
<Company Name>

<Boolean Algebra Calculator> Software Development Plan Version <1.0>

[Note: The following template is provided for use with the Unified Process for EDUcation. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.] **Marked (shaded) areas: items that are OK to leave out.**

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

Revision History

Date	Version	Description	Author
<02/22/24>	<1.0>	<Initial filling out of roles and management>	<Benjamin Weinzirl>

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

Table of Contents *[keep this; say N/A when inapplicable]*

1. Introduction4

1.1 Purpose4

1.2 Scope.....4

1.3 Definitions, Acronyms, and Abbreviations4

1.4 References.....4

1.5 Overview.....5

2. Project Overview5

2.1 Project Purpose, Scope, and Objectives5

2.2 Assumptions and Constraints.....5

2.3 Project Deliverables5

2.4 Evolution of the Software Development Plan5

3. Project Organization6

3.1 Organizational Structure6

3.2 Roles and Responsibilities *[the more details here, the easier your job; include contact info, availability info, expertise, ...]*6

4. Management Process.....7

4.1 Project Plan.....7

4.2 Project Monitoring and Control7

4.3 Quality Control.....8

4.4 Risk Management.....8

4.5 Configuration Management.....8

5. Annexes.....8

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

Software Development Plan

1. Introduction

*[The introduction of the **Software Development Plan** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this **Software Development Plan**.]*

1.1 Purpose

*[Specify the purpose of this **Software Development Plan**. The text below is provided as an example.]*

The purpose for this *Software Development Plan* is to initially organize the team. This document describes how the software will be developed and how the team will be managed throughout the project.

The **project manager** and the **project team members** use this plan to both understand their roles and requirements within the team and allow the **project manager** to track their work and progress.

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 *Software Development Plan* describes the overall plan to be used by the <Boolean Algebra Calculator> project. The details of the individual iterations will be described in the Iteration Plans. The plans as outlined in this document are subject to change based on information given throughout the EECS 348 class.

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 **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.*

Project guideline is provided on canvas by KU's EECS 348 Spring semester course.

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

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 *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

[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 provide a functional Boolean algebra calculator. The scope of this project is based on the KU Spring 348 project guidelines. This project is expected to deliver a functioning Boolean algebra calculator to a GitHub repository.

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.]

Constraints of this project include the small project team along with the semester long timeframe for the project. The calculator should work on KU's server cycle machines.

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

Project deliverables include versions of each iteration of the project along with the final (complete) project in a GitHub repository. Meeting notes will also be included within the repository. Deliverables will also include a Requirements Document, this *Project Management Plan*, a Design Document, and test cases. The project will also include a User Manual which explains how to effectively use the final product.

2.4 Evolution of the Software Development Plan

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

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

3. Project Organization

3.1 Organizational Structure

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

The **project manager** is responsible for not only keeping the team on track to produce deliverables on schedule but is also in charge of developing meeting times and submitting any necessary work to the class instructor. The final compilation of deliverables will also be done by the **project manager**.

The **lead developer** will be responsible for the organization and implementation of code within the project. The development of each iteration is their responsibility, and they must maintain contact with the **project manager** and other **project team members** whenever updating or creating iterations.

The **analyst** is responsible for domain engineering along with developing use cases and class diagrams to better understand the how's of this project and make the project easier to understand at high-level interpretations.

The **implementer/tester** is responsible for ensuring there are no bugs or errors within the code. They are directly tied to the **lead developer** and will have consistent contact with them throughout the project. The tester is responsible writing unit tests.

3.2 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
Benjamin Weinzirl Contact: b062w572@ku.edu Expertise: Leadership Skills, knowledge of Python, C++, and JavaScript. Availability: Mon 9am-1pm Tues: 4pm-10pm Wed: 2pm-6pm Thurs: 4pm-9pm Fri: 9am-10pm Sat: 9am-10pm Sun: 12pm-10pm	Project Manager
Maya Routh Contact: maya.routh@ku.edu Expertise: Python and C coding. Availability: Mon-Fri 4pm-10pm	Lead Developer
Ibrahim Sufi Contact info: ibrahim.sufi@ku.edu Expertise: Unit Testing, C coding Availability: Wednesday 12pm - 2pm, Thursday and Friday after 4pm	Implementer/Tester
Del Endecott	Analyst

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

contact: del.endecott@ku.edu
 expertise: Python programming,
 implementation availability: Thursday
 8am-12:30pm, Mon-Fri: 4:30pm-6pm

Anyone on the project can perform [Any Role](#) activities whenever the project timeline requires, or help is needed for big parts of the project.

4. Management Process

4.1 Project Plan

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

4.1.1 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 1: Requirements Engineering complete. Use case and class diagrams fully formed.

Iteration 2: Initial implementation of operators.

Iteration 3: Expression parsing and true value input complete.

Iteration 4. A smooth and sleek user-interface design will be available.

Iteration 5. Error handling fully consistent and complete.

Note: Evaluation and output will be done within each iteration to properly test each iteration.

4.1.2 Releases

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

Section will be updated with each release done.

Current outline includes a basic demo of iterations 2 and 3, a beta release of iteration 4, and a final release including iteration 5.

4.1.3 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**

Milestone	Deadline
Requirements	03/4/24
Design	03/20/24
Implementation	04/14/24
Testing	05/01/24

4.2 Project Monitoring and Control

[The following is a checklist of items to consider:

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

- *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*
- *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.]

4.3 Quality Control

Defects will typically be found by the **implementer/tester** and communicated to the **lead developer** and the **project manager** will be notified if a deadline would be postponed due to one. The **project team members** will work together to eliminate difficult/big defects during any milestone of the project.

Any defects found during review which are not corrected prior to releasing for integration must be captured and reported so that they are not forgotten.

4.4 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.

4.5 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.

GitHub will be used to keep track of changes and new implementations and should be a sufficient backup for our project.

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*

<Project Name>	Version: <1.0>
Software Development Plan	Date: <12/22/24>
<document identifier>	

Guidelines, Design Guidelines, and other process guidelines. The text that follows is provided as an example.]

The project will follow the UPEDU process. The UPEDU website describes the roles in this plan.

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