Team Reflection w7

Sony Walkman WM-FX 199

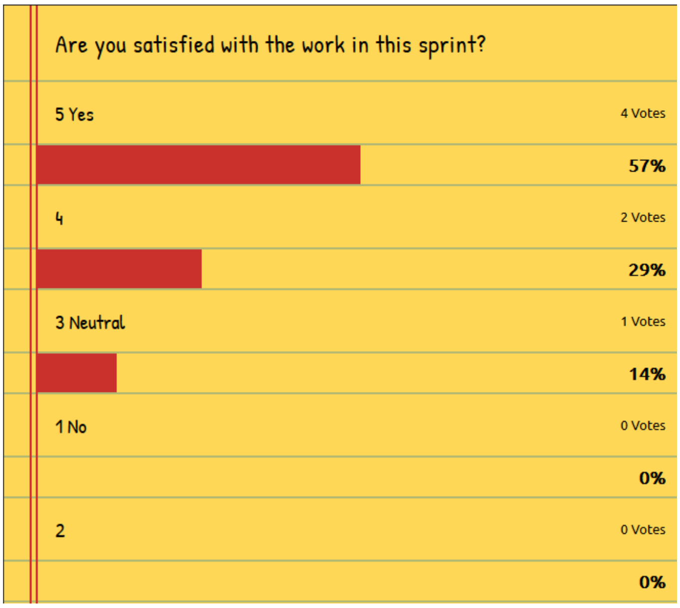
Customer Value and Scope

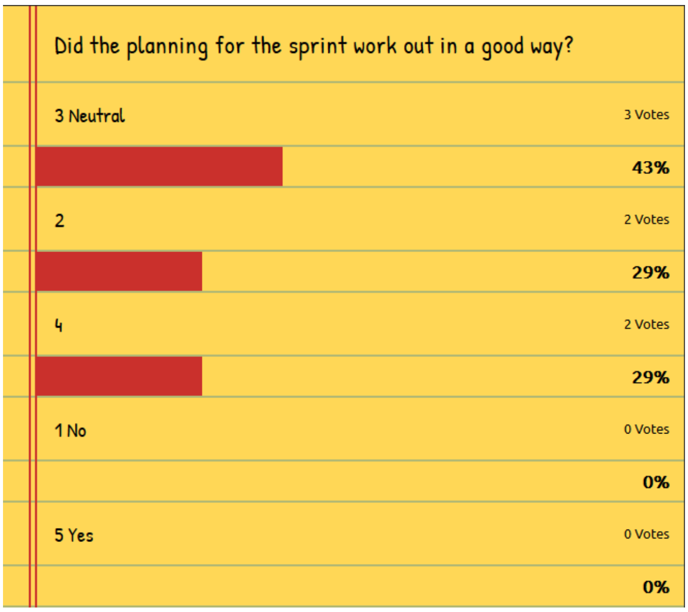
- the chosen scope of the application under development including the priority of features and for whom you are creating value
  - We have narrowed down our scope since the project started, and feel that we have now almost finished our MVP. For the coming weeks we will improve on what already exists and some extra features that would be nice to have, but were not part of the MVP scope. The feature that seemed most important to our PO was to be able to send the invoices in some way to another person, which we implemented recently.

- the success criteria for the team in terms of what you want to achieve within the project (this can include the application, but also your learning outcomes, your teamwork, or your effort)
  - Since we have limited the scope of the project to a working MVP with some iterations the success criteria has subsequently been limited as well. Some conflicts in workflow appeared this week due to unspecified tasks leading to double work. For the upcoming weeks we will aim towards being better coordinated in the Trello-board.

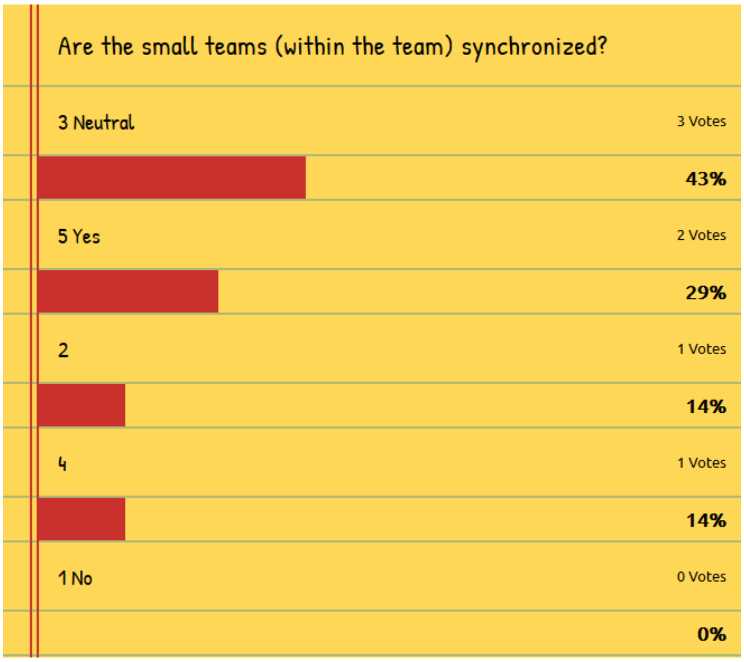
- your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation and how this influenced the way you worked and created value
  - A few weeks ago, we started estimating the sizes of the tasks as S, M, or L, which we found easier than to estimate an exact amount of hours. This estimation was easier to use and actually worked out quite well. This week, two out of three sub-teams however finished a little bit faster than we expected, which is probably a consequence of that the group has started to get used to the way of working, but also that we started to learn the programs/tools we are using which makes it possible to complete the tasks faster. To manage to get the estimates more precise, we figure the best way to get there is to practise more, to actually overshoot one week, undershoot the next, and as time goes by, we might be able to be more right on target.

