Project Proposal & Management Report

[Company name] | [Company address]

5COM1079 – Software Development Exercise

Samuel Blinkhorne, Peter Feehan, Cameron Davidson

2023

Contents

[Managing Software Development Projects 2](#_Toc139011377)

[A Comparison of Common Development Strategies (SB) 2](#_Toc139011378)

[**Waterfall / V Methodology** 2](#_Toc139011379)

[Agile 3](#_Toc139011380)

[Boehm Spiral 4](#_Toc139011381)

[Project Management Framework and Development Strategy (ALL/SB) 5](#_Toc139011382)

[Project Outline 6](#_Toc139011383)

[Problem Statement & Project Background (PF) 6](#_Toc139011384)

[Client Requirements (ALL/PF) 6](#_Toc139011385)

[Business Case - Project Goals & Benefits (ALL/PF) 6](#_Toc139011386)

[General Aims & Objectives (CD) 6](#_Toc139011387)

[Project Management and Development Strategy 7](#_Toc139011388)

[Preferred Approach to Project Management and Development (SB) 7](#_Toc139011389)

[Provisional Project Gantt Chart and Task Schedule (SB) 7](#_Toc139011390)

[Project Risks (ALL / CD) 7](#_Toc139011391)

[Team Structure & Setup 8](#_Toc139011392)

[Team Members & Team Roles (CD) 8](#_Toc139011393)

[Software and Project Management Team Skills Matrix (CD) 8](#_Toc139011394)

[Development, Testing & Deployment 10](#_Toc139011395)

[Target Platform (PF) 10](#_Toc139011396)

[Development and Testing Platform (ALL / CD) 10](#_Toc139011397)

[Project Collaboration and Sharing (CD) 10](#_Toc139011398)

[Link to Online Repository (CD) 10](#_Toc139011399)

[System Requirements 11](#_Toc139011400)

[Existing System Use-Cases (ALL/CD) 11](#_Toc139011401)

[System Requirements 11](#_Toc139011402)

[Project’s Functional Requirements (ALL / PF) 11](#_Toc139011403)

[Project’s Non-Functional Requirements (ALL / CD) 11](#_Toc139011404)

[References 12](#_Toc139011405)

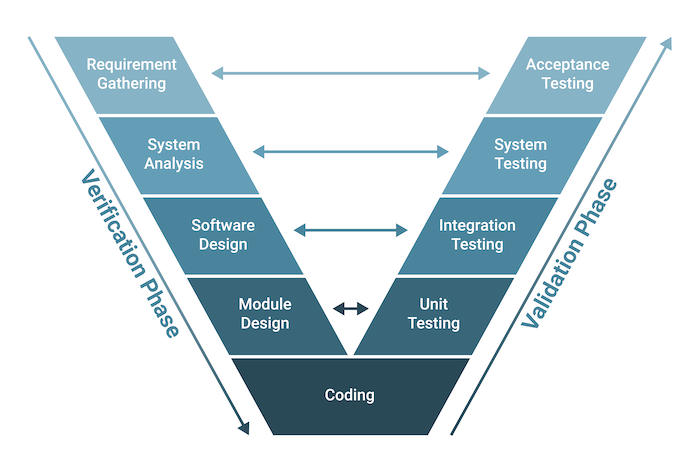
# Managing Software Development Projects

## A Comparison of Common Development Strategies (SB)

This section discusses three of the methodologies considered for the development of our software during this project and then states which development strategy is being used and justification for why this approach is the best for our team.

**Waterfall / V Methodology**

The waterfall method which can also be known as the ‘V’ method is one of the most established ways of carrying out software engineering development. It was coined in the 1970s by Dr. Winston W. Royce (Royce, 1970). It is a methodology that adopts a sequential approach to development meaning you are required to complete a stage of the development before moving on to the next stage. V methodology is an expansion on the standard Waterfall methodology that mitigates some of the disadvantages of the Waterfall methodology.



