Group: Kashyyyk
2022-10-07

Team Reflection: Week 6

Customer Value and Scope

This week the whole team has felt that it's been challenging to reach success with the development of our application. It's been a little unclear what everyone should be working on and how our user stories and their tasks should be interpreted. We concluded what happened after a group meeting to review the week's problems: our scrum planning for this week was lacking in details and clarity in comparison to our previous sprints. Since this week had a lot more technically difficult challenges and functions to implement, this influenced our ability to make user stories that cohered with INVEST and tasks that followed SMART to the same extent as in our previous sprints. We can still find a positive outcome from this, and that is that we have learned what not to do, by doing what we were not supposed to do.

Since this project is “only” for a course, the fact that we didn't deliver all that we had put upon us and planned to achieve this sprint is not that terrible. However if this would be a project done for a real company where we had a PO that was not an internal part of the project team then the fact that we did not deliver on what we promised would be quite devastating. Now instead of fearing the consequences we can take this as a learning opportunity.

Design decisions and product structure

Throughout the development process we have made sure to put comments in the code we have written, through tools like JavaDoc. This sprint we have tackled a wider scope for documentation, which was made more relevant when more time is spent in the backend. Whiteboard documentation has always been of great use, but clean and thorough documentation felt more necessary now with more complex features on the agenda.

Both sequence diagrams and class diagrams were utilized to describe the technical workings of the product. These diagrams make it easier to describe and communicate how a part of the application should be implemented.

The team has continued to ensure the code quality of our application by the use of pull requests. This allows an assigned reviewer or other team members to go over the contributions of others and approve them before they are merged with the main branch of the application. This ensures that not only the author of the contribution is satisfied with their delivery but also a third party not (or for the most part not) involved in the writing of the code.

It should also be stated that our definition of done includes requirements on the quality of the code delivered which should be fulfilled in order for it to be considered fully delivered.