The SNHU Travel project was the first ChadaTech project to use Scrum-Agile after years of using Waterfall. As Scrum Master, I ran planning, daily meetings, reviews, and retrospectives. My job was to keep the work on track and close to what the customer wanted. This reflection looks at the roles, user stories, problems, tools, and communication that helped us finish the project. In the end, Agile-Scrum worked best for this kind of work.

The project only worked because each role did its part. When the developers got stuck because the API documents were missing, I brought the issue to the Product Owner. It was fixed in a few hours.

The Product Owner kept the backlog organized and pushed for mobile compatibility after early feedback from stakeholders. This kept the team working on the most important features.

Well, the developers built the main parts: destination listings, booking, and account management. They also worked together in pair programming to fix a last-minute bug in the payment module.

The tester created and ran test cases for each sprint. This caught problems like the booking form not handling dates right before release. These efforts made the product stronger and more reliable.

Agile made it easier to split user stories into smaller pieces. For example, the story “As a traveler, I want to filter destinations by budget so I can plan an inexpensive trip” was broken into smaller parts and finished in two sprints. This let us release a basic filter early, get feedback, and then improve it later. In Sprint 2, the customer asked for a “Recommended Trip” feature. With Waterfall, this kind of change would have delayed the whole project. With Agile, we moved less important tasks to the next sprint and did a quick research spike to test the idea. This let us keep moving without losing quality.

Clear communication was important. A Slack message about missing error handling in the booking form led to the issue being fixed the same day. A sprint review email listed what was finished, what was not, and what came next. This kept the team and stakeholders on the same page. Both messages worked because they were short, direct, and aimed at the right people.

We also used tools to stay organized. Confluence was great for meeting notes, design choices, and technical details. Well using JIRA helped track the backlog. For example, It showed sprint progress and burndown charts. Such charts and daily standups gave us a quick way to bring up blockers and share updates. These tools together worked well with Scrum’s focus on transparency and adaptability. T hey kept the project steady and on schedule.

Agile had good points and bad point . One good thing was flexibility. We ould change plans when requirements change . Another good part was delivering working features early. This gave stakeholders confidence and let hem give us feedback. Testing all the time als helped. Bugs were found before release.

The bad side was rework. When priorities shifted, we sometimes had to redo things. Daily m etings also took up time, which was harder for remote members. ven with those problems, Agile-Scrum still worked better than Waterfall. It let us keep moving without big delays or high costs.

The SNHU Travel roject showed that Agile can deliver quality software while handling change. Good communicatio , clear priorities, and steady delivery helped the team meet customer needs. The lessons learned here can support other ChadaTech teams as they move toward Agile.