Project Proposal & Management Report

2023

5COM1079 – Software Development Exercise

Samuel Blinkhorne, Peter Feehan, Cameron Davidson

[Company name] | [Company address]

Contents

[Contents 1](#_Toc140526547)

[Managing Software Development Projects 2](#_Toc140526548)

[A Comparison of Common Development Strategies 2](#_Toc140526549)

[**Waterfall / V Methodology** 2](#_Toc140526550)

[Agile 3](#_Toc140526551)

[Boehm Spiral 4](#_Toc140526552)

[Project Management Framework and Development Strategy 5](#_Toc140526553)

[Project Outline 6](#_Toc140526554)

[Problem Statement & Project Background 6](#_Toc140526555)

[Client Requirements 6](#_Toc140526556)

[Business Case – Project Goals & Benefits 7](#_Toc140526557)

[General Aims & Objectives (CD) 8](#_Toc140526558)

[Project Management and Development Strategy 9](#_Toc140526559)

[Preferred Approach to Project Management and Development 9](#_Toc140526560)

[Provisional Project Gantt Chart and Task Schedule 10](#_Toc140526561)

[Project Risks (ALL / CD) 11](#_Toc140526562)

[Team Structure & Setup 11](#_Toc140526563)

[Team Members & Team Roles 11](#_Toc140526564)

[Software and Project Management Team Skills Matrix (CD) 12](#_Toc140526565)

[Development, Testing & Deployment 13](#_Toc140526566)

[Target Platform 13](#_Toc140526567)

[Development and Testing Platform 13](#_Toc140526568)

[Project Collaboration and Sharing (CD) 14](#_Toc140526569)

[Link to Online Repository 14](#_Toc140526570)

[*https://github.com/pf20aai/weatherProject* 14](#_Toc140526571)

[System Requirements 15](#_Toc140526572)

[Existing System Use-Cases 15](#_Toc140526573)

[System Requirements 17](#_Toc140526574)

[Project’s Functional Requirements 17](#_Toc140526575)

[Project’s Non-Functional Requirements 17](#_Toc140526576)

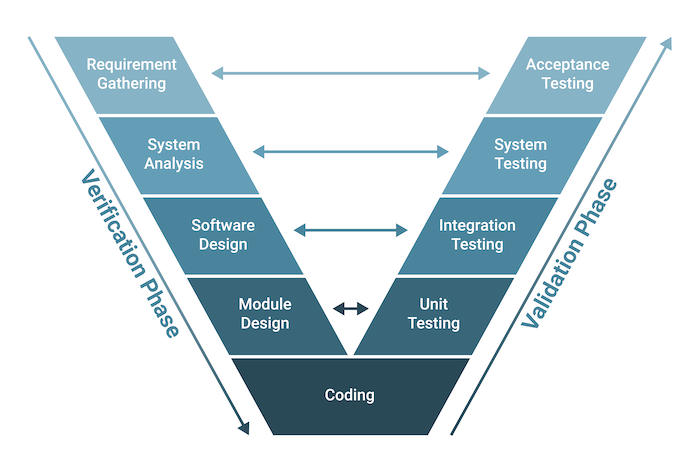
# Managing Software Development Projects

## A Comparison of Common Development Strategies

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 way of carrying out software engineering development. It was coined in the 1970s by Dr. Winston W. Royce (https://herts.instructure.com/courses/104211/files/7255858?module\_item\_id=3173003). 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.

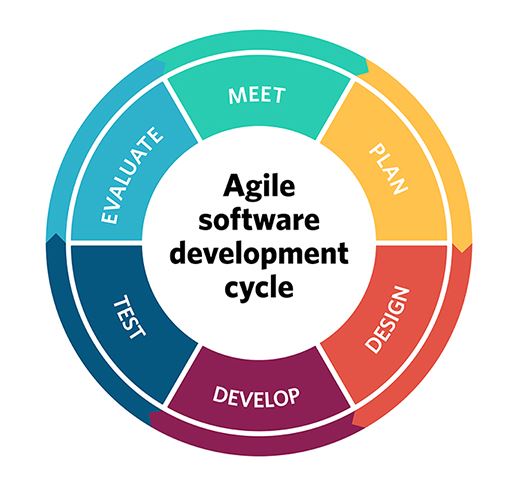
The Waterfall methodology is usually the most intensive when it comes to documentation. This can be an advantage for projects that have to be well documented or researched, such as in safety critical systems. But it can be a disadvantage for small team projects that don’t require a lot of documentation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

