Team Reflection Topics

Customer Value and Scope

So far we have managed to keep our scope without major changes, although this is possibly to change in the coming sprint when we plan to review how much value we can give our external part with the time available. However, we do believe that our MVP is a decent product and we feel confident about the features included.

In this sprint we tried to prioritise making every feature we developed accessible without “magic” URLs, which is URLs that you can enter to access features that do not have buttons. By deleting these we manage to create value for people not internally involved in the project.

Regarding our success criteria we have only touched the subject, and therefore can not really analyze the results from it. We hope to from now continue with having weekly checkups of what the success criteria could be and document changes from individuals.

With insights from our guidance counselors leading into this sprint we improved significantly with user stories, their associated acceptance criteria and task breakdown. Not only did we make them more understandable and easier to implement while coding, we also got a better outlook on the project as a whole. For each and every one of us, the startup time for a new task were shorter this sprint compared to the other sprints. The new tasks were better proportioned so the workload for each task wasn’t as large as it was in the previous sprint.

We now have a standardized method for constructing them, which still is very time consuming. But by having this standardized method with predetermined steps, which we all can apply, we hope to be able to divide the process by working with several tasks at the same time by dividing in little task-groups. This will hopefully lead to better time efficiency.

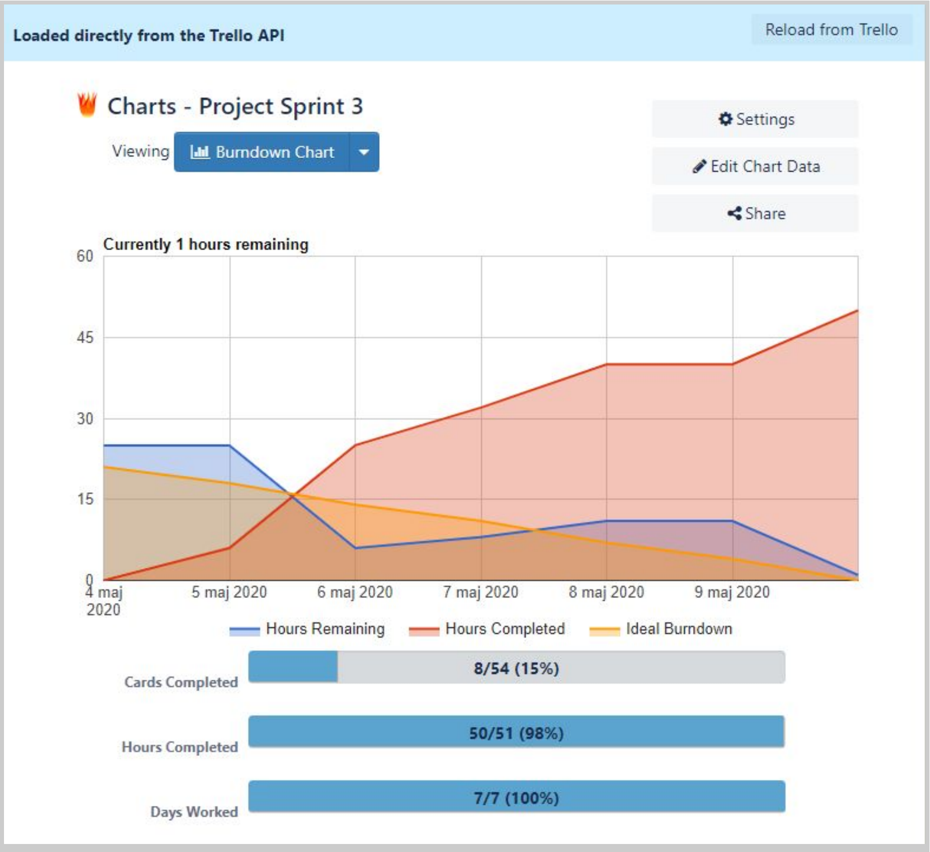
Regarding our acceptance tests, we decided last week that all code reviews should be done in pairs, or at least be discussed in pairs before the pull requests could be accepted. This was unfortunately not followed. Although, we consider this a valid method of reviewing since it helps detect errors regarding functional and structural factors. Therefore we will in the sprints to come apply this way of reviewing, hopefully with better success rate.