<https://builtin.com/software-engineering-perspectives/v-model>

Depending on verification phase you are in you take into consideration the opposite validation stage. This allows you come up with more effective requirements, design and analysis as you take into consideration how the system in the later stages will be tested at the various stages.

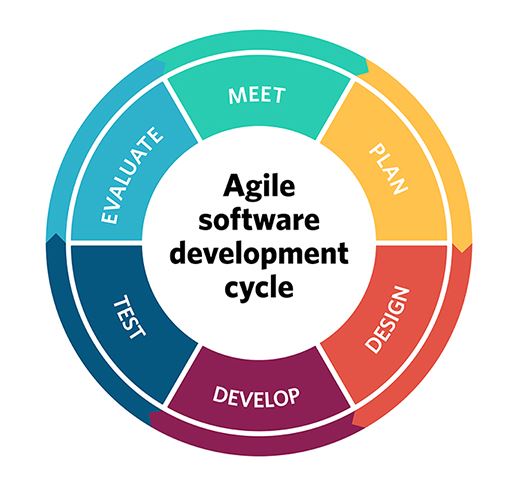
The Waterfall methodology is usually the most intensive when it comes to documentation. This can be an advantage for projects that have to be well documented or researched, such as in safety critical systems. But it can be a disadvantage for small team projects that don’t require a lot of documentation or for a project which needs to be very reactive to requirements changes.

Another advantage of this methodology is that everything has to be well established before moving onto the next stage this can be useful for developers as they have to have a full and complete design before they can begin development, so there is no ambiguity during development

### Agile

The Agile methodology was developed to make software development faster and more reactive requirements changes. It is a development strategy that shares similarities with iterative processes by takes it to a more extreme level. Agile has many aspects to it and can come in a variety of forms, including sprints, scrum meetings, use cases, Kanban and Extreme Programming (XP).

Agile methods rely on a lot of interaction with the customer which can add a lot of value to the development of the product as it allows for the customer to offer constant feedback on what they actually want as a product. The downside of this is if a customer is not responsive it can leave the team ‘in the dark’ with development and cause delays.

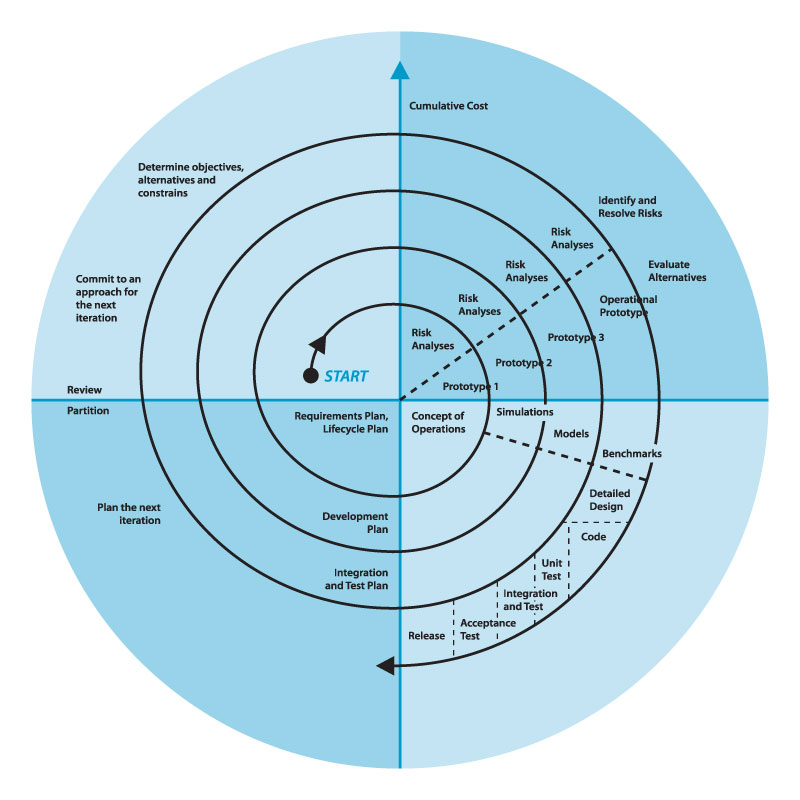


https://project-management.com/agile-software-development-methodologies/

### Boehm Spiral

The Boehm Spiral is a type of iterative development strategy, developed by Barry W.Boemn (<https://ultimatesdlc.com/spiral-model/>). It is a model of development that works on carrying out 4 stages in an iterative way till the final product is developed. The four Stages are Determine Objectives, Identify and Resolve Risks, Development and Test and Plan the Next Iteration.

