Alex Frankel

CS 250 Final Project: Sprint Review and Retrospective

Any good Scrum team works together like an organism. Everyone has their unique role that is of vital importance to the body. We start with a Scrum Master, which is expected to facilitate Scrum events such as sprint planning, the daily scrum, as well as the sprint review and retrospective. The Scrum Master will work with the Product Owner, who represents the customer, to create a product backlog of tasks and features. This backlog can be updated throughout the sprint to reflect changing priorities or additional features. The Product Owner works directly with the customer to define any expectations the customer has and to make sure that a deliverable product is created by the end of each sprint. The backlog was created through user stories, which customers communicated to the Product Owner and were then finalized into actual features. This creates structure for developers and testers whom will add incremental and usable features throughout each sprint. Now that the sprint is planned and the backlog is ready, they can finally begin to create!

As developers, in conjunction with testers, we are responsible for the execution of all of the collective efforts of our team members. Within this sprint we completed a rudimentary representation of one of the user stories that we collected. The final product is a website that displays vacation deals. We broke this down into different features that could each be completed within a single sprint. The final product of this first sprint was a graphical representation of our top 5 recommended wellness and detox vacations. This follows a user story that suggested that the website should include a top 5 or top 10 list. This sprint focused on a single deliverable feature and left others out for future sprints.

I was also given an opportunity as developer to address the customer, the Product Owner, and a focus group of users for their opinions. Facilitating communication is one of the most important things that we can do. Without it we are doomed to fail. The focus group and I (as Product Owner) brainstormed ideas that I would later organize as user stories for potential features. These features included top 5 destinations and options to sort based on preferences and budget. While these were all great ideas, they were not necessarily organized in the way the final product might be. To make the most of them, I (as Developer) would need to communicate with the customer as to which of these features should be included, and which ideas could be consolidated into singular features. Only then could I safely prioritize the product backlog. Specifically, I asked, “What of these features (in particular) are requested for this build?”. If the Product Owner doesn’t pull their weight by representing the customer’s wishes, the whole team will falter. This is why this communication is crucial- so that everyone is on the same page and knows what is expected of them as early as possible.

On that note, this is the perfect time to describe some of the organizational tools of Scrum. Unfortunately, we as a team did not make use of any of these tools I am about to describe and could have benefitted in a real software development setting from their use. First thing in the morning is the best time to conduct the ‘Daily Scrum’, also known as daily stand-up. This is an opportunity for the whole team to briefly catch up with each other on their concurrent progress as well as their plans for the day. While not necessarily required for this event, the Product Owner has an opportunity to communicate with the customer first thing in the morning. This could happen before the standup to solidify the day’s plans and prioritize the product backlog of tasks. Alternatively, they can share the team’s progress with the customer after checking in with the team. A popular solution to this is the use of ‘information radiators’ which are management tools such as a whiteboard or their virtual counterparts that display necessary information to the whole team. This consists of the status of different tasks: an information radiator should have plenty of space for team communications and must be updated frequently in order to provide the greatest value to the team. If we were to use a virtual tool such as Jira, the development process could be completely streamlined. Jira graphically represents the whiteboard, which acts as an information radiator, only it has more functionalities built right in. Instead, we were limited to direct communications. Agile promotes face-to-face communication as the most effective way to engage a team. In small projects such as this it may suffice as long as the whole team is included, and our discussions are substantive. This was apparent when drawing from the simulated discussions of the team members in the provided slideshow. In this way we were successful- for this sprint we were all included in face-to-face discussion and were able to complete a central feature to our product together. Even when the customer wanted us to shift focus mid-sprint we were all present and capable, which I will explain in detail.

Of course, even the best laid plans may fail. The customer threw a curveball at us halfway through the sprint that their business model had changed. If, in this scenario, I was able to communicate the concerns I have previously discussed, this would have varying consequences depending on the planned features and so on. They wanted the website to feature primarily wellness and detox vacations. The team was able to simply reprioritize the planned features and roll out the request within the same deadline. Again, this affects some of the previously planned features, specifically the kind of trips shown on the website. It is important to know which functionalities the customer still wants to implement. Luckily, we had not created anything yet that would have to be scrapped. It instead improved our process! We would move on to create exactly what SNHU travel wanted.

Scrum works very well with change, foregoing a rigid plan-structure to one that is adaptable. Because of this, what Scrum may be lacking is consistency. A waterfall method in this case would attempt to finalize all customer requests long before development began. Overall, the Scrum approach proved to be effective for this type of project. The final product was able to be broken down into incremental deliverables that could be focused and tested individually. Since our user stories followed this format in the first place, we were destined for success!