# Software Engineering Team Project Courses

## Understanding and Supporting Teamwork and Learning

**Nayla Nasir**

Department of Software Engineering
Blekinge Institute of Technology

Licentiate Dissertation Series no. 2026:02

# Software Engineering Team Project Courses

Understanding and Supporting Teamwork and Learning

**Nayla Nasir**

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2026:02

# Software Engineering Team Project Courses

## Understanding and Supporting Teamwork and Learning

**Nayla Nasir**

Licentiate Dissertation in Software Engineering

Department of Software Engineering
Blekinge Institute of Technology
SWEDEN

Blekinge Institute of Technology
Department of Software Engineering

Blekinge Institute of Technology Licentiate Dissertation 2026:02
ISBN 978-91-7295-524-0
ISSN 1650-2140
urn:nbn:se:bth-29240

Printed in Sweden by Media-Tryck, Lund University, Lund 2026

*"Be less curious about people, and more curious about ideas"*

*-Marie Curie*

# Abstract

**Context:** Software Engineering (SE) education equips students for the complexities of the software industry, emphasizing not only technical skills but also teamwork and communication with stakeholders. SE team projects courses with industrial customers provide learning environments where students develop both technical and social skills. However, such courses often face challenges related to teamwork, accountability, and assessment.

**Objectives:** This thesis aims to support teamwork and learning in SE team project courses by i) identifying the challenges students face, ii) exploring peer evaluation as a teamwork support strategy, and iii) evaluating the perceived effectiveness of various teamwork support strategies, including peer evaluation, team contracts, and collaborative peer review.

**Methods:** This research adopts a qualitative, evidence-based approach. We analyzed student reflection reports using qualitative document analysis to capture students' experiences. Teacher focus groups were conducted to gather educators' insights. Additionally, a systematic literature review was performed to create a peer evaluation taxonomy, which was then validated via semi-structured interviews with SE educators.

**Results:** The findings show that challenges in SE team project courses are mainly socio-technical rather than purely technical. Several strategies, such as peer evaluation, team contracts, and collaborative peer review workshops, are used to support teamwork. However, peer evaluation practices in literature vary widely in their design and reporting. To address this, the thesis proposes a taxonomy of peer evaluation with guidelines for designing peer evaluation processes. The thesis also finds that both students and teachers perceive peer evaluation and team contracts as useful strategies; however, they need structured follow-up.

**Conclusion:** This thesis provides empirical insights into teamwork challenges and lessons learned. It also contributes taxonomy and design guidelines for peer evaluation. Further, it provides qualitative evidence on the perceived usefulness of different teamwork strategies. The findings highlight the importance of context-sensitive design and structured follow-up when implementing teamwork support strategies in SE team project courses.

# Acknowledgements

First and foremost, I wish to express my sincere gratitude to Allah Almighty for His guidance, blessings, and strength, which have made the completion of this work possible.

I am deeply grateful to my supervisors, Muhammad Usman and Jürgen Börstler, for their invaluable guidance, support, and encouragement throughout this research. Their mentorship has been instrumental in shaping both this work and my development as a researcher.

I would also like to thank my father for his support, prayers, and confidence in me throughout my studies.

I am especially grateful to my husband, Nasir Minhas, for his unconditional love, patience, and encouragement. His guidance, understanding, and encouragement have been my anchor throughout this journey. From managing daily challenges to celebrating milestones, he has provided both practical support and emotional strength. His belief in me, even in moments of doubt, has motivated me to persevere and complete this work. This achievement is as much his as it is mine.

Finally, I dedicate my gratitude to my children – Hadiyah, Ali, Azka, Ibrahim, and Ayza – whose love, curiosity, and encouragement have been a constant source of motivation.

# List of Papers

## Paper I

**Chapter 2: Nayla Nasir**, Muhammad Usman, Jürgen Börstler, Nina Dzamashvili Fogelström. "Software Engineering Team Project Courses with Industrial Customers: Students' Insights On Challenges And Lessons Learned" – Published in *Journal of Systems and Software 226 (2025): 112441.*

## Paper II

**Chapter 3: Nayla Nasir**, Muhammad Usman, Jürgen Börstler. "Peer Evaluation in Software Engineering Team Project Courses: A Taxonomy and Guidelines for Educators"– In revision with the *Journal of Systems and Software*, 2026.

## Paper III

**Chapter 4: Nayla Nasir**, Muhammad Usman, Usman Nasir. "Using Peer Evaluations and Team Contracts in Software Engineering Team Project Courses" –Accepted for publication in *IEEE/ACM 48th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, 2026.

## Paper IV

**Chapter 5: Nayla Nasir**, Muhammad Usman, Umar Iftikhar. "Collaborative Peer-Review Workshops with Paired Student Teams in Software Metrics Education" – Accepted for publication in *31st Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, 2026.

# Author's contribution to the included papers

We use the Contributor Roles Taxonomy (CRediT) [1] to describe the author contributions for the included papers.

## Chapter 2: Paper I

**Nayla Nasir** Conceptualization (supporting), data curation (lead), formal analysis (lead), investigation (lead), methodology (lead), visualization (lead), writing - original draft (lead), writing - review & editing (supporting)

**Muhammad Usman:** Conceptualization (lead), formal analysis (supporting), investigation (supporting), methodology (supporting), supervision (lead), visualization (supporting), writing - original draft (supporting), writing - review & editing (lead)

**Jürgen Börstler:** Conceptualization (supporting), formal analysis (supporting), investigation (supporting), methodology (supporting), supervision (supporting), visualization (supporting), writing - review & editing (supporting)

**Nina Dzamashvili Fogelström:** Writing - original draft (supporting), writing - review & editing (supporting)

## Chapter 3: Paper II

**Nayla Nasir** Conceptualization (supporting), data curation (lead), formal analysis (lead), investigation (lead), methodology (lead), visualization (lead), writing - original draft (lead), writing - review& editing (supporting)

**Muhammad Usman:** Conceptualization (lead), formal analysis (supporting), investigation (supporting), methodology (supporting), supervision (lead), visualization (supporting), writing - original draft (supporting), writing - review & editing (lead)

**Jürgen Börstler:** Conceptualization (supporting), formal analysis (supporting), investigation (supporting), methodology (supporting), supervision (supporting), visualization (supporting), writing - review& editing (supporting)

## Chapter 4: Paper III

**Nayla Nasir** Conceptualization (supporting), data curation (lead), formal analysis (lead), investigation (lead), methodology (lead), visualization (lead), writing - original draft (lead), writing - review& editing (supporting)

**Muhammad Usman:** Conceptualization (lead), formal analysis (supporting), investigation (supporting), methodology (supporting), supervision (lead), visualization (supporting), writing - original draft (supporting), writing - review & editing (lead)

**Usman Nasir:** Data curation (supporting), formal analysis (supporting), investigation (supporting), methodology(supporting), visualization (supporting), writing - original draft (supporting), writing - review& editing (supporting)

### Chapter 5: Paper IV

**Nayla Nasir** Conceptualization (supporting), data curation (lead), formal analysis (lead), investigation (lead), methodology (lead), visualization (lead), writing - original draft (lead), writing - review& editing (supporting)

**Muhammad Usman:** Conceptualization (lead), formal analysis (supporting), investigation (supporting), methodology (supporting), supervision (lead), visualization (supporting), writing - original draft (supporting), writing - review & editing (lead)

**Umar Iftikhar:** Data curation (supporting), formal analysis (supporting), investigation (supporting), methodology(supporting), visualization (supporting), writing - original draft (supporting), writing - review& editing (supporting)

## Other publications not part of this thesis

Nayla Nasir. "Towards Better Design and Delivery of SE Team Project Courses." Published in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 2.* 2024.

Sara Hooshangi, Asma Shakil, Steve Riddle, Ilknur Aydin, Nayla Nasir, Tejasvi Parupudi, Attiqa Rehman et al. "Evaluating Assessment Practices in Team-Based Computing Capstone Projects." Published in *Working Group Reports on Innovation and Technology in Computer Science Education,* 2025.

# Abbreviations

| | |
|---|---|
| CL | Collaborative learning. |
| GQM | Goal-Question-Metric. |
| LeSS | Large Scale Scrum. |
| LTPC | Large Team Project Course. |
| PBL | Project-based learning. |
| PE | Peer Evaluation. |
| PrBL | Problem-Based Learning. |
| SE | Software Engineering. |
| SoS | Scrum of Scrums. |
| STPC | Small Team Project Course. |
| TBL | Team-Based Learning. |
| TCs | Team Contracts. |

# Table of Contents

# 1 Introduction

## 1.1 Overview of the Work

Software Engineering (SE) education aims to prepare students for professional practice in software industry characterized by technical complexity [2], collaboration challenges [3], and continuous change [2, 4]. In addition to mastering programming and software development methods, SE graduates are expected to work effectively in teams, communicate with stakeholders, manage uncertainty, and apply theoretical concepts in real-world contexts [4, 5]. To address these demands, SE education has increasingly evolved from lecture-based instruction to pedagogical approaches that emphasize active learning, collaboration, and accountability [6] . Among these approaches Project-based learning (PBL), has become central [6–8], as it provides realistic settings in which students can integrate technical knowledge with professional skills.

Project-based learning [6] is a well known approach in software engineering education, providing students with opportunities to apply their technical knowledge while developing the essential soft skills required for success in the software industry. The advantages of this approach are emphasized in higher education guidelines [7–9], as it promotes teamwork, problem-solving, and critical thinking [10, 11]. As a result, project courses have become a cornerstone of SE education at both undergraduate and master's level.

Project courses vary widely in their goals, structure, duration, and stakeholders involved [12, 13], and should not be treated as a single pedagogical category. One form of project based learning is team based project courses that utilize concepts of Collaborative learning (CL) [6, 14], emphasizing engagement through peer interaction, teamwork, and shared problem solving [14].

Another variation is team project courses with external customers (i.e. companies or other partners from industry) [3, 5]. While they offer rich learning opportunities, these courses are challenging to design and facilitate [3, 12] as they closely resemble professional practice [15]. In these courses, students simultaneously need to manage technical complexity, collaborate in teams [10, 12, 16, 17], manage time and resources [10, 18], coordinate work across multiple roles [19], communicate with stakeholders, and manage ambiguous or evolving requirements [3, 18].

Existing research on SE team project courses with industrial customers has largely focused on course design and overall outcomes. Many studies describe how such courses are organized and report benefits such as increased realism [4, 12], motivation [3], and the development of soft skills such as teamwork and communication [16]. Systematic reviews [5, 6] have also mapped common course characteristics and outcomes. However, existing work has payed limited attention to how different course contexts, such as team size, team structure, or course goals shape team dynamics and the challenges students experience.

The thesis addresses this gap by analyzing students' reported challenges and lessons learned across two consecutive project courses. By comparing experiences across different course contexts, the thesis highlights how teamwork challenges evolve and how contextual factors influence students' experiences in SE team project courses. A common response to teamwork-related challenges in SE education, e.g., issues related to communication, contribution, and accountability, is the use of pedagogical strategies such as peer evaluation [20–24], team contracts [25–27], and peer reviews [28]. Although peer evaluation is widely used, our analysis showed that peer evaluation practices remain loosely defined in the literature. Approaches vary widely, and the criteria used to assess teamwork are often implicit. The thesis addresses this inconsistency by proposing a taxonomy of peer evaluation dimensions used in SE team projects, helping to clarify what aspects of teamwork are typically evaluated.

In addition, we identified a lack of studies that examine how the effectiveness of different teamwork strategies depends on the course context. The thesis therefore takes a broader perspective and investigates how team contracts and peer evaluation can be combined to support accountability and collaboration in SE project courses. Specifically, we examine how students and teachers experience peer evaluation and team contracts when implemented across different project-course contexts, highlighting the importance of contextual factors.

Finally, the thesis extends the scope of the thesis beyond teamwork management to explore how another teamwork strategy, collaborative peer review, can support students' learning of complex concepts in SE education.

The thesis utilizes concepts of project based learning, peer evaluation, team contracts, and collaborative peer review, which are discussed in Section 1.2. Research goal and objectives are described in Section 1.3. Section 1.4 presents the research questions that are investigated in this thesis. Research methods employed for the studies included in this thesis are briefly described in Section 1.5. The discussion on the validity threats is presented in Section 1.6. Section 1.7 briefly presents the findings of the included studies. In Section 1.8 we discuss the overall contributions of the thesis along with implications for SE educators. Section 1.9 presents a summary of the thesis, and also presents the research directions that we are interested to investigate in the future.

## 1.2 Background

Various reports indicate that many software engineering (SE) graduates face challenges at the beginning of their careers [29, 30], primarily because the skills acquired during university education do not fully align with the demands of the software industry [29, 31, 32]. In recent years, soft skills such as professionalism, teamwork, and communication have become increasingly valued for their relevance to industry needs [29, 31, 32]. Preparing SE graduates for professional careers therefore requires real-world experience, as classroom instruction alone is insufficient for developing essential skills such as project management, collaboration, planning, progress tracking, and effective communication [33, 34]. While traditional software engineering education can provide a theoretical foundation for these skills, students typically acquire proficiency through hands-on experience [30].

Project courses involving industrial customers are widely regarded as particularly valuable [19], as they simulate professional environments [4] and expose students to realistic constraints, stakeholder communication, and professional expectations. Such courses play a crucial role in developing the socio-technical skills that are relevant for the software industry [4, 30].

However, these courses are also considered among the most complex to design and manage [12]. Planning and delivering them poses significant challenges, including securing industrial partners [5, 19], forming effective teams, coordinating schedules, managing customer expectations, and adopting a supervision style that ensures teams make steady progress toward their objectives [30]. Several studies have noted that project courses of this nature are underrepresented in the literature, with industrial collaboration often limited or absent in reported cases [4, 5, 19, 30, 35]. Even when industrial collaborators are involved, their roles, motivations, and needs are not always clearly described [10, 19].

### 1.2.1 Project-Based Learning and Project Courses in Software Engineering

Project-based learning (PBL) [6, 10, 31] is a student-centered instructional approach grounded in three constructivist [36] principles: learning is context-specific, learners are actively engaged in the process, and goals are achieved through social interaction and knowledge sharing [37]. Kotosaki et al. [37] describe PBL as *"An active student-centered form of instruction which is characterized by students' autonomy, constructive investigations, goal-setting, collaboration, communication, and reflection within real-world practices"*. In PBL settings, students apply theoretical concepts to practical problems while working collaboratively, developing both technical and non-technical skills [35].

PBL often integrates features of other pedagogical approaches, including learning-by-doing, cooperative learning, and collaborative learning [6]. Collaborative learning, in particular, emphasizes active engagement through peer interaction, shared problem solving, and collective knowledge construction [38, 39], aligning naturally with the collaborative nature of professional software development.

In SE education, PBL is widely adopted [6, 8–10], promoting learning through active participation in realistic, practice-oriented projects. SE project courses are a primary implementation of PBL, structuring learning around project experiences [35], while team-based project courses explicitly utilize collaborative learning principles. In team based project courses, students work together to achieve shared goals, exchange knowledge, and support one another's learning.

Project courses are considered central, not peripheral, to the SE curriculum [6]. The importance of PBL is explicitly recognized in the ACM/IEEE Software Engineering Curriculum Guidelines [7], which recommend team-based project experiences as a core component of SE programs. These guidelines emphasize that graduates should be able to work effectively in teams, communicate with stakeholders, and apply engineering principles in real-world contexts. Consequently, project courses are often positioned as capstone or cornerstone experiences [5], integrating knowledge across multiple SE domains.

### 1.2.2 Types of Software Engineering Project Courses

Software Engineering project courses differ substantially in their design, scope, and learning objectives. Fincher et al. [13] provide a useful characterization of computing project courses, highlighting dimensions such as project duration, team size, degree of student autonomy, assessment strategies, and the involvement of external stakeholders. The differences in course structure, project complexity and project deliverables can have important didactical implications [10], as they shape both the intended learning outcomes and the types of challenges students face during project execution. Recognizing this variation is crucial for understanding the expectations placed on students and for designing appropriate forms of instructional support [13].

Project courses can be designed in a variety of ways [10, 12, 35], differing in duration (ranging from one - to two-semester courses), customer arrangements (from a single customer to multiple customers), modes of collaboration (from co-located to globally distributed teams), and project scope (from small, instructor-defined projects with limited ambiguity, to large-scale projects conducted with industrial customers) [12, 35].

Another important variation concerns whether project courses are offered as isolated experiences or as part of a sequence. While many programs include a single capstone project [5], some curricula organize multiple project courses across semesters [12]. Such sequences provide opportunities to study how students' challenges, learning strategies, and perceptions evolve over time. This distinction is particularly relevant for the studies presented in this thesis, which examine student experiences across multiple project courses with similar learning outcomes but different configurations.

SE team project courses with industrial customers are designed to simulate professional software development environments [4, 15], exposing students to real constraints such as incomplete requirements, evolving stakeholder needs, and coordination within teams. In this thesis, we focus specifically on team-based SE project courses with industrial customers. These courses combine technical problem solving, teamwork, and stakeholder interaction, creating rich learning opportunities. However, they also introduce significant pedagogical and organizational challenges that make them difficult to manage and facilitate.

### 1.2.3 Software Engineering Team Project Courses: Challenges and Mitigation Strategies

A substantial body of research has documented the challenges students face in SE team project courses with real customers. These challenges span technical, organizational, and social dimensions, with teamwork-related issues consistently reported as among the most difficult for students to manage.

Prior studies identify problems related to team dynamics [10, 16, 17], communication and collaboration [3, 17], conflicts within teams [12], cultural differences [12], and uneven motivation [12, 40] or participation among team members [2]. Non-motivated team members and free-riding behavior are recurring concerns, particularly in larger teams or long-running projects [2, 12, 40].

Challenges related to collaboration with industrial customers have also been reported, including limited stakeholder availability and difficulties in aligning expectations between students and external partners [10, 30]. In addition, students' personal characteristics, such as openness to change, self-reliance, and sense of responsibility, can influence how they cope with the demands of project work [2, 17, 18].

Beyond teamwork and collaboration, students often struggle with project management tasks such as effort estimation [3], as well as with technical aspects, including unfamiliar technologies and limited experience with version control systems [2, 41]. Despite these challenges, SE project courses are also associated with positive learning outcomes, including the development of soft skills, appreciation of teamwork, and improved understanding of project and process practices [10, 16–18].

Team project courses with industrial customers are challenging not only for students but also for instructors [15, 35]. They are resource-intensive and difficult to scale, as they require close monitoring of team dynamics, and continuous supervision and guidance throughout the project lifecycle [5, 15, 35]. When instructors first implement such a course, they often face a steep learning curve, making the initial semesters particularly challenging and uncertain [30].

Assessment in these courses presents additional difficulties [30, 35]. Instructors need to distinguish between team outcomes and individual contributions, which can be challenging in collaborative settings [5]. Moreover, assessment must account for a multidimensional set of learning outcomes, including both technical competencies and soft skills such as communication, collaboration, and professional responsibility [35].

Together, these findings highlight that SE team project courses are complex socio-technical learning environments. While they offer significant educational benefits, they also require effective pedagogical design to help students navigate the challenges they encounter. Given the prevalence of teamwork-related challenges in SE project courses, several pedagogical strategies have been proposed to support student teams. Some commonly used interventions are peer evaluation, team contracts [42] and collaborative peer reviews [28].

Peer evaluation involves students assessing their peers' performance and contributions toward shared project goals [24]. In SE project courses, peer evaluation has been used to promote collaboration, enhance motivation, support conflict management, and reduce free-riding behavior [20, 43]. It can also encourage students to reflect critically on team members' performance and behavior [21–23]. From an instructional perspective, peer evaluation provides teachers with insights into team dynamics that are otherwise difficult to observe [44–46].

Team contracts, also referred to as team charters or team expectation agreements, are another strategy used to support teamwork [27, 42]. A team contract is a collaboratively developed document that defines expectations, responsibilities, working practices, and behavioral norms within a team [42]. Prior studies suggest that team contracts can help reduce conflicts, minimize social loafing, and enhance accountability by making expectations explicit [25–27, 42, 47].

Although peer evaluation and team contracts have both been studied individually, relatively few studies examine their combined use or analyze how their perceived usefulness varies across different course contexts. Recent work [42] suggests that combining multiple mechanisms of teamwork-support, such as peer evaluation and team contracts, may provide complementary benefits . This observation motivates the empirical investigations in this thesis, which examine how different strategies support teamwork and learning across multiple SE project and metrics courses.

Other teamwork strategies, such as peer review[28, 48, 49] and peer feedback [50–52] have also been discussed in literature. Prior work shows that peer reviews, often considered as evaluation mechanisms can also support student learning [28, 52]. Through structured peer interaction and dialogue, students articulate their reasoning, compare alternative solutions, and refine their understanding [53, 54]. Peer review can promote active learning, critical thinking, and conceptual understanding, especially when combined with scaffolding and instructor guidance [55]. From a socio-constructivist perspective [36], learning in peer review settings is distributed across participants, with students acting simultaneously as learners and contributors to others' learning [28, 56].

## 1.3 Research Goal and Objectives

The overall goal of the thesis work is *to understand and support teamwork and learning in SE team project courses* (See Figure 1.1). We derive three objectives for our investigation based on this goal.

**Objective 1: To identify student challenges in SE team project courses**

While SE team project courses are often recognized as valuable and complex learning environments, there is a lack of empirical studies that systematically identify and synthesize the challenges students face in courses with real projects and real customers [4, 5, 19, 30]. In most studies, challenges related to different socio-technical aspects of project work and learning processes are either discussed in isolation or informally. The first objective for this thesis work was to identify a comprehensive list of challenges and lessons learned by the students in SE team project courses with real projects and real customers. This investigation was necessary to establish a foundation for further investigations to support teamwork.

**Objective 2: To explore and organize the state of the art on peer evaluation**

Peer evaluation is frequently proposed as a strategy to support teamwork in SE team project courses [5]. However, existing research lacks a structured organization of how peer evaluation is designed, implemented, and reported. This lack of structure makes it difficult for educators to understand available design choices, compare approaches, or adapt peer evaluation to their own teaching contexts. The second objective for this thesis work was to explore and organize the state of the art on peer evaluation practices reported in the literature on SE team projects. This investigation was important to have an in-depth understanding of peer evaluation as a teamwork strategy.

**Objective 3: To evaluate perceived usefulness of different teamwork strategies in SE team project courses.**

Several teamwork strategies, such as peer evaluation, team contracts, and collaborative peer reviews, are used in SE education. However, there is limited empirical evidence on how students and educators perceive these strategies in actual course settings. Specifically, little is known about how the course context and design of teamwork interventions can influence the perceived benefits of different teamwork strategies in supporting collaboration and learning. The next objective is to explore how various strategies are perceived for their usefulness in facilitating teamwork and learning in SE team project courses.

## 1.4 Research Questions

Four research questions were formulated to address the objectives outlined in the previous section. A mapping of how these objectives relate to the research questions and the chapters in this thesis is presented in Figure 1.2.

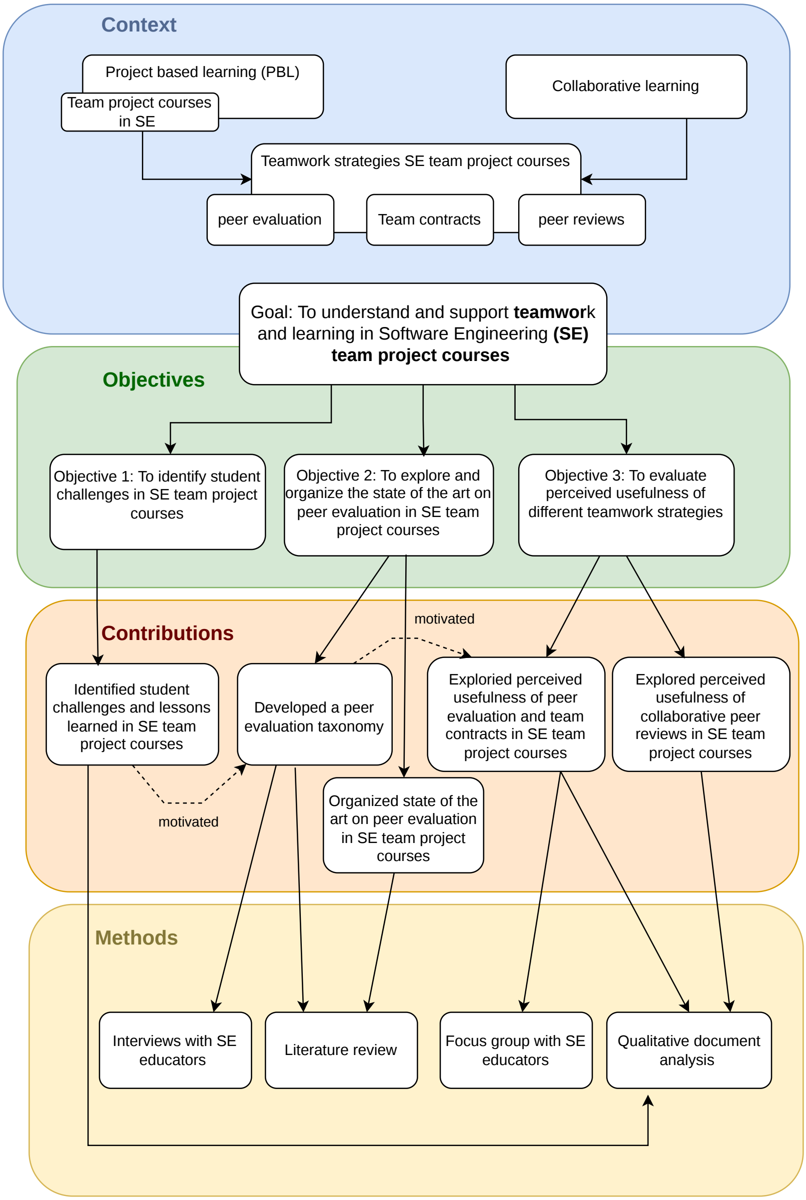



**Figure 1.1:** Overview of the thesis where context is mapped to the overall goal, objectives, contributions and methods

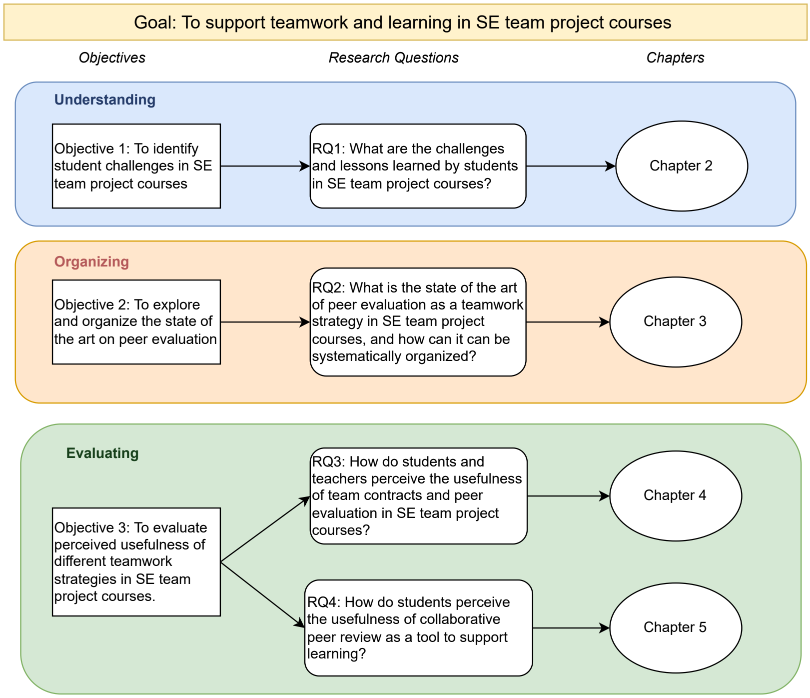



**Figure 1.2:** Mapping between thesis goal, objectives, research questions and chapters

**RQ1: What are the challenges and lessons learned by students in SE team project courses?**

**Contribution:** Identification of student challenges and lessons learned in SE team project courses. Chapter 2 provides an empirical synthesis of student challenges and lessons learned from SE team project courses, with a focus on teamwork, collaboration, coordination, and learning processes. This contribution establishes a problem-oriented foundation for the design of pedagogical interventions in SE project-based education.

**RQ2: What is the state of the art of peer evaluation as a teamwork strategy in SE team project courses, and how can it can be systematically organized?**

**Contribution:** Organization of the state of the art on peer evaluation in SE team project courses. Chapter 3 organizes existing research on peer evaluation in SE team project courses by identifying key design dimensions of a peer evaluation process. This contribution results in a structured taxonomy of peer evaluation that supports educators and researchers in understanding, comparing, and designing peer evaluation mechanisms. Based on the taxonomy, we also provided a set of guidelines for educators and researchers on how to design peer evaluation processes.

**RQ3: How do students and teachers perceive the usefulness of team contracts and peer evaluation in SE team project courses?**

**Contribution:** Evaluation of the perceived usefulness of peer evaluation and team contracts in SE team project courses. In Chapter 4, we analyze reflections of students and educators, to understand how useful peer evaluation and team contracts are considered when implemented in different SE team project courses. This contribution highlights how contextual factors, such as team size and team structure, shape the usefulness of these strategies in supporting teamwork and accountability.

**RQ4: How do students perceive the usefulness of collaborative peer review as a tool to support learning?**

**Contribution:** Evaluation of the perceived usefulness of collaborative peer reviews in student learning. Chapter 5 investigates the use of collaborative peer reviews as a teamwork and learning strategy in SE team project courses. While teamwork and peer review are widely studied separately in SE education, their integration within a structured, time-bounded workshop workflow remains underexplored. Chapter 5 demonstrates how structured peer interaction and peer feedback can support learning of complex concepts in SE education.

## 1.5 Research Methodology

**Table 1.1:** Overview of Research Methods Across the Four Studies

| Study | Research focus | Data sources | Data collection | Data analysis |
|---|---|---|---|---|
| Study 1 | Identify students' challenges and lessons learned in SE team project courses with real customers | Student reflection reports from multiple course iterations | Qualitative document analysis | Thematic analysis |
| Study 2 | Understand how peer evaluation is conducted in SE team project courses, and propose a structured way to organize this body of knowledge | Research papers, Interviews with SE educators | Systematic literature review, semi-structured interviews | Thematic analysis, Iterative taxonomy development |
| Study 3 | Investigate students' and teachers' perceptions of peer evaluation and team contracts in small and large team project courses | Student reflection reports from two courses; Focus group with SE educators | Qualitative document analysis, Focus group discussion | Thematic analysis |
| Study 4 | Investigate perceived usefulness of collaborative peer-review workshops for learning of the GQM framework | Student reflection reports from two course iterations | Qualitative document analysis | Thematic analysis |

We have adopted a mixed-method, qualitative approach that incorporates various data collection and analysis methods (see Table 1.1 for details). A common characteristic of all studies is their empirical foundation in real courses and teaching practices. The researcher plays a dual role as both educator and investigator, allowing for close engagement with course implementation and student experiences.

