EECS 348 Group 10

Web Calculator for University of Kansas Students
Software Development Plan
Version 2.0

Revision History

Date	Version	Description	Author
09/23/2023	1.0	Initial version of project plan	Harlan Williams
12/03/2023	2.0	Updated version	Jacob Kice

Table of Contents

Software Development Plan

1. Introduction

This document contains: an overview of the scope and vision of the website calculator app, henceforth referred to as *The Project*, a description of *The Project* team's organizational structure, the requirements and objectives for *The Project*, references to all outside projects and services that *The Project* will use, and an overview of the management and development processes used by the team.

1.1 Purpose

The purpose of this *Software Development Plan* is to accumulate, organize, and structure the information necessary for the successful development and control of the implementation and execution of a web-based calculator. It describes the approach towards the software development, while factoring and considering the establishment of the requirements provided. Further, it is generated and used by the project manager to direct and coordinate the development efforts necessary.

The following people use the *Software Development Plan*:

- The **project manager** uses it to record changes in the requirements for *The Project*, to record requirements and a timeline for implementing those requirements, and to track progress as the requirements are implemented.
- **Project team members** use it to identify the most recent version of *The Project* requirements and timeline, to use as a guide while implementing the requirements.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by *The Project*, including deployment of the product. This plan describes the requirements of *The Project* as Objectives in Section 2.1. The details of the individual iterations are described in the Iteration Objectives in Section 4.2.2.

1.3 Definitions, Acronyms, and Abbreviations

Web-app: a web-based application.

DigitalOcean: A service that allows hosting of web-based applications on the cloud in the form of *Droplets*.

Droplet: DigitalOcean's term for a Linux-based virtual machine on their servers that will be used to host the web-app.

Operator: A sign that specifies an arithmetic operation.

Infix notation: An arithmetic statement where operators occur between numbers and operate on the numbers to its left and to its right.

1.4 References

Not applicable.

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 of this project is to construct a web-based calculator app. The objective is that the app is easily accessible and useful to college students who frequently need a calculator as they study or complete homework. The calculator will be simple and reusable, and will perform addition, subtraction, multiplication, division, modulo, and exponents. Additionally, it will be able to parse parentheses and handle the correct order of operations.

- Accept as input arithmetic expressions of arbitrary length, evaluate, and return the answer to the user.
    - Perform error checking and report errors to the user
    - Accept: negation, addition, subtraction, multiplication, division, modulo, and exponentiation of integers and floating-point numbers in infix notation.
        - Addition will be specified by a plus sign, +.
        - Subtraction and negation will be specified by a minus sign, -.
        - Multiplication will be specified by an asterisk, *.
        - Division will be specified by a forward slash, /.
        - Modulo will be specified by the percent sign, %.
        - Exponentiation will be specified by a caret, ^..
    - Accept parentheses in the expression and allow them to alter the order in which the expression is evaluated.
    - Correctly perform the arithmetic in the expression
        - Addition, subtraction, multiplication, division, and modulo will be left-associative. Exponentiation will be right-associative.
        - The unary minus sign will have the highest precedence.
        - Exponentiation will have higher precedence than multiplication, division, and modulo.
        - Multiplication, division, and modulo will have higher precedence than addition and subtraction.
    - Support assignment of variables in order to create portable scripts, termed calculator scripts.
    - Support evaluation of calculator scripts when all variables have been entered.
- Incorporate the calculator in a web interface to allow users to access the calculator, provide input to it, and receive evaluations of the expressions entered.

2.2 Assumptions and Constraints

The Project assumes that all members of the team will be able to work on it until the end of the Fall 2023 semester, and that the web-app will be able to be hosted on DigitalOcean.

The Project will implement the calculator and the back-end of the web-app in C++. *The Project* will be hosted as a Droplet on DigitalOcean and will be free for use by anyone. *The Project* must be completed in full by the end of the Fall 2023 semester, both the calculator implementation and the web-app interface to the calculator. The source code and all documentation will be freely available on GitHub at https://github.com/kmdeskin/EECS348-Class-Project.

2.3 Project Deliverables

The Project will deliver: a requirements document, a design specification, working code, a set of test cases that verify that code's correctness, and a hosted web app that can be connected to by users.

2.4 Evolution of the Software Development Plan

The *Software Development Plan* will be revised prior to the start of each Iteration phase. The *Software Development Plan* is subject to revision to reflect changes in requirements for *The Project* or changes in the organizational structure of *The Project*.

Development Plan Version	Revision Date
Initial version of the *Software Development Plan*	9/23/23
Updated Version of the *Software Development Plan*	12/03/2023

3. Project Organization

3.1 Organizational Structure

The Project's team is under the review of Dr. Saiedian and TAs for the course EECS 348. The team is led by Harlan Williams and all artifacts will be reviewed by Jacob Kice, the quality assurance engineer on the team.

The work on *The Project* is split between two teams: one team will work on the calculator that will be incorporated into the web app, and the other team will work on the website front-end and back-end. The calculator team, which consists of Jacob Kice, Teerapat Saengsubin, and Nora Manolescu, will be headed by Jacob Kice. The website team, which consists of Harlan Williams, Kara Deskin, and Gaby Kill, will be headed by Harlan Williams.

3.2 External Interfaces

The Project will use DigitalOcean (https://www.digitalocean.com/), a cloud hosting service, to deploy a Droplet to host *The Project*. A reference for the services that DigitalOcean provides is provided at https://docs.digitalocean.com/reference/ and support can be found at https://docs.digitalocean.com/support/. Deployment of the DigitalOcean droplet will be managed by the website development team.

The Project will use Oat++ as a web framework. The Oat++ project is hosted on GitHub at https://github.com/oatpp/oatpp. The lead developer of the Oat++ project is Leonid Stryzhevskyi, whose GitHub profile is at https://github.com/lganzzzo.

