CS 250 Final Project  
Sprint Review & Retrospective  
SNHU Travel – Scrum Adoption Pilot

Rocky Blasier

8/17/25

**Applying Roles**

As Scrum Master during the SNHU Travel pilot, I coordinated events and removed impediments while keeping the team focused on the Sprint Goal. The Product Owner (PO) refined the backlog to maximize value—e.g., prioritizing the “Search & Filter trips” epic so customers could filter by budget, dates, and destination. The Developers (full-stack and QA) collaborated on a vertical slice for trip search: API integration with a third‑party travel service, a React search UI, and automated tests. For example, when rate‑limiting from the third‑party API blocked testing, I facilitated a quick spike and introduced response caching to stabilize builds. These role-specific contributions align with Scrum’s emphasis on clear accountabilities (Schwaber & Sutherland, 2020).

**Completing User Stories**

Using Scrum’s iterative SDLC flow—refinement, sprint planning, daily scrums, development, testing, and review—helped us break work into INVEST user stories (independent, negotiable, valuable, estimable, small, testable). Two examples reached “Done” with acceptance tests passing: (1) “As a traveler, I can search trips by destination and date so I can quickly see options,” and (2) “As a traveler, I can sort results by price or rating.” Unit testing, API contract testing, UI accessibility testing, and re-documentation were all mandated by the Definition of Done (DoD). The Sprint Review provided stakeholders with immediate feedback that immediately influenced the next Sprint Backlog, and time-boxed sprints established a steady pace (Schwaber & Sutherland, 2020; Atlassian, n.d.).

**Handling Interruptions**

Mid‑sprint, the travel API changed pagination parameters, breaking our list view. In response, we negotiated scope with the PO during a replanning huddle, preserved the Sprint Goal (a usable search MVP), and split the affected story. One slice delivered stable pagination: the other deferred advanced sorting. Scrum's empirical management of transparency, inspection, and adaptation allowed us to alter plans without causing the increment to go awry. This responsiveness also aligns with industry studies that reveal agile practice improves time-to-market and teamwork, especially in small to medium-sized teams (Digital.ai, 2024).

**Communication**

To keep alignment high, I posted concise daily notes on our team channel (What changed, what’s blocked, what’s next) and facilitated asynchronous reviews with short demo clips. Example message: “Yesterday: merged search form validation; Today: wire error states to API; Blocker: sandbox rate limits—requesting higher quota.” These artifacts improved visibility and reduced meeting load. PMI’s research highlights that communication and other “power skills” correlate with organizational agility and project success (PMI, 2023).

Organizational Tools

We used a lightweight toolset: Jira for the Product Backlog and Sprint Backlog, GitHub for PR-based code review, and GitHub Actions for CI. Scrum events amplified tool effectiveness: Sprint Planning clarified the Sprint Goal and slice boundaries, which reduced WIP and hand‑offs. • Daily Scrum surfaced integration risks (API limits), enabling same‑day mitigation. • Sprint Review captured stakeholder feedback (sorting by rating was more valuable than by duration). • Sprint Retrospective yielded two concrete improvements: adopt contract tests for the API and add a mock server to decouple UI testing.

**Evaluating the Agile Process**

Pros: Faster feedback loops; early delivery of a working search MVP; better stakeholder alignment; improved quality via DoD and automated tests; team ownership and transparency. Cons: Context switching due to mid‑sprint changes; initial learning curve for Scrum events and story slicing; dependency risk on external APIs. On balance, Scrum was a strong fit for the SNHU Travel application because requirements evolved with stakeholder discovery and the architecture supported vertical slicing. According to research contrasting agile and structured (waterfall) approaches, agile is preferred when change is anticipated and should be chosen based on project volatility and the need for early value (Sharma et al., 2023). A waterfall or hybrid approach might be better for very stable, compliance-heavy projects with fixed scope and interfaces.

**Conclusion**

The Scrum‑Agile approach enabled our team to deliver a valuable, testable increment quickly and to adapt when requirements shifted. The pilot demonstrated improved collaboration, predictable cadence, and measurable quality gates. With lightweight governance and clear accountabilities, scaling Scrum to additional Chada Tech teams is likely to produce similar benefits, provided teams invest in coaching, Definition of Done, and a robust CI/CD backbone.

**References**

Atlassian, n.d. a thorough examination of Scrum team roles. https://www.atlassian.com/agile/scrum/roles Digital.ai (2024). The 17th Agile State Report. Project Management Institute, https://digital.ai/resource-center/analyst-reports/state-of-agile-report/ (2023). Pulse of the Profession 2023: Redefining project success with power skills. https://www.pmi.org/learning/thought-leadership/power-skills-redefining-project-success

Sutherland, J., and Schwaber, K. (2020). The Scrum Handbook. Sharma, S., Bansal, A., & Sharma, A. (2023) https://scrumguides.org/. Agile versus structured software development. CSI ICT Transactions. https://doi.org/10.1007/s13198-023-01958-5