17 Co.

17 Co. Calculator
Software Development Plan
Version 1.0

Revision History

Date	Version	Description	Author
24/Sep/23	1.0	Initial document creation	17 Co. Team

Table of Contents

Software Development Plan

1. Introduction

The *Software Development Plan* is a guide for every team member working on *17 Co. Calculator* to help guide all phases of development including requirements engineering, design, development, and testing. The development plan should have the highest-level, most-abstract view of the product and process in its contents.

More detailed views of specifics processes are to be actively developed and refined as phases such as requirements engineering and design take place.

Since the project is being developed and broken up into phases, various sections are managed by various people. Each section of this document is to be written by the lead that most closely corresponds to that section to help establish boundaries between phases and reduce the chance of redundant work being completed. Each section is to be reviewed by the Team Administrator to ensure consistency, efficiency, and effectiveness.

This plan is also meant to serve as instruction on how the various phases of development should be completed. The document is not intended to be the answer to every question, instead extended documentation will be referenced within this plan that, throughout development, will be created and developed as the project progresses. Those documents can be found in section 1.4 entitled “References.”

1.1 Purpose

The purpose of this *Software Development Plan* is to create a central source for the general overview of the project, to reduce miscommunications, and to create a more cohesive environment throughout the creation of *17 Co. Calculator*.

The document can be used by a variety of people for a variety of reasons, including but not limited to:

- The **Team Administrator** uses the document to establish meeting agendas, ensure timely completion of requirements, and properly manage the project over the course of several months.
- The various **area leads** use the document to establish boundaries between the various phases of development and roles, ensuring no unnecessary overlapping work; therefore, increasing overall efficiency.
- In addition, **team members** use this document to ensure tasks are completed and processes are followed with respect to their assigned roles.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the *17 Co. 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* to be created by the Requirements and Quality lead.

1.3 Definitions, Acronyms, and Abbreviations

See the *Project Glossary*.

1.4 References

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

- Iteration Plans *(to be developed at the start of each phase)*
- Vision (*to be completed during Requirements Engineering phase)*
- Glossary *(to be created in Phase 2 and maintained throughout the project)*
- Meeting Logs (*to be completed incrementally as meetings are held*)
- Risk List Documents (*to be completed at the start of each Phase)*
- Configuration Management Plan (*to be completed by the technical lead before the start of Phase 4)*

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
    - Provided 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

The purpose and objectives of the project is to develop a calculator in C++ that can interpret and calculate complex math problems, correctly implementing fundamental rules of mathematics (such as the proper order of operations and exponent rules) while being intuitive and simple for the user.

2.2 Assumptions and Constraints

As college students each with our own individual engineering schedules, schedules may be a constraint, especially for meetings during the week. Because of this, we will have to all work around a consistent weekend schedule for meetings and project additions.

As we are working on team documentation through global workspaces, our location for meetings is also a constraint that we must be aware of. Not only will the location need to be accommodating of multiple computers that need to be charged but also have sufficient Wi-Fi to meet the needs of global documentation as so our progress is not limited by our location.

2.3 Project Deliverables

Deliverables for each project phase are identified in the within the appropriate requirements engineering and class-provided documents. Deliverables are delivered towards the end of the iteration, as specified in section *4.2.4 Project Schedule*. The final project is aiming to be delivered by the first week of December.

2.4 Evolution of the Software Development Plan

The *Software Development Plan* will be revised prior to the start of each Iteration phase. More details can be found in section 4.2. When a major version change occurs, the second number in the version number will increase (v1.0 -> v1.1).

Changes can also occur when needed in between or in the middle of phases as changes need to be made to the overall development process. For these changes, the third number in the version will be increased (v1.0 -> v1.0.1)

3. Project Organization

3.1 Organizational Structure

The team will be led by a Team Administrator which will be assisted in their duties by an Assistant Team Administrator. There will be a Requirements and Quality Lead who will be responsible for the elicitation and documentation of requirements and the assurance of project quality and conformity to requirements. The Design Lead will develop a process and plan for project execution and completion and the Technical Lead will lead the actual implementation of that process and plan. Any team member may assist and be utilized by any other member in the completion of their duties.

3.2 External Interfaces

N/A

3.3 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Michael Stang Contact Info: 520-279-6603 michael.stang@ku.edu Availability: Mon, Fri, Sat, and Sun All Day Wed After 4pm Expertise: Leadership Communication Time Management Programming	**Team Administrator** • Organizing & leading team meetings • Establishing agendas & setting internal deadlines • Ensuring proper communication between roles • Coordinating with stakeholders • General team communication with outside entities, in particular in regard to internal relations • Assisting where needed
Jack Bauer Contact Info: 314-221-7656 jack.bauer@ku.edu Availability: Mon after 3pm Wed after 3pm Fri, Sat, Sun all day	**Assistant Team Administrator** • Recording team meetings & taking and posting minutes • Assisting the team administrator with misc tasks • Filling in for the team administrator in extenuating circumstances • Managing the GitHub repository and