The research follows a structured design around four studies (presented as chapters 2-5 of this thesis), each addressing a specific research objective while building on the insights of previous work. The studies collectively move from problem identification, to conceptual organization and empirical evaluation of pedagogical strategies (see Figure 1.2).

Study 1 adopts qualitative document analysis to identify challenges and lessons learned in SE team project courses. Study 2 employs a systematic literature review and iterative thematic analysis to organize existing knowledge on peer evaluation. Study 3 uses qualitative document analysis to investigate student and teacher experiences of peer evaluation and team contracts in practice. Study 4 applies a qualitative intervention-based approach to explore collaborative peer review as a mechanism for supporting learning in SE team projects. A detail of research methods used in each study is presented in the following sub sections.

### 1.5.1 Educational Context

All empirical studies are conducted in the context of SE team project courses at university-level. The courses are characterized by project work, teamwork, and the application of SE methods and tools. Most courses (except the one in Study 4) involve collaboration with real customers increasing the authenticity and complexity of the learning environment.

The courses vary in terms of team size, team structure, and project scope, allowing the research to explore how course context can influence teamwork challenges and the perceived usefulness of pedagogical strategies. These variations can support analytical generalization across similar SE educational contexts.

### 1.5.2 Data Collection Methods

Multiple qualitative data sources are used across the four studies to capture both student and teacher perspectives.

#### *1.5.2.1 Qualitative Document Analysis*

In studies 1, 3, and 4, qualitative data are collected from student reflection reports using qualitative document analysis [57]. Students are asked to reflect on their experiences of teamwork, collaboration, assessment, and learning throughout the project courses. Structured prompts are used to support reflection on challenges, learning outcomes, and perceived usefulness of pedagogical activities. These reflections provide insight into students' experiences and allow the research to capture socio-technical challenges faced by the students.

#### 1.5.2.2 Literature Review

Study 2 employed a literature review to investigate how peer evaluation is designed, implemented, and reported in software engineering (SE) team project courses. The review process consisted of two main stages: study search and data extraction and analysis.

*Study search:* In the study search stage, we followed a two-iteration approach. First, we built on the systematic review by Tenhunen et al. [5], which identified 39 studies published between 2007 and June 2022 on peer evaluation in SE project courses. We selected 37 of these studies, and two studies focusing only on self-evaluation were excluded. In the next iteration, we extended the dataset by searching Scopus for publications from 2022–2025 using the same search strings as our base study [5]. After screening abstracts and applying inclusion criteria based on relevance to peer evaluation, 13 additional studies were selected, resulting in a final set of 50 primary studies.

*Data extraction and analysis:* In the data extraction and analysis stage, we designed a data extraction form to systematically collect information on study characteristics, course context, and peer-evaluation practices. To ensure consistency and reliability, three authors independently extracted data from a subset of studies, compared results, and resolved discrepancies through discussion. The analysis revealed a lot of variation in how peer evaluation practices were reported, which highlighted the need for a structured taxonomy to enable consistent description, comparison, and classification across studies.

#### 1.5.2.3 Interviews with SE Educators

For Study 2, qualitative data are collected through semi-structured interviews with SE educators. We chose semi-structured interviews because they provide flexibility for improvisation and deeper exploration of the research phenomena [58], while also enabling the collection of detailed insights related to the topic under study [59]. We conducted seven interviews with SE educators teaching SE team project courses across different institutions in the world. These interviews aimed to capture the perspectives of educators on peer evaluation practices. Educators were asked to characterize their own peer evaluation setups using the taxonomy and to provide feedback on its completeness, clarity, and applicability. The feedback was incorporated into the final refinement of the taxonomy. An interview guide was developed and pilot tested before conducting the interviews.

#### 1.5.2.4 Focus Groups with SE Educators

For Study 3, qualitative data are collected through focus group discussions with SE educators, in addition to using the student reflection reports. These data capture educators' experiences with peer evaluation and team contracts, including practical constraints, design considerations, and implementation experiences.

### 1.5.3 Data Analysis

Across the studies, qualitative data are analyzed using thematic analysis [60, 61]. The analysis is iterative and reflexive [62], involving *i) familiarization with the data, ii) generation of initial codes, iii) generation of themes and iv) Refinement and naming of themes v) interpretation and reporting.* The frequency of each code was also noted to identify the codes and themes that were reported most frequently. For Study 1, the analysis focuses on identifying challenges and lessons learned by the students in SE team project courses. In Study 2, analysis focuses on identifying peer evaluation dimensions and reporting gaps in the literature. The interviews were conducted with the instructors for the course characterization and taxonomy validation were also thematically analyzed. In Study 3 and 4, analysis emphasizes identifying patterns in perceived usefulness of pedagogical interventions used. To enhance credibility, analysis is conducted collaboratively among authors, and emerging interpretations are discussed and refined through iterative cycles.

## 1.6 Threats to Validity

This thesis employs an empirical research approach grounded in qualitative and interpretive methods within SE educational contexts. The included studies use multiple data sources, including student reflection reports, a literature review, interviews, and focus groups. Since much of the research aims to understand student and educator experiences in real learning environments, the validity concerns identified by Maxwell [63] are particularly relevant.

Following the framework proposed by Petersen and Gencel [64], threats to validity are discussed in terms of theoretical validity, descriptive validity, interpretive validity, and generalizability. This framework is suitable for interpretive empirical SE research, where the goal is to provide credible and transparent interpretations of phenomena within their natural context rather than to establish universal causal relationships. The identified threats were considered throughout the design, data collection, and analysis phases of the studies included in this thesis.

### 1.6.1 Theoretical Validity

Theoretical validity concerns whether the constructs investigated in the studies are well defined and whether the collected data appropriately represents the phenomena under investigation.

Across the studies in this thesis, the primary constructs include student challenges in SE team project courses, peer evaluation practices, and the perceived usefulness of teamwork support mechanisms such as peer evaluation, team contracts, and collaborative peer review. A potential threat is that these constructs may be interpreted differently by participants or may not fully capture the complexity of collaborative learning environments.

In Study 1, challenges and lessons learned were identified from student reflection reports. A risk here is that students may interpret these constructs differently or emphasize aspects that they perceive as important. This threat was mitigated by using structured reflection reports that explicitly asked students to report challenges and lessons learned. Furthermore, the analysis focused on recurring themes across multiple reports to avoid overemphasizing isolated experiences.

In Study 2, which develops a taxonomy of peer evaluation practices, theoretical validity may be affected by incomplete or inconsistent reporting in the primary studies included in the literature review. Since the taxonomy relies on information reported in existing publications, some aspects of peer evaluation practices may not have been fully described. This threat was addressed by explicitly documenting missing information and treating gaps in reporting as a finding of the review. In addition, the taxonomy was validated through interviews with SE educators to assess whether the identified dimensions align with teaching practice.

In Studies 3 and 4, the main constructs relate to the perceived usefulness of teamwork strategies and collaborative learning interventions. These constructs were intentionally defined as subjective perceptions rather than objective effectiveness measures. Data sources, including student reflection reports and educator focus groups, were therefore well aligned with the interpretive nature of the research questions.

Overall, theoretical validity is supported by clear definitions of constructs, careful alignment between research objectives and data sources, and transparency regarding the scope and limitations of the investigated concepts.

### 1.6.2 Descriptive Validity

Descriptive validity refers to the accuracy and completeness with which the collected data represents the observed phenomena.

In three of four studies included in this thesis, student reflection reports were used as a primary data source. A potential threat is that students may omit certain experiences or selectively report events that they perceive as important or socially acceptable. To mitigate this risk, the reflection reports were based on structured templates that guided students to report specific aspects of their experiences. In addition, the reflections were written as part of course activities prior to the research study, reducing the likelihood that the content was influenced by the research process.

Another risk relates to the accurate representation of reported challenges and experiences during data analysis. In Study 1, measures were implemented to avoid over-reporting of challenges or lessons learned. If a student mentioned the same challenge multiple times within a single report, it was coded only once to ensure accurate representation. The analysis also accounted for both the number of unique reports and the number of occurrences of each theme, providing a more accurate overview of the prevalence of different challenges.

In Study 2, descriptive validity relates to the accurate extraction and documentation of information from primary studies. This threat was mitigated by following a systematic review protocol, applying clearly defined inclusion criteria, and documenting the data extraction process. Multiple authors were involved in reviewing extracted data and discussing taxonomy development decisions.

Across the qualitative studies, data management and analysis tools were used to preserve the context of the collected data and maintain traceability between raw data and coded themes. This helped ensure that the findings accurately reflect the data collected from participants.

### 1.6.3 Interpretive Validity

Interpretive validity concerns whether the interpretations drawn from the data accurately represent the perspectives and experiences of the participants.

Since the studies in this thesis rely on qualitative analysis of reflections, interviews, and focus group discussions, researcher interpretation plays a central role in deriving findings. A potential threat is that the researchers' prior experiences or expectations may influence the interpretation of the data.

To mitigate this risk, established qualitative analysis guidelines were followed, particularly thematic analysis approaches recommended in SE research [60, 61]. Coding and theme development were conducted iteratively, and interpretations were grounded in the original data to maintain a clear connection between empirical evidence and reported findings.

In several studies, analysis was conducted collaboratively among multiple authors. Discussions among researchers helped challenge assumptions and refine interpretations, thereby reducing the likelihood of individual researcher bias influencing the results.

Another potential threat is selective interpretation of data. To address this, the analysis aimed to systematically code all relevant sections of the collected artifacts rather than focusing only on examples that support expected findings. Maintaining traceability between coded data and final themes further helped ensure that interpretations were grounded in the empirical material.

Finally, the role of the researchers as educators within the studied courses may introduce potential bias. While this involvement provided valuable contextual understanding of the course environment, it may also influence interpretation of results. This risk was mitigated through collaborative analysis, transparent reporting of the research process, and grounding interpretations in the collected data.

### 1.6.4 Generalizability

Generalizability refers to the extent to which the findings can be applied beyond the specific contexts studied.

Studies 1, 3, and 4 were conducted within SE team project courses at a single institution. This context-specific setting may limit the generalizability of the findings to other educational environments. However, many SE programs include similar project-based courses that involve teamwork, real or simulated projects, and formative assessment activities. Detailed descriptions of the course contexts, project characteristics, and team structures are therefore provided to allow readers to assess the applicability of the findings to their own contexts.

Study 2 strengthens the generalizability of the thesis by synthesizing evidence from a large number of primary studies conducted across different institutions, countries, and course designs. The taxonomy of peer evaluation practices developed in this study is therefore grounded in a broad range of educational contexts. Furthermore, the taxonomy was validated through interviews with SE educators from multiple institutions, supporting its relevance beyond the reviewed literature.

Across the thesis, the findings aim for analytical rather than statistical generalization. The goal is not to claim universal applicability, but to provide insights that may inform the design of SE team project courses and teamwork support strategies in contexts with similar characteristics.

## 1.7 Study Results

This section presents the results of the four studies included in this thesis. Together, the studies present the challenges students face in SE team project courses and explore different strategies to support teamwork and learning in these courses.

### 1.7.1 Study 1: Challenges and lessons learned in Software Engineering Team Project Courses

Study 1 establishes the empirical foundation of the thesis by identifying challenges and lessons learned in SE team project courses involving real customers.

A key finding of the study is that the majority of challenges and lessons learned are related to soft skills rather than technical or disciplinary knowledge. More than ninety percent of the reported challenges, and over two-thirds of the reported lessons learned, concern aspects such as teamwork, communication, collaboration in remote and hybrid settings, and interaction with industrial customers. These findings suggest that SE team project courses function not only as opportunities for applying technical knowledge, but as learning environments for developing social competencies. Among all identified themes, working in a team was most frequently reported. Students reported challenges related to communication, uneven contributions within team, and forming a team with unfamiliar team members. These challenges are more pronounced in large teams and in settings where students have limited prior experience working together. At the same time, teamwork is also the theme associated with the largest set of lessons learned. Students reflect on the importance of effective communication, being respectful to different perspectives, regular team meetings, and an active effort to maintain positive team dynamics.

The study also highlights working with industrial customers as both a major challenge and a valuable learning opportunity. Students struggle with understanding and negotiating requirements, managing scope changes, and aligning expectations with external stakeholders. However, these challenges lead to lessons related to maintaining continuous communication with customers, asking clarifying questions, and adopting visual and interactive approaches to requirements discussions. Working with real customers is perceived as highly motivating and as providing insights into professional practice that are difficult to achieve in simulated settings.

Another important contribution of the study is the analysis of how challenges and lessons learned evolve across a sequence of project courses. By comparing students' reflections from a Small Team Project Course (STPC) and a later Large Team Project Course (LTPC), the study shows that many challenges decrease over time as students gain experience. However, teamwork-related challenges increase in the later course, likely due to larger team sizes and more complex sub-team structures. This also highlights how participation in multiple team project courses with external customers benefits students in progressively adapting to expectations of working on a real software project.

Taken together, the findings suggest that SE team project courses with industrial customers are powerful learning environments, but also inherently demanding. The challenges students face are mostly soft aspects, and most of the lessons learned also concern how to collaborate, communicate, and reflect in complex, real project settings.

### 1.7.2 Study 2: Peer Evaluation in Software Engineering Team Project Courses: A Taxonomy and Guidelines for Educators

The second study focuses on peer evaluation as a teamwork strategy to support challenges around teamwork, including contribution, communication, and team dynamics. We conducted a literature review and a set of interviews with SE educators to investigate how peer evaluation is currently designed, implemented, and reported in SE education research and practice.

Review of the existing literature revealed a variation in how peer evaluation processes are designed and reported. Peer evaluation is most often used for assessing individual contributions and sometimes to enhance student learning or understand team dynamics. Most reported peer evaluation setups involve individuals evaluating their team members and/or themselves, typically using survey-based tools. However, critical design aspects, such as evaluation criteria, frequency, format, and use of peer evaluation results, are frequently underreported. Also, implications of contextual factors such as team size, course duration, and course level on design of peer evaluation processes are rarely discussed, despite their clear relevance. This gap motivated the design of a peer evaluation taxonomy.

The proposed taxonomy organizes peer evaluation along multiple dimensions, including course context, purpose, participants, mechanism, and use of peer evaluation output. Rather than prescribing a single "best" peer evaluation setup, the taxonomy makes explicit the available design choices and their associated trade-offs. Validation of the taxonomy through educator interviews shows that the taxonomy can be used to characterize the peer evaluation processes in practice and supports reflection on why particular design decisions are made in specific contexts. Educators emphasize that peer evaluation design is strongly shaped by course context, including team size, project duration, and student maturity. They report more frequent use of rubrics covering both technical and soft contributions, greater awareness of trade-offs between different design choices, and more systematic use of peer evaluation results, often triangulated with other data sources, when used for grading.

Based on the taxonomy, the study provides design guidelines that guide educators through key decisions involved in implementing peer evaluation. These guidelines emphasize that there is no one-size-fits-all solution and that peer evaluation must be aligned with course goals, team structure, and ethical considerations. The findings highlight that poorly aligned peer evaluation can create tension, bias, or administrative burden, while well-designed peer evaluation can promote accountability, reflection, and learning.

### 1.7.3 Study 3: Using Peer Evaluations and Team Contracts in Software Engineering Team Project Courses

The third study moves from conceptual design to empirical investigation of pedagogical interventions in practice. Building on the taxonomy and design considerations introduced in Chapter 3, this study examines how peer evaluation and team contracts are perceived and experienced by students and teachers in two different course contexts: a small-team project course (STPC) and a large-team project course (LTPC). Using thematic analysis of student reflection reports and a teacher focus group, the study reveals that both interventions are valued, but their perceived usefulness, and limitations vary strongly with team size and course structure.

Peer evaluation was largely perceived as beneficial in small teams. Many students reported that it helped make individual contributions visible, promoted accountability, and encouraged reflection on teamwork and personal performance. Anonymity played a particularly important role in this context, as it gave students a sense of safety to provide honest feedback without harming team relationships. For some students, peer evaluation directly influenced their behavior, motivating them to communicate more clearly or engage more actively.

At the same time, peer evaluation in small teams was not without challenges. Some students raised concerns about fairness, the risk of biased judgments, and the possibility that peers might provide superficial or rushed feedback. Other students also felt that peer evaluation had limited impact on actual team performance, suggesting that awareness and reflection do not automatically translate into behavioral change without further support.

In large teams (where each team consisted of sub-teams), students' perceptions of peer evaluation were noticeably more mixed. While some students appreciated peer evaluation as a way to provide teachers with insights into individual contributions and overall team dynamics, many reported that they lacked sufficient information to fairly evaluate peers outside their own subteams. This lack of visibility was a dominant limitation and contributed to a widespread perception that peer evaluation had little or no impact on team performance in the large-team context. Issues such as peer evaluation results not being shared, limited transparency, and unclear purpose further reduced its perceived value.

Across both contexts, students emphasized the importance of sharing results and follow-up discussions. Without feedback loops or opportunities to reflect collectively, peer evaluation risked becoming a procedural exercise rather than a learning-oriented activity.

Teachers viewed peer evaluation as a tool that can promote accountability and give students a voice, but they consistently emphasized that its value depends on teacher-led interpretation and follow-up. Peer evaluation data was seen as a starting point for conversations, not as evidence on its own. Teachers also raised concerns about fairness, misinterpretation, and the limited suitability of peer evaluation in large teams with subteam structures, questioning whether it should be used at all in such contexts unless significantly adapted. Taken together, the findings suggest that peer evaluation works best in relatively small, cohesive teams, where students have sufficient visibility into each other's work and where teachers actively support interpretation and follow-up.

The study further investigated the perceived usefulness of team contracts in SE team project courses. Team contracts were generally perceived useful by students and teachers in both small and large team contexts. Students reported that contracts helped establish shared expectations, clarify responsibilities, and promote professionalism and accountability at the beginning of the project. Teachers similarly described contracts as useful team-building tools and effective starting points for collaboration.

However, a recurring limitation was that team contracts tended to lose relevance over time. Many students reported that contracts were gradually forgotten and had little influence on day-to-day work once the project progressed. This limitation was especially pronounced when contracts were treated as one-time artifacts rather than revisited or enforced.

Students' suggestions for improvement pointed toward making team contracts living documents. In small teams, students suggested revisiting contracts periodically, for example during sprint retrospectives. In large teams, students emphasized the need to adapt contracts to evolving team needs, include concrete working standards, and introduce clearer mechanisms for accountability. Teachers also suggested that follow-up should be explicitly planned and led by course staff, rather than delegated or postponed.

Overall, the findings indicate that team contracts can support early alignment and accountability, but their long-term impact depends on ongoing reinforcement and integration into course activities.

### 1.7.4 Study 4: Collaborative Peer-Review Workshops with Paired Student Teams in Software Metrics Education

The fourth study extends the focus of the thesis by employing collaborative learning and peer review across student teams to facilitate learning of complex concepts. Two structured workshops were introduced in consecutive course iterations. The first workshop focused on constructing Goal-Question-Metric (GQM) trees, while the second supported students in synthesizing metrics data to answer measurement questions and assess goals. Both workshops were actively facilitated, case-based, and designed around two-stage collaboration: intra-team work followed by inter-team peer review.

Analysis of student reflection reports from two course iterations shows that the workshops were widely perceived as useful, with consistent patterns of benefits across cohorts. Students appreciated both intra-team collaboration and inter-team peer review. Presenting their work to other teams and receiving structured feedback helped them identify weaknesses, refine their reasoning, and consider alternative analytical perspectives. Several students explicitly noted that exposure to different approaches and questions from peers sharpened their thinking and improved the quality of their analysis. These findings suggest that peer review, when carefully structured and facilitated, can function as a learning mechanism rather than merely an assessment activity.

## 1.8 Overall Contributions and Implications

Each chapter in this thesis represents an individual study that is logically connected to the others. The logical flow and interconnections among the studies are presented

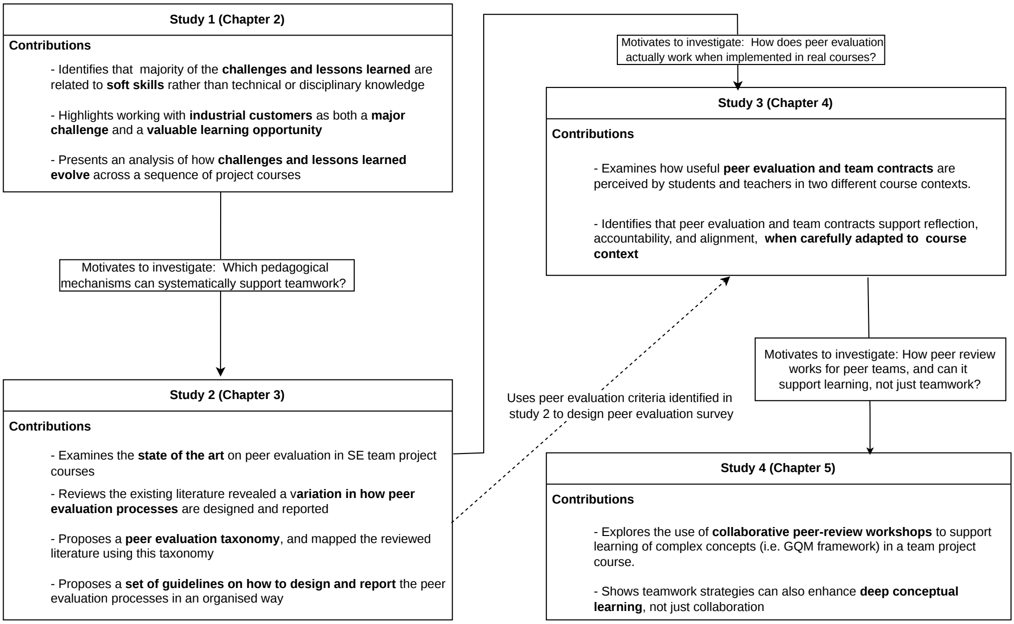



**Figure 1.3:** The logical flow of how studies in the thesis connect to each other

in Figure 1.3.

The first study provides a foundational understanding of challenges faced by students in SE team project courses. The results show that students encounter a range of challenges that extend beyond technical problem-solving and are largely socio-technical in nature. By mapping challenges to corresponding lessons learned, the study provides insight into how students attempt to mitigate challenges they face in SE team project courses. Overall, the study establishes a need to investigate teamwork strategies and pedagogical mechanisms that can support teamwork in SE team project courses.

The second study addresses the need for a more systematic understanding of peer evaluation as a teamwork strategy in SE education. The findings reveal that although peer evaluation is widely adopted, its design varies significantly across courses and is often inconsistently reported in the literature. This lack of transparency limits both its pedagogical effectiveness and its research value. To address this gap, the study develops a taxonomy that organizes peer evaluation along key design dimensions, accompanied by a set of walk through style guidelines for educators and researchers. Together, these contributions support a more structured understanding of peer evaluation and can help educators make informed, context-sensitive design decisions. The taxonomy can also support researchers in consistent reporting and comparison of peer evaluation mechanisms.

The third study moves from conceptual design to empirical investigation of peer evaluation and team contracts as teamwork strategies implemented in real SE team project courses. The findings highlight that interventions such as peer evaluation and team contracts can support reflection, accountability, and early alignment within teams, but only when they are adapted to course context and supported through follow-up and facilitation. Synthesizing these findings with the earlier studies reinforces a central theme of the thesis: there is no universally effective teamwork strategy for SE project courses. Instead, the usefulness of peer evaluation and team contracts depends on how they are designed, implemented, and aligned with course context. This study thus empirically demonstrates the importance of context-sensitive didactical design, building directly on the conceptual framework developed in the second study.

The fourth study extends the scope of the thesis to utilize teamwork strategies to support students' learning of complex SE concepts. It explores the idea of collaborative peer-review workshops to provide structured opportunities for students to discuss, critique, and refine their work with peers. The findings indicate that these collaborative activities support learning by making students' reasoning visible, exposing them to alternative perspectives, and encouraging deeper reflection. Through peer feedback and facilitated discussion, students move beyond superficial application of concepts toward more analytical and integrated understanding.

Taken together, these findings demonstrate that supporting teamwork and learning in SE team project courses requires careful, context-sensitive didactical design grounded in empirical understanding of student challenges. These insights have following implications for both SE education research and teaching practice.

- Teamwork challenges are common in software engineering project courses. These challenges cannot be addressed by assuming that teamwork skills will develop implicitly through project participation. Effective mitigation requires intentional instructional support and structured mentorship built into the course design.

- Pedagogical strategies such as peer evaluation, team contracts, and collaborative peer reviews have the potential to support teamwork and learning, but they are not inherently effective. Their usefulness depends on careful design choices, alignment with course context, and opportunities for facilitation and reflection.

- Course context is important. Factors such as team size, team structure, visibility of individual contributions, and course organization can influence how teamwork strategies are experienced. There is no universal solution, instead, educators must make informed, context-sensitive design decisions.

- Collaborative and feedback-oriented strategies can serve a dual role in SE project courses. They can be used to support teamwork as well as students' understanding of complex SE concepts. When designed thoughtfully, such strategies contribute to deeper learning and reflection.

## 1.9 Summary and Future Work

The thesis establishes that the SE team project courses are didactically complex environments in which students face challenges related to teamwork, accountability, assessment, and knowledge application. These challenges are socio-technical in nature and cannot be addressed through technical instruction alone.

The licentiate work shows that students' challenges and learning needs evolve over time, particularly as they progress through successive project courses. Many of these challenges relate to the soft aspects of software engineering, such as collaboration, communication, and shared responsibility within teams. To better understand how such challenges can be addressed, this research first examined the state of the art on peer evaluation as a teamwork strategy in SE project courses and developed a taxonomy to systematically describe its design and implementation. Subsequent studies then investigated the perceived usefulness of several teamwork support strategies, including peer evaluation, team contracts, and collaborative peer-review workshops. The findings indicate that while these strategies are generally perceived as beneficial, their usefulness depends on how well they are adapted to the specific course context.

The PhD research will further explore how different teamwork strategies are employed to develop teamwork competencies, reflective practices, and professional skills in academic and industry settings. Future work will also aim to move from exploring individual pedagogical mechanisms to propose mentorship solutions for SE team project courses. Building on peer evaluation taxonomy and other empirical findings, PhD research will investigate how multiple strategies, such as peer evaluation, team contracts, and structured reflections, can be combined to support both teamwork and learning in a coherent manner. The goal is to support SE educators in managing and facilitating project courses. In summary, the licentiate thesis lays the groundwork for a PhD research that advances Software Engineering didactics through evidence based and theoretically informed research. By continuing to investigate teamwork, collaboration, and learning in real SE project courses, the PhD project aims to contribute to both educational research and teaching practice in Software Engineering.

# 2 Software Engineering Team Project Courses with Industrial Customers: Students' Insights On Challenges And Lessons Learned

*Nayla Nasir, Muhammad Usman, Jürgen Börstler, Nina Dzamashvili Fogelström*
*Published In: Journal of Systems and Software, 2025*

## Abstract

Team project courses in software engineering allow students to apply their acquired disciplinary knowledge while developing essential skills needed to work in the software industry. This paper examines the challenges and lessons learned by students in two team project courses involving industrial customers. The first course involves small teams and less complex project, whereas the second course, has larger teams and more complex projects. Using thematic analysis, we analyzed 158 reports submitted by two cohorts of students across two successive team project courses. As per our findings most challenges and lessons learned pertain to soft skills, such as teamwork, working in remote and hybrid setting, and collaboration with industrial customers. The results show that challenges and lessons learned evolve as students progress to the second team project course, for example, managing changes and addressing individual skill gaps were more pronounced in the first project course, while students reported greater coordination, communication, and contribution issues in the second team project course. The alignment between the challenges faced and the lessons learned suggests that addressing challenges in teamwork, collaborating with industrial customers, and working in hybrid or remote settings helped students develop effective strategies to mitigate these challenges. This process offers a valuable learning experience for the students, enriching their professional growth.

## 2.1 Introduction

Project-based learning (PBL) is a well-established approach in software engineering education, offering students opportunities to apply their disciplinary knowledge while also developing essential soft skills necessary for success in the software industry. The benefits of this approach are highlighted in higher education guidelines [8, 9], as it fosters teamwork, problem-solving, and critical thinking skills [10, 11]. Project-based courses grounded in PBL not only engage students in hands-on learning but also prepare them to address real-world challenges through collaborative and practical experiences.

Project-based courses can have several designs [10, 12], varying in duration (single- to two-semester), client structure (single to multi-customer), working mode (local to globally distributed), and project scale (toy projects to projects with industrial customers) [12].

According to the ACM/IEEE's *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering* [7], it is essential to incorporate real-world elements into the software engineering curriculum. Collaboration with industrial customers keeps the students motivated [10] and offer a more realistic experience of the practical aspects of a software engineering projects[4, 15, 19, 65]. Students cannot learn to manage complexity and change if they simultaneously act as the client, project manager, and at the end responsible for system acceptance [12]. Without an external customer, students miss the opportunity to understand and negotiate requirements, manage scope changes, and experience the sense of danger and excitement inherent in real-world projects [15]. In certain educational settings, the role of a customer is simulated by teachers or internal stakeholders [5]. According to Cico et al. [4, Sect. 4.2.2], such a *"simulation is only partially effective due to the absence of real external pressure."*

Project-based courses are inherently resource-intensive, requiring continuous feedback and mentorship [5, 15, 35], making them difficult to scale. Furthermore, teachers must devise effective methods for assessing both soft skills and individual performance within teams [5]. Involving industrial customers in project courses adds several other challenges for all stakeholders involved [19]. Risks identified in such collaborations include time, effort, and resources that external stakeholder can commit, as well as misalignment between the expectations of external stakeholders and teachers [65]. Students also face challenges, such as managing teamwork [3, 10, 16], collaborating with stakeholders [10], addressing gaps in technical knowledge [2, 18], and dealing with terminology mismatches between stakeholders and teachers [65].

Though challenging, working on team projects with industrial customers can provide invaluable lessons and learning opportunities for the students [4, 19]. These courses can bridge the gap between theoretical knowledge and its practical application while fostering the skills required to work in the software industry [65].

This study contributes to the existing body of knowledge by analyzing the challenges and lessons learned by students working on software projects with industrial customers. The data for this study was collected from two project courses involving industrial customers. In this context, industry collaboration serves as a tool to achieve the expected learning outcomes of the courses. Both courses emphasize developing soft skills and equipping students with industry-relevant competencies, as detailed in Section 2.3. Our findings can be insightful for designing project courses in similar educational contexts. The contributions of the study are listed below:

**C1:** Identifying challenges students face in team project courses with industrial customers.

**C2:** Identifying lessons learned by the students while addressing the challenges in team project courses with industrial customers.

