SuperCoding

Boogle
Software Development Plan
Version 1.0

Revision History

Date	Version	Description	Author
02/23/2024	1.0	First plan development	SuperCoding
02/25/2024	1.1	Presubmission Revisions	SuperCoding

Table of Contents

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 defines the scope of the project, as well as the purpose of the product. It is the top-level plan generated and used by managers to direct the development effort.

The following people use the *Software Development Plan*:

- The **team leader**, Mr. Dodge, uses it to plan the project schedule and resource needs, and to track progress against the schedule.
- The **project team members**, Mr. Richards, Ms. Rong, Ms. Isenburg, and Ms. Phan, use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

The Software Development Plan will be assembled in large part by the Team Administrator, Mr. Richards.

1.2 Scope

This *Software Development Plan* describes the overall plan to be used by the Boogle project, including deployment 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

See the Project Glossary.

1.4 References

- Iteration Plans

  2024-EECS348-project-description (By: Dr. hossein Saiedian)

- Development Case

  N/A

- Vision

  Our vision is to create a Boolean simulator that will provide accurate results for both simple Boolean operations and truth tables.

- Glossary

  N/A

- Any other supporting plans or documentation.

  N/A

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 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 function of Boogle will be to:

- Take in a Boolean algebra expression (e.g. "(A & B) | C")
  - Operations include OR (|), AND (&), NOT (!), XOR ($), NAND (@)
  - Include support for parentheses
  - Include literals "true" and "false" (e.g. "A | false")
  - Variables uppercase, literals lowercase.
- Take in specific values for each given variable
- Optionally, output a full truth table consisting of every combination of inputs and the expression's result for said inputs, with the selected case bolded. Alternatively, output only the selected line of the truth table.

2.2 Assumptions and Constraints

software/language limitations, student/work schedules, project deadlines, coding development, allocated time to project, new team dynamics

2.3 Project Deliverables

Deliverables for each project phase are identified in the Development Case. 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

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

3. Project Organization

3.1 Organizational Structure

Mr. Dodge is the Team Leader, in charge of coordinating meetings, maintaining deadlines, and delivering artifacts to the customer; Mr. Richards is the Team Administrator, in charge of most notetaking and documentation, alongside Ms. Phan; Ms. Isenburg is the Technical Leader, in charge of writing most of the actual code and designing the project architecture; and Ms. Rong is the Data Administrator and QA Engineer, in charge of making sure the code written is functional, bug-free, and meets the requirements as outlined by Mr. Dodge.

3.2 External Interfaces

N/A

3.3 Roles and Responsibilities

Person	**Unified Process for EDUcation Role**
Brandon Dodge	Role: Project Leader Contact: bdodge@ku.edu Job Description: Submit all work. Coordinate meetings. Lead team efforts. Manage project development. Availability: Monday(2:00pm-5:00pm), Tuesday(4:00pm-6:00pm), Wednesday(N/A), Thursday(4:00pm-6:00pm), Friday(2:00pm-6:00pm)
Jacob Richards	Role: Team Administrator Contact: jacob.richards@ku.edu

	Job Description: Note taking and documentation Availability: Monday(2:00-4:00 pm), Tuesday(N/A), Wednesday (2:00-4:00 pm),Thursday (N/A), Friday (11-4:00 pm)
Eliana Isenburg	Role: Technical Leader Contact: isenburg-eliana@ku.edu Job Description: Architectural design and authorship of most C++ code as required and deliver to QA for testing. Availability: Monday (12pm-7pm), Tuesday (8am-11am), Wednesday (3pm-8pm), Thursday (8am-1pm), Friday (2pm-8pm)
Xiaomin Rong	Role: Data Administrator/Quality Assurance Engineer Contact: xrong123@ku.edu Job Description:Assistant with the code building, ensure the complement of the coding, test the code, report the errors Availability: Monday (12pm - 2 pm, 3pm - 7 pm), Tuesday 4pm- 7 pm), Wednesday(6pm-7pm), Thursday(4pm-7pm), Friday(12pm -2pm)
Lisa Phan	Role: Team Administrator Assistant Contact: lisa.phan@ku.edu Job Description: assisting Team Administrator with following up to assigned tasks, helping preparing meetings and documentation Availability: Monday(12pm-4pm), Tuesday(4pm-6pm), Wednesday(N/A), Thursday(4pm-6pm), Friday(2pm-5pm)

Anyone on the project can perform Any Role activities.

4. Management Process

4.1 Project Estimates

Cost: $0

Time: 4-8 weeks

4.2 Project Plan

Week1: Project Plan (Inception) [Github, VisualStudios, Google Documents, Zoom]

Week2: Software Requirements (Inception - Elaboration) [N/A]

Week3: Software Architecture (Elaboration - Construction) [N/A]

Week4: Test Cases (Construction - Translation)

Week5: User Manual (Translation)

4.2.1 Phase Plan

N/A

4.2.2 Iteration Objectives

Inception: Establish a scope for the Boolean simulator to operate, feasibility, and deadline.
Elaboration: Develop the software architecture and thoroughly documented design.
Construction: Code the software, test, debug, and adjust for project satisfaction.
Translation: Finalize all work and prepare the project for deployment.

4.2.3 Releases

Software release 0.1: N/A

Software release 1.0: N/A

4.2.4 Project Schedule

Deadline 1 (February 25th 2024)	Project Plan
Deadline 2 (N/A)	Architecture
Deadline 3 (N/A)	Coding
Deadline 4 (N/A)	Testing
Deadline 5 (N/A)	Finalize/Deploy

4.2.5 Project Resourcing

N/A

4.3 Project Monitoring and Control

- *Requirements Management:*

  VisualStudios will be providing most of the control mechanisms through error detections and coded constraints/parameters.

- *Quality Control:*

  Quality control within the project will be maintained by the team through thorough examination of each other's work. Visual Studio, a reliable source code editor with comprehensive error detection and debugging features, will be our tool of choice for ensuring accuracy. Regular code testing will be conducted to uphold consistency and reliability.

  For version control management, we will employ GitHub, enabling us to track code changes seamlessly and facilitate collaborative development. Furthermore, meticulous documentation will encompass project plans, meeting minutes, conceptualizations, and alterations, ensuring clarity and coherence throughout the project lifecycle.

- *Reporting and Measurement:*

  N/A

- *Risk Management:*

  To mitigate risk, the approach taken will be through multiple steps of testing and thorough commenting done by both our Technical Leader and Data Administrator/Quality Assurance Engineer.

- *Configuration Management:*

  To process the tacks and monitor changes to the developed systems configuration data, which can include the version control, and its application in a particular software environment.

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 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.

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 QA engineer role.

5. Annexes

The project will follow the UPEDU process.

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

The original project description, as written by Dr. Saiedian.