A Case Study on Student Perspective of Peer Code Review (PCR)

Attiqa Rehman[0000-0003-0375-9867], Rune Hjelsvold[0000-0002-5955-1603], Leonardo Montecchi[0000-0002-7603-9695]

Center for Excellence in IT Education (Excited), NTNU, Trondheim, Norway
{attiqa.rehman; rune.hjelsvold; leonardo.montecchi}@ntnu.no

Abstract. Peer Code Review (PCR) is a professional practice and a learning method. A case study on PCR was conducted in a "Programming Languages" course in the fall semester of 2023 at the Norwegian University of Science and Technology (NTNU). A new protocol for peer code review was implemented where the students received a suggested solution and instructor feedback on their solutions before reviewing their peers' solutions. The motivation for the protocol was to reduce the students' cognitive load, allowing them to focus on assessing the code produced by their peers. A survey among the students showed that they engaged in PCR and found the workload reasonable. The survey also indicates that students found the review work an opportunity for learning even under the new protocol.

Keywords. Assessments, Learning Experience, Programming Languages, Peer Code Review (PCR), Intended Learning Outcomes (ILOs), Student Engagement, Student Perspective

1 Introduction

Peer Code Review (PCR) is a collaborative process used in the IT industry to improve code quality when code is integrated into the codebase [8]. PCR is also practiced in academia to reduce instructors´ workload, build soft skills, train students in giving and receiving feedback, increase their engagement in the course, and, most importantly, involve students in the feedback process [6].

The ability to understand and enhance the code developed by others is essential [17] and makes the integration process easier when fresh graduates join a team involved in an ongoing project [5]. Therefore, PCR is used in a pedagogical setting to give students hands-on experience in scrutinizing the code of their peers [13, 14]. Moreover, practicing PCR at the university provides students with training that is aligned with industry requirements [15]. PCR not only reduces the workload of educators [7] but also boosts students' confidence in their coding skills, deepens their knowledge, and enhances their engagement in their studies [6, 10, 23]. In addition to learning to write quality code, it can also be used in educational settings to provide a way to improve the soft skills and critical thinking abilities in students by requiring students to analyze code, identify errors, and suggest solutions and giving them a chance to deepen their knowledge on the specific topic of the course [23]. Moreover, giving and receiving feedback enhances the learning experience of students [1, 16]. Studies show that the students spend extra time

understanding the topics, increasing their engagement in the course [14]. Lastly, the PCR activity is a student-centric learning activity that can be successfully used to achieve the ILO of programming courses [26].

Along with the potential benefits of PCR for educators and students, specific challenges are associated with peer code review. These challenges at the student end are a lack of knowledge or ability to perform peer code reviews, low learning engagement, and low review quality [10]. The authors [15] reported that students were reluctant to provide feedback on their peers' code due to concerns about their coding experience and validity, reliability, bias, and fairness. According to [13] the challenges associated with PCR, students think they are not competent enough to review code produced by their peers and that the peer assessment activities are considered extra workload, etc.

PCR in educational settings may also act as a form of peer assessment. There are various ways to implement peer assessment in courses [21]. This paper reports on a programming languages course combining PCR with regular assignments to enhance student engagement and deepen their knowledge. PCR was framed to reduce the cognitive load on students and to avoid assigning much extra work for the students. To do this, we revised the PCR protocol: students were given suggested solutions and instructor feedback on their solutions before reviewing their peers' solutions to the assignment. This approach reduces cognitive load by easing the burden of finding working solutions. This paper investigated whether the new protocol would compromise the positive aspects of PCR, such as engagement and learning through PCR. We also aimed to find the student perspective on various issues of PCR (e.g., the difficulty of understanding and evaluating the code of others, the time required to perform PCR, etc.). We also compared the assignments that were failed by the ones failed by the instructor to ensure the accuracy of the PCR output. After performing PCR, the students were asked to complete a survey. The data gathered in the survey was analyzed to answer the following research questions related to the new PCR protocol:

RQ1. How did the PCR activity impact the student's engagement in the course?

RQ2. How did the PCR activity impact the students' assessment of their work?

RQ3. How do the students rate the importance of the various PCR issues?

The rest of the paper is organized as follows: Section 2 presents background knowledge, Section 3 presents the design of the new PCR process and the related basic conceptual framework, Section 4 elaborates on the survey process and the double blinded peer code review assignment, Section 5 presents the results gathered during this study, in Section 6 the results are discussed, and Section 7 concludes this paper and presents some potential future directions.

2 Background

Topping [21] reported that students experience high cognitive load during PCR due to task complexity. Moreover, students may feel that they are not sufficiently competent to perform peer assessment [10, 14], mainly due to a lack of knowledge. A conventional PCR protocol, see [9, 15, 21, 24], has the following steps: 1. The teacher opens an assignment. 2. The students submit their solutions to the LMS. 3. The teacher

distributes the assignments among the peers, possibly after anonymizing the solutions, depending on the policy of institute or country. The teacher may also provide guidelines for the review task. 4. The students submit the reviews to the LMS. The students may then access the reviews. Some variants of the PCR protocol allow students to improve their work after participating in the PCR. The quality and accuracy of the reviews are two significant concerns educational researchers and educators [13, 21, 20] raised about this conventional protocol. In peer review, the cognitive load added by the peer review task comes on top of the cognitive load of solving the assignment [21]. The complexity of the assignment should, therefore, not be too complex.

