AMOS SS25 #1 – EMBArk Orchestrator

PROJECT RETROSPECTIVE

What went well?

Overall, we can state that the project was a lot of fun. We've learned a lot - both through the lecture and the practical experience throughout the project. Even though our team was a bit understaffed on the development side (only four devs), we kept going and achieved the desired results. This was only possible as every part of our project team (POs, SDs, Scrum Master and Industry partner) was highly motivated and pulled their weight.

By utilizing a git branching model including pull requests as well as GitHub Actions to programmatically enforce codestyle, we managed to maintain code quality. This was rather important, as we extended a real-world project.

When we had bigger features that weren't easy to split up, we often used pair programming to solve them. That turned out to be both efficient and fun. Working in pairs helped us move faster and spot mistakes earlier, plus it was just nice to have someone else to talk through the problems with.

While filling 90 minute team meetings seemed like a difficult task at first, we managed to use the time slot efficiently and discuss what needs to be discussed. Almost always, we obtained valuable questions and discussions to further clarify the feature. If something came up between meetings, we just dropped a message on Discord or contacted the industry partner via e-mail.

What didn't go well?

One of the biggest issues we faced was a single feature that ended up taking way longer than expected. Looking back, the main problems were a lack of clarity around what exactly needed to be done, and the fact that we jumped into optimizing things too early. We've learned that we should have invested more time into clarifying the feature - especially once it became clear it wouldn't be finished after two sprints. The result was that it blocked many follow-up issues and thus the time got tight. Unfortunately, we could not put our learnings into practice, as the course was already almost over.

As our AMOS project extended a real-world application which was already in use, our industry partner was interested in upstream pull requests. Creating a pull request was rather tricky, as on one hand we always worked on new issues while a PR was outstanding, and on the other hand, we had to exclude AMOS specific files (e.g. Deliverables, Documentation) in the upstream PR. While the first point is probably not changeable (as AMOS projects are limited in time), we managed the second point by removing the files in upstream PRs, and excluding the removals once we synched our fork.

Another challenge was dealing with special deliverables required by some sprints. While the whole team was technically responsible for making sure these were submitted on time, it didn't make sense for all nine of us to work on them directly every time. We created another role (similar to the Release Manager) which ensured that the deliverables are present once needed. If this dedicated person made a mistake, unfortunately, the whole team ended up being responsible.

2025-07-21