

Individual Reflection

Alexander Neldefors

23-10-2019

This week I have helped the team with connecting all different parts of the application as well as adding the finishing touches to the project (like finishing low priority user stories that have a low estimation). The problem was that because we hadn't connected parts of the application before (like connecting search and result as well as connecting our two apis) we encountered a lot of bugs which kind of broke the develop branch, as code with bugs were uploaded there (instead of a separate branch). The problem was that the app was too reliable on Google Geoservices and the app would not work without it (couldn't search trips and app froze as the request to was on the UI thread instead of a background thread) which it should. That made it hard to work vertically because to make to application testable (manual tests) you had to use an older version of the app which then made it hard to merge with what other people had done because some branches were outdated and so forth (probably the result of not communicating as good as we had done in the other weeks). So this week was basically tracking down bugs and eliminating them, merging without causing other bugs or app crashing problems as well as finishing the app.

What I could and probably should have done differently is to help the team tracking down bugs before finishing low priority tasks.

What I have learned this course that I can take with me to future projects are for starters not to plan out the entire domain and design model during the first sprint. That made it so we couldn't work vertically because all other tasks were too reliable on the model. We also made implementing the model one task which were taken care of by one member making it so the rest of the group couldn't start until the model was done. What we could have done instead was creating the model for that sprint (accommodating it for the user stories that sprint) and then adding to it each sprint. So basically making sure that no task or user story is too reliable on another one so you can work vertically and avoid conflicts, although decisions on the project architecture, framework and coding/testing/documenting standards etc. should be decided before you start coding. I also learned that consistently merging is important, this because it will be a pain to merge if everyone has done too much of their own on each branch resulting in a lot of merge conflicts. I also learned that you always should have a testing protocol which should be followed after every merge to avoid bugs or app crashing problems. Communication is also very important especially if you start working with parts of the app that might affect someone else in the group, you should ask if it is okay to do that or agree on how to do it. That was a problem, because we didn't properly communicate that which made us solve the same problem in different ways or break the code for another member in the group (could probably be avoided if the user stories or tasks are not reliable on another one). You should also start breaking a task down into more tasks if that task starts to grow or if it is bigger than expected so you at least can show that you finished some tasks instead of you being stuck on a really huge task. Another thing is that everyone in the group has a different velocity when it comes to finishing tasks and that instead of taking on a new user story (that is not in the sprint backlog) help others in the group to finish their tasks (comes down a lot to learning how to prioritise correctly).