In this study, we used our newly designed protocol, where students had feedback on their own assignment and a suggested solution before performing PCR. The proposed solution reduces the complexity of the peer code review and the cognitive load.

3 The Intervention

3.1 Conceptual Framework

Researchers in higher education have been studying student engagement and how it can be increased to achieve the intended learning outcomes (ILOs) [22].
The conceptual framework of this study is based on applied learning experiences and active and collaborative learning. Social constructive alignment is used as a lens to study the impacts of the new PCR protocol. This framework has three essential parts: ILOs, Teaching and Learning Activities (TLAs), and Assessment Tasks (AT) [2].

PCR is a student-centric approach to achieve ILOs, where students participate in code writing, code analysis, and feedback exchange. In programming courses, PCR supports ILOs like writing, compiling, debugging, and providing alternative solutions [26]. Social constructive theory already works as a reference for peer review, peer review in group-based project, and problem-based learning. Social constructive theory informs peer review, group projects, and problem-based learning, and PCR enhances participation and engagement [11, 22].

3.2 The PCR Protocol

Figure 1, illustrates the process for evaluating the intervention or new PCR protocol, outlining key steps involved in the process of PCR assignment. The process begins with the instructor publishing the assignment and guidelines, followed by the students working on and submitting their solutions. After submission, the instructor team reviews the submissions, provides feedback to each student individually, and provides comments. These comments are then published, and each student can view feedback on their own assignment. After that, students are given access to both the solutions and guidelines. Assignment number 2 is anonymized, and each student gets one of the peer assignments for review. In the peer review process, students double-blinded review their peers' assignments. Throughout the process, comments and feedback from the instructor and from the peer are shared with the students who was owner of assignment. Finally, students filled out a survey to provide feedback on peer code and experience of the PCR

activity using Nettskjema[1]. The comments on the peer assignment are then extracted from each survey and redistributed to the student who was the owner of the assignment.

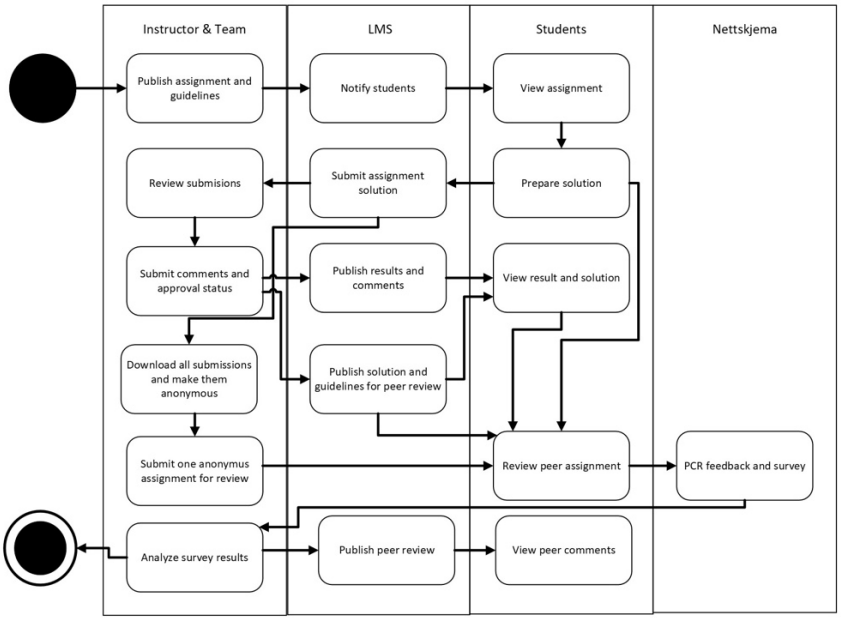

Fig. 1. Peer-Review Protocol

This approach differs from conventional PCR activities as it emphasizes structured interactions between students and the teaching team at multiple stages. One key difference is the involvement of an external tool (Nettskjema) to gather feedback through surveys, also helped to analyze the survey answers. Additionally, the model provides a more guided experience by supplying students with an example solution and guidelines (Appendix B gives an overview of guidelines), which can reduce the cognitive workload as compared to conventional PCR, where students are often asked to evaluate the peer code without an exemplary solution or prior feedback from instructors. This method helps students focus on critical thinking (in the context of debugging and code review) and providing constructive feedback rather than spending excessive time identifying errors.

4 Method

The PCR was orchestrated by a survey hosted on Nettskjema. In this survey, students were asked to blindly review the code assignment of one of their peers. After completing the review, they filled out the survey with open-ended and Likert scale questions. We examined the results of this survey across three phases of peer

[1] https://nettskjema.no/

assessment: Previous Experience (previous experience of peer assessments), Process (peer code review process), and Reflection on the PCR. These stages are explained briefly in the following subsections.

