**7-1 Final Project**

Robert Rak

Southern New Hampshire University

CS-250-R1828

Haruka Konishi

10/15/2023

**7-1 Final Project**

**Team Contributions**

All roles on Scrum-agile Team provided unique and essential contributions to make the SNHU Travel project successful. The Scrum Master, who could be viewed as a mentor or guide for the team, is responsible for setting the example by taking the lead in activities and eliminating any impediments to the team’s success. An example of this leadership role in the SNHU Travel project is how the Scrum Master went first in the Daily Scrum, answering the three primary questions that should be addressed (What did I do yesterday, what will I do today, and what is blocking me?). She modeled the kind of answer that would be expected so that the other team members would be prepared when it was their turn. Another example is where she volunteered to teach team members about Test Driven Development. This kind of initiative is exactly what makes for a great Scrum Master.

The Product Owner develops the user stories which form the basis for the features of the product. This team member is the visionary, responsible for connecting the wishes of the user or stakeholder to the abilities of the team members that can bring them to life. In the SNHU Travel project, there were several suggestions from customers as to what they would like to see in the final result. One customer wanted top destinations listed, another specified how large the range should be, and a third recommended that the list be tuned according to a user’s profile or travel history. This set of suggestions culminated in a “Top Five Destination List” user story with specifics on what should be implemented, which gave clarity to the team on what kind and how much work would be required in order to achieve it.

The Tester/Developer creates the features from the user stories and designs tests to ensure that they meet expectations once completed. Without this team member, there is no product. An example of what this team member did for the the SNHU Travel project is the “Top 5 Destination List Test Case”, which was formed to allow the evaluation of this particular functionality. A set of pre-conditions were described (“being on the main SNHU Travel website”), followed by inputs, and then the expected results. The inputs consisted of specific steps taken, like clicking links or verifying that the destination list was ordered.

**User Story Completion**

The Scrum-agile approach allowed for the maximum in flexibility and transparency, enabling everyone to have what they needed in order to complete the user stories. Given how the Product Backlog was open for the whole group to see, it made it very easy for the Tester to establish the necessary pre-conditions, inputs, and expected results. If there was any ambiguity, a request for a clarification could be issued to the Product Owner. The decentralized nature of the team, (there exists no higher authority among team members) made it possible for everyone to work independently and as quickly as possible until a lack of information became a blocker.

**New Project Direction**

As an example, midway through the project, it was decided that a slideshow style presentation on the destinations would be preferable to displaying them all at once. This necessitated changes to the code, but also the test cases. The agile approach allowed the Tester to request clarification from the Product Owner, who then provided a wireframe in the new format, which allowed the Tester to see which adjustments had to be made to the tests to fit the new requirements.

**Effective Communication Sample**

*“Please share your insights on how these stories should be prioritized, according to your view of both value to the business and dependencies. I’d also love it if we could collaborate both during the upcoming sprint meetings, but also one on one, so that your feedback can help us align the product with your vision.”*

In this sample, I requested insights from the Product Owner, and invited them to work personally with me to increase the chance of success for our project. This demonstrated that I valued their opinion, and that I wanted to work together outside of the context of Sprint meetings. I believe this will not only encourage them to contribute more, but also perhaps inspire them to initiate one-on-one meetings of their own with other team members when they feel the need to do so. I think if communication is limited to sprint meetings, they may begin to feel like hoops that people must jump through instead of being considered as opportunities to acquire more information to help you accomplish your goals.

*“Please adhere to the agreed-upon testing schedule so that we can address issues promptly. Your timely feedback is invaluable if we are to maintain a smooth development cycle. If there are any doubts, or if we can provide any insights into what expected behavior should look like, reach out to the development team with questions.”*

In the prior sample, I requested that the Tester maintain discipline with regards to the schedule, in order to avoid creating blockers for the project by accident. I made sure to ask in a way that was respectful and not accusatory. I also encouraged him to ask questions first if there was any confusion for him. This way he already has the development team on his mind the minute that any questions pop up while reading the requirements.

**Principle and Tool Evaluation**

The number one principle that aided the team was responding to evolving requirements over sticking to a fixed plan. There were countless examples where requirements changed, or something was updated that made another part of the project obsolete. Particularly in the domain of software, with its rapid development, I don’t see how you could operate with a fixed plan. Emphasizing effective communication over processes and tools is another fantastic principle. Everyone has a useful, but limited, perspective on the overall state of the project, so it seems to me that cross-communication is an absolute must to prevent us from wasting time operating on outdated information. Continuous delivery is the final principle that really struck me as useful and common sense. Attempting to achieve perfection on the first try is a major mistake. Generally, it seems better to have smaller features that can be iterated on in the future. This way you have something tangible to demo, something to be excited about, instead of nothing. While the features may be limited at first, there is nothing like actually using a product for yourself when it comes to determining what features it might need in the future. Often you can come up with new features simply by considering what you dislike about the current state of the software. There were many examples of this in the SNHU Travel project, such as the change from displaying all locations to the slideshow. It may not have been possible to know that you preferred the slideshow format until you experienced seeing the whole list and feeling overwhelmed by it. Additionally, the Product Backlog was a great tool for managing expectations and allowing people to narrow their work focus.

**Scrum-agile Evaluation**

The SNHU Travel project began with a set of desires, not detailed specifications. This made Scrum-agile an excellent approach to the project. There was a great deal of ambiguity in terms of what the requirements were, or even what changes needed to be made. In a typical Waterfall project, this would be disastrous and result in large delays. Under Scrum-agile, iteration is not only acceptable, it is preferred. Asking questions and figuring out the details later is not a problem either. Team cross-communication cultivates new ideas and brings perspectives that might improve the product beyond what was envisioned at the start. There were no cons to using Scrum-agile. I do think that in the case where the details are already known, that it may not be the best approach. If the project is simply a matter of implementation where you know the requirements won’t change, then it may become an obstacle in that it could enable communication overhead.