Retrospective

Automatic Project Detection And Tooling For Devs

Answers from group members

Keep

K1 *Weekly standup meetings between the team members* – We held a meeting once per week where every team member was present at a time best for everyone. Here we discussed potential issues that we've ran into and may need help with. This was our primary method for distributing tasks, beside the issue board.

K2 *Well established issue board and milestone system* – The priority of issues were clear due to our extensive use of GitHub's milestone feature and issue board. High priority tasks were always solved before low priority ones, which meant we could stay ahead of the deadlines.

K3 *Fast, asynchronous communication methods* – Communications outside of meetings through social media. We responded quickly to the requests of other team members and could even leave messages for the others to see if they weren't available.

K4 *Balanced division of labour* – Our work distribution was even between team members. Noone was overworked or underworked and nobody had to stress under deadlines. We didn't allow anyone in the team to take up more work than they could handle, which lead to a healthy work-study balance.

Drop

D1 *Multiplatfrom communication* – Our communication was scattered across multiple platforms. Frequently we missed memos, because one member messaged another member on a platform they don't frequently check. This led to some miscommunication and a few delays.

D2 *Retroactive discovery of issues* – Issues were occasionally discovered after they were merged into main. We later started inspecting commits more thoroughly, which certainly helped.

D3 *Slow communication with coach* – Meeting times with the coach were at inconvenient times for most people, which led to difficulties when figuring out key details in the requirements.

Improve

I1 *Non-descriptive issue specifications* – Especially early on, figuring out what key functions were supposed to do with minimal specifications was difficult. Occasionally the type signatures in issues differed from the ones we discussed on other platforms which lead to some major confusion.

I2 *Low priority for documentation* – Documenting code was a low priority task and was shelved in favour of adding features. We would later regret this decision, as certain larger functions became unmaintainable to anyone, but the author.

I3 *Ad-hoc coding conventions* – The coding style varied between team members, which makes written code easy to differentiate. We didn't use any code design principles or patterns which makes the program a bit fragile. The MVP architechture was forced on the project which just made the codebase more complicated.

I4 *Grammatical and linguistic shortcomings* – English is not the first language of any of the team members, which lead to grammar that was rough around the edges. This is more noticable in the earlier documents and writeups.

Votes

Answer	Csüllög Benedek	Laczkó Zsófia	Petes Márton	Merth Borbála	Gergely Dániel
K1	✓				✓
K2	✓		✓	✓	
K3	✓	✓		✓	✓
K4		✓	✓	✓	✓
D1		✓	✓	✓	✓
D2	✓	✓	✓	✓	✓
D3		✓		✓	
I1		✓	✓		✓
I2	✓			✓	
I3	✓		✓	✓	✓
I4			✓		

Answers sorted based on votes

Keep

K4 Balanced division of labour

K3 Fast, asynchronous communication methods

K2 Well established issue board and milestone system

K1 Weekly standup meetings between the team members

Drop

D2 Retroactive discovery of issues

D1 Multiplatfrom communication

D3 Slow communication with coach

Improve

I3 Ad-hoc coding conventions

I1 Non-descriptive issue specifications

I2 Low priority for documentation

I4 Grammatical and linguistic shortcomings