<Company Name>

<Calculator>
Software Development Plan

Version <1.1>

Revision History

Date	Version	Description	Author
<09/12/23>	<1.0>	<Initial additions to the outline>	<Xavier Ruyle>

Table of Contents

Software Development Plan

1. Introduction

A calculator is essential for any task that involves arithmetic. It is the goal of this development team to create a calculator that is able to handle basic expressions. The team will create an algorithm to convert an arithmetical expression passed in by a human to RPN (Reverse Polish Notation) which is also understood as PN (Postfix Notation). The PN will then be sent through a stack, where the expression will be calculated while following correct PEMDAS rules.

1.1 Purpose

The purpose of the Software Development Plan is to layout basic information that is vital to the project. It will guide the team members to correctly carry out the necessary tasks to create a basic working calculator. The plan will not only deal with technical issues relating to the creation of the calculator but also deal with the planning process and resources needed in order to successfully complete the project.

The following people use the Software Development Plan:

- **Project team members** use it to obtain a grasp on the algorithms and data structures that will be used to create the calculator
- The **Project Leader** will use it to make sure deadlines and requirements are met which encompass the entire project.

1.2 Scope

This Software Development plan creates the framework for the development and testing process for the creation of a basic calculator. It contains the details regarding the plan to implement and code the calculator as well as the necessary dependencies and knowledge needed to create it.

1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.4 References

Any references will be found in the Appendix.

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview — Gives an overview of the calculator project’s purpose, scope, and goals. It also defines the features and functions that calculator will have.

Project Organization — Gives a description of the team’s organizational practices and makeup.

Management Process — Gives the time schedule for different deliverables, as well as the cost and different goals for the project as well as how the project’s progression will be analyzed.

Applicable Plans and Guidelines — Gives an overview of the project's software development process, including the algorithms, tools, and best practices it will follow.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The purpose of this project is to create a calculator which can parse and calculate expressions with the correct order of operations (PEMDAS).

2.2 Assumptions and Constraints

It is assumed that the project will be made with C++ and must involve the command line for user input.

2.3 Project Deliverables

- A working calculator
    - An algorithm that converts an expression to postfix notation
    - An algorithm that converts postfix notation to a solution
    - An interface that allows the user to input an expression and receive an output

2.4 Evolution of the Software Development Plan

The *Software Development Plan* will be revised with each iteration phase.

3. Project Organization

3.1 Organizational Structure

The project team involves a product leader who deals with interaction between all of the team members. They make sure that deadlines are met, team meetings are organized, deliverables are submitted, and are the final authority.

The product owner creates features, user stories, manages the git repository, and works on project deliverables.

The quality assurance member is responsible for making sure the project deliverables meet the requirements, testing, and bug reporting.

Finally, the project lead works on project deliverables, documentation, and reports any potential features or requirements that could be added to the project.

3.2 External Interfaces

N/A

3.3 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Michael Hoopes	Product Leader
Xavier Ruyle Connor Williamson	Product Owner Product Lead

Daniel Butler	Quality Assurance

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

N/A

4.2 Project Plan

4.2.1 Phase Plan

N/A

4.2.2 Iteration Objectives

1) Project Start
    - a. Team roles
    - b. Create Github repository
2) Requirements
    - a. Github repository able to be accessed publicly
    - b. C++ development environment setup
3) Coding/Design
    - a. Python representation/rough draft
    - b. Algorithm for parsing expression to postfix
    - c. Postfix to calculation using stack
4) Testing
    - a. Error checking
    - b. Bug checks
5) Release and deployment
    - a. End user experience
    - b. Good quality of life experience for end user

4.2.3 Releases

Alpha: Python implementation of the calculator – a basic outline/rough draft of the design
Beta: C++ implementation of the data structures and algorithms used in the python version

Demo: C++ version with user interface and error checking

Release: Final version of the calculator ready for release

4.2.4 Project Schedule

Alpha: September 29

Beta: November 1

Demo: November 21

Release: November 30

4.2.5 Project Resourcing

N/A

4.3 Project Monitoring and Control

4.4 Requirements Management

N/A

4.5 Quality Control

Bugs/Defects will be reported by the quality assurance team member. The Project lead will give feedback on any features or requirements that many need to be added to the project. Testing will provide the Project Leader with any more defects that will arise.

4.6 Reporting and Measurement

N/A

4.7 Risk Management

Risks will be outlined in meetings. Mitigation of said risks will be monitored by the quality assurance team member or reported by the Project Lead. Risks will be documented in this document should they arise.

4.8 Configuration Management

All changes or potential changes to the project will first go through communication with the team. Project artifacts will be version controlled using Git and the Github repository.

5. Annexes

The project will follow the UPEDU process.

The github repository: https://github.com/hmhoopes/348_project