Sprint Review and Retrospective: Evaluating the Scrum-Agile Pilot for SNHU Travel

Erica Kinch

Department of Computer Science, Southern New Hampshire University

CS250: Software Development Lifecycle

Joseph Rangitsch

February 22, 2026

Sprint Review and Retrospective: Evaluating the Scrum-Agile Pilot for SNHU Travel

ChadaTech initiated the SNHU Travel booking application as a pilot project to evaluate whether transitioning from a waterfall model to a Scrum-Agile framework would improve product delivery and organizational cohesion. As Scrum Master for this initiative, I facilitated the Sprint Review and Retrospective to evaluate team performance, assess the effectiveness of Scrum events, and determine whether the Scrum-Agile framework represents the appropriate methodology for broader organizational adoption.

Applying Roles Within the Scrum Team

The success of the SNHU Travel project depended on clearly defined Scrum roles working collaboratively within the framework.

The Product Owner prioritized features based on client value. For SNHU Travel, this meant delivering core booking and search functionality before implementing cosmetic enhancements. Focus group insights were translated into prioritized user stories centered on personalization, destination filtering, and price-based sorting. During backlog refinement, stories were re-ordered when client feedback emphasized the importance of filtering by price and rating for launch readiness. This ensured development consistently aligned with business priorities, and this alignment illustrates how Scrum-Agile links backlog prioritization directly to stakeholder value, reinforcing the framework’s emphasis on delivering the highest-value features first.

During Sprint Planning, developers evaluated story complexity and adjusted effort estimates when technical uncertainty emerged, particularly around dynamic filtering and thirdparty API integration. Addressing integration concerns before sprint commitment prevented overcommitment and supported predictable delivery.

The Tester translated user stories into detailed test cases and identified ambiguity in acceptance criteria. When the booking confirmation workflow lacked clarity regarding email notifications and sorting behavior, clarification was requested before development proceeded. This reduced rework and strengthened quality assurance.

As Scrum Master, I facilitated Sprint Planning, Daily Scrums, Sprint Reviews, and Retrospectives. I removed impediments and reinforced the Definition of Done. When integration blockers arose, I coordinated technical discussion to resolve the issue within the sprint timeframe. By maintaining focus on sprint goals and timeboxes, I ensured delivery remained predictable while supporting team transparency. This demonstrates how the Scrum Master role safeguards both adaptability and discipline, ensuring flexibility does not compromise delivery stability.

Each role operated within clearly defined responsibilities while collaborating cross-functionally. This balance of accountability and teamwork directly contributed to incremental delivery success.

Completing User Stories Through the Agile SDLC

The Scrum framework enabled incremental completion of user stories through iterative refinement and validation.

Rather than locking requirements at the outset, the team refined user stories during backlog grooming sessions. For example, filtering logic evolved to include price, rating, and availability after stakeholder clarification. This iterative refinement reduced ambiguity before sprint execution. By clarifying requirements before development began, the team minimized rework risk and reinforced the preventive quality controls embedded in Agile practices.

During Sprint Planning, only stories that met the Definition of Ready were included in the Sprint Backlog. Daily Scrums provided structured visibility into progress and impediments, enabling early intervention when blockers arose.

User stories were considered complete only when they satisfied the Definition of Done, which required functional implementation and validated test cases. Unlike waterfall development, where testing occurs after implementation as a separate phase, Scrum embedded testing within each sprint cycle. This approach reduced late-stage defect discovery and increased stakeholder confidence in delivered increments.

The iterative SDLC structure improved transparency and ensured that working software was delivered at the end of each sprint.

Handling Interruptions and Changing Direction

Midway through development, stakeholders introduced new requirements related to promotional discount integration and adjustments to destination ranking logic. Under a traditional waterfall approach, such changes likely would have required revisiting formal documentation, redesigning solution artifacts, and renegotiating timelines.

However, within Scrum, the Product Backlog was reprioritized during Sprint Planning. Lower-priority cosmetic enhancements were deferred, and promotional functionality was elevated in priority. Because sprint timeframes remained fixed while scope remained flexible, the team maintained delivery cadence without destabilizing progress. This experience supports the broader assessment that Scrum-Agile balances adaptability with schedule predictability, a critical factor in client-facing environments.

This experience demonstrated Scrum's capacity to respond to change while preserving schedule integrity, which is an essential advantage in dynamic client environments.

Communication and Collaboration

Structured communication practices played a critical role in maintaining alignment during the SNHU Travel pilot.

