Orange

Arithmetic Expression Evaluator Software Development Plan

Version 1.0

Revision History

Date	Version	Description	Author
09/17/23	1.0	Make initial plan on making an arithmetic expression evaluator program in C++.	Mikaela Navarro
12/03/23	1.2	Update document	Mikaela Navarro

Table of Contents

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of this *Software Development Plan* is to compile essential information essential for project oversight and management. This plan outlines the strategic approach to software development, providing a comprehensive framework for achieving successful results.

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* delineates the comprehensive strategy for the Arithmetic Expression Evaluator project, encompassing product deployment. Specifics pertaining to individual iterations will be detailed in the *Iteration Plans*.

The plans as outlined in this document align with the product requirements as defined in the *Vision Document*.

1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

1.4 References

At this time, there are no external documents or references relevant to this *Software Development Plan*.

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 and objective of this project are to develop a user-friendly software program capable of parsing and evaluating arithmetic expressions provided by the user. These expressions may include operators and numeric constants, as well as parentheses to define precedence and grouping. The primary goal is to ensure the program handles all possible input scenarios effectively and delivers accurate results.

2.2 Assumptions and Constraints

User Input

- It is assumed that users will provide valid arithmetic expressions for evaluation, adhering to the defined syntax and operators. (+, -, *, /, %, ^, and ( )).

Platform Compatibility

- The software will be developed to run on common operating systems (Windows, macOS, and Linux) and assumes that users will have access to compatible hardware.

Development Team Availability

- The development team will maintain consistent availability and ensure sufficient resources are allocated for the project's entire duration. Effective communication will be facilitated primarily through the GroupMe platform, supplemented by regular in-person meetings throughout the semester.

Timeline

- The project is constrained by fixed deliver dates, necessitating efficient time management and prioritization of tasks.

Scope

- The scope of this project is constrained to parsing and evaluating arithmetic expressions; additional features or functionalities may not be accommodated within the current project timeline.

2.3 Project Deliverables

Project Management Plan – 09/24/23

Requirements Document – Finish around the week of 10/08/23

Design Document – Finish around the week of 10/24/23

C++ Code Implementation – Finish around the week of 11/07/23

Test Cases – Finish around the week of 11/16/23

User Manual – Finish around the week of 11/30/23

Implementation – Finish around the week of 12/5/23

2.4 Evolution of the Software Development Plan

The *Software Development Plan* for this project will evolve according to the following schedule and criteria:

Version	Scheduled Revision Date	Criteria for Revision and Reissue

1.1	10/15/23	- We have completed the planning and documentation of our requirements and design
1.2	11/14/23	- We have finished developing the test cases and have identified both the functional aspects that work as intended and those that do not. Additionally, we have devised improvement strategies to enhance the project further
1.3	11/30	- Conclude the ultimate implementation with all components fully updated and finalized

3. Project Organization

3.1 Organizational Structure

Team Administrator – Responsible for project coordination, administrative, tasks, and ensuring that project resources are effectively managed. They serve as a central point of contact for team members and assist in maintaining project documentation.

Utility Developer: They are tasked with developing auxiliary tools, libraries, or utilities that support the primary development tasks. They work closely with the technical leads to ensure seamless integration of utilities into the project.

Project/Tech Lead 1: Holds a leadership role and oversees technical aspects of the project. They're responsible for guiding the development process, making architectural decision, and ensuring alignment with project objectives.

Project/Tech Lead 2: Similar to Lead 1, plays a critical role in project management and technical leadership. They collaborate closely with Lead 1 to ensure a coordinated effort in achieving project goals.

Quality Assurance Engineer: Responsible for conducting rigorous testing, ensuring code quality, and validating that the project adheres to specified requirements. They also contribute to maintaining the overall quality of the software product.

3.2 External Interfaces

Stakeholders

Internal Contact Name: Orange

External Contact Name: Professor and TA

Responsibilities: Regularly communicating project progress and milestones while actively seeking feedback and approval on key deliverables.

End Users

Internal Contact Name: Orange

External Contact Name: Peers/Professor and TAs

Responsibilities: Conduct user interviews and surveys to understand user needs, provide user support during the testing phases, and ensuring that the software aligns with user expectations and usability requirements.

Testing and Quality Assurance

Internal Contact Name: Quality Assurance Engineer

Responsibilities: Conducting quality assessments and actively reviewing and addressing issues and defects identified during testing.

