**Sprint Retrospective: Evaluating Agile for the SNHU Travel Project**

Jordan Richardson

CS-250

4/20/25

Communication issues, not technical ones, were the real cause of delays or misalignment. Throughout this course, I had the opportunity to experience a full development cycle from multiple Scrum perspectives. This gave me valuable insight into the benefits and challenges of applying Agile practices in a team setting and showed how critical each role is to overall project success. It also helped me connect Agile methodology to real-world outcomes in a way that was more hands-on than in a textbook explanation. Going through this project sprint by sprint made the framework feel more like a useful toolkit than a rigid set of steps.

Each Scrum role I experienced contributed to the outcome in a unique way. As the Product Owner, I focused on clearly defining user stories that reflected the business goals of SNHU Travel. One of the main challenges in this role was prioritizing features based on stakeholder input while avoiding scope creep. Creating a well-organized backlog allowed the development team to stay aligned with business goals and deliver incrementally with confidence (Cobb, 2015).

When I stepped into the Developer role, I was able to engage with technical requirements more deeply. I learned how critical it is to interpret user stories correctly and ensure the functionality delivers on the value promised. I found that breaking work into smaller tasks helped me stay focused and feel a sense of progress even within short time frames. Having been involved in early planning through refinement meetings helped avoid rework and helped me stay aligned with the big picture. It also reinforced the importance of cross-functional awareness as I knew my code was impacting other parts of the application which made me think more strategically.

As the Tester, I emphasized quality throughout the cycle. I created test cases based on user stories and offered early feedback during standups and sprint planning. Being integrated from the start and not just at the end helped us catch issues early and kept product aligned with user needs (Cobb, 2015). I began to understand testing not just as bug-catching but as a form of advocacy for the user experience. Good testers bring a different perspective to the team and ask questions that reveal assumptions or edge cases others might miss.

Acting as the Scrum Master allowed me to observe team dynamics from a wider lens. I facilitated standups and helped remove blockers, encouraging transparency and keeping the team accountable. As Scrum Master, I observed that miscommunications, such as unclear acceptance criteria for user stories, rather than technical roadblocks, often led to delay and misalignment. Consistent and well-facilitated standups foster a rhythm of shared awareness and accountability; for instance, a developer may voice a dependency on the design team which led to the scrum master removing blockers quickly. As someone who has worked on teams in other industries, I saw real value in how the Scrum Master promotes team autonomy without micromanaging.

The Agile approach made it easier to complete user stories because we had frequent touchpoints to discuss progress, ask clarifying questions and adjust when needed. In particular, our daily standups and backlog refinement sessions allowed the team to clarify acceptance criteria as we re-prioritize work based on current needs. This iterative feedback loop minimized wasted effort and better aligned development with stakeholder goals; for example, after stakeholders reviewed a demo of a the user story for booking flights, they suggested an additional filter which could be implemented in the next sprint.(Cobb,2015; TechTarget,2023). Even without a “real” client, the frequent check-ins created urgency and purpose around each task. Having clear acceptance criteria helped keep our definitions of “done” consistent, reducing rework.”

During the project, priorities shifted based on new stakeholder input which is quite common in real-world development. For example, when the focused changed to Eco Travel, we adjusted the backlog accordingly. Because Agile welcomes change, we adapted without losing momentum (Cobb, 2015). Rather than seeing changes as disruptions, we learned to treat them as refinements. Agile’s adaptability proved crucial as SNHU Travel’s needs evolved after reviewing initial iterations. The shift towards Eco-Travel required us to reprioritize the backlog and add new user stories focused on sustainable travel options, which we accommodated quickly in upcoming sprints.

Communication was central to everything we did. One example was reaching out to the Product Owner to clarify a vague story. Being specific in that request helped us define the acceptance criteria more clearly. These kinds of conversations promoted transparency and trust within the team (Cobb, 2015). We also had to think about how we provide feedback and its not just during retrospectives but also mid-sprint, which helps the team grow and learn faster. Communication wasn’t just for solving problems, it helped us prevent them too.

Another example was during sprint reviews. Open dialogue during sprint reviews enabled early stakeholder feedback, directly enhancing each iteration’s delivery. These reviews also gave us space to pause and recognize progress, which helps build morale and momentum. It’s easy to get caught in a delivery mindset, but taking a second to reflect with stakeholders often sparked new ideas or simplified what we thought were complex features.

Agile product-management tools like Jira proved invaluable for visualizing tasks, tracking sprint progress and optimizing workflow. For example, Jira’s burndown charts could provide a clear representation of sprint progress, allowing the team to identify and address potential scope creep issues in another sprint. These tools act as digital “information radiator,” increasing visibility and accountability (Cobb, 2015). They are especially useful during daily standups and retrospectives by keeping the team aligned and identifying blockers early. The visual aspect of a task board, even a virtual one, makes the work feel more tangible and keeps priorities in focus. I also found that being able to view progress in real time increases motivation.

The Scrum-Agile approach offered clear benefits for the SNHU Travel project: faster delivery of working features, easy adaptation to changing priorities, frequent stakeholder involvement and higher collaboration across roles. However, challenges included the need for consistent team availability and more frequent communication, which may not be practical in all organizations. Agile’s success hinges on a foundation of trust and autonomy. Teams unaccustomed to rapid decision making and iterative learning may initially find the approach challenging, requiring a shift in organizational culture. That said, Agile was the right for dynamic evolving projects like SNHU Travel (Cobb, 2015; Atlassian, n.d) It allowed for real time adaptation and continuous delivery of value. If ChadaTech wants to build more responsive and innovative teams, this approach would be a smart next step.

**References**

**Atlassian. (n.d.).** *What is agile?* Retrieved April 20, 2025, from <https://www.atlassian.com/agile>

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

**TechTarget. (2023).** *Agile software development.* Retrieved April 20, 2025, from <https://www.techtarget.com/searchsoftwarequality/definition/Agile-software-development>