Freaky Inc.

Freaky 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
<20/09/24>	<1.0>	<details>	<name>

Table of Contents *[keep this; say N/A when inapplicable]*

Software Development Plan

1. Introduction

For this project, we are creating a calculator software application which takes an input from a user, then uses the PEMDAS rules to compile it into a process that the computer will evaluate. The calculator should evaluate each function in PEMDAS order. The main file will include a file which contains our functions that can be called on in main. The application's purpose is to have a smooth running and working calculator a user could use. The scope of the project includes our main file which acts as the UX/UI where the user will add their inputs. Once the user's input is taken, the data is sent to another file to clean and insert into the functions of PEMDAS. We've deployed roles for each member of the group to ensure each part of the project is traceable to the origin requirements and consistency throughout the project timeline.

1.1 Purpose

The Software Development Plan gathers 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 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.

The Software Development plan will aquire input from stakeholders, team members, users, etc. to paint a larger picture of the development process. Completing this first step will help distinguish our project domain and requirements. The Software Development plan will ensure each step we take in the development process in orderly, seamless, and consistent. This is essentially the backbone to our project where ideas, designs, features, and documentation illustrate each goal and requirements for each portion of the project.

1.2 Scope

The scope of our Software Development Plan extensively provides the iterations and updates for each goal. Our first portion will be writing the skeleton of the software with our two files(main and functions), main containing simple input section at first to ensure our functions are working properly. For the functions, we will create a class which contains all of our PEMDAS operations. Each operation will be worked on one at a time to ensure each feature is working as intended before moving on. Once every function is complete for our functions file, we will make perfective maintenance to clean up our input labels and UX/UI design.

1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary:

Project Manager – Team member who plans, leads, and organizes the teams' responsibilities.

Quality Assurance Lead – Looks into the project to ensure that quality standards are met. Develops, implements, and manages test plans.

Configuration (Version Control) Manager - Tracks and manages changes to software code. Must monitor where errors occur and look at previous code if necessary to find bugs.

Scrum Master - Ensures that the team is productive, coordinates the daily scrum. Makes sure cooperation across all roles and functions is being achieved. Removes barriers that prevent team from being effective.

UI/UX Designer – Develops the frontend (User Interface) portion of the project makes sure project is visually appealing.

Technical Lead - Helps provide ideas for the technical side of the project. Makes decisions regarding the technical aspect of the project.

IDE - Integrated Development Environment

Figma – A UX/UI tool used by software developers to create designs/blueprints to aid with the coding stages.

1.4 References

[This subsection provides a complete list of all documents referenced elsewhere in the ***Software Development Plan****. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.*

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

- *Iteration Plans*
- *Development Case*
- *Vision [you may prepare a vision statement of your own: what your vision for the project is]*
- *Glossary*
- *Any other supporting plans or documentation.*

- Purpose
- Requirements
- Iteration Plans
- Vision
- Design
- Glossary
- Meeting Journal

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview – Declares an extensive description of everything the project deals with along with what the project is, time-constrains, goals, and other valuable information.

Project Organization – Acting as our origin, our project organization document is going to clearly and elegently express the structure of the project so we can trace back to the document whenever needed.

Meeting Journal – Essentially our daily scrum meeting, every week we meet up either in person or through zoom to detail what our next steps are and how we can improve productivity. The journal will consists of meeting dates along with the information discussed.

Team Member Roles – A list of roles divided into sections to ensure everyone is on look out for their section of the project but also helping other team members to complete the group work.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The purpose of this project is to allow students to have a well-rounded idea of the software development process and how to implement this process while working with a team. Some of the deliverables this project is expected to deliver is the correct numerical outcome from the input of a calculator following the PEMDAS.

2.2 Assumptions and Constraints

We have the team to work on the coding and designing for the project, but realistically some of the constraints for this project is the materials needed to build the hardware of the calculator. Furthermore, we must assume that our team has enough money to afford the resources and materials needed to build the project.

2.3 Project Deliverables

Deliverables/Artifacts:

- Makefile
- Source Code and Assembly Code
- Test cases and project reports
- Meeting Jornal
- Design Document(UX/UI)

Target Delivery Dates:

Our target delivery dates will occur after each sprint is completed. Time to complete each sprint should be around 2-7 days with our final sprint to perfect the software lasting a total of 14 days at maximum.

Task	Start Day	End Day	Duration(days)
Documentation/Requirements	10/1/2024	10/8/2024	7
Design/Planning	10/8/2024	10/15/2024	7
Setup Skeleton of Project	10/22/2024	10/29/2024	7
Refine () Function	10/29/2024	11/6/2024	7
Refine ^ Function	11/6/2024	11/13/2024	7
Refine * Function	11/13/2024	11/20/2024	7
Refine / Function	11/20/2024	11/27/2024	7
Refine + Function	11/27/2024	12/3/2024	7
Refine - Function	12/3/2024	12/10/2024	7
Refine Front-End	12/10/2024	12/24/2024	14

Semi-accurate start and end dates above to illustrate the duration each sprint will be.

2.4 Evolution of the Software Development Plan

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

Once we finish all our documentation, including what the project is, scope, constraints, etc., we can start working on the design aspect. Our first revisions will start from there as we develop a blueprint of how each function will look on the calculator along with a diagram to portray the different functions and their structure within our files.

