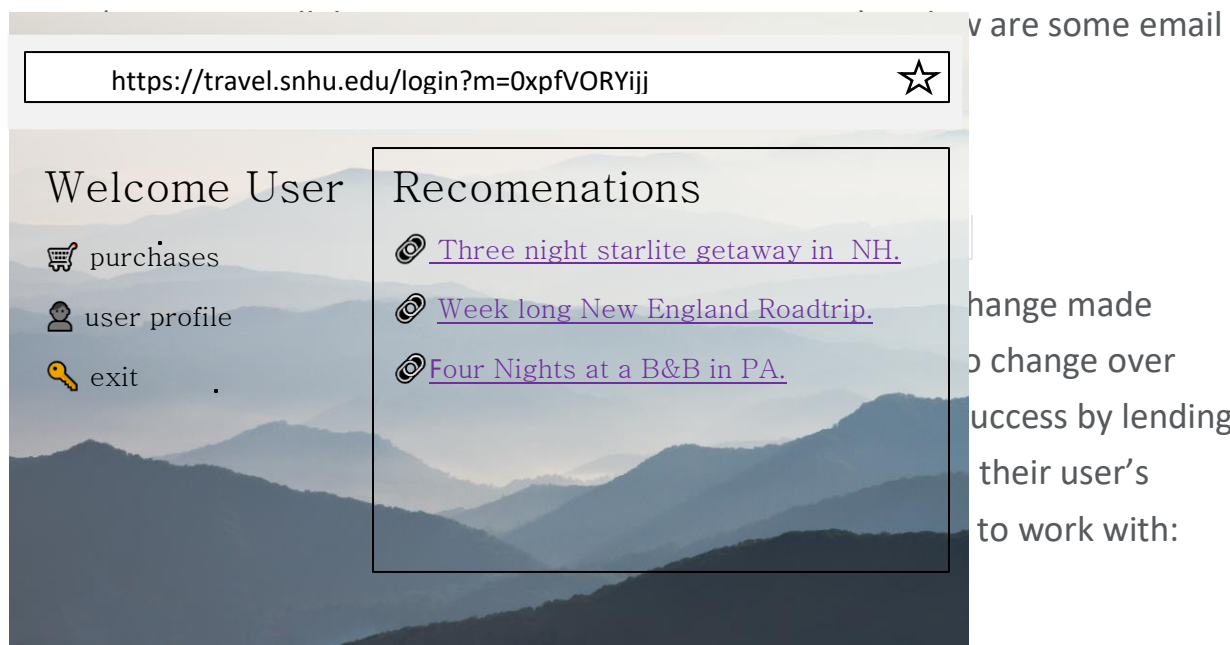


How the Team Wrapped Up

The team came through with all the spirit imparted to them through the Agile Software Development Manifesto. The users, stakeholders, PMs, and developers all had a hand in making the project work as expected. We can antedate success here with the level of quality the group delivered to the project through feedback.

The group took the time to listen to each other (choosing individuals and interactions over processes and tools). They focused on getting the software up and running so they had an end-to-end system to mold into what they all agreed was the finished product (working software over comprehensive documentation). Users and Stakeholders alike connected with the development team and vice



It is through the connections we make that we see a reflection of the success we can distill from our work. We started off with a java rest application and settled

on an html website with a content delivery network better connecting the client to their users.

The PM and lead developer and all the QA work the team focused on lead to adequate prioritization and timely execution of key project deliverables using an Agile software development framework to suit their needs. They worked the backlog like the best of them, spinning up deliverables and applying feedback sprint in and sprint out. We went from manually adding a general list of top destinations to automatically updating a rolling list of recommendations to users based on their search history. (Thomas,2021)

Scrum in Action

Scrum made a good deal of sense for getting a user facing portal out to an existing user base in a month. The team had to compose their work in weeklong sprints and had to focus efforts on balancing bids of functionality with the client's requests.

Our project started off with a charter to align the team to the task at hand and the time limits that be. This defined the commitment the team was making to the project; and solidified of commitment to a good team dynamic. The charter gave us the criteria for a successful deployment:

Key project objectives: End to end site with hosting option with alpha and beta testing done concurrently within five weeks.

This set the tone and motivation for the efforts we put into our team wide communications, including the user base too. From email questionnaires to the daily scrum; we all got to make the project our own, solidifying proofs of how we added value to the project. In particular, the use of user stories and testing their-in was a quantifiable way to demonstrate success to stakeholders and assure quality to consumers of the product.

So that I can <achieve some goal>.

See the most popular locations for travel to educate myself on the best places to go.

Isolate vacation packages of interest to me.

See as many results as I want.

Have my preferences cached and used to present me with a custom destination list.

