



NFT Playbook

AMOS 2022 | Project 7

Project Retrospective

Schwarzmann Sebastian
Schlinger Johanna
Al-Sheikh Tawfeek
Schilling Johannes
Dreesens Philipp
Rotsching Lukas
Wolfrum Lukas
Diktov Hristo
Kurz Noah

Project Retrospective

We started with the expectation that everyone works on every piece of code. We soon realized that the whole project is too big for everyone to understand every single line of code. Also, we decided to split the project into three separate modules that are contained within a mono-repo. So we have the CLI, middleware and backend as separate packages. Everyone was free to work on whatever he/she wanted, but of course some people became “experts” for certain parts of the project and spent most of their time with only one module of the project.

The communication between the teams worked out quite well and after the initial project setup was done, most of the developers worked very flexibly on different parts of the code. Splitting the project into multiple modules and having at least one “expert” for each module was a good choice and drastically reduced merge conflicts. We had almost none of those and those we had were fairly easy to resolve.

Lesson learned was: Putting some effort beforehand in a good project structure and repo-setup saves a lot of resources later on.

Also the Scrum-Process between POs and SM worked very well.

Since the POs do not have that deep of a technical background, some tickets in the beginning were quite unclear for the developers, so we started a so-called backlog-refinement.

This was an additional meeting we conducted every week one day before the main team meeting. The goal of this meeting was to have the POs and some Devs talk about the new tickets that the POs wrote and use the Devs’ feedback to “refine” the issue descriptions. This made the work for all the devs a whole lot easier because we knew exactly what the POs expected for each ticket.

Aside from this we had no additional meetings with the whole team. People that worked on the same parts of the code could use a self-hosted instance of “Mattermost” in order to orchestrate their work. This worked out well.

Lesson learned: You do not need many meetings. Just start the work and communicate with team members where necessary, but do not waste everybody’s time with many meetings.