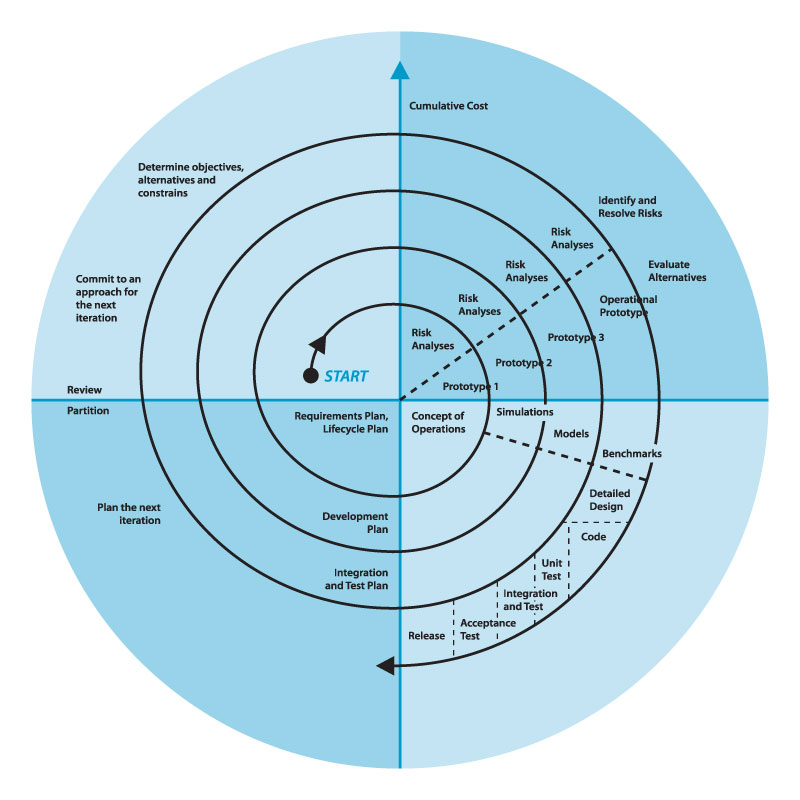


https://project-management.com/agile-software-development-methodologies/

### 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**

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

The client has asked us to create a new application that will forecast the weather for an area when provided data from the user. This differs from most well known and easily available weather predicting applications as it allows the user to input specific data about the current conditions where they are rather than relying on a black box system. Because this technology differs from the more commercially available products currently available the user experience must be intuitive and easy to use. That being said however it is likely that our target audience for this product will be looking for a more comprehensive system and with therefore be more open and willing to use a more complicated system for the pay off of more control. It has been requested that this project be accessible from a web application, this in addition to requiring a database of historic data means we must host the application on a web server. Finally we have been requested to allow logins for users so they may access this application. Because we also need to have administrators we must be careful to create a system that is both secure and keeps separation between different user privileges.

### Client Requirements

Definitions:

* Software: The software produced as a result of this project
* Must: These things must be included for the software to function.
* Should: These things should be included in the final software.
* Could: These things would be nice to have in the final software.

Requirements:

1. Software must run as a web application
2. Software must allow users to login
3. Software must allow users to input current weather data
4. Software must present the user with a prediction after they input their data
5. Software must base predictions on past weather events
6. Software should be accessible from Android 12 and Windows 10 devices
7. Software should store historical data about past weather events
8. Only software admins should be able to create or edit historical data
9. Users should have an account before they are allowed to input data or be shown any predictions
10. Software could be accessible anywhere

### Business Case – Project Goals & Benefits

GOAL 1

Our first goal is to produce a software project that is accessible through a web browser before the 1st of September. It is important that we produce a stable user experience through browser, as the users will only be able to access their predictions from a windows 10 or android device. Being accessible from a web browser is an effective business decision because it does not require the client to employ any third party distributor for the software, as long as they have some form of server hosting capabilities. Before the project begins the client must detail if they have the ability to host the software themselves or if they expect the server itself to be sourced as part of the project. Additionally if they are providing the server for hosting the project team would need it’s specifications. Failure to do this may result in software that is incompatible with the hosting server and reducing the effectiveness of the project.

GOAL 2

Our second goal is to create a piece of software that can make an accurate prediction of the weather basted on current weather data over 75% of the time, to be completed before the 1st of September. This is vital as it is one of the primary requirements set by the client. The software being able to make accurate predictions is important for the credibility of the business providing the software. Accuracy here refers to the frequency of which the software can make a prediction that is later shown to be true. The development team must therefore ensure they set aside enough time to research methodologies for creating a system that can predict weather as to use a method with a high percentage of accuracy. Failure to properly research this will result in a system that is inaccurate and will miss the key and most important objective of this project.

