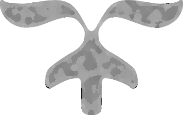


GROUP 4 PROJECT PLAN

Company Financial Analyzer



A purple and white logo

Description automatically generated

CS3343 Software Engineering Practice

SEMESTER A 2024

**Table of Contents**

[1. Introduction 2](#_Toc178706920)

[1.1 Project Background 2](#_Toc178706921)

[1.2 Project Stakeholders 2](#_Toc178706922)

[1.3 Project Team Organization 3](#_Toc178706923)

[2. Project Details 4](#_Toc178706924)

[2.1 Project Objective 4](#_Toc178706925)

[2.2 Project Scope 4](#_Toc178706926)

[2.3 Project Constraints 4](#_Toc178706927)

[2.4 Project Tools 5](#_Toc178706928)

[2.5 Software Development Methodology 5](#_Toc178706929)

[2.6 Work Breakdown Structure 6](#_Toc178706930)

[2.7 Gantt Chart 7](#_Toc178706931)

[2.8 Activity Diagram 8](#_Toc178706932)

# Introduction

## 1.1 Project Background

According to Gartner, an American consulting and technology firm in Stamford, concludes that on average, companies lose an estimated 12.9 million dollars that can be attributed to poor data quality. Poor data quality, in this case, includes incorrect calculations, incomplete reports, mixed-up data, etc. This is especially concerning since the company’s top management will make decisions based on these data reports, which sets the company's direction. This explains why the numbers that companies lose due to this is so big. Faulty data can cause companies to put their resources incorrectly, allocating them to already failing projects, for example. Not only does this make them waste their time and talents on a sinking ship, but it also makes other projects not get their talents and time needed to make them work. While this might at a glance seem to not be our problem as a student working on placement right now, the reality is that wrong resource allocation might result in its employees being put into different projects that they might not necessarily like, or even worse, handling multiple projects at once due to insufficient time. Since no one likes to work overtime, hopefully, you are more aware of the importance of correct data reporting.

This is where our program can come into place. Instead of manually compiling together all of the reports from each department, checking each and every project and its corresponding budget and revenues, and making sure all of the numbers go where they belong, we can instead put a singular text file and let the program do all the work. This can save people long gruelling hours of manual logging, not to mention the anxiety of the possibility that there might be a wrong input somewhere. All departments can share a file where they will record all of the expenditures and types of projects that they are currently working on. This can also increase supervision in each department which may ensure more transparency between each department. Consequently, this will allow top management to be able to make a better decision and save a lot of money that will ultimately trickle down to ordinary employees and shareholders.

## 1.2 Project Stakeholders

This project plan further divides its stakeholders into internal and external stakeholders. It’s a little bit tricky since our program is intended to be used by the company itself, as well as other companies. Then, for simplicity’s sake, the owner will be the top management, and its users will be the owner’s department's accounting lead. Detailed information can be found in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Stakeholder | Internal/External | Responsibilities |
| 1 | Company Owner | Internal | * Gives standard on what the financial report should look like. |
| 2 | Department lead Accountant | External | * Provide and input correct data |
| 3 | Project Team | Internal | * Develop a working software * Adapting to user’s requirements and comments * Flagging potential errors (e.g. Date out of bound, unrecognized department) |

## 1.3 Project Team Organization

This project is a work of collaboration between 6 members. Like all teams, we all had a part to play in creating this project. This section will delve into what our roles are.

|  |  |  |
| --- | --- | --- |
| No. | Name | Role |
| Management Oriented | | |
| 1 | WINOTO Dylan Kenneth | Project Manager |
| 2 | ZHANG Fanlu | Assistant Project Manager |
| Technical Oriented | | |
| 1 | WANG Jinjie | Testing Engineer |
| 2 | HO Ping Hong | Co-Developer |
| 3 | WANG Shenglin | Co-Developer |
| 4 | ZENG Yuhan | Co-Developer |

Further clarifying the role, for our group, Project Managers are responsible for overseeing the entire project, setting objectives, and tracking progress in reports. The project repository will also be made by the PM. As there are a lot of things to track and reports to be made, one more person is needed, which is where the assistant project manager comes in. Moving on to the technical team, developers are responsible for creating the actual project that users will use. Lastly, the testing engineer is responsible for making the test cases and reporting bugs/errors to the rest of the team.

