# Week 1

**1. The chosen scope of the application under development including priority of features and for whom you are creating value**

* Not possible to reflect upon yet, will be possible when we’ve received the project instruction

**2. Your social contract, which means you should create one in the first week**

* The social contract should be a guideline for the groups' work
* At the moment the social contract has been established and works just fine for us
* It's important to constantly reflect upon the social contract in order to coordinate the work within in the team in a better way

**3. The success criteria for the team in terms of what you want to achieve with your application**

* Not possible to reflect upon yet, will be possible when we’ve received the project instruction

**4. Your acceptance tests, such as how they were performed and with whom**

* Not possible to reflect upon yet, will be possible when we’ve received the project instruction

**5. The design of your application (choice of APIs, architecture patterns etc)**

* Not possible to reflect upon yet, will be possible when we’ve received the project instruction

**6. The behavioural overview of your application (for instance through use cases, interaction diagrams or similar)**

* Not possible to reflect upon yet, will be possible when we’ve received the project instruction

**7. The structural overview of your application (such as class diagrams, domain models or component diagrams)**

* Not possible to reflect upon yet, will be possible when we’ve received the project instruction

**8. Your user stories in terms of using a standard pattern, acceptance criteria, task breakdown and effort estimation**

* Not possible to reflect upon yet, will be possible when we’ve received the project instruction

**9. The three KPIs you use for monitoring your progress**

1. Burn up chart (accepted velocity vs points taken)
2. Defects/bugs in the codebase
3. Loc vs velocity, uppdelat per språk

**10. 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**

* No code has been written yet

**11. The roles you have used within the team**

* It’s important to create a good role distribution in order to balance the work time among the group
* At the moment, the scrum master role will rotate among three individuals of the group
* Each group member will review another group group member’s code

**12. The agile practices you have used for the current sprint**

* We have had a reflection on the first part of the project.

**13. The time you have spent on the course (so keep track of your hours so you can describe the current situation)**

* 12h (lectures + meeting + individual reflection).

**14. The sprint review (either in terms of outcome of the current week's exercise or meeting the product owner)**

* We have not had a review yet.

**15. Best practices for using new tools and technologies (IDEs, version control, scrum boards etc.)**

* We will be using git for version control and github as a repository.

**16. Relation to literature and guest lectures (how do your reflections relate to what others have to say**

* We’ve reflected upon the scrum-exercises in a separate document.
* We have nothing further to say on this point yet.