Joseph Silva

SNHU CS-250 Final Project

Chada Tech: Sprint Retrospective

Professor: Ryan Woolever

12/11/2024

In this final paper I will discuss my experiences while taking on multiple roles within an agile/scrum team for the SNHU travel project. I will also discuss some other topics that are important to an agile/scrum team such as completing user stories, handling interrupts, the importance of good communication, organizational tools and the evaluation of the agile/scrum process as it applied to SNHU travel.

The first role we assumed during the course was the scrum master. A major responsibility of the scrum master is to ensure the scrum framework is followed by the team. As we explored this role, we learned that the scrum master is responsible for planning and facilitating the daily standup, sprint planning, sprint review (although in some agile applications, this is facilitated by the product owner), and sprint retrospective meetings. They are also responsible for starting and stopping the sprints and making sure all the scrum events are properly timeboxed. For instance, sprint planning should be kept to four hours for a two-week sprint, the sprint review two hours for a four-week sprint, the retrospective one and a half hours for a two-week sprint and the daily standup should be fifteen minutes daily. It is important the team sticks to the suggested timebox for each event so they can spend their time adding more value by developing and testing the product. In the sprint planning meeting the scrum master will lead the estimation process for user stories which can be done using methodologies such as affinity grouping or agile poker. This is an important part of planning the sprint as the team can match user story points with its overall capacity, so the work is not overloaded. The backlog refinement activity is also set up by the scrum master but could be facilitated by the product owner and should be kept to three hours of a two-week sprint. This activity will ensure the team is familiar with the user stories in the backlog and will be ready to estimate them in sprint planning. The scrum master should also collect impediments in the daily standup meetings and escalate them in scrum of scrum meetings when necessary.

Next, we took on the role of the product owner. The product owner is responsible for creating the product vision, managing the product backlog, collaborating with stakeholders, customers and making themselves highly available in case there are questions about the work that needs to be done. It is important that the product owner stays in contact with stakeholders and customers to ensure the desired requirements and features for the software are being prioritized throughout the development process. The product owner is responsible for creating user stories that are in small enough increments such that they can be accomplished over the course of a two-week sprint. As the product owner, we were tasked with creating user stories to start our product backlog. The requirements for our user stories were gathered from interviews with stakeholders and customers. With this information we were able to use the “who”, “what” and “so that” criteria to clarify the purpose of each user story. Having a clearly defined product backlog makes it easy for the team to estimate and plan its work.

Along with assuming the role of product owner, we had our first development activity of the course. We were tasked with creating a top five destinations list, using a Java form, where we ranked five vacation destinations one through five with one being on top and five being on the bottom. We also needed to provide a photo for each of the destinations we chose. This is relevant because when we took on the role of developer, the requirements changed, and we had to pivot. This was an example of the flexibility benefits of the agile process versus the waterfall method where the change came during the development process as opposed to after the product was fully developed. During the process of developing software for SNHU travel we learned that instead of a top five destinations list, the customer wanted a top five destinations slide show that featured vacation spots that specialized in detox and wellness. This required code changes from the previous implementation. As a developer I would work closely with the product owner to ensure the requested changes were finalized and broken down into clearly defined user stories before I began work. In addition, I would check in with testers on the team to ensure their test cases could be modified to be in line with the requested changes. While face-to-face communication would be ideal, I drafted up an email including the product owner and tester with follow-up questions so we could get on the same page. This planning would allow for the requested changes to be completed in a timely fashion.

Finally, we took on the role of the tester and it was emphasized that the quality of the software is the responsibility of the whole agile/scrum team and not just the testing specialist. While in this role, I determined that the acceptance criteria and software requirements are the most important aspects of user stories for creating test cases. It is advised that the tester works closely with the product owner as they are the ones with a direct line to the end users and stakeholders and could provide valuable input on the most important testing scenarios that needed to be covered. In the context of SNHU travel the testing specialist may have had all their test cases done and then things changed. For this reason, I reached out to the product owner via email asking for clarification with some follow-up questions. Their feedback would be critical in helping me develop my test cases.

