# ASCILITE 2024

## Navigating the Terrain:

*Emerging Frontiers in Learning Spaces, Pedagogies, and Technologies*

# Enhancing Automated Peer Code Reviews in Software Engineering Education with Context-Aware Generative AI

**Pruthvi Patel**
School of Computing and Information Systems, The University of Melbourne, Australia

**Shannon A. Rios**
Faculty of Engineering and Information Technology, The University of Melbourne, Australia

**Andrew Valentine, Eduardo Oliveira**
School of Computing and Information Systems, The University of Melbourne, Australia.

This study investigates the enhancement of peer code reviews in software engineering education through the integration of Generative Artificial Intelligence (GenAI) with contextual awareness. Previous implementations of GenAI, such as ChatGPT, lacked detailed context about the educational content and assessment goals, limiting their effectiveness. Our work-in-progress research addresses this gap by providing GenAI with additional static and dynamic contextual information, including project overviews, pull-request descriptions, and comments. In a controlled study involving 26 students from a 12-week software engineering course, we compared the efficacy of the original GenAI system with a context-enhanced version. Results demonstrated that the context-aware GenAI provided more accurate and useful feedback, as perceived by the students. These findings suggest that incorporating contextual information improves the quality of automated peer reviews, offering a promising tool for educators to enhance student learning and engagement in code review activities.

*Keywords:* peer review, code review, automated code review, genAI, feedback

## Introduction

Peer review has been a well-established teaching method, recognised for its diverse benefits and primary role in offering formative feedback (Liu & Carless, 2006). Defined as a structured critical analysis by a student of another student's work, peer review can be conducted one-on-one, in small groups, or ad-hoc. It can be anonymous or personal, synchronous or asynchronous, and scaffolded or free form. Various classroom implementations exist, such as forums for discussion, critical feedback on presentations, or reciprocal reviews (Pearce et al., 2009). Peer review helps students develop 21st-century skills by applying knowledge, fostering critical thinking, problem-solving, and decision-making abilities (Novakovich, 2016) and it also enhances social, communication, analytical, and evaluative skills while promoting collaboration (Boud & Falchikov, 2008). However, traditional peer review systems face several challenges, including inconsistent quality of feedback, potential biases, and the substantial time investment required from students and instructors alike (Fatima et al., 2018). These challenges can hinder the effectiveness of peer reviews in achieving educational goals.

Recently, generative AI (genAI) has emerged as a possible tool to address some of these challenges where there is a particular need for speed and efficiency, or the workload required to provide a peer review is excessive or overly complex. In this work-in-progress paper we expand upon a prior approach to using genAI to provide feedback in a peer-review activity (Oliveira et al. 2023) by adding additional contextual information to improve the quality and specificity of the feedback it provides. Our study takes place in a software engineering capstone subject where students work on real industry projects with sometimes complex and lengthy code bases. In this subject, students are required to engage in peer code review, a common industry practice, to reflect on and improve the quality of their code.

## Background

**Code review in software engineering subjects**

Code review is a standard industry approach to conducting peer review in software engineering projects. Engaging students in code review processes helps them understand industry coding standards and best practices, which are vital for their future careers in the software industry (Indriasari et al., 2009). It also prepares them to handle complex, open-ended problems with multiple solutions, thus enhancing their ability to work on real-world projects.

Code review is particularly effective in project-based learning environments where students work on real projects and participate in open-source communities. This exposure helps students acquire diverse skills and enhances their motivation by allowing them to contribute to meaningful projects (Alasbali & Bentallah, 2015; Samson & Oliveira, 2023). Moreover, code review processes can help students understand the broader implications of their work and prepare them for ethical decision-making in their careers (Jarzemsky et al., 2023).

Despite its benefits, code review also comes with its set of challenges. The process can be time-consuming, requiring developers to spend considerable time reviewing code changes (Bosu & Carver, 2023; Tufano et al., 2021). This is especially true in large-scale projects with complex codebases, where the volume of changes can be overwhelming (Dey et al., 2020). Additionally, the effectiveness of code reviews depends on the expertise and thoroughness of the reviewers, which can vary widely among team members (Dey et al., 2020; Thompson & Wagner, 2017).

