FIT2101

SOFTWARE ENGINEERING PROCESS AND MANAGEMENT

PROJECT PLAN

GROUP 2

Fahad Ahmad	30685265
Jonathan Wong Leong Shan	31435297
Lam Xin Le	31843549
Lim Zhen Kang	31886876
Matin Raj Sundara Raj	32124260
Yahoo Yang	31868282

Contents

1.0 Introduction and Purpose

This document serves as the project plan for a web application, TechTa, developed for Widgets Are Us (WAU), a contract software development firm. The primary goal of the application is to provide users with real-time or static data manipulation and visualization through the deployment of interactive UI elements. The team assembled for this project is composed of 6 undergraduate students with Software Engineering and Computer Science backgrounds, enrolled at Monash University Malaysia.

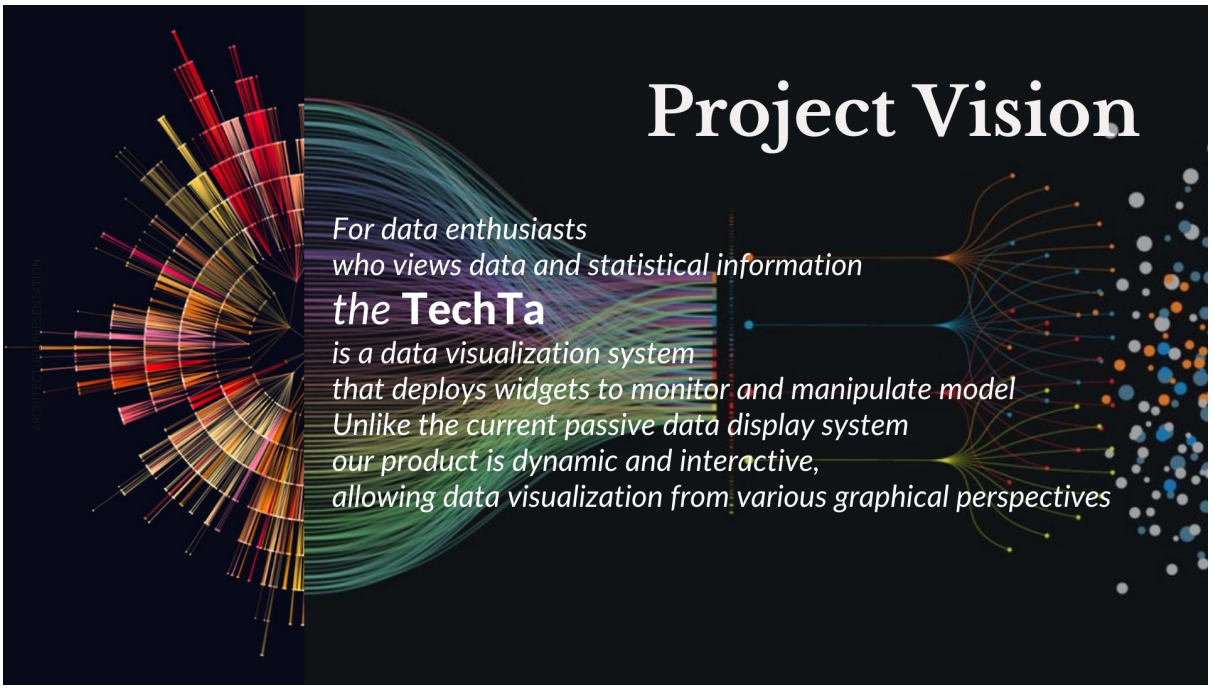
The following documentation is a Project Plan, which is a formal, approved document that defines how the project is executed, monitored, and controlled. This document is intended for the members of this project, as it provides a 7-section preliminary guide to the managerial and technical aspects that are involved in the development of this web application. It aims to keep those involved informed on this project and how it will be conducted.

The first section gives a brief introduction to the nature of this project and the contents of this document. It also establishes a clear vision that sets the direction of this project. The second section gives a detailed background on the situation and what the intended use of the application is. It also lists the deliverables of this project and its associated deadlines. The following section then describes the organizational structure of the team, where the roles & responsibilities for each member involved in this project are tabulated.

In addition, section 4 lays out the framework for how the team will conduct its software development process starting from the process model, which explains the core policies implemented during this project. Next, in section 5, general requirements for testing and QA are highlighted through a list of criteria that signal the completion of a task or a feature implementation. Sections 6 and 7 then explain how tasks are allocated among team members, how their progress is tracked, how product backlog is stored and managed, and how time spent on those given tasks is tracked.

This project management plan is an evolving document and as such, will be subjected to changes and updates in certain sections as the project progresses.

