Sprint Review and Retrospective

Kenya Craw

CS 250 – Software Development Lifecycle

Instructor: Judy Mason

Southern New Hampshire University

October 18, 2025

Sprint Review and Retrospective

Over the course of the SNHU Travel project, I performed each significant role on a Scrum-Agile team; Scrum Master, Product Owner, Developer, and Tester; which deepened my understanding of how Agile practices enhance collaboration, adaptability, and value delivery. This Sprint Review and Retrospective summarizes how the Scrum framework supported project success, how communication and organizational tools contributed to alignment, and how the Agile mindset compares with more traditional development models.

Applying Roles

Each Scrum role brought unique value to our simulated development cycle. As Product Owner, I served as the link between stakeholders and developers, ensuring that user needs were accurately captured and prioritized. Following the INVEST criteria (MoldStud, 2024a), I translated feedback into user stories with clear acceptance criteria, allowing the team to deliver features that met travel-agency requirements.

In the Developer role, collaboration and clarification were critical. I often consulted the Product Owner to refine backlog details such as defining expected behaviors for a new “Flexible Date Search” feature and worked closely with the Tester to confirm edge cases before implementation. Rigby, Berez, and Elk (2020) emphasize that this continuous exchange between roles is central to Agile adaptability and accountability.

Serving as Tester underscored how quality assurance depends on communication. When acceptance criteria were vague, I documented questions and collaborated with the Product Owner to resolve ambiguities, a practice that reflects the continuous feedback loops advocated by Younas and Elnaffar (2022). Well-defined criteria transformed into efficient test cases that verified both functionality and user intent.

Finally, as Scrum Master, I guided the team through core Scrum events. Sprint Planning established sprint goals, Daily Scrums maintained synchronization, and Retrospectives promoted learning. I facilitated time-boxed sessions and used visual tools to ensure transparency, echoing Schwaber and Sutherland's (2020) guidance that a Scrum Master's chief responsibility is enabling self-organization and continuous improvement.

Together, these roles formed a balanced, cross-functional team capable of delivering incremental value while remaining responsive to change.

Completing User Stories

The Scrum-Agile approach enables iterative completion of user stories through structured ceremonies and constant feedback. For example, early versions of the "Set Vacation Preferences" story lacked clarity about default options. Open communication between the Tester and Product Owner clarified those gaps before development began, preventing rework. As Kaufmann and Schmidt (2020) found, frequent synchronization meetings improve understanding and reduce delays caused by miscommunication. Each sprint concluded with a potentially shippable increment, illustrating tangible progress toward project goals.

Handling Interruptions and Change

A key strength of Scrum is its capacity to accommodate shifting priorities. When stakeholders requested a mid-sprint addition, the "Flexible Date Search" capability, the team used backlog refinement to reprioritize and adjust sprint scope without derailing momentum. According to Cobb (2015), Agile teams thrive on such adaptability by welcoming change and leveraging short feedback loops to sustain delivery speed. In practice, our concise sprints limited wasted effort and kept development aligned with stakeholder needs, demonstrating the value of iterative planning over rigid sequencing.

Communication and Collaboration

Effective communication proved to be the foundation of project success. Consistent updates via discussion posts and status emails ensure shared awareness of progress and blockers. For instance, an example of one message I sent to the Scrum Master and Tester read:

"Alex – Could you please confirm that the stories we added for Sprint 3 ('Flexible Date Search' and 'Price Limit Filter') are reflected in the current sprint backlog? Also, I would like to review the sprint goal briefly during tomorrow's Daily Scrum to make sure the team is fully aligned.

Jordan – If you have any testing results or known issues from the last sprint, it would be helpful to review those now."

This concise, task-oriented style kept discussions clear and actionable. Transparent dialogue aligns with Agile's principle of valuing individuals and interactions over processes and tools (Schwaber & Sutherland, 2020). Moreover, maintaining open channels encouraged psychological safety, empowering team members to raise concerns early; an essential factor for successful Agile collaboration (Lankford, 2020).

Organizational Tools and Scrum Events

Our team relied on digital tools and Scrum ceremonies to coordinate work efficiently. Kanban boards and JIRA dashboards functioned as "information radiators" that displayed task status and ownership (Cobb, 2015). This visibility fostered accountability and made progress measurable, consistent with Atlassian's (2020) recommendation that transparency strengthens distributed teams.

Scrum events provided structure and rhythm:

- Sprint Planning clarified objectives.
- Daily Scrums aligned priorities.
- Sprint Reviews gathered stakeholder feedback.
- Retrospectives encouraged continuous improvement.

Regular retrospectives led us to small but valuable process tweaks, such as shortening stand-ups to ten minutes to keep meetings energetic and focused. These structured opportunities for reflection ensured that learning was continuous rather than confined to project closure.

Evaluating the Scrum-Agile Approach

The Scrum-Agile methodology delivered clear advantages for the SNHU Travel project:

Pros

- Continuous feedback improved alignment with stakeholder needs.
- Iterative planning enabled flexibility when requirements evolved.
- Defined roles enhanced accountability and teamwork.

Cons

- Frequent communication demands could challenge larger or remote teams.
- Early ambiguity in user stories sometimes caused initial delays.

Despite these minor drawbacks, Agile proved the optimal fit. SNHU Travel's evolving requirements and emphasis on user experience required adaptability that a Waterfall model, where stages are locked sequentially, could not easily provide. Agile's iterative cycles reduced risk, accelerated feedback, and ensured stakeholder involvement throughout development (Calefato et al., 2020).

Conclusion

This project demonstrated that Agile is not merely a process, but a mindset built on transparency, collaboration, and adaptability. Acting in every Scrum role revealed how interdependent communication and trust are to deliver value. By leveraging Agile tools, practicing open dialogue, and embracing iterative learning, teams can produce software that truly serves user needs. Going forward, I intend to foster environments where clear communication, inclusivity, and continuous improvement remain central to every sprint.

References

Atlassian. (2020, October 1). *Being transparent can ensure remote success — here's how to achieve it.* Atlassian Blog. https://www.atlassian.com/blog/distributed-work/transparency-is-essential-to-remote-success

Calefato, F., Giove, A., Losavio, M., & Lanubile, F. (2020). *A case study on tool support for collaboration in Agile development.* arXiv. https://arxiv.org/abs/2004.00289

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

Kaufmann, A., & Schmidt, R. (2020). Communication patterns in agile software development: A literature review. *Journal of Systems and Software, 166*, 110586. https://doi.org/10.1016/j.jss.2020.110586

Lankford, T. (2020, May 15). *If your Scrum is not fun, you are doing it wrong.* Medium. https://medium.com/serious-scrum/if-your-scrum-is-not-fun-you-might-be-doing-it-wrong-6c5fbce6bf48

MoldStud. (2024a, May 18). *How to create effective user stories for successful software projects.* https://moldstud.com/articles/p-how-to-create-effective-user-stories-for-software-projects

Rigby, D. K., Berez, S., & Elk, S. (2020). The Agile C-suite: A new approach to leadership for the team at the top. *Harvard Business Review, 98*(3), 64–73. https://hbr.org/2020/05/the-agile-c-suite

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

Younas, A., & Elnaffar, S. (2022). Improving software development agility using continuous feedback loops. *International Journal of Agile Systems and Management, 15*(1), 45–61. https://doi.org/10.1504/IJASM.2022.121045