
Scrum Blasters

Big Ole Calculator
Software Development Plan
Version 1.01

Big Ole Calculator	Version: 1.01
Software Development Plan	Date: 09/25/2024

Revision History

Date	Version	Description	Author
09/18/2024	1.0	Initial Plan	Everyone
9/25/2024	1.01	Final Touches on Initial Plan	Everyone

Big Ole Calculator	Version: 1.01
Software Development Plan	Date: 09/25/2024

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 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..... 7*
 - 4.8 Configuration Management..... 7*
- 5. Annexes..... 7**

Big Ole Calculator	Version: 1.01
Software Development Plan	Date: 09/25/2024

Software Development Plan

1. Introduction

This document will provide a detailed plan for designing, implementing, and testing our terminal calculator. The roles of each member and the tentative weekly plan will be documented here. The calculator we create will have an expression parser and evaluation for basic arithmetic functions (+, -, *, /, %, **).

1.1 Purpose

The purpose of this *Software Development Plan* is to lay out all the groundwork necessary to identify all facets of project detail. This will describe the approach to the development of the expression parser and it is the top-level plan generated and used by all team members to direct and follow 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 will need to do it, and what other activities they are most dependent upon.

1.2 Scope

This *Software Development Plan* describes the overall plan to be utilized in the design, implementation, and testing of the Big Ole' Calculator project. This includes deployment of the product, and artifacts to document the development process. The details of the individual iterations will be described in the Iteration Plans.

1.3 Definitions, Acronyms, and Abbreviations

- UPEDU: Unified Process for Education
- UML: Unified Modeling Language

1.4 References

- No references listed.
- The vision for this project is that the team successfully employs industry-standard development practices to create a software that is efficient and able to quickly parse through a given string and determine its validity, and then its value.

1.5 Overview

This *Software Development Plan* contains the following information:

Project Overview	—	The purpose of this project is to create a basic arithmetic expression parser as a means of furthering team experience with software development. As such, this project will deliver numerous artifacts—including this document—in addition to the created software.
Project Organization	—	The project team is broken up into Project Manager, UI/UX Designer, Scrum Master, Technical Lead, Quality Assurance, and Version Control Designer. Each role will have a varying level of importance throughout each stage of development
Management Process	—	The major phases are completed as assigned to us by the stakeholders of EECS 348. The project will be completed by the end of the Fall Semester.

Big Ole Calculator	Version: 1.01
Software Development Plan	Date: 09/25/2024

Applicable Plans and Guidelines — We will follow a waterfall, linear model as we work through each stage of development to create our deliverables. However, during implementation, we shall employ Agile practices of development.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The team plans on delivering a fully functional terminal line calculator that is programmed in C/C++. There will be the 4 basic functions including addition, subtraction, multiplication and division. Along with the 4 basic functions there will be the ability to use exponents and modular division. This will all be packed with an expression parser that will read from the terminal. Testing will be done both manually and through scripts.

2.2 Assumptions and Constraints

The calculator will be able to handle numbers between -2,147,483,648 and 2,147,483,647 . The expressions entered will ONLY be through the terminal. The calculator has to be programmed in C/C++. Git will be used as the version control system.

2.3 Project Deliverables

Requirements Engineering - Documentation and listed objects completed by October 7th

Design - Overall design and modeling of the calculator done by TO BE DETERMINED (Estimated October 14th)

Code - Bulk of the functions and a testable calculator done by TO BE DETERMINED (Estimated November 11th)

Testing - Developing test scripts and going over edge cases done by the end of November.

Deliverables for each project phase are identified in the Development Case. For version planning, see 2.4 *Evolution of the Software Development Plan*.

2.4 Evolution of the Software Development Plan

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

Version	Date to complete by (Subject to Change)
1.0.0 Simple version with addition and subtraction functionality	October 21st
1.1.0 Parentheses Checker	October 28th
1.2.0 Multiple Signs Checker	November 4th
1.3.0 PEMDAS Implementation	November 11th
1.4.0 Error Handling	November 18th

3. Project Organization

3.1 Organizational Structure

Project Manager: Keeps track of the project schedule, assigns tasks, and makes sure everyone meets deadlines. They also handle any issues that come up.

Scrum Master (especially during coding): Helps the team follow Agile practices, organizes daily check-ins,

Big Ole Calculator	Version: 1.01
Software Development Plan	Date: 09/25/2024

and plans work sessions.

Technical Lead: Offers technical advice and ensures the team follows good coding practices. They also help solve any technical problems.

Quality Assurance (QA) Lead: Makes sure the project artifacts meet quality standards. During coding, they plan and run tests to find and fix bugs.

UI/UX Designer: Designs the user interface, especially if a GUI interface is planned. They create sketches and prototypes to ensure the final product is easy to use.

Configuration (Version Control) Manager: Manages the project's version control system. They track changes, manage updates, and ensure everyone is working with the correct version of the project

3.2 Roles and Responsibilities

Person	Unified Process for EDUcation Role
Collin Tullis Email: c336t319@ku.edu Mobile Phone: (785) 492-8024	Version Control Manager
Adira Mongar Email: a523m400@ku.edu Phone: (816) 800-2135	Quality Assurance
Gavin Billinger Email: g859b530@ku.edu Mobile Phone: (785) 592-0478	Project Manager
Matthew Eagleman Email: mattheweagleman@gmail.com Phone: (913) 951 - 6235	UI/UX Designer
Lydia Peng Email: lydia64peng@ku.edu Phone: (816) 547-7615	Scrum Master
Ethan Gao Email: ejgao@ku.edu Phone Number: 913-605-6776	Technical Lead

4. Management Process

4.1 Project Plan

1. Obtain basic knowledge of requisite programming language related to the project
2. Design functionality of the device
3. Spread out different objects between multiple individuals
4. Implement design features through
5. Review requirements and test provided protocols for bug fixing and quality assurance.
6. Success

4.1.1 Iteration Objectives

1. Operators Implementation
2. Parentheses Checker
3. Multiple Signs Checker
4. PEMDAS Implementation
5. Error Handling

Big Ole Calculator	Version: 1.01
Software Development Plan	Date: 09/25/2024

4.1.2 Releases

- Due to constraints in time and scope, there will be one release for the project.

4.1.3 Project Schedule

1. Project is scheduled according to the stakeholders.

4.2 Requirements Management

Each artifact and code iteration will be reviewed by multiple members of the group. After 3 people have check over the artifact or code, it will be added to the main branch of the Git.

4.3 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.

4.4 Risk Management

Risks will be identified in the 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.

4.5 Configuration Management

We will use github to do version control, track changes, and handle change requests.

We will use the github project board in order to keep track of what needs to get done, and when it is done. All of our documentation will be uploaded to the github, as well as any exe files and artifacts.

The change requests will be managed by the version control manager. They will do the reviewing and accepting of the change requests.

5. Annexes

The project will follow the UPEDU process.

Programming Guidelines: Language will be in C++.

Design Guidelines: The program will be laid out with UML. The design should be intuitive enough that the user can use it with minimal instruction.

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