1.1 Project Vision



2.0 Project Information

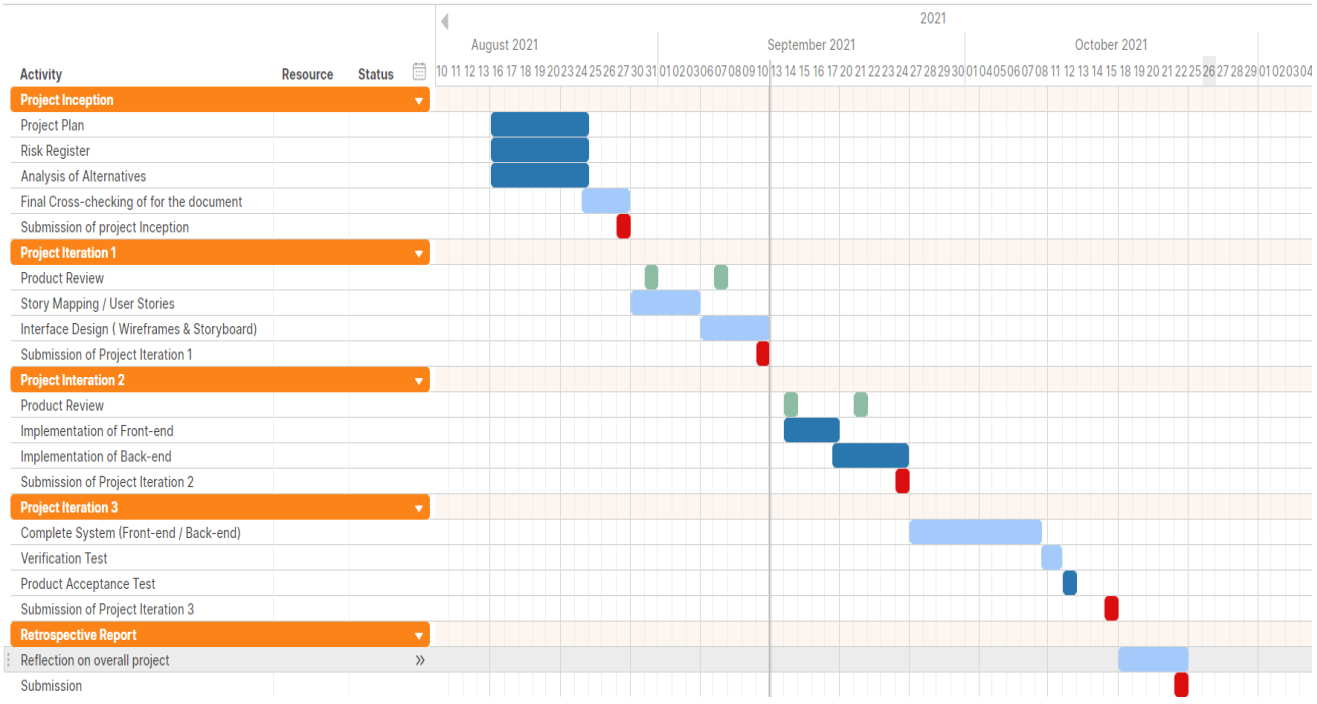
2.1 Background and intended use

In order to keep up with the fast-changing technological landscape, Widgets Are Us (WAU), formerly known as Zenith Widgets, have switched their business from making traditional mechanical widgets to a contract software development firm. They specialize in providing bespoke software solutions for their customers, across a wide range of application domains that deal with real-world problems or satisfy business needs. Nevertheless, their widget-making heritage is still kept alive through the incorporation of widget UI elements that allow for monitoring and control of models and systems in the software.

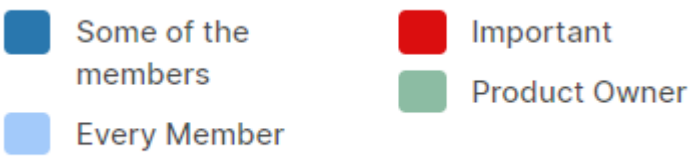
As a team hired by WAU, the task given is to design, develop, implement and test the systems that would be shipped out. However, due to ongoing confidential negotiations between WAU and its clients, the project team is not provided with the detailed business application of the software. Nevertheless, the team will use a multi-layered MVC (model-view-controller) system architecture which provides the opportunity to reuse or design novel widget features according to WAU's client's needs.

Upon attending multiple preliminary meetings with the client (WAU), the team has decided to develop a full-stack web-based application that allows users to manipulate and visualize live and static datasets through the incorporation of interactive widgets. The web application is targeted at computers and smartphones, and it allows for simultaneous access from multiple users. The end-users of the application can be company personnel or the general public, though an account would be required to set up. The purpose of this application is to provide users with meaningful insights into a particular data set. This is done through the incorporation of interactive widgets, which may display graphs and other visualization methods with toggles to filter the data. Some calculations and statistical processing will be done on the data to achieve this. Not only that, the data can be drawn from a variety of sources such as static datasets, APIs, online feeds, or real-time data. All of which are customized according to the client's needs.