4.1 Previous experience

The questions for this stage focused on the students' prior experience with PCR. The survey also included five questions about the students' views on PCR, covering topics like its impact on critical thinking, engagement, self-assessed competency, and on providing constructive feedback. Responses were typically yes, neutral, or no. Appendix A lists the close-ended questions for this age. The open-ended questions addressed the workload of PCR assignments and students' opinions on what constitutes a fair workload, e.g., a PCR assignment should be equal to the workload of one assignment or given weightage of more than one assignment.

4.2 Process

In the process stage, we provided students with review guidelines and peer code review questions. They were asked to review Assignment 2, while the review was part of Assignment 4. Each student received an anonymized submission for evaluation. The review included open-ended and rating questions, with ratings on a scale from 1 (very poor) to 5 (very good), as outlined in Appendix A. The open-ended questions aimed to encourage students to produce constructive feedback to their peers. The reviewers' responses were shared anonymously with the original authors. Sample review answers, not based on actual assignments, were provided to illustrate constructive and non-constructive feedback in the guidelines, see Appendix B.

4.3 Reflection on the performed PCR

In this stage, we examined how much extra time students spent to broaden their knowledge, their opinions on the provided feedback, and their expectations of feedback. The question about "time spent" aimed to assess how PCR impacts course engagement, contributing to RQ1. For RQ2, we asked students what changes they would make if having the chance to resubmit their assignments and reflecting the valuable insights and deeper understanding they've gained through reviewing their peers' code. Students also rated various PCR-related issues (see RQ3) from 1 (most important) to 5 (least important see Appendix A), based on a shortlist from [12, 21]: 1) Difficulty of understanding others code, 2) Difficulty to evaluate other code. 3) Giving more weight to PCR assignment, 4) More time to conduct a PCR Assignment, 5) Tools to facilitate the process, and 6) Training and guidelines. Finally, we asked the students to suggest how the PCR process could be improved.

5 Results

Survey responses from all three parts were analyzed, using descriptive statistics for the Likert scale questions and inductive thematic analysis for the open-ended questions.

Out of 107 students enrolled in the course, 58 students performed PCR and filled out the survey. Out of 58 respondents, 49 consented to having their responses to open-ended questions available for research. For the Likert scale analysis, all 58 responses were included, while the open-ended question analysis was based on the 49 students who provided consent.

5.1 Previous experience

As mentioned above, the first section of the survey focused on students' prior experience with PCR and the activities they performed. We found that 31% had previously participated in peer review. Regarding programming languages, 67%, 44%, 11%, 5%, and 5% of students had done PCR in JavaScript, Java, Python, C, and C++, respectively, while 28% used other languages. In their previous PCR experiences, 83% assessed code correctness and formatting, 56% evaluated program efficiency, 28% checked object-oriented concepts, and 16% reviewed functional programming. Only 5% assessed user interfaces, and 22% couldn't recall specific PCR activities they performed in PCR.

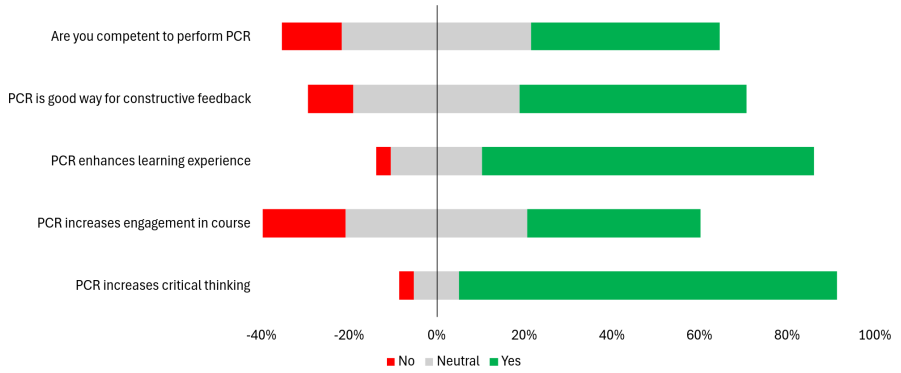

Fig. 2. Impact of PCR and competence of respondent

The students also rated PCR's impact on their critical thinking, engagement, learning, constructive feedback, and their confidence in performing PCR. Figure 2 displays these results, with negative values in red on the left for "No" responses, positive values in green on the right for "Yes," and neutral responses in the middle.

An open-ended question regarding the previous experience stage asked whether the workload of the PCR assignment was comparable to that of a regular assignment. The inductive thematic analysis of the responses revealed that the perceived workload varied depending on the actual solution a student gets for review. Here are some sample responses:

"In general, yes, but it depends on the size of the assignment to be reviewed. Obviously, if the assignment is huge and not well documented it is going to take a lot of time deconstructing the code before reviewing it, and in contrast it is going to take a lot less time if it is short and well documented. I do think peer review is a useful tool for learning diverse ways to solve the same problem though, and that smaller, but more frequent peer reviews are better than one singular big one."