3. Project Organization

3.1 Organizational Structure

The overall structure of our team is very loose, we have our specific roles, but everyone is expected to contribute to each part of the project. The main purpose of our roles is to have points of the project where certain people are doing a larger portion of the project. We are all expected to keep each other on track. Additionally, our sprint durations and mini goals will allow us to keep account of timing in case we are completing the project sooner or later than expected. These will prove vital in the long run when we start getting further into designing and coding where having a simplistic guide aids any team member who is questioning where we're at in the project.

3.2 External Interfaces

Our external interfaces will consists of stakeholders, this can be anyone who influences our project, team members, users, and other actors. The main point of interaction these interfaces will face is our UX and UI which is pivotal to provide a smooth and seamless user experience. To help make this part of the project as best as possible, coordination between members, including feedback, is necessary.

3.3 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Sam Prestigiacomo	Project Manager
Raymond Hu	Quality Assurance Lead
Nick Grieco	Configuration (Version Control) Manager

Zach McCauley	Scrum Master
Brady Boedy	UI/UX Designer
Abhiroop Goel	Technical Lead

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

[Provide the estimated cost and schedule for the project, as well as the basis for those estimates, and the points and circumstances in the project when re-estimation will occur.]

The cost of the project will be $0. Timeline of the project should be around three months with meetings weekly. Creating designs and diagrams to illustrate the flow of our project should take a week. Setting up the skeleton of our code should last a few days. Once we get the skeleton in place, we'll iterate through each function first getting the start of each function with comments to explain what each function should do. This should take around a week to setup our function but after, we will refine each function to ensure they work as intended. Time to complete each function should be around 7 days.

4.2 Project Plan

[This section contains the schedule and resources for the project.]Project artifact as well as iteration schedules]

Requirements - A set of activities that identify, document, and communicate the purpose of the PEMDAS calculator. Our project organization goes further into depth, examining the main mini achievements we must make to reach the release of our project on time.

Design - Transitioning the idea from requirements to a coding format to create the PEMDAS calculator. Designs will be created with Figma to get a satisfying looking design which is sleek but also useable. This ensures that the design doesn't just look good to users but is also functional.

Coding – A calculator that performs PEMDAS. We'll have two files which contain our main file which takes inputs and prints outputs. Our other file will contain the skeleton of our PEMDAS operation, linked to our main file.

Testing – Runs the code and test to make sure it's running properly

4.2.1 Iteration Objectives

Documentation and desings are necessary to complete all the later stages of the development plan so we spend a good amount of time at the start completing this. Each iteration of our project will focus on a different operation of the calculator. We'll start with creating the skeleton of each function, then we'll move from function to function ensuring each one works as intended before moving on to the next. After the operations, refining our main file which acts as our UX/UI to perfect everything.

4.2.2 Releases

[A brief description of each software release and whether it's demo, beta, and so on.]

Each software release will be a small section of PEMDAS, and we will demo each section to a specific group of people ready to demonstrate our product. Testing each portion of the project is important to ensure no bugs/errors are in the code after production. Finally, running each iteration through a demo group we can move on into releasing that portion of code into beta.

4.2.3 *Project Schedule*

[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.] [Limit to major project milestone, e.g., requirements, design, implementaiotn, and testing]

Task	Start Day	End Day	Duration(days)
Documentation/Requirements	10/1/2024	10/8/2024	7
Design/Planning	10/8/2024	10/15/2024	7
Setup Skeleton of Project	10/22/2024	10/29/2024	7
Refine () Function	10/29/2024	11/6/2024	7
Refine ^ Function	11/6/2024	11/13/2024	7
Refine * Function	11/13/2024	11/20/2024	7
Refine / Function	11/20/2024	11/27/2024	7
Refine + Function	11/27/2024	12/3/2024	7
Refine - Function	12/3/2024	12/10/2024	7
Refine Front-End	12/10/2024	12/24/2024	14

4.3 Project Monitoring and Control

Items to consider

- Requirements Management: Ensuring each step in our development plan is for a purpose and is vividly documented to align with further work done down the road.
- Quality Control: Confirm everything from documentation to our final product is exactly the way it was planned. If issues or complications arise, discussions to fix and revise will be made.
- Risk Management: Being able to predict time cycles of each portion and additionally the level of intensity will give us a better perspective of the problem at hand. Decisions to solve various levels of risks will be made to prevent complications down the line.
- Configuration Management: Setting up our structure of the code and files is vital to how our software runs. Our two files will interact with each other to produce the final product. Each section will be worked on until completion, after which, different team members will test. Documentation will simply be named for what it is(ex. Meeting Journal), backup plans for when issues arrive will be quickly resolved via discussion and research to adjust accordingly.

4.4 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.

4.5 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. Obviously, we're going to be reviewing all our code and ensuring everything is complete before passing it off to the final product. We'll structure and organize our issues as complete as possible to ensure no bad code is pushed to production.

4.6 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

*[Additional material of use to the reader of the **Software Development Plan**. Reference or include any project technical standards and plans which apply to this project. This typically includes the Programming Guidelines, Design Guidelines, and other process guidelines. The text that follows is provided as an example.]*

The project will follow the UPEDU process. Other applicable process plans are listed in the references section, including Programming Guidelines.