
Team 12

<Calculator>
Software Development Plan
Version <1.0>

<Calculator>	Version: <1.0>
Software Development Plan	Date: 29/Sept/24
<Project Plan>	

Revision History

Date	Version	Description	Author
23/Sept/24	<1.0>	<details>	<name>

<Calculator>	Version: <1.0>
Software Development Plan	Date: 29/Sept/24
<Project Plan>	

Table of Contents

1. Introduction.....4

1.1 Purpose..... 4

1.2 Scope..... 4

1.3 Definitions, Acronyms, and Abbreviations.....4

1.4 References..... 4

1.5 Overview..... 5

2. Project Overview.....5

2.1 Project Purpose, Scope, and Objectives..... 5

2.2 Assumptions and Constraints.....5

2.3 Project Deliverables..... 5

2.4 Evolution of the Software Development Plan..... 5

3. Project Organization..... 5

3.1 Organizational Structure.....5

3.2 External Interfaces.....6

3.3 Roles and Responsibilities..... 6

4. Management Process..... 6

4.1 Project Estimates..... 6

4.2 Project Plan..... 6

4.3 Project Monitoring and Control..... 7

4.4 Requirements Management..... 7

4.5 Quality Control..... 7

4.6 Reporting and Measurement..... 7

4.7 Risk Management.....8

4.8 Configuration Management..... 8

5. Annexes.....8

<Calculator>	Version: <1.0>
Software Development Plan	Date: 29/Sept/24
<Project Plan>	

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of the *Software Development Plan* is to gather all information necessary to control the Calculator project. It describes the approach to the calculator project's software development 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 to make sure it is on schedule.
- **Project team members** use it to understand what is expected as well as when they are expected to do it. In addition, tell them what other activities require their attention.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the Calculator 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

Definitions, acronyms, and abbreviations will be described in the sections they are used as well as described here when they are created.

See the [Project Glossary](#).

1.4 References

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

- *Iteration Plans 4.2.1 pg 6*
- *Vision*
- *Glossary Available at [Link](#)*
- *Any other supporting plans or documentation.*

1.5 Overview

This document describes the process for creating our software product and serves as technical guidelines. It contains:

Project Overview	—	A technical description of the product, dates, use cases, and constraints.
Project Organization	—	This section describes who will do what to contribute to the success of this project as well as external dependencies of it.

<Calculator>	Version: <1.0>
Software Development Plan	Date: 29/Sept/24
<Project Plan>	

Management Process — A log of the progress made, version control, target dates, milestones, and information to look back on toward the end of the project.

Annexes — A calendar showing the schedule of weekly meeting times.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The purpose of this project is to create a versatile expression evaluator that can take an arithmetic expression, parse it, and calculate the result accurately according to the order of operations. This project will support a variety of operators. It will provide a user-friendly interface that will allow a user to enter expressions, and will then display the calculated results. Finally, it will be able to handle a variety of errors, ranging from improper user input to incalculable equations like divide by 0.

2.2 Assumptions and Constraints

This project will be coded in C++ and utilize a Command Line Interface (CLI) user interface. It will be created by a group of 7 students using personal and school-provided technology, and will be finished by the end of the semester.

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*.

A schedule is to be decided. The project will deliver the following:

Expression parsing - This program will be able to parse arithmetic expressions entered by the user, including order of operations and parentheses.

Operator Support for addition, subtraction, multiplication, division, modulo and exponentiation.

Parenthesis handling - This program will be able to handle expressions enclosed within parentheses

Numeric constants - This program will be able to recognize integers as input

User interface - This program will have a user-friendly and legible CLI that allows users to enter expressions and display results.

Error handling - this project will be able to manage scenarios like division by zero or invalid expressions

2.4 Evolution of the Software Development Plan

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

Version	Date	Changes
1.0.0	9-29-2024	Initial Version

<Calculator>	Version: <1.0>
Software Development Plan	Date: 29/Sept/24
<Project Plan>	

3. Project Organization

3.1 Organizational Structure

The organizational structure of our project team is considered an Agile Scrum model. We have a scrum master who is leading all of our scrum meetings where we plan out what we will work on in the following week. We then have sprints where we return functional snippets of code and are then later able to combine all of our work into a complete product.

3.2 Roles and Responsibilities

Person	Unified Process for EDUcation Role
Carson Schraad	Assistant Project Lead
Caleb Neil	Quality Assurance Engineer
Noah Haskins	Scrum Master
Maxfield Freeman	Project Manager
Will Godderz	UI/UX Designer
Jaydee Brown	Version Control
Maren Proplesch	Technical Lead

Anyone on the project can perform [Any Role](#) activities.

4. Management Process

4.1 Project Estimates

Project is on schedule and should take about 3-4 months to complete.

4.2 Project Plan

-We plan to meet every Monday from 4-5 as retrospective meetings to our week long sprints. At these meeting we will discuss our plan for the next week until the project is complete.

4.2.1 Iteration Objectives

We plan to use an Iterative development model. The first spirit we will release the expression parsing as it is the basis for the program. We then plan to develop the operators starting with addition. We will then do error handling and order of operations adding in parentheses handling.

4.2.2 Releases

Each release will consist of a new functionality added to the program until all requirements are met. We will start with the more simple operations and finish with error handling and order of operations.

4.2.3 Project Schedule

- 9/30: expression phasing- get an input and store it in a stack or tree
- 10/6: Operations support- implement the different math operations to work given correct input
- 10/13: input changing-allow our program to work with floats as well as ints
- 10/20: User interface-create command lines and input lines so our program is user friendly
- 10/30: Error handling to ensure program always runs correctly

4.3 Project Monitoring and Control

- Requirements Management - This section describes what requirements our project has and constraints that our project faces.

<Calculator>	Version: <1.0>
Software Development Plan	Date: 29/Sept/24
<Project Plan>	

- Quality Control - This section describes our approach to ensure quality is exemplary in our artifacts.
- Reporting and Measurement - This section describes our process for scheduling and budgeting our project.
- Risk Management - This section describes how our risks will be identified and what steps we will implement to assess these risks.
- Configuration Management - This section describes how and where we store our artifacts and previous versions of said artifacts.

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

Defects and bugs will be recorded as Change requests and cataloged in the Reporting and Measurement section below.

Deliverables will go through a review process that is described in the Development Case section. This review process is essential for checking that our deliverables are of acceptable quality based on our guidelines.

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 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.

<Calculator>	Version: <1.0>
Software Development Plan	Date: 29/Sept/24
<Project Plan>	

4.8 Configuration Management

Github will be used to store change requests and maintain a database of versioned project artifacts. Github will allow ease of access to make changes and revert said changes to our project.

All source code, data files, and test scripts are maintained on said database. Documentation related to these artifacts are also included on Github. Backups for these artifacts are provided as previous versions on Github as well.

Change requests are assessed by the Project Lead and the Technical Lead to verify said changes are necessary and satisfactory. Project Lead will verify that the change meets the project's naming criteria..

5. Annexes

The project will follow the UPEDU process.
Other applicable process plans are listed in the references section, including Programming Guidelines.
References have practical names to quickly convey to the reader what they represent.

Meeting Agenda #1 September

Monthly Agenda

Monday 4:00pm - 5:30pm
Anschutz 429

9/16/24

- Construct meeting schedule
- Assemble members on a group discussion page
- Discuss steps to complete project plan

9/23/24

- Select roles for group members
- Formulate a plan for completing the project plan
- Assign work to group members

9/23/24

- Discuss what went wrong and how we can improve in making the project plan
- Discuss steps we need to take to complete a software requirements plan

September

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5