Individual reflection DAT256

Adam Andersson

Week 9

What do I want to learn or understand better?

A:
This week mostly consisted of wrapping up the project i.e. working out bugs and fixing TODO:s in the code and therefore not much new was learned during this last week except maybe how you make a project presentable in before a deadline and presentation.

B:
For a future project I would like to learn more about how firebase is setup since during this project I started using the functionality after the implementation of it.

A→B:
In order to be able to implement firebase from a starting point in a project I would have to read up more on how the basic and obligatory parts of the setup is meant to be done, after that I think it would be valuable to try my hands on doing it on a test application i.e. create an application and connect a simple fire database to it.

How can I help someone, or entire team, to learn something new?

A:
Same as earlier sprint weeks.

B:
For a future project I would like to keep being an active member of the group and be open to discuss or help members of the group with their problems in order to try and mediate my experience and knowledge on the subject. I would also continue to be active during meetings and speak my mind since more ideas and takes on a part in the project helps to refine it. I would also like to put more effort into looking at peoples pull requests in order to achieve greater code quality as a whole but also to provide thought trough comments to the writer of the code so that they can explain or change their decisions, hopefully for the better.

A→B:
By continuing to be active and contactable and tell the group that if they want to ask something or just brainstorm they´re welcome to ask me anytime. By trying to convey a friendly atmosphere during meetings were everyone feels like their opinion matters. Being more careful when reading pull requests and taking notes would give a more detailed review in the end that I could then discuss with the writer.

What is my contribution towards the team's use of scrum?

A:
This goes along the same lines as earlier weeks. Being active during the meetings in discussions regarding customer value, velocity, sprint planning etc. following our git routine and social contract.

B:
During this project I didn't take on the role as the scrum master so for a future project I would like to try this role, preferably during a substantial time period to really be able to learn how to duties of a scrum master is performed.

A→B:
Volunteer to be scrum master at the beginning of the project and then try to carry out the scrum master duties as seriously as possible to make to experience realistic.

What is my contributions the team's deliveries?

A:
At the beginning of the week I added functionality for calculating and adding CO2points to the passenger and the driver when a trip was finished, I also did some general code refactoring. At the end of the week I had to fix a bug that was derived from the CO2points functionality I created earlier in the week were to calculated CO2 points was always zero.

B:
For a future project I would like to be more active during the start up of the project in order to complete more tasks related to building the foundation on which the rest of the app rests. I would also like to try my hands on bigger refactoring tasks since for this project I mostly did tasks that forwarded the functionality rather the refactoring already written code.

A→B:
To be a bigger part of the foundation of the application I need to be more active during the start up, sadly for this project I was studying for re-exams at the start and therefore didn't get to take part in the early creation of the application as much as I wanted to. When it comes to taking on bigger refactoring tasks I think it boils down to having the right mind-set and systematically start refactoring so that it is done through many small iterations. Using that method I think the intimidation of changing behaviour and creating code errors that has a big risk of coming along with a big refactoring will decrease. Using tests to make sure that code behaviour hasn't changed is also an aspect that could and should be used.