**Chada Tech Sprint Review & Retrospective:**

**A Comprehensive Analysis of the Agile and Waterfall Approaches**

Austin Siegel

Southern New Hampshire University

CS 250: Software Development Lifecycle

Professor Eugenio Rodriguez

April 15, 2023

Welcome to the SNHU Travel Project Sprint Review and Retrospective. I would like to begin by thanking everyone on the team for playing their part and going above and beyond in demonstrating the tremendous benefit of he agile methodology. We will be spending time in this review and retrospective to look back on the entirety of the project to improve out efficiency for the next project.

Each role within the team played integral parts in order for this project to be a success. The sprint methodology relies on quick cycles and consistent communication across all parties. This is where our scrum master truly shined, as they were the one who led our daily scrums and set the tone for the daily tasks. If not for the scrum master, the revised test cases from the user stories may never have been considered. We relied on the quick agile cycles to implement the original user stories, received feedback from the product owner who relayed that information to the scrum master to be discussed at the daily scrum. Even though there was consistent communication between developers, testers and the product owner, the scrum master served as a conduit for ideas and feedback coming from outside of the team. The product owner was our agent on the outside. The agile methodology demands many team members pay close attention to the daily tasks, leaving little time to get feedback from external sources. Our product owner took the time to develop relationships with testers and potential end users and get their stories to relay back to the team. Lastly, our developers played an integral role in actually developing the software. The population of the team is comprised mostly of developers since their daily tasks can change on a regular basis. They needed to be modular in their ability to focus on different tasks, and, more importantly, being able to leave a task uncompleted if it is no longer considered important by user stories or if another task is suddenly considered a higher priority.

The best example of the team fulfilling all of their roles can be found with User Story Test Case #1: Accept a budget range based on user input. The original project did not call for such a filter and only showed the top destinations based on which locations were being booked most frequently. We took advantage of the work done by Valpak in the past and incorporate an architectural Kanban style to our teams’ style. “The portfolio Kanban is a recommended practice in the Scaled Agile Framework that was implemented by Valpak. The purpose of the portfolio Kanban is to provide a way to plan, prioritize, and manage a portfolio of business epics. It makes upcoming work and works in progress very clear and noticeable, helps with product development flow and can be a key factor in getting to desired goals instead of team or program agility, and making business outcomes more efficient and accurate. (Cobb, 2015). The product owner found this information via user stories and relayed it to the team via the scrum master, who then integrated it into the next daily scrum and assigned it to our developers. Thanks to the Kanban style, all of the priorities were visible to all team members, rather than isolated with a few individuals. The timeline for this event from start to finish was only a few days; just one example of how quickly the agile methodology functions and how much versatility it demands from the team.

Ideas such as this appeared on our radar all of the time and would be neglected if we did not utilize certain agile-based tools when the project took an unexpected turn. We utilized the burn-down chart format to accommodate for such unexpected changes. The burn down chart has an x and y axis with the x-axis representing time to project completion and the y-axis accommodates user story points. As daily tasks were completed and new information came in from the product owner, we were able to perform triage on the user stories and the daily task that we believed demanded more attention. The burn-down chart implementation really set the tone for how our team experienced the agile methodology because it gave us a visual illustration of how quickly our task can change. This conditioned us to be modular in our mindset and to not be afraid to put parts of the project on the back burner if they were no longer considered a priority. This can be seen when our team received revised test cases from our product owner regarding users wanting a spot on the SNHU Travel to leave feedback. This was not originally in the requests for the application, but users proved that demand for such a widget was high. Thanks to our burn-down chart format, we were not taken by surprise and were able to accommodate the request.

The backbone of our success with the agile methodology was our communication within the team. In fact, agile demands effective communication to be successful at all. There are not hierarchal barriers to communication between team members of any status. All ideas were considered during our daily scrums and bounced between team members for constructive criticism in order to refine them to their maximum effectiveness. I spent time as a tester for the SNHU Travel project and had some great conversations with one of our team’s developers, Jariel Armanza. I brought up with them the idea of utilizing burn-down charts to manage our tasks derived from user stories. This was a good idea in its’ own right, but lacked direction and detail for how to implement it. Armanza stated, “I would advise you to be a little more specific about how you want to integrate user stories into your testing procedure. Will you, for instance, utilize particular testing scenarios based on user stories or will you search for software optimization opportunities based on user feedback? Giving additional information on this area of your testing process can assist your team better comprehend your strategy” (2023).

It was through the use of tools like the Gantt Chart and the burn-down chart that we were able to channel our productivity and through the agile principles of respect, openness and communication they were fueled. The ideas coming in through user stories and testers were simple enough to comprehend on a base level, but there needed to be a way to triage them with respect to the time remaining for project completion. The Gantt Chart and burn-down chart gave us a visual illustration of not just what the project tasks were, but how much time they would take to implement. To this effect, we treated the tasks like patients in an emergency room. What needed the greatest amount of attention with the catalyst being the priority of the user story. This is what became our cycle; one that could be implemented at a moment’s notice, even towards the end of the sprint cycle. There was no time for cryptic words, we could only trust in our team members and respect their ideas to best approach the tasks and accept constructive criticisms.

Overall, the SNHU Travel project was a grand success through the utilization of the agile methodology. The pros of agile can be seen in its communication. Honest and objective communication is what drives ideas and tasks during our daily scrums throughout each sprint cycle. There are several tools that are simple to use and illustrate our tasks and their relationship with time on the project. Our team communication brought these tools together and showed why agile is such an effective method. The cons to agile are not in its tools or infrastructure, but in the communication of the team using it. Agile is a wonderful tool, but it will fall short if the team’s communication skills are not strong enough to speak honestly or objectively enough. The team, not the method, is the key to success or failure with agile. The SNHU Travel project had a wonderful team that was open to change and strong/frequent communication. Our baseline skills were strong enough to make the agile approach the right choice for the project.

**References**

Armanza, J. (2023). *Vision quest software case study.* [Discussion]. Southern New Hampshire University: CS-250 Software Development Life Cycle Course. <https://learn.snhu.edu/d2l/le/content/1270022/viewContent/23125420/View>

Cobb, C. (2015). *The project manager’s guide to mastering agile: Principles and practices for an adaptive approach.* Wiley.