- your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders
    - Every pull request should be reviewed by at least 2 team members to ensure that the task has satisfied the acceptance criterias. Apart from that we are not using any other explicit acceptance tests within the team.

- the three KPIs you use for monitoring your progress and how you use them to improve your process
    - 

    - Since we were close to finishing the MVP this week, there weren't many tasks left to do, so we all completed our tasks during our sprint leading to satisfaction.
    - 

    - It seems that most of the group is not completely satisfied with our planning this sprint. We have discussed the reason for this among us and we will in our upcoming sprint devote a bit more work to planning and also try not to rush through it. This sprint some teams were already done with their tasks in the middle of the sprint and we had not planned for this, leading to those teams not having anything to do, which could have been prevented if we had spent more time and put more effort into planning the sprint.

- 

- This sprint, the two frontend teams started to collide in their work, meaning that they were working on the same task independently without knowing that they were both doing that. The backend team also started working ahead of what was planned during the sprint-planning-meeting. This is not a huge problem, but it is one that can easily be avoided, so we should do that. To prevent these sort of things from happening, we will, during our daily standups, also tell the team what we plan to work on during the remainder of the sprint to give heads-up to the other teams what they probably shouldn't start working on. Also, if a team runs out of tasks, or decides to take on a new task during a work meeting (not with the whole team), then they will add that task to in progress as well as notify (through discord) the whole team that they will now start working on that task. We hope, by doing this, there will be no double work and we can *slice the cake*™. This was added to the social contract 7/5.

Application of Scrum

- the roles you have used within the team and their impact on your work
  - We have one scrummaster, and three teams that work on different areas of the project. Splitting us up into teams has been a successful decision, we can get a lot done in a sprint. The scrummaster is usually also the leader of each meeting which makes the meetings effective and goal-oriented.

- the agile practices you have used and their impact on your work
  - We have a meeting with our PO every week where we get feedback, and then we adapt our workflow and priorities according to the feedback given to us.
  - We have daily standups where we show what we did the day before. This helps with synchronization with the rest of the group, so we can <u>slice the cake.</u>
  - We have a planning meeting every monday where we plan our sprint together. We then select the user stories to work on which creates a focused, goal-oriented working environment.
  - After we have selected the user stories that we will work on during our sprint, we have meetings in our smaller teams where we define tasks for our specific team. Then the teams book in meetings during the week and work independently. There is however, low to no risk of not <u>slicing the cake</u> since we have daily standups to synchronize.
  - If we could get perfect information from our PO regarding what he wants us to do and which are his acceptance criteria, we would know exactly what to do. If we also had much more knowledge about programming, we wouldn't need to conduct a lot of research before we can start coding.

- the sprint review and how it relates to your scope and customer value (Did you have a PO, if yes, who?, if no, how did you carry out the review? Did the review result in a re-prioritisation of user stories? How did the reviews relate to your DoD? Did the feedback change your way of working?
  - The reviews are divided into two separate tests where one we conduct internally and another one we conduct with the PO as the final stage of our DoD. After our last review with PO, Oscar, he revealed that we had some unnecessary components in our product, which we have now removed. There hasn't been any re-prioritisation in our latest review, he still wants us to get the whole invoice-component done before we begin on something new.

- best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)
  - We have used Trello for our scrum board, and we use SpringBoot in VSCode/IntelliJ to develop backend and we use ReactJS in VSCode/Webstorm to develop frontend as well as axios to connect frontend to backend.

- At first we did some research on how these frameworks work, but after that we have used the classic “Trial and Error”-framework to learn pretty much everything. This is maybe not the most effective way to learn, but we learn a lot through it so we are satisfied with this way of learning. Some people in our group also have more knowledge than others, and they have shared this knowledge with the rest of the team during sprints, which has helped a lot. In the future, it would be optimal if dont need to watch hours of tutorials before we start coding, but instead just discuss how we should code. To get to this, we probably just need to code more and learn more, and working in smaller teams where you get some sort of expertise in the area your team is working on enables us to reach this goal faster.