**C3:** Analyzing the changes in students' challenges and lessons learned as they progress from one team project course with industrial customers to the next.

We present related work in Section 2.2. Section 2.3 shares information related to the background and context wherein the study is conducted. Section 2.4 describes the research questions and methods we used for collecting and analyzing the data. Sections 2.5 and 2.6 presents the study results, which are further discussed in Section 2.7. Section 2.8 describes the threats to validity of the study. Finally, we present the conclusion in Section 2.9.

## 2.2 Related Work

This section discusses studies reporting challenges and lessons learned in SE team project courses. Studies highlighting collaboration with industrial customers were prioritized, as this aligns closely with the context of the courses analyzed in our study. Several studies have reported the challenges faced by students in team project courses. These challenges (see Table 2.1 for details) include issues related to teamwork (e.g., team dynamics), collaboration with industrial customers, students' personal traits (e.g., openness to changes), project management, and technical knowledge required in a project.

Besides challenges, the literature also reports lessons learned by students participating in team project courses. Team project courses have been found to benefit students in acquiring soft skills, realizing the importance of teamwork, working with an industrial customer, and understanding and practicing important project and process skills. Table 2.2 provides more details about these lessons and maps them with the relevant studies.

Despite its recognized importance, there is little research on collaboration with industrial customers in team project courses [4, 5, 19]. Paasivaara et al. [19] investigated the perspectives of industrial collaborators in team project courses, revealing

**Table 2.1:** Challenges in team project courses presented in the literature.

| Key Aspect | Challenge | Reported by |
|---|---|---|
| Teamwork | Team dynamics | [10, 16, 60] |
| | Communication and Collaboration | [3] |
| | Conflicts in team | [12] |
| | Cultural differences | [12] |
| | Non-motivated team members | [2, 5, 12, 40] |
| Industrial customer collaboration | Stakeholder accessibility | [10] |
| Personal aspects | Openness to change | [17, 18] |
| | Self-reliance and responsibility | [2, 10] |
| Project related aspects | Effort estimation | [3] |
| Technical knowledge | Lack of technical knowledge | [2] |
| | Lack of experience in version control systems (Github) | [41] |

**Table 2.2:** Lessons learned in team project courses presented in the literature.

| Key Aspect | Lesson Learned | Reported by |
|---|---|---|
| Soft skills | Perceived value of soft skills | [10, 16–18] |
| Teamwork | Collaboration within team | [10, 12, 16, 17] |
| | Having roles within team | [10, 12] |
| | Freedom to select team members | [12, 18] |
| Industrial customer collaboration | Perceived value of working with an industrial customer | [10, 18] |
| | Negotiation with the customer | [16] |
| | External communication | [17] |
| Project and process related skills | Project management | [10, 18] |
| | Importance of iterative development | [66] |
| | Importance of documentation | [3, 10] |
| | Continuous testing and quality assurance | [3, 10] |
| | Software configuration management and version control | [3, 10, 41] |

a lack of evidence regarding effective collaboration practices. Additionally, Ståhl et al. [10] identified a gap in the literature concerning the needs and motivations of industry collaborators in these courses. Tenhunen et al. [5] observed the absence of industrial customers in many project courses in their recent systematic literature review.

Similarly, in their mapping study, Cico et al. [4] observed limited engagement from industrial stakeholders in team project courses, with only 26 out of 126 studies reporting such participation.

Murphy et al. [67] reported their experience conducting real projects with real clients in a sequence of project courses. However, the two courses in their study are offered in different programs at two distinct levels—undergraduate and graduate. Two separate groups of students participate in these courses. The first group develops the software projects in the first course, while the second group focuses on evaluation and maintenance tasks for these projects in the subsequent course. The clients for these courses are internal (faculty or graduate students from the same university). Moreover, their analysis primarily addresses the challenges of scaling these project courses and working with an existing code-base in the second course.

In contrast, our study examines a different context: two student cohorts participating in a sequence of two team project courses with industrial clients. This setup provides a unique opportunity to investigate the challenges students and lessons learned by the students and these evolve between the two courses. Details of the courses and their contexts are described in the next section.

## 2.3 Context and Background

This study was conducted in the context of the Bachelor of Software Engineering program at Blekinge Institute of Technology. A distinctive feature of this three-year program (180 ECTS[1]) is its focus on project-based learning, supported by three project courses that build on each other.

The initial project course is a traditional individual project focusing on technical aspects of software development. The second year *small team project course* (STPC) and the third year *large team project course* (LTPC) are 15 ECTS each and involve team-based projects with industrial customers. Both team project courses have a similar general structure, and the same teachers are responsible for both courses.

This study examines STPC instances for 2020 and 2021, and LTPC instances from 2021 and 2022 (following two distinct cohorts - Section 2.4 for more details). In Spring 2020, due to the COVID-19 pandemic, student teams transitioned to remote and hybrid (i.e., a combination of onsite and remote) work setup, which persisted even after restrictions were lifted in 2022.

The details of the two courses are discussed in the following subsections.

### 2.3.1 Intended Learning Outcomes

The intended learning outcomes for the two courses include both technical and non-technical or soft skills. In STPC, the students are expected to demonstrate their ability to work in a smaller team, a professional approach to working in a commitment culture, skills and proficiency for the role of a professional in industry and business, technical knowledge in software development, critical thinking and the ability to reflect. In addition to these, in LTPC the students are expected to work independently on more complex projects within larger teams. These teams must conduct more thorough planning, design, and quality assurance than in the STPC. The LTPC also emphasizes collecting and analyzing metrics (e.g., team velocity, test coverage)

[1]The European Credit Transfer and Accumulation System. One year of full-time studies corresponds to 60 ECTS.

for informed decision-making and expects stricter design and quality assurance practices.

### 2.3.2 Students Teams

At the course start, the teachers allocate students to teams and assign teams to the projects approved by the teaching team. In the STPC, teams of 5 to 6 students develop a small software solution for a customer from the local industry.

In the LTPC, teams of 10 to 12 students develop a larger and more complex system for a customer from the local industry. The students organize themselves as sub-teams (e.g., front-end sub-team, back-end sub-team) within a large team. This sub-team structure not only introduces an additional layer of complexity for communication and coordination but also results in dependencies across sub-teams.

Students assume roles of product owners, Scrum masters, and developers/testers. Team members are encouraged to rotate between project management and development roles to gain diverse experience. Each team uses its preferred ecosystem of project management tools.

### 2.3.3 Course Projects

The project ideas are solicited from customers in the local industry before the start of the courses. The software projects involve the development of different types of software, such as web applications, mobile applications, internal modules of existing systems, plugins, APIs, or proof of concept type applications. The course teachers review and approve these project ideas, and discussions with customers may result in adjustments to the project scope, ensuring alignment with the learning outcomes of the course(s).

In both project courses, student teams are assigned a customer from the company where the project will be carried out. The teams begin by understanding and translating the customer's needs into specific requirements. Customers may specify preferred technologies for the project. Based on these requirements, students propose a solution, and through discussions, requirements specifications, and contracts, the customer and students agree on the final scope of the project.

The students are responsible for managing all stages of the software project, including requirements management, design, development, testing, and product delivery to the customer. They gain access to the customer's working environment, meaning they work on-site at the customer's premises.

Projects in LTPC are comparatively more complex as compared to STPC, due to the number and type of features that need to be developed. These features often require work across multiple layers, such as the database, front end, and back end, as well as the various technologies involved in the project. Table 2.3 presents examples

Table 2.3: Example projects for the two courses

| Course | Example Project Description |
|---|---|
| STPC | The customer wants a solution for growing tomatoes by adjusting led-light levels, intensity and frequency. The provided light armature will be equipped with a raspberry pi, a camera and a separate light sensor. The camera will take a photo of a tomato plant and send it to a central server for image processing and calculation of growth rates. The light sensor will measure light levels and intensity in the greenhouse and send it to the raspberry, which is then combined with the growth-algorithm to ensure the best possible environment for the plant. The best conditions and settings will be saved on a database for continuous improvement. |
| LTPC | The customer wants a solution for a web platform accompanied by an Android application that sends video data to AWS. A system based on various AWS services manages and analyzes video stream data and alerts the web platform in the case that detection of specific labels occurs. Labels can be the identification of specific objects in the data analyzed. The final goal of the product is to provide a platform for operators to view and handle incidents reported by an AWS AI system which analyzes image data gathered from camera devices. |

of projects students worked on in two courses.

### 2.3.4 Software Processes

The projects use iterative software development to plan and execute work in two-week sprints. For STPC, students follow Scrum as a guiding framework, conducting ceremonies such as sprint planning meetings, daily standups, sprint reviews with the customer, and sprint retrospectives.

While Scrum serves as a foundation, it is not rigidly followed; students are encouraged to adapt processes to suit their project and customer environment. For LTPC, teams adopt variants of scaled agile frameworks, such as Scrum of Scrums (SoS) and Large Scale Scrum (LeSS), tailoring these approaches to meet their project needs.

Student teams work closely with customers to plan the project scope. Meetings with the customer serve as important opportunities to gather and clarify the requirements, to request any resources from the customers, and discuss any technical issues related to the project or the customer environment. Requirements are gathered through customer interactions and specified as user stories, with product owners managing the backlog.

Customers provide feedback on requirement changes and different aspects of the system during sprint review demos. After the final demonstration and product delivery, including documentation, customers share their feedback on the team's performance and the delivered product with both the team and instructors.

### 2.3.5 Teaching and Mentorship

The common pedagogical approach in the project courses is “learning by doing” and formative learning and assessment. The focus is not on getting things right from the start but on engaging in project activities, reflecting on things that are not working out, and continuous improvement. Reflection is central in both team project courses as it has been recognized as an effective practice for continuous learning in software development [68].

There are dedicated time-slots for the project work. Apart from having workspaces at the customer premises, the teams have access to project rooms at the university. Teaching in the courses runs parallel to the project work through lectures, seminars, and supervision sessions. Lectures provide theoretical foundations, covering agile process frameworks (e.g., Scrum), requirements management, project planning and estimation, progress tracking and reporting, risk management, and software testing. The course begins with an introduction and a team-building session to set the stage for collaborative learning. Each student team is assigned a mentor from the teaching staff. The mentor acts as the project manager, overseeing the team’s progress and ensuring alignment with the goals set for the sprint through weekly meetings. Mentors assist in resolving technical and soft challenges, e.g., disagreements and conflicts, while providing formative feedback on individual and team performance. This feedback is informed by mentorship sessions and the evaluation of weekly deliverables.

### 2.3.6 Course Deliverables and Assessment

The course includes two individual components:

1. The weekly individual diary (see Appendix A).

2. The individual reflection report, that accounts for two credits within each course (see Appendix B).

The weekly diaries require students to document tasks completed, time spent, challenges faced, solutions, and other reflections on their learning. These diaries are mandatory and receive weekly formative feedback from mentors, guiding students to write better reflections. At the end of both courses, students submit detailed individual reflection reports that describe their project work, the challenges they faced, the solutions they implemented, and the lessons they learned. These final reports are based on the reflections from their individual weekly diaries. Students are explicitly asked to report their personal experiences and reflections in these reports. Mentors are responsible for evaluating both the weekly diaries and the final reports. Since they work closely with the teams throughout the project, they are aware of the challenges faced by the teams and individual students. Mentors do not approve a final report if it does not align with the content of the individual weekly diaries. This process ensures that the content is authentic and accurately reflects the students' experiences. To further ensure the content was not copied from external resources, plagiarism checks are applied.

Students are provided with a detailed template for the individual reflection reports, with sub sections to present an account of challenges and lessons learned. The texts from these subsections were used as data for the study; see Section 2.4.2 for details.

## 2.4 Research Methodology

In this section, we describe the research questions, data collection, and analysis methods used in the study.

### 2.4.1 Research Questions

**RQ 1:** What challenges do students face in team-based project courses with industrial customers?

*Motivation:* Understanding the challenges students face in team project courses helps identify potential areas for improvement in the design and delivery of such courses. This investigation may also provide insights into misalignments between course designs and industry requirements, facilitating better integration of industry-relevant practices into the curriculum.

**RQ 2:** What lessons do students learn in team-based project courses with industrial customers?

*Motivation:* Investigating the lessons students learn in these courses highlights the practical skills and competencies they develop, enabling teachers to focus on these aspects in designing and delivering similar courses. Additionally, this investigation can identify practices that better prepare students for industry challenges.

**RQ 3:** How have the challenges and lessons learned evolved in two team-based project courses with industrial customers?

*Motivation:* Investigating the evolution of challenges and lessons over two team project courses provides insights into how students' experiences change based on variables such as team size, project size and complexity, and student maturity. Analyzing this evolution helps identify overarching themes of challenges and lessons learned, which can inform the improvements in the design and delivery of such courses.

### 2.4.2 Data Collection

This study collected qualitative data by analyzing 158 individual reflection reports detailing challenges and lessons learned in team project courses. Conducting a qualitative document analysis is a valuable research approach [57]. Documents are a reliable data source as they are unobtrusive and nonreactive to the research process [69]. We identified students' individual reflection reports (referred to as reports hereafter) from two courses, the Small Team Project Course (STPC) and the Large Team Project Course (LTPC), as crucial artifacts for identifying challenges and lessons the students learned in team project courses.

We did not need any approval to analyze these reports, as assignments submitted for assessment are public documents. The reports were anonymized to remove the personal identification information of the students. However, we included markers to differentiate individual students and course instances.

A total of 158 reports were analyzed from two cohorts over a period of three years. The first cohort participated in the STPC in 2020 and the LTPC in 2021, while the second cohort undertook the STPC in 2021 and the LTPC in 2022. In total, there were 17 teams for the two instances of STPC and nine teams for the two instances of LTPC. It is important to note that there are 79 unique students across both cohorts. However, the total number of reports is 158 because these reports are gathered from the two courses studied by those 79 students.

The reports have a well-defined structure with separate sections for challenges and lessons learned. The texts from these sections of the reports were extracted into MS Word files. These MS Word files were then imported into NVivo, a digital tool for qualitative data analysis[2], to perform thematic analysis of the collected data. Table 2.4 presents the details of the data used in the thematic analysis.

**Table 2.4:** Size of the datasets per course instance, given as the number of reports, pages, and words per course instance, along with the minimum, maximum, and median number of words.

| | | | Number of words | | | |
|---|---|---|---|---|---|---|
| **Course** | **Reports** | **Pages** | **Total** | **Min** | **Max** | **Median** |
| STPC-2020 | 36 | 59 | 30872 | 218 | 1753 | 794 |
| STPC-2021 | 43 | 80 | 44762 | 264 | 3808 | 953 |
| Total STPC | 79 | 139 | 75634 | | | |
| LTPC-2021 | 34 | 69 | 38902 | 332 | 2887 | 1001 |
| LTPC-2022 | 45 | 84 | 44341 | 543 | 3622 | 779 |
| Total LTPC | 79 | 153 | 83243 | | | |
| TOTAL | 158 | 292 | 158877 | | | |

[2] https://lumivero.com/products/nvivo

### 2.4.3 Data Analysis

After data was extracted to Nvivo, we performed reflexive thematic analysis following the guidelines by Braun and Clarke [60, 62].

1. **Familiarisation with the data:** The first author familiarized with the data by creating cases in NVivo to represent each student in each course instance. These cases were used to characterize the challenges and lessons learned reported by each student.

2. **Generating initial codes:** In step two, the first author analyzed the dataset using a bottom-up approach to assign codes to each piece of text. These codes were then categorized under the main categories "challenges" or "lessons learned". This coding process involved multiple iterations during which the codes were reviewed and refined, while rephrasing and merging codes as new data were coded. After generating initial sets of codes (75 for challenges, 126 for lessons learned), the second author reviewed them in two workshops. The workshops were collaborative and reflexive, ensuring richer interpretations of meaning, as recommended by Byrne [62]. This step was crucial for ensuring accuracy and minimizing bias, as highlighted by Khakurel and Porras [17]. The authors used a checklist suggested by Cruzes and Dybå [61] to ensure the validity of the coding process. Several checks were implemented, including completeness (text-to-code and code-to-text), consistency (inter-rater reliability), and a clear connection between the text and the assigned code. As suggested by Byrne [62], an open coding process was followed to closely represent the students' reflections rather than fitting the text to pre-existing codes.

3. **Generating themes:** Once the codes for all challenges and lessons learned were generated, the authors worked together in a joint session to identify overarching patterns and relationships between these codes. This collaborative effort led to the grouping of codes into a hierarchy of themes and sub-themes. An example of generating codes and themes from students' text is shared in Table 2.5.

4. **Reviewing potential themes:** The first two authors reviewed the initial set of identified themes and sub-themes for possible overlaps and redundancies. They ensured that the generated themes were comprehensive and inclusive of all the codes, as recommended by Cruzes and Dybå [61]. The themes and sub-themes were also checked for coherence, consistency, and distinctiveness [61]. This process resulted in six themes for challenges and nine themes for lessons learned.

5. **Defining and naming themes:** Once consensus was reached between the two authors, the themes were checked and renamed to ensure meaningfulness and

Table 2.5: An example of coding text into codes, sub-themes and themes.

| Student Quotes | Code | Sub-theme | Theme |
|---|---|---|---|
| "The team had problems with communication and unity during the first weeks; however, after that, the team pointed out some issues and cleared the air, and it got better." | Establishing initial communication within team | Communication issues within team | Working in a team |
| "We had one person who did not contribute, or try their best, to contribute to the project as the rest of the team did. This have caused a lot of frustration within the team." | Unequal contribution from team members | Contribution issues within team | Working in a team |

consistent naming conventions. Each theme and sub-theme was then analyzed for its relevance to the dataset and the research questions, as suggested by Bruegge et al. [12].

6. **Producing the report:** The results from steps 1–5 are described in detail in Sections 2.5 and 2.6.

## 2.5 Results and Analysis: Challenges (RQ1 and RQ3)

As shown in Table 2.6, we identified six themes for the challenges faced by the students in our project courses. Tables 2.7 to 2.22 present detailed challenges and lessons learned for the identified themes. These tables provide insights from two perspectives: (1) the number of instances and (2) the number of reports (shown in parentheses) under the columns STPC and LTPC in each table.

Table 2.6: Themes for challenges in the 158 individual reports.

| | Reported instances* | | |
|---|---|---|---|
| **Challenge (themes)** | **TOT** | **REP** | **in %** |
| Working in a team | 243 | 116 | 73.4 |
| Working in a remote and hybrid setting | 209 | 107 | 67.7 |
| Working with an industrial customer | 177 | 87 | 55.1 |
| Working with new technology | 53 | 46 | 29.1 |
| Managing the project | 49 | 43 | 27.2 |
| Maintaining self-confidence and motivation | 12 | 10 | 6.3 |
| **Total** | **743** | **158** | |

***TOT**al number of instances found in REP (individual reports). In %=REP/158.

The three themes *working in a team* (mentioned in 73.4% of reports), *working in a remote and hybrid setting* and *working on a project with an industrial customer* were all mentioned by more than half of the students. All three relate to non-technical or what is otherwise known as soft skills required in software projects. Another challenge related to soft skills is *maintaining self-confidence and motivation*. Taken together, the soft skill themes account for 91.3% (678 of 743) of the total challenge instances and appear in all 158 reports.

Only two themes pertain to the technical or disciplinary knowledge in software engineering: *working with new technology* and *managing the project*. This indicates that students in the investigated team project courses find non-technical issues more challenging than technical ones. Here, it is important to clarify the context of the reported themes, which is "project-based learning in collaboration with the industrial client". The challenges reported by the students may present a combined manifestation of these two aspects and should not be considered in isolation. For instance, *communication within team* is inherently challenging in PBL, but collaborating with an industrial client can introduce additional issues, such as the need to develop a shared understanding of the project. In the following subsections, we further elaborate on each of the themes, while presenting the most frequently reported sub themes. It is important to note that a single student may have reported multiple challenges under each theme and sub-theme.

### 2.5.1 Working in a Team

Working in a team emerged as the most frequently reported challenge theme with 243 reported instances across 116 reports. The theme comprises 14 sub-themes, which are summarized in Table 2.7. Overall, it can be seen that in the LTPC more challenge instances have been reported by more students than in the STPC, although the total number of students was the same in both courses (79), see also Table 2.4).

**Table 2.7:** Sub-themes of *Working in a team.*

| **Challenge (sub-themes)** | **STPC** | **LTPC** | **Total** |
|---|---|---|---|
| Communication issues within team | 18 (16)* | 41 (29) | 59 (45) |
| Contribution issues within team | 11 (9) | 22 (16) | 33 (25) |
| Forming a team with unfamiliar people | 9 (8) | 10 (10) | 19 (18) |
| Managing changes within team | 16 (14) | 2 (2) | 18 (16) |
| Poor team dynamics | 9 (5) | 15 (9) | 24 (14) |
| Coordination issues within team | 0 (0) | 15 (14) | 15 (14) |
| Developing the team mindset | 4 (4) | 8 (8) | 12 (12) |
| Managing conflicts within team | 5 (5) | 7 (7) | 12 (12) |
| Keeping track of team progress | 5 (5) | 7 (6) | 12 (11) |
| Skill gap between team members | 9 (9) | 2 (2) | 11 (11) |
| Different ambition levels within team | 3 (2) | 9 (7) | 12 (9) |
| Adapting to diverse working styles | 3 (3) | 6 (6) | 9 (9) |
| Other difficulties working in a team | 3 (3) | 2 (2) | 5 (5) |
| No prior experience working in a team | 2 (2) | 0 (0) | 2 (2) |
| **Total for the theme** | **97 (53)** | **146 (63)** | **243 (116)** |

*The first number states the number of instances, the number in parentheses states the number of reports.

#### 2.5.1.1 Main sub-themes

*Communication issues within the team* include establishing initial communication, achieving consensus, developing a shared understanding of the project and the process, and hesitation to ask for help when needed. In some cases, language barriers are also mentioned as a communication challenge.

*Contribution issues within the team* include issues such as certain team members failing to participate in group meetings, lacking the ability to offer adequate support when needed by others, and either not completing assigned tasks or submitting them late.

*Forming a team with unfamiliar people* relates to challenges in working with team members students were not already familiar with.

*Managing changes within the team* relates to stressful situations when managing changes within the team. These changes involved re-assigning roles or a team member leaving the team. Both scenarios introduced replanning, redistribution of responsibilities, and increased work.

*Poor team dynamics* relates to conflicts or disagreements, the need to manage diverse perspectives, and "ego issues" among team members.

#### 2.5.1.2 Comparison across the two project courses

Overall, teamwork-related challenges were reported 146 times (in 63 reports) for LTPC and 97 times (in 53 reports) for STPC, suggesting that teamwork becomes more challenging as team size and project complexity increase. This difference can be seen in 10 of the 14 sub-themes. For the sub-theme *coordination issues within team*, this trend is pronounced specifically since it was not mentioned at all in STPC, but by 14 students (17.7%) in LTPC.

However, there are also two notable exceptions to this trend. *Managing changes within team* and *skill gap between team members* were mentioned by 14 and 9 students, respectively, in STPC but only 2 in LTPC. An explanation might be that students become relatively more mature in addressing the challenges due to their prior experiences in STPC.

### 2.5.2 Working in a Remote and Hybrid Setting

The second highest number of challenges were reported on working in a remote and hybrid setting during the pandemic. There are 209 challenge instances highlighted in 107 reports (see Table 2.8 for details). This theme accounts for about 28% of all challenge instances and about 68% of all reports. Due to the pandemic, the student teams were initially forced and later became accustomed to working in a hybrid mode. All team activities including sprint planning meetings, daily stand-ups, sprint reviews, and sprint retrospectives, were conducted online using tools like Zoom or Microsoft Teams. There are 12 sub-themes to this theme, for details, see Table 2.8.

**Table 2.8:** Sub-themes of *Working in a remote and hybrid setting.*

| Challenge (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Communicating with the team remotely | 42 (29) | 33 (28) | 75 (57) |
| Fostering team dynamics in remote setting | 10 (10) | 11 (11) | 21 (21) |
| Staying motivated while working remotely | 17 (17) | 4 (4) | 21 (21) |
| Tracking team progress in remote setting | 6 (6) | 11 (10) | 17 (16) |
| Maintaining team efficiency in remote setting | 8 (7) | 7 (6) | 15 (14) |
| Inability to work in customer premises | 7 (7) | 7 (7) | 14 (14) |
| Adapting to online working mode | 14 (11) | 1 (1) | 15 (12) |
| Keeping focus while working from home | 5 (5) | 7 (7) | 12 (12) |
| Managing sickness | 1 (1) | 7 (7) | 8 (8) |
| Maintaining work-life balance | 6 (6) | 1 (1) | 7 (7) |
| Working on-site during the pandemic | 3 (3) | 0 (0) | 3 (3) |
| Maintaining wellbeing | 1 (1) | 0 (0) | 1 (1) |
| **Total for the theme** | **120 (58)** | **89 (49)** | **209 (107)** |

#### 2.5.2.1 Main sub-themes

*Communicating with the team remotely* includes issues related to ensuring clarity while communicating remotely, the need to communicate more often, and reduced communication effectiveness while working remotely, as this form of communication lacks the richness of face-to-face interactions.

*Fostering team dynamics in a remote setting* relates to difficulty establishing cohesion within the team, compromised ability to interact with the team members, and difficulty in resolving misunderstandings and conflicts.

*Staying motivated while working remotely* posed issues such as lack of peer pressure, lack of social connections, and the monotony of the routine, all contributing to declined motivation.

*Tracking team progress while working remotely* includes challenges of team members not sharing their progress and compromised ability to see the actual work done.

*Maintaining team efficiency while working remotely* has several factors contributing towards this sub-theme, including difficulty staying focused, compromised ability to ask for help, duplication of tasks due to lack of transparency, and team members working outside regular working hours.

#### 2.5.2.2 Comparison across the two project courses

Challenges related to working in remote and hybrid settings during the pandemic were reported 120 times in 58 reports for STPC and 89 times in 49 reports for LTPC. While our analysis focuses on how the challenges evolved across the two courses, a cohort-wise comparison for this theme is relevant due to variations in work settings across the years. Highest number of challenges under this theme were reported for STPC-2020 i.e., 71 instances in 31 reports, indicating that students found it challenging to transition from onsite to remote work. The number of challenges decreased progressively with 49 instances in 27 reports for STPC-2021, 50 instances in 25 reports for LTPC-2021 and 39 challenge instances in 24 reports for LTPC-2022.

At the sub-theme level, several differences were observed. For example, communicating with the team while working remotely was reported as a challenge 42 times in 29 reports for STPC, compared to 33 instances reported in 28 reports for LTPC. Similarly, staying motivated while working remotely was reported 17 times in 17 reports for STPC, compared to only four instances reported in four reports for LTPC. Adapting to the online working mode was another notable challenge, with 14 instances reported in 11 reports in STPC, compared to just one instance reported in one report in LTPC.

On the contrary, some sub-themes have comparatively more instances for LTPC, such as tracking team progress while working remotely and managing sickness, possibly because of a larger number of team members and sub-teams working remotely in LTPC.

### 2.5.3 Working with an Industrial Customer

The third highest number of challenges were reported on working on a project with an industrial customer, with 177 challenge instances reported in 87 reports. This theme accounts for about 24% of all challenge instances and about 55% of all reports. See Table 2.9 for details.

**Table 2.9:** Sub-themes of *Working with an industrial customer.*

| Challenge (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Understanding requirements | 22 (19) | 14 (13) | 36 (32) |
| Managing technical complexity in the project | 16 (14) | 17 (15) | 33 (29) |
| Managing the project scope | 12 (10) | 12 (10) | 24 (20) |
| Understanding customer's system | 17 (13) | 7 (6) | 24 (19) |
| Working in the customer technical environment | 3 (3) | 20 (14) | 23 (17) |
| Communicating with the customer | 1 (1) | 13 (13) | 14 (14) |
| Integrating different parts of the project | 0 (0) | 11 (10) | 11 (10) |
| Lack of experience in project domain | 5 (5) | 3 (3) | 8 (8) |
| High expectations of the customer | 0 (0) | 2 (2) | 2 (2) |
| Working on outdated parts of the system | 2 (2) | 0 (0) | 2 (2) |
| **Total for the theme** | **78 (44)** | **99 (43)** | **177 (87)** |

#### 2.5.3.1 Main sub-themes

*Understanding requirements* has multiple aspects that were challenging for the students, including unclear requirement specifications or the specification being too brief and difficulty in understanding what the customer exactly needs. A few cases also report customers' lack of clarity on requirements, specifically for a new feature or module.

*Managing technical complexity in the project* has different aspects contributing to this challenge including complex feature requirements (e.g., using AI or machine learning), dependencies between multiple modules and/or interfaces, dependencies on certain hardware, and high customer expectations on scalability or performance.

*Managing the project scope* includes challenges such as underestimating the complexity of the customer's system and making an initial commitment without enough understanding of the requirements. This lack of clarity made it challenging to determine what could be accomplished within the allocated timeframe and what aspects needed negotiation with the customer. Additionally, students struggled with managing later changes in the project scope as requested by the customer.

*Understanding customer's system* took time for the students. Factors contributing to this challenge include a lack of good project documentation and several dependencies between different system modules.

*Working in the customer technical environment* posed several challenges for the students. Companies have their own ecosystem of tools, technology, and protocols around that. Factors contributing to this sub-theme include difficulties in familiarizing oneself with the customers' technical environment, changes and updates in the technical environment, issues within the technical environment (e.g., crashing servers), and late or limited access to the resources required to work in the customer environment (e.g., licenses, access codes or specialized hardware).

*Communicating with the customer* involved issues such as use of different terminology by customers, availability of the customer, and obtaining timely responses from the customer.

#### 2.5.3.2 Comparison across the two project courses

Challenges related to working on a project with an industrial customer are reported in almost the same number of reports for STPC (44 reports) and LTPC (43 reports). However, relatively more challenges are reported in LTPC reports (99 as compared to 78 in STPC) under this theme.

At the sub-theme level, students in LTPC found it more challenging to work within the customer's technical environment, with 20 instances reported in 14 reports, compared to only three instances in three STPC reports. Similarly, integrating different parts of the system posed a greater challenge in LTPC, with 11 instances reported in 10 reports and no such instances reported in STPC. These differences may be due to the increased complexity of LTPC projects.

Another frequently reported challenge in LTPC is communicating with the customer, with 13 instances reported in 13 reports, compared to only one instance in STPC. However, some challenges decreased as students progressed from STPC to LTPC. For example, understanding requirements was more difficult in STPC, with 22 instances reported in 19 reports, compared to 14 instances in 13 LTPC reports. Similarly, challenges related to understanding the customer's system were more prevalent in STPC, with 17 instances in 13 reports, compared to only seven instances in six LTPC reports.

