Reflection: Week 1

We have created a github repository for our project as well as a discord channel where we can discuss and coordinate ourselves.

TODO/TOConsider:

- Schedule a meeting before holiday.
- For project: include file linking to doc of each API used (and person who added API)
- Reflection in PDF:s in weekly folders instead of README?

Below we will reflection points from the given template:

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

Not relevant yet.

- your social contract, which means you should create one in the first week

----------SOCIAL CONTRACT----------

- Our ambition as a team is to work on a medium level with max 20h/person spent per week.
- We want experience working in bigger projects and hopefully make some mistakes which we can learn from. To have an ongoing, unified, organic and fun development process which aims to create a product that is relevant.
- We also aim to further our understanding and skills with API's, KPI's and tools such as github and fello.
- We will have at least one physical meeting per week with the whole group where we plan/brainstorm/reflect.
- Since 8 people we will have to work individually and try and sync and organize our work with help of internet platforms.
- If someone wants to work on their spare time it's their responsibility to update the team of any changes.
- Details about the contract needs to be discussed further. The contract is not disclosed yet.

- the success criteria for the team in terms of what you want to achieve with your application

Our goals during this project is to create a functioning application, in sync with the customers requirements. We want to learn how to develop the application as a team unit, using techniques like scrum and agile development. During this week we've seen how challenging upholding a coherent collaboration within the group can be. This is why we feel using these processes can be valuable.

- your acceptance tests, such as how they were performed and with whom

Not relevant yet

- the design of your application (choice of APIs, architecture patterns etc)
- the behavioural overview of your application (for instance through use cases, interaction diagrams or similar)
- the structural overview of your application (such as class diagrams, domain models or component diagrams)

Not available yet

- your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation

At the Scrum exercise we used some standard patterns, including velocity approximation for the different user stories, maximum velocity for a single sprint and task breakdown.

- the three KPIs you use for monitoring your progress

It's hard to imagine what KPIs will be relevant but we thought of some that we might consider. We could measure bugs/time passed with FireBugs to get an estimate of our technical cost. We could us a burn up to get an overall view of our development

progress. We also think that it would be good to have some kind of customer satisfaction to get an outside perspective of our progress, but we don't know how to achieve this yet.

- code quality using a tool such as Findbugs (1 point if your code includes issues concerning correctness or bad style, 2 points if you have dodgy or performance issues and 3 points if the code is fine), only asses the code you have written yourself

N/A

- the roles you have used within the team

In the lego assignment, we assigned one scrummaster to handle scrum master tasks, including: negotiating with the product owner and other group scrum masters. Other people were assigned different tasks, like fetching lego-pieces, but these roles were not static. In the Kata exercise we had a documenter, puzzlers and brainstormers. We realised that not having static roles was helpful in these exercises just to try things out and get perspective on things.

- the agile practices you have used for the current sprint

Doing both the Kata and Lego exercise we used agile development by working in smaller iterations, we also worked flexibly by changing roles and priorities as we went on. One big challenge was that we, as a group, had to communicate with other groups to make the pieces fit together as a whole.

- the time you have spent on the course (so keep track of your hours so you can describe the current situation)

So far the time spent on the course includes lectures and exercises, plus one meeting regarding reflection about the first week. 10 hours on lectures and 2 hours of reflection in two separate groups due to scheduling difficulties. Some additional hours can be spent individually to learn technical stuff like github or API's, or just freshen up general coding skills.

- the sprint review (either in terms of outcome of the current week's exercise or meeting the product owner)

The simulated review at the Scrum exercise gave a good idea about how it's done in the real world. The products produced during the sprint were reviewed by the product owner and we got feedback on weather improvements were needed or not. We learned that by having a dialogue with the PO we could both negotiate and reevaluate exactly what they wanted and what we could deliver.

Reflection Kata-exercise: From the Kata exercise we solved a puzzle in 6 iterations and measured the time spent. We tested different strategies but we learned that it's best to make small changes to the process in order to be able measure the results. We also learned the importance of good documentation, e.g. to get outsiders to understand the process. It's important that everyone in the group gets a say and participates in the project.

Reflection Scrum-exercise: In the scrum(lego) exercise we learned to use some standard patterns: velocity approximation for the different user stories, maximum velocity for a single sprint, task breakdown, product owner negotiation, backlog importance (which user stories to focus on), sprint review and retrospective.

- best practices for using new tools and technologies (IDEs, version control, scrum boards etc.)

Everyone agrees that a familiarity with git is crucial for this project. We also need to use tools like Trello, or other sprint boards, in order to organize our project structure. These tools are helpful for coherent collaboration going forward with the project.

- relation to literature and guest lectures (how do your reflections relate to what others have to say?)

Many of our conclusions is based on information from lectures and slides, or what we learned during the exercises.