2.2 Deliverables / due dates



Legend :



The project starts on 16 August 2021(Week 4) and ends on 22 October 2021 (Week 12). It is to be split into 5 parts which are Project Inception, Project Iteration 1, Project Iteration 2, Project Iteration 3 and Retrospective Reports. Each part of the project contains different tasks and milestones to be achieved. The tasks required to complete each part of the project have been shown in the figure above. There are also legends to indicate which tasks should the members involve themselves in.

3.0 Personnel/HR Management

3.1 Personnel

Fahad Ahmad
Contact number: +92 32240-30777
Email address: fahm0012@student.monash.edu

Jonathan Wong Leong Shan
Contact number: +6017-380 9738
Email address: jwon0100@student.monash.edu

Lam Xin Le
Contact number: +6010-6691138
Email address: xlam0003@student.monash.edu

Lim Zhen Kang
Contact number: +6017-8815315
Email address: zlim0036@student.monash.edu

Matin Raj
Contact number: +6011-2155 7511
Email address: msun0041@student.monash.edu

Yahoo Yang
Contact number: +6017-3559 683
Email address: yyan0227@student.monash.edu

3.2 Roles and Responsibilities

Fahad Ahmad

Roles	Description
Front-end Web Developer	- Specialize in front-end web designing and logic - Determine and design the structure and design of the web pages using front-end web development tools (HTML, CSS and JS)
Application Tester	- Conduct unit testing and integration testing along the coding process - Carry out manual and automatic testing to assess and evaluate the software to ensure it works as expected from usability and technical aspects
Lead UI Designer	- Research on the front-end web templates for the project to lead and oversee the team's UI strategy - Edit and modify user interface design with respect to the front-end developer team - Assess the team's UI design using various design principles to make the UI more user-friendly and appealing: 1. Gestalt Principles 2. Visual Hierarchy with Figure-Ground 3. Norman's Principles 4. Shneiderman's 8 golden rules
Technical Documentation Editor	- Write and completes documents for whether internal or external use

Jonathan Wong Leong Shan

Roles	Description
Front-end Web Developer	- Specialize in front-end web designing and logic - Determine and design the structure and design of the web pages using front-end web development tools (HTML, CSS and JS)
Application Tester	- Conduct unit testing and integration testing along the coding process - Carry out manual and automatic testing to assess and evaluate the software to ensure it works as expected from usability and technical aspects
UI Designer	- Create interactive drafts of front-end web design such as wireframes and storyboard - Edit and modify user interface design with respect to the front-end developer team
Technical Documentation Lead Editor	- Prepare document skeleton before meetings for smooth documenting process - Write and completes documents for whether internal or external use

Lam Xin Le

Roles	Description
Back-end Web Developer	- Build and maintain web applications by handling server-side web application logic - Responsible in helps browsers in communicating with the databases to perform CRUD operations on the database - Manage and deploy web applications on hosting platform
Developer Team Lead	- Manage the coordination of the entire development team - Supervise and keep track of the progress in front-end and back-end

	web development to offer advice or help to the developers
UI Designer	- Create interactive drafts of front-end web design such as wireframes and storyboard - Edit and modify user interface design with respect to the front-end developer team
Application Tester	- Conduct unit testing and integration testing along the coding process - Carry out manual and automatic testing to assess and evaluate the software to ensure it works as expected from usability and technical aspects
Technical Documentation Editor	- Write and completes documents for whether internal or external use

Lim Zhen Kang

Roles	Description
Back-end Web Developer	- Build and maintain web applications by handling server-side web application logic - Responsible in helps browsers in communicating with the databases to perform CRUD operations on the database - Manage and deploy web applications on hosting platform
Scrum Master	- Facilitate discussion and interactions during meetings, helping the team to achieve consensus - Organize sprint activities and select priorities for sprint by helping product owner to manage backlog
UI Designer	- Create interactive drafts of front-end web design such as wireframes and storyboard - Edit and modify user interface design with respect to the front-end developer team

Application Tester	- Conduct unit testing and integration testing along the coding process - Carry out manual and automatic testing to assess and evaluate the software to ensure it works as expected from usability and technical aspects
Technical Documentation Editor	- Write and completes documents for whether internal or external use