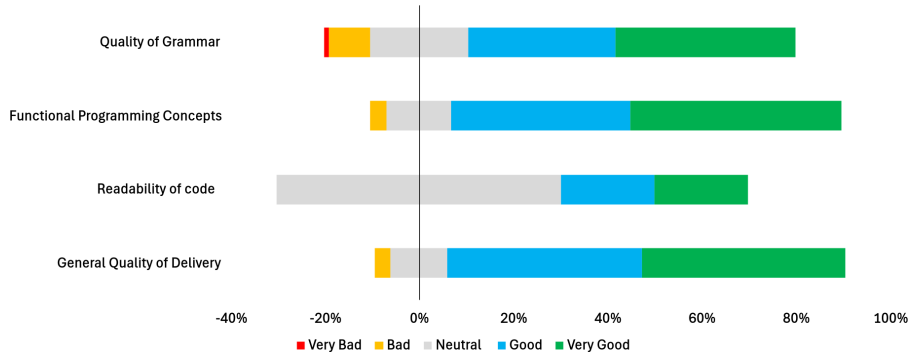

Fig. 3. Students rating of general characteristics of peer assignment

"Yes, in my opinion reviewing one assignment is a fair workload distribution of giving comments on my peers' work. It depends on the magnitude of the assignment, but in this case, it is fair. Additionally, it allows for a more thorough and thoughtful review process, benefiting both the reviewer and the author of the assignment."

5.2 Process

We provided students with a guidelines (See Appendix B) to assist them to give constructive feedback. They were asked to rate, on a Likert scale, the quality of the reviewed assignment, code readability, proper use of functional programming concepts, and the quality of the defined grammar. Students were then asked to report any problems they encountered during the peer assignment. Here are some quotes from their responses:

"I don't find any error. All the functions run correctly, and the output shows the correct answers for each of the three tasks."

"No problem or error, I just had to replace the path to code."

"No immediate issues were found, both task 1 and task 2 ran successfully. There was a large amount of duplicates in the code for task 1 which could have been erased, as each sub-tasks reintroduced functions from previous sub-tasks, reducing the readability of the code".

The next set of questions involved comparing the submitted assignment solutions to the suggested solutions provided by the instructor team and to the students' own solutions. This comparison was conducted separately for each question, resulting in two responses from each student for every question.

Comparison between own assignment and the one reviewed: "My solutions focus more on creating internal functions to the declared functions. One example here is the "Calculate" function my colleague has created. It can (although it does not have to) be defined as an internal function to the "Interpret" function. It makes more sense to it like this since the "Calculate" function is not needed anywhere else for now. Likewise in my implementation I could have split the "InterpretStack" internal function of the "Interpret" function, into more functions to express the intention of my code and its overall readability more clearly."

"The review code is much more compact than mine thanks to clever use of case statements and by using the "FoldL" function in "Interpret" to accumulate the list."

“My colleague's solution is much shorter and simpler, and thereby more favorable here I would argue. Specifically, the ways lists are manipulated is very elegant and something I certainly will apply to my coming assignments.”

“My code reuses the "Lex" and "Tokenize" functions from task 1 to produce a list of tokens. The review code creates new functions to perform this same task”

Comparison between published solution and for the one reviewed: “The published solution uses the "Push", "Peek", "Take" and "Drop" functions from assignment 1 to interact with the stack. The review code, however, does not use them at all, and relies on pattern matching and the "|" syntax instead”

“Fancy pattern matching, and very functional logic. It is worth mentioning that the solution also does more error-handling, in both Task 1 and 2”

In the assignment, we assessed whether students could pass a function as an argument and return a function as a result, both of which are examples of higher-order programming. Students were asked to critically analyze whether the assignments they were reviewing utilized higher-order programming concepts.

“I would not say higher order programming is not used here because no function is strictly passed as arguments. Instead, the result of called functions is passed as arguments to functions”

“Yes, but too a much lesser extent then my code and the published code.”

The students are much stricter than the instructor team.

Students were asked to provide some constructive feedback for question number 3 where they question is regarding grammar.”

“I think the formatting of the grammar could be tidier and easier to read if a markup language was used instead of markdown” “The grammar would be more readable if the author had chosen more descriptive names. "pd" for example is more cryptic than "num" when describing a number.”

“Very good response! The only thing I see is that in Task 3c, it states that task a is context-free. According to the solution on LMS, a is a regular grammar. Otherwise, it seems like the student understands how to formally describe the grammar.”

Finally, we asked students whether the assignment should be accepted as a pass and to provide justification if they felt it should not be accepted. The analysis revealed that students offered well-reasoned critiques, demonstrating their ability to critically analyze the code. This indicates that students can effectively serve as peer reviewers, upholding high-quality standards.

