As part of my coursework with SNHU, I participated in a simulated Agile development project that reflected a real-world scenario at ChadaTech—a company exploring the transition from the traditional Waterfall model to Agile using the Scrum framework. In this simulation, I developed a travel application for a fictional client, SNHU Travel, and took on various Scrum roles throughout the software development life cycle (SDLC). This exercise allowed me to gain hands-on experience applying Agile principles, explore the unique responsibilities of Scrum team members, and evaluate the effectiveness of this approach in delivering a functional and adaptable software product.

During the simulation, I rotated between the roles of Product Owner, Developer, and Scrum Master. Each role provided a different view of the development process and helped me understand how collaboration and flexibility are key to Agile success. As the Product Owner, I defined user stories that translated business needs into clear development goals. For example, I prioritized a story that enabled users to browse travel destinations, which directly supported the app’s usability. As a Developer, I implemented features like updating images within the JAR file to ensure the travel packages displayed the most current visuals. And as the Scrum Master, I guided our process, facilitated Scrum events, and helped resolve hypothetical roadblocks to keep the project moving smoothly.

The use of user stories in this simulation was central to organizing our work. Instead of front-loading all the requirements as in a traditional Waterfall model, we created a backlog of smaller, customer-focused goals. Each story was refined and prioritized, then completed incrementally through Sprints. One example was: “As a user, I want to browse available travel destinations so I can plan my vacation.” This story went through refinement, estimation, and was implemented and tested within a Sprint. Agile allowed for flexibility and adaptability, helping me focus on delivering value continuously rather than waiting until the end for a complete product.

One of the most valuable aspects of this project was how easily the Agile process handled change. Midway through development, a simulated client request added a new feature—user reviews. In a Waterfall environment, this change could have required a major revision of the project scope, timelines, and documentation. Using Scrum, however, we simply created a new user story, added it to the backlog, and prioritized it during Sprint Planning. This responsiveness is one of the key strengths of Agile development—it turns potential disruptions into opportunities to deliver more value.

Communication throughout the project was managed using GitHub, which served as the primary platform for collaboration, task tracking, and updates. GitHub provided an integrated environment for managing issues, pull requests, and ongoing changes to the project. Although this was a simulated environment, GitHub helped simulate real-world collaboration, allowing me to track progress on user stories, submit code updates, and review feedback from peers.

The use of GitHub facilitated transparent and consistent communication, ensuring that updates, tasks, and changes were all documented in one central location. The "issues" feature allowed me to post specific questions, provide status updates, and assign tasks to the appropriate team members. This encouraged prompt feedback and decision-making, preventing delays caused by unclear requirements or assumptions.

Even though we weren’t working in real time, the consistent communication on GitHub helped keep everyone aligned throughout each phase of the Sprint, ensuring the project moved forward efficiently.

GitHub was central to organizing and managing the project, serving as a repository for code, feedback, and collaboration. I used it to store the work I completed, submit code updates, and track issues that arose during development. GitHub allowed me to maintain an organized record of progress while keeping everything in one place for easy access and review.

Reflecting on the experience, I found that the Scrum-Agile approach was highly effective for this type of project. Its biggest strengths included flexibility, responsiveness to change, and continuous feedback that helped deliver features aligned with user needs. That said, there were challenges—especially early on, as I adjusted to the cadence of Sprints and ceremonies. Some user stories also lacked detail at first, requiring extra effort during Planning to clarify expectations and ensure proper implementation.

Ultimately, I believe Agile—particularly Scrum—was the ideal methodology for this project. Its iterative, feedback-driven structure allowed for better alignment with evolving project goals and provided a more dynamic and engaging development process than traditional models. In a real-world context, this approach would likely lead to a stronger final product and better stakeholder satisfaction. If ChadaTech were to implement Scrum organization-wide, the experience I gained in this simulation supports the idea that Agile can drive both innovation and team cohesion across projects.