

FIT 2101 (Project Management Plan):
Zrello Scrum Project Management App

Presented by:

Garret Yong Shern Min - 31862616

Avinash Rvan - 32717792

Aaron Tong Jian Chern - 32946511

Ong Jing Wei - 32909764

Sayyidina Shaquille Malcolm- 32578393

Index

1. Team organisation	3
2. Time and Task Tracking	7
3. Definition of Done	8
4. Vision Statement	9

1.1 Team Organisation

List of Members:

1. Aaron Tong Jian Chern - 32946511
2. Avinash Rvan Ravindran - 32717792
3. Ong Jing Wei - 32909764
4. Garret Yong Shern Min - 31862616
5. Sayyidina Shaquille Malcolm- 32578393

List of Communication and Contact Information

Our team has agreed to use WhatsApp as the primary vector of communication. The following is each of our team member's contact details: -

1. Aaron Tong Jian Chern
H/P :- +6019 336 1928
2. Avinash Rvan Ravindran
H/P :- +6012 236 9284
3. Ong Jing Wei
H/P :- +6012 701 8833
4. Garret Yong Shern Min
H/P :- +6017 307 8780
5. Sayyidina Shaquille Malcolm
H/P :- +6281285966961

Roles and Responsibilities

Aaron Tong Jian Chern

Role: Team member

Responsibilities:

Carry out the tasks

Avinash Rvan Ravindran

Role: Product owner

Responsibilities:

Build and manage the product backlog

Ensure that each team member is clear on the requirements for the product

Give the team guidance on which tasks should be implemented in which order

Ong Jing Wei

Role: Scrum master

Responsibilities:

Host meetings and record aspects that need improvement for upcoming sprints

Ensure the project is on track and check on member's progression

Garret Yong Shern Min

Role: Team member

Responsibilities:

Carry out the tasks given by other members and take initiatives to jump on tasks yet to be undertaken

Malcolm Shaquille Sayyidina

Role: QA person

Responsibilities:

Will be in charge of QA and making sure the project is delivered perfectly

Difference between our process versus Scrum

There are several subtle differences between Scrum and the processes that the team is going to perform throughout the project. For this project, the team is using a variation of Scrum that best suits this project. A Scrum project in general works by adding tasks to a product backlog, and then planning which of those tasks will be done in the upcoming sprint and will be added to a sprint backlog. Then, the tasks are worked on and the sprint will be reviewed once the set time has passed. Then the sprint cycle is repeated until the product is completed. To maintain the Agile methodology, the product backlog is constantly updated with new tasks based on any updated requirements of the client. Scrum projects also have daily scrums which are short daily meetings with all members of the team to discuss and confirm what work would be completed for that day.

For our project, we would be implementing Scrum with some slight variances. Firstly, we would not hold daily scrums as we feel this is unnecessary and would take up too much time since each team member has other projects to attend to as well. Instead, the team members will send a quick and short update message in the team WhatsApp group chat to ensure that everyone is up to date on the progress of the work.

Apart from that, in a Scrum project, no tasks should be added to a sprint once the sprint has commenced. However, for practicality, our team has come to the consensus that certain tasks will likely need to be added and completed in parallel with other tasks in an ongoing sprint. This will allow more flexibility when it comes to planning sprints.

Furthermore, our process will combine the sprint planning and review steps in one. In a Scrum project, sprint planning is the step taken before a sprint begins to determine what tasks need to be done in the sprint. The sprint review on the other hand is a review conducted at the end of the sprint to discuss how effective the sprint was and what could have been improved. Our team intends to combine these two steps into one step. This makes the process very efficient since the team would need to review the previous sprint's performance when planning the upcoming sprint anyway.

Moreover, although Scrum requires teams to have specific roles such as scrum master, product owner and so on, in our project, all team members will be involved in all aspects of the project. The only difference is that the team members assigned to a certain role are in charge of ensuring that role gets carried out. For example, the QA person will not carry out all the quality checking themselves. They may get the assistance of other team members as long as they ensure that the QA is done properly and thoroughly.

Apart from these differences, our project process will be similar to Scrum. This means that all the tasks needed to be carried out will be added to a product backlog and before a sprint, the team will meet and decide on what the tasks to be completed are. Then, the team will divide the work by first placing a difficulty rating for each task. Then the tasks are divided by trying to give each member the same total difficulty rating. For example, a member with a 3-star and 2-star task is considered to be doing equal work to a member doing one 5-star task. Of course, this is a relative scaling system and will be amended as needed. Then, the sprint will commence and the team members will carry out their tasks. Throughout the sprint, the team members will regularly update the team on their current progress to ensure that the team is up to date. Once a sprint is completed, the team will then hold a meeting to discuss how the sprint went and subsequently decide what tasks need to be accomplished in the upcoming sprint. This sprint cycle is repeated until the end of the project.

