2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51719

WIP: Structured Reflection for Stronger Collaboration in First-Year Introductory Programming Course Projects

Prof. Rose Rani John, Baylor University

Prof. John is a Lecturer in the Department of Computer Science at Baylor University. Her research interests include virtual machine placement in cloud computing. She teaches introductory programming courses for first-year students and is interested in integrating an entrepreneurial mindset into her courses while conducting educational research to improve student learning outcomes.

WIP: Structured Reflection for Stronger Collaboration in First-Year Introductory Programming Course Projects

Abstract

This evidence-based work-in-progress paper presents an instructional intervention designed to support effective group collaboration through structured reflection, aiming to help first year students develop stronger and more adaptive teamwork habits. As computing education increasingly emphasizes collaborative, project-based learning, teamwork has become a central component of many introductory programming courses. However, teamwork among first-year students is particularly challenging and these early struggles can negatively affect both learning outcomes and students' learning experiences.

The intervention embeds weekly individual and team reflections within Canvas Groups to support coordination, accountability, and technical integration. The impact of the intervention is assessed using end-of-semester surveys, qualitative analysis of weekly reflections, and performance metrics from project submissions. Preliminary evidence suggests improved role clarity, earlier coordination, more balanced participation, and higher-quality project outcomes compared to previous cohorts.

Introduction

Project-based learning has become an increasingly common pedagogical approach across engineering and computer science undergraduate level courses, as it provides students with opportunities to apply theoretical concepts in authentic, real-time scenarios [1]. By engaging in projects, students move beyond passive consumption of content to actively constructing knowledge through design, implementation, testing, and iteration [2]. Project based learning is often implemented in teamwork settings [3]. Collaborative projects mirror professional practice by requiring students to manage time, make decisions under constraints, and integrate multiple skills toward a shared goal [4]. As a result, team-based course projects are often used not only to reinforce technical learning outcomes but also to foster broader competencies such as problem solving, collaboration, and self-directed learning, making them a central and popular component of many undergraduate courses [5]. However, teamwork is often a challenge for undergraduate students for a variety of reasons.

Teamwork Challenges

Though project-based learning in teams is highly effective, it has several practical challenges especially while applying with undergraduate courses. Role ambiguity in teams refers to the extent to which individuals lack clear, shared understanding of their responsibilities, expectations, authority, and boundaries within the group [6]. When roles and responsibilities are unclear, teams experience reduced accountability [7], inefficient task execution [8, 9], and uneven workload distribution [10], which can contribute to increased stress and interpersonal conflict. It may further complicate coordination, integration, and delay progress [11]. This may also limit the opportunities for meaningful skill development, making role clarity a critical factor in effective collaborative learning [12, 13].

Unequal contribution among team members is a persistent challenge, where some individuals contribute significantly more than others. This phenomenon, often referred to as social loafing, can reduce motivation and overall team effectiveness [10]. Poor communication such as infrequent updates, or unclear messaging can hinder coordination and lead to misunderstandings, especially in technical tasks requiring integration. Teams often face challenges aligning individual contributions into a cohesive final product, particularly when tasks are interdependent. Lack of coordination can delay progress and reduce quality. Especially with programming project coordination, students struggle to manage the coding environment differences while integrating [11]. Interpersonal or task-related conflicts may arise due to differing work styles, expectations, or priorities. When unmanaged, conflict can negatively impact team cohesion and outcomes [14]. Teams often fail to reflect on their processes, which limits their ability to learn from experience and adapt strategies over time [15]. Without reflection, ineffective behaviors may persist. Especially in first-year or novice contexts, students may lack prior exposure to structured teamwork, resulting in uncertainty about how to collaborate effectively [16].

Instructional Context and Motivation

The instructional intervention described in this paper was implemented in a first-year introductory programming course primarily enrolling computer science major first-year students. The course includes a multi-week, team-based programming project designed to reinforce foundational programming concepts and self-learning of SDL 2.0 while introducing students to collaborative software development practices. The objective of the project is to design and develop a 2D infinite runner game using C++ and SDL2.0. Students work in small teams to design, implement, integrate, debug, and test multiple components of a shared codebase. While the project is intended to simulate authentic technical collaboration and promote peer learning, many students enter the course with limited prior experience working in structured technical teams, particularly in contexts that require coordinated design decisions, code integration, and shared responsibility for debugging and testing.