“Task 1 is completely correct and working and task 3 is answered. But Task 2 seems to not have a valid attempt. This means that the requirements of the assignment are not met, and thus the assignment should not be given a pass. Had the student explained their thought process on Task 2 more this might have been a pass.”

“While I would give the code a pass, the missing formal notation and (E)BNF in task 3, makes the submission "not accepted".

5.3 Reflection on the performed PCR

The last part of the survey focused on students' reflections on the PCR process and their perspectives on various PCR issues. The first question asked how much extra time

students spent reviewing or deepening their knowledge on the assignment topics (see RQ1). 71% reported spending extra time. When asked if PCR deepened their knowledge of the course topics, 72% of students answered that it did.

Students were also asked if they would like to make changes to their assignment solution after participating in the PCR activity and whether these changes were related to the assignment they reviewed (see RQ2). About 88% indicated they wanted to make changes, attributing these to insights gained from the review. Here are some of their responses:

"I have learned that there are different notations for grammar, beyond BNF and EBNF, as well as that there are more efficient ways of performing the assignment (such as not having long "if-elseif statements"). In addition, I have learned that there are shorter ways of writing the same code."

"I will definitively try to find more places to treat the control-statements of the Oz-languages as values to minimize some code, and I will begin to weed out edge-cases before doing pattern matching, instead of doing that while pattern-matching. I have found that to be much more readable."

"I would change some of my code to better use pattern matching, which I learnt from the code I was reviewing."

"I now understand better how I can handle lists in Oz so I can make my programs shorter and more readable."

"The assignment I reviewed made me realize I made some errors in my own assignment, which I do think was surprisingly educational. "

"Yes, I have gained a more comprehensive understanding of the topics and see mistakes I made in the assignment that I wasn't aware of when I first submitted it."

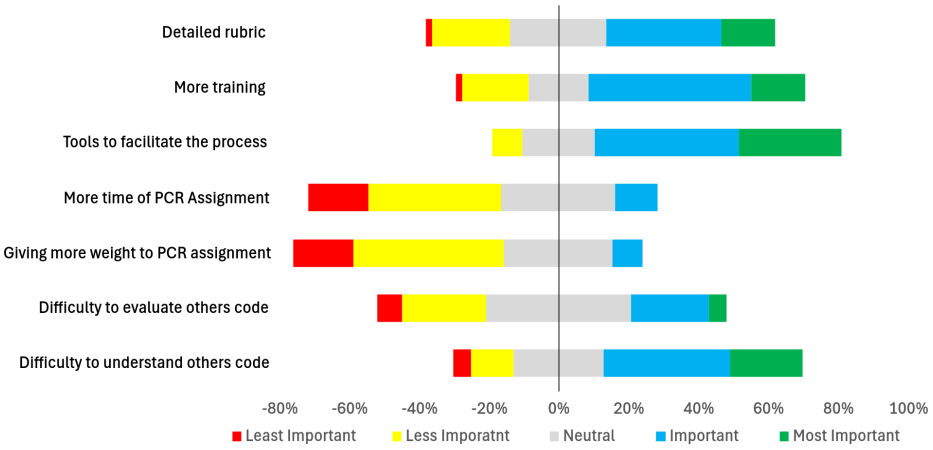

Fig. 4. Peer Code Review Issues on the Likert Scale

Figure 4 and Table 1 present the ratings from the Likert scale and descriptive statistics regarding various PCR issues from the students' perspectives, where 1 is most important and 5 is least important (see RQ3). The results indicate varied viewpoints on the importance and difficulty of these issues. High mean values for "Difficulty to understand the code of others" and "Difficulty to evaluate the code of others" suggest that

many students viewed these as neutral or less important. In contrast, factors like “Tools to facilitate the process,” “Detailed guidelines,” and “Training” had lower mean values, indicating high importance from the students' perspective. The standard deviations reflect moderate variability in student responses. The median values giving an insight of central tendency, the values shows that what was the most common response of students for a specific issue.

Table 1. Descriptive Statistic of Likert Scale Questions

	Factors	Mean	Median	Standard Deviation
1	Difficulty to understand others code	2.4	2	1.1
2	Difficulty to evaluate other code	3.0	3	0.97
3	Giving More weight to PCR Assignment	3.6	4	0.85
4	More time to perform PCR Assignment	3.6	4	0.90
5	Tools to facilitate the process	2.0	2	0.91
6	More training	2.4	2	1.02
7	Detailed guidelines	2.6	3	1.04

6 Discussion

Most students claimed they had no former performed peer review experience, even though peer review should be a mandatory activity in some of the previous courses. As discussed in the results section, Figure 2 shows that about 50% of students felt competent enough to perform PCR, while 35% remained neutral and 15% did not feel competent. A known barrier to PCR is students' lack of confidence in their abilities [13]. However, this study indicates that students are confident in their competency, suggesting that the newly designed PCR protocol may improve their self-efficacy and may contribute to overcoming this barrier.

