Apex Software

Tech Calculator
Software Development Plan
Version <1.0>

[Note: The following template is provided for use with the Unified Process for EDUcation. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.] Marked (shaded) areas: items that are OK to leave out.

Revision History

Date	Version	Description	Author
09/27/2024	1.0	Initial Project Plan	Gage Weaver

Table of Contents

Software Development Plan

1. Introduction

The purpose of this project is to make a working calculator that can parse expressions and provide accurate results. The tasks to make the calculator will include functions for adding, subtracting, dividing, modulo, exponents, and multiplying, along with expression parsing and an output

1.1 Purpose

The purpose of the software development plan is to outline the process for making the calculator app and to document progress throughout planning, development, and testing.

1.2 Scope

This plan will be used to develop the Tech Calculator, and to keep track of progress throughout.

1.3 Definitions, Acronyms, and Abbreviations

N/A No definitions, acronyms, or abbreviations needed

1.4 References

N/A

1.5 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.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

*This project aims to develop an arithmetic expression parser. This program will be capable of evaluating and parsing arithmetic expressions +, -, *, /, %, and **. This will have to be done along with correctly interpreting parenthesis.*

2.2 Assumptions and Constraints

Assumptions:

- *The program will need to be developed in C++*
- *7 team members will work consistently in their roles*

Constraints:

- *Group meetings will need to be scheduled around classes and team members jobs*
- *The project will need to be completed by the specified date in the course syllabus*

2.3 Project Deliverables

This project will include a calculator that is user-friendly. This is done with C++ code and some frontend. 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*.

2.4 Evolution of the Software Development Plan

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

Revision Number:	Iteration Phase	Summary of changes
1.0	Project Planning	Initial filling out of requirements

3. Project Organization

3.1 Organizational Structure

Team Leader / Project Manager (Gage Weaver)

Change Control Manager (Matt Nash)

Analyst (Jal Maru)

Backend Developer (Aryan Ghorpade)

Integrator (Shayton Wright)

Tester/Reviewer (Abel Herzberg)

3.2 External Interfaces

N/A

3.3 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**	**Contact**	**Expertise**
Gage Weaver	Team Leader / Project Manager	Gage_Weaver@ku.edu	Backend, collab work
Matt Nash	Change Control Manager	nash.matthew@ku.edu	Testing, documentation, debugging
Jal Maru	Analyst	jalmaru@ku.edu	QA, Python
Aryan Ghorpade	Implementer	aryanghorpade@ku.edu	Backend, OOP
Shayton Wright	Integrator	shayton23@ku.edu	Coordination, planning
Abel Herzberg	Tester/Reviewer	abel.herzberg@ku.edu	Backend, collab work
Delaney Gray	Designer	delaney.gray@ku.edu	UI Design/Frontend

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

N/A

4.2 Project Plan

Goal	*Date To Be Completed By*
Project Management Plan	*9/29/2024*
Project Requirements	*10/13/2024*
Project Architecture and Design	*11/10/2024*
Project Implementation	*12/12/2024*
Test Cases	*12/12/2024*
User Manual	*12/12/2024*

4.2.1 Phase Plan

N/A

4.2.2 Iteration Objectives

We can start by creating the backend functions of the calculator, and a disconnected frontend. Future releases will focus on linking these functionalities together and getting rid of bugs. Basic operation of the calculator will build the foundation of our program before implementing more complex functionality such as PEMDAS.

4.2.3 Releases

The Final Release date will be December 12 2024, the project will not be public before then.

4.2.4 Project Schedule

Goal	*Date To Be Completed By*
Project Management Plan	*9/29/2024*
Project Requirements	*10/13/2024*
Project Architecture and Design	*11/10/2024*
Project Implementation	*12/12/2024*
Test Cases	*12/12/2024*
User Manual	*12/12/2024*

4.2.5 Project Resourcing

N/A

4.3 Project Monitoring and Control

Requirements Management: In order to make sure all requirements are met, code will be tested and analyzed frequently. If requirements are updated or changed, code will be changed and updated accordingly, with documentation to match.

Quality Control: Tester/reviewer will ensure the functioning of the project at the end of each deliverable through debugging and preventative maintenance.

Reporting and Measurement: Every meeting will be logged, along with progress updates based on the timeline.

Risk Management: The tester/reviewer will look for any potential risks associated with the software before approving pushes to the repository.

Configuration Management: Changes to the project are submitted through pull requests in GitHub. These will be approved by the tester/reviewer. These are tracked by GitHub automatically in the repository. GitHub also automatically retains and backs-up previous versions with time and user logs. This will provide sufficient coverage in the case of a mistake in the updating, and reverting to old code will be simple.

4.4 Requirements Management

N/A

4.5 Quality Control

The quality control will be handled by the reviewer/tester as outlined in the group member roles table aforementioned in the document. Any issues will be brought up in the next group meeting after they are found. Issues will be resolved according to preferences by those higher up in organizational structure.

4.6 Reporting and Measurement

N/A

4.7 Risk Management

There is a risk that files can be accidentally deleted/corrupted while working, so project backups will be saved to the github repository. Change control will ensure these backups are up-to-date.

4.8 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.

5. Annexes

The project will follow the UPEDU process.