The SNHU Travel project provided a hands-on simulation of Agile teamwork, allowing me to rotate through Scrum roles and experience how Agile supports collaboration, planning, and delivery. This Sprint Review and Retrospective reflects on how each role helped the team succeed, how user stories were completed within sprints, how the team adapted when scope shifted, and how communication and organizational tools helped maintain progress. It also evaluates the effectiveness of Scrum-Agile compared to traditional Waterfall methods. Throughout this process, the structure of Agile and its supporting tools and ceremonies helped ensure the delivery of a usable product through cross-role collaboration, transparency, and iteration.

Each role I performed served a distinct purpose in keeping the team on track. As Scrum Master, I focused on clearing blockers and guiding the team through its ceremonies. I facilitated Daily Standups and ensured that updates were communicated clearly. In the Developer role, I worked on tasks pulled from the Product Backlog and stayed responsive to changing priorities. When acting as Product Owner, I managed the backlog and prioritized user stories that reflected the client’s goals. As Tester, I defined acceptance criteria and worked with Developers throughout the sprint to identify bugs early. According to Cobb (2015), “A Scrum team is a self-organizing and cross-functional team that has all the competencies to accomplish the work without depending on others not part of the team” (p. 47). The roles were structured to prevent bottlenecks and keep the work balanced across the team, and I experienced that clearly in each sprint. Having this clear division also reduced stress since I always knew which part of the process I was responsible for and when to ask for support.

Scrum helped user stories get completed because of its focus on iterative planning and collaboration. Each sprint started with Planning, where the team selected a small group of high-priority stories. As a Tester, I helped ensure those stories had clear criteria from the start. Developers could focus on delivery while testers ran checks in parallel. The end of each sprint included a Review to present the work and a Retrospective to evaluate what could be improved. This structure kept feedback moving quickly. Boogaard (2019) wrote, “Shortening your feedback loops makes your team more effective and reduces stress.” That described our experience exactly. We stayed productive because we didn’t wait until the end to make corrections. With every completed story, the team gained confidence in their progress and felt more in control of what was coming next.

Scrum also supported the team when priorities changed. During one sprint, the requirements for SNHU Travel shifted and several stories had to be replaced mid-sprint. The Product Owner updated the backlog, and the team adapted quickly. Developers changed direction, and testers adjusted their work. In a Waterfall model, that change would have delayed everything. Agile made it possible to continue delivering usable software even with new scope. As Valacich and George (2020) wrote, “Agile methods are best suited for projects with rapidly changing or highly uncertain requirements” (p. 84). This was one of those situations, and Agile made it manageable. We didn’t need to pause the entire sprint or wait for approvals. We just shifted and kept working.

Communication was constant throughout the project and was central to every sprint. In the Daily Standups, team members identified what was done, what was next, and what was blocked. This made it easier to resolve issues early and kept the team aligned. Everyone had a clear role, and no one was waiting for a handoff. As Tester, I used that time to ask questions about unclear requirements. As Product Owner, I explained shifts in priority. Ambler (2020) defines an information radiator as “a physical display of information that is visible to everyone and updated regularly.” While we did not use actual tools like Jira, we followed this principle by documenting our updates and communicating openly during ceremonies. That made a big difference in how smoothly each sprint progressed. It also helped build trust between roles since everyone could see that work was happening even if they weren’t directly involved in every task.

Even without live Agile tools, we followed the logic of Agile workflows. The team simulated a Scrum Board using shared updates and structured check-ins. The backlog acted as our single source of truth. Cobb (2015) described these tools as “a collection of critical project information that the team updates continuously and displays prominently” (p. 139). Our work followed that idea closely. We knew what was in progress, what was complete, and what needed attention. Atlassian (n.d.) highlights that Scrum Boards help teams “visualize work, limit work-in-progress, and maximize flow.” Even though our version was manual, we applied the same method. It showed me how even simple tools can be effective as long as everyone commits to using them consistently.

Overall, the Scrum-Agile approach was the right fit for this project. There were challenges, especially when priorities shifted and required immediate response. But the structure of Agile made those shifts easier to handle. Hamilton (2014) explained that “Continuous improvement is about removing the things that get in the way of your work.” Each Retrospective helped the team do exactly that by talking through what slowed us down and deciding how to improve in the next sprint. Agile kept us focused, open to feedback, and able to deliver value incrementally. Waterfall would not have supported that level of flexibility, and I believe Agile led to a better process and a better product. The experience also gave me insight into how Agile can be taught, practiced, and expanded beyond a single project. I can now see why companies are shifting in that direction.

**References**

Ambler, S. W. (2020). The Agile modeling (AM) home page. http://agilemodeling.com/essays/informationRadiators.htm

Atlassian. (n.d.). Jira software guide.<https://www.atlassian.com/software/jira/guides/getting-started/introduction>

Boogaard, K. (2019, November 7). How to shorten your feedback loops (and reduce your team’s stress). *DZone*.<https://dzone.com/articles/how-to-shorten-your-feedback-loops-and-reduce-your>

Cobb, C. G. (2015). *The project manager’s guide to mastering Agile: Principles and practices for an adaptive approach*. Wiley.

Hamilton, B. (2014, September 24). Top 20 quotes for Lean business inspiration. *LeadingEdge Group*.<https://www.leadingedgegroup.com/top-20-quotes-lean-business-inspiration/>

Valacich, J. S., & George, J. F. (2020). *Modern systems analysis and design* (9th ed.). Pearson.