During the development process for SNHU Travel, we worked on the user story for creating a top 5 destinations list for the project. To do this, we had to update a Java form to include the top 5 destinations of our choice and include a picture for that destination. Once the coding was complete, we tested it to make sure all the functionality was working as expected. For my version of the application this included validations such as making sure everything was correctly listed, there were no misspellings, my name as well as SNHU Travel were properly displayed, and all pictures were roughly the same size, so the display was uniform. Once the development and testing were complete, the user story could be considered done. This is an example of a product increment that could be added to the larger project and demonstrated in the sprint review for end users and stakeholders. In fact, the demonstration of the current product increment could have potentially led to an interruption as changes were requested. It is feasible that while the top 5 destinations list was being demonstrated during the sprint review, the stakeholders or end users did not like the list format and requested a slideshow featuring vacation options specializing in detox and wellness. Using the agile development process for this project gives the team the flexibility to implement these changes possibly as the next product increment. This could theoretically become a new user story to be picked up over the next couple of weeks, where the code could be updated and the changes tested. Had this been decided later in the development process it could have resulted in higher costs, and more risk.

It takes good communication within an agile team for it to run efficiently. During our development process we needed to reach out to the product owner and tester as a developer via email. This communication was a crucial step from the development side to verify exactly what was being implemented after changes were requested. From the developer, some sample questions to the product owner were:

1. I understand they want a slide show instead of a countdown list.
   1. Are there any details about the slideshow I need to know?
   2. Do they want anything else added when compared to the current list format?
2. They want the focus to be on detox/wellness vacations.
   1. Do they want to feature spas, yoga retreats, both or something else?
3. Are these the only changes that are requested?
4. Are we adding one or two user stories for these changes?

And to the tester:

1. How big of a lift will it be to change your test cases to be in line with these changes?
2. Will you be able to modify any regression suite?

I feel these questions are clearly communicated and would kick start the discussions about the newly requested changes for the team. These questions should be answered before the developer and tester start working on the new user story.

The organizational tools that would help the team be effective during the SNHU Travel project would be Jira, Microsoft Teams and Zoom. I am most familiar with Jira as a tracking tool for agile software development. I find it to be user friendly when creating and updating stories and more importantly it has many features that benefit agile software development. Within Jira teams can estimate user stories using interactive sessions of agile poker, provide anonymous feedback during sprint retrospectives and analyze data from velocity, burndown and burnup charts to determine the team’s effectiveness. In addition, it provides powerful query filtering tools and board customizations so the team can track all their work in one place. Microsoft teams and zoom are great for remote meetings if all team members were not all in the same location. For our scrum team the daily standups would have been very valuable, especially after things changed, so the team could ensure they were all on the same page with development and test. The sprint planning meeting would also be important as in this case the product backlog may have needed some reprioritization.

The pros of using the agile scrum development process for the SNHU Travel project was certainly the flexibility agile offers by using incremental development and quality of the product. Since end users and stakeholders are looped in throughout the process, changes can be requested and more easily implemented. Had the waterfall method been used the need for change would have been discovered much later in the process and the scope of making changes would be much larger costing the development team time and adding risk due to a potentially more limited testing runway. Product quality should be higher with agile as testing would be done early and often alleviating the possibility for late breaking code bugs and releasing a product with deferred defects. A con would be some of the user stories may have been too large for a product increment and more refinement was probably needed. Additionally, inviting change can always lead to unforeseen additional time, cost and resources that can affect work estimates. In our case the product backlog would have needed to be revisited after changes were requested and tasks developers were prepared to start working on could have been pushed out. I believe agile was the right choice here. Website development is something that could be fluid as design-based changes such as organization of a page, look and feel seem likely. Having constant check-ins with end users would make this easier as opposed to developing it end to end and having it not be what the customer expects.