Evaluate how personal profiling tools predict team performance in a non-trivial project.

**Evaluating Software Team Performance**

I evaluated how profiling tool can predict team performance. I tried to follow Big Five model, which breaks personality into five dimensions. (Kloda, 2025a) Although our team didn’t use profiling tools like Big Five during the project, creating profiles retrospectively helped me better understand behaviour of team members. For example, Martin’s high openness and extraversion aligned with his success in generating new solutions and driving collaboration, especially during brainstorming meetings and communication on Slack. In contrast, Iain, who eventually dropped out of the project, seemed to have low conscientiousness, which means he might have struggled with staying organized and following through on tasks.

Early in the project, we also assigned specific roles within our team to improve coordination and accountability. (Kloda, 2025b) I contributed to informing this team structuring by considering individual strengths, previous experience, and communication styles. This helped us play to team members' strengths and ensure smoother collaboration. To support team dynamics, we created a shared document outlining our approach to conflict mitigation. It included agreed-upon steps for addressing disagreements, such as raising issues during retrospectives, using private channels for sensitive conversations, and involving a neutral team member if necessary. (Kloda, 2025c)

Our team’s performance was evaluated through Agile metrics and retrospective meetings, which let us evaluate team performance and measure productivity. We used Bitbucket and Jira to measure how many tasks we managed to complete. (Kloda, 2025d) Pull requests and amount of commits also indicate steady progress. Slack allowed our team active problem solving. (Kloda, 2025e) Code reviews and pair programming allowed to get feedback and balance review load to prevent bottlenecks.

As a result, profiling tools might offer insight and help predict team performance. In our case, retrospective analysis played a crucial role in enabling continuous improvement. It highlighted unclear task descriptions in Jira, which led us to refine and clarify task details. We noticed that breaking down tasks into smaller, well-defined tasks allowed us to progress faster and contributed to successful sprint outcomes. Communication via slack and teams and pull requests supported bug detection. Commits activity showed participation across team members.

# References

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/LongTailOfTheSoftwareCycle/BigFive.xlsx [Accessed 14 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/RoleAssignment.png [Accessed 21 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/movie\_recommendation\_system/blob/main/team\_documents/Conflict%20Mitigation%20Plan.docx [Accessed 21 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/EpicsProgress.png [Accessed 2 May 2025].

Kloda (2025). [online] github. Available at: https://github.com/Jkloda/LOJuliaKloda/blob/main/Assets/CommunicationSlack.png [Accessed 2 May 2025].