### 2.5.4 Working with New Technology

Students reported several challenges when working with new tools and technology, with 53 challenge instances reported in 46 reports. This theme accounts for about 7% of all challenge instances and 29% of all reports. Although these numbers are not as high as the previous three themes, they still reflect the difficulties that some students face while working with new tools and technology. See Table 2.10 for details.

**Table 2.10:** Sub-themes of *Working with new technology.*

| **Challenge (sub-themes)** | **STPC** | **LTPC** | **Total** |
|---|---|---|---|
| Lack of experience with tools & technology | 24 (22) | 15 (15) | 39 (37) |
| Lack of support in chosen technologies | 5 (5) | 0 (0) | 5 (5) |
| Making appropriate technology choices | 2 (2) | 2 (2) | 4 (4) |
| Lack of experience with version control SW | 3 (3) | 1 (1) | 4 (4) |
| Equipment-related challenges | 0 (0) | 1 (1) | 1 (1) |
| **Total for the theme** | **34 (29)** | **19 (17)** | **53 (46)** |

#### 2.5.4.1 Main sub-themes

*Lack of experience with tools and technology* required to work on the project emerged as the most reported sub-theme. Students struggled to propose alternative technologies when customers had already suggested specific ones. These involved several front-end and back-end technologies, e.g., angular.js, NoSQL databases e.g., Firebase, programming languages, e.g., Python, cloud technologies, e.g., OpenStack, and AI and machine learning technologies, e.g., TensorFlow.

*Lack of support in chosen technologies* was another challenge students reported. Students sometimes make technology choices while working on customer projects by looking at the initial requirements. In some cases, it turned out later that the selected technologies did not provide enough support for the features they were developing.

*Making appropriate technology choices* was challenging for some students. Students have reported investing significant time in researching appropriate tools and technology. Some students expressed skepticism about the appropriateness of their technology choices. In few cases, initial technology choices had to be abandoned, costing a lot of time.

*Lack of experience with version control software* made some students struggle with version control systems like Git and Gerrit. This challenge has four instances across four reports.

#### 2.5.4.2 Comparison across the two project courses

Challenges under this theme were reported more frequently for STPC, with 34 instances in 29 reports, compared to 19 instances for LTPC in 17 reports. At the sub-theme level, the most prominent challenge was the lack of experience with certain tools and technologies, reported 24 times in 22 reports for STPC compared to 15 instances in 15 reports for LTPC. Other challenges, such as lack of experience with version control systems and lack of support in chosen technologies, also decreased in LTPC compared to STPC. This suggests that students became more proficient with tools and technologies as they progressed from STPC to LTPC.

### 2.5.5 Managing the Project

Students have also reported challenges in managing the project. In total, 49 challenge instances were reported in 43 reports. This theme accounts for about 6.6% of all challenge instances and about 27% of all reports. See Table 2.11 for details.

**Table 2.11:** Sub-themes of *Managing the project.*

| Challenge (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Making correct effort estimates | 18 (15) | 7 (7) | 25 (22) |
| Making initial plans | 8 (8) | 2 (2) | 10 (10) |
| Lack of clear work distribution | 1 (1) | 7 (7) | 8 (8) |
| Lack of initiative to take project management roles | 0 (0) | 2 (2) | 2 (2) |
| Lack of project management experience | 2 (2) | 0 (0) | 2 (2) |
| Even distribution of tasks | 0 (0) | 1 (1) | 1 (1) |
| Ineffective time management | 1 (1) | 0 (0) | 1 (1) |
| **Total for the theme** | **30 (24)** | **19 (19)** | **49 (43)** |

#### 2.5.5.1 Main sub-themes

*Making correct effort estimates* was the most challenging aspect of project management. Students often underestimated the complexity of the tasks and overlooked the time needed for testing.

*Making initial plans* was difficult for some students. This involved difficulties in understanding the tasks, setting up the platforms for collaborating on the project, and assigning roles and responsibilities.

*Lack of clear work distribution* was reported by students despite using project management tools like Jira. Students had issues with unclear task descriptions and certain teammates not reporting their progress in Jira. This, sometimes, led to people taking up wrong tasks or multiple people doing the same task.

#### 2.5.5.2 Comparison across the two project courses

Challenges related to project management were reported more frequently for STPC, with 30 instances in 24 reports, compared to 19 instances in 19 reports for LTPC. At the sub-theme level, the trends varied. For instance, making correct effort estimates and initial plans were more challenging in STPC than in LTPC. However, the lack of clear work distribution was more prominent in LTPC, where we have relatively larger teams that are further organized as sub-teams.

### 2.5.6 Maintaining Self-confidence and Motivation

Maintaining self-confidence and motivation also emerged as one of the themes, with 12 instances in 10 reports. This number accounts for about 1.6% of all challenge instances and about 6% of all reports. See Table 2.12 for details.

**Table 2.12:** Sub-themes of *Maintaining self-confidence and motivation.*

| Challenge (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Doubts in ability to achieve project goals | 6 (5) | 1 (1) | 7 (6) |
| Difficulty staying motivated | 1 (1) | 4 (3) | 5 (4) |
| **Total for the theme** | **7 (6)** | **5 (4)** | **12 (10)** |

#### 2.5.6.1 Main sub-themes

*Doubts in the ability to achieve project goals* includes issues like nervousness and lack of confidence in achieving project goals, especially at the start of the project. These doubts emerged from the lack of experience in the project domain, the need to work with new technologies, and the project requirements being too complex. Some students were overwhelmed by the idea of working with an industrial customer in a professional setting.

*Difficulty staying motivated* had several factors contributing to this challenge including the team environment, dependability on others for completion of tasks, lack of interest in the project domain, and lack of experience in the technologies used.

#### 2.5.6.2 Comparison across the two project courses

Challenges under this theme are reported more for STPC than in LTPC, i.e., seven compared to five challenge instances. Staying motivated was more difficult in LTPC (four instances in three reports) than STPC (one instance). On the other hand, students doubted their ability to achieve project goals more in STPC (six instances in five reports) than in LTPC (one instance in one report).

## 2.6 Results and Analysis: Lessons Learned (RQ2 and RQ3)

We identified nine themes of lessons learned by the students (see Table 2.13).

**Table 2.13:** Themes for lessons learned in the 158 individual reports.

| | **Reported instances*** | | |
|---|---|---|---|
| **Lessons learned (themes)** | **TOT** | **REP** | **in %** |
| Working in a team | 395 | 140 | 88.6 |
| Managing the project | 189 | 113 | 71.5 |
| Working with an industrial customer | 141 | 96 | 60.7 |
| Working in a remote and hybrid setting | 111 | 77 | 48.7 |
| Managing the product quality | 80 | 59 | 37.3 |
| Setting personal values and ethics | 79 | 57 | 36.0 |
| Formalizing the processes | 40 | 33 | 20.8 |
| Working with new technology | 40 | 31 | 19.6 |
| Taking advice from the mentors | 9 | 8 | 5.0 |
| **Total** | **1084** | **158** | |

***TOT**al number of instances found in REP (individual reports). In %=REP/158.

As for challenges (RQ1), most of the themes for lessons learned relate to soft skills such as *working in a team* (mentioned in 88.6% of the reports), *working with an industrial customer* 60.7%), and *working in a remote and hybrid setting* (48.7%). Additionally, a significant number of lessons were reported regarding individual aspects, such as *setting personal values and ethics* (36.0%). Altogether, the soft skill themes account for 67.8% (735 of 1084) of the total lessons learned instances reported in 158 reports.

The themes about technical or disciplinary knowledge include *managing the project* (71.5%), *managing product quality* (37.3%), *formalizing the processes* (20.8%), and *working with new technology* (19.6%). In total, these technical and disciplinary knowledge-related account for 32.2% (349 of 1084) of all lessons learned instances. This indicates that although working on team project courses contributed to students' technical and disciplinary knowledge, the students perceive that they contributed more to developing and improving their soft skills.

In the following subsections, we further elaborate on each of the themes, while presenting the most frequently reported sub themes.

### 2.6.1 Working in a Team

The most prominent theme is working in a team, with a total of 395 instances reported in 140 reports. This theme accounts for 36.4% of all instances of lessons learned and appears in 88.6% of the reports. The theme comprises 12 sub-themes which are summarized in Table 2.14. It can be seen that there are more lesson instances reported for LTPC than in STPC.

**Table 2.14:** Sub-themes of *Working in a team.*

| **Lessons learned (sub-themes)** | **STPC** | **LTPC** | **Total** |
|---|---|---|---|
| Ensure good team communication | 74 (48) | 58 (43) | 132 (91) |
| Create a conducive team environment | 29 (25) | 20 (16) | 49 (41) |
| Create good team dynamics | 32 (23) | 20 (15) | 52 (38) |
| Manage team meetings effectively | 13 (13) | 27 (23) | 40 (36) |
| Develop a team mindset | 17 (15) | 14 (12) | 31 (27) |
| Share knowledge within team | 10 (10) | 19 (17) | 29 (27) |
| Create sub-groups in large teams | 0 (0) | 14 (12) | 14 (12) |
| Appoint a team leader | 4 (4) | 10 (7) | 14 (11) |
| Ensure good coordination within team | 0 (0) | 11 (10) | 11 (10) |
| Set clear goals for the team | 5 (5) | 6 (4) | 11 (9) |
| Encourage reflection and feedback | 2 (2) | 5 (5) | 7 (7) |
| Define a code of conduct for the team | 2 (2) | 3 (2) | 5 (4) |
| **Total for the theme** | **188 (66)** | **207 (74)** | **395 (140)** |

#### 2.6.1.1 Main sub-themes

*Ensure good team communication* is the most reported sub-theme for lessons learned for working in a team. Students reported specific lessons, such as adjusting conversation according to the audience's knowledge and experience, being open and clear while communicating within the team, not hesitating in asking for help, establishing effective channels for communication, and following up to ensure everyone is on the same page.

*Create a conducive team environment* includes specific lessons such as being open to understanding each other's perspectives, ensuring that everyone in the team gets to share their thoughts and ideas, understanding and acknowledging differences, and maintaining a good team spirit through appreciation and motivation.

*Create good team dynamics* has been suggested several times. Specific lessons reported by students include being humble and considerate while working in a team, making an effort to know the team members, discussing the importance of having a good team dynamic, having fun together as a team, having some team building sessions, maintaining trust within team members, and approaching and resolving conflicts as early as possible.

*Manage team meetings effectively* includes several lessons, including defining the scope and duration of the meetings, clearly communicating the meeting agenda, ensuring maximum team member participation in the meetings, having meetings only when necessary, realizing and communicating the importance of team meetings. Moreover, in the context of LTPC, students preferred replacing meetings requiring the participation of the entire team with a Scrum of Scrums. In this setup, only representatives from each sub-team attended the meeting to share critical updates, coordinate dependencies, and address blockers.

*Develop a team mindset* includes lessons on ensuring team cohesion as early as possible, identifying 'we' needs from 'me' needs, having a shared responsibility of project outcomes, supporting each other in accomplishing tasks, and maintaining team integrity while working in larger teams.

#### 2.6.1.2 Comparison across the two project courses

In aggregate, lessons related to teamwork are reported more frequently for LTPC as the team size grows. There are 207 lesson instances reported in 74 reports for LTPC, compared to 188 instances reported in 66 reports for STPC. There is a varying trend across the two courses at the sub-theme level. For certain sub-themes, more lesson instances are reported for STPC. These include ensuring good team communication, creating good team dynamics, establishing a conducive team environment, and developing a team mindset. Conversely, for some other sub-themes, more lessons are reported for LTPC compared to STPC. These include lessons on managing team meetings effectively, sharing knowledge within the team, appointing a team leader, creating subgroups in a larger team, ensuring good coordination within the team, setting clear goals for the team, encouraging reflection and feedback, and defining a code of conduct for the team.

### 2.6.2 Managing the Project

The second most reported theme for lessons learned while working on a team project is managing the project. There are 189 instances of lessons reported in 113 reports. This theme accounts for about 17.4% of all lesson instances and 71.5% of all reports. Lessons learned under this are categorized into seven sub-themes, which are summarized in Table 2.15.

#### 2.6.2.1 Main sub-themes

*Invest time in planning* includes several lessons, on project planning, such as making detailed plans on different aspects of the project (scope and intended outcomes, project timelines, etc), planning iteratively, and strictly following the plan, trying to avoid any deviations.

**Table 2.15:** Sub-themes of *Managing the project.*

| **Lessons learned (sub-themes)** | **STPC** | **LTPC** | **Total** |
|---|---|---|---|
| Invest time in planning | 31 (26) | 15 (14) | 46 (40) |
| Manage work distribution effectively | 16 (14) | 28 (23) | 44 (37) |
| Manage requirements& changes efficiently | 11 (11) | 19 (17) | 30 (28) |
| Invest time in effort estimation | 14 (13) | 13 (13) | 27 (26) |
| Make informed choices for the project | 11 (11) | 12 (8) | 23 (19) |
| Focus more on risk management | 5 (5) | 5 (5) | 10 (10) |
| Use project management tools effectively | 4 (4) | 5 (4) | 9 (8) |
| **Total for the theme** | **92 (55)** | **97 (58)** | **189 (113)** |

*Manage work distribution effectively* includes lessons such as defining owners for different parts of the project, dividing tasks evenly to increase efficiency, having clear delegation of roles and responsibilities, proactively keeping track of team progress, having a shared responsibility for project management tasks, and communicating tasks clearly with all the team members.

*Manage requirements and changes efficiently* includes several lessons such as documenting the requirements clearly, specifying the scope and definition of done for each requirement, prioritizing the requirements carefully, being flexible and adapting to changes quickly, keeping track of changes, and focusing on the minimum viable product.

*Investing time in effort estimation* includes breaking down the larger tasks into smaller ones, making careful effort estimates to avoid underestimating, considering time for testing and bug resolution for each task, using effort estimation techniques, and updating estimations later in the project. Effort estimation should not be confused with planning. Effort estimation focuses on calculating the time and resources required to complete specific tasks, whereas planning implies broader project-level decisions on scope, outcomes and timelines.

*Make informed choices for the project* implies important lessons on making decisions grounded in thorough system analysis. It involves gathering and leveraging information about the system, making optimal use of online resources, utilizing existing solutions and frameworks, avoiding third-party modules as primary system features, and not postponing project work due to research. This initial phase of project development sets the foundation for effective planning.

#### 2.6.2.2 Comparison across the two project courses

Lessons on managing the project are reported comparatively more for LTPC, with 97 instances reported in 58 reports, compared to 92 instances in 55 reports for STPC. Despite having experience working on STPC, students have reported a similar number of lessons learned in LTPC. This suggests that there were still new lessons to be learned in LTPC, likely due to the increased complexity of the projects. At the sub-theme level, the most prominent differences can be observed for the top three sub-themes. Lessons on project planning are reported more in STPC, with 31 instances in 26 reports, compared to 15 instances in 14 reports for LTPC. In contrast, more lessons on managing work distribution are reported for LTPC, with 28 instances in 23 reports compared to 16 instances in 14 reports for STPC. Similarly, lessons on managing requirements and changes are reported more for LTPC, with 19 instances in 17 reports, compared to 11 instances in 11 reports for STPC. The rest of the sub-themes have comparable values across the two courses.

### 2.6.3 Working with an Industrial Customer

Lessons learned while working with an industrial customer have been reported 141 times in 96 reports, making up 13% of all lesson instances and about 61% of all reports. These lessons learned concern different dimensions of students' interactions with the customer including communication with the customer, understanding requirements from the customer's perspective, conducting meetings with the customer and so on. For the details of the sub-themes, please refer to Table 2.16.

**Table 2.16:** Sub-themes of *Working with an industrial customer.*

| Lessons learned (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Communicate effectively with the customer | 34 (27) | 23 (22) | 57 (49) |
| Understand requirements from customers' perspective | 32 (27) | 17 (15) | 49 (42) |
| Conduct effective meetings with customer | 9 (8) | 5 (5) | 14 (13) |
| Keep commitments with customer | 6 (6) | 5 (5) | 11 (11) |
| Optimize learning in customer environment | 7 (7) | 3 (3) | 10 (10) |
| **Total for the theme** | **88 (53)** | **53 (43)** | **141 (96)** |

#### 2.6.3.1 *Main sub-themes*

*Communicate effectively with the customer* includes lessons on maintaining open, clear, and frequent communication with the customer. Specific lessons under this sub-theme include defining clear communication channels with the customer, frequently updating the customer on project progress, being proactive in requesting meetings and resources from the customer, and using informal communication when appropriate.

*Understand requirements from customers' perspective* include lessons on using customers' help in understanding requirements, discussing requirements in meetings instead of emails, maximizing engagement with customers for better requirement understanding, having a dialogue on customers' priorities, engaging with all stakeholders, using visual representations for clearer communication, and seeking regular customer feedback.

*Conduct effective meetings with the customer* include suggestions on proactivity in scheduling customer meetings, i.e., securing time for future meetings as early as possible, having a well-defined scope for each meeting, preparing well for the meeting, taking notes, and documenting the meeting with the customer.

*Keep commitments with customers* emphasizes the importance of keeping commitments with the customer. Students advise caution while making commitments to prevent over-commitment. Unmet commitments can negatively impact customers' interest and motivation in the project.

*Optimize learning in customer environment* relates to appreciating the experience of working with an industrial customer. Students have suggested capitalizing on this opportunity. Some have mentioned the steep learning curve, particularly when the customer is technically proficient, which presents both challenges and insights.

#### 2.6.3.2 Comparison across the two project courses

Lessons learned while working with an industrial customer have been reported more frequently for STPC compared to LTPC. There are 88 lesson instances in 53 reports for LTPC, compared to 53 instances in 43 reports for STPC. This suggests that students learned more lessons when they first started working on projects with industrial customers in STPC. At the sub-theme level, the trend is similar. For each sub-theme, more lessons are reported for STPC than for LTPC. For certain sub-themes, the difference in lesson instances is even more pronounced. For example, understanding requirements from the customer's perspective has 32 instances in 27 reports for STPC, compared to 17 instances in 15 reports for LTPC.

### 2.6.4 Working in a Remote and Hybrid Setting

Students have specified several lessons they have learned while working in a remote and hybrid setting during the pandemic. There are 111 instances of lessons in 77 reports accounting for 10.2% of the total lesson instances and about 49% of the reports. There are 12 sub-themes under this theme (See Table 2.17).

**Table 2.17:** Sub-themes of *Working in a remote and hybrid setting.*

| **Lessons learned (sub-themes)** | **STPC** | **LTPC** | **Total** |
|---|---|---|---|
| Make good use of collaboration tools for remote work | 20 (19) | 8 (8) | 28 (27) |
| Working on-site is better | 6 (5) | 29 (21) | 35 (26) |
| Working remotely is better | 14 (14) | 9 (8) | 23 (22) |
| Setup team schedule for remote work | 15 (14) | 7 (7) | 22 (21) |
| Use camera in online meetings | 0 (0) | 3 (3) | 3 (3) |
| **Total for the theme** | **55 (39)** | **56 (38)** | **111 (77)** |

#### 2.6.4.1 Main sub-themes

*Make good use of collaboration tools for remote work* was the most reported sub-theme. Students have suggested making efficient use of collaboration tools such as Discord, microsoft teams, and Zoom. Other project management tools such as trello have also been sugegsted.

*Working on-site is better* was one of the takeaways for students during the pandemic. The key lessons are related to understanding the importance of face-to-face interactions and having a proper workspace.

*Working remotely is better* was a lesson for several students driven by factors like having an introverted nature, travel constraints, and the comfort of working in a home environment.

*Setting up team schedule for remote work* was suggested by a significant number of students. An aligned schedule for teamwork allowed for creating good communication and run-time resolution of any conflicts or misunderstandings, improving overall team efficiency.

#### 2.6.4.2 Comparison across the two project courses

Working in a remote and hybrid setting has a similar number of lesson instances reported in both courses. There are 55 instances in 39 reports for STPC and 56 instances in 38 reports for LTPC. However, at the sub-theme level, there are differences. The most frequently reported lesson by students in LTPC is realizing their preference for the on-site working mode, with 29 instances reported in 21 reports.

On the other hand, the rest of the sub-themes have more lesson instances for STPC than LTPC. These include making good use of collaboration tools and setting up team schedules for remote work. Additionally, working in a remote setting is appreciated more for STPC than LTPC.

### 2.6.5 Managing the Product Quality

There are 80 instances of lessons on managing the product quality specified in 59 reports accounting for 7.3 % of the total lesson instances and 37.3% of all the reports. The sub-themes relate to different aspects of software quality, such as code quality, testing, software design and architecture, and product usability. For details of these lessons (See Table 2.18).

#### 2.6.5.1 Main sub-themes

*Focus more on code quality* include specific lessons such as writing clean code, integrating codes in smaller commits, not pushing code to the main branch directly, making use of static analysis tools, setting coding standards for the team, reviewing code to improve code quality, and making use of pair programming technique to improve quality.

**Table 2.18:** Sub-themes of *Managing the product quality.*

| **Lessons learned (sub-themes)** | **STPC** | **LTPC** | **Total** |
|---|---|---|---|
| Focus more on testing | 21 (19) | 12 (12) | 33 (31) |
| Focus more on code quality | 6 (5) | 27 (21) | 33 (26) |
| Focus more on design and architecture | 8 (8) | 4 (3) | 12 (11) |
| Take into account product usability | 1 (1) | 1 (1) | 2 (2) |
| **Total for the theme** | **36 (28)** | **44 (31)** | **80 (59)** |

*Focus more on testing* includes lessons regarding testing, such as focusing on testing as early as possible, having a formal test plan, considering testing at all levels and not only the unit testing, making use of automated testing techniques, and delivering the product as early as possible to have time to address post delivery errors and bugs.

*Focus more on design and architecture* includes lessons on carefully planning the program structure, considering different design aspects such as maintainability in terms of coupling and cohesion, and using clear guidelines for interface design at the team level.

While usability is also an important aspect of product quality, it is mentioned only once for the each of the courses.

#### 2.6.5.2 Comparison across the two project courses

Lessons learned on managing the project are reported more frequently for LTPC compared to STPC. This could be because students need to work on a more complex project in LTPC. There are 44 lesson instances in 31 reports for LTPC, compared to 36 lesson instances in 28 reports for STPC.

At the sub-theme level, there is a varied trend. Focusing more on code quality is a more prominent challenge in LTPC, where the code is submitted by more team members, with 27 lesson instances in 21 reports compared to 6 instances in 5 reports in STPC. Conversely, focusing more on testing, design, and architecture are more frequently reported lessons for STPC than for LTPC.

### 2.6.6 Setting Personal Values and Code of Ethics

Students have reported several lessons on setting personal values and a code of ethics while they worked in team project courses. There are 79 instances of lessons in 57 reports under this theme. This accounts for 7.2% of total lesson instances and 36% of all the reports. There are 12 sub-themes under this (See Table 2.19).

#### 2.6.6.1 Main sub-themes

*Try new technologies without hesitation* has been suggested by students. There are instances where students have mentioned initial discomfort or hesitation in trying new

Table 2.19: Sub-themes of *Setting personal values and code of ethics.*

| Lessons learned (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Try new technologies without hesitation | 10 (10) | 6 (5) | 16 (15) |
| Have confidence and a positive attitude | 8 (8) | 5 (5) | 13 (13) |
| Maintain self discipline | 4 (4) | 8 (8) | 12 (12) |
| Be respectful to others | 4 (4) | 5 (5) | 9(9) |
| Voice your opinion | 5 (5) | 3 (3) | 8 (8) |
| Have breaks from work to refresh | 4 (4) | 3 (3) | 7 (7) |
| Ask questions and take initiatives | 2 (2) | 2 (2) | 4 (4) |
| Keep record of time spent | 3 (3) | 0 (0) | 3 (3) |
| Keep your commitments | 0 (0) | 3 (3) | 3 (3) |
| React early if subjected to bullying | 1 (1) | 1 (1) | 2 (2) |
| Be proactive to solve problems | 1 (1) | 0 (0) | 1 (1) |
| Identify your own skillset | 0 (0) | 1 (1) | 1 (1) |
| **Total for the theme** | **42 (31)** | **37 (26)** | **79 (57)** |

technologies. One of the takeaways for the students from team project courses is to step out of their comfort zone and try new tools and technologies with confidence.

*Have confidence and a positive attitude* is a lesson learned by several students suggesting that self-confidence is reflected in actions and can boost up entire team's morale. Students have also learned to believe in themselves and their team while overcoming the initial doubts and insecurities in doing the project.

*Maintain self-discipline* has been suggested 12 times across 12 different reports. This involves setting up a personal schedule, respecting meeting times and deadlines, being transparent to the team and the customer, and doing one's best at all times.

#### 2.6.6.2 Comparison across the two project courses

Setting personal values and code of ethics has 42 lesson instances in 31 reports for STPC compared to 37 lesson instances in 26 reports for LTPC. The difference is not significant enough. The number of reported instances is similar across the two courses, even at the sub-theme level. Getting to try new technologies without hesitation, having confidence, and having a positive attitude are reported comparatively more for STPC, whereas maintaining self-discipline is a more frequent lesson for the LTPC.

### 2.6.7 Formalizing the Processes

Students have suggested formalizing the process while working on a software engineering team project. There are 40 instances of lessons under this theme, reported in 33 individual reports. This accounts for 3.6% of the total lesson instances and about 21% of all the reports. (See Table 2.20).

**Table 2.20:** Sub-themes of *Formalizing the processes.*

| Lessons learned (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Follow agile practices | 12 (11) | 8 (8) | 20 (19) |
| Document the project and the process | 14 (10) | 2 (2) | 16 (12) |
| Maintain shared repository of documents | 3 (3) | 1 (1) | 4 (4) |
| **Total for the theme** | **29 (23)** | **11 (10)** | **40 (33)** |

#### 2.6.7.1 Main sub-themes

*Follow agile practices* is a lesson learned by several students who have have suggested principally following Agile (SCRUM) practices such as sprint planning meetings, daily standups, sprint review meetings, and sprint retrospectives. Students have also suggested opting for project management roles to understand these practices better.

*Document the project and the process* include lessons on maintaining detailed documentation of the process and the project. This includes documention on the topics researched for the project, changes in the project requirements and even smaller details such as minutes of the meeting. To record and monitor the process, students have suggested using techniques like burndown charts and team velocity metrics, enabling analysis of the team's day-to-day progress.

*Maintain a shared repository of documents* is suggested by several students through setting up this repository using Google Drive, Microsoft Teams, or in the form of a Wiki.

#### 2.6.7.2 Comparison across the two project courses

The need to formalize processes is reported more frequently as a lesson for STPC. There are 29 lesson instances in 23 reports for STPC compared to 11 instances in 10 reports for LTPC. A similar trend follows at the sub-theme level. For all sub-themes, more lesson instances have been reported for STPC compared to LTPC. For instance, documenting the project and the process has been emphasized more in STPC, with 14 instances in 10 reports compared to two instances in two reports for LTPC.

### 2.6.8 Working with New Technology

Students have reported several lessons on working with new technology. There are 40 lesson instances (3.6% of total) in 31 (19.6 % of total) reports for this theme (See Table 2.21).

#### 2.6.8.1 Main sub-themes

*Invest time in learning new technologies* includes lessons like investing appropriate time in learning new technologies and to find solutions for potential issues while

**Table 2.21:** Sub-themes of *Working with new technology.*

| Lessons learned (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Invest time in learning new technologies | 5 (4) | 19 (16) | 24 (20) |
| Make good use of version control system | 10 (10) | 5 (5) | 15 (15) |
| Learn one new technology at a time | 1 (1) | 0 (0) | 1 (1) |
| **Total for the theme** | **16 (12)** | **24 (19)** | **40 (31)** |

working with new technology. Several students found it rewarding to learn a new technology.

*Make good use of the version control system* includes suggestions on making optimal use of version control systems, e.g., Git, Gerrit, and Bitbucket. Specific lessons include creating separate branches for each individual in each sprint, managing changes in separate branches, having a code review, and solving conflicts before merging. Here its important to mention that the idea of separate branches implies using individual branches for specific tasks during the sprint, which are then regularly merged into shared branches (e.g., sprint or main branches) throughout the sprint, in alignment with continuous integration practices.

#### 2.6.8.2 Comparison across the two project courses

Lessons related to working with new technology have been reported more frequently for LTPC than STPC. At the sub-theme level, investing time to learn new technologies has been reported more for LTPC, with 19 instances in 16 reports, compared to five instances in four reports for STPC. On the other hand, efficient use of the version control system is a more frequently reported STPC, with 10 instances in 10 reports, compared to five instances in five reports for LTPC.

### 2.6.9 Taking Advice from the Mentors

Another lesson learned by the students is to seek active advice from the mentor. This lesson has been reported nine times in eight reports. While there are no specific sub-themes, students found it valuable to seek guidance on resolving conflicts, building strong team dynamics, and interacting effectively with customers. Additionally, they reported receiving advice on time management, setting project commitments, project planning, and handling technical challenges. See Table 2.22 for details.

**Table 2.22:** Sub-themes of *Taking advice from the mentors.*

| Lessons learned (sub-themes) | STPC | LTPC | Total |
|---|---|---|---|
| Take advice from the mentors | 6 (5) | 3 (3) | 9 (8) |

#### 2.6.9.1 Comparison across the two project courses

This theme has more instances for STPC when it might be the first experience for students to work in this kind of setting. There are 6 instances in five reports for STPC and three instances in three reports for LTPC.

## 2.7 Discussion

Our study identified ten themes, of which five surfaced in the challenges as well as in the lessons learned (see Table 2.23). Most challenges and lessons learned students reported relate to soft skills, such as working in a team, working in a remote and hybrid setting, and working on a project with industrial customers. This finding aligns with the existing literature [10, 17] on team project courses suggesting that these courses play a crucial role in nurturing the soft skills students require to succeed in the software industry. In their study [16] have mentioned that the students' perceived value of soft skills increases, while that of technical challenges decreases after completing a project course. Findings from our study also highlight more focus on soft skills as the number of challenges and lessons learned related to the soft aspects are quite high as compared to that of technical aspects. (i.e., 91.3 % of all the challenge instances relate to soft skills, similarly, lesson instances reported for soft skills account for 67.8% of the total lessons learned).

In the following subsections, we further discuss and compare with the related works the main results for RQ1 and RQ2 in Section 2.7.1 and RQ3 in Section 2.7.2. Section 2.7.3 discusses the study's implications for teaching practices.

