# **CS 250 Final Project**

Benjamin Verrill

[Benjamin.verrill@snhu.edu](mailto:Benjamin.verrill@snhu.edu)

Southern New Hampshire University

During this Sprint, we were tasked to develop a web application for SNHU Travel that would allow their customers to see potential vacation destinations to visit. ChadaTech made the decision to trial Agile methodologies with this project being the pilot. We created a team consisting of a Scrum Master, Product Owner, Developer, and Tester to work in conjunction with each other in order to complete the task.

The Scrum Master’s role was to organize and coordinate Scrum events for the team to attend and participate in. These included Sprint Planning, Daily Scrums, Backlog Refinement, Sprint Reviews, and Sprint Retrosprectives. Each of these events were moderated by the Scrum Master to ensure they stayed on track and that Scrum principals were followed. The Scrum Master also acted as a coach and advisor, ensuring that everyone understood not only the Scrum principals but why they were being carried out, as well as sharing their own experiences to help the team when facing challenges. Finally, the Scrum Master worked to remove any bottlenecks the team faced. During each Daily Scrum, each member of the team was to speak out about what was impeding them. The Scrum Master would then do their best to come up with solutions to remove these roadblocks so that the team could focus on their work and not on their impediments.

The Product Owner’s role was to interpret the client’s needs to add to the backlog, and to prioritize the backlog. In our project, the Product Owner helped the team to understand what SNHU Travel was trying to do, which was create a vacation destination application that would drive more traffic, and thus more sales, to their website. They worked with the team to develop user stories, which were then added to the backlog and prioritized so that the team could determine what was to be included in the upcoming Sprint. The Product Owner also updates the backlog as requirements from the client changes, and helps the team revise their strategy for meeting the Sprint goal. We saw this when the SNHU Travel decided that they wanted the focus of their destinations to be on Wellness/Detox destinations. This required an update on the user stories and the potential rearrangement of prioritized user stories within the backlog so that the completion date stayed the same. The Product Owner also helped with the Definition of Done, which describes when a project is completed enough to be released.

The Product Tester ensured that the product worked the way it was intended, and that the code was written as efficiently as possible. Taking the user stories, the tester would have an idea of what functionality the client was looking for, but they would then have to determine the reasoning behind those feature requests so that the product met the client’s expectations. Once this was determined, the tester developed test cases for each user story. These were pass/fail type tests, such as clicking a link titled “Most Popular Destinations” taking the user to a list of the most popular destinations. These tests would include a list of inputs and expected results. These ensured the user stories worked as intended.

Finally, the developer wrote the code. They took the information from the user stories and the goals of the Sprint, estimated the level of effort for those user stories, and wrote the code to create the product. They worked with the Product Owner to ensure that they understood the user stories and the reasoning behind them, and helped define what user stories would be included in the Sprint. In the SNHU Travel project, they wrote code that, in the beginning, created an application that showed a list of 5 top destinations that included a picture, location, and description of each. As the project requirements were revised, this became a slide show of Wellness/Detox destinations. They also attended the Daily Scrum to update the team on what they’ve done so far, what they’re going to be working on, and what was holding them up. This allowed for greater collaboration and for the Scrum Master to help them more efficiently get their jobs done.

Utilizing Scrum practices, the team was able to develop effective user stories that ultimately led to the type of product the client was asking for. The Product Owner met with the client and potential users to define what sort of features were important to them. This included things like a list of popular destinations that they could see as well as ways to filter that list. During the Sprint Planning phase, the Product Owner was able to work with the rest of the team to better define those user stories and prioritize them based on the desires of the client. These user stories included things like there being a list of top destinations, what each destination visual would include, how lists could be filtered, etc. This allowed the team to define what was going to be included in the upcoming Sprint, and to help set expectations of what would be released at the end of the Sprint with the client. The Daily Scrums helped the team to understand what everyone was working on and what areas they may have needed assistance on so that the project stayed on track. This level of collaboration helped the team to run more efficiently for meeting deadlines.

One of the major benefits of using Scrum practices is that it allows for the ability to be flexible when project requirements change. During the SNHU Travel project, the team was told that the client had decided to narrow their focus to only destinations related to Wellness/Detox. Since the team met every day during Daily Scrums, they were able to be updated as to these changes in a fast and efficient manner and immediately make adjustments to the user stories and Sprint goals so that these changes wouldn’t effect deadlines. The Product Owner was able to collaborate with the developers and testers to ensure that the backlog was reprioritized to meet deadlines and that the developers and testers had the information that they needed to make the required updates. Using a waterfall approach, this would not have been possible as all requirements are determined up front and changes are only considered after a project is completed.

