TEAM 20, SPRINT 4

Team members: Rebecka Jakobsson, Cecilia Michelsen, Jacob Messinger, Tina Samimian, Elin Nilsson, Linnéa Fransson, Norbert Laszlo & Oscar Helgesson.

Customer value & scope

This sprint we had some problems with the estimation of the user stories. Half of the user stories were properly estimated, while the other half took longer than planned. Especially the user story that was supposed to achieve the search function in the company list. It turned out that this function was more complicated than expected when we wanted to connect it with our search method and fragment. Another coding pair had two user stories which in total achieved correct estimation, but the user story we thought was the easy one turned out to be more complicated than the other one, which was easier instead. Inaccurate estimation may be because we are quite new to Android Studios, which makes it more difficult to estimate the user stories.

To achieve a minimum valuable product, we have listed the most important and central functions that create value for the customer which are:

- Different views for different types of information, relevant to the students
- As a user, be able to “save” companies
- Show all companies and make it possible to filter and search between them

Being able to filter among the companies is the only feature that remains to be implemented. That being said, we have reached our MVP when this function is implemented. We estimate that this will be done in the last sprint and that we then will have reached our goal.

KPI:s

Effectiveness: Estimated number of invested work hours VS Actual number of invested hours for each sprint.

This sprint's estimated total time was **26,5h** and the actual spent time this week added up to **37h**, it gives us a percentage of **140%** which we consider as the borderline for "Ok" rating in our Effectiveness KPI. Some of the User Stories this week took a lot longer to finish than planned. As we dig deeper in special Android Studio functionality more difficult tasks emerge, making it really difficult to estimate time, as sometimes there might be a Stack Overflow post describing exactly your issue, and sometimes you have to try different solutions for several hours before finding a solution. For example, one group was assigned two User Stories and we estimated one to be a bit easier than the other. The actual time spent was considerably less on the one we thought would be more difficult and a bit more time than estimated was spent on the "easier" one.

Productivity: Total score of points by completed user stories in a certain sprint vs. estimated total score one sprint.

This week we estimated the user stories to totally cover **27** points and we finished **22** of them. One of the user stories was estimated to take one-quarter of the actual time it took to finish it. Therefore the coding pair that was responsible for that user story and another one didn't have the time to start on the second one. Except that we almost finished all the other user stories and got all points allocated to them. As a result we had **81 %** coverage which means that our estimation is considered "Good" in terms of our Productivity KPI.

User Satisfaction: A measurement of satisfaction from our stakeholder.

1. Does our work meet the user's expectations on functionality?
2. Is the user interface convenient?

Our stakeholder gave us 8/10 for the functionality. She considered the search function to be fast. She liked the fact that the user will be able to mark a company as a favourite from the company page. As for the user interface convenience she gave us 8/10. Her comments were that she liked the size of the search function and that it was easy to find and use thanks to that. She liked the size and placement of the heart that the user will use to mark a company as a favourite. She had some comments about the part that shows what the company is looking for and what it offers. In her opinion it was a bit unclear what was what. Also she wanted an added company logo to the company page. On the “my page” page she wanted a bit more structure and clarity of what list belonged to what title.

Social contract and effort

One of our goals this week was to review our social contract and update it based on our experience during the project. We still kept most of the contract but made a few changes. One point in the social contract was that we wanted to appoint a president and a secretary for each meeting. This is something we haven’t done, this task has instead fallen on the appointed scrum masters for each week. We also added a definition of the scrum master and product owner in the contract to clarify what they are supposed to do.

As in the previous sprints we’ve continued to work in programming pairs but this sprint we’ve started to work cross teams for some tasks that proved to be more challenging than expected.

The time each team member has spent on this project during the previous sprints lives up to our initial requirements that we specified in our social contract. Some sprints we would have liked to accomplish more in the timeframe we had planned for but we still delivered sufficient progress to satisfy both the groups and the product owner's expectations. So far we’ve reached the goal of presenting a noticeable progress in the apps functionality and look for every sprint which has held us on the set timeline (with some deviation) that we’ve

planned. Even though we've liked to accomplish more in some sprints that we managed we're still on the path of completing our MVP in the timeframe that we've planned.

Design decisions and product structure

As a team we have made the decision to use MVVM (Model-View-Viewmodel) as our architectural design framework. We have heard from previous courses that using MVVM is beneficial when developing in Android Studios but it also provides value to our customers. Using the MVVM patterns makes it easier for both us and our stakeholder to understand, maintain and modify the code base. Furthermore, using such large architectural frameworks makes it easier for us as developers and the code base in general to follow the SOLID principles when coding. Another added benefit of using the MVVM patterns is that it allows us to write Unit tests on both the Model and Viewmodel. This of course helps ensure code quality and value for our stakeholder. However, this is the first time any of us have used this pattern and it has required effort from our part to make sure we are following it to the greatest possible extent. The pattern isn't just plug-and-play but we have to actively pursue it, but this has become a lot easier the more we have used it.

In the last team reflection, we made a list of three criterias to the Definitions of Done (DoD). All of the coding pairs have been trying to work based on DoD during this sprint. The team has increased the amount of documentation and it's quality. This made it easier to understand the work we have done. We have also been checking each other's code, to ensure good quality of it. The criteria of using unit-test for java code is an area of improvement. We have been testing some of the code, but far from everything. This will be every team member's responsibility to use in the next sprint. The use of unit-test will be followed up in the next team reflection.

Application of Scrum

Our application of Scrum has not changed from the last sprint. Therefore we don't have anything to add in this reflection.