

# Reflection

This week the group focused on finishing up all current features on the app and a lot of debugging has been done. The navigation to new fragments and recent fragments was the main focus. It seems that the application is ready to be shipped.

## Customer Value and Scope

- **the chosen scope of the application under development including the priority of features and for whom you are creating value**

The chosen scope of the app this week has been to tie together all the loose ends. This means to finish off the overall look and experience for our stakeholder, within the scope we delivered within the timeframe.

- **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)**

Success for the majority of the team has been learning outcomes, as in learning how to navigate and work effectively and independently in Android Studio as well as learning how to apply SCRUM. The success criterias were to not only be able to implement a certain functionality but also to be able to understand it thoroughly.

Success criteria for the application has been to be able to rate a medium and be able to view it afterwards. During the project the success criteria has slightly changed from the beginning. It went from three types of medium (movies, series and movie series), to only one as our top priority. Therefore we believe being able to rate a movie is a success for the application and the other two mediums could be added in future development instead.

- **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**

User stories gradually became less detailed. As everyone got a better understanding of the expected definitions, less time was spent on defining them. The task breakdown did really make it easier to know what to do and when to do some stuff. It also made us, the developers, more thorough in our work and minimized the risk of forgetting important parts of a user story. Every defined user story had a velocity, but the team didn't fully take advantage of the velocity concept. However it didn't influence how we worked.

- **your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders**

The acceptance tests have mostly been performed by trying out the app and the specific features, trying different scenarios to make sure that it's very unlikely that a bug will occur. This was done by the developers themselves, as they know best how the features are supposed to work. On top of that, the stakeholder tests the app every week which gives a different perspective, focusing on if things are working in an intuitive way. We believe that with these two perspectives we have great coverage of preventing bugs and making the application run as it is expected to, and this is the resulting value of our acceptance tests.

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

1) How satisfied is the stakeholder:

The stakeholder was satisfied with our product and he thought the carousel at the home screen was a nice feature! He thinks the search bar is good, despite some minor opinions on the "hitbox". Overall the stakeholder was very satisfied.

2) Percentage of finished commitments: we started with 19 user stories and finished 16.

3) How many stories were finished: We finished 16. However, the other user stories are nearly done. This has been the final sprint so technically they will not be finished on time, but theoretically we could have finished them if we increased this week's sprint to Sunday, but we are too tired to do so.

## Social Contract and Effort

- **your social contract, i.e., the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project**

The group did follow the social contract in the best way possible this week. We did meet twice as a full team and we did communicate a lot on different platforms. Nowadays the team is better at communicating whenever obstacles appear.

One thing that has changed during the project is meeting protocols. We decided that we wanted to have two meetings a week that would be documented by protocols. We did this in the beginning but we later realised it wasn't that effective and that our team reflection and scrum board was enough documentation.

- **the time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)**

On average the team has spent around 15-20 hours per week per person. Some sprints when team members have been absent a lower sum of hours have been spent on the course. Usually this has been compensated by putting in more hours on other sprints when a lot of bugs appeared and when we took on a lot of stories. Some weeks our deliverable has been less than others, not because less time was spent, but rather that more problems were encountered which made certain user stories take longer than expected.

## Design decisions and product structure

- **how your design decisions (e.g., choice of APIs, architecture patterns, behaviour) support customer value**

Our design was built mostly for our stakeholder to which he was a very big part off. The stakeholder choose many design choices to which we needed to adapt to and implement. We tried to implement the mvvm pattern so that the lifecycle of the application is extended and the stakeholder does not only get a product but also a high quality application. We have used also singeltons to access a single object and also tried to make as many interfaces as possible. Something that came up is also calling the api which was done on the main thread which is not good because if the api call fails the application fails. So what we did was to implement it in a parallel way so that the main thread still runs if the api fails and also make it faster.

- **which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)**

We have used and tried to be consistent with writing documents in drive and however class diagrams, models and component diagrams have not been our priority. However our trello where we have our user stories have been consistent.

- **how you use and update your documentation throughout the sprints**

Whenever we have completed or come a very long way with our task/user story we make sure our trello board is updated and also our github has all new implementations and information needed. Sometimes we have updated user stories with extra tasks when things have come up in the middle of a sprint, also sometimes noting down bugs related to the user

stories, so that anyone in the team can take a look at the problem without having to ask what problems exist.

- **how you ensure code quality and enforce coding standards**

Our group decided early on that we would follow certain naming conventions for classes, packages, XML-files, etc. :

- Capital letters for classes
- lower case letter for packages
- lower case letters for XML-files and underscore between words

Although these guidelines were put in place at the beginning of the course, sometimes it has not been applied. Therefore some variations exist.

## Application of Scrum

- **the roles you have used within the team and their impact on your work**

Jenny was the product owner. This week Ida was our SCRUM-master. Ida sent out daily SCRUMs to see how team members are doing and to see if any obstacles have appeared. Jenny gathered some input from the stakeholder during the week which was useful for the development of the GUI. Other practices like reviews, retrospectives, planning meetings and such were the same as the last weeks.

- **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)**

Same as previous weeks.

- **the agile practices you have used and their impact on your work**

Same as previous weeks.

- **the sprint review and how it relates to your scope and customer value**

Same as previous weeks.

- **relation to literature and guest lectures**

Nothing new this week.