Matin Raj

Roles	Description
Front-end Web Developer	- Specialize in front-end web designing and logic - Determine and design the structure and design of the web pages using front-end web development tools (HTML, CSS and JS)
Product Owner	- Set the direction for product development or project progress - Attend client meetings to gather client requirements and communicate them with everyone involved in the project - Updates the requirements into the scrum product backlog and ensure that every detail translates to the vision
UI Designer	- Create interactive drafts of front-end web design such as wireframes and storyboard - Edit and modify user interface design with respect to the front-end developer team
Application Tester	- Conduct unit testing and integration testing along the coding process - Carry out manual and automatic testing to assess and evaluate the software to ensure it works as expected from usability and technical aspects
Technical Documentation Editor	- Write and completes documents for

	whether internal or external use

Yahoo Yang

Roles	Description
Back-end Web Developer	- Build and maintain web applications by handling server-side web application logic - Responsible in helps browsers in communicating with the databases to perform CRUD operations on the database - Manage and deploy web applications on hosting platform
Minutes Taker	- Attend client meetings with the product owner - Take down minute meetings to summarize the outcome of internal discussion and client meetings
UI Designer	- Create interactive drafts of front-end web design such as wireframes and storyboard - Edit and modify user interface design with respect to the front-end developer team
Application Tester	- Conduct unit testing and integration testing along the coding process - Carry out manual and automatic testing to assess and evaluate the software to ensure it works as expected from usability and technical aspects
Technical Documentation Editor	- Write and completes documents for whether internal or external use

4.0 Process Model

The process model to be used in this project is the Scrum methodology. Scrum is a framework utilizing an agile mindset for developing, delivering, and sustaining products, with lesser emphasis put on the production of documents than traditional process models.

4.1 Reasons Why Scrum is Chosen

Due to signing of MoU by the team's company, detailed requirements for the business application that the team is developing will not be disclosed to the team. There will be a client meeting with the Product Owner once every week to relay the latest product requirements and provide feedback regarding the team's developed product features to date. Therefore, the product requirements are likely to be changed frequently before the end product is delivered, the project team may then build the solution in small increments, typically working in two-week sprints/iterations and acting on feedback as the team goes along. Not to mention that the current project is typically short in nature, which is around 3 months. In short, using scrum methodology allows the adaptation of changes to the requirements with regards to the client.

In addition to that, to satisfy the product versatility, which is one of the client's expectations, the team will have to work out a full-fledged solution as described earlier in this Project Plan. As such, the team is made up of 6 developers, each with different backgrounds and expertise in the aforementioned technical aspects. This is said to be a cross-functional, self-organizing team of size 6 members, thus one of the best process management models for this scenario is Scrum.

4.2 How Scrum will be Implemented in the Project

4.2.1 Plan & Estimate Phase

4.2.1.1 Step 1

The product owner in the team will be getting product-related inputs and feedback from the client representative at least once every week (every Tuesday 1pm to 4pm MYT). He will then decide which of the client requirements should go into the Product Backlog and refine it based on each of the items' priority.

4.2.1.2 Step 2

At the end of each sprint, before the team begins the next sprint iteration, the team will hold a sprint planning meeting and the Product Owner shall brief all team members on the content of the Product Backlog. After that, the team will modularise high-level user stories and requirements into smaller detailed tasks. The team shall then discuss, as a team,

which of those Product Backlog items should go into the next sprint backlog. This usually happens in two phases, however, due to the tight schedule in this project (7 weeks remaining), the team shall compress it into a single phase process. This means that as soon as the briefing by the Product Owner and the modularisation by the team are done, the team will have to decide on items to be included in the upcoming sprint iteration.

4.2.2 Implement Phase

4.2.2.1 Step 3

Once the team has finalised the new sprint backlog, each team member will be assigned a portion of the sprint backlog according to their skill sets and then start a new sprint iteration. Each sprint iteration will be at most 2 weeks long, this is because the team is expected to have at least 3 sprint iterations before the delivery of the final working product. This is to ensure the team will have enough iterations to not only produce a runnable artefact but also a well-tested and well-designed artefact.

4.2.2.2 Step 4

In each of the sprint iterations, the team will be holding a standup meeting every 2-3 days, instead of a daily standup meeting, to keep everyone updated on each other's progress and possibly seek help from each other on any issues that any of the team members are having. The team will not be doing daily standup meetings because all the team members are not working full-time in this project and each member has other obligations outside of this project to attend to.

4.2.3 Review & Retrospect Phase

4.2.3.1 Step 5

At the end of each sprint, the team shall collectively conduct a product review and a sprint retrospective at the same time. The aim of the product review is to validate the product changes the team has made in that sprint iteration to conform to what the team members have all agreed on at the beginning of the sprint iteration. On the other hand, the sprint retrospective aims to pinpoint the issues and weaknesses with regards to the management of the development process the team encountered during the sprint iteration and to arrive at solutions which the team shall employ in the next upcoming sprint iteration.

4.3 Policies Tailored to Scrum Process Model

The standup meeting held every 2-3 days is led by the Scrum Master and should not last more than 30 minutes. Each team member will have to answer 3 questions:

- How are things going on your side?
- How are you going to proceed in the next few days?
- Do you have any issues on your side?

