Voodoo Dough Boys

Arithmetic Expression Evaluator in C++
Software Development Plan

Version <1.0>

Revision History

Date	Version	Description	Author
16/02/2026	1.0	Created the name for our company and our project. Started general reformatting	All
17/02/2026	1.0	Filled out sections of the Project Overview and Project Organization section of the report.	Joseph Ubanda
18/02/2026	1.1	Filling out rough versions of the Software Development Plan and Project Overview	Vu Nguyen
18/02/2026	1.2	Revised the SDP and Project overview	Maddy Dang
19/2/26	1.3	Added more to the software development plan introduction, and purpose	Lanale garry
20/2/2026	1.4	Filled out sections of Project Estimates of the report	Tram Dinh

Table of Contents

Software Development Plan

1. Introduction

What you will see in our Software Development Plan is an outline of the steps we took to create a C++ program, “The Calculatorinator." It covers the purpose, scope, abbreviations, references, and an overview of this program.

1.1 Purpose

The purpose of the *Software Development Plan* is to gather all the 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 to 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.

1.2 Scope

This *Software Development Plan* describes the plan to be used by the *“Calculatorinator”* 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*.

1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.4 References

Project Outline:

EECS 348: Software Engineering Spring 2026 - Term Project in C++: Arithmetic Expression Evaluator, Professor Hossein Saiedian.

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 seeks to create a mathematical expression evaluator using C++. The program will have the functionalities of addition, subtraction, multiplication, division, modulus, and exponentiation. The program should be able to run on any input and tell the user what errors occur. The project should include a user-friendly graphical interface and intuitive use.

2.2 Assumptions and Constraints

The program should be created using C++ and be able to run without crashing given any input. The program should be able to handle parentheses, unary operators, and the correct order of evaluating expressions. The program should be able to handle numeric constants and give a correct result.

2.3 Project Deliverables

- Project Management Plan
- Requirements Document
- Design Document
- Program Test Cases
- Completed C++ program with documentation
- User manual README with examples

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. The changes should reflect the flow of the project and any changes that are discussed during the weekly Scrum meetings. Any member of the team is able to revise the plan.

3. Project Organization

3.1 Organizational Structure

The project will be modifiable by any member of the group, and they should all work towards the current deliverable of the sprint. The Scrum Master will help organize meetings for the team where everyone can discuss the flow of the project and which items to focus on.

3.2 Roles and Responsibilities

Person	**Availability and Contact Information**	**Unified Process for EDUcation Role**
Micah Lehman	Phone Number: 913-337-6753 Email: m3231924@ku.edu Availability: MF 10p-2p.	Any Role
Joseph Ubanda-Ruiz	Phone Number: 816-536-6585 Contact: jubanda@ku.edu Availability: M F: 1:00 - 5:00pm T: 5:00 - 6:00pm TH: 4:00 - 6:00pm	Developer
La`Nale Garry (la-nail)	Contact info: Email: l915g097@ku.edu Phone number: 913-429-1595 Availability: M T W TR F 5:00pm-12:00am	Developer
Tram Dinh	Phone number: (785) - 331 -8199 Email: tram.dinhnguyet@ku.edu	Any Role
Vu Nguyen	Phone number: 816-599-1562 Email: v212n917@home.ku.edu Availability: M T TR F 3:30pm-9pm	Scrum Master
Maddy Dang	Phone Number: 913-687-6806 Email: m825d168@ku.edu Availability: M W 11-1 pm	Any Role

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

Estimated effort per phase:

- Project Planning (Deliverable 1) – 1 week
- Requirements Engineering (Deliverable 2) – 3 weeks
- Architecture & Design (Deliverable 3) – 3 weeks
- Implementation & Unit Testing – 4 weeks
- Final Integration, Test Cases, and Documentation – 2 weeks

Re-estimation will occur:

- After completion of the Software Requirement Specification (March 15)
- After Architecture approval (April 5)

4.2 Project Plan

This section contains the schedule and resources for the project.

4.2.1 Iteration Objectives

- Software Development Plan
- Requirements Document
- Design Document
- Implementation
    - Expression Parsing
    - Correct PEMDAS Order Parsing
    - Parenthesis Handling
    - Unary Operators
    - Numeric Constants
    - Error Handling
- Test Cases
- Write Test Cases
- Create User Manual

4.2.2 Releases

All releases will be updated on the repository:

https://github.com/maddytd/The-Calculatorinator

Timeline of releases:

No releases as of V1.0

4.2.3 Project Schedule

- Software Development Plan Feb 15 - Feb 22
- Software Requirements Specification Feb 23 - March 15
- Software Architecture Document March 16 - April 5
- Implementation of Code April 5 - May 7
  - Expression Parsing TO BE SET
  - Correct PEMDAS Order Precedence TO BE SET
  - Parenthesis Handling TO BE SET
  - Numeric Constants TO BE SET
  - Unary Operators TO BE SET
  - Error Handling TO BE SET
- Write Test Cases April 5 - May 7
- Create User Manual April 5 - May 7

4.3 Project Monitoring and Control

- Requirements Management: Requests should be filed through Github to keep an up-to-date log of requested items.
- Quality Control: Every commit should be manually reviewed by another member to ensure that code will function once added to the main project. Github's commits will be used to track updates.
- Risk Management: Code that is waiting to be pushed should be organized by size and importance to make sure items are given the proper attention and check for any fatal errors. Github's commit tracker will be used to repair previous versions if code fails to function
- Configuration Management: Code should be reviewed manually before committing into the main function. Every commit should have a message that includes the purpose and what changed as well as a properly documented name for each commit that includes the name, which subfile, what version of code, and if it is a functional or non-functional requirement.

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 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.

Refer to the Project Measurements Document (AAA-BBB-X.Y.doc) for detailed information.

4.6 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.

Refer to the Risk List Document (CCC-DDD-X.Y.doc) for detailed information.

4.7 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 (EEE-FFF-X.Y.doc) 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.