Dear future AMOS Team,

Our semester has started well. In the first weeks we have familiarized ourselves with the project:

For the POs this has meant discussing the abstract wishes of the customer and deriving tickets with concrete requirements. Our industry partner Sick was always involved throughout the project, but without losing its role as a customer.
Lesson learned: The customer should not micromanage the project.

For the SDs this meant to think about the architecture of the components to be developed and to get familiar with the tech stack. For the fleet management, we were quite free to choose since we wrote it from scratch. We chose Python as the programming language and common open-source frameworks like Flask for the backend and Vue for the frontend.
Lesson learned: Choose a programming language that most members are already familiar with and that is not overly complex. The same applies to the frameworks used, for which there should be enough documentation and tutorials, since not all team members have the same level of knowledge.

For the TurtleBot control software, we were given ROS as a requirement. There are many packages to communicate with the hardware built into the robot or to perform complex tasks like navigating from point A to B.
Lesson learned: Using opensource software saves you from having to reinvent the wheel, even if you have to learn the ropes first.

One challenge was the room situation. We were the first AMOS hardware project. We got a key for an office from the secretary, which allowed us to pick up the robots in the evenings and/or on weekends. During these times it is easier to find a free seminar room and to find a slot where all necessary team members are free. Also, demonstrating finished features is more difficult for hardware projects than for pure software projects, since you can't just let a robot drive around during a Scrum meeting. So, we worked a lot with screenshots and videos. Furthermore, testing with robots costs a lot of time, which is the reason why a simulation is helpful.
Lesson learned: Permanent 24/7 access is crucial for hardware projects. Videos are useful for demonstration. Simulation is helpful for testability.

In the middle of the semester, we were able to present a first minimally viable product. We continued to work on it until the end of the semester. We managed to implement all desired features and to make our industry partner happy.
Lesson learned: The end of the semester comes faster than expected and debugging eats up a lot of time, especially for hardware projects. Pay attention to clean commits already during the semester to reduce the clean-up effort at the end. Keep that in mind to have less crunch time in the end than we did.

Your AMOS TurtleBot Team