### 2.7.1 Challenges and Lessons Learned (RQ1 and RQ2)

#### 2.7.1.1 Themes related to soft skills

Among the themes related to soft skills, teamwork emerged as the most prominent theme, with challenges primarily centered around communication, contribution, and forming teams with unfamiliar people. Students frequently reported communication difficulties *("it was very difficult to have good communication between all the group members because there were so many of us")*. Another key challenge was uneven contributions within teams *("there was one person in our group who did not do so much work... took it easy because there are others who do everything")*. This issue is also discussed in the literature as "slackers" [12], "free riders" [5], or non-motivated or non-participative team members [2], resulting in unequal contributions among the team members. Additionally, students also found it challenging to form teams with unfamiliar peers *("[i]t was a challenge to identify who we were in the group, what different qualities and prior experiences we have, and how we could use them as efficiently as possible")*. The challenge of forming teams with unfamiliar people is

Table 2.23: Comparison of themes for challenges and lessons learned.

| Theme | Type | Cs% | LLs% | Ref | Leading sub-themes |
|---|---|---|---|---|---|
| Working in a team | S | 73.4 | 88.6 | 2.7.1.1 | Cs: communication and contribution issues, working with unfamiliar people<br>LLs: ensure good communication, create a better team environment and dynamics, effective team meetings |
| Working in a remote and hybrid setting | S | 67.7 | 48.7 | 2.7.1.1 | Cs: remotely communicating with the team<br>LLs: Effective use of collaboration tools for remote work |
| Working with an industrial customer | S | 55.1 | 60.7 | 2.7.1.1 | Cs: understanding requirements<br>LLs: Effective communication with the customer, account for customer perspective while understanding requirements |
| Working with new technology | TD | 29.1 | 19.6 | 2.7.1.2 | Cs: Lack of experience with new technology<br>LLs: Invest time in learning new technologies |
| Managing the project | TD | 27.2 | 71.5 | 2.7.1.2 | Cs: Effort estimation<br>LLs: Invest time in planning, better work distribution |
| Managing the product quality | TD | – | 37.3 | 2.7.1.2 | LLs: Focus more on testing and code quality |
| Setting personal values and ethics | S | – | 36.0 | 2.7.1.1 | LLs: Don't hesitate to try new tools, maintain self-discipline with confidence and positive attitude |
| Formalizing the process | TD | – | 20.8 | 2.7.1.2 | LLs: Follow agile practices, document project and process outcomes |
| Maintaining self discipline and motivation | S | 6.3 | – | – | Self explanatory with only two sub-themes |
| Taking advice from the mentors | S | – | 5.0 | – | Self explanatory theme without any sub-themes |

**Type** = Non-technical or Soft skills (S), Technical or Disciplinary (TD)
**Cs%** = Challenges as a % of 158 reports; **LLs%** = Lessons learned as a % of 158 reports
**Ref** = Subsection where this will be discussed

also reported in the literature [12, 18].

Despite these challenges, teamwork facilitated several valuable lessons learned. Studies such as [17, 18] have identified teamwork and collaboration as primary soft skills developed through project-based learning. The lessons learned include realizing the importance of effective communication *("communication is the key to success for any project....we learned to make sure that good channels and procedures for communication are established")*, and respecting each other's opinion to maintain a conducive team environment *("I learned how to work in a group where everyone has their own ideas and experiences and how to handle all inputs as a group with* *everyone's opinions in mind.")*. The students also noted the value of positive team dynamics *("it was very informative and helpful to learn more about how to work in a team through the group dynamics workshops")*. Other lessons include having effective team meetings and developing a team mindset. The literature also highlights the significance of regular team meetings for improving collaboration and team dynamics [3, 18, 66]. Furthermore, Paasivaara et al. [3] reported that participation in project courses improves students' perception of the importance of communication and collaboration within the team.

Another valuable aspect of team project courses is the opportunity to work with an industry customer. Key challenges in this area included understanding customer requirements *("the main challenge was to work in a team and to understand what the customer wants")*, managing technical complexity in the project *("project was about artificial intelligence and machine learning which just by itself is quite daunting")*, and managing project scope *("the scope changes did not only confuse us but also impact our performance")*.

While mitigating these challenges, students have learned to maintain effective communication with the customer *("[i]t is extremely important to keep the external communication with the customer throughout the project so that everyone is on the same page and no confusion is made")*. Such external communication has been identified as a critical outcome of project courses [17]. For understanding requirements, students have suggested visual representations *("[v]isual feedback is very powerful when working with a customer")*, asking clarifying questions from the customer *("...we asked the customer how they would prefer it")*, and discussing requirements in meetings rather than relying on email exchanges. Working with industrial customers proved insightful *("[w]orking with a real customer and a real company made me see what it can look like in the future and that was industrially exciting")*. The literature suggests that developing real projects for industrial customers motivates students to complete their work [10]. Spichkova [18] also noted that students appreciate the opportunity to practice their soft skills in industry projects, as it helps to bridge the gap between academia and industry. Paasivaara et al. [3] found that students recognized the increased importance of customer communication after completing the course.

Working in remote and hybrid settings introduced additional challenges related to communication *("communication became a bit harder virtually")*, motivation *("being home all the time can make me lose motivation")*, and creating good team dynamics *("it took us a bit longer to create a good group dynamic when working on distance")*.

It is important to note that some sub-themes, such as managing sickness, inability to work on customer premises, and maintaining well-being, are specific to the pandemic. However, many other sub-themes, such as remote team communication, fostering team dynamics in a remote setting, staying motivated while working remotely, tracking team progress, maintaining focus while working from home, and ensuring work-life balance remain relevant to remote and hybrid work environments persisting beyond the pandemic. Therefore, these findings remain relevant to post-pandemic scenarios in those contexts where remote and hybrid work environments continue to exist.

Students reported difficulties due to the lack of in-person interaction, leading to confusion and demotivation. These findings are consistent with [3], where students noted the difficulties of distributed work and the perceived importance of co-location for project success.

On the other hand, some students appreciated the flexibility of remote work, particularly those with long commutes *("...my teammates are a bit introverted and they had a good time working remotely")*. This variation in student perspectives may be because the reports were collected over three years, beginning in 2020, during the onset of the pandemic. This suggests a learning curve as students gradually became more comfortable with remote work and adapted to the strategies necessary for remote and hybrid work settings. Key lessons learned included making effective use of collaboration tools *("[w]e have adapted well and used communication tools to be still able to discuss and work together")* and aligning schedules for teamwork *("having a strict work schedule and being present in discord during work hours also helped a ton as this made the team not miss the social aspects of working")*. These insights point toward a hybrid model that leverages the strengths of both remote and on-site work, tailored to the individual and team needs. The hybrid work style (i.e., working partly onsite and partly from home) has continued post-pandemic in the software industry as well [70–72]. Like the students in our study, practitioners have also expressed positive (e.g., greater flexibility, avoiding commuting time) and negative (e.g., social isolation, team coordination challenges) views of working remotely from home [70, 71, 73]. Companies have formulated policies for a minimum mandatory onsite presence – an attempt to balance the flexibility of remote work with the need for onsite presence [70]. In team project courses as well, such a hybrid working style may be employed, where students and teams must spend a minimum amount of time on campus. However, there is no one-size-fits-all solution for hybrid work [71].

#### 2.7.1.2 Themes related to technical or disciplinary knowledge

The students also reported several challenges and lessons learned related to technical and disciplinary knowledge. These include learning new technologies and managing the team project. Working with the new technologies required for the project posed a steep learning curve for some students, especially given the limited time for the project *("the entire team had a severe lack of knowledge in the specific languages and tools necessary to execute this project. This led to a longer period of research and learning")*. Similar findings were reported by Majanoja et al. [2], who identified insufficient technical knowledge and skills as challenging for the students. However, confronting these challenges broadened students' perspectives on approaching new technologies. They learned not to hesitate when faced with unfamiliar tools *("it* *is always possible to learn new frameworks and programming languages no matter what").* Most students found it rewarding to invest time in learning new technologies *("using new technologies is something I am definitely taking with me").* Similar insights are provided by Spichkova, Raibulet et al. [18, 41], who reported that students often appreciate the opportunity to learn new technologies despite initially facing challenges.

Project management, including effort estimation, project planning, and work distribution, also posed challenges for some students *("[t]he largest challenge we had to overcome in this project was planning...our time estimations ended up being a way off and very ambitious.").* Realizing the importance of planning, the students suggested investing enough time in planning *("I realize the importance of proper planning so that both project teams and clients can accurately predict offerings and deliver a product that meets the agreed requirements").* Similar findings have been reported in Bastarrica et al. [16] and Paasivaara et al. [3], where students' perception of the importance of project planning increased after taking the project course. To ensure clear and equal work distributions, students suggested defining owners for different project parts, clearly communicating the tasks, explicitly delegating roles and responsibilities, proactively tracking team progress, and sharing responsibility for project management tasks. Defining roles within a team is also suggested by Staahl et al. [10], as it can impact the team members' overall experience. Dedicated project management roles, such as timekeeper and minute taker, have also been discussed [12]. While using version control systems, students suggested utilizing the branching feature in GitHub. This recommendation aligns with the findings of Raibulet et al. [41], which highlight the use of branching and merging functions to facilitate concurrent work among team members.

While managing project requirements and scope, students learned the value of flexibility and adapting to scope changes *("we remained flexible and adaptable, we did not let these things ruin the project and instead accepted them").* Being open and flexible to changes is identified as an essential skill acquired by students while working on a project with an industrial customer [17, 18]. The students realized the importance of prioritizing software testing, code quality, and design. The literature also mentions that team projects allow students to realize the importance of testing [10] and quality assurance. Some students have also mentioned how participating in a team project course helped them recognize the value of formalized processes, such as following agile practices and documenting both the project and the process.

We found several studies that focus on specific aspects of team projects, such as teamwork, the involvement of industrial clients, and remote and hybrid work settings. These studies identify challenges and lessons learned in these contexts. However, there are few studies that specifically report on SE team projects involving industrial clients [4, 5]. Our study provides an overview of the challenges and lessons learned from team projects with industrial clients, considering multiple facets of these projects. We not only identify high-level themes but also present a hierarchy of sub-themes, challenges, and lessons learned within each theme. Another contribution of our study is the mapping of challenges to corresponding lessons learned. This offers insights into the strategies students employ to address the challenges they encounter in team projects.

### 2.7.2 Evolution of Challenges and Lessons Learned across Team Project Courses (RQ3)

**Table 2.24:** Increase or decrease in the challenges and the lessons learned from STPC to LTPC (decreases are negative). The differences are reported as the number (%) of more or less reports/students for a theme.

| **Theme** | **Challenges** (LTPC-STPC) | **Lessons** (LTPC-STPC) | **Comments** |
|---|---|---|---|
| Working in a team | 10 (6.3%) | 8 (5.1%) | Relatively more students reported teamwork challenges and lessons in LTPC, indicating teamwork's continued and enhanced importance in the second project course with large teams. |
| Working in a remote and hybrid setting | -9 (5.7%) | -1 (0.6%) | Relatively fewer students reported challenges under this theme in LTPC, potentially due to enhanced familiarity of students with the hybrid way of working. |
| Working with an industrial customer | -1 (0.6%) | -10 (6.3%) | Relatively fewer students reported challenges and lessons in LTPC under this theme, which may indicate their improved familiarity to work with the industrial customer after having completed the STPC. |
| Working with new technology | -12 (7.6%) | 7 (4.4%) | Relatively fewer students reported challenges and more reported lessons under this theme in LTPC. This may indicate that, on the one hand, the students have become more experienced in working with new tools and overcoming challenges (hence fewer challenges). On the other hand, they have more to reflect on and share as lessons about learning new tools. |
| Managing the project | -5 (3.2%) | 3 (1.9%) | Like the previous one, fewer students reported challenges and more reported lessons related to managing the project in LTPC. Relatively fewer students found it more challenging to estimate effort and make initial plans in LTPC than in STPC, indicating improved project management skills when they moved to LTPC. |
| Managing the product quality | – | 3 (1.9%) | Only lessons learned are reported under this theme, with only a few more students reporting lessons in LTPC. |
| Setting personal values and ethics | – | -5 (3.2%) | Only lessons learned are reported under this theme, with only slightly fewer students reporting lessons in LTPC. |
| Formalizing the process | – | -13 (8.2%) | Only lessons learned are reported under this theme, indicating that the students don't perceive following development processes as challenging. The focus on the process and product documentation in lessons seems to have reduced in LTPC, which could be an interesting topic for teachers to look into. |
| Maintaining self discipline and motivation | -2 (1.3%) | – | Not much difference between STPC and LTPC for highlighting any topic. |
| Taking advice from the mentors | – | -2 (1.3%) | Not much difference between STPC and LTPC for highlighting any topic. |

Table 2.24 summarizes the evolution of challenges and lessons learned as students progressed from STPC to LTPC, highlighting the differences in the number of reports or students reporting each theme. A negative value indicates a decrease in the number of reports as students progressed from STPC to LTPC.

For most of the themes, the results show that students in LTPC faced relatively fewer challenges and shared fewer lessons, potentially due to their prior experiences in STPC. This can be attributed to the learning curve of the students as they progressed from STPC to LTPC.

However there is an exception to this trend, i.e., working in a team, which shows a higher number for both challenges and lessons learned as the students progressed from STPC to LTPC. Students have reported more issues related to communication and coordination in LTPC, potentially due to the large team size and sub-team structure. The exposure to such a dynamic also led more students to reflect on the lessons they learned during the project.

Increasing trend can also be observed for lessons learned, across three themes relate to technical or disciplinary knowledge (i.e., working with new technology, managing the project and managing product quality). This can be attributed to increased technical complexity of the projects that exposed the students to more diverse set of technologies leading to an increased number of lessons learned. This may also be due to students' increased ability and confidence to reflect on these technical or disciplinary topics as they become more expert in working with the new technology, product quality, and project management.

Exploring how the challenges and lessons learned by students evolved as they progressed through a sequence of courses represents a unique contribution of our study. To the best of our knowledge, we have not found any similar studies in the existing literature that track and analyze this progression over multiple courses.

### 2.7.3 Implications for teaching team project courses

Our results indicate that software engineering team project courses not only enhance students' technical and disciplinary knowledge but also foster the soft skills essential for their careers. These findings from this study are communicated to the teaching staff to identify areas in the two project courses that can be improved to further enrich the students' learning experience. The study has the following implications for teachers:

1. Since the students struggle more with soft issues (e.g., communication, contribution, and coordination issues), early awareness and support about these issues through, for example, lectures, seminars, and mentor guidance, may help students to effectively manage and address these soft issues.

2. Besides soft issues, students also reported challenges related to learning, managing, and selecting new tools and technology. The complexity of the required

tools and technology may vary among projects and industry customers. To address this, it is important for teachers to ensure that students receive timely guidance from team mentors and industrial customers to make informed technology choices and invest enough time to learn new technologies.

3. The evolution of challenges and lessons learned across the two team project courses show that participation in multiple project courses with industrial customers benefits students in progressively adapting to expectations of working on a real software project. Including multiple team project courses instead of a single one toward the end of software engineering degree programs seems more beneficial for students' learning.

4. The involvement of industry customers has its own challenges, such as requirements and scope management. However, as the students' reflections on the lessons learned show, it provides students with a valuable real-world learning experience, which otherwise may not be easy to offer in a simulated setup.

5. Team project courses provide a valuable learning setup for teachers, industry customers, and students to experience the dynamics of hybrid work before students enter the industry. Companies continue to adopt policies for a hybrid working style that balances the flexibility of individuals with the needs of teams and organizations [70].

6. Learning by doing and reflection are very important practices to prepare students as future software engineers. Team project courses with industry customers expose students to real challenges. Reflecting on the challenges faced and the lessons learned is essential for continuous learning [68].

7. Students reported challenges related to unclear requirements and misaligned expectations when working with course teachers, team mentors, and industry customers. To address this, effective coordination among teachers, mentors, and industry customers is needed, starting early to define project goals and continuing throughout the course during critical activities such as requirement gathering and sprint deliveries. This may help to reduce misaligned expectations and mixed messages to students from teachers, mentors, and industry customers.

## 2.8 Threats to Validity

We conducted a qualitative study, making the validity concerns identified by Maxwell [63] relevant to our work. We used the framework developed by Petersen and Gencel [64] to identify specific threats related to theoretical validity, interpretive validity, and generalizability of the collected data. The threats to validity were considered throughout the design and execution of the study, and guided data collection and analysis.

### 2.8.1 Theoretical Validity

To mitigate risks associated with *poorly defined constructs*, we defined a clear scope for the study. The artifacts used for data collection, i.e., the individual reflection reports, contained well-defined sections for challenges and lessons learned by students in the two project courses, allowing us to identify and collect all relevant data.

The use of final reflection reports as the only data source posed a risk of omitting significant reflections from weekly diaries, potentially introducing *social desirability bias*. However, this was addressed by ensuring students based their final reports on weekly diaries. Further the mentors triangulated the content of these reports with the weekly diaries. The fourth author, who is also an instructor within the team project courses provided further insights into the credibility of the process. Threats related to *evaluation apprehension* and *reactivity* are irrelevant in our study, as the artifacts used for data collection predated the study, ensuring that the student reflections were not influenced by the study.

To address potential *biases in subject selection*, we included all reports submitted by students across two courses and two cohorts, thereby ensuring a representative sample. Data was manually collected and recorded in MS Word files, eliminating concerns regarding *poor instrumentation*. The first two authors further reviewed the data extraction process, ensuring completeness and accuracy in capturing relevant information.

### 2.8.2 Descriptive Validity

Threats related to *quasi-statistics* were mitigated by implementing measures to avoid over-reporting. Specifically, if a student mentioned the same challenge or lesson learned multiple times within a single report, it was coded only once to ensure accurate representation. Additionally, the analysis accounted for both the number of unique reports and the instances of challenges reported, thereby reflecting the actual number of students who identified a particular challenge in the course.

### 2.8.3 Interpretive Validity

To ensure *interpretive validity*, we followed thematic analysis guidelines [60–62] and used a checklist from Cruzes et al. [61].

To eliminate *researchers bias*, the first author led the thematic analysis, and the first two authors reviewed all phases collaboratively, ensuring consistency and rigor. Details of this process can be found in Section 2.4.

To avoid *selective interpretation*, we ensured that all the text in the relevant sections of the reports was systematically coded, maintaining a clear and traceable connection between the text and the assigned codes.

To mitigate the risk of *loss of context*, data from all course instances was stored in separate MS Word files and imported into NVivo as separate files. This approach preserved the context of each code instance, ensuring the integrity of the data during analysis.

### 2.8.4 Generalizability

The generalizability of our findings is limited to contexts similar to those of our team project courses. To facilitate understanding and applicability of our findings, we provided a comprehensive description of the study context (see Section 2.3), which would aid readers in evaluating the relevance of our study and its results to their contexts. One of the themes identified in our study relates to remote and hybrid work settings. As discussed in Sections 2.3, 2.5, and 2.6, these setups emerged as a consequence of pandemic related restrictions. However, within the context our project courses and overall the software industry, remote and hybrid work settings continued to persist even after the pandemic. There is an increasing shift towards remote and hybrid setups, making our findings relevant beyond the context of pandemic (apart from a few sub themes which were specific to pandemic).

We also reviewed our approach to the identification and mitigation of threats to validity using the checklist proposed by Lago et al. [74]. Apart from the item related to the discussion of the trade-off among various threats to validity, our approach addresses all other checklist items (e.g., characterizing threats to validity using the checklist by Petersen and Gencel [64]), partially or completely.

## 2.9 Conclusion

The study results show that the challenges related to soft skills, particularly teamwork, remote and hybrid ways of working and working with an industrial customer, are prevalent and correspond closely with the most important lessons learned. The main themes for challenges and lessons learned are closely aligned, indicating a connection between the difficulties faced and the educational outcomes achieved.

Involving industrial customers in team project courses can be challenging, but it can substantially impact student experiences and learning outcomes. Students gain invaluable insights into industry practices, project management, and collaborative problem-solving by working on real projects with industrial customers. In addition, the experience of working as a team in a remote and hybrid setting also prepares students for a scenario that is becoming increasingly prevalent in the software industry. These experiences equip students with the competencies and skills to integrate successfully into the software engineering workforce.

Finally, the evolution and progression of students' learning experiences in the two team project courses highlight the importance of exposing students to such team projects early and frequently in software engineering degree programs.

# Appendix A: Weekly Individual Diary- Template

Weekly diary-blog is a mandatory part of the course. Create and update a new entry each week. Contents

## 1. Week

- Week number.
- Hours worked this week.
- Total project hours so far.

## 2. Activity & result

- Main activities completed.
- Outcomes or progress achieved.

## 3. Challenges & solutions

- Key challenges faced.
- How they were addressed/resolved.

## 4. Reflection

- Key insights and learnings from the week.

## 5. Submission instructions

- Use tools like Canvas (e-Portfolio), Google Docs, or OneDrive for your diary.
- Update the document weekly.
- Share the document link here as the submission.

# Appendix B: Individual Reflection Report-Template

## 1. Introduction

- *Project overview:* Purpose, main functionality, and system architecture.
- *Customer details:* Name and contact information.

## 2. Project results

- *Deliverables:* Delivered features, satisfying features, and features that need improvement.
- *Deviations:*
  - Commitments: Features not delivered or delivered beyond expectations.
  - Costs: Planned vs. actual effort and total hours spent, including your contribution.

## 3. Learning

- *Development methods:* Briefly describe known models (Sequential, Iterative, Agile) and how your team structured the work.
- *Requirements analysis and customer interaction:* Compare standard practices with your team's approach.
- *Planning and control:* Estimation accuracy, activity order, and improvement opportunities.
- *Technology:* Summarize technical solutions, tools used, and new knowledge gained.
- *Testing and quality assurance:* Outline testing methods and quality assurance practices.
- *Risk management:* Highlight identified risks, monitoring, and contingency plans.
- *Tools and techniques:* List tools used for requirements, planning, version control, and testing.

## 4. Personal experiences

- Challenges faced during the project. –(Focus of this study)
- Key lessons learned during the project -(Focus of this study)
- Major events/decisions impacting the project.

## 5. References

List all sources with proper citations.

# 3 Peer Evaluation in Software Engineering Team Project Courses: A Taxonomy and Guidelines for Educators

*Nayla Nasir, Muhammad Usman, Jürgen Börstler*
*In revision with the: Journal of Systems and Software, 2026.*

## Abstract

Peer evaluation is widely used in team project courses in software engineering to assess individual contributions, promote accountability, and improve teamwork. However, current practices in peer evaluation vary significantly in terms of purpose, design, and implementation, which makes it challenging to compare them and establish best practices.This study systematically investigates peer evaluation in software engineering team project courses. In a literature review spanning from 2007 to 2025, we identified 50 articles that discuss the use of peer evaluation in these courses. By iteratively coding the findings and employing a thematic synthesis approach, we developed a taxonomy to characterize peer evaluation practices across four dimensions including their context, participants, mechanisms, and outcomes. Using this taxonomy, we organize the findings from existing literature and characterize how peer evaluation is applied across different courses. Additionally, we interviewed seven SE educators to collect their perspectives on the quality and applicability of the taxonomy. Based on their feedback, we refined the taxonomy and provided a set of guidelines to support educators and researchers in designing and implementing peer evaluation interventions, as well as reporting them in a structured and informed manner.

## 3.1 Introduction

Team project courses play a significant role in preparing students in software engineering (SE) or related fields for their work in real-world software development environments [5, 75]. These courses provide opportunities to develop technical competencies as well as essential professional skills such as collaboration, communication, and problem-solving [16, 18, 76]. As team project courses are central to SE education, they often simulate industry practices, where effective teamwork and clear communication are critical for success.

Despite their pedagogical value, SE team project courses are not without challenges for both students and educators [2, 75, 77]. One of the most prominent challenges in SE team project courses is the development of effective team dynamics [76]. Students often struggle with coordination and communication, leading to misaligned expectations, conflict, and delays in project progress [2]. Another recurring issue is unequal contribution from team members, often referred to as "free-riding" behavior [5], or the presence of non-participatory members [2]. This unequal contribution can undermine team cohesion, frustrate engaged members, and affect the project's overall success. These challenges are detrimental to the learning experience and hinder the development of essential interpersonal skills, which are crucial in the field of software engineering.

Peer evaluation has emerged as a valuable tool for addressing these issues in SE team project courses. By enabling students to evaluate contributions of other students [78], peer evaluation promotes accountability, encourages engagement, and provides structured feedback at both the individual and team level [79]. It can mitigate free-riding by making contributions visible and valued while promoting self-reflection and critical thinking that extend beyond technical skills [44]. In addition, peer evaluations give instructors insights into team dynamics, including each member's performance and role, enabling them to detect potential problems early, such as low participation or interpersonal conflicts [79].

The literature describes various methods and approaches for peer evaluation that are employed across different educational contexts. These methods include anonymous peer reviews [80], 360-degree feedback [81], and hybrid approaches combining self and peer evaluation [44, 82, 83]. While these approaches address common challenges, their effectiveness is highly context-dependent, and there is little agreement on best practices. Furthermore, there are differences in the terminology used for peer evaluation (e.g., peer review, peer feedback), the aspects of performance being assessed (e.g., technical or other contributions), the evaluators (individuals or teams), and the evaluatees (e.g., team members, team, other teams). Additionally, existing studies do not consistently report peer evaluation. Many studies miss contextual and procedural details on peer evaluation practices. For instance, some studies [51, 84, 85] do not provide any information about the evaluators or evaluatees. Similarly, many studies do not specify the evaluation criteria used for peer evaluation or how often peer evaluation was conducted [84, 86–88]. The differences and inconsistencies in the reporting make it difficult for educators to make informed design choices.

Peer evaluation has been discussed in previous secondary studies [5, 89]. However, these works examine peer evaluation in broader educational contexts or treat it as a secondary aspect within SE courses. To our knowledge, no prior work provides a structured classification that systematically organizes peer evaluation practices across multiple dimensions in SE team project courses. This gap highlights the need for a structured classification to capture different aspects of peer evaluation in SE team project courses. Such a classification would facilitate the comparison of findings across studies and aid educators in designing evaluation strategies tailored to their specific course contexts. It would also enable more rigorous and comparable reporting in research on peer evaluation in SE team project courses.

This study addresses this gap by systematically investigating peer evaluation in SE team project courses. We reviewed literature on SE team project courses published between 2007 and 2025 and identified 50 papers discussing the use of peer evaluation in such courses. Through iterative coding of the literature findings and employing a thematic synthesis approach [61], we developed a taxonomy to categorize peer evaluation practices. This taxonomy considers four key dimensions, each consisting of 2 to 4 facets. The main dimensions include: i) *Participants* – who evaluates whom?, ii) *Context* – why conduct peer evaluation?, iii) *Mechanism* – how peer evaluation is operationalized?, and iv) *Outcome* – What are the peer evaluation outcomes and how to use them? An overview of the proposed taxonomy is shown in Figure 3.6.

In the next step (see Figure 3.1 in Section 3.3) , we interviewed seven SE educators who utilize peer evaluation in their courses and gathered their insights on the quality and applicability of the taxonomy. Based on the interview findings (see Section 3.4.4, we refined the taxonomy, which now includes five dimensions (a new dimension: *Course context* was added. There are also some changes at the facet level– see Fig 3.6 in Appendix C).

Based on our taxonomy, we also provide a set of guidelines to assist educators and researchers in designing and implementing peer evaluation interventions, as well as in reporting them in a structured and informed manner

The contributions of this study are as follows:

C1. A taxonomy for peer evaluation practices in SE team project courses.

C2. A categorization of peer evaluation practices in the existing literature on SE team project courses using this taxonomy.

C3. A set of guidelines, based on this taxonomy, to support the design and implementation of peer evaluation interventions in SE team project courses.

## 3.2 Background and Related Work

**Table 3.1:** Peer evaluation aspects reported in related work

| Aspect | Value | Studies |
|---|---|---|
| Terminology used | Peer avaluation | [78, 79, 90–92] |
| | Peer assessment | [21, 93, 94] |
| | Peer feedback | [50, 51] |
| | Peer review | [48, 49] |
| | Self- and peer rating | [95] |
| | Self- and team evaluations | [96] |
| Forms of evaluation | Individuals evaluating team-mates | [23, 78, 79, 94] |
| | Teams evaluating other teams | [97] |
| | Hybrid (using multiple forms of peer evaluation) | [98, 99] |
| Evaluation model | Peer Rating | [21, 24, 45, 78, 79, 94, 100, 101] |
| Frequency of PE | Iterative | [23, 79] |
| Use of PE | Used in grading | [79, 88] |
| | Validated with teacher feedback | [79, 88, 95] |
| | Validated with objective records of effort | [23, 92] |
| Evaluation criteria | Evaluation criteria specified | [78, 79, 94, 101] |

Peer evaluation is an effective approach for enhancing teamwork, motivation, and conflict resolution in SE education [20]. Adapting from the definition of peer evaluation given in [79], we define peer evaluation as *"a process in which individuals [21, 77–79, 93] (or, in some cases, teams [97]) assess the extent, value, and quality of their peers' contributions, performance or produced artifacts, mostly by rating them [21, 24, 45, 78, 79, 94, 100, 101] against different aspects of quality or performance, or in some cases, providing qualitative feedback as well."* The existing literature on peer evaluation in SE team project courses refers to peer evaluation using diverse terminology (see Table 3.1), including "peer evaluation", "peer assessment", "peer feedback", "peer review", "self- and peer rating", and "self- and team evaluations", with some studies using multiple terms interchangeably.

Peer evaluation is implemented in various forms depending on the evaluators and evaluatees involved. Some studies focus on individual students evaluating team members [23, 78, 79, 94], while others examine team-to-team evaluations [97]. A few adopt hybrid approaches that combine multiple forms of evaluation (e.g., self, peer, and team assessments) within a single course context [98, 99].

Different models of peer evaluation have also been discussed, including the peer-ranking model, the peer-nomination model, and the peer-rating model [24]. Depending on course objectives, these models differ in complexity, granularity, and applicability.

Table 3.2: Systematic reviews reporting peer evaluation in context of computing education

| Title | Ref | Findings on peer evaluation |
|---|---|---|
| A systematic literature review of capstone courses in software engineering | [5] | Identified studies employing peer evaluation as one of the forms of assessment in SE team project courses, but they did not investigate peer evaluation in detail. |
| Does peer assessment in on-line learning environments work? a systematic review of the literature | [89] | Analysed the state of the art on the use of peer assessment techniques in on-line educational environments. However, this study was not conducted in context of SE team project courses, and the scope of its findings is also different. It considers aspects like contexts and educational levels where peer review has been applied most, goals of peer assessment, its impact on student learning, its usefulness for teachers, and challenges involved. |

Many studies report the motivation for and frequency of peer evaluation; however, only a few investigate it in depth [23, 79, 88, 92, 95]. These studies explore iterative peer evaluation [23, 79], integration of peer evaluation into grading schemes [79, 88], triangulation with teacher input [79, 88, 95], and validation using objective records such as GitHub contributions [23, 92]. Similarly, only a limited number of studies explicitly specify evaluation criteria [78, 79, 94, 101], with most providing minimal detail.

