Project Retrospective – Metadata Analyzer WS 2024/25

What went well

- The team size was adequate and worked smoothly. Though nine people is a large number to handle our experience was positive because of consistently open and direct communication.
- Apart from minor roadblocks our industry partners were responsive and gave useful feedback in a friendly manner using a shared slack channel. The scope of the project was fitting as well.
- The team members were a good pick. Because some of them were familiar with each other it set the precedent for a positive work environment and straightforward communication.
- We achieved a successful scrum workflow, because of this the learning experience was generally effective. That the course required focus on scrum practices and not only results promoted the correct strategies. Following these guidelines also improved the development process.

What could have gone better

- The homework specification was sometimes unclear, as there were multiple ways to interpret the assignments in the document. For example, it states that the feature board „... must be in a clean state […] ready for action.“, what exactly this entails is not self-evident.
- The homework document is also not consistent with the slides, e.g. there are mentions of demo day posters in the slides but not in the work section of the homework document. This causes confusion if the assignment is something that should nevertheless should be submitted.
- There are too many sources to get information from. Explanations are either in the slides, the homework document, the planning document, or the main schedule document, and require attentive cross-referencing in order not to miss anything.
- Making the final release plan is tedious because everything in it must be noted mutlitple times. Updating the issues in all places causes excess work with no actual additional value.
- The grading process is very unclear. Lots of data plays into what seems like a black box, e.g. feature boards, tests, actual code, number of commits and behavior in team meetings. In the end
- Lots of extra effort was put into the project that could not be documented. Examples are bugfixing, helping others with tasks, and making conceptual decisions.
- The presence of grader in team meetings works against the confidentiality proposed by the Las Vegas rules for agile teams. Some people may hesitate to talk about things that did not work optimally, but could decrease the grade of person who caused it.
- The project timeline was chaotic, with lots of meetings and deadlines where it was not always clear what is due when and how. Take for example the deliverables, it wasn’t clearly stated if it was ok for them to be added to a secondary branch, or if they need to be merged to main before a team meeting.
- Some work that needed to be done for the submissions was virtually duplicate. The build concept, architectural design and user guide were already documented in readmes and through our wiki, but also needs to be put into document form for the deliverables.

Suggestions for the Future

- It is very convenient to have a general slack channel for all pressing questions. However it would be great if the questions could be answered in a more timely manner.
- The introduction of a technical lead position could help the team make solid decisions early on. It would require a more experienced developer who takes guiding position for design choices.
- Reducing the organizational overhead, for example by not documenting the same things in github, the planning document and the code would leave more time to concentrate on the actual work.
- One central file that contains all relevant information would save a lot of time.
- The disadvantages of scrum come to light when working with it over prolonged period. It would be nice to talk about it in more detail in lectures, and maybe compare it to other agile methods.
- One could slim down the final release plan. The weekly updates were too many, and clearer instructions on how to handle it would be appreciated.
- Team meetings have lots of agenda points to work through, so staying within the time frame is tough even with a structured approach, especially if you take the review and planning seriously. Moving the sprint retrospective to an additional meeting could address this.
- Building on that, two week sprints would allow for longer time to distribute tasks. One could do a review planning session at the end of each sprint, and the sprint retrospective in the off weeks.