One clear example occurred during test case development, when the Tester identified ambiguity in several user stories and sent clarification questions to the Product Owner. These questions addressed issues such as whether top destinations should appear on one page or as a slideshow, whether results should be sortable by rating, and whether price filters should allow user-defined ranges. This proactive communication prevented incorrect assumptions from being embedded into development and ensured that acceptance criteria were measurable before sprint execution.

Additionally, backlog refinement sessions became more structured after recognizing that some user stories required decomposition. Identifying that a mobile application request functioned more as an epic than a sprint-sized story improved future sprint planning accuracy. By clarifying scope prior to commitment, the team reduced the risk of oversized sprint goals.

Daily Scrums further reinforced transparency by creating a consistent forum for surfacing blockers and coordinating next steps. Rather than functioning as status reports, these events supported inspection and adaptation within the sprint timebox.

Finally, Sprint Reviews enabled stakeholder feedback on implemented filtering and personalization features. This inspection point ensured that adjustments could be incorporated into subsequent backlog prioritization rather than discovered late in development.

Taken together, these communication practices strengthened collaboration, minimized ambiguity, and reinforced Scrum's core principles of transparency, inspection, and adaptation.

Organizational Tools and Scrum Events

Several Scrum artifacts and events were instrumental to project performance. The Product Backlog ensured value-based prioritization, while the Sprint Backlog protected sprint focus. The Definition of Done maintained consistent quality standards, and Sprint Planning aligned team capacity with realistic commitments. Daily Scrums surfaced impediments early, Sprint Reviews validated stakeholder expectations, and Sprint Retrospectives supported continuous improvement. These recurring inspection points contrasted with the linear progression of the waterfall model and significantly reduced late-stage surprises. By embedding inspection and adaptation into each sprint, Scrum reduced cumulative risk that often surfaces late in traditional phase-based models.

Evaluating the Effectiveness of the Scrum-Agile Approach

The Scrum-Agile methodology presented both strengths and operational challenges during the SNHU Travel pilot. Overall, the framework improved adaptability and transparency;

however, its effectiveness depended heavily on disciplined execution and stakeholder engagement.

Advantages Observed

Flexibility in accommodating evolving requirements.

During development, stakeholders introduced changes related to promotional discount integration and destination ranking logic. Because Scrum fixed time while allowing scope to adjust, the team was able to reprioritize backlog items without destabilizing the sprint schedule. This adaptability would have been significantly more disruptive under a waterfall model.

Continuous stakeholder visibility and feedback.

Sprint Reviews provided regular inspection points where stakeholders evaluated working increments. This visibility reduced uncertainty and ensured that development aligned with business expectations before proceeding to additional features. In doing so, Scrum reinforced accountability through frequent inspection, reducing the likelihood of misalignment at later stages.

Early identification of ambiguity.

The Tester's clarification of acceptance criteria during backlog refinement prevented misinterpretation of filtering logic and booking confirmation requirements. By addressing ambiguity prior to sprint commitment, the team reduced rework and improved delivery quality.

Strong cross-functional collaboration.

Daily Scrums and structured refinement sessions fostered shared accountability. Developers, testers, and the Product Owner collaborated rather than operating in isolated phases, strengthening problem-solving and accelerating decision-making.This collaborative structure highlights one of Scrum's primary strengths: dissolving functional silos to enable faster, more integrated delivery.

Limitations Observed

Requires disciplined backlog management.

The effectiveness of Scrum depended on maintaining a well-refined and prioritized Product Backlog. Without structured refinement, user stories risked becoming oversized or ambiguous, potentially undermining sprint predictability.

Dependent on active stakeholder participation.

Sprint Reviews and backlog refinement sessions required timely stakeholder input. If feedback had been delayed, the iterative advantage of Scrum would have diminished. This dependency illustrates that Agile effectiveness is contingent upon sustained engagement, making stakeholder commitment a critical success factor.

Cultural adjustment for teams transitioning from waterfall.

Moving from phase-based delivery to iterative increments required a mindset shift. Team members accustomed to comprehensive upfront documentation needed to adapt to evolving requirements and continuous inspection.

Overall Assessment

For the SNHU Travel pilot, Scrum-Agile proved highly effective because the project involved evolving customization features and dynamic stakeholder input. The benefits of flexibility, transparency, and embedded validation outweighed the coordination demands required by the framework.

While waterfall may remain suitable for highly stable or regulated environments, this project demonstrated that Scrum-Agile enhances responsiveness, transparency, and delivery quality in innovation-driven initiatives.