**GenAI code review**

The emergence of GenAI, has opened new possibilities for automated code review, offering timely, standardised feedback beyond the capabilities of standard automated tools (Dey et al., 2020; 2022; Wong et al., 2023). GenAI represents a significant advancement in the field of artificial intelligence, characterised by its ability to generate new data and content based on existing inputs. In the context of software development, GenAI has shown considerable promise in automating and enhancing various processes, including code review (Wong et al., 2023).

The application of GenAI in code review offers several benefits. One of the primary advantages is the ability to automate repetitive and time-consuming tasks, allowing developers to focus on more complex and creative aspects of software development. GenAI systems can rapidly identify syntax errors, code smells, and potential bugs, providing immediate feedback to developers. This not only accelerates the development process but also enhances the overall quality of the code (Oliveira et al. 2023; Wangoo, 2018). Moreover, GenAI can facilitate a higher level of standardisation in code reviews. Unlike human reviewers, who may have varying levels of expertise and subjective biases, GenAI systems apply a consistent set of rules and criteria to evaluate code. This leads to more uniform and objective feedback, which can be particularly beneficial in large teams and open-source projects where maintaining consistency is crucial (Batarseh et al., 2020).

In our previous work (Oliveira et al. 2023) we demonstrated that integrating GenAI into the peer review process in an educational setting not only increased student engagement but also identified a larger number of code issues in a shorter time, leading to more fixes. This suggests that GenAI can enhance both the educational value and the practical outcomes of code reviews.

## Methodology

**Activity design**

This earlier study (Oliveira et al. 2023) identified a clear shortcoming of the AI generated peer review attributed to the genAI model having a lack of contextual understanding of the project and the reasoning for the changes being made to the code (Version 1). To minimise this problem, we created an updated version of the genAI peer review process that was provided with the following additional files (Version 2):

- *Static Context*: This was provided by the project overview document (typically called a README file in software engineering). This document provides an overview of the project, its purpose and what it does

and includes any relevant setup and usage instructions for the project as well as licensing and authorship information.

- *Dynamic Context*: This included the pull-request (PR) description and comments, which offer insights into the rationale behind code changes and the discussions among team members.
- *GitHub Pull Request Form*: To further facilitate the inclusion of relevant contextual information, an optional GitHub PR form was provided. This form was designed to capture key information about the changes being proposed.

These additional files were appended as text to the end of the prompt described in our previous study. The genAI review Version 2 was integrated into the student's normal workflow and as such, did not require any additional tasks to receive it. Over the course of the project, students completed two code review stages, the first occurring in week 8 using Version 1, and the second in week 10 using version 2. For more information on how the genAI peer review process was structured, please refer to our previous study (Oliveira et al. 2023).

**Research design**
**Participants**

Participants were invited from the Software Project (COMP90082) capstone subject (School of Computing and Information Systems, The University of Melbourne), with a total of 60 students (10 groups of 6 students each) deciding to participate in the study out of 186 enrolled students. All students, regardless of participation in this study, used the same tools and followed the same procedures as part of the subject expectations, but data for this study was only collected and analysed for those who consented to participate. Although 60 students agreed to participate, only 26 completed the final survey.

Ethics

Ethical approval for this study was obtained from the University's ethics committee (Ethics approval #24272). All participants provided informed consent, and their data was anonymised to protect their privacy. Students who chose not to participate in the study were not disadvantaged in any way and continued to receive the same educational experience.

Evaluation method

Participants were surveyed in week 11 after they had used both genAI review systems (Version 1 and Version 2) and asked to compare their experiences with both with a mix of quantitative (Likert scale) and qualitative questions (open-ended). In the following section we will present the results of this survey and discuss their implications.

## Results and discussion

Out of the 26 students who participated in the survey, only a few responded to the open-ended questions. Despite the limited qualitative feedback, several important themes emerged. When asked for suggestions to improve the context-enhanced GenAI code review system, the most common suggestion (3 out of 6) was to reduce the length of the code reviews. For example, one student noted, "Code reviews are too long and have too many trivial suggestions". This feedback highlights a potential area for refinement, suggesting that future iterations of the system should aim to provide more concise reviews without compromising on quality.