The word “Oriented” is used because both management-oriented teams and technical-oriented teams will do some of the tasks from other teams; PM doing project development and developer doing reports.

# Project Details

## 2.1 Project Objective

Our project aims to minimize user input and maximize user output, which is why this program should be able to generate a lot of information from just one text file. This program is expected to create a company financial report containing the expenditure of all the different departments and their top 3 spending categories, respectively. Additionally, this program will also be able to take in profit reports from the company's product lineup. This will in turn allow the program to calculate the profit or loss of the entire company. Lastly, this program will allow its users to generate a monthly report and a quarterly report in addition to the usual annual report, allowing its users to gain more information.

## 2.2 Project Scope

This project encompasses the entire technical aspect which includes coding both the framework and the details, finding and debugging bugs, creating and testing multiple cases, and also potentially adding features. Moreover, non-technical aspects such as recording test results, writing reports, and creating presentations are also under this project’s scope. However, graphical user interface design, program updates, and security concerns fell outside the scope of this project.

## 2.3 Project Constraints

Due to the limited time, knowledge, and resources present, this project has a few constraints which will be detailed below:

1. User must use a text file; Unfortunately, even if the contents are exactly the same, this program cannot accept other file formats such as pdf, docx, word, excel, etc. This is due to the Java scanner function which will be used.
2. Java application environment will be the only kind of user interface; As GUI lies outside the scope of this project, additional graphical user interface will not be made, which might reduce the comfortability of the user.
3. The program results will not include any visual data; Due to limited knowledge, the program will only output a simple series of text and numbers. This might also reduce user comfort, especially since conventional financial reports are filled to the brim with bar graphs and pie charts
4. File must follow a very strict format; Due to dealing with a machine, any slight mistakes in the format will result in an error. While our program will give out specific error messages to help the user fix its report, the program will only work if and only if the text file follows all of the formatting rules that we have set.

## 2.4 Project Tools

Below are all of the tools necessary for creating this project

|  |  |  |
| --- | --- | --- |
| No | Tools | Usage |
| 1 | Eclipse IDE | Program Developing Environment |
| 2 | GitHub | Program Version Controller |
| 3 | JUnit4 | Program Tester |
| 4 | JDK | Program Language |
| 5 | Google Drive | Gather all the necessary document |
| 6 | Google Docx | Review people’s documents / Document collaboration |
| 7 | Microsoft Word | Writing Documentation |
| 8 | Notepad | User File Input (also used for tests) |
| 9 | Visual Paradigm | Create Sequence and Activity Diagram, Use case, etc. |

## 2.5 Software Development Methodology

This project will be developed by using the Agile Methodology. To further explain, the agile method involves splitting the project into multiple phases. Think of each phase as a whole mini project, complete with objectives and tests. Each phase (or cycle) is called a sprint. Below is an image to help visualize the agile concept:

