Gigi Cruz

June 2025

CS 205 – Sprint Review and Retrospective

As the Scrum Master for ChadaTech’s pilot Agile project—the SNHU Travel application—I led a cross-functional team through a full sprint cycle to develop a client-facing web app. This engagement marked a strategic shift from ChadaTech’s traditional waterfall development practices to a more iterative, flexible Scrum-Agile framework. The primary goal was to determine whether Agile could improve responsiveness to change, team collaboration, and product quality. Throughout the project, I rotated through the roles of Product Owner, Tester, and Developer, applying Agile principles at each stage of the SDLC (Software Development Lifecycle). This retrospective offers a technical and reflective overview of the work completed, the impact of Agile on project outcomes, and the lessons learned.

The Agile framework encourages defined roles with focused responsibilities to promote autonomy and accountability (Schwaber & Sutherland, 2020). As Product Owner in the project’s initial sprint, I was responsible for defining and prioritizing the product backlog. The backlog included user-centric stories like: “As a user, I want to view the top 5 or 10 destinations based on my preferences,” “As a user, I want to personalize my profile to receive tailored recommendations, ” or “As a user, I want to filter travel packages by region, theme, or cost.” These stories were crafted to meet INVEST criteria—ensuring they were Independent, Negotiable, Valuable, Estimable, Small, and Testable (Wake, 2003). This helped set the foundation for modular, testable development work.

During the next sprint, I shifted into the Tester role. Here, I focused on refining acceptance criteria and drafting test cases for each user story. For instance, the filtering feature included test cases for sorting by lowest price, filtering by region, and validating empty search results. These cases enabled the development team to verify functionality met user needs with greater precision and reduced rework.

As a Developer in the final sprint, I collaborated on implementing a basic list view component and connecting it to sample destination data. However, midway through the sprint, the client shifted their strategic focus, requesting the site promote wellness and detox retreats instead of general travel options. This required us to pause, solicit clarification from the client, and refactor parts of the application to align with the new vision, which, due to its iterative nature, Agile made easy to handle.

One of the key advantages of the Scrum-Agile approach is its ability to guide teams toward delivering incremental value with each sprint. By working in short, focused iterations, our team could break down user stories into manageable tasks and continuously deliver shippable components.

For example, the “Top Destinations” feature began as a high-level user need. Through backlog refinement, we decomposed it into tasks such as setting up a destination schema, building the list view, and applying basic ranking logic. Each task was paired with a definition of done and a test case, ensuring clarity. Agile ceremonies, like Sprint Planning and Backlog Grooming, helped us align on scope and avoid gold-plating features that weren’t prioritized by the client (Cohn, 2010).

Thanks to this approach, we were able to push functional code at the end of each sprint, improving confidence in both the product and the process.

A significant test of our agility came during development when the client abruptly changed their focus from general travel to wellness-focused packages. In a waterfall model, such a pivot might have required full documentation updates, stakeholder reapprovals, and change control procedures, which could’ve caused delays.

Instead, we used the Agile framework to embrace this shift. During Sprint Planning, we held a discussion to clarify the new business goals, re-prioritized the backlog, and updated key user stories. Our team replaced generic destination categories with wellness themes and updated UI elements to reflect the new brand direction. Agile's built-in flexibility enabled us to handle this shift without derailing project velocity (Rigby, Sutherland, & Takeuchi, 2016).

Communication during the project was structured around Agile principles of transparency and collaboration. Scrum ceremonies, such as Daily Stand-Ups and Sprint Planning, helped surface blockers and align development efforts. Outside these scheduled events, direct communication proved essential when the client introduced a major scope change mid-sprint.

One effective example involved sending an email to the Product Owner, Christy, and the Scrum Tester, Brian, requesting clarity around the new wellness-focused direction. The message specifically asked for: A prioritized list of features or changes needed for the new wellness and detox theme, guidance on which elements from the existing travel functionality should be repurposed or deprecated, and updated test case scenarios that aligned with the newly defined user needs. This targeted outreach ensured that all parties were aligned before significant development work resumed. The email prompted a backlog refinement session that allowed the team to reorient efforts quickly and accurately, reducing guesswork and misalignment. Agile thrives on fast feedback and iterative improvement, and this example illustrated how direct communication outside of scheduled Scrum events could accelerate decision-making and enhance team efficiency (Rigby, Sutherland, & Takeuchi, 2016).

By ensuring that feedback loops remained short—even outside of formal meetings—project risk was minimized, and the final output remained consistent with evolving client goals.

The team used structured documentation to organize the product backlog, track progress, and plan each sprint. This included simple tools like shared documents and status updates submitted during each phase of the project. Tasks were clearly listed, prioritized, and updated throughout the sprint to reflect current progress.

Story points were used to estimate the effort for each task, which helped plan realistic sprint goals and maintain a steady development pace. Scrum ceremonies—such as Sprint Planning, Sprint Review, Daily Stand-Ups, and Retrospectives—provided regular check-in points to align priorities, address blockers, and review progress.

These structured events helped maintain focus, encourage collaboration, and support quick adjustments when priorities shifted. Even without complex software tools, the core Scrum principles—like iterative delivery, team accountability, and continuous feedback—ensured that development stayed on track and responsive to change (Schwaber & Sutherland, 2020).

The Scrum-Agile process demonstrated significant benefits throughout the SNHU Travel project. One of the clearest strengths was its ability to support rapid changes in direction. When the client shifted focus toward wellness and detox destinations, the Scrum framework provided the flexibility needed to reassess priorities, update user stories, and adapt development tasks without derailing progress. Regular ceremonies and short development cycles made it possible to incorporate new requirements incrementally, rather than starting over entirely, as would likely have been necessary in a waterfall approach.

Additionally, team communication and collaboration improved as a result of Scrum’s structured touchpoints. Frequent interactions, including stand-ups and retrospectives, encouraged transparency and alignment, while the emphasis on delivering small, functional increments helped maintain forward momentum and stakeholder engagement.

Despite these strengths, the Agile process presented challenges as well. Rotating through different roles each week—while beneficial for understanding the full development lifecycle—required frequent context switching and occasionally slowed momentum as responsibilities shifted. There was also a learning curve associated with adopting Agile concepts and practices, particularly in areas like backlog prioritization, estimating story points, and writing acceptance criteria.

Even with these challenges, the benefits of using Scrum far outweighed the drawbacks. The methodology supported a clear structure for communication, encouraged adaptability, and helped maintain consistent progress despite shifting goals. Compared to a traditional waterfall model, which is less suited for handling mid-project changes, Scrum provided a more resilient and responsive approach to development. Based on this experience, Scrum-Agile can be considered a highly effective choice for projects where client input evolves or where early feedback is critical to success.

The SNHU Travel project served as an effective pilot for Agile implementation within ChadaTech. Scrum practices—focused on user story prioritization, adaptive planning, and iterative development—enabled the delivery of a product aligned with changing client expectations. Clear role definitions, structured ceremonies, and the use of supporting tools contributed to the team’s ability to respond effectively and deliver high-value features. These results suggest that broader adoption of the Scrum-Agile framework could improve development outcomes across the organization.

References

Highsmith, J. (2013). *Adaptive Software Development: A Collaborative Approach to Managing*

*Complex Systems*. Dorset House Publishing.

Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). *Embracing Agile*. Harvard Business Review.

<https://hbr.org/2016/05/embracing-agile>

Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The*

*Rules of the Game*. Scrum.org. <https://scrumguides.org/>

VersionOne. (2019). *13th Annual State of Agile Report*. Digital.ai. <https://stateofagile.com>