CS250 Project

As this sprint comes to a close, it's important to look back on the contributions of certain parts of the team and what the team as a whole accomplished in a Sprint Review. In this review, we'll analyze what went well and what could be improved upon for the next and future sprints.

The Product Owner played a well-rounded role in facilitating and managing while acting as the mediator between the customer and the development team. They also spent a good amount of time maintaining and updating the backlog and refining and clarifying user stories. Ultimately, the main goal of the Product Owner is to ensure that the project vision is reached. A good example of this occurred when the user stories had to be created. The Product Owner was responsible for generating the "what," "why," and priority based on relevancy to the overall vision of the project for each user story. They provided just enough detail to allow the team to implement, but not so much detail so that the developers had to ask for clarification. In this way, the Product Owner did their part in keeping communication alive.

The developers, who make up the bulk of the team, played a crucial role. They were responsible for building the software for the product by interpreting the user stories and writing testable code. A big part of their responsibility resided in making sure the code they wrote was clean and tested, while also asking questions from the tester or Product Owner to make sure they were on the right track. We saw this when they had to develop questions to ask the testers and Product Owner to satisfy the conditions of the project vision as it related to the user stories. The purpose of this was to make sure development followed the path of the project vision. Collaboration is key when it comes to being a developer; the more details and higher level of understanding they have, the more likely they will be able to create a product that will satisfy the customer.

The testers may seem like they play a small role, but in the agile methodology, they play an overarching role in making sure code development is progressing smoothly and not regressing as the project moves along. Their job is to make sure the code is up to standard with the project vision, and they do this in a multitude of ways.

The agile approach significantly streamlined the completion of the user stories compared to the waterfall methodology. The iterative nature of the sprints allowed us to update, add, or remove user stories as necessary as the project went on. Instead of testing at the end, as we would under the waterfall methodology, test cases were made and edited as needed. We also created an order of priority that allowed us to focus on what was important first and things that were more of a "want" than a "need" later. This, however, was made possible with the daily scrum meetings and constant backlog refinement, which allowed us to talk about these user stories and evaluate them as needed.

Agile is built on the idea of change. Being able to change something at a moment's notice is what makes it better than other methodologies like the waterfall methodology. We saw this happen when the Product Owner instructed the team to change a search filter functionality. The team was able to quickly incorporate the change, bringing the team closer to being able to satisfy the customer by making the product embody the product vision more. Had we practiced the waterfall methodology, we could have faced a significant setback, or it would have been too costly to incorporate the change at all. The added flexibility provided by agile gave us the edge needed to accomplish our goals.

In these sprints, we facilitated easy methods of communication through different facets, from scrum boards to low-fidelity user stories. If information wasn't clearly visible, it was encouraged to seek it out. Developers did a great job of communicating anytime they sent emails to the Product Owner asking clarifying questions and letting them know why certain information was needed. The tools we used have also been great at facilitating this communication. Jira, for example, was a great tool that allowed us to organize all our information, send out tasks, and update the backlog. Our use of daily scrum meetings allowed us to gain new information and insight as it became available and keep everyone on the same page. This sprint retrospective also allows us to review what we've done and where we can improve.

Now, agile isn't perfect it has its ups and downs. Agile was a good way to keep everyone updated, using the tools that we used. Its broad amount of flexibility saved time and money. With each iteration of the product, we were able to see our stakeholders' reactions in real-time and get feedback almost immediately. This allowed us to create a quality product, as we were able to fit the model of our project vision each time we got something wrong or a substantial change was made. On the flip side, this methodology could have been unproductive had we not had a productive team or good communication skills. Its reliance on multiple members of our team working together in conjunction with one another could have proved non-beneficial had we had bad team chemistry. Agile is also not something easily picked up by many. It requires a shift, a shift in thinking, a shift in how we operate, and a shift in work, put simply. Thankfully, that is why I am here. Helping everyone adjust to this new methodology is a job that takes a lot of experience and expertise.

All in all, was agile the best approach? Overall, the answer is yes. The ability to garner feedback from our customers and other stakeholders was crucial to our success, and the money and time saved by our use of this methodology allowed us to create a great product. So, in the future, I hope Chadatech adopts this methodology for its continued success in a world where things are constantly and rapidly changing.