

Group I – Retrospective Sprint #2

Stories:

The Team has committed 15 stories out of 25 remaining. All the functionalities of stories were implemented. For some stories, the team didn't complete the test part because we acquired the knowledge about testing only halfway the sprint and we had also to define tests for the stories committed in the previous sprint.

-Story points implemented = 18 , Committed = 21 , Story points not implemented = 3 (Tests)

What went well:

The choice of using Laravel as framework resulted in a complete success: after the initial training, it allowed us to implement features following the MVC (Model-View-Controller) pattern in a simple and vastly efficient way. So, we were confident enough to commit these many stories and we actually managed to complete all of them (apart some testing).

Also, by getting to know each other's programming style, we managed to improve the synergy between us: multiple members working on a same story or feature found themselves comfortable in adapting to the code written by the companions.

Problems and improvements:

Our biggest fault was that, even though we managed to add features from all stories we've committed to the release, we worked more than expected, resulting in an **overcommitment**: we spent all our time budget for the sprint and also an extra 18 hours (3 person-hours for each member, approximately). This was due to two main factors:

1. We needed to improve our knowledge on testing and, since none of the stories we've committed last sprint were tested, we made this our main priority. Too many hours were spent on training and, not to make them waste, we committed extra hours to actually perform them for that set of stories.
2. We wanted to complete all features for the committed stories: leaving a feature incomplete or its graphic not polished enough was simply not an option in our minds.

Another problem was about the setting of the Docker, in particular the compatibility with our database. This was mainly due to our **lack of knowledge** in that field and also **lack of communication**.

For the next sprint, we aim to be more strict about our time budget (we don't want to make extra work without getting paid!) and to be even more specific and precise in detailing the hours we'll spend on the project on the timesheet.

Also, we'll try to be even more polished in the creation of code (more comments, clearer name of methods, more frequent code review...) and to improve our communication (more meetings, higher presence on the Telegram group...).