Brett Sears

Sprint Review and Retrospective

When developing applications, some techniques work better than others depending on the intended goal. For this class we've learned and analyzed the Agile approach. Prior to this methodology, developing teams typically preferred a Waterfall approach. The Waterfall standard was to split the process into stages such as analysis, design, and development sequentially. An Agile approach instead separates the project into numerous sprints each compromised of their own analysis, design, and development. Taking things in chunks like this allows for flexible changes and fixes without significantly stunting progress. Additionally, Agile's setup alters the team composition and internal components of development.

Within an Agile team, there are normally Product Owners, Scrum Masters, Testers, and Developers. The Product Owner communicates between the team and stakeholders. Product Owners also develop product backlogs of user stories which contain a prioritized list of project segments and numerous details on each. This is significant because obviously without knowing what stakeholders are interested in and communicating those details, any products developed will fall flat with the audience. As the project progresses, communication continues which may lead to changes, and it's the Product Owner's responsibility to make those desired changes known as well. During the SNHU Travel project, the stakeholders decided it would be a better idea to focus on detox and wellness travel locations instead of standard tourism travel locations. The Product Owner had to make these changes known to the team and alter user stories to, for instance, list top five wellness and detox travel destinations, which the testers and development team then had to change the product to match.

Scrum Masters are a part of the development team, but have responsibilities to host daily meetings, help remove obstacles in the way of testing or development, and setup the team's charter stating the project's goals and the team's expected behaviors. Daily meetings allow team members to discuss their progress and any issues in the project, but each meeting is limited to 15 minutes, so the Scrum Master holds the team to these and other standards. One example charter was created during this SNHU Travel project, which listed ways the team is expected to treat others with respect, have daily meetings, communicate in various ways, and more. This is one simple way the Scrum Master can help set the team up with tools necessary to properly proceed through development.

Testers and Developers work with each other to piece together and create the working applications based on the user stories the Product Owner listed. Testers list user inputs and each input's expected results as well as clarify details of user stories with the Product Owner. Based on those inputs, expected results, and other details, the Developers create the application to meet those qualities. After any necessary clarification or changes, the tests may change, and with that the Developers must alter the application to meet new criteria. This happened in the SNHU Travel project as the Product Owner clarified the destination layout to be a slideshow instead of a list format.

User stories in an Agile approach are worked through quickly and effectively partially because they're small chunks of the product. Any changes are planned and implemented within a short sprint, roadblocks are tackled immediately, communication of any issues or clarifications are daily or immediately with some tools. Additionally, due to the details of each user story and test, placeholder development is straightforward and simplified.

The earlier example of change in the SNHU Travel project from standard travel destinations to detox and wellness showed a strength of the Agile method. Where a Waterfall approach might have planned everything and possibly designed some of it prior to the change, thus requiring an entire reformatting of the project like a domino effect, these Agile sprints allow changes to be designed and implemented on the spot. A change is communicated, current tests are restructured to fit the new criteria, and the developmental changes are worked in to meet them. It might have some rippling effects, but certainly not as much as if the entire project was already planned out. In some cases, such as to user stories not worked on yet, the changes may not even affect any existing code, which means very little is altered as opposed to again having to restructure because of a full plan from a Waterfall method.

Similarly to any team situation with the need for complexity or flexibility, communication is of utmost importance. One of the aspects of Agile is segmenting production into quick short sprints, and anything blocking progress such as personal issues, tech issues, changes in design requiring alterations, or otherwise needs to be quickly solved. To do so, Agile implements daily meetings, face to face or over webcam meetings, and open trusting communication. An example of relaying information and requesting clarification is the emails we made as testers to the Product Owner. Something such as "I am using your provided user stories to create testing procedures for features to conclude if they function sufficiently. I would like some clarification and additional details concerning some of the required criteria. Would you mind providing these?" With details provided after, this becomes a quick and easy solution to a lack of information. Similarly, during daily meetings, people are encouraged to speak up about their progress and what is blocking any. Opening to others about one's contribution and availability for help creates a working team environment.

These daily meetings are one of the tools Agile provides. Another is the concept of information radiators. These are when tasks are posted onto a board, assigned to people, and organized in categories such as in progress, stuck in place, completed, or any other the team considers proper. The boards are also typically visible to the Product Owner, other teams, or possibly more. This gives people a good idea of the entire project and how it's going without even the need for verbal communication, although further contact is suggested. There are also online boards available for teams to use which they can use from afar and update from wherever. Ease of access makes tools such as these all the more useful, which increases efficiency and availability of communication. Additionally, Agile uses charting and tracking methods to estimate completion based on what's completed during sprints, which allows the team to properly communicate to stakeholders when and what can be accomplished at various stages.

Overall, Agile seems extremely useful in developing applications. While Waterfall approaches might be more simplistic and controlled, Agile allows for flexibility and continuous progress because of short sprint completion of project segments in user stories, has free flowing and encouraging communication in daily meetings, boards, and a more equal team structure, allows for alterations and suggestions deep into the project's production, uses more accurate product estimations from sprint completion and timing, and more. Due to these benefits, Agile was the better approach for the SNHU Travel project. Changes from the stakeholders would have clogged up progress, but Agile's adaptability allowed them to be implemented without being a major setback. Juxtaposing design with development allows quickly finding and fixing issues rather than requiring significant replanning. Communicating with the rest of the team allowed immediate clarification and progress. None of these would be as efficient in a Waterfall approach.