

Community Code Project

aka Continuous Code Reviews

Tobias Dürschmid and Patrick Rein

Motivation of the project

- Give **feedback** on code quality to the Squeak community
- Identify clean code as **good examples for students**

Research Problem

- Contemporary code reviews are done only once before a pull request is merged
 - No continuous feedback on code quality (especially of legacy code)
 - No support for questions of new developers concerning existing code
 - Forces the developers to leave their IDE for commenting on code -> Context Switch
- How to give **continuous feedback** on code quality?

Walkthrough of the approach

The developer Alice reads a piece of code relevant to her current issue. She does not understand, why this is done like it is and **adds a comment** on this piece of code. Because Bob did the last change to this code, he gets **informed about the new comment** by a notification in the IDE. He opens the concerned method and answers the questions. Alice recommends Bob to refactor the code to directly reveal this intention. He does so and **pushes the done button** of the comment in order to hide the comment of the discussion. After Alice has finished her issue, Bob enjoys reading her code and therefore pushes the **"I like" button** of the new method in order to help new developer getting to know the coding styles of the project.

Open Questions

- Who should be notified?
 - All developers that performed a change to the code
 - The last developer who performed a change
- How to annotate the existing comments?
- How to scale the system?
 - Download all comments for all classes
 - Request comments for classes names contained in the image