Regarding securing the quality of the code we looked at technical debt as a KPI. After discussions with supervisor Jean-Philippe we realised we interpreted technical debt wrong, and what we actually measured was process debt. Process debt we define as what is left to be completed from the last sprint, which is a KPI easily measured in work hours. We have some process dept from the first sprint. It is a task we didn't solve in either of sprint 2 or 3 and it is not solved yet. We need to fix this task before it develops into more problems, so it should be our first prio the sprint to come. This weeks sprint resulted in minor process debt. This was in the case of a task not accepted in a code review on the last day of the sprint.

Regarding our wellbeing we use a survey about wellbeing. The survey and the comparison process is working fine, although we have not found a good way to compare the results so it is planned to made manually, by looking at different results. These are some of our findings while we compared the results:

- One person did not think they got the support needed from the team, in contrast till last week where no one thought so. This should be improved.
- General well being have improved this sprint, which is a success!
- We feel more confident about what needs to be done to pass the course.
- We have continued variety in opinions regarding how to feel about the technologies we use. Some are very comfortable and some are not at all.
- We are more confident in our ability to define user-stories. You could draw a correlation with the improved user-stories we make this week.
- We were more happy with what we achieved this week.

The burndown chart is used to evaluate our process and progress through the sprint. We had some problems implementing this. The first week we tried to implement it as a plug-in to our trello, which failed. Then we found the website https://www.burndownfortrello.com/ which worked fine, till we realized we had to manually update enter the site so it could refresh our information once a day, which resulted in misleading results. But for this week we will definitely make it work. The chart for Sprint 3 can be found in the image below. Note that the blue curve, hours remaining has a positive direction between 6-8 may. This is a result of forgetting to write the hours planned in the sprint planning, and was then filled in on 6 may when the task was started. This is a small deviation we are aware of. Below the burndown we show out Velocity Chart so far. The points are estimated to rise since we felt this sprint had low workload, and we are still searching for a good balance.



Loaded directly from the Trello API

Charts - Project Sprint 3

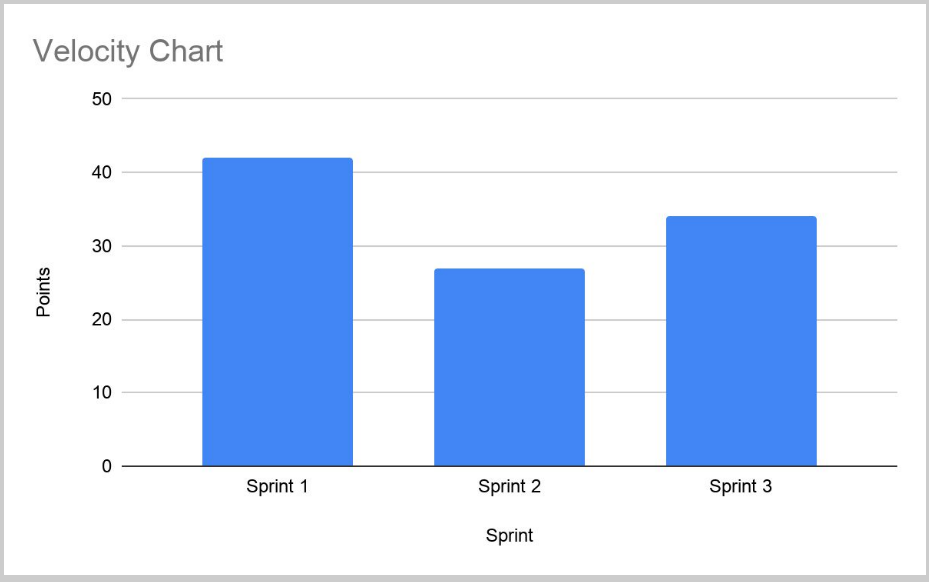
Viewing Burndown Chart

Currently 1 hours remaining

Cards Completed 8/54 (15%)

