FIT2101 – Software Engineering Processes and Management: S2, 2021
Assignment 5 Individual Report

Name: Chan Wai Han
Student ID: 31555373

Topic 1: Agile software development practices

The way in which our team developed the program was quite different to the way in which Agile development is carried out in "by-the-book" industry projects. Our team's process did its best to follow Scrum practices, defined as the iterative, incremental framework for projects and product or application development. Before project inception, our team practiced delegating roles, such as having a product owner, scrum master, front-end team, as well as a back-end team. The scrum master acted as the leader of the team, setting up meeting times and asking the client questions regarding the product that we had to develop. However, the product owner was more of just a formality. In reality, throughout the assignment, all of us had our fair share in deciding the content of the product backlog, as well as deciding the priority of each backlog entry. Besides that, our product did not have a business value, thus we did not have to predicate our product backlog decisions based on a business value. Furthermore, each team member was responsible for maintaining the product backlog, so we technically did not have a legitimate product owner that only had to care about product backlog related issues. In other words, everyone had to contribute to the development of the software. In other words, this meant that all our team members had flexibility in responsibility, where each of us were aware of our responsibility to check the product backlog and to ensure that somebody is doing it. Consequently, quality assurance was present during our team's development process.

In Sprint 1, our team did not properly practice sprint meetings, as well as the implementation of Trello. The development process was messy, with information being stored all over the place in a single Google Drive which consist of multiple documents. Besides that, we did not properly plan out any meetings as required by Agile practices. Despite satisfying the Agile practice of not having a "big boss" in making decisions, and instead having team members come to a collective consensus, proper documentation was not done and team members tend to forget about the decisions that were made final. Not only that, we did not properly document the client's requirements of the product as user stories. This made us assume many requirements and the assumptions were indeed wrong, in which the client consequently told us to change.

Therefore in Sprint 2, our team had a hard reset. We properly documented information into Trello. Our team practiced the single-platform-use of Trello, and stored anything related to the project into Trello, except for long documents or tables that had to be typed out, of course. In Sprint 2, we re-organized our documents and created a singular document called Project Inception. The Project Inception document consist of our team's vision, communication method, progress tracking, process model, definition of done, team management, risk management, and stakeholder management. Next, our team created proper documentation for different information within Trello, such as the product backlog, sprint goals (MVP), sprint 1 backlog, sprint 2 backlog, sprint 3 backlog, as well as a section for only user stories. This made the software development process extremely efficient, by implementing user requirements based on user stories. Besides, we also had a time tracker for each team member.

Consequently, Sprint 3 was done in a quick and efficient manner, with all information being in proper places. These practices allowed our team to realize a few values and principles as laid out in the Agile Manifesto. For example, through frequent Sprint meetings and stand up meetings, we were able to value individuals and interactions over processes and tools, due to the frequent communication among team members, which also improved the team's chemistry along the way. Besides that, our team was able to smoothly respond to change over following a plan, with user stories being the sole focus of team members when developing a software. When a requirement is needed by the client, or changed by the client, it was easy to change up what had already been done, due to the software being developed based on user stories. Sections that needed changing were easy to identify and change based on the client's requirement.

In order to prevent the same mistakes from being made in new Sprints, our team had retrospective meetings to identify problems from the previous sprint. We would list them down and record them in Trello. This was to keep track of the mistakes that were made in the previous Sprint and so that we could prevent those mistakes from reoccurring, as well as to improve from our previous Sprint in the future Sprint. The retrospective meetings that we held as well as the things we recorded in Trello helped us significantly in improving our team's performance, especially from Sprint 1 to Sprint 2. In Sprint 2, our efficiency was much better compared to Sprint 1. By Sprint 3, we had gotten the hang of the good practices and as a result, Sprint 3 was a smooth and fast process for our team. Without the retrospective meetings, we would not have identified our teams problems and the same mistakes would most likely reoccur, consequently causing our team to be unproductive and inefficient.

Due to the coronavirus outbreak, all of us had to stay at home during the semester. I believe that this had a big impact on our software development practices, mainly due to the fact that we were always at our computer, so it was easier to plan for Sprint meetings and stand up meetings. If it were to be a real-life setting, I believe that the way we held our meetings would not be good, as we planned them about 1 or 2 days prior.

Thinking back over the project, I believe that the one thing we should have done differently would be to implement Scrum practices from the start. Due to ignoring Scrum practices in Sprint 1, a lot of our work was ad-hoc and based on our own assumptions. However, after implementing it in Sprint 2 and 3, our work became very smooth and I am glad that we were late than never.

Topic 2: Working in teams

Coordination with the team members was good. Despite having team members from different countries, most of the team members already had an unhealthy sleep schedule, so matching the time that we were awake was not an issue (despite it being unhealthy). Communication between team members were done through WhatsApp, Discord, and Zoom during tutorials. Task allocation was done with mutual consent, in which every team member chose the task that they wanted to do, based on the strengths and weaknesses. Thus, the arrangement was satisfactory to everyone.

I found the team selection based on blind CV cards a bit useless, due to the fact that everyone's CV card looked identical, as the students were mostly from the same cohort and thus took the same units and did the same projects. I believe that regardless of who the team members were, we would have been able to still achieve the stated aims to help create inclusive and diverse cross-functional teams, as all of us had a similar goal in mind – to ace the unit with high marks. Consequently, we would still be doing our best as a team to ensure we get the highest marks possible.