In previous offerings of the course, it was observed that recurring challenges in team collaboration negatively impacted both the student experience and project outcomes. Teams frequently struggled to define clear roles and responsibilities, resulting in duplicated effort, uneven participation, and delayed integration of project components. Communication within teams often focused on task completion rather than coordination of design choices, leading to late discovery of incompatibilities, duplicate work, and technical issues. These patterns suggested that simply assigning group projects was insufficient to support effective collaboration for novice programmers. These observations motivated the development of a more intentional instructional framework that explicitly scaffolds teamwork processes through structured weekly reflection. The reflection included prompts for individuals as well as group response. The goal of the intervention was to help students develop early awareness of collaboration as a learnable skill and to support more effective coordination, accountability, and technical integration throughout the project lifecycle.

Literature Review

Reflection in an educational context refers to a deliberate and structured process in which learners critically examine their experiences, actions, and outcomes to make meaning of what they have learned and to infer better strategies for future practice. Through reflection, students analyze their decisions, evaluate the effectiveness of their strategies, and connect experience with underlying concepts, enabling deeper understanding and adaptive learning. Reflection is not merely descriptive but involves interpretation, judgment, and intentional planning for improvement. This process supports metacognitive development by helping learners become more aware of how they learn and how their behaviors influence outcomes [17,18].

Reflective practice has its roots in educational and organizational theory, with early work by Dewey emphasizing reflection as an active, deliberate process through which learners make meaning of experience [19]. This concept was later formalized by Schon, who described reflection as a mechanism through which individuals examine their actions during and after practice to improve future performance [17]. Building on individual reflection, subsequent research extended these ideas to group contexts, introducing the concept of team reflexivity, the collective process by which teams reflect on their goals, strategies, and processes and adapt them in response to experience [20].

Use of reflection in Computer Science courses

Reflection has increasingly been incorporated into computer science courses to support deeper conceptual understanding, metacognitive development, and professional skill formation. In computing contexts, reflective activities are commonly embedded in programming assignments, labs, and team projects to help students analyze problem-solving strategies, debugging processes, and design decisions. Structured reflection can improve students' ability to reason about code, articulate learning challenges, and adapt approaches in subsequent tasks [19]. Additionally, reflection has been shown to support the development of collaboration and communication skills in team-based programming environments, where coordination and integration are critical to success [18].

Various forms of reflection have been implemented across computer science contexts. In a human computer interaction programming course, structured self-reflection was incorporated through three reflective writing assignments [21]. Reflection has also been integrated with computational thinking instruction; for example, a seven-week intervention with middle school students embedded metacognitive activities to promote awareness of problem-solving strategies and learning regulation [22]. In advanced programming courses, reflective practices have been used to support design evaluation. One study examined student reflective analyses of their initial and final project designs [23]. Another study used reflective journaling in programming curricula as an ongoing mechanism for students to document challenges, debugging strategies, and conceptual insights [24]. Another study was found on continuous reflection in engineering project-based courses, where students engage in iterative reflective activities throughout the development lifecycle to connect theory with practice [25]. Another study found in the literature used structured tools such as metacognitive strategy cards in computer programming courses [26].

Use of Reflection in Teamwork

In design-based learning environments, it is found that self-reflection in a collaborative environment supports learning by enabling students to externalize thinking, negotiate roles, and make sense of shared design experiences, particularly when reflection is supported through mediating tools [27]. Similarly, structured group reflection in a Web 2.0 learning environment is a significant predictor of group performance, with instructor-supported reflection producing the strongest outcomes among different reflection approaches [28]. Research on team-based project learning further indicates that reflective activities help students evaluate their performance, identify areas for improvement, and strengthen collaboration skills [29]. In multidisciplinary design education, combining active project work with individual and group critical reflection improves student understanding of group dynamics and supports the development of negotiation and decision-making skills [30]. A thorough study on the quality and quantity of team reflections and the impact can be found in [31]. Group reflection is a common and valuable assessment practice in higher education, while variation in reflection quality depend on group composition [32]. Collectively, these studies underscore the importance of intentionally scaffolding collaborative reflection to enhance both learning and teamwork in project-based courses.