For each of the standup meetings as well as the ones between the Product Owner and the client representative, the minutes taker in the team should take note of all the conclusions, solutions arrived, issues arose as well as agreements made during the meeting. The notes should be available on the shared working space as soon as possible so that everyone in the team can refer to it.

It will be every team member's responsibility (not only the Product Owner) to make sure that all items in the product backlog are fine-grained. All members of the team shall participate in grooming the product backlog from time to time. Should an item be found to be unclear, the Product Owner should be informed about it as soon as possible. This is so to ensure that the team can very quickly decide on which of those items shall be moved into the next sprint backlog once a sprint Iteration is complete.

When in doubt about a proposed technical solution, the team shall create a spike and put it into the product backlog or sprint backlog depending on the urgency of the solution. If the spike goes straight into the sprint backlog, it shall be implemented as soon as possible to give clues to developers regarding the feasibility of the solution.

4.4 How Process Model Differs From Scrum

There are several aspects in the process model deployed by the project team that differs from Scrum itself. First of all, the team will not be doing Release Planning. The purpose of Release Planning is to let internal stakeholders, primarily software developers, to get an idea about the timeline of the project for when a product feature shall be implemented etc. Release Planning can be thought of as a roadmap to the final product. The reason why the team will not be doing Release Planning is because the team is certain that the final product will be shipped in 7 weeks later. On top of that, the software requirements in this project are volatile, subject to changes in the weeks to come and hence a roadmap with pre-defined milestones is not suitable in this case.

On the other hand, the project team would not be using Scrum Epics as well because there are no detailed product requirements at this stage yet. Scrum Epics requires the Scrum team to have a clear, well-defined goal product which can then be modelled as a

Scrum Epics. Scrum Epics serves the purpose of modularising a big chunk of a project into smaller shippable pieces. However , the project team at this moment still does not know about the exact details of the final product, thus a Scrum Epic in this case is not suitable.

5.0 Definition of Done

There should be some general requirements for a feature to be considered complete and ready to roll up so that everyone on the team agrees that the feature is completed and can switch focus to the next feature. This is known as the Definition of Done, acting as an agreement among the team members to avoid any conflicting view on the product that is said to be completed. For instance, Done may mean that a specific task has met all the client requirements, coded to standards, and can be executed with no errors and no known defects, peer-reviewed, pass all functional tests with unit test coverage > 85%, works cohesively with other portions of the program, and is well documented.

However, the Definition of Done may vary depending on the features implemented, based on its complexity and client requirements. As a general definition, the team will adhere to the following Definition of Done checklist, whereby a task is only deemed done when all the criteria in the list are fulfilled.

Definition of Done Checklist:

1. Client requirements are met
2. Code is complete
3. Code can execute with no errors and no known defects
4. Code is peer-reviewed and approved by all members of the team
5. All unit and functional tests with > 85% coverage are passed
6. Works cohesively with other portions of the program
7. Works perfectly across a wide range & variety of data
8. Well documented
9. Acceptance test passed and software is approved
10. Product Review for the demonstration of the current status of the product

6.0 Task Management

Task management is defined as the process of handling the entire life-cycle of a task, from planning to tracking, and finally to execution. It helps the teams to track tasks from the beginning, setting deadlines, prioritizing tasks, and assigning them to the appropriate people to complete them, returning the final product. In the case of software development for this project, there are three main aspects in the task management process to be taken into account, namely task allocation, progress tracking as well as backlog storage and management.

6.1 Task Allocation

In Scrum, task allocation involves a process known as sprint planning. The project team is required to allocate weekly individual tasks about the project accordingly. This is to ensure that everyone is aware of what to complete for the week, keep track of individual contributions - that everyone is contributing equally - and ensure the project is submitted on time. Scrum Master is responsible for facilitating the sprint planning meeting, ensuring that everybody's voice is heard, and to mediate in case of conflicting views. Note that there should be a maximum amount of time set for each sprint planning called timeboxing. For instance, the sprint planning meeting for a two-week sprint would be no longer than two hours.

The Product Owner shoulders the responsibility for deciding the content and priority of each product backlog entry, then determining what goes into the product backlog on the basis of business value. During the first phase of the sprint planning meeting, the Product Owner will give a detailed briefing on the highest-priority items in the product backlog with the team, answer any question about them as well as receive feedback on their feasibility from the developer team. The team will then break the high-level user stories and Product Backlog Items down into smaller, more detailed tasks, giving a clearer scope covering the functionality and features of the final product.

Then, the team will then decide which sets of tasks they will attempt to implement in this sprint by estimating the time required for each task. Then, the team commits to a selection of tasks that they expect to get done by the end of the sprint. Those selected sets of tasks get arranged into the sprint backlog. The team will then discuss to decide the person-in-charge for taking up the most appropriate task from the backlog based on their skills set as well as their availability.