When students were asked about the impact of the additional context on the quality of the GenAI code review, the majority (5 out of 8) indicated that the quality had improved. One student commented, "PR description helped the GenAI to understand the task for each PR better and gives more context-related feedback for each file it reviews". Another added, "It seemed to give the AI more context and therefore better feedback overall (more relevant feedback)." These comments underscore the perceived benefits of providing contextual information to the GenAI system, leading to more relevant and useful feedback.

A comparison of student perceptions between the original checklist-based system (Version 1) and the new context-enhanced system (Version 2) is presented in Table 1. The data shows a slight preference for the context-enhanced system, particularly in terms of overall satisfaction and the clarity and usefulness of feedback.

Specifically, 92% of respondents were satisfied with Version 2 compared to 77% for Version 1. Similarly, 85% of respondents found the feedback from Version 2 to be clear, compared to 77% for Version 1. This suggests that the inclusion of contextual information positively influences student satisfaction and the perceived clarity of the feedback. Moreover, when comparing the two systems, most students found Version 2 to provide more accurate (16 out of 26) and useful (17 out of 26) feedback, with only a few preferring Version 1. This reinforces the idea that additional context enhances the overall quality of the reviews.

The ability of the GenAI system to accurately identify logical and structural code issues, however, did not see a notable improvement, with 77% agreement for Version 1 and 73% for Version 2. This discrepancy could be due to several factors, including the possibility that there were fewer logical and structural errors at this stage of the project or limitations in the type of contextual information provided.

*Table 1*
Comparison of the results of the Likert Scale questions and percentage of respondents who agree

| | **Strongly Agree** | | **Agree** | | **Neutral** | | **Disagree** | | **Strongly Disagree** | | **% agree or higher** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Question** | **V1** | **V2** | **V1** | **V2** | **V1** | **V2** | **V1** | **V2** | **V1** | **V2** | **V1** | **V2** |
| I was satisfied with the GenAI code review system | 5 | 8 | 15 | 16 | 6 | 2 | 0 | 0 | 0 | 0 | 77% | 92% |
| The GenAI system accurately identified trivial issues (syntax errors, standard code anomalies) in the code | 7 | 6 | 15 | 19 | 4 | 1 | 0 | 0 | 0 | 0 | 85% | 96% |
| The GenAI system accurately identified issues (logical and structural code issues) in the code | 3 | 4 | 17 | 15 | 6 | 6 | 0 | 1 | 0 | 0 | 77% | 73% |
| The feedback provided by the GenAI system was clear | 5 | 7 | 15 | 15 | 3 | 3 | 3 | 1 | 0 | 0 | 77% | 85% |
| The feedback was useful in improving the code | 4 | 6 | 15 | 16 | 4 | 3 | 3 | 1 | 0 | 0 | 73% | 85% |

*V1 – Version 1, V2 – Version 2

## Conclusion

This work-in-progress study has demonstrated that incorporating contextual awareness into GenAI-powered peer code reviews enhances the perceived quality and usefulness of the feedback provided to students in software engineering education. By enriching the GenAI system with static and dynamic contextual information, we addressed a key limitation of previous implementations, resulting in more relevant and actionable feedback as reported by the students.

The findings indicate that students preferred the context-enhanced version of the GenAI system, noting improvements in feedback accuracy and usefulness. Despite these positive outcomes, the study also identified areas for further improvement, particularly in reducing the length of the reviews and enhancing the AI's capability to detect logical and structural code issues. Given the small sample size and the subjective nature of student perceptions, these results should be interpreted with caution. Another limitation of this study was that the quality of the responses was not checked, however previous research (Oliveira et al., 2023) has shown that a genAI code review can outperform student peer review in identifying trivial and medium difficulty problems, and the students did not raise any issues with the quality of the genAI feedback.

Future research could focus on refining the contextual inputs, objectively assessing the quality of GenAI feedback, and exploring the broader impacts on student learning outcomes, including self-efficacy and critical review skills.

