**Final Project**

CS 250: SDLC

December 13, 2024

Cole Flournoy

Every member of the SNHU Travel team played a significant role in the success of the project. One of the most critical team members was the Product Owner, because she acted as the bridge that connected the rest of the team to direct insights from the client and potential users. It was the Product Owner who conducted the interview with three potential users who were each using another competing service. This interview set the foundation for the early requirements and scope of the project. (She also communicated change with the wellness thing). Another member of the team who had a big role was the Scrum Master. He started the project off on the right foot by creating the Agile team charter and sharing it with the rest of the team. This document specifies the roles of each team member and serves as a foundation for how the team will work together, make decisions, and collaborate throughout the project. It has some high-level aspects, such as the objectives and risks of the project and some agreed upon values and principles for the team to uphold. The team charter also has some more detailed elements, such as specific communication guidelines and the frequency of Scrum events. The Scrum Master brought a lot of value in these structural elements that helped organize the team and hold them accountable to Agile practices. Agile development is very flexible by nature, and intentionally so, but having someone to guide the team through and maintain that structural foundation is very important. Although they had a lesser variety of different responsibilities, the Developer and Tester were no less important for the success of the project. They worked closely together to handle the implementation of functional, well-tested code that met the requirements of each story. This is critical because these two roles determined not just the quality of the product shipped to the client and end users but the very existence of the product in the first place. Specifically, the Developer created a list view of top ten travel destinations in the application based on a user story detailed by the Product Owner, and the Tester took three user stories and translated them into tests that had detailed steps based on user actions and specific pass or fail criteria for each step.

The Scrum-Agile approach was very helpful in developing and executing on user stories, especially because this approach prioritized user feedback and adapted quickly to changes. The team started by gathering requirements for stories by speaking directly to potential users of the product, which may seem like an obvious way to understand user needs, but this is actually a hallmark of Scrum-Agile development that isn’t always a given in other approaches. Having tickets that described user needs to be addressed rather than tasks to be executed kept the users’ needs at the forefront of each team member’s work. It also gave the opportunity for a significant amount of flexibility and freedom in the implementation and testing of each story. The Scrum-Agile approach, with its organized backlog and short sprint cycles, kept the team focused on the highest priority stories and manageable chunks of work at all times. These practices combined allowed team members to work quickly and without excessive pre-planning or bureaucratic oversight. Because team members were never too deeply invested in large, complex projects with rigid implementation details, this approach also allowed the team to change course quickly and effectively as needed. When the original list view for top destinations was changed to a slideshow view, the Tester updated his test cases seamlessly and quickly enough that the release did not need to be delayed. He did so by requesting additional information on the new requirements so that the scope of the changes was clear. When the content of the top destinations page was changed to focus on wellness and detox travel packages, the Product Owner shared this in a face-to-face meeting with all of the other team members present. This gave the team the opportunity to talk through the change together, ask clarifying questions, and ultimately feel secure in how the change would impact their work. Prioritizing face-to-face communication is another key aspect of the Scrum-Agile approach.

Communication in general is a big part of what allows the Scrum-Agile approach to work so freely. Agile is meant to be extremely collaborative. As with the structural organization that Scrum events bring to a project, frequent open communication is essential for reigning in the freedom that Agile gives each individual on the team. One main criticism of the Waterfall methodology is that it isolates individual members of the team, making the development lifecycle very susceptible to inefficiencies. Our Agile team can be just as inefficient if we are not able to communicate regularly and effectively. Scrum events like daily stand ups play a significant role in fostering this collaborative communication, but additional communication is also required. The following is an email that I sent to the Product Owner when acting as the Tester for this project:  
 *Hi Christy,*

*I’m drafting the initial test cases for SNHU Travel, and I have some clarifying questions about one of the user stories. For user story #2:*

* *Is the user profile page going to be limited to only travel history and preferences, or does it have a larger purpose?*
  + *For example, do we plan to put other user settings on this page?*
* *Do you have any additional guidance or design plans for what the user travel history will look like?*
  + *Will the travel history show only travel booked on our site?*
  + *Do users need a form to input their previous travel history booked elsewhere?*
  + *What information will be included in the history and how will it be displayed?*

*Thanks!*

*Cole*

What makes this effective is foremost the directness of the email. It is relatively short and doesn’t have any fluff. Everything communicated is either essential context or a direct question to be answered. The email is also very organized; the Product Owner does not need to go looking for the point of the email, because it is clearly stated in the very first sentence. She is also able to glance over the email to see my questions easily, because they are formatted as a bulleted list, rather than a long paragraph.

One of the key Agile principles that made our team effective is related to the communication described above. As I mentioned before, we prioritized frequent in-person communication as a team, and this had a big impact on our ability to work efficiently throughout the project. This priority can be seen in Scrum events such as the daily standup. These standups also helped us understand each other’s work and take advantage of opportunities to collaborate, especially cross-functionally. Prioritizing interaction and direct collaboration with team members who have different roles is another important Agile principle, which can be seen in the open dialogue of emails such as the one above and in pair programming sessions that can be introduced when progress is discussed as a group during daily standups. It is especially important for developers to work closely with business people so they can maintain a connection to the larger goals of the company and of the end users. Two other Agile principles that helped us to be successful were delivering working software frequently and welcoming changes to requirements, even late in the development process. The first can be seen in the short sprint cadence chosen, the sprint meetings scheduled and managed by the Scrum Master, and the segmentation of user needs into individual, manageably-sized user stories. The latter principle is evident in the fluidity of our team’s response to changing requirements.

The benefits of using the Scrum-Agile approach for the SNHU Travel project have been discussed at length already, but it allowed the team to do a few key things that would have been more challenging with a more traditional strategy: collaborate effectively, particularly cross-functionally; prioritize users’ needs and feedback; manage the scope of individual chunks of work and thereby deliver working code more regularly; and adapt easily to changes while minimizing wasted time or resources. In my opinion, using the Scrum-Agile approach was the right choice for this project for all of those reasons. In a system like Waterfall methodology, each of the aforementioned benefits is harder to achieve because of the more rigid, pre-planned structure of its software development lifecycle. Agile is not well-suited for projects with more rigid constraints build in, such as a hardware project where iteration and continuous development might be much more challenging or, in some cases, impossible. But this does not apply to SNHU Travel’s situation. They are trying to build a product that meets the needs of everyday users, and they can, and should, iterate quickly in that situation. The consequences for being wrong aren’t dire, and the Agile approach encourages feedback and adaptation in the face of changes. Many of the downsides of using the Scrum-Agile approach for this project are difficult to see until further down the road. For example, long-term planning and maintaining documentation can both be challenging when the priority of Agile is to move as quickly as possible in short sprints. While these are good considerations for SNHU Travel to keep in mind as the project continues, I do not think that they outweigh the benefits gained from using the Scrum-Agile approach.