A recent study [20] proposes a structured approach for peer evaluation, showing that structured peer evaluation can enhance fairness and student satisfaction. However, the authors have considered only a few aspects of peer evaluation including transparency, anonymity, structured assessment, and qualitative feedback.

Only a few secondary studies (see Table 3.2) synthesize the current knowledge on peer evaluation in computing education. Tenorio et al. [89] conducted a systematic review on peer assessment in online learning, discussing its goals, impact on student learning, benefits for teachers, and implementation challenges. Whereas, in a literature review on capstone courses, Tenhunen et al. [5] examined peer evaluation as an aspect in SE team project courses, but they did not investigate peer evaluation in detail.

The discussion above demonstrates that peer evaluation practices vary significantly in terminology, purpose, and approaches, making cross-study comparison challenging. To our knowledge, no structured classification exists to organize and synthesize the state of the art on peer evaluation in SE team project courses. This highlights the need for a structured classification that can serve as a framework for designing, analyzing, and comparing peer evaluation practices in a consistent manner.

## 3.3 Research Methodology

The main aim of this study is to characterize peer evaluation practices in team project courses in SE.

### 3.3.1 Research Questions

- **RQ1:** What peer evaluation practices have been reported in the existing literature on SE team project courses?

- **RQ2:** How can peer evaluation practices in SE team project courses be characterized?

- **RQ3:** How useful is a structured characterization of peer evaluation practices from the educators' perspective?

We used a mixed-method research design that combined a literature review with semi-structured interviews. This design helped us examine peer evaluation practices in software engineering (SE) team project courses from both research and practical perspectives.

Using a literature review, we explored how peer evaluation has been described and implemented in SE team project courses. During data extraction, it became apparent that the descriptions of the peer evaluation practices employed in the primary studies varied substantially in terms of the level of detail and the type of information provided.

We therefore developed a taxonomy for peer evaluation in SE team projects to facilitate a systematic and consistent data extraction.

As shown in Figure 3.1, this led to an iterative process of data extraction, taxonomy development, and eventually a mapping of the peer evaluation practices reported in the primary studies to this taxonomy. That is, insights from the literature informed the design of the taxonomy, while the evolving facets and dimensions in the taxonomy guided subsequent rounds of literature analysis to identify missing or overlooked aspects.

To answer RQ3, we conducted semi-structured interviews with seven SE educators who had experience using peer evaluation in their team project courses. The aim was to explore the usefulness and applicability of the taxonomy in the context of these courses.

The mapping of the taxonomy to the existing literature and the interviews helped ensure that the taxonomy was evidence-based and practically relevant.

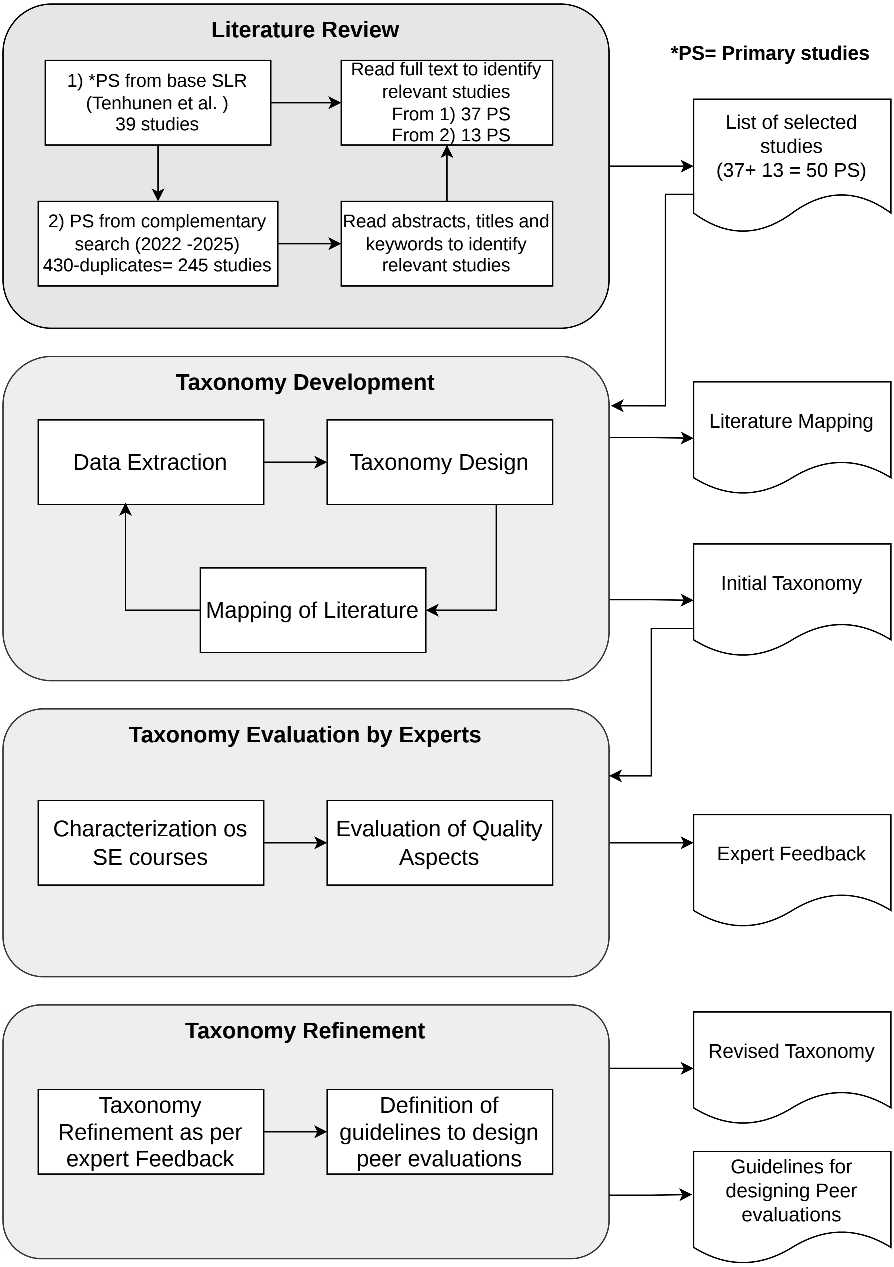



**Figure 3.1:** Overview of the research methodology.

### 3.3.2 Literature Review

To understand the current state of research and practice on peer evaluation in SE team project courses, we conducted a literature review. The literature review aimed to identify, analyze, and synthesize published studies that discuss peer evaluation practices.

The process consisted of two main steps:

- *Study search:* The search was conducted in three phases. In Phase one, we utilized the systematic review by Tenhunen et al. [5], which identified 39 studies published between 2007 and June 2022 that reported peer evaluation in SE project courses. Two of those 39 studies were excluded as they focused solely on self-evaluation.

  In Phase two, we complemented this initial set of 37 primary studies by searching for more recent publications (2022–2025) in Scopus. We employed two separate search strings, following the approach used by Tenhunen et al. [5]. The first search string was *"software AND project course"*, and the second was *"software AND capstone"*. In Phase three, we expanded the search further and applied the same search strings in the ACM Digital Library and the IEEE Xplore. In total, 430 new primary studies were retrieved using the search strings on three databases. Building on Tenhunen et al. [5], we restricted the search results from the three databases to publications from Jan 2022 to Dec 2025.

  After removing the duplicates, we had a set of 245 unique primary studies. We scanned through the abstracts to identify studies that mentioned peer evaluation and/or its related terms (e.g., peer assessment, peer feedback, peer review) and/or discussed student assessment or evaluation in general. The selected studies were further screened for inclusion or exclusion based on whether they discussed any aspect of peer evaluation. After applying the inclusion criteria, thirteen additional studies were included, bringing the total number of studies to 50 ($N_s = 39 - 2 + 13 = 50$).

- *Data extraction and analysis:* We developed a data extraction form[1] to capture the characteristics of the study (e.g., publication year, location), the context of the course (e.g., level of education, size of the teams, duration), and peer evaluation aspects (e.g., why do peer evaluation, who was evaluated by whom, what was evaluated, and how peer evaluation was implemented). For calibration purposes, data from ten studies were independently extracted by three authors, then compared and discussed to ensure consistency. Subsequently, the first author continued extracting data from all identified studies. Once extraction

[1] Data Extraction form and mapping of literature to the revised taxonomy

was completed, the data were collaboratively analysed by all authors using thematic synthesis [61].

The extracted data showed that peer evaluation practices were reported in very different ways. Some studies described the process in detail, while others omitted essential contextual or methodological details. This made it difficult to extract relevant data from the primary studies in a consistent way and motivated the need for a structured taxonomy, supporting a systematic way to describe, compare, and classify peer evaluation practices across contexts.

### 3.3.3 Taxonomy Development

The taxonomy development followed an iterative and collaborative process combining bottom-up data extraction with top-down conceptual refinement (see Figure 3.1). Insights from the literature informed the design of the taxonomy, and the evolving taxonomy structure guided further analysis of the literature. For taxonomy development, we adopted the guidelines proposed by Usman et al. [102].

The taxonomy development process consisted of the following phases:

- *Select classification structure:* We chose faceted classification because it allows the subject matter to be classified from multiple perspectives (facets) [103]. This approach provides flexibility by allowing each facet to have its own attributes, making it suitable for multifaceted concepts like peer evaluation.

- *Identify data sources:* Our primary data sources were the 50 primary studies from the literature review described in Section 3.3.2).

- *Identify facets:* Using their prior knowledge in peer evaluation, the first and the second author identified an initial set of peer evaluation aspects (e.g., why do peer evaluation, who was evaluated by whom, what was evaluated, and how peer evaluation was implemented) to create the data extraction form. The first author then extracted data on peer evaluation from the primary studies. The extracted data was categorized thematically to identify peer evaluation facets. This was an iterative process in which data extraction led to the identification of facets, and the emerging set of facets, in turn, guided the data extraction process (see Figure 3.1). The facets were recorded in the data extraction form and then reviewed collaboratively with the second and third authors through discussions.

- *Identify dimensions:* All three co-authors collaboratively categorized related facets into candidate dimensions. Three of the dimensions, i.e., context, participants, and mechanism mapped to the initial peer evaluation aspects used to design the data extraction form, whereas dimension Outcome emerged from thematic analysis of the identified facets. Disagreements on categorization were resolved through discussion until consensus was reached.

- *Refinement:* Identified facets and dimensions were refined in several iterations. Redundant or weakly supported facets that did not contribute meaningfully to peer evaluation processes were removed (pruning). The authors also analyzed the taxonomy for any missing but necessary design elements based on their experience of teaching SE project courses.

- *Define values for each facet:* For each facet, a preliminary set of values was defined based on the data extracted from the primary studies. These values provide a foundation for using the taxonomy or adapting it to specific contexts.

### 3.3.4 Mapping of the Literature to the Taxonomy

The mapping of the reviewed literature to the proposed taxonomy is presented in 3.4.3.

### 3.3.5 Interviews with SE Educators

To investigate the taxonomy's usefulness and applicability beyond our set of primary studies, we conducted semi-structured interviews with seven SE educators. All interviewees had experience using peer evaluation in team project courses.

The first author conducted the interviews during an international conference on computing education in 2025. Participants were recruited from a working group on team project courses. The first author was one of the 12 members of that working group. All other working group members were invited to participate, of whom six agreed. Participants were selected using convenience sampling [104], but represented a demographically rich sample, allowing us to explore peer evaluation practices from diverse perspectives.

No formal ethical approval was required to conduct these interviews under Swedish regulations, as the study involved voluntary participation by adult professionals and did not collect sensitive personal data.

The interviews consisted of two parts. In the first part, the interviewees were asked to fill out a spreadsheet template to characterize their peer evaluation practices using the taxonomy. In the second part of the interviews, the focus was on evaluating the quality of the taxonomy, inspired by Kaplan et al.'s [105] evaluation method for taxonomies. Each interview lasted about 60 minutes, was conducted with participants' consent, audio-recorded, transcribed, and later analyzed qualitatively. The interview guide is provided in Table 3.16 (Appendix B) and was pilot-tested with two experienced educators.

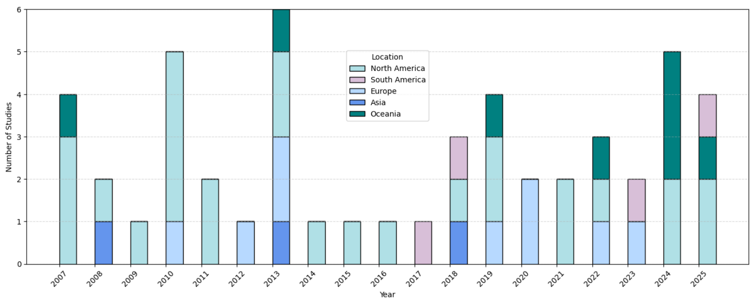
## 3.4 Results and Analysis

The results are organized into four subsections. In Section 3.4.1, we provide an overview of the reviewed literature, including demographics on the courses reported in this literature. Section 3.4.2 introduces the proposed taxonomy. In Section 3.4.3, we use this taxonomy to structure and organize the current state of the art on peer evaluation practices in SE team project courses. Section 3.4.4.1 presents how interviewees characterized their peer evaluation practices using the taxonomy, and Section 3.4.4.2 reports their perceptions of different quality aspects of the taxonomy.

### 3.4.1 Literature Review: Overview

We identified 50 primary studies (see Appendix A: Table 3.14), *published between 2007 and 2025*, and *discussing some aspect of peer evaluation in SE team project courses*. A base set of 37 studies, published between 2007 and 2022, was identified from [5] (S1–S37 in Table 3.14). Thirteen more studies (SS1–SS13 in Table 3.14) were identified using a complementary search for the years 2022 to 2025.

Figure 3.2 shows the distribution of primary studies over the time period (2007–2025) across different venues and geographical areas. The majority of primary studies report courses from North America (26 of 50), with the USA as, by far, the country with the most contributions. Some European countries, including Finland, Germany, Greece, Sweden, Norway, and Portugal, also contributed studies (9 of 50). The remaining studies exhibit a large spread across the rest of the world, with at most four studies from a single country. Most studies are published in conferences (35 of 50) , rather than in journals (15 of 50).




| | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Conference | 3 | 1 | 1 | 4 | 1 | 0 | 6 | 0 | 1 | 1 | 1 | 1 | 3 | 2 | 1 | 1 | 1 | 5 | 2 |
| Journal | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 2 | 1 | 0 | 1 | 2 | 1 | 0 | 2 |

**Figure 3.2:** Distribution of the primary studies over time ($N_s = 50$) by study location (upper part) and publication venue type (journal, conference; lower part).

Figure 3.3 shows the course demographics for the courses discussed in the primary studies. Most studies (40 of 50) were conducted on undergraduate courses (see left part of Figure 3.3). The most common course setup (10 of 40) in these courses is small software development teams (3–5 students) and course durations of two semesters (around six months). Medium-sized software development teams (6–10 students) were also common at the undergraduate level, with seven studies reported for courses lasting for two semesters. Only two undergraduate courses had large software development teams (11–15 students), and none had very large teams. Three studies do not specify the duration of the undergraduate courses. Two studies do not mention the team size.

Only 6 of 50 studies reported graduate-level courses (see middle part of Figure 3.3). Most involved small to medium-sized software development teams, with course durations ranging from one semester to three semesters. One study reported using a very large software development team (20–35 students) with no specified course duration.Four studies involved courses taught at both undergraduate and graduate levels.All four were conducted with small teams and in shorter courses.

The majority of our primary studies reported on courses that were conducted at the undergraduate level (40 of 50) with small sized teams (20 of 40) and medium sized teams (11 of 4, respectively. Out of 50 studies, 24 studies reported on courses that were two semesters long (3 to 6 months).

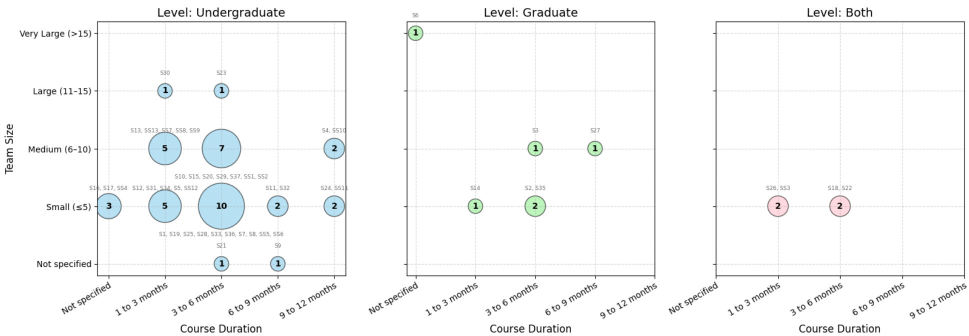



**Figure 3.3:** Course level, duration, and team sizes across courses using peer evaluation in the primary studies ($N_s = 50$, including two studies with no team sizes specified).

## 3.4.2 Peer Evaluation Taxonomy: First Version

The proposed faceted taxonomy comprises four main facets (or dimensions): **participants**, **context**, **mechanism**, and **outcome**. Each of these four dimensions has 2–4 subfacets, see Figure 3.4 for details. The taxonomy was later refined based on the interview findings. The revised version is presented in Appendix C, Figure 3.6.

### *3.4.2.1 Participants*

The participant dimension captures *"who evaluates whom?"*. Evaluators are the entities (individuals or teams) carrying out the evaluation. Evaluatees are the entities (in-

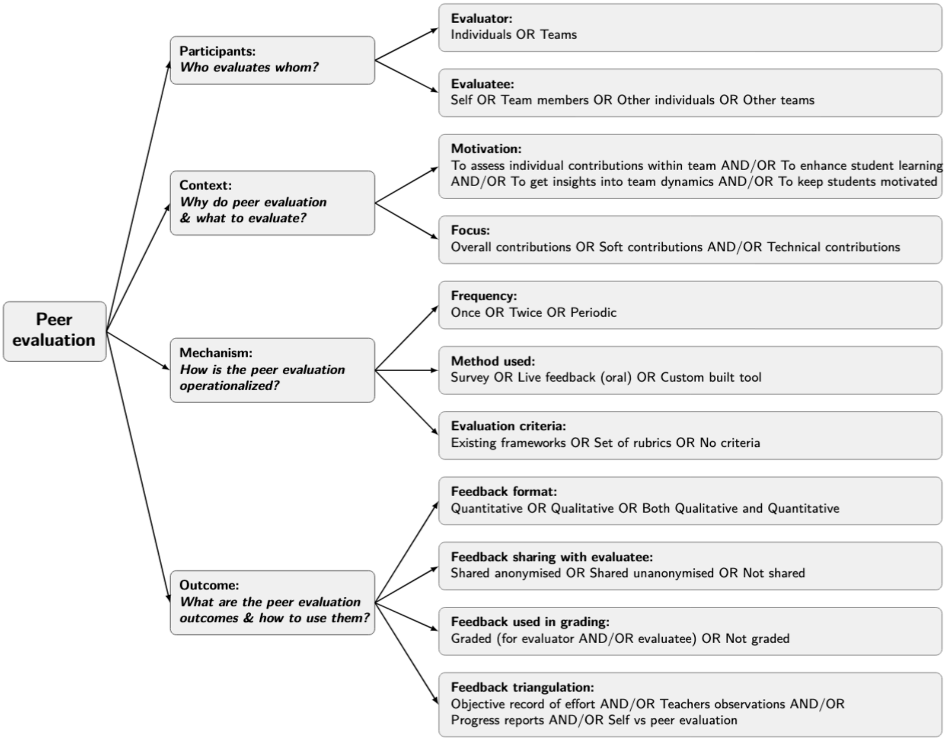



**Figure 3.4:** Taxonomy for peer evaluation (V1): Dimensions (Participants, Context, Mechanism, Outcome), facets, and values.

dividuals or teams) being evaluated. This captures multiple setups of peer evaluation, for instance, individuals evaluating themselves or their team members, individuals evaluating their own team or peer teams, etc.

#### 3.4.2.2 Context

The context dimension captures the rationale for peer evaluation *"why and what to evaluate?"*. Peer evaluations may have various motives, including assessing individual contributions, enhancing learning in terms of promoting reflection and critical thinking, understanding team dynamics, and promoting accountability, motivation and engagement. The focus or object of an evaluation can vary from high level, overall team contributions to either soft contributions (e.g., communication, teamwork, collaboration, leadership etc) and/or technical contributions (e.g. project artifacts and their quality).

#### 3.4.2.3 Mechanism

The mechanism dimension captures *"operational details"* of the peer evaluation process, which are captured in the facets: frequency, method, and evaluation criteria. Evaluations may occur once, twice, or periodically throughout the course, providing either one-time or continuous feedback. Methods range from structured surveys to custom-built tools. Evaluation criteria may utilize evaluation frameworks (e.g., CATME [106]), a set of rubrics (see, e.g., [107]), or be open-ended questions to facilitate qualitative observations.

#### 3.4.2.4 Outcome

The outcome dimension captures *"what are the peer evaluation outcomes and how to use them"*. Peer evaluation can be quantitative, qualitative, or a combination of both, offering objective metrics and detailed insights. The evaluations may be shared anonymously, openly, or only used internally by teachers, and may or may not be used in grading of the evaluators or evaluatees. To ensure fairness and accuracy, peer evaluation data can be triangulated with objective records of effort, e.g., using data from Git repositories, teacher observations, or iterative progress reports submitted by the students. In cases where peer evaluation is used in combination with self-evaluation, peer evaluation data can be compared with self-evaluation data to reduce the risk of self-enhancement bias [44, 95].

### 3.4.3 Literature Review: Mapping to the Taxonomy

In this section, we align the findings from the 50 primary studies listed in Table 3.14 with the taxonomy outlined in Subsection 3.4.2. The following subsections present a detailed analysis of the literature, structured according to the dimensions and facets of the proposed taxonomy.

#### 3.4.3.1 Participants

Figure 3.5 gives an overview of the participants ("who evaluates whom"), defining the peer evaluation setups in our primary studies. In our sample of primary studies ($N_s = 50$), seven peer evaluation setups were identified (represented as a–g in the lower part of Figure 3.5). These setups are represented as suffixes (a–g) appended to the study IDs. In total, we observed 75 instances ($N_p = 75$) of these peer evaluation setups in 50 primary studies. The dot matrix in Figure 3.5 shows that some studies use more than one peer evaluation setup, e.g., S7 and S10.

The most common setup for peer evaluation is individuals assessing their team members (40 of 75 instances), indicating a strong emphasis on assessing individual contributions within teams. The second most common setup is self-evaluation i.e., individuals evaluating themselves (16 of 75 instances). It can be noted that all instances of self-evaluations are used in combination with team member evaluations,

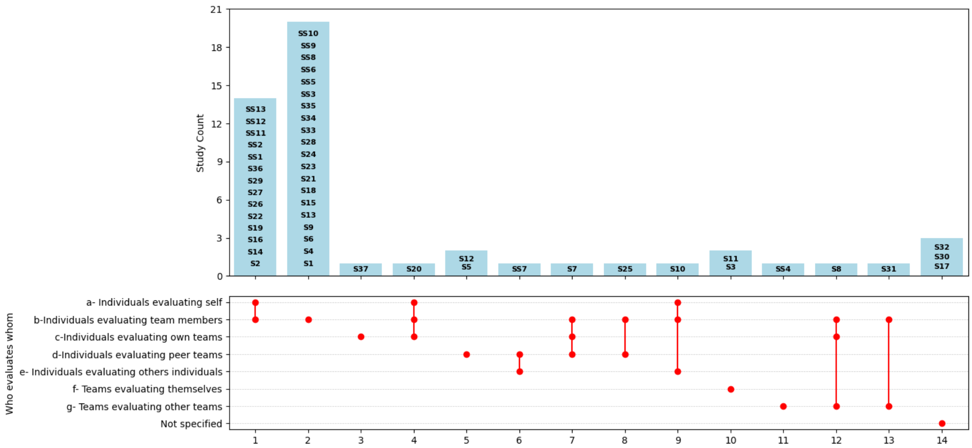



**Figure 3.5:** Distribution of peer evaluation setups ("who evaluates whom") in the primary studies ($N_s = 50$). In total, there are $N_p = 75$ instances of the peer evaluation setups. The bar chart shows the number of studies reporting the combination of peer evaluation setups a–g shown in the dot matrix below.

**Table 3.3:** Peer evaluation setups identified across primary studies ($N_s = 50$). Several studies implemented multiple peer evaluation setups (see the dot matrix in Figure 3.5), which are denoted by suffixes appended to the study ID (a–g). The total instances of peer evaluation setups is $N_p = 75$.

| | Evaluatee | | | | | | |
|---|---|---|---|---|---|---|---|
| **Evaluator** | **Self** | **Team members** | **Own teams** | **Peer teams** | **Others** | **Not specified** | **TOT** |
| Individual | S2a, S10a, S14a, S16a, S19a, S20a, S22a, S26a, S27a, S29a, S36a, SS1a, SS2a, S11a, S12a, S13a | S1, S2b, S4, S6, S7b, S8b, S9, S10b, S13, S14b, S15, S16b, S18, S19b, S20b, S21, S22b, S23, S24, S25b, S26b, S27b, S28, S29b, S31b, S33, S34, S35, S36b, SS1b, SS2b, SS3, SS5, SS6, SS8, SS9, SS10, S11b, S12b, S13b | S7c, S8c, S20c, S37 | S5, S7d, S12, S25d, SS7d | S10e, SS7e | | 67 |
| Teams | S3, S11 | | | S8g, S31g, SS4 | | | 5 |
| Not specified | | | | | | S17, S30, S32 | 3 |
| TOT | 18 | 40 | 4 | 8 | 2 | 3 | 75 |

indicated by the line connecting a and b.

Other peer evaluation setups include individuals who evaluate their own teams (4 of 75 instances), individuals evaluating peer teams (5 of 75 instances), teams evaluating themselves (2 of 75 instances), and teams evaluating peer teams (8 of 75 instances). There are only two instances (S10e and SS7e), where individuals evaluate artifacts produced by individuals outside their teams. In total there are 59 instances where individuals are evaluators and five instances where teams are evaluators. For three studies (S17, S30, and S32), participants are not specified. Of the 50 studies, 26 use only one peer evaluation setup, indicated by a single dot in the lower part of Figure 3.5. The remaining studies employ up to three setups, e.g., S7 and S8. See Table 3.3 for a detailed compilation of peer evaluation setups in our primary studies.

### 3.4.3.2 Context

**Table 3.4:** Peer evaluation contexts (motivations and focus areas) in the primary studies ($N_s = 50$, $N_p = 75$).

| | Focus area in terms of contribution | | | | |
|---|---|---|---|---|---|
| **Motivation** | Overall | Soft | Technical | Not specified | **TOT** |
| Assess individual contributions within team | S1, S2a, S2b, S6, S16a, S16b, S18, S20a, S20b, S20c, S27a, S27b, S35 | S4, S7b, S25b, S26a, S26b, SS1a, SS1b, SS3, SS5, SS6, SS8, SS9, SS10, SS11a, SS11b | S4, S7c, S7d, S25b, S26a, S26b, SS1a, SS1b, SS3, SS6, SS8, SS9, SS10, SS11a, SS11b, SS13a, SS13b | S14a, S14b, S19a, S19b, S21, S32, S33 | 52 |
| Enhance student learning | S16a, S16b, S18, S29a, S29b | S4, S13, S34, SS1a, SS1b, SS5, SS12a, SS12b | S4, S8g, S25d, SS1a, SS1b, SS4, SS12a, SS12b | S30, S37, SS7d, SS7e | 25 |
| Get insights into team dynamics | S4, S16a, S16b, S20a, S20b, S20c, S27a, S27b | S25b, SS2a, SS2b, SS5, SS6, SS10, SS11a, SS11b | S25b, SS10, SS11a, SS11b | S8b, S14a, S14b | 23 |
| Keep students motivated | | | S10a, S10b, S10e | | 3 |
| Enhance students' collaborative skills | S18 | | S8g | | 2 |
| Assess individual and team performance in risk identification | | | S3 | | 1 |
| Not specified | S7b, S23, S24, S31b | S11, S36a, S36b | S5, S11, S12, S31g, S17 | S8c, S9, S15, S22a, S22b, S28 | 18 |
| **TOT** | 31 | 34 | 39 | 20 | 124* |

*The total is larger than $N_p$, as there are multiple motivations and/or focus areas for some instances.

Table 3.4 gives an overview of the motivations and foci for peer evaluations setups in our study. Some setup IDs appear in more than one motivation category and/or more than one contribution focus column, as these setups have multiple motivations or foci for peer evaluation.

Assessing individual contributions is the most common motivation for using peer evaluation (52 of 124 instances). The second most common motivation is to enhance student learning (in terms of reflection and critical thinking – 25 of 124 instances), followed by gaining insights into team dynamics (23 of 124 instances).

Regarding the foci of peer evaluation, 31 of 75 instances target overall contributions, 34 target soft skills, and 39 target technical contributions. In 14 peer evaluation instances, motivation is mentioned, but the focus of evaluation remains unspecified. Similarly, 12 peer evaluation instances specify the evaluation focus but do not indicate a motivation. In six instances, neither the motivation nor the focus is explicitly stated.

When categorizing peer evaluation setups by motivation and evaluation focus, assessing individual contributions emerges as the most common motivation, appearing across all focus areas. Enhancing student learning also shows a balanced distribution between overall, soft, and technical contributions. Studies motivated by gaining insights into team dynamics primarily focus on overall or soft contributions. Less frequent motivations include enhancing collaborative skills and maintaining student motivation, which appear only in a few studies. One study S3, has a study-specific focus and motivation for peer evaluation, which aligns with the goals of the study (i.e, individual and team performance in risk identification).

#### 3.4.3.3 Mechanism

**Table 3.5:** Peer evaluation mechanisms (frequency, methods used, and evaluation criteria) in the primary studies ($N_s = 50$, $N_p = 75$).

| | | Evaluation Criteria | | | |
|---|---|---|---|---|---|
| **Frequency** | **Method** | **Frameworks** | **Rubrics** | **Not specified** | **TOT** |
| Once | Survey | | S6, S7c, S7d, S12, S25b, S27a | S22a, S22b, SS4 | 9 |
| | Live feedback | | | S5 | 1 |
| | Configurable PE tool | | SS12a, SS12b | | 2 |
| | Not specified | | | S19a, S19b, S27b, S30, S31b, S31g, S35 | 7 |
| Twice | Survey | | S3, S11, S13 | S36a, S36b | 5 |
| | Live feedback | | | S8g | 1 |
| | Not specified | | | S31, S37 | 2 |
| Periodic | Survey | SS3 | S2a, S2b, S4, S8c, S18, S20a, S20b, S20c, S25d, SS1a, SS1b, SS8 | SS6 | 14 |
| | Live feedback | | | SS7d | 1 |
| | Custom built tool | | SS9, SS10 | S10a, S10b, S10e | 5 |
| | Written feedback (LMS) | | SS7e | | 1 |
| | Configurable PE tool | | | S13a, S13b | 2 |
| | Not specified | | SS11a, SS11b | S1, S8b, S9, S15, S34, SS5 | 8 |
| Not specified | Survey | | S26 | | 1 |
| | Custom built tool | | | S17 | 1 |
| | Not specified | | | S7b, S14a, S14b, S16a, S16b, S21, S23, S24, S28, S29a, S29b, S32, S33, SS2a, SS2b | 15 |
| TOT | | 1 | 29 | 45 | 75 |

The mechanisms used for peer evaluation across the 75 peer evaluation instances varied in frequency, method, and evaluation criteria (Table 3.5). Three patterns of evaluation frequency emerged: evaluations conducted i) once, ii) twice, iii) periodically, or no frequency was specified.

