

# Team reflection

## Customer Value and Scope

- **The chosen scope of the application under development including the priority of features and for whom you are creating value**
  - **A:** Feels good, this week has mostly been us getting started so we don't know our velocity good enough yet to tell if the scope is reasonable. We are currently focusing on developing quintessential parts of the product, while exploring the news technologies we have to learn. These important parts include views, basic navigation in the app and basic communication between phones through firebase.
  - **B:** Ideally we want our scope to be just right. Something that the team can accomplish but doesn't push them over the 20h/week ambition. We also want to provide features in line with our clients priorities in an efficient manner.
  - **A → B:** We primarily need to get comfortable with the technologies we are using. That way we can get a better sense of what actually might be sensible to ambition for, and what might actually be harder/easier than expected.
- **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)**
  - **A:** We have a fairly good idea of what we want the finished product to be like, but it will be refined during each sprint. Success in terms of the application would be getting it to a "finished" state that provides the values we have intended. In terms of learning outcomes, we of course want to learn scrum and the different tools we have used. We also keep track of our individual learning goals each week.  
Success in terms of teamwork is to maintain a good mood among the group members, make sure that everyone works efficiently and feels that they have a balanced workload.
  - **B:** We want a concrete and well defined success criteria for the application, learning outcomes and teamwork.
  - **A → B:** We will discuss during an upcoming meeting how to refine the success criteria. They will probably also naturally get clearer as the development progresses.

- **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**
  - **A:** We have spent a lot of time breaking epics into user stories, and further subdividing them into tasks with acceptance criterias. We have not estimated effort beyond creating a backlog that we all agreed was reasonable for a single sprint. However, we plan to estimate all issues on Monday next week as we'll have a better understanding by then about the technologies needed to solve these issues, so that we can estimate and measure team velocity later on. It has helped us split tasks between group members to get started and work more efficiently.
  - **B:** We want to be able to accurately estimate the remaining work and use that information to further improve efficiency and optimize for value during each sprint.
  - **A → B:** We will begin estimating effort for all tasks and keep track of our velocity.
  
- **Your acceptance tests, such as how they were performed, with whom, and which value they provided for you and the other stakeholders**
  - **A:** We haven't interacted with any stakeholders as of now. The idea as we understand it is that one of us is the product owner, but everyone should also be developing the app. So we solve this by enforcing that PR:s are to be reviewed by group members who haven't helped out on that task, and thus simulate a third-party reviewing the supposed feature that PR brings.
  - **B:** Ideally we want a clear idea of how our relationship with the product owner actually works and with that a way to clearly define what values a feature really gives.
  - **A → B:** We will ask a lot of questions regarding this on Monday, and try to get a clear picture.
  
- **The three KPIs you use for monitoring your progress and how you use them to improve your process**
  - **A:** Currently, we have not decided on all KPI metrics that our team wants to use. We have implemented build status with tests/linter/building for our code using Travis CI.
  - **B:** Having and making use of KPI metrics to be able to evaluate the teams velocity and productivity, and also enabling us to discover and eliminate potential bottlenecks that unnecessarily slow down our workflow.
  - **A → B:** The team needs to meet and decide on KPIs that are reasonable for a project like this and the Scrum workflow that we are assuming. This should be

done sooner rather than later so that we can utilize the full potential of these tools ASAP.

## **Social Contract and Effort**

- **Your social contract, i.e. the rules that define how you work together as a team, how it influenced your work, and how it evolved during the project (this means, of course, you should create one in the first week and continuously update it when the need arrives)**
  - **A:** We have a social contract that we wrote and agreed upon during the groups very first meeting. It was largely based on contracts from previous group projects that the members had taken part in, refined and then combined together.
  - **B:** A fully functional, tried-and-tested, completely covering social contract that aids in the teams process and provides solutions for all kinds of situations.
  - **A → B:** Continuously updating the social contract as the need arises, which so far hasn't happened. But we should review the social contracts to find points that we want to update, possibly already during our meeting on Monday.
- **The time you have spent on the course and how it relates to what you delivered (so keep track of your hours so you can describe the current situation)**
  - **A:** Team members have had less time than desired this week, and there are some learning curves to the technologies that we will be using. However, we have gotten a proof of concept for the app code, the firebase code and people are working in parallel on different issues and communicating actively on Slack.
  - **B:** Ideally we would like to maintain a 20h/week and still finish our sprint tasks along with course related documentations, and perform good results.
  - **A → B:** Get more experience with the technologies so that we can make better decision and make better estimations of problem complexities.

## **Design decisions and product structure**

- **How your design decisions (e.g. choice of APIs, architecture patterns, behaviour) support customer value**
  - **A:** We've chosen Kivy/Python for the mobile app, Firebase for the backend/database and a MVP-like architecture pattern for the app code. We have yet to get going with a map API, but plan to use Google Maps' API. As we've had to experiment a bit with how Kivy apps should be designed the specifics of the architecture isn't set in stone yet, but we should be done with that by the end of this week. We've setup Travis with a linter, tester and an app build exporter to ensure code quality and to handle deployment. We've set our main branches to protected and require PR:s to have some approving reviews and pass the Travis checks.