This development strategy allows for larger projects to be broken down into smaller chunks that can be defined and then developed 1 after the other. Allowing a smaller team to develop a bigger system without getting overwhelmed or having to define the entire system at the start of the project. This development strategy can be considered a middle ground between waterfall and agile as it shares features of both



https://eternalsunshineoftheismind.wordpress.com/2013/03/09/the-spiral-model-5/

## **Project Management Framework and Development Strategy (ALL/SB)**

The chosen development strategy for our project is using the Boehm Spiral. This is the most appropriate development strategy for our team as it allows us to break down the task into stages. This will work better for us as a smaller development team, meaning we can more confidently take on the work by splitting it into workable packages. This strategy also allows us some level of flexibility in development and derived requirements change as we will be developing a number of prototypes through the development lifecycle where we can make adjustments to the original plans.

Boehm spiral will also allow us to carry out a more continuous testing strategy as we will be testing compared to if we used a waterfall method, for a small short-term project this works out more effective as we have a tighter time constraint so carrying out mass testing and bug fixing at the end of the project will not be possible. With a Boehm spiral we carry out a subset of testing and regression testing at each prototype stage meaning there should be a more polished demonstration model at the end.

The Agile methodology did not fit our development strategy as it relies on constant contact with the customer and feedback from them, which for our project will not always be available as we have been given set requirements at the beginning of the project.

# Project Outline

### Problem Statement & Project Background (PF)

### Client Requirements (ALL/PF)

### Business Case - Project Goals & Benefits (ALL/PF)

### General Aims & Objectives (CD)

# Project Management and Development Strategy

### Preferred Approach to Project Management and Development (SB)

### Provisional Project Gantt Chart and Task Schedule (SB)

### Project Risks (ALL / CD)

# Team Structure & Setup

### Team Members & Team Roles (CD)

### Software and Project Management Team Skills Matrix (CD)

|  |  |  |  |
| --- | --- | --- | --- |
|  | SB | CD | PF |
| **Technical Writing** | 3 | 2 | 2 |
| **Software Development Techniques** | 3 | 1 | 3 |
| **Project Management** | 4 | 3 | 3 |
| **C# Software Language** | 3 | 0 | 0 |
| **Database Management** | 1 | 1 | 2 |
| **Database Building** | 1 | 1 | 1 |
| **HTML Language** | 1 | 3 | 1 |
| **Web Development** | 1 | 1 | 2 |
| **Server Development** | 2 | 1 | 2 |
| **Requirements Development** | 4 | 3 | 3 |
| **Jira** | 4 | 0 | 4 |
| **Github** | 1 | 1 | 1 |
| **Weather Algorithms** | 0 | 0 | 0 |
| **Weather Forecast Techniques** | 0 | 0 | 0 |
| **Design Document Writing** | 4 | 2 | 2 |
| **UI Design** | 3 | 4 | 3 |
| **UX Design** | 3 | 3 | 2 |

# Development, Testing & Deployment

### Target Platform (PF)

### Development and Testing Platform (ALL / CD)

### Project Collaboration and Sharing (CD)

### Link to Online Repository (CD)

# System Requirements

### Existing System Use-Cases (ALL/CD)

### System Requirements

### Project’s Functional Requirements (ALL / PF)

### Project’s Non-Functional Requirements (ALL / CD)

# References

Royce, D. W. W., 1970. *MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS.* s.l.:IEEE WESCON.