Differences	Justifications
No daily scrums	This is too time-consuming and sending a text update is a much better alternative
Tasks can be added once the sprint has started	Allows for unpredictable tasks which may sometimes crop up

Combining the sprint planning and sprint review	Makes the process more efficient as they go hand in hand and it is more time consuming to separate them
Each task won't be necessarily done by only the appointed members	The project has little manpower and this can help to achieve the greatest potential. This can also help balance the workload between members.

1.2 Time and Task Tracking

Usage of Git:

Git will be an integral part of our application development and communication (code-wise). All codes written will be shared on Git and will be worked on simultaneously by all members.

Each member is expected to use proper naming conventions accompanied by clear but documentation when writing their code to ensure a smoother and swifter peer review process. In addition, appropriate commit messages will be uploaded to the cloud upon satisfactory completion of any tasks.

Each section of code (block of code), will be written in a separate branch from the master to ensure that there will be minimal conflict. Upon the completion of each branch and/ or function, approval will be requested from the QA personnel and 1 other team member to cross-check the correctness of the codes to ensure seamless execution and minimal conflict.

Task Allocation:

All tasks will be analysed and given an approximation of difficulty based on the complexity of the tasks and the expected task duration. This difficulty score will serve as the base for the task distribution that will occur before a sprint and will help to ensure fairness amongst the tasks allocated.

Task allocation will be done through Trello, all of the allocated tasks for each member will be visible there. Google Drive will be used for our storage relating to documents such as reports.

Time Tracking:

Each member of the group will be given tasks to complete with a set of deadlines.

This allows each member to be clear on what and when to deliver. Time estimation may not always be fully accurate, this is why everyone in the group will adapt and start to give close to perfect time estimations of their work overtime.

The equal working time between members will also be done, this will go hand to hand with our task allocation process to ensure the ultimate goal is achieved fairly between members. When a member can't achieve their deadline with their full potential, another member will also help the member in need thus hopefully finishing their task with minimal delay. The member that was helped before should also be helping their helper in their task which creates total fairness between the members. To help with keeping track of the time spent by each member on this project. The team will be making use of the Planyway plugin which is available within Trello itself. This allows us to monitor how much time each member spends doing their tasks. Thus, this will give a clearer understanding on how tasks should be allocated in future sprints.

1.3 Definition of Done

The Definition of Done is a collection of criteria that must be fulfilled before a project can be deemed complete. A checklist will be made to ensure that all the conditions or acceptance criteria that a website application must satisfy are met so that it can be accepted by the client.

Final Checklist

Code is peer-reviewed
Code passes all the unit tests that have been created
Refactoring done
Code is well documented
Any build modification and configuration are documented
Documentation is updated
All the features that are requested by the client are in the application
Project is error-free
Bugs are fixed
Quality assurance performed and all issues are resolved
Launching the web application on a test platform that is identical to the production platform
No work is left unchecked at any stage

User Experience(UX) is tested
Approved by Product Owner
Approved by Stakeholders
Approved by team members

1.4 Vision Statement

For project managers who need to keep track of their scrum project and task, Zrello is a project management app that categorise the product backlog which will sort according to its prioritisation and organise the sprints to make the project more manageable unlike the normal collaborative white board system, our product provides well-organised templates which clearly outline the progression of the task, allows collaboration for better communication and includes in depth review of time spent on the project.

2 Risk Register

Before starting any project, it is very important to first identify all the risks associated with the project. This is called a risk register. By creating a risk register, it can help a team identify possible places where a problem can occur in the project. The risk register will also include some descriptions of the estimated likelihood and impact of the risk, a monitoring strategy, and a mitigation plan in case the risk does occur. This process can greatly help a team get ahead of any problems by having a plan to monitor and overcome them before it even occurs. Of course, there is a very high likelihood that there will be problems that occur which cannot be predicted. However, it will still be very beneficial to try and identify any predictable or possible risks before starting the project.

For this project, there are 4 risks associated with it. Each risk will be listed along with its estimated impact and likelihood, monitoring strategy and mitigation plan. The estimated impact and likelihood of the risk will be listed by making use of the risk matrix as shown below:

		Impact →				
		Negligible	Minor	Moderate	Significant	Severe
Likelihood ↑	Very Likely	Low Med	Medium	Med Hi	High	High
	Likely	Low	Low Med	Medium	Med Hi	High
	Possible	Low	Low Med	Medium	Med Hi	Med Hi
	Unlikely	Low	Low Med	Low Med	Medium	Med Hi
	Very Unlikely	Low	Low	Low Med	Medium	Medium

The risk matrix shows the severity of each risk based on its impact and likelihood.