The literature on reflection in education consistently demonstrates that structured reflection supports metacognition, self-regulated learning, and deeper conceptual understanding, with studies in computer science education further showing benefits for problem-solving, professional skill development, and awareness of learning processes in project-based settings. Research on team reflection highlights its role in improving coordination, shared understanding, and overall team performance. However, most existing studies treat individual reflection and team reflection as separate pedagogical interventions, focus on unstructured or end of project reflections, rather than ongoing team dynamics. There is a notable gap in empirical work examining the sustained use of structured, recurring reflections that intentionally integrate both individual and team-level prompts within group projects, especially in introductory computer science programming courses for first year students, and in understanding how such combined reflection practices influence the evolution of collaboration, accountability, and interpersonal dynamics over time.

Structured Weekly Reflection for Collaboration

Overview of the proposed methodology

The methodology employed an educational research approach to examine the impact of structured, recurring weekly reflections on team dynamics and collaboration in an introductory computer science course. The study was conducted in a 16-week first year-level introductory programming course offered in two sections, comprising a total of eight student teams. There were five teams with five members, one team with four members and two teams with three team members. The objective of the project was to design and develop a 2D infinite runner game using C++ and SDL2.0.

A structured reflection intervention was implemented over seven key weeks of a semester-long group project, with reflection activities embedded directly into each team's Canvas group home page to support collaborative engagement. Each weekly reflection consisted of two components: individual reflection prompts that encouraged students to articulate their roles, contributions, challenges, learning progress, and self-improvement goals, and a single team-level prompt that required collective discussion and consensus based on the individual reflection responses. The prompts were scaffolded across different phases of the project from expectation setting, through design and implementation checkpoints, to final reflection to capture the evolution of team dynamics over time.

The output of this experiment includes individual student reflections, team reflection artifacts collected across multiple weeks, and an end-of-semester survey administered to students in both sections to capture their retrospective perceptions of collaboration, the usefulness of reflection, and overall team functioning. Student reflection responses were collected and reviewed to identify recurring patterns related to collaboration levels, teamwork styles, and learning experiences, while survey responses provide complementary descriptive and comparative insights across sections and teams, enabling triangulation of findings related to the role of integrated individual and team reflection in supporting collaborative learning among novice computer science students.

Summary of each week's reflection prompts

Across the seven reflection checkpoints, the prompts were intentionally scaffolded to align with evolving phases of team development and project execution. Early-stage prompts emphasized individual self-awareness, role clarity, and expectation setting. Mid project prompts shifted toward monitoring, coordination, and adaptive collaboration, design refinement, timeline alignment, and workload balancing. Later stage prompts focused on integration, knowledge consolidation, collective efficacy, and learning transfer, while addressing collaboration challenges related to code integration, interdependence, and collective problem-solving. Table 1 gives clarity about the expectations of this structured weekly reflection's impact on teams.

Table 1: Weekly Reflection Expectations & Summary

Week theme	Individual prompt focus	Team prompt focus	Team dynamics outcome
Team formation and setup	Roles, strengths, expectations	Role assignment, define responsibilities	Role clarity, early trust, accountability
Planning & setting up expectation	Understanding goals, personal plans	Task planning, short term goal setting	Alignment on objectives, coordination
Design & plan development	Design progress, challenges, suggestions	Design refinement, decision making	Collaborative problem solving
Collaboration checkpoint	What worked, what needs improvement	Progress evaluation, workflow adjustment	Open communication, conflict mitigation
Collaboration checkpoint	Identify contributions, area of improvement	Progress evaluation, workflow adjustment	Individual accountability, mutual support
Integration & coordination	Learning gain, code integration issues	Integration planning, dependency management, progress evaluation	Interdependence awareness, coordination efficiency
Reflection & knowledge transfer	Lessons learned	Retrospective analysis and advice to future teams	Shared learning, transfer of collaborative practices

These structured reflection activities were implemented through each team's Canvas group home page. However, participation varied across teams, with not all groups engaging in the reflections consistently throughout the project. In addition to the weekly reflections, an end-of-semester survey was administered to students across both course sections to capture overall perceptions of the reflection process.