Hours Completed 50/51 (98%)

Days Worked 7/7 (100%)



Social Contract and Effort

Feeling that out meetings were unstructured and more time consuming than they had to be, we have in the sprint to come added a role to a specific person in the team. This role has an administrative function to structure meetings, remember deadlines, remind other people about deadlines and what needs to be done, creating agenda to have efficient meetings etc. This is updated in our social contract.

Actions have been taken to structure occasions where people can work together. In our agenda there is a section dedicated so people can say when they want to work, find possible team members where the schedule matches and therefore better plan their week and their workload. Up until now this has been done very spontaneously with non-impressive results, and this will hopefully have a positive impact which we look forward to evaluate next week.

We are still fighting with our balance regarding workload. WE have been cautious due to the fact that our teachers have been saying that when learning SCRUM you almost always schedule more tasks than you can handle. We have had that in mind, which resulted in planning our sprints with not enough tasks and workload. This could also be seen in our velocity chart, which in most cases mirror the time spent on each task, since 1 hours is equal to 1 point in the velocity chart. We have less value delivered, even though the hours could be the same, but probably are not. For the sprint to come we look to schedule more tasks so we know our upper limit.

Design decisions and product structure

We planned to define our code review standards, branch strategy as well as our general Definition of Done but this has not been done properly. This is mostly affected by problems in dictation of assignments. Also, this assignment requires knowledge/experience n the area to do it properly. To not let one person do it alone we decided we would take it at another meeting, but then it fell between the chairs. But the goal is that this week have a unified approach to secure that all coding will be reviewed in the same areas.

Application of Scrum

As mentioned above, we added an administrative role which impacted our work in the regard that not all people have to bother about administrative details, bringing the product owner met with our external part which went well, we have not yet received any bad feedback which is good. He seems satisfied, and look forward to our progress to come.

We have developed our own take on daily standups this. We take daily standups every meeting but not more than that so far. This have been discussed, and a suggestion is to have one more meeting, over lunch, where we just check up on each other including the daily standup. But mostly we believe it is extensive and unnecessary to have each day, and therefore we have chosen our approach. Another factor for this decision is that all people can’t work with the project every day, so the schedule does not match.

Regarding our methods to learn to use the tools and technologies so far implemented, we understood that to write code you don't have to understand everything. We have some who are great at programming which can give guidance and have concrete answers, and therefore we choose to take many of the questions arisen through them. It is in fact that simple that some of the applications are not that commonly used while coding, and to out time for all to learn this seems like a waste of time if we want to hit our targets. So the rest of the group does not really get too much into these applications and technologies, they rather focus on other technologies which are more widely used.

We also try to code in at least pairs for a more intuitive learning experience, giving a less experienced codewriter an instructor with more knowledge, guiding what to write, in what files and why. This is also a gateway to natural discussions about functionality and different implementation ways, where the guide gets insight of alternative ways it could be programmed. If not influenced, the guide still has to back its implementations with arguments, leading to a critical analysis while needing to explain what needs to be done. This has proven to be effective towards learning the technologies and the different connections between them, and we will continue to do so.

In the lecture from Husqvarna they had scheduled virtual fika-breaks. Our take on that is to have unscheduled, spontaneous meme-sessions. They can last from 30 seconds to 30 minutes, depending on the mood of the group and our state of mind, including how many hours we worked and the level of tiredness. This could to some sound counterproductive, but that is a matter of perspective. If we feel great we do great work, but it is definitely a tradeoff regarding how time-efficient we would like to be. Nevertheless, these meme-sessions is definitely a lifesaver sometimes, and for long meetings also inevitable. The downside is that a 2 hour meeting could become 3 hour long.