The sprint backlog is similar in format to the product backlog but with a column indicating person-in-charge working on each task. The sprint backlog will be arranged in Trello, including the brief descriptions about each task, their respective deadlines as well as the name(s) of the person-in-charge for the task.

6.2 Progress Tracking

In project management, progress tracking is important because it demonstrates a real-time visual representation of the status of a project. As such, the project team will instantly know the status of different tasks and where most attention is needed. Not only does progress tracking help the team stay focused on priority to reaching the project goal, but it also aids in identifying potential obstacles and strategies on how to overcome them. Online tools such as project management and collaboration tool - Trello, cloud-based file storage solution - Google Drive, web-based Git repository - GitLab and cross-platform centralized instant messaging service - Whatsapp as well as regular meetings such as the Daily Scrum are used for progress tracking in this project.

6.2.1 With Online Tools

6.2.1.1 Project Management and Collaboration Tool - Trello

Trello offers features to decompose tasks as well as to add assignees, due dates and checklist to monitor progress. The task board in Trello is organised into a Kanban style where tasks can be divided into different stages for progress tracking. In Agile, a kanban is a signboard that allows the users to see implementation status at a glance. It can be used to show what's coming up next and what still remains to be implemented. It can have multiple columns to reflect the workflow of a task. For illustrative purposes, 'Todo', 'Working on it / Doing', 'Ready for Review', 'In Review', 'Ready to finalise', and 'Done'. Status of individual user stories or tasks is indicated by their presence in one of the columns.

The following table depicts the workflow that happens when a particular task is under a particular column on Kanban.

Kanban Column	Workflow
'Todo'	Planning of the task for a particular user story to be implemented
'Working on it / Doing'	When the person-in-charge working on a particular task assigned to him labelled in the 'Todo' column
'Ready for Review'	When the development for the task is completed by person-in-charge and the product is ready for check by the team members or leader
'In Review'	When the product is being reviewed to check for its completeness and correctness

'Ready to finalise'	When the product review is completed, then it is ready to be either finalized to be merged into main codebase
'Done'	Indicates that the task for the particular user story is done completely without a hitch

From the table above, it can be clearly seen that progress tracking is further simplified using Trello and whenever there is a change to the status of a task, a reminder email will be sent to the team members to update and inform them of the progress. Not only is the status updated, the assignee or reviewer can even add comments under each task card to be revised by any of them such as missing functionality, mistake or bug and any other useful information.

6.2.1.2 Cloud-based File Storage Solution - Google Drive

Not only does the project team have to make regular project contributions, which include, but are not limited to, writing documents such as the requirement and design documents, the team also needs to make preliminary notes and record the meeting minutes as a result of internal or client discussion. This is to ensure that the project will be completed in time and that the result will be of high quality. Therefore, Google Drive, which is a cloud-based storage solution that allows online file saving and can be accessed anywhere from any smart device, is used in progress tracking for written documents. With the use of Google Drive, team members are allowed to create a shared drive so that members can securely upload their files and receive those files and documents for editing and modifications.

For any reports or documents, Google Docs is utilised as it allows other members to attach comments for any work done, multiple members to edit at the same time and all contributions to be recorded. Google Docs prevents the loss of work as it has the capability of auto-saving all contributions and restoring past versions of work. On the other hand, for general task explanations or timeline, Google Sheets may be used to summarize the tasks for the particular week along with the deadline and task descriptions apart from the use of Trello in a table form. Multiple related sheets can be saved in a single file in the form of Google Sheets, allowing grouping of related information for easy access and understanding.

In short, Google Drive allows the tracking of work progress of each member to check the new changes made as well as who made the new changes. Thus, google drive with various tools for file systems is utilised for tracking the work progress and for the collaborators to edit and work on files, to see live changes, allowing the virtual side-by-side work.

6.2.1.3 Web-based Git Repository - GitLab

Git is a distributed version-control system for tracking changes in source code during software development. A team repository is cloned to the team members' local machines for editing and pushing to the online repository. It is designed for coordinating work among programmers as well as for tracking changes in any set of files. To allow the tracking of progress in members in terms of code development, GitLab, which is a web-based DevOps lifecycle tool that provides a Git repository manager providing wiki, issue-tracking and continuous integration and deployment pipeline features, is utilized by the project teams.