Following is an example of a reflection provided in Week 1. The first three are individual reflection prompts and the last one is a team reflection prompt.

1. What role do you see yourself playing in this project (e.g., coding, design, testing)?

2. What are your strengths that can help the team succeed?
3. What concerns or challenges do you anticipate for this project?
4. Based on everyone's responses, what is our agreed-upon plan for roles and responsibilities?

Results & Discussion

An analysis of data collected from the end semester survey is discussed below. Towards the end of the semester, students took on a survey which was used to understand the impact of the weekly reflection across three constructs such as Team Communication & Trust, Self-awareness & Reflection, and Project Planning & Execution. Figure 1 presents a consolidated stacked Likert analysis of the end of semester survey results. As summarized in Table 2, Self-awareness & Reflection exhibited the strongest positive perception, with 62.5% of students agreeing or strongly agreeing with the highest mean Likert score of 3.59. This indicates that structured reflection was particularly effective in helping students identify personal strengths, recognize areas for growth, and better understand their teammates. In contrast, Project Planning & Execution showed a moderate positive impact of 45.3% and 3.33, suggesting that reflections supported planning, progress tracking, and adaptive decision-making, though these benefits were not uniformly experienced.

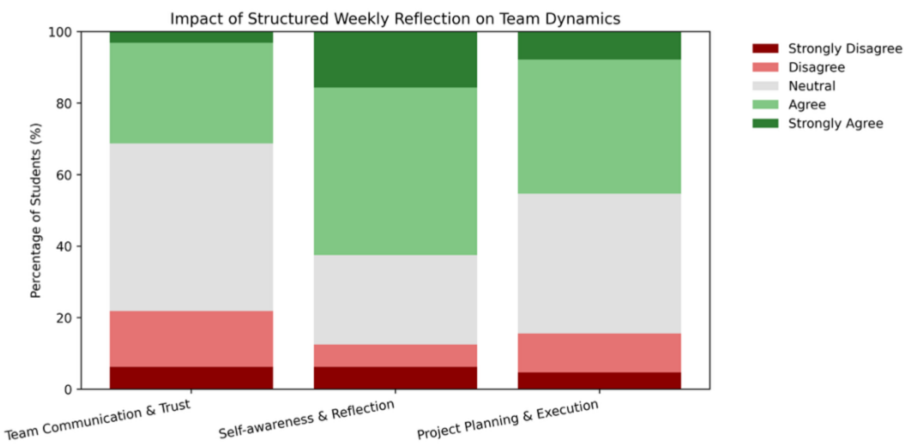

Figure 1. Impact of structures reflection of team dynamics

Team Communication & Trust received the lowest positive agreement and the highest proportion of neutral responses, implying that improvements in interpersonal dynamics may require longer-term or more explicitly guided reflective interventions.

Table 2. Impact of structured reflection in team dynamics

Construct	Agree + Strongly Agree %	Mean Likert Score ± SD (n = 16)
Project Planning & Execution	45.3	3.33 ± 0.82
Self-awareness	62.5	3.59 ± 0.99
Team Communication & Trust	31.2	3.06 ± 0.87

Figure 2 illustrates student perceptions regarding the statement "Consistent weekly reflection would improve team performance." Responses show a predominantly neutral to positive trend, with approximately 44% of students agreeing or strongly agreeing, while a smaller proportion expressed disagreement.