I can measure our team's performance based on the Tuckman and Jensen group development model which was proposed in 1965, in which consists of forming, storming, norming and performing. During the forming stage, our team members did not know each other very well. Thus, we had to put in more effort to understand each other, as well as to understand how each of our working style is like, in order to sort out the task allocations. Besides that, there were times when team members did not understand the task at hand, and therefore we had to patiently explain the requirements to them. There was not much collaboration as we have yet to learn how to work together.

During the storming stage, we did come into a bit of conflict. It was a small conflict however, in terms of small decisions such as what programming language should be, or small little details in the code. This was just a small matter of differences in work preferences or opinions. Regardless, the group dynamic stayed strong and never looked like it was falling apart. The team's productivity was not really affected, as we managed to resolve the conflicts pretty quickly, understanding that these were just small issues that were bound to occur.

During the norming stage, our team managed to reach a healthy level of trust and respect between teammates. This occurred around our Sprint 2 and 3, where we started to be efficient in our development process and managed to complete many user stories in a short amount of time. This is where most of the team members began to take on unofficial roles within the team , such as the "mood calmer", or the "document creator". Team members would voluntarily take on tasks for the benefit of the team. On a side note, during this period, we began to play some video games together after completing a certain task. I believe that this helped improve our understanding of one another and made us better with communication. It helped the team members bond and we could consequently perform better.

Subsequently, our team entered the performing stage, where everyone understood how the team functions. There was a high level of trust between team members, and it the journey in working towards completing the client's product was smooth and easy.

The adjourning stage is occurring right now, where the client's product has been completed and the team members rarely talk to each other now. Regardless, we still try our best to keep in touch, and sometimes even play some video games together.

I believe that our team's teamwork practices would hold up very well if we were programming in a small company or in a larger organization. The only thing that I would change would probably be certain roles such as the product owner role. This is because as mentioned earlier, all of use were technically product owners. Thus in a real-life setting, we would not really know how a product owner is supposed to do his task. Other than that, I believe that our team's practices were efficient in getting work done.

Topic 3: Managing change

Our team had a big problem during the transition from Sprint 1 to Sprint 2. Basically, in Sprint 1, our team used Wix as our platform for the client's website. Despite having easy templates to design our website, the client's requirements complicated matters, in which we had to collect certain user data that Wix did not allow us to. Hence, we had to leave behind all our progress in Wix and start over by coding the website from scratch using Javascript, HTML and CSS.

Despite this huge platform change, I believe that our team handled the change well because none of us complained, gave up, or slacked off. All of us understood how important the change was and why the change had to occur. Nobody blamed anyone and we just continued to do our allocated work. We knew that we had to complete this in order to get marks for the assignment. In a real-world setting, problems like this will always occur. We understood that the most important thing in a situation like this, was how we solve this problem together as a team and make it work.

As the semester progressed, client's requirements and expectations were not a huge problem as they were properly stored as user stories into Trello. This made it easy for the team to implement these requirements into the software.

This huge platform change for the project arose from outside the team, not within. So we understood that there as nothing that we could do about it and there was nobody to blame for it. We understood that it was technically the client's requirement that forced this change to occur and it was just an unlucky circumstance due to the fact that Wix could not perform what the client wanted.

I believe that these sorts of challenges will always arise in industry projects. The way to manage this is to work together as a team. After all, it is not an individual project. I believe that team members should be there for each other, and create a positive atmosphere, which is motivating to each other. When we have our backs against the wall, team members should always be there to support each other and help each other solve whatever problems that are being faced. After all, we are all in the same boat.

Topic 4: Making decisions

Our team initially chose Python as the language to develop the website. This decision was made due to the fact that all members were most familiar with Python. However, it was hard to integrate the front-end with a Python back-end code. Then during a tutorial, the lecturer allowed us to use Wix. Since we discovered Wix and it had a lot of ready-made templates, we decided to use it. Despite being easy to design due to the free templates, we could not obtain user data from it, which was a client requirement. Thus, we decided to just go ahead and code the website ourselves. Although we were not good Javascript programmers, we managed to find some templates and code online and thus we were able to integrate them into our software to meet the client's requirements. These changes may seem messy and may look like extra work, but it was a result of the team members's consensus in that moment in time, with perfectly valid reasons. Thus, we did not think it was annoying to change so many decisions. We understood the root of the problem and we just took the challenge head on and tried solve it in the best way that we can.

We are extremely confident that the requirements we elicited from the client are correct due to the fact that we recorded the user requirements as user stories. Each user story is specific enough that it does not contradict with any other user story. We also programmed the software based on the user stories, so there will not be problems in the implementation. I believe that the final product serves the client's needs well because as mentioned earlier, we satisfied all user stories, which meant we satisfied all user requirements.

One thing that I may want to do differently for future projects, be it in industry or in class, would be to perhaps make user stories more specific from the start. This is because in Sprint 1, our user stories were pretty generic, and hence we had to make a lot of assumptions. These issues were only fixed from Sprint 2 onwards, which smoothened the entire software development process.
Therefore, it would be nice to have specific user stories from the start, to enable us to have a smooth and efficient software development process. On a side note, even if user stories were too generic, I believe that it would be better to discuss with the team when making certain assumptions. At least those assumptions are agreed by the entire team, rather than having the assumptions be made by the team member allocated with the task.

Overall, I believe that this entire process was aa great experience, in terms of giving us a little taste of the outside world. I am grateful for this experience and I cannot wait to implement what we learnt in this unit, to collaborate with others as a team in the real-world working industry.