**7-1 Final Project Submission**

Paul Kudelsky

Southern New Hampshire University

CS-250-11741-M01 Software Development Lifecycle

Sathish Gopalakrishnan

06/25/2025

**7-1 Final Project Submission**

**Introduction**  
 Throughtout this course, I have had the opportunity to work as part of a Scrum team, and this team was part of the SNHU Travel Project for a company called ‘ChadaTech’. This project was meant as a practical (and theoretical) simulation of Agile software development practices and to help mark a transition from the Waterfall model to the more-used, more Scrum-based Agile methodology. As the Scrum Master of the project, I operated and planned the daily activities, the collaboration between the team members, and helped the team with their respective sprint goals and what they should expect and what the outcome *should* be. The retrospective is meant to provide both insight and a review of how the team utilized Agile, applied their Scrum roles, how they finished their user stories, managed any sudden interruptions in their work, and lastly how they used communication tools to help deliver a final product.  
**Applying Scrum Roles**  
 Being able to understand and more importantly, implement Scrum roles was the single most vital part to any of the project’s now-success. The Product Owner acted as the liaison for the client, collecting their requirements and translating them to the worker’s eyes, which was the user stories. These stories helped to prioritize the backlog of requests and tasks and ensured that the development team stayed aligned to their own and the client’s expectations of the project. As the Scrum Master, it was my responsibility to remove blockers, help facilitate meetings and be sure that the team maintained a proper path to completion. During this as well, I organized sprint planning, the daily scrum, and the retrospectives that made sure the team was flowing in all the same direction.

The developers and testers were the core of the team when it came to technical execution. The developers developed the user stories into something tangible; working. They made sure the features were functional while our testers guaranteed each iteration of features not only met the criteria but worked properly. The team dynamic between all the respective roles helped maintain the ability to be both flexible and efficient. Every single member was useful in some capacity.  
**Completing User Stories**  
 The user stories are the spine, the brain, of the backlog. Every story represented a unique user goal. As an example of this, “As a traveler, I want to view top destinations so I can choose a vacation.” A story like the above is prioritized and estimated the most during the sprint planning(s). The team ended up using ‘planning poker’ to help estimate a story’s points based on the story’s complexity and the effort behind it.  
 Following SDLC stages within Scrum: the planning took place during the sprint planning sessions, and these developed during sprint cycles. Testing was aligned with the ‘definition of done’ (DoD) review meetings. For example, when the client had requested a slide show for a top wellness vacation spot, these details of a story were added to our backlog, prioritized, and completed without further delay. It was decompressed into different tasks, developed, tested, and integrated into the sprint. Allowing these small details and requests to be had also helped maintain momentum without the quality of the product decreasing.

**Handling Interruptions and Change**

One of the many moments during this project that really defined the team as competent was when there was a sudden, imperative, and large client request that was for a slide show feature for wellness destinations. Even though the change came late in the sprint, it didn’t deter the team The Scrum allowed the team to pivot from one objective to another, without disruption, and for the team to have several meetings about how and when they need to complete these newfound tasks. Technical issues, difficulty and feasibility, and the sprint backlog were all discussed and tackled accordingly.  
 By being able to leverage Scrum’s interpretative/iterative design, the tasks that were in that project/sprint were able to be moved around and prioritized “correctly”. From a Waterfall model perspective, there would have been many different delays. With Scrum, there was adaptation to the new issues, and they didn’t end up derailing the project or the team’s timeline.

**Communication**  
 Clear and timely communication was a reason for the success of the team. Daily standups meetings were used to provide updates and address blockers, which prevented a larger and longer timeframe for backlogs. During these meetings, team members shared what they completed, what they were working on, and where they needed help. These quick updates ensure alignment and accountability.  
 Outside of those standup meetings, tools like Slack and Trello were kept and used for discussions, tasks, and progress. This digital communication was helpful in tracking work asynchronously. The team also held regular retrospectives to review what went well and what could be improved, which helped refine our process in each sprint.  
**Organizational Tools and Agile Practices**  
 The team used Trello to visualize the product backlog and sprint board. Columns were organized into “To Do,” “In Progress,” “In Review,” and “Done,” mirroring a Kanban workflow. This allowed all team members to quickly see task ownership and status.

Sprint planning and retrospectives were instrumental in managing efficiency and workload management. During planning sessions, we committed to a set, but realistic, number of story points based on team capacity. At the end of each sprint, retrospectives allowed us to reflect on our performance and implement improvements. These Agile moments enhanced our ability to iterate, learn, and grow as a team.  
**Evaluating the Agile Process**  
 Scrum proved to be an ideological masterpiece for the SNHU Travel project. It being iterative made it easier to incorporate client feedback, track progress, and maintain quality. Benefits included better teamwork, clearer accountability, and the ability to adjust to change.  
 However, Scrum also presented some challenges. Sprint planning sometimes resulted in overcommitting, and prioritizing work was not always unanimous. These issues were overcome through open communication and retrospectives. Compared to a Waterfall model, which relies heavily on up-front planning, Scrum gave us the flexibility needed for this type of evolving project.  
 In the end, the Scrum-Agile framework helped our team stay focused, engaged, and productive. We were able to deliver a working product that met client expectations, even with late-stage changes. This confirmed that Scrum is not only a viable option for modern software development, but often the best one for collaborative, client-focused projects.  
**Conclusion**  
 This project gave me first-hand experience applying Agile methodologies in real-world simulation. As a Scrum Master, I learned how to coordinate teams, manage changing requirements, and ensure continuous delivery. Our ability to collaborate, communicate, and adapt was what ultimately led to our project’s success. Looking ahead, I feel better prepared to lead or contribute to Agile development teams in a professional setting.