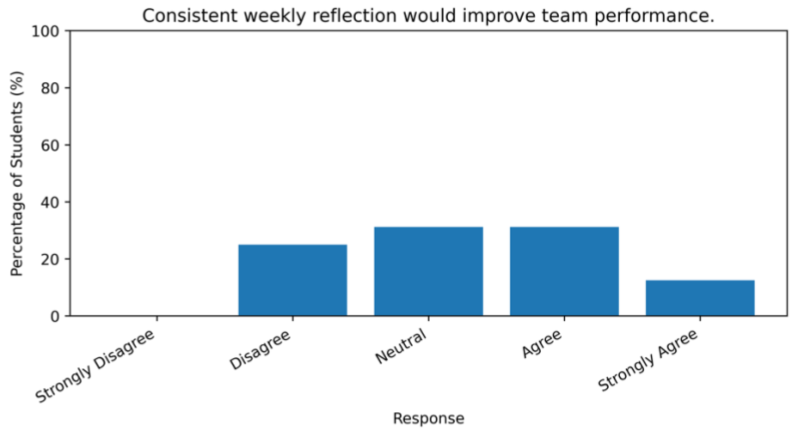

Figure 2. Student perception of consistent weekly reflection

This distribution highlights that the reflective practices appear to depend not only on individual engagement but also on consistent participation.

Future work

This study demonstrates that structured weekly reflection can serve as an effective mechanism for supporting collaboration and team dynamics in first year level computer science programming course group projects. The reflection framework encouraged self-regulation, role clarity, and shared accountability, even though student engagement with the activity was uneven. The findings suggest that embedding reflection in a familiar platform such as Canvas can foster collaborative awareness without imposing significant instructional overhead. Collectively, the reflection artifacts and end-of-semester survey responses indicate that sustained, structured reflection holds promise as a scalable pedagogical strategy for improving teamwork experiences in introductory programming courses and warrants further investigation into strategies for increasing student engagement and longitudinal impact. Since student participation in the reflection activity was uneven, the plan for the upcoming semester is to introduce faculty intervention and provide feedback based on the weekly reflection responses. Since only 41% of the students participated in the end semester survey, the inferences are suggestive and not conclusive from this pilot study. Hence, planning to repeat this experiment at least a couple of semesters to reach a reliable inference on the impact of structured reflection in the context of team-based projects in first year programming courses.

Acknowledgement

I would like to thank Dr. Cassie Wallwey, my mentor from the KEEN Entrepreneurial Mindset in the First Year (EMIFY) program, for her valuable guidance and constructive feedback throughout the development of this paper.

References

[1]. de Los Rios, I., Cazorla, A., Díaz-Puente, J. M., & Yagüe, J. L. (2010). Project–based learning in engineering higher education: two decades of teaching competences in real environments. Procedia-Social and Behavioral Sciences, 2(2), 1368-1378.

[2]. Frank, M., Lavy, I., & Elata, D. (2003). Implementing the project-based learning approach in an academic engineering course. International Journal of Technology and Design Education, 13(3), 273-288.

[3]. Vogler, J. S., Thompson, P., Davis, D. W., Mayfield, B. E., Finley, P. M., & Yasseri, D. (2018). The hard work of soft skills: augmenting the project-based learning experience with interdisciplinary teamwork. Instructional Science, 46(3), 457-488.

[4]. Ritenour, A. P., Ferguson, C. W., Gardner, P., Banther, B. R., & Ray, J. L. (2020, June). Collaborative project-based learning capstone for engineering and engineering technology students. In 2020 ASEE Virtual Annual Conference Content Access.

[5]. Murzi, H. G. (2014, June). Team-based learning theory applied to engineering education: A systematic review of literature. In 2014 ASEE Annual Conference & Exposition (pp. 24-1175).

[6]. Kahn, R. L., Wolfe, D. M., Quinn, R. P., Snoek, J. D., & Rosenthal, R. A. (1964). Organizational stress: Studies in role conflict and ambiguity.

[7]. Belbin, R. M., & Brown, V. (2022). Team roles at work. Routledge.

[8]. Tubre, T. C., & Collins, J. M. (2000). Jackson and Schuler (1985) revisited: A meta-analysis of the relationships between role ambiguity, role conflict, and job performance. Journal of management, 26(1), 155-169.

[9]. Beauchamp, M. R., Bray, S. R., Eys, M. A., & Carron, A. V. (2002). Role ambiguity, role efficacy, and role performance: Multidimensional and mediational relationships within interdependent sport teams. Group Dynamics: Theory, Research, and Practice, 6(3), 229.

[10]. Karau, S. J., & Williams, K. D. (1993). Social loafing: A meta-analytic review and theoretical integration. Journal of personality and social psychology, 65(4), 681.