In our study, 55% of students considered PCR a suitable method for obtaining constructive feedback, with 35% neutral and 10% lacking confidence. These findings align with [7, 19, 24] suggesting that PCR can effectively provide constructive feedback. Thus, we successfully addressed barriers without compromising the positive aspects of PCR, such as learning through peer feedback. A significant majority, about 85%, and 75% agreed that PCR increases critical thinking and enhances the learning experience. Our results indicate that PCR positively impacts the learning process in programming courses when students are not overwhelmed by the task of determining what reasonable solutions should look like. The newly designed protocol tries to reduce cognitive load by providing students feedback on their own assignments and suggested solutions before reviewing other students’ work. This encourages students to deepen their knowledge of the assignment topics and spend extra time studying while reviewing. The responses to the open-ended question about workload indicate that students find the workload reasonable when the work to be reviewed is well-documented, minimizing the extra time needed to understand the code being reviewed. Thus, the workload seems highly dependent on the code they receive for review.

In the first step of the review, students evaluated various aspects of the assignments, including overall quality, grammar, the extent of functional programming applied, and code readability, as shown in Figure 3. The anonymity of the assignments aimed to minimize bias in ratings. Positive ratings emerged, with about 73% of students rating the grammar as good or very good and approximately 80% rating the extent of functional programming positively. The grammar reviewed by students was formal grammar, which is one of the topics of the course and is used to define the syntax of programming languages. The question asked students to rate the grammar delivered by their peers. The collective ratings for overall quality and readability were 82% and 75%, respectively; only a few grammars were given low ratings. However, when asked about using higher-order programming (HOP), only 43% agreed that it was utilized, while 57% did not view the reviewed assignment code as using HOP. This indicates that students were quite critical in their assessments. These results may contradict those presented in other studies [3, 4], which claimed that students tended to give more positive reviews than the instructor.

Students were asked to compare the assignment solution they reviewed with their own and the instructor's solution. Many students demonstrated a positive attitude toward accepting critical feedback to their own code. They responded positively by using peer and instructor solutions to identify errors and weaknesses in their own solutions. This feedback acceptance indicates learning through PCR. Some students also asserted that their own code was better than the code they reviewed, reflecting confidence in their own competence. These comparisons suggest that students learned collaboratively by engaging with each other's work [2] and demonstrated increased engagement [11, 22]. PCR facilitates community engagement and helps achieve course ILOs [25] through collaborative feedback. Additionally, it fosters dialogue among learners and integrates interactive technologies like Nettskjema and the LMS at NTNU.

PCR is a student-centric learning activity where students assess each other's code. Successful PCR requires significant time and cognitive effort. In this study, students reflected on their own solutions, leading to increased behavioral and cognitive engagement. RQ1 focuses on PCR's impact on engagement. Notably, 71% of students reported spending extra time to broaden their knowledge and fulfill their role as assessors, while 72% indicated that they deepened their understanding of the assignment topics. This demonstrates that PCR activities enhance engagement, supporting findings in [2, 6, 11, 23]. Therefore, PCR effectively increases engagement by encouraging students to invest additional time in their assignments, as noted in [6, 14, 23]. So, the newly designed protocol for PCR seemed to have positive effects on the student confidence (i.e., overcoming one of the barriers of PCR) without negatively affecting the positive aspects of PCR, such as engagement and learning through PCR. In our study, 56 students concluded that the code they reviewed was good enough for passing the assignment and that only two solutions were too weak for passing. The instructor team also failed these two solutions while passing the other 56 solutions. This indicates that the newly designed PCR protocol enabled students to provide valid and accurate assessments of their peers' work.

RQ2 examines the impact of PCR on students' assessment of their own work. When asked if they would change their assignment solutions if given the chance to resubmit

and whether this change would be due to PCR, about 88% of students responded affirmatively, citing the reviewing activity as a source of alternative solutions. The results show that students corrected their perceptions of the quality of their own assignment solutions after participating in the PCR and that the review task helped them identify errors and weaknesses in their code. Therefore, it seems that the PCR activities positively affected the course learning outcomes [25, 26] . Thus, our study suggests that the new PCR protocol aligns with social constructive alignment [2].

To answer RQ3, we assessed the importance of various PCR issues, as rated by the students [12, 21]. These issues include difficulty understanding and evaluating others' code, PCR training, guidelines, tools, and assignment workload. Figure 4 presents students' perspectives on these issues. A high mean value of 3.6 for factors such as the need for more weight and more time for PCR assignments indicates that these are the least important. The difficulty in evaluating others' code is rated as neutral, with a mean of 3.0. Detailed guidelines, with a mean of 2.6, are considered somewhat important. The difficulty in understanding others' code and the need for more training are rated as significant, with a mean of 2.4. The most important factor is the availability of tools to facilitate PCR, with a mean of 2.0. The median value of 2 for the difficulty in understanding others' code, tools, and training reflects the importance of these factors. Difficulty in evaluating code and detailed guidelines have a median of 3, suggesting a neutral stance, although most students rated guidelines as important. More time and weight for assignments are the least important, with a median of 4. The high variance in ratings for the difficulty of understanding code, guidelines, and training suggests that student opinions on these issues are highly dispersed.