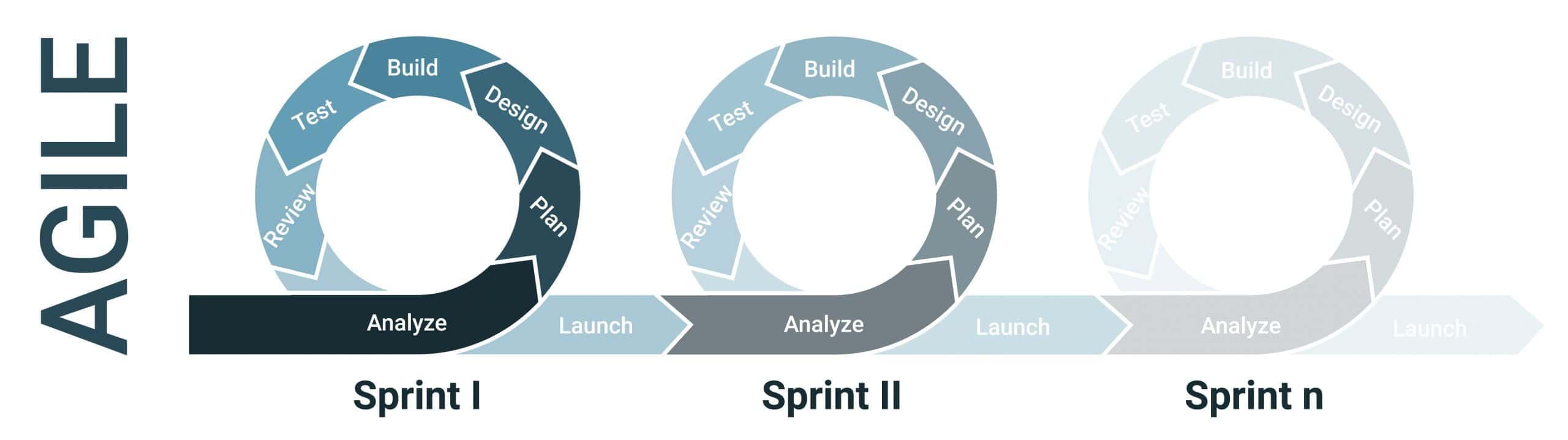


Figure : Agile Method Chart

The Agile method is chosen for 2 reasons. The first reason is to find and eliminate bugs quickly before a new feature is added. As can be seen from the image, each sprint will include a test and a review. This will ensure that before new features are added, the program itself will be free of bugs. Moreover, this has the additional benefit of the team to discuss the feasibility of the next phase, whether some things need to be slightly adjusted or changed completely. Both of these increase flexibility for the developers and make sure that the project is completed on time.

The second reason is to ensure all of our members can work as soon as possible. If a conventional model, like the waterfall model is used, it will result in the testing engineer having to wait until the very end of the project to create tests. Besides this, debugging will be significantly harder as it can only be done once the project is near its completion. This will result in the report being halted as well.

## 2.6 Work Breakdown Structure

A diagram of a work breakdown

Description automatically generated

Figure : Work Breakdown Structure

The image above is the work breakdown structure for this project. While at first glance 6 sprints sound like a lot for a single project, the decision ultimately comes from the team’s desire to save time in debugging in exchange for more features. As can be seen from the image, the software development is divided into 3 as we think that debugging while we have already implemented 4 different functions might be very challenging. Besides, having 6 short sprints and 4 long ones does not really have that much of a difference in regards to the time it takes to complete.

## 2.7 Gantt Chart

A screenshot of a graph

Description automatically generated

Figure : Gantt Chart

The image above is the Gantt chart of our project. One of the reasoning behind why the deadline for the use case diagram, class diagram, and sequence diagram is longer than the deadline for making the algorithm is that our team may need some time to think about the very details of the project. Also because we plan to finish most of the management in sprint 2, each member will deal with at most 1 project. On the other hand, in sprints 3-5, as there won’t be a lot of management operations going on, 3 developers with the assistance of both the project manager and assistant project manager will focus on implementing the functions of the program, which although difficult, with the help of 5 people, 10 days should suffice. Lastly, as most of the report is continuously updated on sprint 2-sprint 5, the last sprint will most likely be focused on making the final presentation and polishing the reports for the final submission. All in all, the project should be finished on 17th November 2024.

## 2.8 Activity Diagram

A diagram of a process

Description automatically generated

Figure : Activity Diagram

The image above is the activity diagram for our project. As stated previously, our project aims to minimize user input, which is the reason why our activity diagram is very simple. As long as the user has the correct format file, the user only needs to input the time at which the financial report will be on.