Reflection Report

Topic 1. Agile software development practices

As the product owner, during the sprint planning, me along with my team decided on high priority and set the product backlog. This was done before the start of any of the sprints after client requirements were elicited. After deciding on the backlog we moved to phase 2 of the sprint planning where we decided on how we were gonna implement the requirements such that we can complete it by the end of the sprint.

We followed the need for the daily scrum in out own way by having a weekly standup and sometimes even weekly twice. This was because we had other units that we also had to work on and also often a daily standup would be pointless since there would not be much of a progress report and thus we agreed on a weekly standup for a more impactful review. During these standups the Scrum Master used the product backlog to show the team's progress and reflects on how the progress has been and reviews the sprint accordingly.

During the sprint we would sometimes have to refine the backlog to help with the future sprints. At the end of all the sprints, we hold a sprint review where we will demo the product to the client and get feedback from him of the sprint and consequently also get new requirements for the next upcoming sprint

These scrum practices worked really well with us. We started the first sprint completely disorganised but once we learnt and started following the scrum practices the development tracking and organising and the process itself was better after its implementation.

Initially we only ever made contact with the client when it was required over the period. It was only during the regular scheduled meetings that we gave the client the updates. Over time we learnt that we could host a client negotiation and we also learnt to request for the client meeting outside of the scheduled times. This helped us a lot during the sprints to ensure that the final product aligns with the client's requirements.

We also made sure during elicitation that we got into intricate details about every single requirement that the client needed from us in the final project. It allowed us to provide the full satisfactory product after the end of all the 3 sprints to the client.

The biggest impact due to the academic environment was the commitment to the other units of the semester. But due to the fact that the other team members had similar units, the workload was similar and thus we were able to manage the time spent on the project. The workload of the other units also had an impact on the quality of the first sprint since I had not yet learnt to balance the workload. I did not have any other problems due to the academic environment

I think the project realised many of the principles laid out by the manifesto. We included customer collaboration over fixing a contract prior to the sprint. Even though we did have comprehensive documentation for the work we did, we made sure to prioritise working software before the documentation and were very versatile for any unforeseen circumstances. I feel like out project did realise all the values laid out by the Agile Manifesto

Looking back at the project, the only thing that I can think of that we could have done differently is the way we handled the first sprint. During this sprint we did not follow any of the scrum practices leaving the team disoriented and disorganised and not aware of what was to be done. What I would have done differently is ensuring that the team was aware of all the scrum practices and the weekly responsibilities of updating the team. We should have also done our first sprint planning which would have reduced the need to halt in the middle of the sprint to re-orient ourselves

Topic 2. Working in teams

I coordinated with my team members by communicating with them through every step of the project. As the product owner, I ensured that each small decision needed to be taken had to go through me as a form of quality control. This was so that I can make sure that the team follows the contract that we had planned to follow before the start of the sprints and the development process.

As a developer, I made sure to communicate what I needed from the other team members so that I could finish my side of the process as well. As I was developing the project I made sure to inform the other team members of my progress since we did not have daily sprint standup meetings to update each other on the progress daily.

I communicated with the team through our agreed mode of communication i.e. whatsapp. I ensured that I did not leave any one in the dark while I was working on features so that the dependent features do not get held back due to me. This was the way all the team members communicated with each other

Initially we allocated tasks according to the capabilities of each of the team members without considering the different developing roles at first. This was because we were already behind and needed to get a start on the sprint. As a product owner I did not feel like this was a satisfactory way to split up and allocate tasks since it was very messy. After the first week of the sprint is when we decided to split the team members into the roles of backend and frontend developers and also set aside members to integrate the two. This was a far more efficient way since it involved two members under each role to work together.

In this way we were able to complete the following 2 sprints more quickly and efficiently than the first sprint after we reached the client with more quality and satisfaction.

Within a small company a lot of these practices would change. The way the tasks are allocated would change drastically since the development process only involves a small group of people. There would not be a need to split the team members into specific roles since if any block was to happen to any of the members the other members must have a capability to assist and pick up his work as well. The integration would be done by the developers themselves and not need other people to do so.

The daily standups would not require physical attendance of the members and could be finished in a quick call. The meetings would also not last that long as compared to the average meetings and would only take around 5 mins each due to the small size of the group. The code quality will also increase since there will be lesser need for code reviews and there would be no need for a product owner to take care of decision making and code reviewing

In the beginning of the team formation everyone followed the first stage of the Bruce Tuckman's Team Formation model by being kind and polite to each other since we did not know each other yet and wanted to make a good first impression. I could see that we were all optimistic about what we were gonna accomplish and were trying to get to know each other

Following that when we got into the details of the sprint we went into the next stage where the roles needed to be decided and decisions had to be made. We did not follow much in regard to the Storming phase since we only disagreed with the decisions made since we were not yet sure of each other's strengths and weaknesses and had to learn. We passed this phase smoothly without much team tension and decided on the roles and responsibilities. This led us to passing the Norming phase as well since we all agreed upon all our responsibilities by the first half of the sprint

We worked our way to the performing stage as we completed features and we moved through sprints. We did not move to the adjourning phase at all throughout the development process. This caused us to maintain the performance level and team chemistry leading to efficient development and healthy relationships till the end

The only conflicts that occurred in the duration of our projects was due to the confusion in roles between team members. This resulted in other team members doing the work they were not supposed to. Due to this we had to rollback a lot of changes causing major setbacks in the development. This occurred due to the lack of communication and lack of clear structure and definition in what is expected of each member.

