
Team8 Co

Calc. Co
Software Development Plan
Version <1.0>



Calculator	Version: <1.0>
Software Development Plan	Date: <26/SEPT/24>
01	

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<1.0>		<name>

Calculator	Version: <1.0>
Software Development Plan	Date: <26/SEPT/24>
01	

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: <26/SEPT/24>
01	

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of this document is to provide all the necessary information required to complete our project and keep it running smoothly. It outlines the approach we will be taking for software development and creates a plan.

1.2 Scope

This Software Development Plan outlines the overall strategy for the Calculator project, including how the product will be released. Specific details for each development phase will be covered in separate Iteration Plans. We will use this to track progress and keep on schedule.

1.3 Definitions, Acronyms, and Abbreviations

Ticket- A request to fix an issue in our program

Backlog - A prioritized list of things that need to be done

Calculator Operations - This refers to the mathematical operations (addition, subtraction, multiplication, division, etc.) we will program our calculator to do.

1.4 References

Iteration Plans

Date: 26/09/2024

Source: [Iteration Plan](#)

Vision Document

Date: 26/09/2024

Source: [Calc.Co Vision Plan](#)

Glossary

Date: 09/26/2024

Source: [Glossary](#)

GitHub

Date: 09/19/2024

Source: <https://github.com/BigIronDestroyer/Calc.Co>

1.5 Overview

Project Overview: Details the purpose, scope, objectives, and expected deliverables.

Project Organization: Describes the team's structure and roles.

Management Process: Covers estimated costs, schedules, major phases, milestones, and monitoring strategies for the project.

Calculator	Version: <1.0>
Software Development Plan	Date: <26/SEPT/24>
01	

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

This project aims to develop a user-friendly calculator application that performs basic operations. Our calculator can add, subtract, multiply, and divide. Our application's intended use is for ourselves and our teacher.

2.2 Assumptions and Constraints

This project assumes standard mathematical operations such as addition, subtraction, multiplication, and division, trigonometric functions, exponents, and logarithms. This project will operate under a desktop client and strive for an intuitive user interface. Currently, the project is constrained to be done in 13-16 weeks using a team of 6 individuals with varying levels of experience and roles. Hardware is limited to student laptops, and those will also be the testing environments.

2.3 Project Deliverables

We will create a few artifacts throughout this project. There will be requirements documentation, source code, test cases, design specs, and results. The target dates for these deliverables will be in section 4.24.

2.4 Evolution of the Software Development Plan

The Software Development Plan will be updated before each new phase starts. Changes will be made as needed if there are unexpected issues or changes in project requirements.

3. Project Organization

3.1 Organizational Structure

This project involves a team with six specific roles, each role has unique responsibilities. While each role focuses on particular tasks all members need to help with most duties like coding, meeting logs, and reports. Cooperation and communication are necessary for this project

3.2 Roles and Responsibilities

Person	Unified Process for EDUcation Role	Contact	Expertise
Kai Barnhart	Project Manager	913-963-6360	Python
Abdulaziz Ali	Technical Lead	913-401-5539	C++, UI/UX
Muhammad Abdullah	Quality assurance Lead	929-301-2903	Python, HTML, Java
Ibaad Khatib	Scrum Master	913-972-3412	Python, Java, C++
Julian Gutierrez	UI/UX Design	316-204-2121	Python
Mason Moore	Configuration (Version Control) Manager	505-365-9365	Python, HTML

ROLE DESCRIPTIONS

Calculator	Version: <1.0>
Software Development Plan	Date: <26/SEPT/24>
01	

Project Manager: Keeps track of the project schedule, assigns tasks, and makes sure everyone meets deadlines. They also handle any issues that come up.

Scrum Master (especially during coding): Helps the team follow Agile practices, organizes daily check-ins, and plans work sessions.

Technical Lead: Offers technical advice and ensures the team follows good coding practices. They also help solve any technical problems.

Quality Assurance (QA) Lead: Makes sure the project artifacts meet quality standards. During coding, they plan and run tests to find and fix bugs.

UI/UX Designer: Designs the user interface, especially if a GUI interface is planned. They create sketches and prototypes to ensure the final product is easy to use.

Configuration (Version Control) Manager: Manages the project's version control system. They track changes, manage updates, and ensure everyone is working with the correct version of the project

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

4. Management Process

4.1 Iteration Objectives

Here is our [iteration plan](#).

4.1.1 Releases

Date released	Brief description of the change
	Basic parsing calculator. This will be the most basic form and first release of the calculator. It will be able to handle simple calculations but nothing more. This release will be a demo
	Add support for operator precedence (PEMDAS) and expressions with parentheses and additional operators. This release is a beta because it will have most of our features but not all of them.
	Improve the robustness of error handling, design a command-line interface for users, and add more calculator features. This will be a beta because it will be an almost finished product and is ready for user testing.
	Will focus on debugging and finalizing the code. Will test the code and make sure it's perfect and works as it supposed it and move the code over

Calculator	Version: <1.0>
Software Development Plan	Date: <26/SEPT/24>
01	

	github.

4.1.2 Project Schedule

Date	What needs to be done- iterations and phases, release points, demos, and other milestones	Target date for Completion
Software Requirements Specification	Determine the requirements needed for the project, figure out everyone's roles, introduce the functional requirements needed	10/01/2024
Design	Determine overall structure, use cases, and create models of how program will function	10/14/2024
Construction	Code various use cases using C and create UI to garner user input	11/15/24
Testing	Test code for defects while making sure overall functionality is not lost	12/01/24

4.2 Project Monitoring and Control

- **Requirements Management:** We will track and control the product requirements using a shared project management tool to log all feature requests, changes, and updates. All team members will update the requirements status regularly, and any changes will require approval from the project lead. Weekly meetings will review the status and impact of changes to ensure the requirements are aligned with the project's goals and timelines.
- **Quality Control:** The quality of the calculator deliverables will be controlled through regular code reviews and testing. We will use metrics such as efficiency, and accuracy of the calculator's mathematical functions (addition, subtraction, multiplication, division, etc.). Peer review will be conducted before each significant milestone. The final evaluation will focus on UI/UX usability and performance benchmarks to ensure the calculator works smoothly and delivers the expected results.
- **Risk Management:** We will identify risks (Incorrect Calculations, Missed deadlines, or Miscommunication) during the planning phase. These Risks will be analyzed by their likelihood and impact on the project. Risks with a higher priority will require more attention and plans to mitigate their likelihood such as quality assurance checking, additional testing, and backup team members.
- **Configuration Management:** Our version control system (Git) will be used to manage all project artifacts, including the calculator source code, documentation, test scripts, and results. Each version will be tagged and named according to project milestones (e.g., V1.0 for the initial

Calculator	Version: <1.0>
Software Development Plan	Date: <26/SEPT/24>
01	

functional version, V1.1 for bug fixes, etc.). The project repository will include backups, and we will follow a disaster recovery plan by keeping copies of the latest version in the cloud. All changes will undergo a pull request review process before being merged into the main branch to ensure stability.

4.3 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 the review that are not corrected before releasing for integration must be captured as Change Requests so that they are not forgotten.

4.4 Risk Management

Risks will be identified in the 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.

4.5 Configuration Management

Appropriate tools will be selected to 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, such as design documentation, is also included in the baseline. All customer deliverable artifacts, including executables, are included in the final baseline of the iteration.

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.

5. Annexes

The project will follow the UPEDU process.

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