### **1. Introduction**

We built the SNHU Travel web app over a four-week Scrum sprint. This review summarizes how our roles, ceremonies, and tools helped us deliver features, handle changes, and learn Scrum’s strengths and limits.

### **2. Applying Roles**

* **Product Owner:** Gathered requirements, wrote user stories (for example, filter by price), and reviewed demos.
* **Scrum Master:** Ran planning, stand‑ups, review, and retrospective; removed blockers such as code merge conflicts; tracked burndown.
* **Developers:** Turned stories into working code (for example, built a “Top‑5 Destinations” slideshow); broke work into small, shippable tasks.
* **Testers:** Wrote test cases, ran QA, logged bugs in JIRA; caught issues such as currency conversion early so fixes happened within hours.

### **3. Completing User Stories**

We used Planning Poker to size stories, committed only what matched our two‑week capacity, and held daily syncs. When an API formatting issue arose, we re‑estimated and re‑assigned immediately, keeping us on schedule so every story shipped as planned.

### **4. Handling Interruptions**

Mid‑sprint, the client asked for a new “Wellness Retreats” slideshow. We added it to JIRA, reprioritized lower‑value work, and adjusted our sprint plan in one backlog refinement. The change was live within a day and did not derail our goal.

### **5. Communication**

Daily 10‑minute stand‑ups surfaced blockers early, once a tester spotted an integration bug and we fixed it the same morning. We also used Slack threads per story to keep discussions organized and searchable, so questions and answers stayed tied to the right work item.

### **6. Organizational Tools and Ceremonies**

JIRA tracked every story’s status (To Do → Done), made progress visible, and supported quick re‑planning. A virtual Kanban board reinforced status updates during stand‑ups. We held sprint reviews to demo features and retrospectives to note improvements, creating a steady rhythm of delivery and adjustment.

### **7. Evaluating Scrum**

* **Pros:** Flexible to change, transparent progress, frequent demos build confidence.
* **Cons:** Ceremonies can feel heavy in small teams, risk of scope creep without tight backlog control.
* **Verdict:** Scrum was the right choice; its time‑boxed sprints and clear roles let us respond to evolving needs and deliver quality faster than a Waterfall approach would have.

### **8. Conclusion**

Scrum enabled close collaboration, quick pivots, and reliable delivery. Key lessons: define roles clearly, limit sprint scope to real capacity, and pair daily check‑ins with a strong tool such as JIRA. For future small teams, shorten ceremonies to keep focus on shipping value.