

**Arithmetic Expression Evaluator**  
**Software Development Plan**  
Version <1.0>

Arithmetic Expression Evaluator	Version: <1.0>
Software Development Plan	Date: 9/25/2024
Alexander Carrillo	

## Revision History

Date	Version	Description	Author
9/25/2024	0.0	Initial meeting, discussion of meeting times and assigning roles for the project	Alexander Carrillo

Arithmetic Expression Evaluator	Version: <1.0>
Software Development Plan	Date: 9/25/2024
Alexander Carrillo	

# 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..... 5

**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..... 5**

    3.1 Organizational Structure..... 5

    3.2 External Interfaces..... 6

    3.3 Roles and Responsibilities..... 6

**4. Management Process..... 6**

    4.1 Project Estimates..... 6

    4.2 Project Plan..... 6

    4.3 Project Monitoring and Control..... 7

    4.4 Requirements Management..... 7

    4.5 Quality Control..... 7

    4.6 Reporting and Measurement..... 7

    4.7 Risk Management..... 8

    4.8 Configuration Management..... 8

**5. Annexes..... 8**

Arithmetic Expression Evaluator	Version: <1.0>
Software Development Plan	Date: 9/25/2024
Alexander Carrillo	

# Software Development Plan

## 1. Introduction

### 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 **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 outline for the construction of our Arithmetic Expression Evaluator, it will include descriptions of current iterations and near in the future iterations to keep an organized and concise project plan

### 1.3 Definitions, Acronyms, and Abbreviations

See the Project Glossary.

### 1.4 References

*[This subsection provides a complete list of all documents referenced elsewhere in the **Software Development Plan**. Identify each document by title, report number if applicable, date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

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

- *Iteration Plans*
- *Development Case*
- *Vision* *[you may prepare a vision statement of your own: what your vision for the project is]*
- *Glossary*
- *Any other supporting plans or documentation.*

Arithmetic Expression Evaluator	Version: <1.0>
Software Development Plan	Date: 9/25/2024
Alexander Carrillo	

## 1.5 Overview

C++ project to make a working arithmetic evaluator which can handle operators +, -, \*, /, %, \*\*. It also needs to be able to handle parentheses and grouping.

## 2. Project Overview

### 2.1 Project Purpose, Scope, and Objectives

2.1.1 A user interface calculator that can handle 6 operators and ordering, we will break it into pieces: each of the operators, ordering, and user interface.

### 2.2 Assumptions and Constraints

2.2.1 scheduling might be difficult but I think the scope of the project isn't very big where I feel we can organize over group chats if everyone is doing their part

### 2.3 Project Deliverables

2.3.1 working addition, subtraction, multiplication, division, modulation, and exponents with pmdas

2.3.1.1  $1 + 12 = 13$

2.3.1.2  $3 - 2 = 1$

2.3.1.3  $4 * 8 = 32$

2.3.1.4  $12 / 3 = 4$

2.3.1.5  $4 \% 2 = 0$

2.3.1.6  $10 ** 2 = 100$

2.3.1.7  $(3 * (2 - 1)) / 2 = 1.5$

### 2.4 Evolution of the Software Development Plan

2.4.1 addition subtraction

2.4.2 multiplication division

2.4.3 modulation exponents

2.4.4 pmdas

2.4.5 user interface

## 3. Project Organization

### 3.1 Organizational Structure

3.1.1 Our group is divided into provided roles of scrum master, project manager, technical lead, configuration manager, QA lead, and UI designer. we will divide more technical problems to the hands on roles while more documentation is on the scrum master and project manager.

### 3.2 External Interfaces

*[Describe how the project interfaces with external groups. For each external group, identify the internal and external contact names. This should include responsibilities related to deployment and acceptance of the product.]*

Arithmetic Expression Evaluator	Version: <1.0>
Software Development Plan	Date: 9/25/2024
Alexander Carrillo	

### 3.3 Roles and Responsibilities *[Identify the project organizational units that will be responsible for each of the disciplines, workflow details, and supporting processes. The text below is provided as an example.]*

Person	Unified Process for EDUcation Role
Viren Chowdary Padarathi	UI/UX Designer
Noah Hickman	Scrum Master
Riley Backus	Quality Assurance (QA) Lead
Sina Asheghalishahi	Configuration (Version Control) Manager
Alexander Carrillo	Project Manager
Wyatt Payne	Technical Lead

Anyone on the project can perform [Any Role](#) activities.

## 4. Management Process

### 4.1 Project Estimates

*[Provide the estimated cost and schedule for the project, as well as the basis for those estimates, and the points and circumstances in the project when re-estimation will occur.]*

### 4.2 Project Plan

4.2.1 weekly meeting with team members to create group documentation in addition with the personal documentation. We will start with operators then pemdas then user UI. Work will divided based on team members responsibility

#### 4.2.2 Phase Plan

*[Include the following:]*

- a Gantt chart showing the allocation of time to the project phases (Not necessarily detailed to the activity level; this type of Gantt Chart is providing along with the Iteration Plans themselves; Provide an Overview of the project Timeline with the major miles stones)
- identify **major milestones** with their achievement criteria

*Define any important release points and demos.]*

*[If available, refer to the related **Iteration Plan Documents** for more details]*

#### 4.2.3 Iteration Objectives

4.2.3.1 the plan currently is to start with most basic operators then move on to more difficult ones, then pemdas, then user UI. Documentation will be provided weekly from all team members

#### 4.2.4 Releases

*[A brief description of each software release and whether it's demo, beta, and so on.]*

Arithmetic Expression Evaluator	Version: <1.0>
Software Development Plan	Date: 9/25/2024
Alexander Carrillo	

#### 4.2.5 Project Schedule

*[Diagrams or tables showing target dates for completion of iterations and phases, release points, demos, and other milestones.] [Limit to major project milestone, e.g., requirements, design, implementation, and testing]*

#### 4.2.6 Project Resourcing

*[Identify the numbers and type of staff required here, including any special skills or experience, scheduled by project phase or iteration.]*

*List any special training project team members will require, with target dates for when this training should be completed.]*

### 4.3 Project Monitoring and Control

#### 4.4 Requirements Management

The requirements for this system are captured in the Vision document. Requested changes to requirements are captured in Change Requests, and are approved as part of the Configuration Management process.

#### 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.

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

#### 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.

Arithmetic Expression Evaluator	Version: <1.0>
Software Development Plan	Date: 9/25/2024
Alexander Carrillo	

*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

*[Additional material of use to the reader of the **Software Development Plan**. Reference or include any project technical standards and plans which apply to this project. This typically includes the Programming Guidelines, Design Guidelines, and other process guidelines. The text that follows is provided as an example.]*

The project will follow the UPEDU process.

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