
EECS 348 Group 14

**Arithmetic Expression Evaluator in C++
Software Development Plan**
Version 1.0

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu sdp	

Revision History

Date	Version	Description	Author
19/09/23	1.0	First Draft	Vincent Dick

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu_sdp	

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 (N/A).....	6
3.3 Roles and Responsibilities.....	6
4. Management Process.....	6
4.1 Project Estimates (N/A).....	6
4.2 Project Plan.....	6
4.3 Project Monitoring and Control.....	7
4.4 Requirements Management (N/A).....	7
4.5 Quality Control.....	7
4.6 Reporting and Measurement (N/A).....	7
4.7 Risk Management.....	8
4.8 Configuration Management.....	8
5. Annexes.....	8

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu sdp	

Software Development Plan

1. Introduction

1.1 Purpose

The purpose of the Arithmetic Expression Evaluator Software Development Plan is to organize necessary information in order to achieve project goals in an organized and timely manner. It describes the approach and development of the arithmetic software, and is a top-level plan to be used by members of the team to track and plan progress.

The following people use the Arithmetic Expression Evaluator 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.
- The project quality assurance members use it to periodically keep track of changes made in the software and give feedback on what changes are beneficial to the project.

1.2 Scope

*[A brief description of the scope of this **Software Development Plan**; what Project(s) it is associated with and anything else that is affected or influenced by this document. The text below is provided as an example.]*

This Arithmetic Expression Evaluator Software Development Plan describes the overall plan to be used by the EECS 348 Group 14, specifically the development 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

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the Software Development Plan. This information may be provided by reference to the project's Glossary.]

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 Arithmetic Expression Evaluator Software Development Plan, the list of referenced artifacts includes:

- Iteration Plans *[for example: plan to implement + and -; plan to implement * and /, ...]*
- Development Case (N/A)
- Vision *[you may prepare a vision statement of your own: what your vision for the project is]*
- Glossary

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu_sdp	

- C++ Official Documentation

1.5 Overview

*[This subsection describes what the rest of the **Software Development Plan** contains and explains how the document is organized. The text below is provided as an example.]*

This Arithmetic Expression Evaluator 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 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

[A brief description of the purpose and objectives of this project and a brief description of what deliverables the project is expected to deliver.]

The purpose of this project is to develop a calculator that follows PEMDAS order of operations. The deliverables are the code so that the user can enter in math equations and get a desired, correct result. More deliverables are this document, as well as the logs of all of our meetings.

2.2 Assumptions and Constraints

[A list of assumptions that this plan is based and any constraints, for example. staff, equipment, schedule, that apply to the project.]

We assume that we all have sufficient knowledge in C++ in order to get this done. We also assume that we all have access and knowledge to navigate through Linux. Some constraints are that we all have different classes, and therefore have different schedules on when we can work on this project. Another constraint is that this project is to be completed within the semester and to the requirements described in the project description.

2.3 Project Deliverables

[A list of the artifacts to be created during the project, including target delivery dates. The text below is provided as an example.] Requirements, design specs, test cases, code

The deliverables for the project are as follows:

- Project management plan
- Project requirements
- Project architecture and design
- Project code
- Test cases
- User manual

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

*[A table of proposed versions of the **Software Development Plan**, and the criteria for the unscheduled*

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu sdp	

revision and reissue of this plan. The text below is provided as an example.]

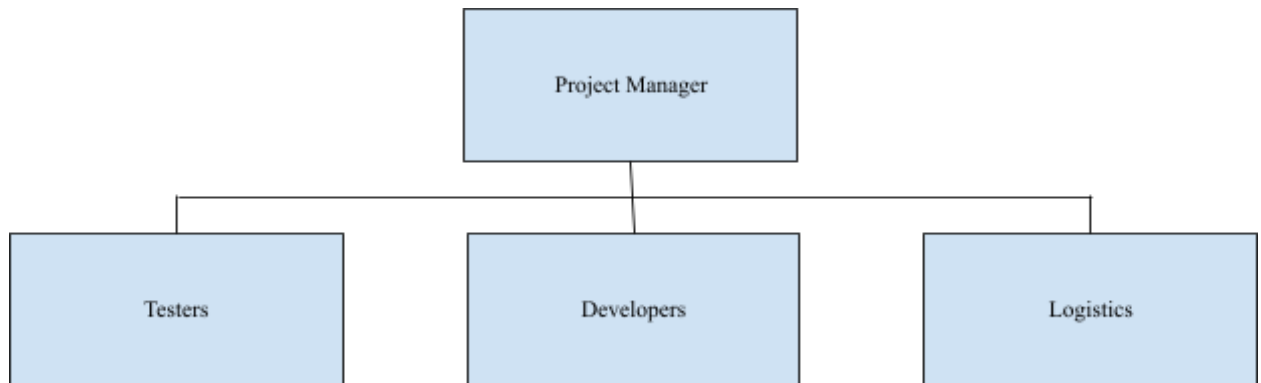
The *Software Development Plan* will be revised prior to the start of each Iteration phase.

3. Project Organization

3.1 Organizational Structure

[Describe the organizational structure of the project team, including management and other review authorities.]

The project team for the arithmetic expression evaluator will be as follows:



3.2 External Interfaces (N/A)

[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.]

