**Rock Chalk Instruments**

**C++ Calculator**

**Software Development Plan**

**Version <1.0>**

**Revision History**

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 9/22/2023 | 1.0 | Beginning to outline project requirements and definitions | Owen Deines |
| 9/24/2023 | 1.0 | Defining and assigning roles  Completing Development Plan | Carson Abbott  Owen Deines  Brisa Andrade  Jordan Mcdaniels |
|  |  |  |  |
|  |  |  |  |

**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........................................................................................................................................................ 4*

**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 ........................................................................................................................................... 6**

*3.1 Organizational Structure ............................................................................................................................... 6 3.2 External Interfaces......................................................................................................................................... 6 3.3 Roles and Responsibilities............................................................................................................................. 6*

**4. Management Process........................................................................................................................................... 9**

*4.1 Project Estimates........................................................................................................................................... 9 4.2 Project Plan................................................................................................................................................... 10 4.3 Project Monitoring and Control.................................................................................................................... 10 4.4 Requirements Management............................................................................................................................ 10 4.5 Quality Control.............................................................................................................................................. 10 4.6 Reporting and Measurement.......................................................................................................................... 11 4.7 Risk Management........................................................................................................................................... 11 4.8 Configuration Management........................................................................................................................... 11*

**5. Annexes................................................................................................................................................................. 11**

**Software Development Plan**

**1. Introduction**

This document outlines our goals for the project and how we intend to work toward those goals. The purpose of this document is to serve as a reference point for team members as we work through the project. Information about our development methods, timeline, domain, and division of work will be contained within. Any terminology used throughout development will be defined here in case it needs to be referenced back to later. Contact information is also included in order to facilitate communication within the development team. This is a living document and is subject to change in the event that any part of it is not allowing us to fulfill the requirements of the project.

**1.1 Purpose**

The purpose of the *Software Development Plan* is to gather 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 **Team Leader and Assistant Team Leader** uses it to plan the project schedule and resource needs, and to track progress against the schedule.

• The **Technical Lead** uses it to understand the project schedule and the iteration goals and make sure the Developers are on track to completing deliverables

• The **Developers** use it to understand the Functional Requirements and Design Specifications needed to complete the project

• The **Quality Assurance Engineer** uses it to understand the various test cases that will be used to validate the calculator’s functionality.

**1.2 Scope**

This *Software Development Plan* describes the overall plan to be used by the C++ Calculator project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans. Purpose, Scope, and Objectives are outlined in section 2.1

**1.3 Definitions, Acronyms, and Abbreviations**

DE - Domain Engineering

VCS - Version Control System

GH - GitHub

**1.4 References**

No current references

**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**

* The purpose of this project is to create a calculator in C++
* The calculator will be able to process and utilize the +, -, \*, /, %, and ^ operators to create outputs from user inputs
* The project will be expected to deliver a working C++ script that implements the functionality detailed above

**2.2 Assumptions and Constraints**

* Final product will meet requirements outlined in project assignment including:
  + Being written in C++
  + Implement the necessary arithmetic functionality
  + Be submitted by the deadline
* Development will need to be scheduled to work around team member’s schedules
* Team members will have access to necessary Linux lab machines
* Final product will be compatible with Lab machines
* Documents and code will be stored in GH repository

**2.3 Project Deliverables**

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*.

* Requirements
  + Expression Parsing
    - Take operator precedence and parentheses into account
  + Operator Support
    - + (addition), - (subtraction), \* (multiplication), / (division), % (modulo), ^ (exponentiation)
  + Parenthesis Handling
    - Determining the order of evaluation
  + Numeric Constants
    - Recognize and calculate numeric constants within the expression
* Design Specifications
  + User-friendly Command-Line interface
    - Allows user to input expressions and get the calculated results as output
  + Error Handling
    - Managing scenarios like division by zero or invalid expressions
* Test Cases
  + To test the validity of our program, we will utilize the “Examples of Valid and Invalid Expressions” section outlined in the 00-Project-Description.pdf on Canvas.
* Code
  + Will be written in C++

**2.4 Evolution of the Software Development Plan**

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

**3. Project Organization**

**3.1 Organizational Structure**

The Technical Leader will schedule and delegate work to Developers. Developers will work on deliverables, such as Functional Requirements and Design Specifications, and push deliverables to the GH repository. The Quality Assurance Engineer will review the work of Developers, ensuring everything is functional, while reporting any issues to the Technical Lead. The Assistant Team Leader will check on the progress of deliverables with the Technical Lead and communicate that progress with the Team Leader. The Team Leader will schedule/manage meetings, take notes of said meetings, and track overall progress of the project.

**3.3 Roles and Responsibilities**

**Team Leader:** Carson Abbott

Contact: carson.abbott@ku.edu

Availability:

* Monday: 2:00 PM and later
* Tuesday: 10:00 AM - 12:00 PM, 7:00 PM and later
* Wednesday: 7:30 PM and later
* Thursday: 6:00 PM and later
* Friday: 6:00 PM and later
* Saturday: Anytime
* Sunday: Anytime

Expertise

* Major: Computer Science
* Relevant Coursework: EECS 140, 168, 210 (enrolled), 268, 348 (enrolled)
* Relevant Skills: Python, Java, some HTML and CSS

Responsible for:

* Setting up and managing team meetings
* Taking notes on what happens in meetings
* Maintaining and Updating various Project documents
* Handling Canvas submissions
* Bringing questions to the TA/Professor if needed

**Assistant Team Leader/Developer:** Owen Deines

Contact: owendeines150@ku.edu

Availability:

* Monday: 1-3pm
* Tuesday: after 4pm
* Wednesday: 1-3pm, after 8pm
* Thursday: after 6pm
* Friday: after 6pm
* Saturday: May be available all day or not at all
* Sunday: All day