[11]. Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. ACM Computing Surveys (CSUR), 26(1), 87-119.

[12]. Dillenbourg, P. (1999). What do you mean by collaborative learning?. Collaborative-learning: Cognitive and computational approaches., 1-19.

[13]. Johnson, D. W., Johnson, R. T., & Smith, K. (2007). The state of cooperative learning in postsecondary and professional settings. Educational psychology review, 19(1), 15-29.

[14]. De Dreu, C. K., & Weingart, L. R. (2003). Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. Journal of applied Psychology, 88(4), 741.

[15]. Fluijt, D., Bakker, C., & Struyf, E. (2016). Team-reflection: The missing link in co-teaching teams. European journal of special needs education, 31(2), 187-201.

[16]. Oakley, B., Felder, R. M., Brent, R., & Elhajj, I. (2004). Turning student groups into effective teams. Journal of student centered learning, 2(1), 9-34.

[17]. Schon, D. A., & DeSanctis, V. (1986). The reflective practitioner: How professionals think in action.

[18]. Kolb, D. A. (2014). Experiential learning: Experience as the source of learning and development. FT press.

[19]. Dewey, J. (1933). How we think: A restatement of the relation of reflective thinking to the educative process.

[20]. West, M. (1996). Reflexivity and work group effectiveness: A conceptual integration. In The handbook of work group psychology (pp. 555-579). John Wiley & Sons, Ltd.

[21]. Demmans Epp, C., Akcayir, G., & Phirangee, K. (2019). Think twice: exploring the effect of reflective practices with peer review on reflective writing and writing quality in computer-science education. Reflective Practice, 20(4), 533-547.

[22]. Li, W., Liu, C. Y., & Tseng, J. C. (2023). Effects of the interaction between metacognition teaching and students' learning achievement on students' computational thinking, critical thinking, and metacognition in collaborative programming learning. Education and Information Technologies, 28(10), 12919-12943.

[23]. Coffey, J. W. (2017, March). A study of the use of a reflective activity to improve students' software design capabilities. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (pp. 129-134).

[24]. Alhazbi, S. (2014, April). Using e-journaling to improve self-regulated learning in introductory computer programming course. In 2014 IEEE Global Engineering Education Conference (EDUCON) (pp. 352-356). IEEE.

[25]. Burden, H., & Steghöfer, J. P. (2018). Teaching and fostering reflection in software engineering project courses. In Agile and Lean Concepts for Teaching and Learning: Bringing Methodologies from Industry to the Classroom (pp. 231-262). Singapore: Springer Singapore.

[26]. Chetty, J., & van der Westhuizen, D. (2014, June). Implementing metacognition skills for learners studying computer programming. In EdMedia+ Innovate Learning (pp. 726-731). Association for the Advancement of Computing in Education (AACE).

[27]. Chen, W., Zhang, J., Lou, H., Zhou, X., & Wang, B. (2024). The effect of multiple team membership on creativity: the role of self-reflection and need for cognition. Current Psychology, 43(35), 28264-28274.

[28]. Kim, P., Hong, J. S., Bonk, C., & Lim, G. (2011). Effects of group reflection variations in project-based learning integrated in a Web 2.0 learning space. Interactive Learning Environments, 19(4), 333-349.

[29]. Nurpratiwi, S., Amaliyah, A., & Romli, N. A. (2022). Learning by Project: Develop Students' Self-reflection and Collaboration Skills Using Team-Based Project. Hayula: Indonesian Journal of Multidisciplinary Islamic Studies, 6(2), 267-284.

[30]. O'Brien, W., Soibelman, L., & Elvin, G. (2003). Collaborative design processes: an active- and reflective-learning course in multidisciplinary collaboration. Journal of Construction Education, 8(2), 78-93.

[31]. Otte, K. P., Konradt, U., & Oldeweme, M. (2018). Effective team reflection: The role of quality and quantity. Small Group Research, 49(6), 739-766.

[32]. Adelopo, I., Asante, J., Dart, E., & Rufai, I. (2017). Learning groups: The effects of group diversity on the quality of group reflection. Accounting Education, 26(5-6), 553-575.