Group 32

Arithmetic Expression Evaluator in C++

Software Development Plan

Version 1.1

Revision History

Date	Version	Description	Author
20/09/23	1.0	Opened the file and shared access with teammates Derek N. and Ryan G. Minor edits such as adding dates and the project name. Derek N. assigned Project Manager role.	Jordan B.
22/09/23	1.1	Edited various empty sections to include items such as a tentative schedule, a project overview, objectives, etc. Jordan Burns assigned Quality Control role.	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.
23/09/23	1.2	Editing Section 3.1, 1.4, 1.5, and 5 Defining team member roles. Quality control checks.	Derek N., Jordan B
24/09/23	1.3	Continuing quality control and filling in final undone portions.	Jordan B.

Table of Contents

Software Development Plan

●

1. Introduction

1.1 Purpose

The *Software Development Plan* document's purpose is to unify all group activities under a well-defined resource detailing sources, intention, roles, and approach to the "Arithmetic Expression Evaluator in C++" project. The project manager uses the document as a point of reference to instruct team members and ensure the project progresses smoothly. The team members use the document as a means of self-instruction and to understand the purpose and schedule of their work.

1.2 Scope

The *Software Development Plan* is the instructional first portion of a multi-deliverable project, culminating in the multi-function calculator program "Arithmetic Expression Evaluator in C++." The base of this *Software Development Plan* document is provided by the Project Plan document. Further instruction for the project overall is provided by the Project Description document. Both of these sources are defined in Section 1.4 (References.)

1.3 Definitions, Acronyms, and Abbreviations

Github	A cloud service for software development, and a tool for sharing versions of files between members of a team.
C++	A high level programming language, which the Arithmetic Expression Evaluator will be written in.
Arithmetic	Relating to operations on numbers. In this project, it refers to the abilities of the calculator.
Test case	Sample inputs for code to test for debugging purposes.
User Documentation	Comments in the code left behind by developers that allow users to understand each part's function and other developers to understand the logic behind each part's construction.
TA	A teaching assistant, a figure of authority and informative resource for the project.

1.4 References

- Team Github
  - Team member profile sheet(09/24/23, Group 32)
- All EECS 348 class materials as provided on Canvas
  - Project Description (09/07/23, Professor Hossein Saiedian, University of Kansas)
  - Project Plan (09/15/23, Professor Hossein Saiedian, University of Kansas)
  - EECS 348: Software Engineering Lecture Slides (08/21/23 – Current, Professor Hossein Saiedian, University of Kansas)

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 is to evaluate mathematical expressions, including handling operators and parentheses. Our objectives include developing a user-friendly interface, optimizing performance, and providing error handling. Deliverables will include the C++ source code, design, requirements, an executable application, a user manual, user documentation, finalized project plan, and test cases. This project will provide a tool for mathematical computations and educational use.

2.2 Assumptions and Constraints

This project must be done in C++, and therefore assumes that users and developers alike will have access to the necessary environments for constructing and executing C++ programs. The project also assumes that users will input valid arithmetic expressions using standard mathematical notations. It will not process non-standard symbols or notations. The project is constrained to one flow of execution, where functions are not used in parallel, but rather one at a time. Other constraints include the project timeline, availability of team members, and level of proficiency in the undergraduate group.

2.3 Project

1. Project Plan: Due by 9/24
2. Source Code Framework
3. Expression Evaluation
4. Test Cases
5. Design Specs
6. Requirements
7. Finalized Project Development Plan
8. C++ Source Code
9. UI Model
10. Error Handling Code
11. Testing
12. Full executable program
13. User Manual
14. User Documentation
15. Optional enhancements

With the exception of the *Software Development Plan* document and the finalized project, which have deadlines of 09/24/23 and the week of 12/05/23, respectively, no due dates are given. However, a schedule can be assumed around the dates that deliverables are assigned. This hypothetical schedule is detailed in Section 4.2.1, Phase Plan.

2.4 Evolution of the Software Development Plan

The assignment dates for each new project iteration are listed in Section 4.2.1, Phase Plan. The *Software Development Plan* is to be updated as the project progresses, and as such, will follow the (subject to change) schedule outlined in the Phase Plan. This section will be updated as the schedule is elaborated upon in the future.

Version	Date Issued	Author	Description

1.0	20/09/23	Jordan B.	FIle created, shared with teammates. Minor edits made. Project Manager role assigned to Derek N.
1.4	24/09/2023	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.	Completed Part 1: Project Management Plan Part 1 due
2.0	19/09/23	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.	Part 2: Project Requirements assigned
3.0	24/10/23	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.	Part 3: Project Architecture and Design assigned
4.0	31/10//23	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.	Part 4: Project Implementation assigned
5.0	14/11/23	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.	Part 5: Project Test Cases assigned
6.0	28/11/23	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.	Part 6: Project User Manual assigned
7.0	Week of 05/12/23	Derek N., Priyatam N., Jordan B., Manu R., Ryan G.	Final Project Implementation submission due