Expertise: Logistics & Organization Documentation C++ & Microsoft Office	ensuring completion of deliverables • Assisting where needed
Holden Vail Contact Info: 620-515-4501 h190v184@ku.edu Availability: Mon, Wed, Fri: After 8pm Tues, Thurs: After 4pm Sat, Sun: After 8pm Expertise: Programing Communication Software Engineering	**Requirements and Quality Lead** • Lead the implementation of the Design Lead's requirements • Assign technical tasks to team members but be ultimately responsible for their completion • Assisting where needed
Logan Smith Contact Info: 331-213-6811 l500s632@ku.edu Availability: Wed, Fri-Mon Expertise: Programming Design Problem Solving Communication	**Design Lead** • Lead the creation of design deliverables and design requirements • Ensure technical implementations are completed in accordance with design requirements • Assisting where needed
Blake Carlson Contact Info: 815-277-7415 blake.carlson@ku.edu Availability: Wed, Fri-Sun Expertise: Object-Oriented Programming Project Management	**Technical Lead** • Lead the Requirement Engineering phase of the project • Ensure quality and completion of design and implementation in accordance with designated requirements • Ensure tasks are completed by deadlines and follow up with team members to minimize blockers • Assisting where needed

Anyone on the project can fill in and perform the tasks of the other roles.

4. Management Process

4.1 Project Estimates

N/A

4.2 Project Plan

This section contains the schedule and resources for the project.

4.2.1 Phase Plan

N/A

4.2.2 Iteration Objectives

The scope of the iteration plans is to be explained in more detail in the plan documents and refined as development commences.

4.2.3 Releases

Periodically throughout development, at the end of the corresponding phase, there will be different releases product releases of varying types. The releases and their types are as follows:

- Demo 1 will be an un-implemented mockup used to show how the software will function when it is implemented.
- Alpha 1 will be the first release of the software without full functionality.
- Beta 1 will be a fully featured but unrefined release of the product.
- Final Release will be a fully functional and well-refined release of the product to the public.

4.2.4 Project Schedule

The phases of the project are as follows, based upon given stakeholder deadlines:

Week (Tuesday class date – Thursday class date)	Phase End	Phase Begin	Associated Release	Milestones
3 (9/5 - 9/7)	N/A	1	N/A	Project Management Plan begins development
5 (9/19 - 9/21)	1	2	N/A	Requirements Engineering process begins
10 (10/24 - 10/26)	2	3	N/A	Project Architecture and Design begins
11 (10/31 – 11/2)	3	4	Demo 1	Project Implementation begins
13 (11/14 – 11/16)	4	5	Alpha 1	Project Testing begins
15 (11/28 – 11/30)	5	6	Beta 1	Project User Manual begins development
16 (12/5 – 12/7)	6	N/A	Final Release	Project Completion due

Phases 2-6 will include multiple iterations to be developed and scheduled at the start of each phase.

4.2.5 Project Resourcing

N/A

4.3 Project Monitoring and Control

- *Requirements Management:* Quality will be ensured through systematic check-ins with the Requirements and Quality Lead for all other team members. When contributing new code, it will be necessary to ensure resiliency and compatibility. Additionally, the management of the all the changes will be handled via git to ensure the project requirements are met.

- *Quality Control:* This project is going to be split into six distinct phases and will be developed using an agile development model. Meaning the phases are both iterative and incremental. During each of these phases we will first develop a list of our backlogged tasks. Then, once the tasks are delegated and they begin to be completed the completed tasks will be tested for resiliency and either logged as completed or rejected for review or correction. Additionally, at the end of each sprint all the tasks both previous and current will be reviewed for quality, resiliency, and compatibility.
- *Reporting and Measurement:* N/A
- *Risk Management:* Through proper documentation and extensive planning done by each lead of each phase, risks can be seen before any real development occurs, reducing the need for fixing and refactoring. One big way this can be expanded upon is by having both the Team Administrator and Assistant Team Administrator do check ins on the various phases to ensure there are no miscommunications and everything is being planned correctly.
- *Configuration Management:* Due to the nature of the team structure, each phase is handled by a different person. If a change is needed within a phase, the various people working on that phase will report to the lead in charge of said phase. However, if a change is needed across phases, the Team Administrator should be contacted, and the communication between phases should be facilitated by either the Team Administrator or Assistant Team Administrator.

4.4 Requirements Management

N/A

4.5 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, which is to be implemented by the current phases' lead and added to this section with the appropriate software development plan update. 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.

4.6 Reporting and Measurement

N/A

4.7 Risk Management

Risks will be identified at the start of each 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 Documents for each phase after Phase 2 for detailed information.

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.

Refer to the Configuration Management Plan 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.

This section is a major source of expansion in future versions of the document, including references to design, requirements, and programming documents.