21/9-2021

- Discuss KPI in the next sprint meeting7
- Remember team reflection and individual reflection
- Pull requests on tasks or on Userstories?
- Molly new SCRUM master
- Update DoD for testing

Sprint review:

Erik: created a sample application and finished it. Started to work on the camera implementation, realised it was a lot to work with android permission. Asked if anyone else had issues with it and the rest of the group shared that it was also a problem. Erik created a permission handler and created a new user story and finished it. It brought value to the team. Continued with user story 14 which is almost finished. Requested a PR for it.

Daniel: Location handler used to get location of the device. Might not be as useful for OSM as Mapforge.

Valeria: worked with notifications and a drop down menu in case there is a new challenge for the user. User story 7.2. Still working on perfecting it

Joakim: working on 3 classes to get the notifications to calculate the time. User object to get listed routes and getting update manu to read the last item from the list and check timestamp. New approach for long term scalability.

Molly: created and finished task 3.1 with creating the map.

How/when will the remaining tasks be finished?

Daniel will continue implementing code for map location today, tasks 3.3 and 3.4.
Joakim will need to do some testing.
Valeria will need to merge work with Joakim. Valeria and Joakim is not coordinating as well as they could?

What went well this sprint?

Everybody has been making progress on their user stories. A lot of them are all finished. The workflow was good among everyone. Good cooperation and response between team members. Daily reports were good to see if anyone else had the same questions. It helped keep the group “alive” and not be dormant.

What could've been done better?

The questions can be handled better, it's easy for them to get forgotten when tehyre in their separate text channel that is dedicated for only Daily Updates.

The team reflections could be handled better. Miscommunications from teachers when it should be done and confusion in the group.

Discussion on how to set up Java code. Right now we're adding classes whenever and wherever we need. Should we create a design for our code, like Model,View,Controller? Each team member could plan their design, then merge with the rest? The diagraming/planning conflicts with the SCRUM workflow. It would be good to have a general idea on how we structure things

- the chosen scope of the application under development including the priority of features and for whom you are creating value
    - Scope already in Canvas.
    - Priority of features: taking pictures, showing a map, creating destination.
    - Priority of features: In GitHub
    - For who are we creating value?: Already in Canvas
- 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 criteria for the team: Better understanding of the Agile work process
    - Success criteria for the team: Coordinating skills within a group
    - Success criteria for the team: Create an app that tells the user to walk to a certain point on the map and that the app is able to take photos and save them and show them to the user
- 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
    - The standard pattern gives us some general understanding on what to create, the acceptance defines it more clearly, helps the group understand when something is finished.
    - Task breakdown and effort estimation helps the group coordinate our work and avoids mistakes and encourages communications. Large user stories encourages work together as teammates
- your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders
    - Every code reviewer for the pull requests checks that the task meets the acceptance criteria, and that the code is runnable and is compatible.
- the three KPIs you use for monitoring your progress and how you use them to improve your process
    - The group hasn't used any KPI's so far