### General Aims & Objectives (CD)

**Aim:** To work effectively as a team

**Objectives:**

* Organise weekly check-up meetings to review progress, answer questions, and set new tasks
* Be contactable outside of the meetings should anything urgent come up
* Use Jira to set tasks and create a timeline for when tasks should be completed

**Aim:** High quality user experience

**Objectives:**

* Create an easy to navigate GUI (Graphical User Interface) so Users can sign in/up
* Create an interactive graphic for Users to see the predicted weather for that day

**Aim:** Accurate weather predictions

**Objective:** Set up a database to store weather data

**Aim:** Store User’s information securely

**Objective:**

* Do not store passwords in plaintext
* Use a hash function to encrypt passwords

**Aim:** Keep historic weather data up to date for reliable predictions

**Objective:**

* Have members of the Admin team responsible for keeping the data up to date
* Make sure that historic data is up to date by 9:00 am every day

**Aim:** Collect weather data

**Objective:**

* User’s inputted data will be stored and added to the historic weather database for future weather forecasts
* When adding to the weather database, verify new entries are correct

# Project Management and Development Strategy

### Preferred Approach to Project Management and Development

This project will be run using an adaption of the Boehm Spiral technique with some Agile features to allow for faster development due to the project being time critical. This project will run shorter Boehm spirals which will last around 14 days each. these spirals will contain the 5 key phases of the development cycle, Requirements Analysis, Design Concept, Development, Testing and Validation, Cycle Review & Risk Management.

As the team is quite small with a lot of tasks being carried out we are adopting a Kanban development strategy to track all of the work that needs to be carried out on the project during these cycles. The board will be maintained by the project manager who will analyse the board during each cycle review to see if any additional tasks need to be added along with adding in the next cycles worth of work. This allows the team to be adaptive in how the workload is tackled and makes sure that no tasks get missed or forgotten about. This also allows the team to prioritise work if the project becomes at risk of running out of time.

Using this technique will also allow our team and project to be flexible in its development of the prototype by allowing us to select requirements for each of the cycles and development them in an order that is seen to be fit while allowing us to update the designs, test spec and other documentation throughout the project. The reasoning for this development style is because we have a small development team it allows the project and requirements to be broken down making the development more manageable for a small team

Boehm spiral also has the advantage of catering for risk during development as at the start of each cycle while carrying out requirements analysis for the next stage of development the risks of the project are taken into account which means they can be mitigated at each stage when they arise, and due to the flexibility of the development style it allows you to react to the risks earlier in the development which can save time.

One of the risks on our project that this development strategy will help is there is also a lack of expertise in certain areas of the project as can be seen by section (XXXX) a Boehm spiral approach means that we don’t have to have high levels of expertise in areas from the very start of the project it allows the team to grow their skills as the project and requirements grow.

### Provisional Project Gantt Chart and Task Schedule

Our Gantt chart has split the project into 4 major milestones of our project as we are using the Boehm spiral approach these are:

Cycle 1: This will be the end of the first cycle of development where we will have an initial prototype with basic functionality and designs to work forward with

Cycle 2: This will be our intimidate prototype with higher fidelity and more advanced functionality with more refined design documents

Cycle 3: This will be our full prototype with completed designs and full testing

Demonstration: This is when the official demonstration to the customer is so will need everything completed by then

With regards to the task lengths of each of the cycles they take a very similar approach. The exception being that the initial development spiral is slightly longer in the design and development stages. This is due to the fact there is no basis we are working from on the development and design and it is being done from scratch, with the team working on ‘ramping up’ on workflow so will have slightly reduced efficiency.

### Project Risks (ALL / CD)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Risk | Impact | Likelihood | Fallout | Mitigation | Impact After | Likelihood after |
| Lack of experience in weather prediction algorithms | High | Medium | The site will incorrectly predict the weather | We all need to research weather prediction algorithms and how they work/can be implemented | Medium | Low |
| Missing a Project meeting | Medium | Medium | If someone misses the meeting, they will not be up to date with progress, the next tasks and deadlines | Minutes of each meeting will be recorded. All tasks will be updated on Jira with due dates, and assignees | Low | Low |
| Programming Issues | High | Low | Incorrect algorithms/ Broken GUI | Extensive testing throughout development | Medium | Low |

# Team Structure & Setup

### Team Members & Team Roles

Sam Blinkhorne (Project Manager):

As the manager of this project, Sam will oversee directing our team meetings, which tasks are prioritised and who will complete them. They will also manage and collate the content for any documentation made throughout this project.

Cameron Davidson (GUI Designer):

As the main graphic designer of the team, Cameron will be responsible for making sure that the end-user is able to use and navigate the app to an acceptable level.

Peter Feehan (Database Manager)

Peter will be in charge of the implementation and upkeep of the database used to store the weather data. As well as managing the development of the website.

Regardless of team role, each member of this project will be involved with writing documentation, programming the application, and decision making.

