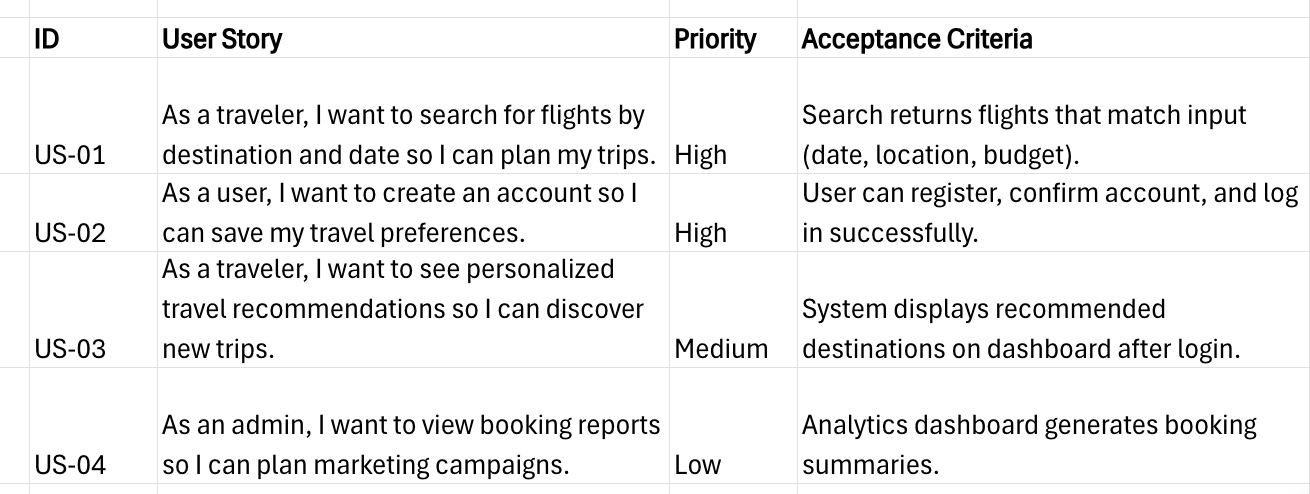
# **Final Project:** SNHU Travel

Stephane Nidri:

P**art 1: Team Artifact:**

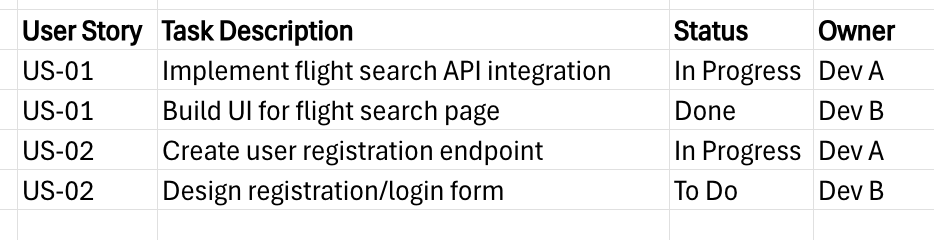
1. Product Owner Artifact: Product Backlog

**Role:** Product Owner  
 **Artifact:** SNHU Travel Product Backlog



### **2. Developer Artifact: Sprint Task Board (Sprint 1 Example)**

**Role:** Developer  
 **Artifact:** Sprint Task Board



### **3. Tester Artifact: Test Plan (Sprint 1 Excerpt)**

**Role:** Tester  
 **Artifact:** Test Plan

**US-01: Flight Search**

* Verify search results match entered departure/arrival dates.
* Verify invalid date input returns an error.
* Verify filtering by price works.

**US-02: Account Creation**

* Verify account creation with valid email/password.
* Verify error messages for invalid fields.
* Verify successful login with valid credentials.
* Verify failed login with incorrect credentials.

### **4. Scrum Master Artifact: Sprint Burndown & Event Notes**

**Role:** Scrum Master  
 **Artifact:** Sprint Notes

* **Sprint Goal:** Deliver basic search and user account features.
* **Burndown:** Story points reduced steadily, with a spike mid-sprint due to bug fixes.
* **Daily Standups:** Identified API documentation delays early.
* **Sprint Review:** Demoed working search and partial account functionality.
* **Retrospective:** Agreed to increase test automation to reduce manual bug retesting.

**Part 2: Sprint Review & Retrospective**

Introduction:

ChadaTech launched a pilot initiative to evaluate the Scrum-Agile methodology by developing a travel booking application for SNHU Travel. The project’s objective was to determine whether Agile could deliver incremental value, enhance collaboration, and adapt to change more effectively than the company’s traditional waterfall model. This Sprint Review and Retrospective highlights the contributions of each Scrum role, the management of user stories and project interruptions, communication practices, organizational tools, and an overall assessment of the Agile process applied to this pilot.

