**Sprint 4 Retrospective**

**Group H**

Macro statistics

**Stories committed: 7**

**Stories done: 6**

**Total points committed: 31**

**Total points done: 18**

**Hours planned: 55h30**

**Hours spent: 62h45**

Detailed statistics

**Hours per task – avg: 57mins, std: 37mins**

**Total task estimation error ratio: 1.19**

**Sprint Backlog:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Story # | Committed | Completed | Story Points | Story Points Gained | Tasks |
| 22 | Yes | Yes | 2 | 2 | 5 |
| 23 | Yes | Yes | 5 | 5 | 7 |
| 24 | Yes | Yes | 3 | 3 | 6 |
| 25 | Yes | Yes | 3 | 3 | 5 |
| 26 | Yes | Yes | 2 | 2 | 5 |
| 27 | Yes | No | 13 | 0 | 9 |
| 28 | Yes | Yes | 3 | 3 | 5 |

**Time Management:**

|  |  |  |
| --- | --- | --- |
| Story # | Estimated Time | Actual Time |
| 22 | 6h | 5h15 |
| 23 | 8h45 | 9h25 |
| 24 | 7h | 3h55 |
| 25 | 4h15 | 3h |
| 26 | 6h | 6h20 |
| 27 | 18h30 | - |
| 28 | 5h | 3h20 |

**Add to the total:**

* **(More) learning: 2h45.**
* **Make application logic compliant with product owner requirements – Tasks 2 (3h40) and 4 (45 mins) respectively.**
* **Bug fixes: 9h20.**
* **Refactoring: 6h45.**
* **Real-world database generation: 8h15.**

Unit testing

**Hours estimated: 6h30**

**Hours spent: 3h25**

**Nr of unit tests written in this sprint: 8**

**Coverage – of new code: 100%, total: 98.40%**

E2E testing

**Hours estimated: 4h**

**Hours spent: 6h15**

Code review

**Hours estimated: 9h**

**Hours spent: 9h**

Technical debt management

**Hours estimated: 3h**

**Hours spent: 5h15**

Overall technical debt at the demo\*

**Number of days: 47min**

**Debt ratio: 0.0%, Rating: A**

<https://sonarcloud.io/dashboard?id=Piryus_Electronic_Student_Record_Management_System>

\* at the demo, something unexpected happened and SonarCloud was showing we had 135 bugs in our code. That was because we committed the unit tests coverage file (coverage.html), without excluding it from the sources of the SonarCloud analysis. Indeed, all 135 “bugs” were related to the autogenerated file coverage.html.

Assessment

* Errors in estimation are very frequent in frontend tasks (always underestimated). That’s because both some team members are not proficient with the technology in use and because the design of new features is not done a priori. It is difficult to know how much time the frontend team will need to implement an interface, if there is no clear description of how it should be.

Errors in estimation happen sometimes on unit testing tasks as well, but there the slight overestimation is deliberately done because it is not possible to predict which functions will require more attention, more edge cases, more test data.

* We would have preferred to have the real-world database some sprints ago, since that would have prevented us from making lots of mistakes. For example, having just one school class in the database was ok for the first sprint, but it was not enough from the second on.

In this sprint we experimented doing tests right after writing the code, and not at the end of the sprint like we did in previous sprints. That allowed us to discover bugs right away, even before frontend/backend integration, and saved us a tremendous amount of headaches.

We learned that technical debt is important and definitely a good thing to care about, but sometimes SonarQube (and similar tools) just complain about nonsense stuff. So, like everything, you don't have to make a big deal out of it. Just use it to spot bad, insecure, potentially dangerous code.

In this sprint more than in any other, we discovered that modifying other’s code is not that easy. We all should write code not just to make things work, but also having readability and clarity in mind. After 4 sprints roles and “experts” were slowly defined, but we never endorsed code ownership or exclusive competences.

* Sprint 3 improvement goals have been achieved to a great extent. Most of product owner requests have been implemented in the application and coordination between team members has reached a remarkable level, even though most of the work is done remotely (due to each team member personal and academic duties).
* The team is very proud of the developed product. It is easy to use, functional and straightforward. We encountered obstacles, and we have always been able to overcome them. We would help each other tirelessly, because working in a team does not mean assigning tasks and then “playing the sport” individually. Being in a team means understanding what are the worries, flaws and difficulties of others and doing your best to help them grow.