GitLab indeed provides an online storage space that can be shared among all team members for code storage. This is to ensure that everyone in the team is using the most updated version of codes and to incorporate each member's codes together for the web application to function. All team members are to commit any codes or changes into the team repository on a separate branch on the GitLab Server once their code development is completed. This can be done simply through Visual Studio (VS) Code, where team members first have to "pull" the codes from the team repository for receiving the latest version editing on local machines. Then, after updating their local version for editing, the team members can then commit and "push" any change made to the current version to update the version on their working branch in the team repository. After code review and multiple integration testing to ensure that the latest changes function correctly, the changes from the separate working branch can then be merged into the main branch. This indeed results in an efficient workflow and collaborations as every member of the team is allowed to work in parallel and concurrently for different tasks without the fear of losing the work.

In a nutshell, GitLab also allows the viewing of the complete history of changes committed to the group repository by each member and time is recorded with proper commit messages. Therefore, the progress of code development by each team member can be tracked. Note that the commit hash for each change made to the team repository allows the restoring of the previous code version if there is any bug to the current version.

6.2.1.4 Cross-platform Centralized Instant Messaging Service - WhatsApp

Even with the use of online tools for task tracking, there is still a need for having an easy-access communication channel among the team members to interact with each other for reporting task progress. In this case, Whatsapp is used instead of other mediums such as Gmail as it allows for fast responses and more convenient communication. By having a Whatsapp group chat among all the team members, each member can easily report their task status and answer any question from other team members to check up with their work progress.

Team members are required to respond to these messages within five hours during day-time, in consideration of online class hours, and within eight hours during night-time, in consideration of the sleeping time. If response time is exceeded, then either decisions will be made without their input. This helps to keep everyone on the same page as every member

should hold account for the task allocated to him or her and should be responsible in updating with one another about task status.

Whatsapp may also be used to schedule a meeting time for updating work progress or to request for task extensions. However, task extensions should be requested at least a day before the deadline and must be with valid reason. To prevent the issue, team members should ask for help earlier if they are facing difficulties and the team should set deadlines much earlier than project submission date to accommodate for late submissions and checking of work.

6.2.2 With Meetings

6.2.2.1 The Daily Scrum (or StandUp)

The daily scrum is a 15-minute meeting held at the same time and place everyday. It is sometimes called a standup and it is compulsory for all the team members to attend to report their work progress to each other. The daily scrum is considered a platform to provide better transparency, efficiency and clear work progress for each team member. It serves the purpose to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

The daily scrum is to be led by the Scrum Master and each team member is required to explain what he did the day before, what he plans to do today as well as what prevents him from doing the work (if anything). By focusing on what each member accomplished yesterday and will accomplish today, the team gains a better insight into what work has been done and what work remains. The daily scrum meeting is not a status update meeting in which a boss is collecting information about who is behind schedule. Rather, it is a meeting in which team members make commitments to each other. After the meeting, the Scrum Master takes action to get rid of anything blocking team members' productivity and to try to resolve any other issues that come up.

Bear in mind that daily scrum is often held as a very short meeting to keep everyone on the same page on the task status. It should not involve excessive meetings to avoid taking up too much of members' time. This is because the members should devote more time and energy towards the development stage which is usually more time consuming.

6.3 Backlog Storage and Management

A backlog is a prioritised list of the functionality to be developed in a product or service and it may be derived from the roadmap and its requirements. The most important items are shown at the top of the product backlog so the team knows what to deliver first. The development team may pull work from the product backlog as there is capacity for it by iteration (scrum). A well-prioritized agile backlog not only simplifies the release and iteration planning, it also broadcasts all the things that the project team intends to spend time on.

It is imperative to note that during the requirement gathering process, the Product Owner will communicate with the clients as well as other stakeholders involved in the project to collect their requirements which will then be used to write user stories. The user stories are created and phrased in terms of the needs of different user groups for identifying the concrete client requirements. Thus, the user stories are generally the preferred items to track development tasks within a backlog. The Product Owner will then specify the priority of each of the user stories in order to indicate which are most important for the system. The user stories will then be divided into tasks and have their difficulty estimated by the developers.

These user stories will then be stored in a product backlog on Trello, which is also an online backlog management software. However, it is more often than not when the todo list of user stories might get too complicated. The team members may then have a discussion together to break user stories down into individual development tasks. Therefore, it is said that the product backlog will change over the course of the project. Some product backlog items will be added, some might be removed, and many will be expanded -- possibly broken down into smaller items with a more well-defined scope during the team discussion.

After further categorizing of the tasks, a Trello board may then be created to add all the smaller, broken down tasks, whereby each task may be contained in a card. Then, a list of such cards (tasks) sorted with priority known as backlog is then built. For each product backlog item, it also needs to list out its description, estimate of the effort to complete it and its priority as agreed by the Product Owner and the rest of the team. Other information is often stored in the backlog as well. For example, user stories may contain a link to UI mockups, or perhaps to detailed algorithms if these algorithms form part of the specification. Not only that, information such as bugs, design changes, technical debt, customer requests, action items from the retrospective may also be included in the backlog. This ensures everyone's work items are included in the overall discussion for each iteration.