## **1. Contributions of Scrum Roles and Artifacts**

The project’s success depended on the unique contributions and artifacts from each Scrum role:

* **Product Owner:** Managed and prioritized the Product Backlog, ensuring user stories reflected SNHU’s business goals. For example, US-01 (flight search) was marked as a top priority, guiding the team to deliver high-value functionality first.
* **Developer:** Maintained the Sprint Task Board, translating stories into smaller actionable items. This artifact provided transparency: for instance, showing that the flight search UI was completed while API integration was still in progress.
* **Tester:** Produced a Test Plan to validate features against acceptance criteria. Early in Sprint 1, login validation issues were detected and resolved before the Sprint Review, demonstrating proactive quality assurance.
* **Scrum Master:** Facilitated ceremonies and tracked progress through the Sprint Burndown chart. This allowed quick escalation of obstacles, such as delays in the API integration, and promoted collaborative resolution.

Together, these contributions established accountability, visibility, and alignment with the Sprint Goal.

## **2. Completing User Stories**

Scrum enabled the incremental completion of user stories, guided by clearly defined “Definition of Done” criteria.

For example, US-01 (flight search) involved both UI design and API integration. Once developed, testers validated filters for dates and pricing. Because Scrum emphasizes delivering potentially shippable increments, stakeholders could interact with the flight search feature in the first Sprint: long before final project delivery.

In contrast, under waterfall, this functionality would likely have been demonstrated only after months of sequential work, increasing the risk of misalignment with client expectations.

## **3. Handling Interruptions and Changes**

Midway through development, the external flight API provider updated its documentation, disrupting ongoing integration work. In a waterfall model, this change would have required a lengthy change request process and delayed multiple phases.

With Scrum, the Product Owner quickly re-prioritized the backlog, the Developer adjusted integration tasks, and the Tester revised the associated test cases. The team adapted within the sprint, showing Agile’s ability to manage change without losing momentum.

## **4. Communication and Collaboration**

Communication was structured, frequent, and highly effective:

* **Daily Standups:** Team members shared progress and identified blockers. For instance, “API integration cannot continue until new documentation is released.” This allowed immediate escalation and resolution.
* **Sprint Review:** Stakeholders provided real-time feedback on the search functionality, confirming alignment with customer needs.
* **Retrospective:** Team members openly suggested improvements, such as expanding automated testing.

These communication practices were effective because they were concise, role-focused, and encouraged collaboration, fostering transparency and shared ownership.

## **5. Organizational Tools and Scrum Events**

Key artifacts and tools ensured productivity and alignment:

* **Product Backlog (PO):** Ensured development followed business priorities.
* **Task Board (Dev):** Made task progress visible and manageable.
* **Test Plan (Tester):** Guaranteed quality and alignment with acceptance criteria.
* **Burndown Chart (SM):** Tracked progress and sprint velocity.

Scrum ceremonies amplified their usefulness: Sprint Planning aligned efforts with goals, Daily Standups eliminated blockers, Sprint Reviews engaged stakeholders early, and Retrospectives drove continuous improvement.

## **6. Evaluating the Agile Process**

**Pros:**

* Early and continuous delivery of working features.
* Strong adaptability to external changes (for instance API update).
* Frequent stakeholder involvement and feedback.
* Transparent workflow supported by artifacts and ceremonies.

**Cons:**

* Initial challenges with estimating story points.
* Daily meetings occasionally felt repetitive.
* External dependencies, such as the API provider, sometimes slowed progress.

**Assessment:**  
 Scrum-Agile was clearly the most effective methodology for this project. It enabled early client engagement, reduced rework, and allowed rapid adaptation to external changes. A waterfall approach would have delayed feedback until the late stages, increasing both risk and the cost of revisions.

## **Conclusion**

The SNHU Travel pilot demonstrated that Scrum-Agile significantly improves delivery speed, adaptability, and collaboration compared to waterfall. Each role’s artifacts enhanced visibility, accountability, and alignment with project goals. Although some challenges arose, the overall outcomes strongly support adopting Scrum-Agile more broadly across ChadaTech’s development teams.

Refereces:

Class resource: [*The Project Manager’s Guide to Mastering Agile*](https://search.ebscohost.com/login.aspx?direct=true&custid=shapiro&authtype=athens,ip&db=nlebk&AN=937009&site=ehost-live&scope=site)