Periodically conducted peer evaluations are the most common (31 of 75 instances), followed by one-time evaluations (17 instances) and evaluations conducted twice (8 of 75 instances). The remaining setups (19 of 75 instances) do not specify the frequency. When peer evaluation was conducted only once during a course, it most commonly took place at the end of the course. Only one study (SS4) reported conducting it mid-course. In cases where peer evaluation was conducted twice, it typically occurred mid-course and at the end of the course. Periodic peer evaluations showed a lot of variation, conducted weekly, biweekly, at the end of each sprint, or twice per semester in courses longer than one semester.

With respect to methods used to conduct peer evaluations, surveys are the most commonly used method (29 of 75 instances). A smaller but notable proportion of setups use custom-built tools (6 of 75 instances), live feedback sessions (3 of 75 instances), configurable peer evaluation tools (i.e. TeamMates2 and ITPMetrics) were used in two instances, whereas written feedback embedded in learning management systems was employed in one instance. In several cases, the method is not explicitly specified (32 of 75 instances).

Regarding evaluation criteria, 45 of 75 instances do not specify any formal criteria for peer evaluation. Rubric-based evaluation appears in 29 of 75 instances, making it the most common approach. Whereas one of the setups (SS3) utilizes an existing framework (CATME [106]) for peer evaluation.

Rubrics mentioned in the reviewed literature are either developed by the teaching team, or adapted from previous works. S13 and SS8, for example mention using the rubrics proposed by Silvestre et al. [107]. Similarly, SS11 mentions using the evaluation criteria developed by Winger-Haunty [108].

More details on specific rubrics used in several studies are provided in Table 3.6. The peer evaluation instances (32 of 75) that mention the use of rubrics, include rubrics on contributions (13 of 75 instances), technical contributions (10 of 75 instances), and rubrics related to communication (7 of 75 instances).

#### *3.4.3.4 Outcome*

The outcomes of peer evaluation include the format of the peer evaluation (quantitative, qualitative, or both), whether it was shared with the evaluatee (anonymised or unanonymised), its use in grade setting, and whether it was triangulated through additional sources.

Table 3.7 summarizes the formats of peer evaluation and the ways in which this evaluation is shared in the 75 peer evaluation instances. Peer evaluation is either quantitative, qualitative, or a combination of both. Quantitative evaluation (10 instances) is slightly more common than qualitative one (8 instances). The largest group, however, consists of studies using both quantitative and qualitative evaluation (19 instances). This combined format often appears in studies where numerical feedback is used for ratings or ranking, while open-text comments are included to justify ratings or provide richer peer insights. A significant portion of the setups (36 of 75 instances) do not specify the evaluation format at all.

Regarding how peer evaluation is shared, 9 of 75 instances explicitly mention

**Table 3.6:** Evaluation criteria for the 20 (of 75) instances that provide detailed rubrics.

| Evaluation theme | Evaluation sub-theme | Evaluation Criteria (Study identifier in parenthesis) | TOT |
|---|---|---|---|
| Team contribution | Attendance | How much was each team member present? (S26b); Always appear for groupwork (S18); List all team meetings you attended / missed (S20a); On time for meetings (S25b) | 4 |
| | Contributions to project work | Completes an equal share of work (S18); Specific contribution to the work project (S2b, S20b); How much did each team member contribute to the deliverables? (S26b), productivity (SS10), level of contribution to the team's work (SS12) | 6 |
| | Comparing contributions | Who on your team deserves recognition this sprint? (S20b); What letter grade does your team deserve? (S20c); What letter grade do you deserve? (S20a); I think this student has performed better/as/worse than average (S6); I cannot estimate the performance of this student (S6) | 4 |
| | Time spent | Number of hours spent on the project (S2a, S2b); How many days per week did you work on this project? (S26a); How many hours did you spend on the entire project? (S26a) | 3 |
| | Participation | How actively did each team member participate? (S26b); How eager was each team member to participate? (S26b); I worked with my partner (S4); Describe your biggest personal contribution this sprint (S20a), performance during meetings (SS10) | 4 |
| Technical contributions | Quality of work / deliverables | Produces high quality work (S18, SS10); Quality of work produced by team (deliverables) (S5, S7d, S11), Completes assigned tasks in a satisfactory manner, the quality of his/her work contributes to project success SS9) | 6 |
| | Implementation, testing, tools | Contributions to implementation (S4, SS1a, SS1b); Contributions to testing (S4, SS1a, SS1b); Contributed good tools and technology to the project (S25b) | 4 |
| Communication | Peer & client communication | How was the team member's communication? (S4, SS1b); Maintained good communication with clients and users (S13, SS8); Evaluation about the effectiveness of the team's communication (S11); Shows a communicative attitude with peers (S13, SS8), communicates well with the rest of the team (SS9), effective and frequent communication, information sharing, seeking and incorporating teammates' feedback, seeking teammates' input before taking action (SS12). | 7 |
| Professionalism | Respect, adaptability & attitude | Very positive and pleasant to work with (S18); Able to admit mistakes and accept criticism (S13, SS8); Demonstrates interest in improving performance (S13, SS8); Respects differences of opinions (S18), admits mistakes and accepts criticism (SS9), respectfulness, ability to handle criticism (SS10) | 6 |
| Initiative | Initiative & leadership | Takes initiative in solving problems (S18); Demonstrates initiative to achieve the project success (S13, SS8); Eager to plan and execute tasks (S18), shows initiative towards ensuring project success (SS9), proactiveness (SS10), planning and organizing the work, monitoring progress, anticipating and addressing blocking issues (SS12) | 6 |
| Reflection & learning | Retrospection (evaluating own team) | What has enhanced and/or handicapped their team's performance (S2a, S2b); what went well / What did not go well / Any other comments (S4); Describe the top two achievements by your team this sprint (S20c), self-reflection (SS12) | 5 |
| Team culture | Group behaviour & team dynamics | Contributions to team culture (S4); Group dynamics within their team (S7c, S11); How did group behaviour influence perceived meaningfulness? (S26b) | 4 |
| Collaboration | Mutual help | Assumes the project as a team effort, providing support to peers (S13); Is willing to work with others for the success of the group (S18); Do team members cooperate to accomplish the goals? (S11), offers support in tasks that go beyond his/her assigned role (SS9) | 4 |
| Accountability | Responsibility & norms | Do team members hold each other accountable for timelines, commitments and results? (S11); Respects and follows group norms (S25b); Meets deadlines and responsibilities (S18), Timeliness and accuracy of submitted work (SS12) | 4 |
| Coordination | Coordination & deadlines | Effectively coordinates task (S18); Meets deadlines and responsibilities (S18); Routinely monitors the effectiveness of the group (S18); Has the team defined goals/timelines? (S11), Effectively coordinates his/her work with the team's work (SS9) | 3 |
| Expertise and skills | Role maturity | Each team member has acted as a software development expertise; How mature was each team member in their expert role? (S26b), Demonstrating needed capabilities (SS12) | 2 |
| Project management | Project planning & risks (evaluating own team) | Is the project likely to be a success? (S11); Can the team complete the project within schedule? (S11); What are the risks or difficulties? (S11); Do we have the right people for the project? (S11) | 1 |
| Presentation | Presentation quality (evaluating other teams) | Group presentation; Presentation organization; Explanation thoroughness; Preparedness; Quality of presentation material; Presenter formality/appearance; Clear summary (S25d) | 1 |

**Table 3.7:** Peer evaluation formats and sharing ($N_s = 50$, $N_p = 75$).

| | Feedback sharing | | | | | |
|---|---|---|---|---|---|---|
| | Shared | | | | | |
| **Evaluation format** | Anonymised | Unanonymised | Anonimity not clear | Not shared | Not specified | TOT |
| Quantitative | S34 | | | | S3, S6, S7b, S18, S20, S26a, S26b, SS3, SS8, SS13a, SS13b | 12 |
| Qualitative | | S5, S8g | SS4 | | S11, S20a, S20b, S20c, S21 | 8 |
| Both quantitative and qualitative | S4 (anonymised for evaluator), S8c, SS9, SS10 | SS7d, SS7e | S25d | SS1a, SS1b, SS6 | S2a, S2b, S7c, S7d, S13, S25b, S27a, SS12a, SS12b | 19 |
| Not specified | S22a, S22b | | | S8b, S28, S31b, S33, S35, SS5 | S1, S9, S10a, S10b, S10e, S12, S14a, S14b, S15, S16a, S16b, S17, S19a, S19b, S23, S24, S27b, S29a, S29b, S30, S31g, S32, S36a, S36b, S37, SS2, SS11a, SS11b | 36 |
| TOT | 7 | 4 | 2 | 9 | 53 | 75 |

**Table 3.8:** Peer evaluation triangulation and its use in grading in our primary studies ($N_s = 50$, $N_p = 75$).

| | Peer feedback used in grading | | | |
|---|---|---|---|---|
| **Triangulation** | Yes | No | Not specified | TOT |
| Objective record of effort | S18, S28, SS13b | SS6, SS13a | SS3, SS5 | 7 |
| Teachers evaluations | S2a, S2b, S4, S28 (for evaluator), SS7d, SS7e, SS13b | S6, SS6, SS13a | S13, S25b, SS3, SS11a, SS11b | 15 |
| Progress/reflection reports | SS7d, SS7e, SS10 | | S25b | 4 |
| Student grades | | | S3 | 1 |
| Self vs peer feedback | | | S26a, S26b, SS3 | 3 |
| Not specified | S7b, S7c, S7d, S8c, S9, S10e, S12, S15, S16a, S16b, S19a, S19b, S21, S22a, S22b, S23, S25d, S34, SS2a, SS2b, SS8 | | S1, S5, S8b, S8g, S10a, S10b, S11, S14a, S14b, S17, S20a, S20b, S20c, S24, S27a, S27b, S29a, S29b, S30, S31b, S31g, S32, S33, S35, S36a, S36b, S37, SS1a, SS1b, SS4, SS9, SS12a, SS12b | 54 |
| TOT | 34 | 5 | 45 | 84* |

*The total is larger than $N_p$, as some instances have multiple triangulations.

not sharing the peer evaluation results with evaluatees. In majority of instances (53 of 75 instances), sharing of the peer evaluation data is not specified. Anonymised sharing occurs in seven instances, unanonymised sharing is reported in 4 instances, and in two instances feedback sharing is mentioned but whether it was anonymised is not mentioned.

Overall, the table presents diversity in how peer evaluation is formatted and shared. While mixed-format feedback (both quantitative and qualitative) is the most common approach, most studies do not clearly specify how peer evaluation results are shared. This suggests that details of sharing peer evaluation remain inconsistently reported in the literature.

Table 3.8 summarizes how peer evaluation results are validated in our studies and which triangulation sources are used for such a validation. The table indicates that validation is not discussed in the majority of the reported peer evaluation instances (54 of 84). For the remaining 30 instances, the teachers observation is the main source for triangulation (15 instances), followed by objective records of effort (5 instances). In some instances, multiple triangulation sources are used (e.g., S28 and SS3).

In the 34 of 84 instances where the peer evaluation results are used for grading, triangulation is discussed in only 13 cases. Use of peer evaluation results in grading without employing any form of triangulation may introduce potential bias in grading. In 45 of 84 peer evaluation instances, neither triangulation nor grading was mentioned.

#### 3.4.3.5 Other Considerations

Apart from the taxonomy dimensions discussed above, some studies have also considered other aspects related to peer evaluation. S6, S16 and S26 suggest that peer evaluations are susceptible to self-enhancement and unfair ratings. In S6, the authors argue that teachers should always set final grades, as friendships can influence student grades. Strategies to address these concerns include triangulation and teacher oversight, as suggested in SS3, and awareness training, as suggested in S18. S4 suggests that a set of rubrics and guidelines improve consistency and reduce subjectivity.

S18 highlights other aspects of peer evaluation, including validity and reliability, which can be affected by students' experience or confidence, clear assessment criteria to ensure fairness, level of formality (i.e., teacher involvement and grade impact), and student concerns about bias or discrimination, which can affect participation and trust. The cultural aspects of peer evaluation have also been discussed in S18, where peer evaluation was found to be unsuccessful in evaluating student group work in Spain. In contrast, a similar strategy worked well for students in the United Kingdom. The study emphasizes the need for cultural adaptation to ensure inclusiveness and fairness.

Finally, S6 suggests peer evaluation may feel like an overhead to project work, often diminishing students' motivation to provide quality feedback.

### 3.4.4 Interview Findings

We conducted semi-structured interviews with seven SE educators (T1–T7 in Table 3.9) who had used peer evaluation in their SE team project courses. The first author interviewed these educators during an international conference on computing education. The interviewees represented various institutions from different continents around the world. The demographically diverse sample allowed us to explore peer evaluation practices from various perspectives.

**Table 3.9:** Demographic characteristics of interview participants (Exp = Teaching experience in years).

| # | Country | Exp | Experience in team project courses |
|---|---|---|---|
| T1 | New Zealand | 19 | Implemented peer evaluation in seven iterations of a SE team project course while serving as course coordinator and examiner. |
| T2 | Australia | 6 | Implemented peer evaluation in six iterations of a SE team project course while serving as course coordinator and examiner. |
| T3 | USA | 15 | Implemented peer evaluation in six iterations of a SE team project course while serving as course coordinator and examiner. |
| T4 | USA | 3 | Implemented peer evaluation in one iteration of a SE team project course while serving as course coordinator and examiner. |
| T5 | UK | 25 | Applied peer evaluation in multiple SE team project courses while serving as course coordinator and examiner. |
| T6 | Finland | 20 | Applied peer evaluation in 15 instances of a SE team project course while serving course coordinator and examiner |
| T7 | Sweden | 37 | Applied peer evaluation in multiple SE team project courses while serving as course coordinator and examiner. |

### 3.4.4.1 Characterization of peer evaluation practices by educators

In the first part of the interviews, educators were asked to describe the peer evaluation practices in their courses using the proposed taxonomy. It is worth noting that none of the interviewees had previously encountered or utilized a taxonomy for peer evaluation practices. The course contexts are summarized in Table 3.10, and the detailed characterizations of peer evaluation practices in these courses is presented in Table 3.17.

**Table 3.10:** Context of courses characterized using the taxonomy (UG= Undergraduate, G= Graduate).

| Course (Teacher) | Level | Students | Team size | Duration |
|---|---|---|---|---|
| C1 (T1) | UG (final year) | 120–250 | 5–6 | 1 semester |
| C2 (T2) | G (final year) | ≈ 600 | 5–6 | 1 semester |
| C3 (T3) | UG (year 3/4) | 24–27 | 3–5 | 1 semester |
| C4 (T4) | UG (final year) | 45–50 | 4 | 1 semester |
| C5 (T5) | G (final year) | ≈ 60 | 5–6 | 1 semester |
| C6 (T6) | G (final year) | 60–90 | 5–6 | 1 semester |
| C7 (T7) | G (first year) | 25–50 | 5–6 | 1 semester |

Overall, the educators did not report difficulties in characterizing their peer evaluation processes, indicating that the taxonomy is clear, practical, and effective for capturing real-world practices. Their resulting classifications also suggest that the taxonomy extends well beyond the contexts represented in the published literature we reviewed for taxonomy development.

All courses were one semester long and used small software development teams of 3–6 students (see Table 3.17). This is consistent with the common setup observed in the literature. Software development in all of these courses is project-based, iterative, and single delivery (software product to be delivered towards the end of the course). Motivations for peer evaluation align with our earlier findings, i.e., most courses used peer evaluation to assess individual contributions and to understand team dynamics, and sometimes for other goals such as promoting collaboration or supporting student learning. Educators frequently employed rubrics, both quantitative-qualitative feedback, and some form of triangulation (e.g., teacher evaluations, artifacts, or monitoring data) to validate peer assessments.

Despite these general consistencies, there are certain deviations that provide meaningful insights. In one course, the team collectively evaluated the performance of each team member. This approach is not identified in our literature review but the taxonomy was able to accommodate it. We also found that, although most courses used peer evaluation for grading the evaluatee, two courses graded the evaluator as well. In one course, only the evaluator was graded and in another course peer evaluation was used to set the grades for both the evaluator and the evaluatee This practice was not so common in the literature.

Evaluation methods in the characterized peer evaluation processes were also diverse. These included surveys (two courses), configurable tools (one course), live feedback (one course), and Excel sheets (two courses). Using a set of rubrics as evaluation criteria is more prevalent and reported in six of seven courses (see Table 3.17 for details). Additionally, peer evaluation triangulation was common across all courses, using teachers observations, artifacts, or monitoring mechanisms to triangulate the peer evaluation results from students.

The interview data suggests that peer evaluation results were usually not shared (four courses), and were only used by the teaching team. These results were shared anonymously in two courses. Open, unanoymised feedback appeared only in courses instance where the team collectively evaluated each team member. Overall, this points to a predominantly summative use of peer evaluation rather than as a formative tool for students, since feedback is rarely shared. This also suggests an absence of structured follow-up activities based on peer evaluation results (e.g, mentoring, re-evaluation, or conflict resolution). The interviewees confirmed that follow-up was rarely formalized, though some educators used results to start informal conversations.

The characterization process demonstrates that educators can use the taxonomy to classify a diverse set of peer-evaluation setups, and that the taxonomy successfully captures both common and less common practices. It also helped reveal variations (e.g. teams evaluating individual team members), that extend beyond what was found in published studies, highlighting the value of applying the taxonomy in real educational contexts.

### 3.4.4.2 Educators' perceptions on the quality of the taxonomy

**Table 3.11:** Improvement suggestions on different quality aspects of the peer evaluation taxonomy.

| Quality aspect | Suggestion | Reference | Corresponding changes |
|---|---|---|---|
| Completeness | Need to add "timing" explicitly as a distinct facet | T3, T6, T7 | "Timing" facet added to the "Mechanism" dimension |
| | Course context needs to be included | T2, T7 | Course context dimension has been added to the taxonomy with its respective facets. |
| | Missing how evaluation workload is distributed when evaluating as a team | T6 | We add it as a guideline when having teams as evaluators |
| | Taxonomy should include a "why" (rationale) for deciding among different values for each facet of the taxonomy | T1, T7 | Included this rationale in the guidelines for using the taxonomy (see Table 3.13. |
| Conciseness | No suggestion for improvement | | N/A |
| Clarity | Confusion over "feedback triangulation" term, suggest rename | T4, T6 | The facet has been renamed to "Output validation" |
| | "Context" dimension is confusing with course context, suggestion to rename to "Purpose" | T1, T4, T7 | The context dimension has been renamed to "Purpose". |
| | Need for more detailed guidance/support for users | T2, T4, T5, T7 | Detailed guidelines have been provided along with the taxonomy (see Table 3.13. |
| Orthogonality | No suggestion for improvement | | N/A |
| Conceptual coherence | Some facets in the "Outcome" dimension, e.g. feedback sharing and triangulation seem more like subsequent use of feedback | T1, T4 | Feedback format is moved to "mechanism" and renamed as Evaluation format. "Outcome" has been named as "Use of Output" to include respective facets on the use of peer evaluation output. The word feedback also seemed confusing so it is replaced with "output" in all facets in this dimension. |
| Applicability | Uncertainty if taxonomy directly helps with evaluation design unless supported with guidelines | T7 | Detailed guidelines have been provided for educators who are developing peer evaluation practices for the first time. |
| Adaptability | Need for more flexibility in values | T6, T7 | "Other" has been added to the set of values to make the taxonomy more flexible to accommodate more values. |
| Innovation | No suggestion for improvement | | N/A |
| Usability | Need for more walkthrough-style or example-based guidance | T2, T4, T5, T6 | Detailed guidelines have been provided along with the taxonomy (see Table 3.13. |
| | The taxonomy should include minimal documentation to avoid overloading the users | T5 | To keep the documentation minimal, the guidelines have been complemented with a walkthrough through style table. |

After characterizing their courses using the proposed taxonomy, the interviewees were asked to provide feedback on the quality of the taxonomy. The questions for this part of the interview were inspired by the taxonomy quality metrics proposed by Kaplan et al. [105] and included, among others, questions about the completeness, clarity, and orthogonality of the taxonomy. The full set of quality aspects and corresponding interview questions can be found in Table 3.16 in Appendix B.

All interviewees appreciated the taxonomy and were able to use it to characterize their peer evaluation processes. The interviewees provided several suggestions to improve the proposed taxonomy, which are summarized in Table 3.11. The suggestions were discussed by the authors and helped us further improve the taxonomy (see column "Corresponding changes in Table 3.11.

The revised taxonomy is presented in Appendix B (see Figure 3.6). An updated mapping of the literature, aligned with the revised taxonomy, is available in an electronic supplement at Zenodo.[2].

[2] Data Extraction form and mapping of literature to the revised taxonomy

## 3.5 Discussion

In this section, we report the summary of the main findings from the literature interviews. In addition, we also share implications for educators and researchers for SE team project courses.

### 3.5.1 Summary of Findings

Table 3.12 gives a side-by-side overview of our main findings from the literature review and interviews with educators.

**Table 3.12:** Summary of the main findings from the literature and the interviews. PE=peer evaluation.

| **Taxonomy dimension** | **Findings from the literature** ($N_s = 50$, $N_p = 75$) | **Findings from the interviews** ($N = 7$) |
|---|---|---|
| Course context | Course level and team size are not discussed as aspects influencing the design of PE interventions. PE is mostly used in undergraduate courses with small or medium-sized teams. | The roles of course level, team size, and course duration in designing PE interventions are emphasized. PE is mostly used with small teams in undergraduate and graduate courses. Small teams are said to allow for more detailed PEs, whereas larger teams require careful scaling. |
| Purpose | The three most commonly reported motivations are (1) assessing individual contributions (most common), (2) enhancing student learning, and (3) understanding team dynamics. | Assessing individual contributions was the most common motivation. Team dynamics was mentioned as a secondary goal by some interviewees. |
| | The focus of the PEs is mostly on technical contributions 34 instances), soft contributions (32 instances) and overall contributions (27 instances) respectively. Only 12 instances focused on both technical and soft skills. | The focus of the PEs was a combination of both technical and soft contributions. |
| Participants | The two most common PE setups for PEs are individuals evaluating their team members and self-evaluation. In eight PE instances, peer teams were evaluated by individuals or teams. | The most commonly reported PE setup was individuals evaluating team members and self-evaluation. Two of seven interviewees reported using teams as evaluators. |
| Mechanism | The most common frequencies for PEs are periodic, followed by once. However, for a quarter of the PE instances, the frequency was not reported. | The most common frequencies for PEs were periodic and once (a tie between two values). All interviewees reported the frequency of their PE setup(s). |
| | Survey-based tools with rubrics are used most frequently, but only a few of them provide details about the evaluation criteria used (given in Table 3.6) | A variety of tools were used, including configurable digital tools, spreadsheets, surveys, and live feedback. |
| | The rubrics suggested that some soft aspects, such as initiative and coordination, were not frequently assessed. | Rubrics were used in almost all courses, covering both technical and soft contributions. |
| | The format of the PEs varied (qualitative, quantitative, as well as both). However, for about half of the PE instances, the evaluation format was not reported. | The format is typically qualitative or a combination of both qualitative and quantitative. In none of the courses was it quantitative alone. |
| Use of PE output | For the majority of PE instances (53 of 84), the sharing of PE results was not reported. Of the 11 PE instances that shared PE results, four did so anonymously. Limited reporting on whether the PE outcomes were shared with the evaluatees points to a potential underutilization of peer evaluation as a formative feedback tool. | PE results were usually not shared with students (four courses). When shared, they were mostly anonymous. |
| | There are 34 of 84 peer evaluation instances where peer evaluation results are used in grading. However, only 13 of these 34 instances report triangulation of this data with other data resources. Use of peer evaluation outcomes to set student grades without reporting triangulation mechanisms raises concern on fairnes of grading | PE results were used for grading in five of seven courses. Triangulation of PE results was mentioned in most cases. Different, and sometimes multiple, sources were used for it (artifacts, effort logs, monitoring, teacher observations). |

The table shows that interviewees generally provided more comprehensive descriptions of their peer-evaluation practices and the contextual factors that shape them. Since we explicitly asked interviewees to provide information according to our taxonomy (see Table 3.15), this was expected and is an indication of the usefulness of our taxonomy in reporting peer-evaluation practices.

Due to a lack of reporting in the literature, it is difficult to compare the findings from the literature and those from the interviews. However, we can still see several similarities and differences between the sources. Both sources lack information on peer evaluation practices for large teams and predominantly report peer evaluation setups, where individuals evaluate their team members and/or themselves, periodically and/or once.

Differences between the findings from the literature and those from the interviews are pronounced in the mechanisms for peer evaluation and the use of the peer evaluation results in grading. The use of rubrics and evaluation of both technical and non-technical aspects is reported to a higher degree in the interviews. The interviews also report the use of peer evaluation results in grading to a much higher degree, together with different and sometimes multiple sources for triangulation.

A very few studies specified sharing of peer evaluation results with evaluatees. Whereas interview findings suggested limited sharing of peer evaluation results with the evaluatees, suggesting a more summative use of peer evaluation than formative.

### 3.5.2 Implications for Educators and Researchers

The taxonomy provides a structured framework for educators to design peer evaluations and for researchers to report them consistently. The following guidelines (see Table 3.13) are based on the revised taxonomy and the reflections shared by the educators during the interviews.

**Consider the course context** Since there is no one-size-fits-all peer evaluation setup and practice, it is important to consider the course context when designing peer evaluations (T2, T5). The effort required by students to participate in peer evaluation surveys increases with the size of the team (T5, [21]). Therefore, lengthy peer evaluations would not be feasible for larger teams. Additionally, the duration of the course should be taken into account when determining the frequency of peer evaluations. For shorter-duration courses, it may not be feasible to conduct peer evaluations multiple times during the course.

**Define the purpose of using peer evaluation:** The purpose of using the peer evaluation is important to articulate, as it would not only clarify the reasons for conducting the peer evaluation, but it would also facilitate deciding the other taxonomy dimensions [21]. The purpose can be clarified by selecting the appropriate values for its two facets: Motivation and Focus. For example, if the motivation is to assess individual contributions, team members should be evaluated individually, not as a team (T2). Likewise, if the motivation is to enhance learning, it would be appropriate to share the feedback with the team so that they could reflect on and improve their performance (T6).

**Table 3.13:** Guidelines for designing peer evaluation practices: A walk-through based on the revised taxonomy.

| Dimension | Facet | Example values* | Reason(s) to select this value | Constraints |
|---|---|---|---|---|
| Course-Context | Course-level | Course level is already decided for a given course context | | Student maturity may influence how students perceive peer evaluations |
| | Duration | Course duration is already decided for a given course context | | Longer courses may imply more peer evaluation cycles and more administrative costs. |
| | Team size | Team size is already decided for a given course context | | Team size affects granularity of contributions, workload, and complexity of evaluation. |
| Purpose | Motivation | Assess individual contributions | Promotes accountability, discourages free-riding, can inform fair grading. | Risk of biased or strategic ratings (friendship or conflict-driven), can create tension in teams if not managed ethically. |
| | | Enhance learning | Encourages reflection on teamwork and personal performance, builds critical evaluation and feedback-giving skills | Without guidance or training, peer feedback may be superficial or unhelpful. Students may see it as extra work unless linked clearly to course learning outcomes. |
| | | Understand team dynamics | Can identify emerging conflicts, workload imbalances, or communication gaps. | Risk of focusing more on relationships than actual contributions, more tensions among team members if raised issues are not managed carefully. |
| | Focus | Overall contributions | Simple for students to evaluate. Captures a general sense of fairness and contributions | Too vague. Doesn't reveal what kind of contribution was made. |
| | | Soft contributions | Highlights teamwork skills often overlooked in technical projects. Encourages students to reflect on interpersonal skills. | Non-tangible contributions, more prone to interpersonal biases or popularity. |
| | | Technical contributions | Useful when workload distribution matters. Can identify hidden free-riding or skill gaps. | Students may not fully know the technical details of team members' work. Risks undermining soft contributions. |
| Participant | Evaluator | Individuals | Rich feedback, accountability | More workload, interpersonal tensions |
| | | Teams | Simpler, less overhead | Requires distribution of evaluation effort, less detail |
| | Evaluatee | Self | Promotes reflection | Inflated/deflated self-ratings |
| | | Team members | Accountability, captures individual contributions | Personal biases, more work |
| | | Teams | Emphasizes collective outcomes, less data to analyze | Hides unequal participation |
| Mechanism | Evaluation criteria | Frameworks | Validated, comparable across studies | May not fit context |
| | | A set of rubrics | Aligned to context, consistent scoring, easier analysis | Rigid, may miss certain behaviors |
| | | Free form | Rich insights | Inconsistent, harder to analyze |
| | Evaluation format | Quantitative | Easy aggregation/comparison | Lacks in-depth insights |
| | | Qualitative | Rich explanations | Hard to standardize/analyze |
| | | Both | Balanced evidence | Higher effort for both educators and students |
| | Evaluation method | Surveys | Scalable, analyzable | Survey fatigue (specially in longer surveys,) no possibility of clarification. |
| | | Live feedback | Quick, promotes dialogue | Students may avoid criticism to prevent conflict, harder to record and analyze |
| | | Custom built tools | Tailored to context, privacy options | Build/maintenance cost |
| | | Configurable tools | Off-the-shelf solution with no maintenance costs | Needs to be adjusted to context, privacy concerns, subscription fees may imply |
| | Frequency | One-time | Low effort, easy summative use | No detailed insights or progression |
| | | Twice | Balance of insight and effort | Some fatigue for teachers and students |
| | | Periodic | Tracks progress over time, detailed insights on progression | More workload, quality may decline overtime |
| | Timing | Early in course | Issues diagnosed early | Too premature, lack of detailed insights |
| | | Mid- course | More informed, corrective actions possible | Needs careful scheduling |
| | | End of course | Full information available | Recall bias, no chance for correction after feedback |
| Use of output | Output sharing | Shared anonymous | Sharing promotes learning and reflection, anonymity builds trust and protects students | Limited accountability |
| | | Shared not anonymous | Encourages reflection, responsibility and dialogue | May discourage criticism, poses risk of retaliation or biased responses |
| | | Not shared | Informs teachers in grading | Lost learning opportunity for evaluatees. |
| | Output used in grading | Graded | Motivates students to give well thought and careful input | Strategic or biased ratings (based on conflict or friendship) |
| | | Not graded | Encourages honesty, reflection | Risk of low participation or less effort in providing feedback |
| | Output validation | Objective record of effort | Improves fairness, reliability | Requires extra data collection and effort |
| | | Not validated | Simpler, less administrative workload | Risk of being be unfair or inaccurate |

*Note: The set of values provided in this table is based on the literature review and can be further expanded.