Risk 1: Team members get sick

The first risk associated with this project is that there is always a chance that team members may get sick and this will slow down the progress of the project. The likelihood of this happening is likely, and the impact is significant. This means that the severity of this risk is medium-high according to the risk matrix. It is likely to happen because, over the course of 12 weeks (which is the duration of this project), it would not be unusual for at least one team member to get sick during this time, especially with Covid-19 still being an issue. If a team member does get sick, this would cause a significant impact to the project since they would be unable to do their part of the project. Thus, the project's progress would slow down significantly as the work being handled by the sick team member will be delayed.

Monitoring strategy:

This risk can be monitored with weekly updates from individual team members on their current health status. This can be done via the team WhatsApp group chat. By constantly updating the team, it will not come as a surprise if one of the members gets sick and the team will have time to deal with this issue.

Mitigation plan:

In the likely event that a team member does get sick during the project, there needs to be a mitigation plan to ensure that it does not affect the project too heavily. First, the sick team member will determine whether he/she is still able to carry out their tasks at home (assuming the sickness is mild). If that is the case, then the project will carry on as usual. Assuming the team member is unable to continue their work, the team will then hold a short group meeting online to discuss how the workload of the sick team member can be shared. Thus, even though a team member is sick, their work will still get completed and the project will not be delayed.

Risk 2: Code does not function as intended

The next risk is that the codes written for this project will not function as intended or planned. This is very likely to happen, and the impact would be severe. This means that the severity of this risk is high according to the risk matrix. It is very likely to happen because bugs and errors in code are very common as codes are written by humans and this will tend to lead to human errors. Even minor errors such as a misspelt variable name could lead to a system failure thus it is almost certain that the code will not function as intended on the first try. The impact is severe because, with malfunctioning code, it would mean that the product does not function as the client intended which would result in a faulty product.

Monitoring strategy:

Since this risk is unavoidable, the team will carry out code testing consistently throughout the whole project. Instead of only testing the whole system at the end, it would be far better to test individual sections of the code throughout the project. This allows the team to detect any issues with the code in smaller sections which would make it easier to find the root of the problem. This testing can be done by testing the input and output of functions that are created or by using print statements to ensure that the code is doing as intended. The testing should also be carried out by a team member who didn't code that section. This is to ensure that there are no biases when testing and that every part of the code gets tested.

Mitigation plan:

Once the testing is conducted, the problems can be identified. However, the root cause of the code failure would still need to be identified. This can be done using debugging. This is also a built-in feature in most coding IDEs where you can step through each line of code one by one to identify the problematic line of code. This allows the team to quickly identify and fix the issue. After the testing is done, the team member in charge of that code will take the list of problems identified and carry out debugging to ensure that the problems are fixed.

Risk 3: Client's requirements are misunderstood

One of the other risks that can cause a problem to the project's success is that the team may misunderstand the client's requirements. The likelihood of this happening is possible, and the impact is severe which makes this risk's severity medium-high. It is possible because when the client explains their requirements, it is possible for the team to understand it in a slightly different way compared to how the client explains it. In fact, it is even possible for the client to be describing something different to what they want because even the client might be unsure of what they want. However, this risk will have a severe impact on the project since the entire project's success depends on how much the final product matches the client's requirements. Thus, if the requirements are misunderstood, then the project would be deemed a failure.

Monitoring strategy:

To monitor this and to make sure that the team is not working on something which is not according to client requirements, the team will make sure to ask follow-up questions to the client to ensure that the understanding of the client's requirement is the same for both parties. Apart from that, the team will also come up with mock-ups of what the final product would look like to ensure that the client is happy with it. If the client agrees to the mock-up of the product, then it would be justified that the team has not misunderstood the requirements and that it matches what the client requires.

Mitigation strategy:

In the event that the team realises halfway through the project that the product being developed is not following the client's requirements, then the team will conduct a meeting with the client to discuss and clarify the actual requirements for the product. Once the true requirements are clarified, the team will then hold a meeting on their own to discuss which parts of the currently developed product can be reused for the actual requirements. This would allow some time to be saved when redeveloping the product to match the client's requirements. However, it should be noted that this is only in the worst-case scenario that this occurs, and the monitoring strategy should be carried out regularly to ensure that this does not happen in the first place.