### References


Alasbali, N., & Benatallah, B. (2015, October). Open source as an innovative approach in computer science education A systematic review of advantages and challenges. In 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE) (pp. 278-283). IEEE. https://doi.org/10.1109/MITE.2015.7375330

Batarseh, F. A., Mohod, R., Kumar, A., & Bui, J. (2020). The application of artificial intelligence in software engineering: a review challenging conventional wisdom. Data democracy, 179-232. https://doi.org/10.1016/B978-0-12-818366-3.00010-1

Bosu, A., & Carver, J. C. (2013, October). Impact of peer code review on peer impression formation: A survey. In 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (pp. 133-142). IEEE. https://doi.org/10.1109/ESEM.2013.23

Boud, D. and Falchikov, N. (2008). Rethinking assessment in higher education: Learning for the longer term. Routledge, London. https://www.routledge.com/Rethinking-Assessment-in-Higher-Education-Learning-for-the-Longer-Term/Boud-Falchikov/p/book/9780415397797

Dey, T., Mousavi, S., Ponce, E., Fry, T., Vasilescu, B., Filippova, A., & Mockus, A. (2020, June). Detecting and characterizing bots that commit code. In Proceedings of the 17th international conference on mining software repositories (pp. 209-219). https://doi.org/10.1145/3379597.3387478

Honig, C., Rios, S., & Oliveira, E. (2023). A tool for learning: Classroom use - Cases for generative AI. The Chemical Engineer, June 2023, 38-42.

Indriasari, T. D., Luxton-Reilly, A., and Denny, P. (2020). A review of peer code review in higher education. ACM Transactions on Computing Education, 20(3), 1–25. https://doi.org/10.1145/3403935

Jarzemsky, J., Paup, J., & Fiesler, C. (2023, March). "This Applies to the Real World": Student Perspectives on Integrating Ethics into a Computer Science Assignment. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (pp. 374-380). https://doi.org/10.1145/3545945.3569846

Liu, N. F., & Carless, D. (2006). Peer feedback: the learning element of peer assessment. Teaching in Higher Education, 11(3), 279-290. https://doi.org/10.1080/13562510600680582

Novakovich, J. (2016). Fostering critical thinking and reflection through blog-mediated peer feedback. Journal of Computer Assisted Learning, 32(1), 16-30. https://doi.org/10.1111/jcal.12114

Oliveira, E., Rios, S., & Jiang, Z. (2023). AI-powered peer review process: An approach to enhance computer science students' engagement with code review in industry-based subjects. *ASCILITE Publications*, 184-194. https://doi.org/10.14742/apubs.2023.482

Pearce, J., Mulder, R., and Baik, C. (2009). Involving students in peer review: Case studies and practical strategies for university teaching. Centre for the Study of Higher Education, University of Melbourne, Parkville, Vic. https://melbourne-cshe.unimelb.edu.au/__data/assets/pdf_file/0006/3590943/Involving-students-in-peer-review.pdf

Samson, A., & Oliveira, E. (2023). University learning partnerships: Enhancing learning, enabling innovation and addressing challenges in schools. *ASCILITE Publications*, 531-535. https://doi.org/10.14742/apubs.2023.460

Thompson, C., & Wagner, D. (2017, November). A large-scale study of modern code review and security in open source projects. In Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering (pp. 83-92). https://doi.org/10.1145/3127005.3127014

Tufano, R., Pascarella, L., Tufano, M., Poshyvanyk, D., & Bavota, G. (2021, May). Towards automating code review activities. In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE) (pp. 163-174). IEEE. https://doi.org/10.1109/ICSE43902.2021.00027

Wangoo, D. P. (2018, December). Artificial intelligence techniques in software engineering for automated software reuse and design. In 2018 4th International Conference on Computing Communication and Automation (ICCCA) (pp. 1-4). IEEE. https://doi.org/10.1109/CCAA.2018.8777584

Wong, M. F., Guo, S., Hang, C. N., Ho, S. W., & Tan, C. W. (2023). Natural language generation and understanding of big code for AI-assisted programming: A review. Entropy, 25(6), 888. https://doi.org/10.3390/e25060888

# ASCILITE 2024

## Navigating the Terrain:

*Emerging Frontiers in Learning Spaces, Pedagogies, and Technologies*

Patel, P., Rios, S.A., Valentine, A., & Oliveira. E. (2024). Enhancing Automated Peer Code Reviews in Software Engineering Education with Context-Aware Generative AI. In T. Cochrane, V. Narayan, E. Bone, C. Deneen, M. Saligari, K. Tregloan, & R. Vanderburg. (Eds.), *Navigating the Terrain: Emerging frontiers in learning spaces, pedagogies, and technologies*. Proceedings ASCILITE 2024. Melbourne (pp. 647-652). https://doi.org/10.14742/apubs.2024.1446