- **B:** Well thought-out code architecture that is sustainable in the long run, without needing major refactoring every time a new feature is implemented in the codebase. Using APIs and external libraries that are supported and maintained. Have reliable code that won't break in production.
  - **A → B:** To reduce the gap we need to finish the overall architecture for the frontend.
- **Which technical documentation you use and why (e.g. use cases, interaction diagrams, class diagrams, domain models or component diagrams, text documents)**
    - **A:** We use user stories to target and prioritize what our clients want and need. We use a design mockup to internally share a sense of the app we are building, and how we imagine features to be implemented and how the user will interact with the features we propose. We have also started documenting code using Sphinx.
    - **B:** We want fully documented code that is as clear as possible, so that any team member can understand what every part of the code does by reading its documentation. This also increases the teams truck factor. We also want to have the things stated in **A**, such as making use of user stories and their benefits and a global design mockup.
    - **A → B:** We need to carry on with everything that is listed in point A, but also need to actively document the code we are writing and committing, because if documentation is not present from the beginning, it will never exist at all (from experience). This can be aided with tools like the coverage extension for Sphinx, which measures the documentation coverage.
- **How you use and update your documentation throughout the sprints**
    - **A:** User stories are added and refined as the need arises. The design mockup is also being updated as the need arises.
    - **B:** Ideally we would like to be flexible and refine our documentation quickly as soon as we have to.
    - **A → B:** Continuously discussing how each user story as well as the mockup is relating to the project so far. If there are any issues we would like to solve them as soon as possible.
- **How you ensure code quality and enforce coding standards**
    - **A:** We have some continuous integration tools to enforce the PEP8 style guide and some other style guidelines. Our CI also runs tests, and we require PRs to

be approved by at least 2 others to be merged. And since our main branches are protected all code has to be reviewed to be merged. We could look for additional code quality tools in the future.

- **B:** No broken/untested/undocumented/inconsistent code can be merged to any main branch.
- **A → B:** We need code coverage and also ways to check for undocumented code.

## Application of Scrum

- **The roles you have used within the team and their impact on your work**
  - **A:** Anyone can select any issue they feel like, so we have no fixed roles, people have naturally gravitated towards different types of areas for now though (e.g. Martin: CI, Theodor: Firebase)
  - **B:** We might need to have someone in the group assume the role of product owner.
  - **A → B:** Who the product owner should be, and if we need to change anything, will be discussed on the monday meeting. Otherwise the current situation seems good as is.
- **The agile practices you have used and their impact on your work**
  - **A:** We have set up a slack bot for daily stand up meetings, they have been great. It's a good way of ensuring that everyone can work effectively without being stressed. It also helps to bring a good picture of everyone's daily situation and their progress. The topics that everyone answers each daily scrum are: *How are you feeling today?*, *What did you do since yesterday?*, *What will you do today?*, *Is there anything blocking your progress?*. Since it's written it also serves as documentation. We also have weekly meetings on Monday/Friday.
  - **B:** Use the input from the daily standups to better understand what opportunities and challenges we're facing as individuals and as a team, and to better understand what and how much we are able to deliver during a sprint.
  - **A → B:** By looking back on the documentation of the daily standups, we will be able to see any trends in how we function as a team and if there are any individual issues that need to be addressed to make sure that everyone have the best possible conditions to do their work and feel like they're contributing.
- **The sprint review and how it relates to your scope and customer value (in the first weeks in terms of the outcome of the current weeks exercise; in later weeks in terms of your meetings with the product owner)**

- **A:** This is the first sprint, so we haven't had any sprint reviews yet. In previous weeks we've participated in the lego and the puzzle exercise. These have helped us understand the basics of working in an agile manner and delivering value to the customer, as well as evaluating our process and improving it.
- **B:** The sprint review should help guide us on the right track to delivering the final product.
- **A → B:** We'll do our best with the sprint review when this sprint is done.
  
- **Best practices for learning and using new tools and technologies (IDEs, version control, scrum boards etc.; do not only describe which tools you used but focus on how you developed the expertise to use them)**
  - **A:** We're using PyCharm as our IDE, git for version control, GitHub issues (backlog) and projects (scrum board). Most of the group members have worked with these (or very similar) tools before and try to help the rest of the group to also develop an expertise in them.
  - **B:** Become more adept with using the tools and methodologies, so that the tools themselves are not a problem and instead help us provide value.
  - **A → B:** Since we have a lot of different experience and knowledge backgrounds within our team, there's a bigger chance that someone has worked with the particular technologies or methods before and therefore has a chance to instruct the others. This will be part of our weekly work and it will let the team members approach the "T Developer" concept over time.
  
- **Relation to literature and guest lectures (how do your reflections relate to what others have to say?)**
  - **A:** We have not looked into any course literature, but there have not been any guest lectures, so this is not applicable yet.
  - **B:** We should think about how we can incorporate any guest lecture or literature knowledge into our workflow in order to become more efficient.
  - **A → B:** Since there are no scheduled guest lectures that we are aware of at this time, we will have to bring our own knowledge. The course literature would probably help too, but we currently think that our time may be better spent by putting it towards the project and applying scrum in practice rather than reading about it.