A major part of efficiently completing these types of endeavors requires good communication. We talked about how Daily Scrums allowed the teams to daily update each other on what each member had done, what they were going to be working on, and what was impeding them. We saw how this helped when requirements changed midway through the project. However, it’s not always possible to meet face to face. Another effective means of communication is through email. In the following example, I used an email as a developer to the Product Owner and tester to better define the user stories. Because of the updates to the project requirements, I needed additional detail to ensure that what I was going to build was going to meet the expectations of the client. To do this, I asked pointed, detailed questions that gave me a better idea of what to do. This eliminated any requirements that could be misinterpreted and gave us a better chance of increasing customer satisfaction. That email went as follows:

*It’s my understanding that we need to make some updates to the website based on some changes in requirements from our client. It sounds like they’ve decided that the destinations they want made visible to their end users are going to be more specific and based on wellness and detox types of destinations. We can make that happen, but I was hoping to get some clarity before we move forward:*

1. *Will we only be showing these types of destinations/vacations on the site, or is this just going to be at the forefront and end users can search for other styles of destinations as desired?*
2. *Are we keeping the same type of information/summary for each destination as before, or is there anything else we need to add?*
3. *Has our deadline for completion been pushed back, or is the plan to deprioritize other user stories?*

*Once we get some clarification here, our team will happily move forward on making these changes. Looking forward to your feedback!*

Another email I used was as a tester to the client. This was done at the beginning of the project and used for similar purposes as the email to the Product Owner above in that I was looking for additional details from the client to ensure the website was going to work the way they expected. Here, I broke the email down by user stories and asked for additional details for each one so that the client would be satisfied when they saw the finished product. Using the response from the client, I would then be able to make any updates needed to fit the requirements. That email went as follows:

*I received the user stories for the features you’re looking to have added to the product, and it’s my job to make sure these features are tested so that they work as expected. While we can definitely provide what’s being asked, I wanted to get some additional details so that we can make sure it’s built the way you’re looking for:*

*User Story One:*

* *How do you want the list of most popular destinations to be presented? As a list? A slideshow? In more of a grid view?*
* *How should they be listed? In ascending or descending order?*

*User Story Two:*

* *Do you want only vacation types that have been marked as liked or purchased in the past as being suggested as relevant, or also to include recent searches?*
* *How should they be listed? In ascending or descending order?*

*User Story Three:*

* *How do you want the price range to be displayed? As a sliding scale? Text boxes?*
* *Where should the feature be placed on the site?*
* *If the user has a price range set in their user account, should they still be able to customize it from the list view?*

During the project, we used agile project-management tools like Jira to more efficiently communicate with each other. This was a valuable tool to use because we are geographically spaced out. These types of tools gave us the ability to quickly understand what everyone was focused on and what had been completed and still needed to be completed. When we needed to update the requirements to the project, agile project-management tools gave us the flexibility to do so without any great interruption to our processes. Combined with the Daily Scrums, this allowed us to effectively pivot our efforts so that we could make the required changes and still meet our Sprint goals.

As a pilot, this project was meant to determine whether a Scrum-agile approach was a more efficient way for completing a project. In this example, I would say that it certainly was. This approach gave us the flexibility to make changes to the project as needed and to quickly pivot, whereas a waterfall approach would have been more rigid. This was because of the constant communication the team had with each other and with the stakeholders. The transparency of what everyone was working on or challenged with also helped from a collaborative standpoint to ensure tasks got completed on time. This also helped to ensure customer satisfaction, because such communication means that we were able to update our goals quickly to meet the client’s changing needs during the Sprint, whereas with a waterfall approach any suggested changes would have to wait until the end. The client also was able to see a finished product faster. This was because of the adoption of Sprints, which allowed for short term releases to get a functional product in the hands of the client in weeks instead of the months it might take for a waterfall approach. However, the agile approach was not better in every way. One of the keys to the agile approach is to have consistent teams working together. This allows for more efficient estimation of effort for projects. Without this consistency, more unpredictability gets introduced within projects and estimations are harder to accomplish. Another downside to consider is that documentation tends to become less important. Since the teams are in constant communication, less documentation is needed to understand why code was written for different areas of the project because the intention is already known. However, should another team go to work on that same piece of code, it would be more difficult to understand why things were done the way they were. That said, I think that the benefits far outweigh the cons in the Scrum-agile approach and that it was the best approach for this project and should be considered as the methodology for future projects.