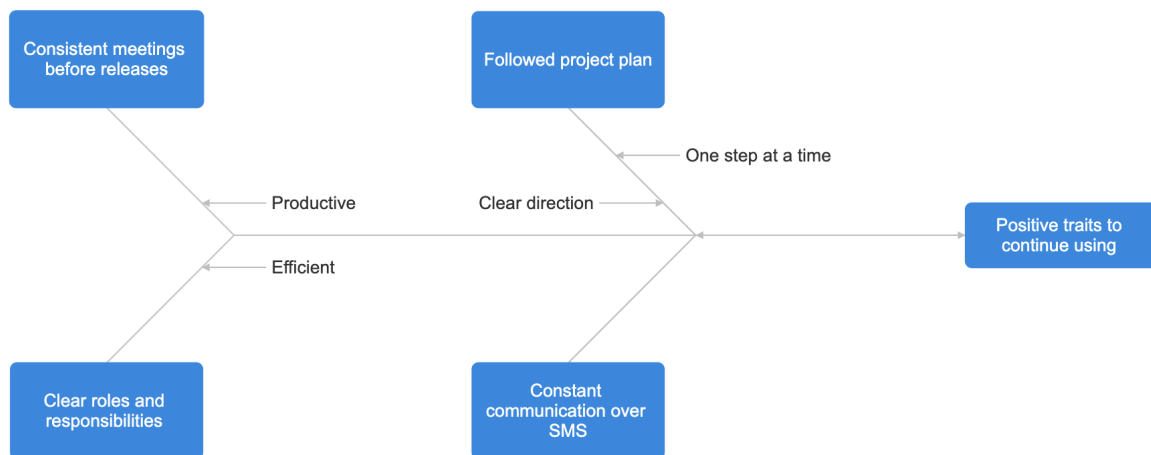**One thing that went well:**
Constant communications and meetings, it kept everyone in the loop and prepared us well for the release presentations. Since we are all on the same page, it was very easy to move forward with our software engineering objectives from our clients, and if anyone was behind, it was easy to catch them up to speed. We were all also very active in the group chat, which also helped us to be on the same page.
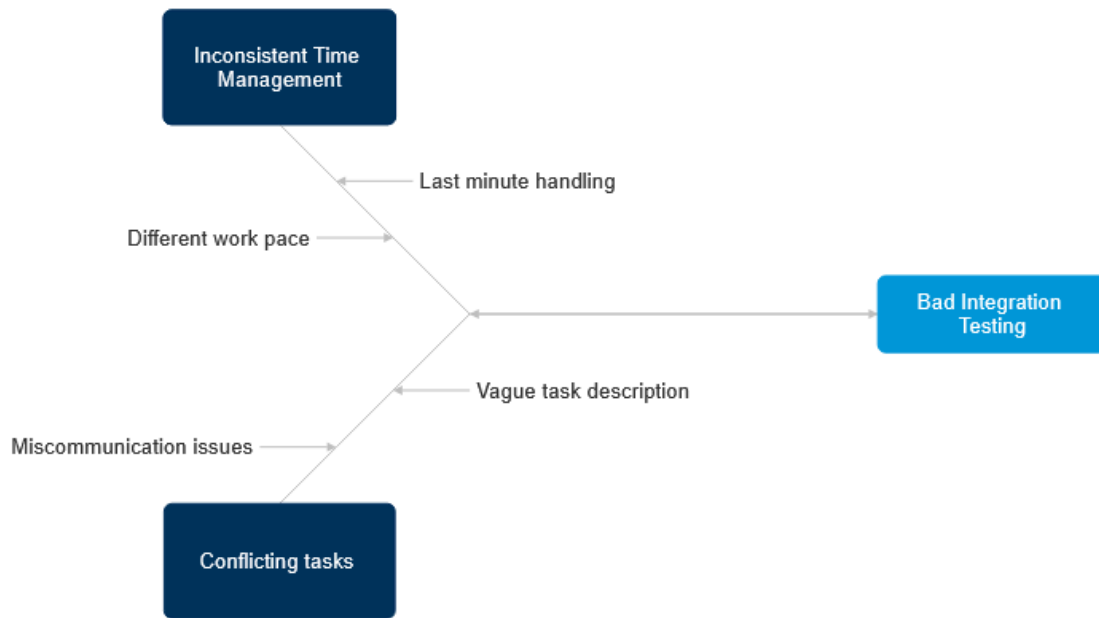
We will maintain this habit of good communication and meeting by making sure everyone will have free time to meet at least once a week to work on programming the project, as well as objectives we have completed/and will work on. We will also communicate with each other in a timely manner in our text group chat.



**One thing that did not go well:**
One thing that did not go as well was integration testing, because we always left it right before the deadline to do. Each of us would have our own modules that we worked on to improve/add/debug features, and if we want to piece everything together, we have to extensively test them and make sure they all work together. Integration testing did not always go smoothly, and it always took a long time to figure out right before the presentation.

Not only that, integration testing ran into some confliction issues during last minute due to unclear task descriptions between the members of the group, making it confusing for the team to understand what part of the system the members are responsible for. This resulted in some occasions in this project where members tested two different codes for the same module and feature.

Inconsistent Time Management

Last minute handling

Different work pace

Bad Integration Testing

Vague task description

Miscommunication issues

Conflicting tasks

We believe starting integration testing early could help us avoid this problem in the future, with clearer task description for clarity within the workgroup, so issues would pop up early instead of all right before the deadline.