The use of user stories with user feedback via interviews made sure that stakeholders could see the project composing itself with strategic

functionality that added value from the user perspective in line with the project charter.

(Rodrigues, 11/12/2023)

Measuring by User Story: Design, Build, Test, Review

User stories bring us the aggregate of functionality that we want to see in the project. We designed the software by visualizing user stories as code. After Building the solution, we work to test and ultimately review the code to better adapt the solution to the user's needs. We can prioritize and change the user story backlog as feedback rolls in. The process of making sure that stakeholder and user feedback changes design decisions will be checked with the use of test cases applied to user stories.

	required components.	link to the top travel-package-deal booking.

(Rodrigues, 11/17/2023)

This feedback loop that involves designing, building, testing, and reviewing to make sure that there is a series of filters that give us the most value out for effort in. User stories are locked into our memory at each stage of the sprint and form the scaffolding of everything we build.

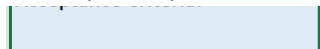


(mendix,2023)

After agreeing to the priority of user stories we build code using a backlog like this:

4	4	Large	End User	have my destination list taylored to my travel history	Have my preferences cached and used to present me with a custom destination list.
---	---	-------	----------	--	---

As a developer we then go on to isolate the user story and draw conclusions to how best complete the solution and piece together with the remaining stories; all while being aware of future changes to come through numerous rounds of feedback seen throughout an iterative development pattern. Below, a developer responded to the user's feature request.



Later in the project, the team moved from a list display to a slide show display in order to present suggestions. Ultimately, we integrated the project into a website. This was all made possible because the team kept close tabs on the feedback that they got and made sure to suggest the most appropriate

design implementation to take on. Code was easily refactored and re-implemented to reflect changes made to the project.

Tools of the Trade: The Good, the Bad, the Unknown

No one project management style is the know all be all. Using a framework like Nexus or a style like Scrum can move mountains for teams, even have them save face despite stakeholders' doubts.

(Graham,2012) On the flip side, locking the team into any one development style or tool, like Scrum, can be counterproductive in certain cases. We need to be flexible and stick to the basic tenants of the Manifesto; even Dave Thomas doesn't like to test (Thomas, 2015). We see that teams can do great things when they have the autonomy to get a feel for what sort of Agility they'd like to achieve as is the case with Amazon's Amazonians and their two-pizza teams. (Leather,2017)

Ultimately, for the project to come to fruition we must see progress being made at a rate that is effective. Estimation is a key factor in getting the job done right and on time. Teams being comfortable with the development conditions is paramount. In some cases, can have Scrum masters leading teams astray, developers not focusing on the team effort (taking on too much work without collaboration or not collaborating all together) There had been instances where developers take little interest in integration of various deliverables, or where ownership is not effectively shared. Consequently, leading to diminished project quality. (Rodrigues, 11/05/2023)

Meetings and communication methods are great venues for group members to align their efforts and avoid the folly of poor team dynamics. Other members can help guide team members to take on better behaviors and encourage them to aid in the development effort. In our team exercises in module 6, our team gave key reports on the user story metrics to guide our stakeholders using kanban style backlog of artifacts. I had too suggested the use of a project management platform like the ones used on GitHub that allowed us to document our development efforts all within a platform that allows us to version control but automate continuous integration as we develop. (Rodrigues, 12/2/2023) In the end communication seems to be at the root of the distribution of quality.

References

Rodrigues Eduardo. User Stories. 11/05/2023/ CS 250 Module Two Reflections on the Daily Scrum . SNHU. CS-250.

Rodrigues Eduardo. User Stories. 11/12/2023/ CS 250 Module Three User Stories Assignment. SNHU. CS-250.

Rodrigues Eduardo. Test Case Series. 11/17/2023/ CS 250 Module Four Test Case Assignment. SNHU. CS-250.

Rodrigues Eduardo. Weekend Review. 12/02/2023/ CS 250 Module Six Group 5. SNHU. CS-250.

Graham, Dorothy. Fewster, Mark. (2012) Experiences of Test Automation: Case Studies of Software Test Automation, Addison-Wesley Professional.

Ken Schwaber, accessed 2023. Nexus. scrum.org. <https://www.scrum.org/team/ken-schwaber>.

Leather, Ellen. 2017. <https://medium.com/frontira/amazonian-agility-e3720ff004f7>.

Mendix. 2023. <https://www.mendix.com/wp-content/uploads/agile-feedback-loops.svg>

Thomas, D. Et al. 2001. Manifesto for Agile Software Development. <https://agilemanifesto.org/>

Thomas, D. 2015. Agile is Dead. Goto; conference. <https://www.youtube.com/watch?v=a-BOSpxYJ9M&t=4s&pp=ygUNYWdpbGUgaXMgZGVhZA%3D%3D>