Summary of
Feedback in Context: Using a Code Review Tool for Program Grading

https://dl.acm.org/doi/10.1145/3478431.3499402

by Groza Vlad, Guia Alex, Hideg Paul

Delivering constructive feedback on student code is a cornerstone of effective programming education. However, achieving this consistently can be difficult. Traditional methods often lack structure, making it challenging to ensure all aspects (functionality, design, clarity) are addressed and that feedback is clear and actionable. This tool represents a novel approach that provides rubric-based feedback directly tied to student code. This streamlines the process for instructors and teaching assistants (TAs) by creating a systematic framework for code comments. The tool utilizes a predefined rubric, guiding TAs through code review and ensuring consistent evaluation across submissions. Additionally, it eliminates the need for separate comment documents, allowing feedback to be directly embedded within the code itself.

This research investigates the effectiveness of a code review tool designed to address the limitations of traditional feedback methods. The tool's key innovation lies in its ability to provide rubric-based feedback directly tied to student code. This offers several advantages such as: **structured feedback**, which refers to the predefined rubric that ensures all aspects of code quality (functionality, design, clarity) are systematically evaluated, **consistency**, meaning the use of a rubric promotes consistency in feedback delivery across TAs and submissions. Another advantage is **contextualization** (feedback is directly tied to specific code segments, improving student understanding of issues and their impact). The last advantage is **efficiency**: The tool streamlines the commenting process, saving time for instructors and TAs.

The student survey results were overwhelmingly positive, with over 90% of respondents agreeing at least somewhat with key statements. Students reported that the feedback format, with comments tied to specific code sections, helped them understand their grades and the rationale behind them. Furthermore, a significant majority (over 90%) agreed that the feedback improved their ability to make corrections and improve their programs. Importantly, 85% of students indicated the feedback made them "a better programmer," and 80% felt it enhanced their learning in the course. These results suggest that the code review tool offers a valuable approach for enhancing code review in programming education, promoting student learning and development as programmers. The high satisfaction ratings, even from students with concerns about the overall grading process, further highlight the tool's effectiveness in providing clear and actionable feedback.