Paul Alicea

Sprint Review and Retrospective Meetings

**Scrum Master:** Welcome to our sprint review and retrospective. Usually these are separate meeting. For the sake of our current time restraints, we will be doing them together. Let’s get started on the sprint review portion of this meeting and go over how we can improve the product. So far we have gone over a few working prototypes of the final product. We have tried putting together a top 5 vacation packages page, we have implemented a search bar and a profile page that allows the user to customize and filter out their search results. These are all great features for the application. As of recently we have also implemented a top 5 wellness and relaxation resorts page. I am confident that all these features will lead to a popular product among users. As we know, the product is not finished until our product owner acknowledges that all user stories are complete, and the stakeholders say we are done. I do believe that we can allow cross functionality of our previous features to our backlog to allow for a more robust user experience. I have went over these details with the product owner and they agree that this would be an amazing feat. The product owner has also run this by the stakeholders and they are excited to see the results of this cross functional feature. To elaborate, the goal is to be able to create the ability to use a multi-layered filtering system to make it easy to find the vacation you were dreaming of. Let’s say in my profile I already filtered out that I only want a camping vacation. However, from that camping vacation page I can also filter my search further with lets say I want bike trails. So now I type in bike trails in the search parameter. Now what comes up is a page with camping sites that also offer bike trails.

Now to move into our sprint retrospective part of this meeting. As you know, there have been a lot of changes to our software development life cycle. Previously we were using a waterfall methodology to complete our product. Using that methodology, we may have struggled quite a bit with this product so far. Creating a definite plan on paper sounds nice but is not feasible in this situation. The stakeholders have made several changes to the priorities of the product within the last few sprints. In a waterfall style of planning, we would have lost a lot of time and labor in developing this product due to the fact that major changes would have led to essentially restarting huge portions of the product. Using agile we are able to test the product along the way so we know there is a working copy of the product and how the product behaves. This is made possible by having our developers accomplish the most amount of work in the least amount of code. Our pairs of developers create their parts of software with a working product being the result of every sprint. Out testers create test cases based on the intended functions of the product and make sure they emulate the user experience in as many situations as they can to make sure there are no flaws or bugs. Our developers also use test driven development to test the code as it’s being written. This makes sure the code works as it’s being written so that there is less time being spent on error checking and finding bugs in the product later. Previously we had our testers help with test driven development as we had not reached a point of having a working product yet. That is no longer the case and now they are focused on their test cases. Once again in a waterfall style of planning there is no working product until the end in most cases, therefore many bugs may have been found at the end of development cycle or after release. This leads to overworked and overwhelmed development teams and frustrated users. Agile prevents these problems from occurring.

Using Microsoft Azure as well as scrum events we can stay on task. Azure allows us to create our backlogs, organize our meetings, set up Kanban boards and gives us access to our slack chat rooms and our GitHub account. Using this we can all communicate and review any work completed. The daily scrum helps us keep on task and is a great way for us to briefly describe our accomplishments and roadblocks so that we can reflect and move on to the next phase of our day. The user story refinement meetings helped us get on track with updated requirements from the stakeholders thus increasing the value of the product.

Using agile instead of waterfall for our team was an interesting change. We are still working on improving our agile approach. Even though, we are not quite masters of agile yet, we were still able to save time, money, and quite a bit of headaches during development. Using our paired programming methods and test driven development we saved time on finding bugs later. Having our testers test the software along each step of development has saved us time by identifying bugs as the product is developed instead of trying to fix bugs at the end of development. It is much easier to fix bugs within smaller sections of code. My job as scrum master is to keep you guys on track and assist with maximizing efficiency in agile principles as well as helping maintain morale and reducing roadblocks. This is a great team we have and I have confidence that we will only get stronger as a team. Our goal as a team is to assist each other in not just developing the software but also developing each other and I feel that using scrum based agile is the best way to accomplish that.