3. Project Organization

3.1 Organizational Structure

Project Manager: Derek Norton

- Plan project schedule
- Address resource needs
- Track progress against schedule
- Organizing team meetings
- Tracking meeting timeframes
- Organizing and conducting meetings

Deputy Administrator: Ryan Grimsley

- Supervises project engineers
- Logging team meeting details
- Identifying project requirements

Configuration Management Engineer: Manu Redd

- Monitor consistency of project
- Track changes in product development
- Review work and test developed code in conjunction with Quality Assurance Engineer

Quality Assurance Engineer: Jordan Burns

- Review work and test developed code in conjunction with Configuration Manager
- Reporting emerging issues during project development to the team
- Reporting any technical issues to professor

Project Lead 1: Priyatam Nuney

- Directing project development
- Leading software development phases
- Compiling project deliverables

3.2 External Interfaces

The only external groups Group 32 will be working with will be Professor Hossein Saiedian, who oversees all groups and the project itself, and the EECS 348 teaching assistants to deploy, review, and accept project instruction and deliverables.

3.3 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Derek N.	Project Manager
Jordan B.	Quality Control
Priyatam N.	Project Lead 1
Ryan G.	Deputy Administrator
Manu R.	Configuration Manager

4. Management Process

4.1 Project Estimates

The completed project has a deadline of the week of 12/05/23. As the Fall 2023 academic semester stops classes on 12/08/23, no extensions may be provided. However, if deemed necessary, Professor Hossein Saiedian, who is overseeing all groups, may adjust all other deadlines. With the exception of Part 1, no deadlines for each deliverable portion have been given. However, a general schedule outline can be seen in Section 4.2.

The project is expected to come at no cost (that is, $0.00) to any Group 32 member. Regarding class tuition and any funds spent on necessary materials (such as laptops for programming, cell phones for group communication, etc.,) as these have been pre-paid and are not used exclusively related to this project, or purchased in anticipation of this project, these costs will not be considered. If future assignments require any monetary spending on behalf of Group 32, it will be noted.

4.2 Project Plan

Segments of the project are assigned section by section. With the exception of Part 1, due 09/24/23, no due dates are provided. However, the assignment dates are listed on the EECS 348 Syllabus, and therefore due dates can be assumed around the time the next portion is assigned.

Part 1: Project Management Plan	Assigned week of 08/29/23, due 09/24/23
Part 2: Project Requirements	Assigned week of 09/19/23, due date unknown
Part 3: Project Architecture and Design	Assigned week of 10/24/23, due date unknown
Part 4: Project Implementation	Assigned week of 10/31/23, due date unknown
Part 5: Project Test Cases	Assigned week of 11/14/23, due date unknown
Part 6: Project User Manual	Assigned week of 11/28/23, due date unknown

According to the syllabus, the final project implementation must consist of all parts and deliverables, a finalized project plan, requirements, design, test cases, code, and a user manual. This is due the week of 12/05/23.

To assist in meeting this deadline and delivering all artifacts in an optimal state, members of Group 32 may at any time access the following resources:

- University of Kansas EECS staff
  - EECS 348 Professor Hossein Saiedian
  - EECS 348 teaching assistants
- All EECS 348 class materials as provided on Canvas
  - Project Description (09/07/23, Professor Hossein Saiedian, University of Kansas)
  - Project Plan (09/15/23, Professor Hossein Saiedian, University of Kansas)
  - EECS 348: Software Engineering Lecture Slides (08/21/23 – Current, Professor Hossein Saiedian, University of Kansas)
- KU EECS Computing Lab, equipped with Linux OS and various programming software

4.2.1 Phase Plan

Dates given in the table below are the assignment dates, not the due dates, as most due dates are not known.

Task Name	8/29	9/19	10/24	10/31	11/14	11/28	12/05
Project Management Plan	---						
Project Requirements		---					
Project Architecture and Design			---				
Project Implementation				---			
Project Test Cases					---		
Project User						---	

Manual							
Final Project Implementation							---

MAJOR MILESTONES AND ACHIEVEMENT CRITERIA:

Part 1: Project Management Plan: Deliverables for this portion include this document, completed and up-to-date for each new iteration. Functions as a guideline and progress tracker for the project.

Final Project Implementation: Deliverables will include the C++ source code, design, requirements, an executable application, a user manual, user documentation, finalized project plan, and test cases. This is the final submission for the project, and as such all portions must be fully functional and free of bugs and errors.

For all other portions, beyond names, no details are given. As such, no achievement criteria can be listed.

Each date marks the assignment of a major milestone with its associated deliverables. With the exception of the first and final portions, detailed assignment information is unknown.

The project timeline and associated major milestones are shown in Section 4.2, Project Plan.

4.2.2 Iteration Objectives

Utilize version control from phase Project Implementation onwards.

