Kevin Schroeder

5/20/20201

Sprint Review and Retrospective

Our team was tasked with developing a new website for SNHU Travel Agency that focused on selling travel packages to niche locations. We chose to work within an agile framework for this project and learned a lot about how an agile team functions. Our review will analyze the team's performance and recommend whether or not we should use the agile framework for all projects moving forward.

Team Dynamic

Our Scrum Master did a great job at starting the project off on the right foot with the sprint planning meeting. They worked with the entire Scrum team to decide what can be done in the coming sprint and come up with practical steps that will make that happen. Our Scrum Master did an excellent job in this meeting setting up time boxes for each task, including the planning meeting which was strictly held to 2 hours. They were also in charge of facilitating our Daily Scrum meetings in which everyone talked about what they did the previous day, what they were going to work on that day, and vocalized any potential roadblocks. They did a great job at keeping these meetings to a strict 15-minute timeline, stopping any side-tangents, and setting up additional meetings if needed to discuss topics more in-depth. The Scrum Master did a great job at using monday.com to refine our project backlog. The backlog broadcasts all the things our team intended to spend time on. The backlog is important because it set expectations with stakeholders and other teams and makes development time a fixed asset.

The Project Owner acted as a middle man between the development team and the stakeholders at SNHU Travel Agency and product users. The Product Owner did a lot of research about the company and its stakeholders to make their communications feel more personal and purposeful. In all communication with the stakeholders, the Project Owner was honest about the project's purpose, scope, risks, and approach and did a great job with building trust with the client. SNHU Travel Agency mentioned a few times that they felt like partners with us rather than a typical client-agency relationship. The Project Owner then developed user stories after gathering information from the stakeholders and the current users. These user stories gave a voice to the people that we are trying to serve and promoted collaboration between the Scrum team, the Product Owner, and the users to determine project requirements. The Product Owner finalized 3 high-quality user stories that shaped the development done in this sprint.

The Developers were the ones who brought this project to life. They were the ones who actually completed the items in the project backlog. Since we used an agile framework, the developers were focused on making small, production-level changes that they could show the rest of the team. They were also responsible for developing test cases to send to the Testers with detailed instructions on how to interact with the site to ensure that all functionality was working. The Testers took those test cases and conducted a thorough analysis of the site's behavior while going through these tests. There was a lot of o communication between the Testers and the Developers to clarify exactly what needed to be done and what the results should be.

Agile with User Stories

The Scrum-agile approach to the software development life cycle gave us a set of steps that we followed to complete each user story. The first step was to create the product backlog, a

list containing features to be implemented during the sprint. The agile approach uses the following template to fill out their user stories: As a [User Role], I want to [feature body] so that [User profit]. The agile principle the business people and developers must work together daily throughout the project helped us order the product backlog to first accomplish the most important user stories. For example, the Project Owner determined that creating a user profile that saves travel preferences is the most important to the customers. Since the agile manifesto states that the highest priority is to satisfy the customer through early and continuous delivery of valuable software, we knew that the user profile would be what we prioritize. The Developers told us that it was a large lift of effort, so we set the sprint timeline for 4 weeks in hopes that we could build out all of the user stories.

Once the backlog was ordered and the sprint timeline was set, it was now time for the Developers, Testers, and the Scrum Master to work on building the software. Scum meetings, which are a part of the agile approach, were conducted every morning to update the entire team on the progress of the sprint. We used a Trello board to keep track of the tasks. The Scrum Master would bring up the board in the morning with the team and make sure that every item is in the right column. The Scrum Master kept everyone aware of the priorities and made sure that the highest priority user stories are being accomplished before moving on to the lower priorities. For this project, we didn't start on the 'click a link to view top destinations based on my past vacations,' user story until the last week of the sprint. The agile approach is very strict with their time periods so if we weren't able to accomplish this task, we understood that it would be moved to the next sprint. Luckily, we were able to complete all user stories on time.

Agile with Handling Changes

A great aspect of the Scrum-agile approach is that it is set up to handle changes in requirements. When SNHU Travel requested that we change the requirements to focus on relaxing vacation packages, the Developers requested that the Project Owner updates the project backlog with the new requirement. Since the product backlog is a prioritized list, they were able to see how important this new requirement is compared to the other requirements in the sprint. Then, the Developers produced a time estimate for the new requirement and determined the requirements in the current sprint that may not get done or get pushed to the next sprint to fulfill the new requirement. For example, the Developers let the team know that they may not have time to start on the 'click a link to view top destinations based on my past vacations,' user story. The Developers request that the Project Owner communicates the estimate with the client to ensure that everyone is on the same page and that the client knew the effect this request will have on the final product and/or project timeline. The Tester had some initial clarification questions on how the new requirement is supposed to function in our situation. These questions helped speed up development time because the testing process went by faster. The Developers requested that the Testers test out the current requirement and imagine what the functionality might be like with the new requirements added. From there, the Testers can develop a list of initial clarification questions that can guide some of the details of the requirement development and speed up final testing time.