During the sprint planning meetings, the project team will then determine the sets of tasks from the product backlog to get implemented in the current sprint. Then, the selected sets of tasks will get arranged into the sprint backlog. The team will then allocate the task to the most appropriate member for designing and coding each item according to priority, keeping track of progress on active tasks using a kanban.

In conclusion, backlog serves the purpose for allowing prioritisation of user stories and associated developer tasks, and assignment of tasks to individual developers. The backlog also functions as the connection between the product owner and the development team. The product owner is free to re-prioritize work in the backlog at any time due to customer feedback, refining estimates, and new requirements. But once the team has committed to a sprint backlog, the Product Owner should not add anything to the sprint once it has started. Any new or changed requirements should go in the product backlog.

6.3.1 Backlog Refinement

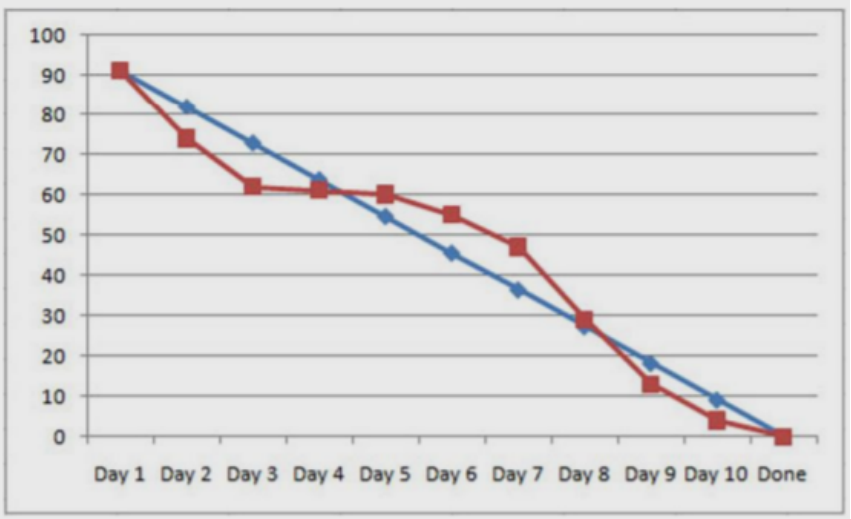
Once the product backlog is built, it is also crucial to regularly maintain it to keep pace with the program. Product owners should review the backlog before each iteration planning meeting to ensure prioritization is correct and feedback from the last iteration has been incorporated. Regular review of the backlog is often called backlog refinement in scrum. If the product backlog items are not well-described, they need to be fleshed out; if they are too big, they should be broken into smaller items, and so on. The team may offer help to the Product Owner to give a rough estimate on those product backlog items.

7.0 Time Management

Time management is known as the process of organizing and planning how to divide time between specific activities. Since the project team is made up of members from different backgrounds and each of them has his own commitment, therefore, good time management to keep track of time spent by each member on project tasks is critical to ensure the smooth workflow. Good time management enables the team to work smarter – not harder – so that the team can get more done in less time, even when time is tight and pressures are high, achieving high work efficiency.

7.1 Keeping Track of Time Spent on Project Tasks

In each sprint, the team will keep track of the time spent on each task using a sprint burndown chart. A sprint burndown chart is a graphical representation of work left to do versus time. At the end of each day, the team will record the amount of tasks completed and not completed where a team member then updates the sprint burndown chart. An example of a burndown chart is shown below.



Reference: Screenshot from Week 5 Lecture Slide Page 25

The project team will be using a slightly modified chart similar to the one shown above. The days of the sprint will be signified by the x-axis where it spans from Day 1 to the last day of the sprint. The number of story points yet to be implemented will be shown on the y-axis.

Not only that, to make sure accurate records are taken, a custom timesheet will be created for team members to enter their tasks and total time taken spent on completing a certain task every day. By doing this, it is easy for every team member to record their completion time and the person in charge of updating the burndown chart will be able to do his/her job efficiently.

There will be no standard working hours and neither there is maximum or minimum commitment as all team members are free to work at any time that they are free. This is due to the reason that most team members have different timetables to follow as not every member takes the same units. To set standard working hours, maximum or minimum commitment would be impossible and unjust to team members struggling to obtain a similar amount of free time for the project. However, the strict requirement for each team member is that they complete their task before the sprint ends or before the deadline given for a specific task.

Keeping track of the team's effort and the time needed for the team to complete each task or feature will allow the team to know who might be in need of assistance, to know who is not doing their tasks in the amount of time agreed, and to determine the velocity of the team's progress. From obtaining this information, the Scrum Master and/or team members will be able to devise a solution after identifying the problem that has been hindering the team from the completion of the project.