Sprint Review and Retrospective

Phillip Gooden

Southern New Hampshire University

CS 250: Software Development Cycle

Professor Morrison

October 23, 2025

Sprint Review and Retrospective

Over the semester, I worked through different Scrum roles and learned how each one contributes to a successful Agile project. As the developer, I learned how to turn changing requirements into functional code, especially when the Product Owner shifted our theme to wellness travel. Asking clarifying questions early helped me prevent rework and keep the sprint on track. As the tester, I focused on acceptance criteria and the Definition of Done, creating clear test cases that matched user stories. When I acted as Scrum Master, I realized how much communication drives collaboration. I scheduled standups, encouraged transparency, and used Agile tools to make progress visible to everyone. Each role relied on open dialogue, which connected to Cobb’s (2015) idea of transparency and information radiators supporting self-organizing teams.

Using Scrum made completing user stories manageable because I could link every task back to a shared sprint goal. During development, I broke features into small, testable parts and reviewed progress daily. This reflected Agile’s focus on continuous delivery and adapting to change. Story points and burn down charts helped me track work without overplanning. I learned from Cobb (2015) that Agile estimation values flexibility and incremental updates over rigid forecasting. Tracking our velocity made it easier to estimate what we could realistically complete each sprint. Constant feedback from the Product Owner kept my work aligned with the overall business value.

In one sprint, the Product Owner revised the project’s direction halfway through. Instead of stalling, my team and I paused to reassess the backlog and redefine “done.” Agile’s iterative structure made it easy to pivot without wasting effort. I saw that Agile teams expect change and use collaboration to manage uncertainty rather than resist it. Having a shared backlog and transparent goals allowed us to reprioritize user stories quickly while maintaining momentum.

Communication became the skill I developed the most this term. I used short, focused emails and standups to share updates, confirm details, and solve problems quickly. For example, I sent a progress email asking for updated acceptance criteria and testing expectations before continuing development. These small interactions improved teamwork and prevented confusion. Cobb (2015) notes that Agile depends on frequent, face to face or virtual communication to maintain alignment and reduce delays. Using Azure Boards also helped me keep communication open by showing everyone’s progress in real time.

I relied on Azure Boards and Kanban views to visualize progress, ownership, and blockers. These tools acted as digital information radiators and mirrored Cobb’s concept of transparent status tracking for distributed teams. Scrum events kept my workflow organized. Sprint Planning helped me set goals, Daily Scrums exposed blockers, Sprint Reviews gathered feedback, and Retrospectives helped me plan improvements for the next sprint. These tools and events gave my work rhythm and structure while leaving room for flexibility.

The Scrum Agile approach worked well for the SNHU Travel project. It improved transparency, adaptability, and team learning. It helped me respond to shifting requirements and deliver steady progress. Communication tools reduced confusion and improved accountability. At first, estimating time was challenging because our team’s velocity fluctuated, and some sprint goals were ambitious. Remote collaboration also required extra coordination. Still, Agile fit this project better than a waterfall model because the scope changed often, and feedback arrived mid sprint. Agile’s iterative nature supported learning and flexibility, matching Cobb’s idea that modern software projects benefit from adaptive management instead of rigid planning.

By the end of the semester, I saw how Scrum turns theory into practice. The clear roles, open communication, and short feedback loops helped me turn uncertainty into progress. I learned that Agile is more than a process—it’s a mindset that values teamwork, transparency, and adaptability.

# References

Cobb, C. G. (2015). The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach. Wiley.

Schwaber, K., & Sutherland, J. (2020). *The Scrum guide.* Scrum.org. https://scrumguides.org/