- **Project Management Plan** - Delegate sections of each phase to group members, find times to work on the project as a group, create a rough time frame for each project phase.
- **Project Requirements** - Account for all external resources needed to create the project, such as version control and group communication. Have the group leader read through the project rubric and relay important information to group members.
- **Project Architecture and Design** - Create a UML use case model of the calculator, and decide the best way to implement each operation and user command line communication.
- **Project Implementation** - Each group member writes their respective section of the project code; utilize group feedback to optimize and debug the code.
- **Project Test Cases** - Using multiple test trials by different group members, make sure that the program fulfills all required functions defined by the rubric.
- **Project User Manual** - Have each group member write a user manual for their respective section of the program and have other group members review, edit, and test the manual.
- **Final Project Implementation** - Present the program, user manual, and all miscellaneous deliverables in a polished format. Review that the program covers the design requirements and use cases. Perform final bug testing.

4.2.3 Releases

No current software releases.

4.2.4 Project Schedule

(Subject to change) **Tentative Project Schedule**

Section of Project	Target completion date
Project Management Plan	Week of 08/29/23
Project Requirements	Week of 09/19/23
Project Architecture and Design	Week of 10/24/23
Project Implementation	Week of 10/31/23
Project Test Cases	Week of 11/14/23
Project User Manual	Week of 11/28/23
Final Project Implementation	Week of 12/05/23

4.2.5 Project Resourcing

The only staff needed will be the Project Manager's teaching assistant. We should not need any specialized skills or experience from staff members, excluding proficiency in C++ and a general understanding of project requirements.

The only special training required by team members will be:

- Every team member must be competent in C++ by the start of the Project Implementation phase (The week of 10/31.)
- Some team members must additionally be competent in using git and GitHub by the Project Design phase (Around the week of 9/19.)

4.3 Project Monitoring and Control

- Requirements Management: To effectively manage product requirements, each deliverable will be traced from the project's initiation to the final implementation. Team members will consult each other regarding any desired changes, requiring approval by all team members before implementation.

- Quality Control: Each skill, environment, and needed resource (such as git/GitHub) will be laid out, to ensure all team members have a basic understanding before using it. Only once everyone has enough proficiency will the project move forward with using each resource. Every team member should be proficient in the programming language C++, so that all members can work on any part of the code, whether simple or complex. This allows for an environment where work is easily reviewed and corrected. If a team member corrects another's work (e.g., an error in documentation or faulty code,) the correction should be noted or communicated in some way. If team members are not proficient enough with a given tool or resource, extra time will be allotted for building skill prior to moving forward. Each iteration of the project should be thoroughly inspected and tested prior to submission, and consistent communication between team members allows for a deeper understanding of the code itself and subsequent fixes.

- Reporting and Measurement: There will be weekly progress reports, either written down or discussed verbally between team members. The report will summarize the status of the project, the milestones achieved, review/reconstruction, and any deviations from the original project plan.

- Risk Management: Group 32 will regularly identify, analyze, monitor, and prioritize risks. To do this, a list will be kept with a description of identified risks, as well as potential risks that could occur if there is deviation from

the project plan. Also included is the potential impact, likelihood, and mitigation strategies to these risks. Since risk assessment is an ongoing process, Group 32 will discuss risk reviews weekly.

- Configuration Management: A Configuration Management Plan will include the process of identifying a problem, reviewing it, and submitting it to the Change Requests log for later fixing. The project artifacts will be documented, ensuring version control throughout the project. Each new artifact deployed will have a unique version. It is the responsibility of the team to keep each artifact up-to-date with the newest iteration.

4.4 Requirements Management

Requirements for this system are described in the provided documents (listed in References). Any further development on system requirements will be communicated via teaching assistants, Professor Hossein Saiedian, or by documents provided with each new iteration phase.

4.5 Quality Control

Any significant defects in the project, code or otherwise, should be noted in Change Requests and thoroughly communicated between team members. Each project piece should be reviewed by multiple team members, and should undergo strict testing. The Quality Control team member has a role in checking completed and in-progress work for errors and communicating these errors to the team.

4.6 Reporting and Measurement

The end of each iteration phase should have detailed reports of progress, schedule changes, and changes to requirements. The Minimal Set of Metrics, as described in the RUP Guidelines, calls for:

Earned value for completed tasks, for reschedule and budget analysis.

Total defects, open or closed, to track remaining workload.

Acceptance test cases passing, to display progress.

Requests for defective code or faulty information fixes will be tracked as Change Requests. Further guidelines for Change Requests are defined in Section 4.5, Quality Control.

4.7 Risk Management

Risks are identified in the Inception Phase. Any further risks will be identified as the scope and format of the project changes for each new iteration. Risks must be discussed with team members to prioritize proper spending of time and resources.

4.8 Configuration Management

Group 32 will continuously review and update Change Requests (adding or removing,) as well as keep a GitHub repository of all project deliverables. Team members are to keep artifacts of source code, test scripts, data files, design documents, and documentation regarding these, for review and submission. All artifacts must remain up-to-date and functional with each new iteration.

5. Annexes

The project will follow the UPEDU process.

Additional process plans and instructions are described in the documents listed in the References section.