Corey Skinner

8/12/2022

Sprint Review and Retrospective

During the course of the SNHU Travel project there were several important factors that played into it coming to fruition. Overall there were challenges, but using the agile method many of them were overcome. The Agile method allowed us to make quick changes and incorporate changes based on customer feedback. This proved beneficial to the end product.

There were several changes made based on customer feedback. The first major change was the layout of the site. Originally we had implemented a list on the front page with the travel options. This was changed based on customer feedback. This has changed to a slideshow to reveal the top five travel destinations. The slideshow has buttons to interact with for skipping to the next, as well as going back to the previous spot.

Following this there was another major change. The product owner found out from the customers that the most desired vacations were detox and wellness type vacations. This put a bit of a damper on things as we had to find new vacations to fit into this category. Luckily through agile and the planning process we were able to identify that this was a desired outcome soon enough and make the proper changes.

The second change may have not been as big of an issue though had we taken other previous user stories into account. Some other suggestions that had gone unresolved this sprint were, price range options, learned preferences for users based on their history, and vacation categories. The price range would of course be a nice feature that would have been easy to implement. The learned preferences would have probably been ambitious and underwhelming at this point in the product's life. The one that would have made a big difference in the project would have been having categories of vacation types. This type of forward thought could have saved significant time by arranging the program differently. This is however something that could be changed in the future as the core of the program still exists and works.

The change of requiring all vacations to be wellness and detox vacations had other issues aside from what was already mentioned. Most of the team leads had pushback exclaiming how it would be detrimental and take too much time to implement. Product owner insisted on keeping the same deadline, but forgoing other user stories. The developer was concerned that most of the code would have to be scrapped. This turned out not to be the case.

One good thing that happened at the meeting was that each member did acknowledge what they could do to support the change. I know this was mentioned before but to bring it up again, the vacation type changing should not have had any effect on the program itself. Had the development team implemented vacations as an object, then the primary code would not have had to change, just the database. It was helpful to have the SCRUM meeting to be able to discuss changes, and get them implemented, but the type of vacation should not have been as big an issue.

When updating the code there were minor changes that had to happen, but they mainly rotated around finding new vacations and updating the ones in the slide show. In my previous vacation slide show I had mostly beach locations featured. The pictures, vacation packages, descriptions, and locations all had to be changed. The other change I had made at the same time was to implement alternate colors for the text. This I believed would help the user understand that each slide was different and unrelated to the last.

The following part was based around teamwork. We had a discussion post where each member talked about the project, taking on a different role. Our project owner took initiative to volunteer for his spot first, sending out an email and making sure that each of us were aware of the project. This was helpful as I had not checked up on the project yet. Our product owner laid out clearly what he was going to address and left us to decide what roles we wanted and ideas for those roles.

The other member that posted aside from myself was the developer. They had some ideas as to planning ahead to set us up for success. Another idea was to utilize Jira for a ticketing system. For the organization of the developers he was looking to divide and conquer the tasks.There was absolutely an eagerness about the communication though which was inspiring.

My post was based around myself being the tester. I chose that role since I have experience with hardware testing and writing scripts to test the hardware. I had a couple goals. The first was to ensure that developers and testers worked closely together. This would increase productivity and effectiveness of the testing. The second was to avoid the issues of the previous project and that was to have the team work with multiple teams at once, and not get stuck in one particular group. This is to avoid getting locked in to working with one group. There would absolutely be some initial issues with it, but in the end, each member would be versatile enough to cover for each other. This would provide greater coverage, a backup plan if somebody is out, or leaves, and also make it easier to take on new team members by having more team members able to help.

In the replies we all supported the ideas of each other and mentioned what we could do to help out with their plans for their teams. The overall attitude was to keep up with communication and give accurate, up to date reports from each of our ends. Other ideas that were tossed around were each of us taking a Scrum course. Our whole group agreed that this would be beneficial. The other part was to utilize a small group mentality in order to minimize bureaucracy and increase efficiency.

As a side note, I would have enjoyed having a discussion board where we did get a user story and got the chance to have a discussion board on it. Following the post, we have to fulfill our roles in the group for a week or two. That could have been a really interesting project.

The SNHU travel project did not make the most of agile. There were many points where they could have made improvements, but did not. One glaring example was the lack of a price fixing option. Customers would never want to see a bunch of vacations outside of their price range. They will instantly go elsewhere. Then there is the issue again of the specific vacation type. This crossed a line with a previous customer suggestion. That suggestion was to have vacation categories. Had the team implemented this, or made the program more modular, then they could have gotten away with far less changes.

Due to the time constraints and that they threw out most of the customer suggestions, agile may not have been their best choice. Now the group could call it a learning opportunity and if they do learn from it, then great. The biggest problems are that they listened to the customer feedback, but then ignored most of it. If the group can plan things out better, and make a more modular, and adaptable product, then that could work out great for them. But having to change massive portions of the code is what agile is supposed to avoid. I would say if the group is not going to take advantage of the incremental progress that makes agile so effective, they might as well use waterfall, especially when they are on a short timeframe.