3.3 Roles and Responsibilities

Person	**UPEDU Role**	**Responsibilities/Key Tasks**
Mikaela Navarro	Team Administrator	- Administrative tasks for the team - Team coordination
Elijah Pijawnowski	Utility Developer/Quality Assurance Engineer	- Development of utility components - Testing and quality control - Issue tracking and reporting
Adam Albee	Project/Technical Lead 1	- Project planning and strategy - Technical leadership
Mahgoub Husien	Project/Technical Lead 2	- Technical oversight - Collaboration with Lead 1
Jackson Yanek	Quality Assurance Engineer	- Testing and quality control - Issue tracking and reporting

4. Management Process

4.1 Project Estimates

This project doesn’t involve direct financial costs. However, resource allocation and time management will be essential factors in project planning and execution. The schedule for this project is outlined in the *Project Plan*, which details milestones and task timelines. Re-estimation will occur at key project milestones or when significant changes in scope, resources, or timelines occur.

4.2 Project Plan

Devise Project Management Plan
- Start: 09/17/23
- Due: 09/24/23

Project Requirements
- Start: 09/21/23
- Due: 10/15/23

Project Architecture and Design
- Start: 10/24/23
- Due: 11/12/23

Project Implementation
- Start: 10/31/23
- Due: 12/03/23

Project Test Cases
- Start: 11/14/23
- Due: 12/03/23

Project User Manual

- Start: 11/28/23
- Due: 12/03/23

Update Project Implementation

- Start: 12/5/23
- Due: 12/03/23

4.2.1 Phase Plan

N/A

4.2.2 Iteration Objectives

The project will be divided into multiple iterations, each with its specific objectives. Refer to the related *Iteration Plan Documents* for more detailed information. The objectives for the project are as follows:

Iteration 1 – Requirements Gathering and Initial Design (09/21/23 – 10/24/23)

- Collect and refine project requirements based on stakeholder input and analysis
- Develop an initial architectural design for the software
- Define the user interface specifications
- Create a comprehensive Requirements Document

Iteration 2 – Detailed Design and Development (10/24/23 – 10/31/23)

- Elaborate on the architectural design and create detailed technical specifications
- Begin the implementation of core functionality
- Develop utility components and libraries
- Commence work on the Design Document

Iteration 3 – Implementation and Testing (10/31/23 – 11/14/23)

- Continue and finalize the implementation of the software
- Develop rigorous test cases based on the requirements
- Conduct unit testing and integration testing
- Refine and expand the Design Document

Iteration 4 – User Documentation and Quality Assurance (11/14/23 – 11/28/23)

- Create comprehensive user documentation, including a User Manual
- Begin user testing and quality assurance activities
- Identify and address defects and issues
- Prepare for the final implementation phase

Iteration 5 – Final Implementation and Project Closure (11/23/23 – 12/05/23)

- Conclude the ultimate implementation with all componenets fully updated and finalized
- Conduct comprehensive system testing
- Prepare for project closer activities, including final reviews and documentation

4.2.3 *Releases*

Software Release	Type	Description
Internal Testing	Alpha	Initial version of the project, primarily intended for testing within the team. It’ll focus on implementing core parsing and evaluation functionality
Limited User Testing	Beta	Advanced version of the project made available to a limited group of external users for testing. Includes expanded parsing capabilities and initial operator support
Release Candidate	Release Candidate	Close-to-final version of the project, with complete parsing and operator support. Stable and ready for user acceptance testing
Official Release	Production	Fully polished and stable version of the project, equipped with comprehensive parsing, operator support, and error handling. Ready for general availability and public use

4.2.4 *Project Schedule*

N/A

4.2.5 *Project Resourcing*

N/A

4.3 Project Monitoring and Control

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

N/A

4.7 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.

4.8 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 (EEE-FFF-X.Y.doc) for detailed information.

5. Annexes

Development Process – The project will adhere to the software development process defined by UPEDU.

Programming Guidelines – The team will follow a set of Programming Guidelines that outline coding standards, conventions, and best practices for C++ development. These guidelines ensure consistency and maintainability in the codebase.

Design Guidelines – Principles and standards to be followed during the architectural and software design phases. They help maintain a well-structured and modular design for the project

Process Guidelines – Detailed instructions on various aspects of the software development process, including project management, requirements gather, testing, and quality assurance. These guidelines serve as a reference for project team members to ensure uniformity in processes