Team Reflection Topics - Sprint 5

Customer Value and Scope

Before the sprint started we suggested potential features we wanted to include, and then narrowing these down to an MVP. The MVP was fulfilled but not too much time was given to implement other features, e.g a direct connection to a Discord-bot. Our ambition with this bot was that it would simplify the communication with tournament hosts and teams. The reason for this was the restrained timeframe and resources we had, where we much rather focused on factors more valued by our external part. Therefore, our scope did not change that much during the sprint and project as a whole.

The acceptance tests we formulated in a previous sprint have been used in an unanimous way. The steps are the following:

1. Should have an approved code review.
2. The acceptance criterias needs to be checked if done. Them itself should be formulated from the SMART-principle.
3. Travis needs to control the code.

This has been a method well structured and has been followed throughput, even though our external part has not been part of the process. Nevertheless, that we have a continuous method gives value to our external part, since we can present our actions and from that not only ensure code quality, but trust to that person. The iterative problem identification has proven effective when it comes to code quality and structure.

Design decisions and product structure

This week we structured our technical documentation to get a better overview. This means we have not updated it week to week, which probably is the best method for this. However, our reflection was not so positive on maintaining this documentation for our project. Our analysis is that the scope is too small for technical documentation to really give value, compared to the time it takes to maintain and formulate properly. If you would have a bigger project it could possibly work better. For instance, the more experienced programmers in our team can right now know the entire system, which shouldn't be the case. Instead we believe that SCRUM works best when the project is too big for one person to overview it all. When this is

the case, the documentation proves useful since you should need to know the documentation to know the entire project.

As mentioned in last sprints team reflection, our ambition was to update this weekly. However, due to restricted time, this has failed completely. With that said, we do believe that if we were to have proper documentation we would need to structure a way to make sure that the documentation is updated each sprint. An outdated documentation barely gives value due to the uncertainty it could bring. Therefore a way to make sure of this is to make this a part of our acceptance test, ensuring that a task is complete when the documentation has been overlooked. In other words, to make sure that possible code changes that have been made, and what structure changes this could bring, is documented.t