Expertise

* Major: Computer Science
* Relevant Coursework: EECS 268, 210 (enrolled), 348 (enrolled)
* Relevant Skills: Python, C++, some HTML and CSS

Responsible for:

* Checking in on the progress of deliverables
* Maintaining and Updating various Project documents
* Implementing Requirements and Design Specifications
* Pushing completed code to the GH Repository
* Communicating deliverables’ progress to Technical Lead

**Technical Lead:** Brisa Andrade

Contact: bandrade@ku.edu

Availability:

* Monday: After 12:00 pm
* Tuesday: After 5:00 pm
* Wednesday: 11:00 am - 2:00 pm
* Thursday: After 5:00 pm
* Friday: After 4:00 pm
* Saturday: Unavailable
* Sunday: All Day

Expertise

* Major: Computer Science
* Relevant Coursework: EECS 168, EECS 268, EECS 330: Data Structures and Algorithms, EECS 388: Embedded Systems
* Relevant Skills: C, C++, C#,

Responsible for:

* Managing project code in the GH Repository
* Dividing/Delegating work to Developers
  + Can also assume the responsibilities of Developers
* Communicating with the Team Leader or Assistant Team Leader on deliverables’ progress

**Developer:** Jordan Mcdaniels

Contact: mcdanielsjordan32@ku.edu

Availability:

* Monday: After 5 Pm
* Tuesday: After 4
* Wednesday: Anytime
* Thursday: After 4
* Friday: After 6 (My lab time is 4 to 6)
* Saturday: Early- 7pm
* Sunday: 8 am - 1 pm

Expertise

* Major: Computer Science
* Relevant Coursework: 268, 388,
* Relevant Skills: Strongest python, C, Little HTML

Responsible for:

* Implementing Requirements and Design Specifications
* Pushing completed code to the GH Repository
* Communicating deliverables’ progress to Technical Lead

**Developer:** Jaret Priddy

Contact: jaret\_priddy@ku.edu

Availability:

* Monday: after 5pm
* Tuesday: after 5pm
* Wednesday: after 5pm
* Thursday: after 5pm
* Friday: after 5pm
* Saturday: All day
* Sunday: All day

Expertise

* Major: Computer Science
* Relevant Coursework: EECS 268, 388 (enrolled) 210 (enrolled)
* Relevant Skills: Python, C++, Limited Knowledge: C, HTML, JavaScript and CSS

Responsible for:

* Implementing Requirements and Design Specifications
* Pushing completed code to the GH Repository
* Communicating deliverables’ progress to Technical Lead

**Quality Assurance Engineer:** Connor Schroeder

Contact: Connors@ku.edu

Availability:

* Monday: after 12pm
* Tuesday: after 4pm
* Wednesday: after 12pm
* Thursday: after 4pm
* Friday: after 6pm
* Saturday: after 12pm
* Sunday: after 1pm

Expertise

* Major: Computer Science
* Relevant Coursework: EECS 140, 168, 210 (enrolled), 268, 348 (enrolled), 388 (enrolled), 468 (enrolled)
* Relevant Skills: Python, JavaScript, C, HTML

Responsible for:

* Making sure code is functional and without errors
* Reporting any issues to the Technical Lead
* Providing suggestions to the Technical Lead or Team Leader

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 Iteration Objectives*

Iteration 1: Create a plan for the project, determine requirements for the project, define and assign roles for the project.

Iteration 2: Outline code based on requirements for the project and determine if changes need to be made to the plan early in development

Iteration 3: Begin development of key features. Perform tests on individual features as they are completed. Key features to be included:

1. Production of a file that can take a user input of functions and place it into a stack (Include cases of bad operations)
2. Making the program follow PEMDAS and parenthesis rules using different cases
3. Implementation of the + and - Operators
4. Implementation of the \* and / Operators
5. Implementation of the % operation
6. Implementation of the ^ operation

Iteration 4: Combine code for all key features into complete program and test

Iteration 5: Complete testing and bug correction. Submit final product

*4.2.2 Releases*

*N/A*

*4.2.3 Project Schedule*

| **Iterations** | **Descriptions** | **Expected Completion Date** |
| --- | --- | --- |
| Iteration 1 | Completion of software development plan | 09/24/2023 |
| Iteration 2 | Creation of code outline | 10/08/23 |
| Iteration 3 | Code implementation | 10/22/23 |
| Iteration 4 | Assembling program | 11/12/23 |
| Iteration 5 | Program testing and bug correction | 11/19/23 |
| Project Completion | Submission of completed assignment | 12/03/23 |

*4.2.4 Project Resourcing*

C++ is the only skill required for completion of the project. Considering the size and scope of the project, extensive training in C++ doesn’t need to be completed. Team members who are not already familiar with C++ will simply become familiar with it as the project continues.

**4.3 Project Monitoring and Control**

The Technical Lead will report progress on Functional Requirements and Design Specifications to the Assistant Team Leader. Both the Assistant Team Leader and Team Leader will be responsible for managing these reports using a checklist. The Quality Assurance Engineer will also check off the functionality of these deliverables using the same checklist.

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, 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.

4.6 Reporting and Measurement

Updated schedule estimates, and metrics summary reports, will be generated at the end of each iteration.

The Minimal Set of Metrics, as described in the RUP Guidelines: Metrics will be gathered on a weekly basis. These include:

Earned value for completed tasks. This is used to re-estimate the schedule and budget for the remainder of the project, and/or to identify need for scope changes.

Total defects open and closed – shown as a trend graph. This is used to help estimate the effort remaining to correct defects.

Acceptance test cases passing – shown as a trend graph. This is used to demonstrate progress to stakeholders.

4.7 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.

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.

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