7 Conclusion

This study used social constructive alignment to examine the impact of a new protocol for PCR. The protocol, which was used in the “Programming Languages” course at NTNU, had students review the solutions of peer students’ submitted assignments in the course. In traditional PCR protocols, students need to analyze the code to be reviewed without having detailed reference points for the review. In the new protocol, students received suggested solutions and feedback on their own solutions before reviewing their peers’ code. The objective was to reduce the cognitive load required for students when reviewing the code developed by peers. A survey was conducted in Nettskjema to evaluate whether the new protocol would reduce the cognitive load and if the PCR activity still had positive effects on the students’ learning.

RQ1 focused on engagement. The results show that students invested time and effort in deepening their knowledge. Students expressed a desire for more PCR assignments and an appreciation for developing critical analysis skills and their involvement in the evaluation process. These results align with the results shown for traditional PCR protocols [1, 2, 6]. RQ2 explored student learning. Nearly 90% of the students participating in the survey indicated they would modify their assignments after PCR, where they got to see and judge alternative solutions. Hence, the new protocol seemed to have a positive influence on learning and self-assessment. RQ3 studied how students rate known

PCR issues when following the new protocol. The students identified tools, training, and guidelines for effective peer review as more important. At the same time, time spent on PCR and the weighting of the PCR activities within the course were considered less important. Future work will involve gathering instructor and educator perspectives through interviews and focus groups to improve the understanding of PCR and its challenges.

References

[1] Almeid Fernando 2018. Framework for Software Code Reviews and Inspections in a Classroom Environment. *International Journal of Modern Education and Computer Science*. 10, (2018), 31–39. DOI:https://doi.org/10.5815/ijmecs.2018.10.04.

[2] Biggs, J. 1996. Enhancing Teaching through Constructive Alignment. *Higher Education*. 32, 3 (Oct. 1996), 347–364. DOI:https://doi.org/10.1007/BF00138871.

[3] Burgess, A. et al. 2021. Peer Review in Team-based Learning: Influencing Feedback Literacy. *BMC Medical Education*. 21, 1 (Dec. 2021), 426. DOI:https://doi.org/10.1186/s12909-021-02821-6.

[4] Carbonaro, A. and Ravaioli, M. 2017. Peer Assessment to Promote Deep Learning and to Reduce a Gender Gap in the Traditional Introductory Programming Course. *Journal of e-Learning and Knowledge Society*. (Sep. 2017), Vol 13 No 3 (2017): EMEMITALIA Conference 2016. DOI:https://doi.org/10.20368/1971-8829/1398.

[5] Cc2020 Task Force 2020. *Computing Curricula 2020: Paradigms for Global Computing Education*. ACM.

[6] Dutta, S. et al. 2023. Enhancing Students' Engagement and Learning through Peer Assessment in Group Projects. *Journal of Educational Research and Reviews*. 11, 6 (Oct. 2023), 93–104. DOI:https://doi.org/10.33495/jerr_v11i6.23.120.

[7] Duzhin, F. and Narayanan, A.S. 2020. Peer Grading Reduces Instructor's Workload without Jeopardizing Student Learning in an Undergraduate Programming Class. *New Directions in the Teaching of Physical Sciences*. 15 (Jul. 2020). DOI:https://doi.org/10.29311/ndtps.v0i15.3466.

[8] Hundhausen, C.D. et al. 2013. Talking about Code: Integrating Pedagogical Code Reviews into Early Computing Courses. *ACM Transactions on Computing Education*. 13, 3 (Aug. 2013), 1–28. DOI:https://doi.org/10.1145/2499947.2499951.

[9] Hyyrynen, V. et al. 2010. MyPeerReview: an online peer-reviewing system for programming courses. *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (Koli Finland, Oct. 2010), 94–99.

[10] Indriasari, T.D. et al. 2020. A Review of Peer Code Review in Higher Education. *ACM Transactions on Computing Education*. 20, 3 (Sep. 2020), 1–25. DOI:https://doi.org/10.1145/3403935.

[11] Kuh, G.D. 2009. What Student Affairs Professionals Need to Know About Student Engagement. *Journal of College Student Development*. 50, 6 (Nov. 2009), 683–706. DOI:https://doi.org/10.1353/csd.0.0099.

[12] Li, H. et al. 2020. Does Peer Assessment Promote Student Learning? A Meta-Analysis. *Assessment & Evaluation in Higher Education*. 45, 2 (Feb. 2020), 193–211. DOI:https://doi.org/10.1080/02602938.2019.1620679.

[13] Middleton, J. et al. 2024. Barriers for Students During Code Change Comprehension. *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon Portugal, Apr. 2024), 1–13.

[14] Mirzakhanov, R. 2024. Enhancing Methodology for Developing Students' Competence in Computer Technology. (Feb. 2024). DOI:https://doi.org/10.5281/ZENODO.10720428.

