**ChadaTech Sprint Review and Retrospective**

Jason Robbins

CS 250: Software Development Lifecycle

Tarik Iles

December 11, 2022

**Adapting to the Roles for Agile Methodology**

As far as the change in roles for adapting to the Agile methodology I feel as though there were minimal changes to them outside of our development and testing team. Our Project Manager came to be known as the Product Owner, keeping in close contact with the business sponsor and kept a backlog of items for the team to accomplish throughout the “Sprints” that we worked through while maintaining the overall vision of the project. Between the Product Owner and the Development Team. Was the Scrum Master, which provided a Supporting role to ensure that the team was a self-sufficient unit and led our daily Scrum Meetings to facilitate the overall progress of the project. The Scrum Master proved to be more necessary later, as the Dev. Team faced some issues regarding a change in direction from the business sponsor and found themselves lacking in self-organization when combining the different teams. Outside of these roles the Development, Quality and Testing teams merged to form one Development Team unit and worked together through-out the Sprint instead of the Individual teams through the waterfall methodology that we have been using up until now.

**Fulfilling User Stories**

As far as reaching our goals for the User Stories, I believe that we did exceptional in fulfilling all of the user stories that were provided thanks to the Agile approach. Each user story was ranked by ease of implementation and importance, with the more important core functionalities prioritized and the less important functionalities left for the team to work on later. Later in the project though we found that a core functionality of the site could not be completed in a day and was very difficult for some of our less experienced devs. What we found was the higher difficulty functionalities could be broken down into smaller parts so that the more difficult parts of the functionality could be broken down even further and easier parts given to the less experienced devs and the more senior devs taking the more demanding work. This allowed the functionality to only take a single day to completely implement with the team working with each other to compile the work. By breaking down the work in this manner we found an increase in development that we were pleasantly surprised with. It is thanks to the Agile approach that we were able to break up the work in this manner and use our teams’ strengths to the fullest.

**Adjusting to a change in Direction**

We did find a little bit of trouble during our development with a change in the core vision of the product. However, I believe that due to our team following the Agile method as opposed to the Waterfall method, we were able to react to the change in the direction much better than if we were still using the waterfall method. With SNHU Travel changing its core vision of the product to a service that offers health and wellness retreats at the forefront, Our Product took to the time to add and reevaluate all of the user stories for the product catalog. This was then brought to the attention of the team through one of our daily Scrum meetings and we focused on readapting the work we have done to fit the new vision of the product. If we were still following the waterfall method, I believe that we would be halfway through the testing phase of development, which means that we would have spent more time just to go back to the planning phases and essentially restart the entire project. Because of this I believe that the agile method is much better at reacting to sudden changes in design than the Waterfall method.

**Communication is the key to success with Agile**

Our success with Agile is mostly due to our priority change involving communication. We find that communication is important to the success of the Agile method, either through an Agile Project management software such as Jira, through face to face interaction with our daily Scrum Meetings, or from clarifying production details through email. An example of such communication is below when our testers had questions about integrating into the Development team when the Agile method was first introduced to our team:

“Hello Jason!

In order for our development to be successful, the development team needs to stay in close touch with the testers. The developers will keep the testers updated on our implementation plans for the items that get pulled into the sprint backlog. This way, test cases can be developed simultaneously as we begin developing the implementations for those items in the sprint backlog. How would you prefer we communicate our implementations to you? Would you like updates via email every couple of days? I believe the more we communicate about our implementations, the easier it will be for the testers to come up with a complete set of test cases.

I would also recommend getting in touch with our product owner and requesting thorough acceptance criteria in the user stories. Detailed acceptance criteria will help us in our implementations and test cases. As a developer, I would like to explain to the testers how we plan on proceeding with the implementations for meeting those acceptance criteria. I hope this will help you in forming your test cases. Please reach out to me if you have any questions/suggestions regarding the development process and/or our implementations!”

It is with communication like the correspondence above that provides reassurance to the fact that the Agile method is more collaborative than the waterfall method. It also demonstrates that testing now works more alongside development, rather than an after thought to development.

**Effectiveness of Agile-Centered Tools**

One of the tools that we use to communicate in such a way is Jira, an Agile Project Management tool that not only acts as a way for the devs to communicate with each other but also acts as a hub that allows the development team to see who is working on what user story that day and to also keep track of the total progress of the project. It does this by offering a variety of features including Scrum Boards to emulate the Project board used in our daily scrum meetings, roadmaps of long term planning and dependency management to make sure the higher priority functions are implemented first. Of course the tools do not replace the Daily scrum meeting but act as a guideline for team outside of the meeting and encourage communication apart from face to face. It also implements a report that tracks the progress of the project which can be used at the end of the week for the Sprint Review.

**Reviewing Agile methodology with SNHU Travel**

Switching over to the Agile method definitely had a learning curve for the team as we had to break out of our comfort zone and communicate more to ensure that we were meeting the vision of the project everyday versus maybe only communicating every other day to possibly every week for updates. I feel as though if our team was not as cooperative as they were, the Agile framework would not have worked as well as it did. With so much communication needed, I believe that if this were to be implemented in a very large team that it would not work as well as it did with our 6 member team, as too many people present at daily scrum meeting might lead to gaps in who could present what they did and what they are doing for the day. Also if the team was comprised of more junior level developers that it would be much more difficult to get consistent daily progress.

I believe that because our team was small and had mostly experienced developers that Agile was the better approach for the SNHU Travel project, compared to the waterfall method. Our team collectively completed our goals, communicated effectively and overcame challenges by modifying those goals and tracking our progress through the entirety of the project together. I am not sure how effective the Agile method would be for larger teams or longer term projects, but I can fully endorse the Agile method for smaller projects for ChadaTech in the future.