3.3 Roles and Responsibilities

Harlan Williams
Project Manager
Website Team Lead
Contact: hrw@ku.edu
Availability:
Wed: 12:00 pm - 2:00 pm

Mon-Fri 5:15 pm and later
Major: Computer Science MS (Spring 2025)
Programming Languages/Libraries: C, C++, Javascript, Python 3, Flask, Django
Responsibilities:

- Overseeing project direction
- Defining project features and requirements
- Communicating with the professor and with TAs
- Submitting all project artifacts to the professor
- Manage website team
- Website implementation

Gaby Kill
Meeting Organizer
Website Team
Contact: cat@ku.edu
Availability:
Tuesdays 6pm-8pm
Major: Computer Science BS (Spring 2025)
Relevant Coursework: EECS 168, 268, 101
Programming Languages/Libraries: Python, LaTeX, Java, HTML, C++
Responsibilities:

- Organize meeting schedule
- Take and maintain meeting notes
- Website implementation

Jacob Kice
QA Engineer
Calculator Team Lead
Contact: jakekice@ku.edu
Availability:
Monday and Wednesday: 7:15 – 8:30 pm
Tuesday: 5:30 – 8:30 pm
Thursday: 4:00 – 8:30 pm
Friday: 6:15 pm and later
Saturday and Sunday: Anytime
Major: Computer Science BS, Interdisciplinary Computing in Geography BS (Spring 2026)
Relevant Coursework: EECS 138, 168, 268, GEOG 560; currently enrolled in EECS 210, 348
Programming languages/Libraries: Python, LabVIEW, limited exposure to MATLAB, C++, R
Responsibilities:

- Develop test cases based upon project requirements
- Test functionality of code products
- Review project artifacts for completeness, accuracy, and formatting
- Manage calculator team
- Calculator implementation

Kara Deskin
Configuration Engineer
Website Team
Contact: kara.deskin@ku.edu
Availability:
Tuesday at 6pm
Thursdays at 9am

Major: Computer Science BS (Spring 2025)
Relevant Coursework: EECS 168, 268, 140; currently enrolled in EECS 388, 210, 348
Programming Languages/Libraries: C++, C, Python, Pandas, HTML, NumPy
Responsibilities:

- Maintain GitHub repository
- Manage project versioning
- Website implementation

Nora Manolescu
Requirements Engineer
Calculator Team
Contact: n559m268@ku.edu
Availability:

Monday: after 7:00 pm
Tuesday: after 3:45 pm
Wednesday: after 4:30 pm
Thursday: after 3:45

Major: Computer Science BS (Spring 2025)
Relevant Coursework: EECS 140, 210, 268; currently enrolled in EECS 330, 348
Programming Languages/Libraries: Python, C, C++, MATLAB
Responsibilities:

- Oversee determination of project requirements
- Oversee development of requirements documentation
- Calculator implementation

Teerapat Saengsubin (Top)
Design Engineer
Calculator Team
Contact: topark@ku.edu
Availability:

Monday: after 2:00 pm
Tuesday: after 4:00 pm
Wednesday: after 3:00pm
Thursday: after 4:00pm
Friday: after 2:00pm
Saturday and Sunday: anytime

Major: Math BS, Computer Engineering BS (Spring 2025)
Relevant Coursework: EECS 138, 140, 168, 268; currently enrolled in EECS 210, 348, 388
Programming Languages/Libraries: Python, C, C++, MATLAB
Responsibilities:

- Develop project design
- Oversee development of design documentation
- Calculator implementation

4. Management Process

4.1 Project Estimates

Not applicable.

4.2 Project Plan

4.2.1 Phase Plan

Not applicable.

4.2.2 Iteration Objectives

The calculator team's planned iterations are:

1. A calculator that supports only integer types and addition
2. Subtraction functionality
3. Multiplication functionality
4. Division functionality
5. Modulo functionality
6. Exponentiation functionality
7. Parenthesis functionality
8. Invalid expression handling
9. Float type functionality
10. Script functionality

The website development team's planned iterations are:

1. A basic back-end that can serve static HTML
2. The ability to handle form input through HTML
3. The ability to serve dynamic HTML in response to user input
4. Interface that links to the calculator app
5. Front-end work to polish the user interface.

4.2.3 Releases

The Project will be able to be released in beta and demo form when the web-app is able to interface with the calculator. *The Project* will be released in its final version at the end of the Fall 2023 semester.

4.2.4 Project Schedule

9/19: begin Requirements

10/24: begin Architecture & Design

10/31: begin Implementation

11/14: begin Testing

11/28: begin User Manual

12/5: Final Project Implementation – including completed and updated:

- Project management plan
- Requirements
- Design
- Test cases
- C++ code
- User manual

Completion will also include successful demos.

4.2.5 Project Resourcing

Every team member will begin studying and reviewing C++ beginning on 9/26/23. The website team will begin studying web development beginning on 10/3/23. Both of these trainings should be completed by the beginning of the Architecture & Design phase on 10/24/23.

4.3 Project Monitoring and Control

4.3.1 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.

4.3.2 Quality Control

Defects will be recorded and tracked as Pull Requests on GitHub, 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. The review will include a series of test cases designed to detect possible defects by checking against the actual output against the expected output from many input combinations.

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.

4.3.3 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.

4.3.4 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.

4.3.5 Configuration Management

GitHub will be used to host *The Project*'s Git repository and to manage Pull Requests and control versioning of all 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 Pull Requests are reviewed and approved by the quality assurance engineer, and merge conflicts will be handled by the configuration engineer.

5. Annexes

The Project will follow the UPEDU process.

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