[15] Oliveira, E. et al. 2023. AI-powered Peer Review Process: An Approach to Enhance Computer Science Students' Engagement with Code Review in Industry-based Subjects. *ASCILITE Publications*. (Nov. 2023), 184–194. DOI:https://doi.org/10.14742/apubs.2023.482.

[16] Panadero, E. and Alqassab, M. 2019. An Empirical Review of Anonymity effects in peer assessment, peer feedback, peer review, peer evaluation and peer grading. *Assessment & Evaluation in Higher Education*. 44, 8 (Nov. 2019), 1253–1278. DOI:https://doi.org/10.1080/02602938.2019.1600186.

[17] Rehman Attiqa and Divitini, Monica 2023. Categorizing Projects for Software Engineering Capstone Courses. *NIKT: Norsk IKT-konferanse for forskning og utdanning Vitenskapelig artikkel* (Norway, 2023).

[18] Song, X. et al. 2020. Using Peer Code Review as an Educational Tool. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (Trondheim Norway, Jun. 2020), 173–179.

[19] Sun, Q. et al. 2019. Formative Assessment of Programming Language Learning based on Peer Code Review: Implementation and Experience Report. *Tsinghua Science and Technology*. 24, 4 (Aug. 2019), 423–434. DOI:https://doi.org/10.26599/TST.2018.9010109.

[20] Topping, K. 2021. Peer Assessment: Channels of Operation. *Education Sciences*. 11, 3 (Feb. 2021), 91. DOI:https://doi.org/10.3390/educsci11030091.

[21] Topping, K.J. 2009. Peer Assessment. *Theory Into Practice*. 48, 1 (Jan. 2009), 20–27. DOI:https://doi.org/10.1080/00405840802577569.

[22] Trolian, T.L. 2023. Student Engagement in Higher Education: Conceptualizations, Measurement, and Research. *Higher Education: Handbook of Theory and Research*. L.W. Perna, ed. Springer Nature Switzerland. 1–60.

[23] Turner, S.A. et al. 2018. Peer Review in CS2: Conceptual Learning and High-Level Thinking. *ACM Transactions on Computing Education*. 18, 3 (Sep. 2018), 1–37. DOI:https://doi.org/10.1145/3152715.

[24] Wang, Y. et al. 2012. Assessment of Programming Language Learning based on Peer Code Review Model: Implementation and Experience report. *Computers & Education*. 59, 2 (Sep. 2012), 412–422. DOI:https://doi.org/10.1016/j.compedu.2012.01.007.

[25] WebPage: Intended Learning Outcomes(ILOs) of the Course TDT4165 - Programming Languages. *https://www.ntnu.edu/studies/courses/TDT4165#tab=omEmnet*.

[26] Yanqing, W. et al. 2011. Learning Outcomes of Programming Language Courses based on Peer Code Review Model. *2011 6th International Conference on Computer Science & Education (ICCSE)* (Singapore, Singapore, Aug. 2011), 751–754.

Appendix A

Table 1. Previous experience Questions

1	Do you think peer evaluation increases critical thinking?
2	Do you think peer evaluation increases engagement in the course?
3	Do you think peer evaluation enhances the learning experience?
4	Do you think peer evaluation is a good way to get constructive feedback?
5	Do you feel you are competent enough to perform peer evaluation

Table 2. Likert Scale Questions for the PCR Process

1	Rate the overall quality of the delivery
2	Rate the readability of the code
3	Rate the extent to which functional programming concepts have been applied in the code
4	Rate the grammar defined in the delivery. Consider correctness of the notation and ability to represent the strings they are meant to represent.

Table 3. Likert Scale Questions for the PCR Reflection

Rate the importance of PCR issues from 1 to 5, where 1 is most important and 5 is least important	
1	Difficulty to understand others code
2	Difficulty to evaluate other code
3	Giving More weight to PCR Assignment
4	More time to perform PCR Assignment
5	Tools to facilitate the process

Appendix B

Table 1. Guidelines to distinguish between Constructive and Non-Constructive feedback

Question	**Non-Constructive Feedback**	**Constructive Feedback**
What did you like about this submission?	It is good	This delivery uses proper indentation in defining the cases and pattern matching, so it is easy to understand what the possible values are returned by the function.
Describe one significant difference between your own submission and the one you get for review.	My code is faster. My code is short, does the same better	The author's code needs more iterations than mine, so my code is more efficient for module XYZ
How the code can be improved?	Not enough documentation. I don't understand the code.	The author should add the proper indentation in function ABC. The functions named XYZ and TRQ are performing the tasks so one of them should be removed.
Did you find any error? If so, briefly describe them.	Function ST is not working	Function ST when assigned 0 as second arguments crashes, as it is used as a denominator. Proper exception handling is thus required.
Do you have other constructive feedback for the author?	This code is perfect	The base case for recursion in function ABC could be defined in a simpler way. In particular...

Do you think it should be accepted? Explain your reasoning.	It is not working.	It should be accepted but functions STV and ATR should be improved, because they provide incorrect output for inputs XYZ