Team Communication

To promote transparency in an Agile development environment, I used a digital information radar. Information radars are a large, highly visible display of critical team tasks and information that can be accessed and edited by each member of the team. This enabled

transparency because everyone on the team is aware of the work of all other members and how that affects the end goals of the team. It kept the Product Owner and the stakeholders aware of the Scrum team's progress and allows them to identify concerns. The tool that I used for this information radar was Trello. This is a free tool that is easy to use and understand. It allows you to make digital cards with all of the requirement information and a checklist for specific tasks to do to fulfill the requirement. The team would then move the card into different columns that are labeled with progress categories. For example, SNHU Travel used a Trello board with columns for Upcoming, Gather User Information, User Stories, Tech Specs, Build, Review, Test, Deploy, Announce, and Completed. In the beginning, all requirements were put in the Upcoming column, ordered by priority. Once the Project Owner started gathering user information for the user stories, the card was moved to the second column. This process repeats in every column. The whole team had the ability to see exactly how each requirement is progressing and identify pain points throughout the process.

The daily Scrum meetings were also good for teammates to check the status of cards and voice their concerns about being held up by another requirement's progress. I tried to promote a culture of honesty and transparency during these meetings by giving the good and the bad about how my tasks are going. For example, I let my team know that I was behind on my work for this project and voiced the user story that may be affected by this. The hope is that everyone would feel comfortable enough to share their struggles and successes. Ultimately, the Trello board was the source of truth on progress.

Tools and Agile Principles

We used a variety of different digital communication tools to help us be successful. As mentioned previously, the tool that I used for our information radar was Trello. Our team valued the information that they were able to gather on the Trello board and how easy it was to use. The Trello board was updated every day at the daily Scrum meeting to make sure that everyone was on the same page. I even set up my own Trello board to keep track of what we did well and what we need to improve based on the feedback from our upcoming Sprint Review and Retrospective meetings. We also used email to communicate if team members had questions that were longer-form (defined as 3+ sentences). For shorter-form questions or situations where we expected a lot of back and forth conversation, we used Slack as an instant messaging tool.

As a team, we did our best to follow 12 agile principles that are based on the agile manifesto and we gathered from Agile Alliance (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). The first principle says that our highest priority is to continuously deliver valuable software throughout the project (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). We did this by producing production-level code by the end of this sprint. The second principle is to welcome change requests late into the project, which we were faced with in this sprint and had a method to handle (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). The third principle is to produce bits of production-level code on a preferably short timeline (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). We were able to accomplish this by timeboxing our sprint to 6 weeks with the understanding that if we didn't get the lowest priority items done, they'd be moved to the next sprint. Principle four is that business people and developers should communicate daily, which they did through digital communication methods such as Slack and email (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). Email and Slack worked for quick questions or questions between teams. However, the

development team followed the agile principle that the most efficient and effective method of conveying information to and within a development team is face-to-face conversation (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). We did a lot of work with Google Calendar to make scheduled meetings or quick round-ups. The sixth principle challenged us to build a culture around trust and motivate the people on the projects (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). The daily Scrums did a great job of keeping people motivated as our Scrum Master would always end the meeting by reminding us of the purpose and celebrating our personal accomplishments. Principle seven stresses that working software is the most accurate measure of progress, which we followed by making sure that the functionality of the current user story was finished and tested before moving forward (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). The eighth principle states that the agile processes need to promote sustainable development (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). It's easy for Developers to feel burnt out on a project if they are on a time crunch and are working late hours. To prevent this we always included Developers in the conversations about time estimates and encouraged them to add in extra time for inevitable unforeseen problems. Principle nine pushed us to strive for technical excellence and good design (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). Every one of the team was made aware at the start of the project that good work done slower than planned is better than sloppy work done quickly. The Testers really helped us determine if code or a design was made poorly. Principle ten is to promote simplicity, which our Developers were able to do by making good designs before coding and our Scrum Master was great for promoting this in our daily Scrums and additional meetings because they would stop any conversation from spiraling into something that is getting too complicated for the task at hand (*12 Principles Behind the Agile*

Manifesto | Agile Alliance, 2015). Principle eleven states that the best work is done by self-organizing teams, which is how our team was constructed (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). Finally, principle twelve states that the team should come together at regular intervals and reflect on what is going well and what needs improvement and our behavior adjust accordingly (*12 Principles Behind the Agile Manifesto | Agile Alliance*, 2015). We were able to accomplish this through open communication in the daily Scrums and our upcoming Sprint Review and Retrospective.

Final Recommendations

The Scrum-agile approach was very effective for the SNHU Travel project, and we would recommend using this type of approach for similar projects moving forward. The biggest benefit to the agile approach is that it allowed us to produce production-level code much faster than we have on past projects. Agile also helped us manage the large change request that the client made during development. In the past, our waterfall approach wasn't flexible enough to handle such a request and would cause us to lose multiple workdays to strategize how to handle this change. We also saw an improvement in communication and transparency with the daily Scrum meetings and the Trello board. The biggest con to this approach is that documentation gets left behind. This could end up hurting us in the future if SNHU Travel continues to work with us on retainer for site maintenance and we hire someone new. There will be a learning curve, but I think the pros far outweigh the cons.

Sources

12 Principles Behind the Agile Manifesto | Agile Alliance. (2015, November 4). Agile Alliance |; Agile Alliance. https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/

Atlassian. (2021). *User Stories | Examples and Template | Atlassian*. Atlassian. https://www.atlassian.com/agile/project-management/user-stories#:~:text=What%20are%20agile%20user%20stories,the%20end%20user%20or%20customer.

Dmitry Gurendo. (2020, January 15). *Scrum Methodology Phases which Help in Agile SDLC Process: 5 Key Steps*. XB Software. https://xbsoftware.com/blog/software-development-life-cycle-sdlc-scrum-step-step/#:~:text=Scrum%20framework%20allows%20implementing%20Agile,ready%2Dto%2Duse%20product.