Sprint Review and Retrospective

During our Scrum sprint, we took on three main roles: the Scrum Master, the Product Owner, and the Development Team. I will explain how each role specifically contributed to Chada Tech’s travel product so far.

The Scrum Master started us off by teaching us how Scrum works and what philosophies it stands for. They taught us about sprints, daily Scrum meetings, product backlogs, what the different roles mean, and how to interact with each other. After we knew what Scrum was and how to implement it, the Scrum master then held us accountable. They would make sure we were coming to the meetings, contributing to those meetings, and giving examples of how to give constructive criticism and communicate honestly. On the days the Development Team was doing great the Scrum master would sit back and let the team operate. On the days the Development Team was struggling the Scrum master would intervene and get everyone back on track.

The Product Owner wasn’t worried about how productive the team was being but was focused on the product that was being developed. They assembled user stories, both initially and throughout, in the product backlog to guide the Development Team. They got estimates from the Development Team on how difficult each item would be and then used that information to order the items by priority. They talked to clients to develop more user stories which allowed the Development Team to work distraction-free. Thanks to the Product Owner the Development Team always knew what they should be working on and in what order.

The Development Team oversaw the delivery of features from the product backlog and participated in the Scrum framework. They went to all the meetings and contributed their technical insights like how difficult a feature is, and how long a feature would likely take if something were possible. They always collaborated in an honest and constructive manner. They debated amongst themselves and used tools like voting poker to make sure a group census was made when making a technical decision. The Development Team would also work on implementing features from the product backlog when they were outside of meetings.

Using the Scrum-agile approach to the SDLC allowed us to accomplish all the user stories and it also helped us to get the right user stories done first. The main advantage of Scrum over Waterfall when accomplishing user stories is that Scrum's focus is constantly being adjusted so that the most relevant or important stories are being done first. This allows the team to always work at peak efficiency allowing them to accomplish more stories quickly. Any given story might take longer in Scrum but that would only be because it was a low priority, and low-priority stories don’t need to be done until the end anyway. Theo, because of the way Scrum events are structured, the whole Development Team is aware of what the main stories being worked on are, which gives team members an opportunity to give advice if they have any.

The Scrum-agile approach allowed us to complete the project even when it was interrupted or changed directions. In Scrum, nobody really has any expectations about what they will be working on later. The Product Owner keeps the product backlog up to date with the highest priority stories at the top. So, the product backlog could change at any point and the team is aware of and ready for it. The only thing guaranteed is the work that was assigned for the day at the daily Scrum meeting. Typically, major changes won’t happen till after a sprint and the backlog refinement meeting, but Scrum can account for it if it does. Scrum's modular incremental approach allows it to be built from whichever user stories bring the most value first.

We communicated effectively using the Scrum framework. I would like to give a couple of examples of exactly what was said and explain why it was helpful. In the very beginning when we were at our first daily Scrum meeting, the Scrum master helped set the atmosphere. They gave us three suggested questions to answer: 1. What was done yesterday? 2. What is the goal for today? 3. What impedes me? The Scrum master then gave an authentic personal example of the answer to those three questions. The answers were not relevant on a technical level, because the Scrum master doesn’t do any developing, but they helped to establish the kind of honest communication that was expected.

Another example of how we communicated effectively was during the sprint review meetings. Everyone made sure not to clap and give too much praise to encourage transparency. You might think that praising good work would be motivating but our team decided it encouraged team members to make things look better than they were. So, we decided that it was best to go with no applause and no excessive praise to encourage transparency.

As far as the effective tools and principles we used for this project, I can see two categories: general purpose and event-specific purpose. Probably the most important tool you can’t go without is the Kanban board, or equivalent. The team must be able to see what’s in the product backlog and they must be able to see what they should be working on. Another helpful tool is time boxing. Time-boxing is basically allocating a certain amount of time for anything, for our intents and purposes, Scrum meetings. This forces the team to talk about something for the allotted time, which encourages them to collaborate. Pair programming can be used if you want two people to be on the same page.

Some tools are only applicable in certain Scrum events but are still crucial for making things run smoothly. For backlog refinement meetings using user stories as product backlog items was good enough for our project. Estimation cards or voting poker were useful when we were trying to get the team to decide on the difficulty of a user story, or anytime a group census needed to be made. “Side baring” unrelated topics by writing them on the whiteboard to discuss later, helped us keep on track. Using the “what I did, what I will do, what impedes me?” really helped the team format their thoughts. Using safety checks during the sprint retrospective allowed us to make sure everybody felt safe to speak freely, just because one person feels they can share doesn’t mean everyone does.  
 In conclusion, I will spend these last couple of paragraphs assessing the effectiveness of the Scrum-agile approach. I will start off by listing some of the pros and cons. One of the biggest pros was that we were able to change the project halfway through, we changed the webpage to a keynote format. That wouldn’t have been possible in Waterfall, and the product is much better because of it. One of the only real cons was that we had to spend some time learning Scrum and we had to develop our teams’ relationships. Which is a short-term upfront cost that I think will continue to pay dividends down the road.

If I had to say whether the Scrum-agile was the right approach for this project, I would say that it was. The Product Owner didn’t have a definite idea about what the product should be, and Scrum allowed us the framework to find it along the way. This way is better because it is often easier to figure out what is best after having something to work with. Planning everything out in advance requires so much forethought that it is easy to get lost in the weeds. Focusing on what makes the most sense for the current sprint is more practical and more easily understood. Which allows the product to be built piece by piece in a more fluid and efficient manner.