblins>

<Calc-U-Later>

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <21/09/23> | <1.0> | <This was the initial completion of the product outline. > | <Daniel, Brett, Anna, Sabeen, James, Kaden> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

This Software Development Plan provides an overview of the document and is the embodiment of our program and its development timeline. The document includes dates, iterations, team dynamics, and scope providing a detailed view of the purpose and process of developing this calculator with C++.

## 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 Calc-U-Later 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

See the Project Glossary.

## References

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

* Iteration Plans
* Development Case
* Vision
* Glossary
* Any other supporting plans or documentation.

## 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

## Assumptions and Constraints

We are assuming that for this project we are going to be able to complete it this semester and that the user will use good faith when entering input. Though we are assuming the user will enter in input to the best of their ability, we are also ready to try and catch bad input (things such as parentheses that don’t close and other things like that).

## 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*.

## Evolution of the Software Development Plan

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

# Project Organization

## Organizational Structure

We have broken our team up into one scrum master who is going to be making the management decisions, and the rest of the team is going to be developing and working on the project. The scrum master will also contribute code. The scrum master will be the person who holds people accountable if code breaks production.

## External Interfaces

N/A

## Roles and Responsibilities

|  |  |
| --- | --- |
| **Person** | **Unified Process for EDUcation Role** |
| Daniel Neugent | Team Administrator |
| Brett Balquist | Product Owner/Utility Developer |
| Anna Ross | Project Lead 1 |
| Kaden Huber | Project Lead 2 |
| Sabeen Ahmad | Project Lead 3 |
| James Tormohlen | Project Lead 4 |

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

# Management Process

## Project Estimates

*The estimated cost for this project is $0, as all materials were previously obtained by team members and managers. The schedule for the project will be followed using the following:*

***Deliverable***         ***Due Date***

* + *Project Management Plan*         *9/24*
  + *Project Requirements*         *10/24*
  + *Project Architecture and Design*        *11/2*
  + *Project Implementation*        *11/16*
  + *Project Test Cases*        *11/30*
  + *Project User Manual*        *12/7*
  + *Final Project Implementation*        *12/15*

## Project Plan

*Project Part 1 assigned: Project Management Plan – 09/05-07/2023*

*Project Part 2 assigned: Project requirements – 09/19-21/2023*

*Project Part 3 assigned: Project architecture and design – 10/24-26/2023*

*Project Part 4 assigned: Project implementation – 10/31/2023*

*-There will be no additional resources required for this project. All software and hardware has previously been acquired including but not limited to MacBook's, Laptops, and VS Code.*

### Phase Plan

#### N/A

### Iteration Objectives

**Iterations:**

**Expression Parsing**

* This part of the project will handle the input and subdivide it and read each part of the input. It will create a data structure such that it can then analyze each part of the input to determine whether it is an operator or value.

**Operator Precedence**

* This iteration of the project will include PEMDAS rules and used the defined structure to then analyze which part of the equation to compute.

**Parenthesis Handling**

* This will handle parentheses and evaluate which parts of the function to compute first. Additionally, there may be a needed data structure to determine if the parentheses are valid making it a real equation.

**Numeric Constants**

* This iteration takes in the numeric values and addresses any inconsistencies with input such as decimals. Additionally, this will be implemented based on the previous iterations and determine whether the values are computer and what order they are computed in.

**Erorr Handling**

* The second to last iteration will include error handling. If there is input that is invalid it will be caught at this iteration and allow the program to continue running.

**User Interface**

* Finally, this will handle interaction with a user. This iteration will clean up output and input functions and ensure that the user has a friendly interface to interact with on the command line.

### Releases

|  |  |
| --- | --- |
| *Expression Parsing <v1.1> Alpha* | 11/3 |
| *Operator Precedence <v1.2> Alpha* | 11/5 |
| *Parenthesis Handling <v1.3> Beta* | 11/7 |
| *Numeric Constants <v1.4> Beta* | 11/9 |
| *Error Handling <v2.0> Beta* | 11/18 |
| *User Interface <v2.1> Demo* | 11/11 |

### Project Schedule

|  |  |
| --- | --- |
| *Project Management Plan* | 9/24 |
| *Project Requirements* | 10/24 |
| *Team Meeting* | 10/28 |
| *Project Architecture and Design* | 11/2 |
| *Implement Expression Parsing* | 11/3 |
| *Implement Operator Precedence* | 11/5 |
| *Implement Parenthesis Handling* | 11/7 |
| *Team Meeting* | 11/8 |
| *Implement Numeric Constants* | 11/9 |
| *Implement Error Handling* | 11/11 |
| *Implement User Interface* | 11/16 |
| *Team Meeting* | 11/17 |
| *Project Implementation* | 11/18 |
| *Team Meeting* | 11/25 |
| *Project Test Cases* | 11/30 |
| *Project User Manual* | 12/7 |
| *Final Project Implementation* | 12/15 |

**Deliverable** **Due Date**

|  |  |
| --- | --- |
| *Project Management Plan* | 9/24 |
| *Project Requirements* | 10/24 |
| *Project Architecture and Design* | 11/2 |
| *Project Implementation* | 11/16 |
| *Project Test Cases* | 11/30 |
| *Project User Manual* | 12/7 |
| *Final Project Implementation* | 12/15 |

### Project Resourcing

There are six student team members, all with prior experience with Python and other languages like C++. Using skills obtained from EECS 348 throughout the semester, they will be able to complete the project and no special training will be required. One team member has experience on a live team and previous experience involving NodeJS and JSON.

## 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.
* Reporting and Measurement: Describe reports to be generated. Specify which metrics should be collected and why. **OR** if available, refer to the **Project Measurements and Project Measurements** document
* 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

## **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.

## **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**

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.*

## **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.