# **CS 250 Module Seven Sprint Review and Retrospective**

Sarah Copeland

sarah.copeland@snhu.edu

Southern New Hampshire University

**Introduction**

The SNHU Travel project was used as a structured Agile simulation to practice the Scrum framework in a full software development life cycle. Although this project did not take place in a real business environment with a live development team, the role rotations and deliverables allowed me to understand how Agile supports iterative development, prioritization, and adaptability. This retrospective explains how each Scrum role contributed to the project, how user stories were completed, how Agile handled change, how communication supported progress, and how Scrum tools helped organize work. It also evaluates whether Scrum was the right approach for this project and provides lessons learned for future Agile work.

**Applying Roles**

Throughout this project I rotated through the roles of Product Owner, Developer, Tester, and Scrum Master. Each role showed me how Agile depends on collaboration and role clarity. When I served as Product Owner in the first sprint, I focused on building and refining the product backlog. I learned that well written user stories make development smoother. The Project Management Institute explains that Agile delivery depends on breaking down work into small, testable increments that can be prioritized by business value (PMI, 2021). I experienced this directly when I rewrote several user stories that were too broad and difficult to estimate. Adding acceptance criteria made it easier to understand what done meant for each story. When I moved into the Developer role, I shifted my focus from planning to execution. Working in small increments helped me complete features without losing track of time or scope. When I acted as the Tester, I verified whether each story met acceptance criteria before completion. I connected test steps to each story to make sure development stayed aligned with requirements. Finally, when I served as Scrum Master, I facilitated the structure of the sprint. Scrum Alliance teaches that the Scrum Master supports delivery by removing obstacles and keeping the process moving (Scrum Alliance, 2023). Even in a simulated environment, I saw how important that leadership role is in keeping the team focused on progress and improvement.

**Completing User Stories**

User stories were completed consistently because we focused on achievable goals per sprint rather than trying to deliver the full application at once. This follows a key Agile principle of delivering small increments of usable functionality. Angela Druckman explains that progress in Scrum is measured by completed product increments that provide value, not by time spent or tasks checked off (Druckman, 2012). That is exactly how this project worked. Each sprint produced deliverables that contributed to the travel application. The acceptance criteria defined what successful completion looked like and helped prevent partial delivery. User story completion also relied on maintaining a clear definition of done which included functionality, basic testing, and alignment with requirements. This structure made it possible to show meaningful progress over short cycles, which would not have been possible if we treated the project like a traditional waterfall plan.

**Handling Interruptions**

Scrum helped handle changes in direction and new requirements. In traditional waterfall development, change is considered a problem. In Agile it is expected. The client request to add a destination review feature is an example of changing requirements. Instead of reworking the entire plan, this new story was added to the product backlog and prioritized for a future sprint. The Agile Practice Guide explains that adaptive planning is a core Agile concept which allows a team to adjust scope without resetting the entire development process (PMI, 2021). This flexibility prevents delays and protects progress. Interruptions can also come from technical blockers. When a blocker occurred during development, it was addressed by reviewing priorities and resolving it before continuing. Scrum handled these interruptions by using time boxed sprints, backlog prioritization, and regular review cycles that support changing needs.

**Communication**

Communication in this project took place through written sprint updates, backlog revisions, user story clarification, and reflective journal entries rather than live team meetings. Even though this was an individual academic project, communication was still an important part of Agile practice. Progress was communicated by documenting work completed for each role. Priorities were communicated through product backlog organization and sprint goals. Testing outcomes were communicated through test case results and defect explanations. As Scrum Master I communicated sprint outcomes in the sprint review and improvement opportunities in the retrospective. Korn Ferry research states that Agile teams succeed when they operate with transparency and shared awareness (Korn Ferry, 2022). Communication in this project supported that by showing progress clearly through documentation, reflection, and alignment with sprint objectives.

**Organizational Tools**

Agile tools and Scrum ceremonies helped structure the work even in an academic setting. The product backlog organized user stories by priority. The sprint backlog identified what work would be completed in each sprint. The definition of done ensured consistent delivery standards. Sprint planning created a realistic plan. Sprint reviews provided a chance to reflect on progress and product value. Sprint retrospectives focused on improvement. These tools prevented confusion and kept the project moving. I also used the MoSCoW prioritization method to separate critical stories from lower priority ones. This helped prevent overcommitting to too much work in one sprint and supported decision making based on value. These tools support clarity and flow, which are central to Scrum practices (Scrum Alliance, 2023).

**Evaluating Agile Process**

The Scrum Agile approach was effective for this project because it aligned with the goals of iteration, learning, and improvement. Scrum made it easier to organize work and deliver results over time. It also encouraged reflection, which improved planning and execution with each sprint. There were challenges such as balancing sprint scope and being disciplined about priorities, but those challenges made the learning experience stronger. Waterfall would not have supported late requirement changes or continuous feedback. Scrum kept the focus on delivery and learning without slowing progress. Based on this experience, I believe Scrum is a strong methodology for software development and should be used for projects where requirements may evolve or customer input is important.

**Conclusion**

This project gave me practical experience applying Agile beyond theory. Working through each role made it clear that Scrum depends on communication, planning discipline, and iterative delivery. It also showed that Agile is not about moving fast without structure. It is about moving forward with purpose and transparency. The process helped build stronger habits around planning, prioritization, and team alignment. I would recommend that Chada Tech continue exploring Agile methods because they support continuous improvement and stronger product outcomes.

**References**

Druckman, A. (2012). The Scrum Field Guide: Practical Advice for Your First Year. Addison-Wesley.  
  
Korn Ferry. (2022). Leading Agile Teams. Korn Ferry Institute. https://www.kornferry.com  
  
Project Management Institute. (2021). Agile Practice Guide. PMI.  
  
Scrum Alliance. (2023). Scrum Foundations. https://www.scrumalliance.org