Refl. Team

When working with small projects, it could be easier to work individually. You will have a good overview of the project since you have written all the code. That would be a great advantage, since you know how every class should interact. While it works better in small project, it is more efficient to work in teams when the projects are bigger. It would take more effort and time for one person to make it to the goals than it takes for a team mostly.

Assign tasks
One thing we thought was important was to make sure at the beginning of the sprint that everyone had different tasks. Same people should not work on same classes most of the time when writing them from scratch. As a group of four, it was not as complicated as it could be with assigning the tasks, though sometimes it was hard to get a clear overview of the tasks that was taken. It would have been more efficient if we would just check the tasks that was taken by someone in the team.

Misunderstanding and communication
Something that occurred more than one time was misunderstanding and lack of communication. When everyone is working on different classes, we must sooner or later piece them together. This requires much communication between the parts that have classes that should be pieced together. This was not a critical flaw in our project since every functionality was defined before the coding, so the problems that occurred was only minor. They were taken care of through communication between the both parts. Even though it was not flawless, we think that these kind of misunderstandings are common in every project.

More knowledge
Even if it is harder to have some kind of structure when there are more than one in a project, it also means that there is more knowledge to share. Since everyone in the project is reachable, if anyone gets stuck, they could always ask if any other has the knowledge. People are good at different kind of things, so it comes natural that someone may write or sometimes even take over a class if the person is good at it.

Refl. Documents

When documents were created, everyone is involved. We have special times (like now) where everyone in the group is creating documentation. We also have all our documents in Google Drive, giving us real time updating. Right about now we are all working in this document with different parts.

We think Google Drive is almost the perfect way for us to collaborate, even though there are some things we miss. There is really nothing to complain about when in a document and writing together; everyone sees the same thing and it is easy to write and discuss at the same time. You can add comments and mark different parts of the text which is instantly visible for the others in the group, there is also a chat function which is quite good when we're not sitting together and work.

To the parts which aren't so good. When creating a document you always have to invite the group one person at a time, which can be quite annoying when making lots of documents. It would be better if we could automatically share the ChessFeud folder we all have created. Another thing that could be better is that now you have to export the document to a PDF and then move it to GIT where the updated version is pushed. It would be better with some kind of autosave where we don't have to download all documents and compare them before pushing.

As stated above Google Drive makes it easy for us to write documents though the disadvantages is that we only push the latest version to GIT right before a hand in or release. We always have the latest version here but we're a little bit too lazy to put updates in GIT.

Refl. Software Engineering

In this project, we worked with the scrum technique. It’s an agile process which are supposed to let us spend less time in documentation and more time working. We believe that scrum is a very good method to use. However, it felt like we knew too little about it. We really did not know what we were supposed to do in all the situations.
We learned about the different parts of scrum, but when we created our documentation we had some troubles to know what actually to write. We think that it would be better with some examples of the different backlogs as well as the sprint planning document. This would be easier if you have any experience with it before though, which none of us had. It is also a bit hard to have such short time when the sprints only last for one week, because most examples we found to learn from had much bigger sprint- and product backlogs.

One more thing that slowed us down in the beginning was the thinking of creating one feature at a time instead of building everything progressively at once. However, when we got used to it, it actually worked pretty well.