Alexis Prazak

6/25/2024

Tarik Iles

Sprint Retrospective

Retrospective for SNHU Travel

It’s important that everyone understands their roles before starting a sprint. To help teams understand what is meant by “roles”, In the 2020 release of the Scrum Guide, the term “role” was replaced with “accountabilities”. It prioritizes an ownership mindset necessary to execute Scrum well(*Accountability, responsibility and roles, n.d.*). This shift in thinking helps people understand their roles, even if the work is ambiguous.

The responsibilities of each role are all equally important. The product owner may have the most contact with stakeholders, but they could not do their job without the development team and Scrum Master supporting them. Before the Sprint the product owner spoke with customers about the features they would want in a travel booking service. They use these notes to create a product roadmap. The owner then worked with the scrum master to create user stories. Then the development team created acceptance criteria and testing criteria for these stories.

It was clear during the SNHU project that the team had experience creating great user stories. The information had everything a Scrum Master could want, including acceptance criteria(Administration, n.d.). We can look at the user story requesting the ability to set price limits as an example of this being done well. As a tester it was easy to imagine what this feature needed to do. I knew it had to accept an integer input, be included in search filters, and be accurate everytime. If our user story just asked for a way to filter trips would we have gotten this acceptance criteria?

When user requests become vague problems follow. The desire to shift to a wellness-focused application may be a good market move. However, there needed to be more information on how these new features should perform. This led to developers spending time investigating the stakeholders' meaning instead of writing code. To clarify requirements the Product Owner conducted focus groups with customers, taking notes about what they wanted when booking travel.

While working on the project management decided to change direction. They wanted to make a wellness-focused application due to changing market demands. The user interface would have to be adjusted to accommodate this. New requirements were added to the backlog. We would also have to test while writing code since Scrum does not keep the testing phase separate(Maniraj, 2021). Testing at the same time allowed us to deliver a working product more quickly. Under the Waterfall method, this would be done after building. Waiting to write test cases would have only made development take longer. This is not something SNHU Travel could afford in this competitive climate.

In an ideal world the team would all be located in the same place. Face-to-face meetings allow for better communication and include important non-verbal cues. Since I am remote this isn’t always possible. When possible video conferencing and instant messaging tools bridged this gap. Emails were also a valuable tool for communication.

A good example of effective email use is the one our tester sent to clarify the new slideshow’s details:

“I appreciate your clarification on the last round of testing. We are currently revising our test cases to incorporate these requirements. Before the next round of tests, I’d like more information about the mobile application and top destinations page.

* Would you like the mobile app to also show destinations as a slideshow? Or would you like a scroll bar, which is typically more mobile-friendly?
* When a user is not logged in what sort of trips would you like destinations to show? A few options include the most booked trips, the highest-rated trips, or a list curated by the marketing team.”

The best part of this email is how concise it is. The bullet points make it easy to find key information. Communicating in Scrum should get directly to the point no matter the format.

Once we had information about the slideshow I needed to organize this work. As the Scrum Master I use a combination of techniques to set us up for success. Throughout development we all accessed a digital “whiteboard” on Microsoft Excel. This housed the product backlog and allowed the team to reference it while working. It was used as a reference point during daily standup meetings. I could use it to ensure it met acceptance criteria before presenting the product to stakeholders.

Trying anything new requires patience. My biggest disadvantage when using Scrum at ChadaTech was the time it took to train myself and the team. Even after this training, we made mistakes that are common for newcomers. When creating testing criteria for the slideshow I added in features that hadn’t been requested. This type of “scope creep”, where a project’s requirements keep growing, is common when there are ambiguous criteria. Scrum is commonly used for situations with unclear requirements, which teams used to rigid guidelines struggle with.

Due to the short timeframe of the project, Scrum was the best choice for the Chada Tech team. It allowed us to pivot quickly when changes needed to be made. A waterfall method would have worked better if the requirements weren’t going to change. But they did multiple times. We can look at changing the layout to a slideshow to see where waterfall fails. Any significant change means adjusting the cost and acceptance criteria. All of those are determined during the requirements and design stage of the traditional Waterfall method. If we were using that method we would’ve needed to return to the first stage of the process.

References

*Accountability, responsibility and roles*. Scrum.org. (n.d.). https://www.scrum.org/resources/accountability-responsibility-and-roles

Administration, G. S. (n.d.). *Writing effective user stories*. Writing Effective User Stories - Tech at GSA. https://tech.gsa.gov/guides/effective\_user\_stories/

Maniraj, S. (2021, July 30). *A guide for software testing in Scrum*. Software Testing Magazine. https://www.softwaretestingmagazine.com/knowledge/a-guide-for-software-testing-in-scrum/