We did not assign roles to any of the team members leading to confusion. This could have been avoided if we had defined clear roles to each team member before the start of the sprint so that we could have avoided these problems and reduced the initial setback.

Topic 3. Making decisions

Our initial choices for language, platforms and tools were good ones for the scope of the project, since the project required a static website with not much dynamic content as the MVP.

We decided to move with HTML and CSS for the front-end since it was a language that can be easily learnt within a short span of time. For the back-end we went ahead with javascript since it integrated with HTML well even though it was a bit on the harder side to learn but the pro of the integration far outweighed the learning spike.

For project tracking we used Jira since it had a more modern look to it and was easy and intuitive to use. Initially for the sprint standup we decided to just do a recap within ourselves without using any tools so the documentation would just be a video but later we realised that it was not a good way to do so and we changed the way we did the standups

For the documentation side we stuck with using google services like docs and used notion and these were good tools since everyone can collaborate on the same document at the same time without any issues. And for version control we used Gitlab services since this helped us keep track of changes made to the code at any point of time

We had to change the way we did our standups since in the beginning it was not as detailed as it should have been. We changed from just recording ourselves in the standup to using zoom as an interface where each member of the team would use jira and share their screens to show their progress and it helped the scrum master show the team’s progress using visual aids rather than just with words.

We also had to change the way we did the sprint review with the client. At first we decided to just let the product owner do the demo. This decision was changed since we wanted to include the participation of all the members in the team so we decided to split the features between the members and let them demo the features assigned to them.

As the product owner,I was fully aligned with the team's choices. Since I took an active role in the decision-making process, I was able to ensure every team member's perspective was effectively integrated into the final decisions. Our methodology involved a thorough reflection on the potential pros and cons of each suggestion, allowing us to accurately assess its impact on the team's overall progress and goals.

I am confident that the requirements that have been elicited from the client are correct. This is due to the fact that during the elicitation period we made sure to clarify each and every requirement the client provides. This ensured that we met the requirements accurately to what the client needed even when it was not clearly stated. We even showed mockups in the second elicitation round to clarify on the expectations of the client and following that, every client meeting we showed the progress of the software to see if we were meeting with his expectations and the client was satisfied with the current software.

This ensured that we were on track to the software requirements of the client and that towards the end of the final sprint the software is exactly what the client expected

I think that the final product serves the client's needs well since throughout the development process, we continuously took input from the client. Initially we did not know how to elicit the requirement properly and accurately and it led to the first sprint having a lot of negative feedback. But after we learnt to properly get the client's feedback, we were able to include him in the development process. We made sure to clarify each requirement that the client provided by going into intricate details of each feature.

In future projects, the only things I would do differently is the interaction with the client and how the work was split between the team members initially. I also would have preferred more standup meetings and clear communication between the members of the team through the agreed mode of communication

Topic 4. Ethical aspects

In an industry setting the developer may be pressure to cut corners on security features or data protection services due to budget costs or a deadline to be met. This may lead to the developer to use weak, poorly encrypted customer data when handling it. This violates the ACS code **1.2.1 The primary of the Public Interest.** A good mitigation strategy would be for the organisation to enforce strict security standards and code review process before the release of a product

A developer may also exaggerate his capabilities during the development process for a personal benefit when he might be able to fulfil the role. This is in direct breach of the ACS code **1.2.3 Honesty.** A good mitigation strategy is for the organisation to do a string background check and a qualification process to ensure that everyone put on the team is capable of accomplishing the requirements by the project

References

Tuckman's Stages of Group Development - WCU of PA

https://agilemanifesto.org/iso/en/manifesto.html

https://d25zr1xy094zys.cloudfront.net/c9/31/c931b791677ae939c33786b6935d5822dbfbd799?response-content-disposition=inline%3Bfilename%3D"Code-of-Professional-Conduct_v2.1.pdf"&response-content-type=application%2Fpdf&Expires=1762463040&Signature=4bAkNDfnu~ALQU1S6fG7zGsLplSGMG~5gOrZ2MAj~IRxjCVWfjn0JR1rUilVcHq3kdRSVStNUscdH97NESpWC3ekxZoz7xPGCSrMPiBxq4KD-IYDKkJBIvZ83WTHik64JjUS9998Ib37ws2Ad62nAxwbTZIxFcJd3~EI2LN9TbUaxVDoLeSqM2RD7yLq-HerBcx4X5pi8dCcbDxIv-IZxFxYv~UJPrElm-vHU94o81wefGOzNShmBgmmJWKEDnAKcWoIwPLxHvHwJebUGjKjtIk9nIcUKm2LsanwzPziU6KhAXLzw1U-WDeKdUCysjUhjrhb~NDvmgZsJDVB-n7pOw__&Key-Pair-Id=K1I589YUQOO6ZB