Retrospective - third iteration

Syntax Highlighters

April 19, 2018

What worked well?

Gradle: We switched build systems from Maven to Gradle, and we have noticed a marked improvement in build system satisfaction and ease of use since then. Gradle is a lot easier to work with, and offers several advantages over Maven, so it was well worth the switch.

Continuous integration: We've added continuous integration, which is really helpful to make sure that all the pushed changes are working. It helps the entire team to check which commits break which things, which is especially useful because it's hard to remember to always test before you push.

Attendance: We've still maintained high attendance, and people are still very diligent about notifying the group of their absence well ahead of time.

Previous iteration buffer: Because our project already implemented much of the functionality specified in this iteration's assignment, and as other parts were already nearly supported, not to mention the mostly sound backend, we didn't have as much we needed to do during this iteration, which freed up resources to do some of the things that weren't strictly required but which we wanted to implement.

Improved use of issues: We have become better at creating and distributing issues. If it's something small, we still often just fix it, but for many things we've been able to document the process of discovering a problem, formulating what the problem is, assigning someone to work out the solution, and solve the problem.

What didn't work so well?

Frontend shift: Because most of the logic was already in place, there was not a lot to do on the backend, whereas the frontend work didn't decrease as much, so the amount of work on the frontend and UI became comparatively high. This was a problem because we have relatively few people who are able to work on the UI, and trying to train people in this area would be a costly endeavor. We've tried to shift some resources around to deal with this, but it's difficult due to the next problem...

Role entrenchment: We've noticed that the main problem of having more or less fixed roles or areas of responsibility is that it is difficult to shift resources around - this is especially noticeable regarding the frontend shift, but also a problem in other areas of the project. This leads to inflexibility. The only solution we can see to this problem is if everyone steps up their game and attempts to get an overview of the parts of the project they're not too familiar with, but there is the issue with time and other courses which demand our attention, which makes this a difficult thing to prioritize.

Task scheduling difficulties: Although better than the last iteration, due to the sheer amount of work we had to do at the time, we've still found it difficult to spread the workload evenly over time, and as such, more work remained at the end of the iteration than we ideally would have. In order to combat this we should be more firm regarding deadlines, and give smaller tasks stricter deadlines to ensure that they're done as soon as possible so we can move on with the project. Perhaps we should make more use of the built-in deadline information in Gitlab issues.

Focus during group meetings: This continues to be a difficult thing. It's also something we clearly have no idea how to improve on, seeing as it has been a problem through three iterations. Related is our general lack of a solid agenda before group meetings - some people in the group argue that we should be better at going through certain topics every time, such as what issues have been completed and which resources we have available to redistribute, but the problem of holding everyone's attention for extended periods of time still stands. Still others argue that the use of the group meeting time as sort of a ”group programming session” is fine.

Neglection of testing: We haven't been as diligent in testing the program as we should have been. It's particularly difficult this iteration because there have been relatively few logic changes, and much UI changes. The logic part is easier to test.

Some issues still large: There have been some issues that have been quite large also during this iteration, and as such, have been ongoing for most of the assignment. We should try to improve on splitting such issues into more achievable subtasks.

Split communication channels: Some people prefer to use Facebook over Discord, and vice versa, and it doesn't look like either side is willing to give ground. It appears that the only solution is to double post anything that is sufficiently important and several people need to see.

The future

Some of our greatest problems during this iteration have been lack of flexibility and poor resource management. We need to address this for the next iteration, preferably shifting from a role-based system to one where everyone can in principle do anything. However, whether we'll actually be able to accomplish this is a different issue entirely. It's really difficult to address many of the issues described above when you find that time and motivation are of the most precious resources.

We do not have any particular technologies we would like to consider for the next iteration for the time being, as we are happy with the current system. However, we would like to get better at using the tools we are currently using, and we still need to consider potential improvements to the process, as partly outlined in the previous section. There might also be some technologies we are forced to start using based on the coming assignment requirements, but it's impossible to plan for this ahead of time. All we can do is try our best, and hope for the best.