Risk 4: Disagreements among team members

Another risk that can affect this project is disagreements among team members about any aspect of the project (design, task division, etc.). This risk is likely to happen and the impact of it is moderate. According to the risk matrix, this makes the severity of this risk medium. It is likely to happen since, in a group project, we would be working with all kinds of people with various personalities. Thus, it is likely for these varying personalities to clash, especially when making decisions as a group. However, the impact is moderate since disagreements are expected and team members will still be respectful even if they have disagreements. This will ensure that the project still gets completed on time and with the high work quality.

Monitoring strategy:

Although it is not possible to “monitor” disagreements before they occur, the team will however always have group discussions before taking any major decisions. During these discussions, each team member will be allowed to voice their own opinion and the group can come to a consensus on the best possible choice to make. This will ensure the least amount of friction among team members since everyone will get to have their say in the decisions.

Mitigation strategy:

If disagreements do occur, they will be solved maturely and respectfully. The team members who are in disagreement will both argue their points to the group as a whole and why they think their decision is the one the team should take. This gives the team a chance to hear both sides of the story before making a choice. If the disagreement escalates, then the Scrum Master will ensure that the team still functions as a team and comes to a rational choice based on the reasons supplied. If a centreline cannot be reached, the team will decide to use a simple majority voting system to make things simple. Although this is not the best way to solve the issue, it will ensure that a small disagreement does not bring the project to a halt.

Risk 5: Insufficient Team Expertise

Another risk that can affect the progress of the project is the lack of team's knowledge on a certain topic. This risk is very likely to occur and had a significant impact on the project. The chances that the team would face a new task that they have yet to learn on is to be expected especially considering the team comprises university students. However, if no steps are taken to overcome this risk, the worst case scenario would lead to the complete stagnation of the project's progress. Hence, a proper plan has to be devised to ensure that this issue would be resolved and the project progresses as planned.

Monitoring strategy:

It would not really be possible to avoid these types of situations as they are almost certain to occur. However, this does not mean that there are absolutely no step that can be taken to monitor it. One possible solution would be to identify the topics that would give the team problems in future and read up on those topics well before the implementation. By doing so, this would cut down on the time taken for research that would be needed during the sprint itself.

Mitigation strategy:

The mitigation strategy reflects similarly to the monitoring strategy but should this event fall upon the team during the sprint the team would have to prepare to conduct a bunch of research on those topics. Other, more knowledgeable, members of the team should step and guide those members having problems understanding the material where possible. The research should be done early into the sprint so that there is ample time for the implementation and testing of features to be developed.

Risk 6: Family Emergencies

Another risk that can affect the project is when a team member is suddenly faced with a family emergency (death of a family member, divorce, home robbery, etc.). This risk is very unlikely to occur but has a impact rating of significant to severe. According to the risk matrix, this would make the severity of this risk, medium. It is unlikely that these sort of situations would occur all the time (if not at all during the period of the semester) as these situations are rarely occurring events that are unforeseeable. However, the randomness of the possible timing and the severity of the emergency would certainly leave a significant impact on the progression of the team. For every type of emergency, there are certain conditions and steps that are required to be carried out in order to manage this risk, the important thing is to adjust to the situation accordingly and make compromises to ensure the efficacy of the sprint.

Monitoring strategy:

These situations are usually quite random and hard to foresee so there is no proper monitoring strategy in place for risks such as these. The only possible course of action would be a swift implementation of the mitigation strategy.

Mitigation strategy:

If these sort of scenarios do occur, the affected member is expected to immediately report their situation to the group. Whether or not they can proceed with their tasks is completely up to that member and their capability to cope with the situation they are facing.

Assuming the worst-case-scenario, whereby the affected member is unable to do their tasks for an extended period of time (and would affect the project's ability to deliver a satisfiable product by deadline), the tasks of that affected member would have to be divided amongst the rest of the team's members. Where possible, the team should take on the minimal amount of tasks needed to cover for the affected member's lost time. However should the need arise, full coverage of their tasks might have to occur. So, the divided tasks are to be divided up fairly amongst the rest of the team; the priority of these tasks would have to be weighed on the impact of those tasks on other parts of the system but, when possible, should be left to after the members have completed their own tasks. The affected member should also always be contactable and should try their best to return to their duties to the team as soon as possible. Where applicable, other compromises should be made. For instance, family issue which takes the affected member overseas, bring a device along with them so they can continue working despite being away from home. Or another example, an incident where the affected member would need some time to recover, negotiate a time where the member can recover but still has enough time to complete their tasks.

By carrying out this mitigation plan, the project can still go on, albeit at a slower pace, but the product should (hopefully) be able to be delivered as promised, on time.