3.3 Roles and Responsibilities *[the more details here, the easier your job; include contact info, availability info, expertise, ...]*

[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
Project Manager	Secures resources for team use, manages priorities, and generally keeps the team focused on goals. Manager should also ensure that the product and its artifacts are kept to a quality standard.
Developers	Primarily responsible for the analysis and design, implementation, and configuration of the software. Provides support and insight to the Project Manager
Logistics	Primarily responsible for managing the requirements for the software. Provides support and insight to the Project Manager.
Testers	Primarily responsible for test design, test execution, evaluation of test execution and recovery from errors, and assessing the results of testing and logging defects. Provides support and insight to the Project Manager.

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

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu sdp	

4. Management Process

4.1 Project Estimates (N/A)

[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.]

This project does not require any monetary estimates and the project schedule is projected to span over the months of September to December.

4.2 Project Plan

[This section contains the schedule and resources for the project.] **Project artifact as well as iteration schedules!**

4.2.1 ~~Phase Plan~~ (N/A)

[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]*

4.2.2 Iteration Objectives

*[Briefly list the objectives to be accomplished for each of the iterations and Refer to the related **Iteration Plan Documents** for more details.]*

Iteration	Objectives	Artifacts
Project Management Plan	<ul style="list-style-type: none"> - The project management plan is created and populated to serve as a basis for development moving forward. - Team roles are defined, and logistics (contact information, meeting times, etc.) are coordinated among team members 	The initial version of the Project Management Plan
Project Requirements	<ul style="list-style-type: none"> - The purpose, functionalities, and constraints of the Arithmetic Expression Evaluator are identified and documented. 	An updated version of the Project Management Plan

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu sdp	

Project Architecture and Design	- The architecture and overall design of the Arithmetic Expression Evaluator are defined.	Models, diagrams, and specifications corresponding to the design of the project
Project Implementation	- The Arithmetic Expression Evaluator is programmed according to the specifications and designs of the previous 3 iterations.	The project code
Project Test Cases	- The Arithmetic Expression Evaluator is run against test cases to ensure the accuracy and consistency of the program.	The code and descriptions of the tests
Project User Manual	- The document detailing the information needed for a user to successfully operate the Arithmetic Expression Evaluator is written.	The project user manual

4.2.3 Releases

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

The beta release of the software is scheduled for the 13th week of the semester (the week of November 13th), corresponding to the completion of the Project Implementation iteration. The final release of the software is scheduled for the 16th week of the semester (the week of December 4th).

4.2.4 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]**

Project Aspects and Phases	Target Dates
Project Management Plan	September 24th, 2023
Project Requirements	TBD
Project Architecture and Design	TBD
Project Implementation	TBD

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu_sdp	

Project Aspects and Phases	Target Dates
Project Test Cases	TBD
Project User Manual	TBD
Project Implementation	TBD

~~4.2.5~~ **Project Resourcing (N/A)**

[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

[The following is a checklist of items to consider:

- *Requirements Management: Specify the information and control mechanisms which will be collected and used for measuring, reporting, and controlling changes to the product requirements.*
- *Quality Control: Describe the timing and methods to be used to control the quality of the project deliverables and how to take corrective action when required. Include techniques, metrics, criteria, and procedures used for evaluation— this will include walkthroughs, inspections, and reviews. Note that this is in addition to the Test Plan, which is not enclosed in the Software Development Plan.*
- ~~*Reporting and Measurement: Describe reports to be generated. Specify which metrics should be collected and why. **OR** if available, refer to the **Project Measurements and Project Measurements** document (N/A)*~~
- *Risk Management: Describe the approach that will be used to identify, analyze, prioritize, monitor and mitigate risks. If available, refer to the **Risk List** document.*
- *Configuration Management: Describe the process by which problems and changes are submitted, reviewed, and dispositioned. Describe how project or product artifacts are to be named, marked, and numbered, including system software, plans, models, components, test software, results and data, executables, and so on. Describe retention policies, and the back-up, disaster, and recovery plans. **OR** if Available, Refer to the **Configuration Management Plan** document*

The text that follows is provided as an example.]

~~4.4~~ **Requirements Management (N/A)**

Requirements Management: Specify the information and control mechanisms which will be collected and used for measuring, reporting, and controlling changes to the product requirements.

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

Quality Control: Describe the timing and methods to be used to control the quality of the project deliverables and how to take corrective action when required. Include techniques, metrics, criteria, and

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu_sdp	

procedures used for evaluation— this will include walkthroughs, inspections, and reviews. Note that this is in addition to the Test Plan, which is not enclosed in the Software Development Plan.

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)

Reporting and Measurement: Describe reports to be generated. Specify which metrics should be collected and why. OR if available, refer to the Project Measurements and Project Measurements document

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

Risk Management: Describe the approach that will be used to identify, analyze, prioritize, monitor and mitigate risks. If available, refer to the Risk List document.

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

Configuration Management: Describe the process by which problems and changes are submitted, reviewed, and dispositioned. Describe how project or product artifacts are to be named, marked, and numbered, including system software, plans, models, components, test software, results and data, executables, and so on. Describe retention policies, and the back-up, disaster, and recovery plans. OR if Available, Refer to the Configuration Management Plan document

Appropriate tools will be selected which provide a database of Change Requests and a controlled versioned repository of project artifacts.

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Development Plan	Date: 19/09/23
upedu sdp	

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.