### 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 |

Based on the completed skills matrix, it was discovered that as a team, we have a collective understanding of nearly everything that our project will require.  We also found that each of our strongest skills can be categorized into a section of the project (Project management, Databases, Graphic Design, etc). This will allow us to split the tasks into groups that can be covered under each of these aspects. As the Skills Matrix shows, none of us have any experience with weather algorithms or forecasting techniques. This means we will have to do some extra research to investigate how each of these works and how we can factor them into our website

# Development, Testing & Deployment

### Target Platform

Our client requirements state that we are primarily developing a system accessible on both smartphone and PC. Therefore, to effectively suit these needs we require a web-server as a primary platform. This way we can effectively deliver a system that works for both Windows 10 and 11 devices as well as latest Android versions. To effectively achieve this we plan to write our platform in c# as we have team members who are very capable with the language and its flexibility will allow for easy creation of a functional and effective web-server. Additionally for our web client we intend to use HTML5 alongside CSS as multiple team members are effective with it though have little experience so the large amount of documentation will assist us in creating an effective functional system.

### Development and Testing Platform

In terms of programming, we will be using the latest version Microsoft Visual Studio to write our application. This is due to the majority of the team having experience in using this software and will therefore be able to do their tasks more efficiently and assist the team members who may have less experience. By using the same development tool, we can ensure that all the code we have created will not cause any issues regarding compatibility

We will be saving all our project content on GitHub. This will allow us to share content with our team members quickly and effectively.

As an add-on from GitHub, we will be also using GitKraken, a GUI that allows us to separately download project files, create branches, and make changes, without interrupting the active version of the project. This means we are able to carry out tests at each stage of development without causing issues to the work that has already been committed to the final product.

### Project Collaboration and Sharing (CD)

### Link to Online Repository

### *https://github.com/pf20aai/weatherProject*

# System Requirements

### C:\Users\davidsca\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\F8E10076.tmpExisting System Use-Cases

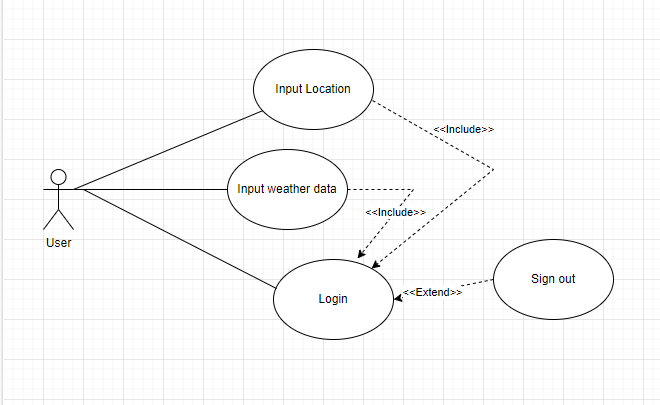
An Admin accessing the Weather Database

Actor: Admin

Goal: To Create, view, edit, or delete weather data

Precondition: Authorised user has logged in as an Admin

Summary: Admin user can CRUD the prediction weather data



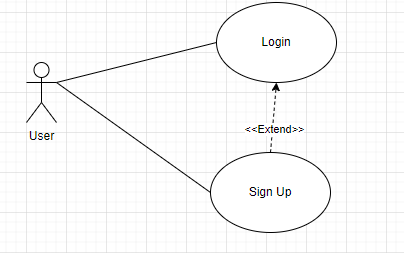
User inputting data

Actor: User

Goal: To enter current data to receive a weather prediction

Precondition: The User has already created an account

Summary: A User can login, and input their current weather and location data to get a prediction

User logs in/Creates a new account

Actor: User

Goal: To allow a user to log into their account. To allow a new user to create a new account

Precondition: None

Summary: A User will be able to log in to their account. If this is to fail, the User is given the option to sign up and create new account

### System Requirements

### Project’s Functional Requirements

* Users should be able to make an account using an id email and password
* Users should be able to login using their account
* Users should be able to input detail about the current weather conditions including:
  + location
  + pressure
  + humidity
  + temperature
* Admins should be able to input data into the database
* Admins should be able to reset users passwords
* Users shall be shown a prediction of weather after inputting data

### Project’s Non-Functional Requirements

1. Users shall be able to access the website from a Windows 10 PC
2. Users shall be able to access the website from any Android 12 mobile device
3. The database shall be able to store data including temperature, pressure, location, humidity, etc
4. User details should be stored securely
5. The weather prediction output shall be based on historic weather data
6. The user shall receive their weather prediction within 1 minute of inputting their current weather data
7. The user will need to have an account before they are able to input weather/location data
8. The Weather Prediction will show the predicted weather for that day only
9. Weather data from the previous day becomes historic weather data by 9:00am the next day
10. Only Admins will be able to access/edit the historic weather database