**Select participants of peer evaluation:** Decisions around participants (evaluators and evaluatees) have important trade-offs. Selecting individuals or teams either as evaluators or evaluatees have different consequences, including the workload of teachers and students [21], richness of the feedback, and the potential benefits of peer evaluation. For example, when individuals act as evaluators and evaluatees, it generally leads to richer and diverse feedback (T1). However, it may also introduce interpersonal tensions and more workload for students and teachers (T6). Similarly, if self-evaluations are also a part of the peer evaluation process, they can skew the results from students underrating themselves, [21], or potential self-enhancement [44, 95]. In addition to these potential consequences, the purpose of using peer evaluations in the project course should also be considered when selecting participants for peer evaluation.

**Decide mechanism for peer evaluation:** The mechanism supports decision-making regarding the operationalization of peer evaluation practices. First, it is important to define suitable evaluation criteria. Educators can either use existing frameworks (e.g., CATME [106]) or define their own set of rubrics or criteria for their peer evaluation setup [79]. The former approach makes scoring transparent and consistent; however, it often needs to be adapted to fit the course context (T1). The latter approach requires a clear description of rubrics and may necessitate some training or awareness sessions to clarify the evaluation criteria and associated rubrics [24]. In addition to these two approaches, peer evaluation could also consist of open-ended feedback [48]. It can lead to deeper and more meaningful insights [22, 92], but this type of feedback can be more challenging to interpret and analyze. A more balanced approach could involve using a combination of methods. For example, educators might begin with their own set of rubrics tailored to a specific course context and then compare these with existing frameworks to ensure that no important aspects are overlooked. Additionally, offering students the opportunity to provide free-text comments can enhance the evaluation process. However, educators should be mindful to keep peer evaluation surveys or forms concise, including only essential criteria or questions to avoid overwhelming the students.

Choosing an evaluation method involves balancing data quality, scalability, and accessibility. While surveys are convenient, lengthy ones can cause fatigue [21]. Live feedback is quick but may lack honesty due to the absence of anonymity, and can be hard to analyze if not recorded. Custom tools can be tailored to needs but require significant development and maintenance (T1). The selected method should be scalable, ensure consistent data collection, protect privacy, and facilitate quick analysis to provide timely feedback to students and teachers.

Educators also need to consider the evaluation format. Quantitative ratings are efficient for comparison but often lack depth; qualitative feedback (i.e., free-text comments) provides richer perspectives [22, 92] but is more challenging to analyze systematically. Combining both can lead to a more balanced approach, although it may require additional effort from students and teachers.

The frequency and timing of peer evaluations should balance meaningful data collection and workload [21]. Frequent evaluations may lead to fatigue , while infrequent ones can miss important insights. Early evaluations can help guide learning and team dynamics, but might be inaccurate if conducted too soon. Mid-course evaluations allow for adjustments while projects are ongoing, whereas end-of-course evaluations capture overall performance but offer limited opportunities for any corrective action and can also be affected by recall bias.

**Decide how the output of peer evaluations will be used:** An important decision is whether peer evaluation output is shared with students or used solely by teachers (e.g., for grading purposes). The sharing of peer evaluation output decides the summative or formative use of peer evaluation. If the purpose of using peer evaluation in a course was to enhance learning and reflections, the results of peer evaluation should be shared with the students (T1, T5). Sharing feedback anonymously protects students from social pressure, fear of retaliation and encourages honest input (T1). On the down side, it can limit direct discussion and accountability. Non-anonymous evaluations promote transparency and dialogue but may discourage critical comments due to fear of conflict. Another approach can be to provide an option for both public and private comments (only visible to the teacher) [109].

Integrating peer evaluations into the grading can incentivize thoughtful input, but also risks bias or strategic behavior based on conflicts or friendships [21]. Non-graded peer evaluation may encourage honest reflections and emphasize learning over grades, but students may invest less effort in providing detailed evaluations. Validation of peer evaluation results through other sources (e.g., teacher observations or contribution logs) [23, 110] increases reliability and fairness, but it requires extra effort. Such a validation becomes more relevant and necessary when peer evaluation results are used for summative purposes, i.e., an input for deciding students' grades (T1, T2, T4, T6).

**Summary:** The taxonomy and associated guidelines can help SE educators reflect on different trade-offs involved in designing peer evaluation setups for their courses. By carefully considering different dimensions of the taxonomy, educators can design peer evaluation practices that are both pedagogically valuable and ethically sound. Ensuring a positive peer evaluation environment develops a culture of reflection and learning. It is important to encourage constructive feedback and prevent personal attacks. Without these measures, peer evaluation can lose credibility and potentially harm students' learning experiences. Moreover, the proposed taxonomy and guidelines can serve as a resource for SE education researchers to improve the reporting of peer evaluation practices within their studies. For instance, the taxonomy dimensions could help researchers to reflect on the appropriateness of their reporting.

## 3.6 Threats to Validity

This study employs a mixed-method approach, combining a literature review with semi-structured interviews. We adopted the guidelines proposed by Ampatzoglou et al. [111] to systematically identify, discuss, and mitigate threats to validity for both phases of the study.

### 3.6.1 Study/Subject Selection Validity

*Literature Review:* Study selection bias may occur if relevant literature is omitted due to limitations in the search process. We based our review on a recent and systematic literature review (SLR) [5], which included studies based on credibility, rigor, and quality of reporting. To capture newer research, we extended this work with an updated Scopus search covering studies published between 2022 and 2024. While using a single database and the exclusion of gray literature may limit coverage, combining these two steps gives us reasonable confidence that the selected studies offer a representative set of literature.

*Interviews:* Selection bias in interviews may arise if interviewees do not represent the broader SE education community. The participants were recruited from a working group on team project courses at an international computing education conference, of which the first author was also a member. This use of convenience sampling may limit the generalizability of the findings. However, the interviewees represented diverse institutions and countries, providing a range of perspectives that helps mitigate this threat.

During a complementary database search conducted after the interviews (see Phase three in Section 3.3.2 for more details), we identified one publication co-authored by one of the interview participants. This study had not been included in our reviewed set at the time the interviews were conducted. Also, the interviews focused on general peer evaluation practices and on the perceived usefulness and quality aspects of the proposed taxonomy, rather than on evaluating specific publications.

### 3.6.2 Data Collection Validity

*Literature Review:* Threats at this stage include inconsistencies in data extraction and potential interpretation bias. To mitigate these, we designed a structured data extraction form in Excel to capture course context and peer evaluation characteristics systematically. A pilot extraction on ten studies was conducted independently by all authors to ensure consistency and shared understanding. Discrepancies were discussed and resolved collaboratively. There were several iterations of data extraction and refinement to ensure a good quality of the dataset.

*Interviews:* Interview responses may be affected by misunderstandings or leading questions. To mitigate this, we piloted the interview protocol with two educators before the study and refined ambiguous questions. All interviews were recorded and transcribed to minimize data loss.

### 3.6.3 Data Analysis Validity

*Literature Review:* Subjectivity in coding and synthesis poses a threat to data analysis validity. Data were coded using a iterative thematic synthesis, and coding disagreements were discussed until consensus was reached. Triangulation across multiple studies was used to strengthen interpretations.

*Interviews:* Analysis bias may occur if researchers impose their interpretations on interviewee responses. To address this, we employed thematic analysis guided by the proposed taxonomy, with multiple researchers reviewing and validating the codes. Member checking was conducted by sharing summaries with interview participants for confirmation.

### 3.6.4 Conclusion Validity

*Literature Review:* Inconsistent reporting in primary studies posed a threat to conclusion validity, as missing or ambiguous information limited the reliability of the synthesized findings. To mitigate this, we developed the taxonomy iteratively that provides a structured framework for describing peer evaluation practices, enabling more consistent data extraction and comparison across studies.

*Interviews:* Conclusions drawn from interviews may be limited due to the relatively small sample size. While our seven interviewees represented diverse institutions and countries, the findings may not capture the full range of peer evaluation practices globally. We mitigated this by combining interview insights with evidence from the literature review, thereby strengthening the overall credibility of results.

## 3.7 Conclusion

This study addressed the need for a structured framework to organize and analyze peer evaluation practices in software engineering team project courses. We developed a taxonomy of peer evaluation in SE team project courses and used it to characterize the state of the art and practice. The results show that while peer evaluation is widely used in SE team project courses, its implementation remains diverse and often underreported. Peer evaluations are used in SE team project courses primarily to assess and track the contributions of individual team members (e.g., to address free riding). The potential of peer evaluation as a tool for enhancing collaboration, reflection, and learning in SE team project courses is relatively underutilized. Moreover, although triangulation with teacher observations and project artifacts is common in practice (interviews), it is less frequently documented in research, leaving these safeguards against potential bias in peer evaluations largely invisible.

We also developed guidelines based on the proposed taxonomy for designing peer evaluations in SE team project courses. The proposed taxonomy offers researchers a consistent framework for reporting, comparing, and extending peer evaluation studies. Together, these contributions support a more systematic use of peer evaluation in SE team project courses and improve the reporting of related research findings in the field.

# Appendix A: List of reviewed literature

Table 3.14: Papers included in this study ($N_s = 50$). S1–S37 represent studies from the base set ([5]), and SS1–SS13 represent studies from the complementary search.

| Study | Ref | Title | Year | Location | Venue |
|---|---|---|---|---|---|
| S1 | [77] | A case study of classroom experience with client-based team projects | 2008 | USA | Journal of Computing Sciences in Colleges |
| S2 | [78] | A decade review of a masters-level real-world-projects capstone course | 2011 | USA | Proc. Information Systems Educators Conf. (ISECON 2011) |
| S3 | [100] | A look at software engineering risks in a team project course | 2013 | USA | 26th Int. Conf. on Software Engineering Education and Training (CSEET) |
| S4 | [79] | A Three-Year Study on Peer Evaluation in a Software Engineering Project Course | 2022 | New-Zealand | IEEE Transactions on Education |
| S5 | [50] | A Top-Down Approach to Teaching Web Development in the Cloud | 2018 | Japan | IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) |
| S6 | [21] | Assessing large-project courses: Model, activities, and lessons learned | 2012 | Sweden | ACM Transactions on Computing Education |
| S7 | [101] | Benefits and struggles of using large team projects in capstone courses | 2007 | USA | American Society for Engineering Education (ASEE) Annual Conference and Expositon |
| S8 | [45] | Best practices for industry-sponsored CS capstone courses | 2014 | USA | Journal of Computing Sciences in Colleges |
| S9 | [93] | Capstone project: From software engineering to Informatics | 2010 | USA | EE Conference on Software Engineering Education and Training (IEEE - CSEET) |
| S10 | [99] | Competitive and agile software engineering education | 2010 | USA | IEEE South- eastCon 2010 (SoutheastCon) |
| S11 | [96] | Deployment of capstone projects in software engineering education at Duy Tan University | 2013 | Vietnam | IEEE Learning and Teaching in Computing and Engineering |
| S12 | [90] | Emphasizing soft skills and team development in an educational digital game design course | 2009 | USA | Proceedings of the 4th international Conference on Foundations of Digital Games, |
| S13 | [94] | Enhancing the Student Learning Experience in Software Engineering Project Courses | 2018 | Chile | IEEE Transactions on Education |
| S14 | [82] | Evolution of a graduate software engineering capstone course – A course review | 2019 | USA | International Journal of Engineering Education |

| | | | | | |
|---|---|---|---|---|---|
| S15 | [112] | Evolution of capstone courses in software engineering: a finishing school | 2010 | USA | Journal of Computing Sciences in Colleges |
| S16 | [44] | Experience Report on Key Success Factors for Promoting Students' Engagement | 2020 | Germany | IEEE World Conference on Engineering Education (EDUNINE) |
| S17 | [84] | Experiments with Adding to the Experience that Can be Acquired from Software Courses | 2010 | Canada | Seventh International Conference on the Quality of Information and Communications Technology |
| S18 | [24] | Fair assessment in software engineering capstone projects | 2013 | Greece | 6th Balkan Conference in Informatics |
| S19 | [83] | Fostering cooperative learning with Scrum in a semi-capstone course | 2018 | USA | Journal of Information Systems Education |
| S20 | [98] | Hurricanes and pandemics: adapting software engineering courses | 2021 | USA | Journal of Computing Sciences in Colleges |
| S21 | [86] | Industry-emulated projects in the classroom | 2015 | USA | 6th Annual Conference on Infor- mation Technology Education (SIGITE ) |
| S22 | [18] | Industry-oriented project-based learning of software engineering | 2019 | Australia | 24th International conference on engineering of complex computer systems (ICECCS) |
| S23 | [87] | Information systems development course: integrating competencies | 2010 | Portugal | IEEE Transforming Engineering Education |
| S24 | [49] | Maintaining High Process Capability in a Student Project Course | 2007 | Australia | 20th Conference on Software Engineering Education & Training. |
| S25 | [48] | One-semester CS capstone: A 40–60 teaching approach | 2013 | USA | 0th International Conference on Information Technology |
| S26 | [95] | Peer assessment in experiential learning in agile capstone projects | 2013 | Finland | IEEE frontiers in education conference (FIE) |
| S27 | [113] | Practical Software Engineering Capstone Course Framework | 2019 | Finland | Computer Supported Education: 10th International Conference (CSEDU) |
| S28 | [88] | Reflections on 10 years of sponsored senior design projects | 2007 | USA | Journal of Systems and Software |
| S29 | [114] | Simulating industry: innovative software engineering capstone course | 2013 | Australia | IEEE frontiers in education conference (FIE) |
| S30 | [51] | Splat! er, shmup? A post-mortem on a capstone production experience | 2016 | USA | IEEE frontiers in education conference (FIE) |
| S31 | [115] | Student teamwork: capstone course in game programming | 2007 | USA | IEEE frontiers in education conference (FIE) |

| | | | | | |
|---|---|---|---|---|---|
| S32 | [85] | Systems Development Projects | 2008 | Jordan | 3rd International Conference on Information and Communication Technologies: |
| S33 | [116] | Teaching agile methodology in a software engineering capstone course | 2011 | USA | Journal of Computing Sciences in Colleges |
| S34 | [80] | Teaching software engineering with open source development | 2019 | Canada | Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS) |
| S35 | [17] | Effect of Real-World Capstone Project on Soft Skills Acquisition | 2020 | Finland | Proceedings of the 32nd Conferenceon Software Engineering Education and Training (CSEET) |
| S36 | [117] | Using Agile and Active Learning in Software Curriculum | 2021 | USA | Proceedings of the ASEE Virtual Annual Conference Content Access. |
| S37 | [16] | What can students get from a software engineering capstone course? | 2017 | Chile | IEEE/ACM 39th International Conference on software engineering: software engineering Education and Training Track (ICSE-SEET) |
| SS1 | [22] | Think Before You Scrum – Daily Scrums to Aid Reflection | 2024 | New Zealand | 36th International Conference on Software Engineering Education and Training (CSEET) |
| SS2 | [46] | ScrumBoard: Project Management Tool for SE Education | 2024 | New Zealand | 36th International Conference on Software Engineering Education and Training (CSEET) |
| SS3 | [23] | Assessing individual contributions to software engineering projects | 2022 | USA | Computer Science Education |
| SS4 | [97] | Guiding Peer-feedback in Learning Software Design using UML | 2022 | Netherlands | ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Train- ing, |
| SS5 | [91] | Guiding Principles for Assessing Software Engineering Teams | 2024 | Canada | IEEE frontiers in education conference (FIE) |
| SS6 | [92] | Using GitHub Analytics to Assess Collaboration Quality | 2024 | Canada | IEEE frontiers in education conference (FIE) |
| SS7 | [118] | Feedback as a process in a large semi-capstone course | 2023 | Norway | 27th International Conference on Evaluation and Assessment in Software Engineering |
| SS8 | [119] | Impact of Grading on Teamwork Quality in SE Capstone | 2023 | Chile | IEEE Access |
| SS9 | [120] | The Influence of Software Engineering Education on Developing Teamwork Skills in Both Genders | 2025 | Chile | IEEE Access |

| | | | | | |
|---|---|---|---|---|---|
| SS10 | [109] | Project Pulse: Enhancing Peer Evaluation and Team Accountability in Senior Design Projects | 2025 | USA | Proceedings of the International Conference on Software Engineering and Knowledge Engineering |
| SS11 | [121] | Game Studio Simulator: Integrating Industry Practice and Formative Individual Assessment in Collaborative Game Development Capstone Projects | 2025 | Australia | ACM Transactions on Computing Education |
| SS12 | [122] | Understanding Students' Mindset While Reflecting on Their Teamwork Experiences | 2025 | Canada | World Engineering Education Forum - Global Engineering Deans Council (WEEF-GEDC) |
| SS13 | [123] | Enhancing Student Engagement in Large-Scale Capstone Courses: An Experience Report | 2024 | New Zealand | Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1 |

# Appendix B: Interview guide and categorization of example courses

**Table 3.15:** Interview guide part 1: Peer evaluation practices – characterization sheet.

| **Course Context** | | | | |
|---|---|---|---|---|
| Course year and level | | | | |
| Number of students | | | | |
| Team size and number of teams | | | | |
| Is peer evaluation used | | | | |
| **Peer Evaluation Practices** | | | | |
| **Dimension** | **Facet** | **Description** | **Value** | **Comments** |
| Participants | Evaluator | Entities who evaluate other entities (individuals, teams). | | |
| | Evaluatee | An entity that is being evaluated by another entity. | | |
| Context | Motivation | Motivations for carrying out peer evaluations. | | |
| | Focus | Specific skills, behaviours, or competencies targeted for evaluation. | | |
| Mechanism | Frequency | How often peer evaluation is carried out in a course. | | |
| | Method used | The method used to collect data on peer evaluation. | | |
| | Evaluation criteria | Criteria or rubrics used to guide the peer evaluation process. | | |
| Outcome | Form of feedback | Qualitative or quantitative feedback. | | |
| | Feedback sharing | Whether the feedback is shared with the evaluatee. | | |
| | Feedback used in grading | Whether feedback is used to set the course grade (evaluator/evaluatee). | | |
| | Feedback triangulation | Whether the feedback is validated using different sources. | | |

**Table 3.16:** Interview guide part 2: Questions on quality aspects of the taxonomy.

| **Quality aspect** | **Interview question** |
|---|---|
| Completeness | What key aspects of peer evaluation does this taxonomy capture well? Are there any important areas you feel are missing? |
| Conciseness | Can you identify any elements in the taxonomy that seem redundant or unnecessary? Why? |
| Clarity | How clear and understandable is each of the dimensions and facets? Were there any that you found confusing or ambiguous? |
| Orthogonality | Did you find any dimension or facet as too broad, too narrow, or overlapping with others? Can you suggest changes or refinements? |
| Conceptual Coherence | How well do the dimensions and facets within each dimension fit together conceptually? |
| Applicability | Do you think this taxonomy can help in designing and reporting a peer evaluation intervention? |
| Adaptability | How adaptable do you think this taxonomy is across different types of software engineering courses or team-based projects? |
| Innovation | Have you used other peer evaluation taxonomies? How does this one compare in terms of structure or comprehensiveness? |
| Usability | How user-friendly do you find this taxonomy for educators or researchers? What kinds of support or documentation would help make it easier to use? |

**Table 3.17:** Characterization of peer evaluation practices in courses C1–C7 using the proposed taxonomy.

| # | Participants | | Context | | Mechanism | | | Outcome | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Evaluator** | **Evaluatee** | **Motivation** | **Focus** | **Frequency** | **Method Used** | **Evaluation Criteria** | **Feedback format** | **Feedback Sharing** | **Feedback Use in Grading** | **Feedback triangulation** |
| C1 (T1) | Individuals | Self and team members | To assess individual contributions within team, to get insights into team dynamics, to enhance students collaborative skills, to keep students motivated | Both soft and technical contributions | Periodic: four times (weeks 5, 9, 11, and 13) | Third party PE tool (TeaMates) | A set of rubrics designed by the teaching team | Both quantitative and qualitative | Shared anonymised | Graded for both evaluator and evaluatee | Reflection documents, github commits, progress meetings with TAs, regular journals (written weekly, submitted every two weeks). |
| C2 (T2) | Individuals | Team members | To assess individual contributions within team, to get insights into team dynamics | Overall Contributions | Periodic: three sprints with peer evaluation during the last week of each sprint | Survey | Set of rubrics | Both quantitative and qualitative | Not shared | Graded for evaluatee | Teachers observations |
| C3 (T3) | Individuals | Self and team members | To assess individual contributions within team, to get insights into team dynamics (secondary motivation) | Both soft and technical contributions | Once: at the end of the course and the project work | Form in Microsoft word | Set of rubrics | Both quantitative and qualitative: ratings on 1–10 scale and some free form text | Not shared | Not graded (but the final grade depends on peer evaluation) | Peer evaluation is the last option to validate, teachers mainly rely on their observation, & github contributions |
| C4 (T4) | Individuals | Self and team members | To assess individual contributions within team, to enhance student learning, to get insights into team dynamics | Participation, communication, individual contribution in project (technical) | Periodic: at the end of each sprint | Free form written responses | Set of rubrics (prompts for students) | Qualitative | Not shared | Graded for evaluatee (20 points/ peer evaluation, 1000 points for assignments in total) | Teachers observations, progress reports and logs |
| C5 (T5) | Teams | Individuals (team sits collaboratively to decide score for each team member) | To assess individual contributions within team, to get insights into team dynamics, to enhance students collaborative skills | Communication, project management, technical development, research for the project | Once: at the end of the course | Students record peer evaluation results on an excel sheet and submit to teacher. Participation in team session is mandatory. Unsupervised session. | Team signs a team contract which serves as the evaluation criteria for peer evaluation | Both quantitative and qualitative (contribution percentages and free form text) | Shared: everyone in the team does it together | Team gets a grade on the group report then the individual grade is adjusted according to the peer evaluation (overall effect is around 40%) | Individual reports, and monitoring of the groups. |
| C6 (T6) | Teams | Peer Teams (team artifacts, i.e., requirements document, project plan, initial technical documents) | To see if a project is feasible from another team's point of view. | Technical contributions (artifact) | Once: when project plan is in place. | Live feedback (during a class presentation) | Set of rubrics | Qualitative (oral feedback) | Shared unanonymised | Graded for evaluator (1–5 points) | The evaluatee give their feedback on quality of received feedback. Teacher also checks the submitted artifacts to validate peer feedback. |
| C7 (T7) | Individuals | Self and team members | To assess individual contributions within team | Both technical and soft contributions | Twice: mid and end of course | Survey | Set of rubrics | Both Quantitative and Qualitative (List of 10 questions on a Likert-type scale for each of team member, plus 2 extra questions) | Not shared (only used by teacher for corrective actions) | Not graded but informs grade to a small extent | Checked whether the peer evaluation matches evaluations in other documents (e.g, weekly reports) and suggestions for distributing credits between team members. |

# Appendix C: Revised taxonomy


**Figure 3.6:** Revised taxonomy for peer evaluation: Dimensions (Course context, Purpose, Participants, Mechanism, Use of peer evaluation output), facets, and values.

# 4 Using Peer Evaluations and Team Contracts in Software Engineering Team Project Courses

*Nayla Nasir, Muhammad Usman, and Usman Nasir*
*Accepted for publication in: IEEE/ACM 48th International Conference on Software Engineering-Software Engineering Education and Training (ICSE-SEET), 2026.*

## Abstract

In Software Engineering (SE) team project courses, students work in teams to develop a software solution for a customer. Besides technical, working in teams in such team project courses involve several non-technical challenges related to teamwork as well (such as unequal contributions, communication and coordination, team conflicts). Peer evaluation (PE) and team contracts (TC) are well known pedagogical interventions in project-based education, including software engineering (SE). We applied these two interventions in two SE team project courses at our university: a small-team project course (5–7 students per team) and a large-team project course (10–12 students per team, typically organized into subteams), to test if they help to address the teamwork challenges. The analysis of students and teachers' reflections shows that they found team contracts as a useful intervention for early team building in both contexts but quickly lost relevance without structured follow-up. Peer evaluation was perceived as more effective in small teams, where students had more visibility into peers' work, while its value was limited in large teams due to subteam structures and reduced visibility of individual contributions. We also share implications for teachers of SE team project courses on how to introduce peer evaluation and team contracts in such courses.

## 4.1 Introduction

Team project courses are an important part of software engineering (SE) education. They give students an opportunity to work on real problems, often with external stakeholders, while practicing collaboration, communication, and professional skills [124]. These courses require students to collaborate within teams, share responsibilities, manage conflicts, and collectively deliver a project. Collaboration is important in SE team projects because software development involves frequent requirement changes, tight coupling between modules, and shared use of tools like version control and testing frameworks. The need for maintainable, error-free code requires regular communication, reviews, and knowledge sharing.

Team project courses face challenges related to a lack of commitment towards the project, unequal participation, conflicts, and coordination problems [124]. These issues can undermine learning outcomes and collaboration, causing student teams to fail to deliver. For course managers and examiners, it becomes a challenge to ensure that the project team plans the project, does fair and equitable work allocation, and each member works continually throughout the project's life.

Peer Evaluation (PE) [20] and Team Contracts (TCs) [42] are two well-known approaches that can help address these challenges. PE enables students to provide structured feedback on their team members' performance and behavior, while TC encourages teams to establish and agree upon shared expectations and behavior at the start of the project. Both interventions have been widely studied and adopted across disciplines, including SE education. However, the existing literature treats these interventions in isolation or within a single course context. Less attention has been paid to how their usefulness in improving teamwork may vary depending on the context of the SE project courses, such as team size, the presence of subteams, and the partial visibility of work inherent in subteam structures. Without these contextual details, educators may still find it difficult to perceive when and how these interventions are likely to be effective.

At Blekinge Institute of Technology (BTH), team project courses are a significant component of SE education. We also face the challenges discussed above; therefore, we designed and tested two interventions, for PE and TC, in two of our SE team project courses. These two courses differ in team size and internal structure i.e., a small-team project course (5–7 students per team) and a large-team project course (10–12 students per team, organized into subteams).

This paper makes the following contributions. First, we present an analysis of student and teacher reflections on PE and TC in small team versus large team project courses. By comparing experiences from small- and large-team project courses, we aim to understand the benefits and limitations of these interventions to support teamwork, as well as opportunities to improve their use in the context of two project courses. Second, we provide insights into how team size and the presence of subteam organizations within a large team influence accountability and feedback mechanisms. Third, we provide practical implications for teachers who plan to adopt these interventions in their own courses.

The rest of this paper is organized as follows: Section 2 presents the background of the shared experience and related work, Section 3 introduces the setup of the courses, including teaching and assessment mechanisms. The data and analysis methods are presented in Section 4. Section 5 shares how we designed and implemented the two interventions. Section 6 presents the results, i.e., analysis of students' and teachers' reflections. Section 7 further discusses the results and also shares the implications for teachers. Section 8 concludes the report.

## 4.2 Background and Related Work

This paper presents our experience of designing and using peer evaluation and team contracts in two software engineering team project courses. The details about the setup of these courses will be shared later in Section 4.3. The analysis of the students' reflections in the anonymous course evaluations from previous years, teacher workshops, and discussions among the teaching staff highlighted that students find it more challenging to deal with non-technical challenges (e.g., communication within the team, free-riding behavior of some team members, handling disagreements, etc.) than technical challenges. The teaching team decided to introduce changes to the team project courses to support students in addressing these challenges. The teaching staff identified two interventions that could potentially help student teams address the challenges: peer evaluation and team contracts.

Peer evaluation refers to the process in which students evaluate their peers' performance and contributions toward achieving a common goal [24]. Peer evaluations have been used in project courses to promote collaboration, enhance motivation, and support effective conflict management [20]. Other motivations to have peer evaluation include encouraging students to critically reflect on their team members' contributions [21–23] in terms of both performance and behaviour [24]. Peer evaluation may also reduce free riding behaviour and improve individual performance [43]. In addition to that, peer evaluation also provides teachers with valuable insights into team dynamics [44] that are otherwise difficult to observe [45, 46].

Teamwork expectations are also formalized through a team contract (at times also referred to as team charter or a team expectation agreement [27, 42]). Team contract is a collaboratively developed document that outlines each member's responsibilities and commitments [42]. Such contracts can include expectations, responsibilities, team etiquette, working procedures, meeting policies, and consequences for violations [42]. Prior research has shown that team contracts can help minimize conflicts, reduce social loafing [25–27, 42], and enhance accountability by making expectations and behavioral norms explicit [42, 47].

While many prior studies have investigated peer evaluation and team contracts, including their benefits and limitations, there is a lack of studies that have discussed the use of both interventions and analyzed how their perceived usefulness varies with team size and internal team structure.

In his recent study, Barman [42] suggests that team contracts should be complemented with strategies such as anonymized peer feedback and individual contribution-based grading to enhance teamwork and learning outcomes further. In our reported experience, we implemented both peer evaluation and team contracts within two SE team project courses at BTH. We analyzed how PE and TC supported teamwork, with their perceived value and limitations shaped by the course context, including team size and internal team structure. By comparing their use in two SE project courses with different team configurations, we aim to deepen understanding of when, why, and how these interventions support teamwork in SE education.

## 4.3 Setup of the Courses

We have two types of software engineering team project courses at our university:

- Small Team Project Course (STPC): In this project course, students work in teams (5-7 member teams) with an industry customer. Before starting this course, they have covered basic computing courses such as programming, databases, etc. The focus of the course is on applying core SE practices in a realistic but relatively contained team setting.

- Large Team Project Course (LTPC): In this project course, students work in larger teams (10–12 team members) on more complex projects with industry customers. Teams typically self-organize into subteams (e.g., frontend, back-end, testing). Before starting this course, students should have covered more computing courses and also passed a small team project course. Compared to the small-team course, this course places greater emphasis on coordination, architectural decision-making, and large-scale agile practices.

Both are semester-long (around 20 weeks) courses. Students work around 50% of full-time on these courses. We used the two interventions (team contract and peer evaluation) in the two project courses (one LTPC and one STPC) in 2025.

### 4.3.1 Learning Outcomes

In both project courses, students are expected to demonstrate a professional approach to software development, including working as an independent team, working with an industry customer, reflecting on their progress individually and as a team, along with technical proficiency in software development (from requirements management to software delivery). In LTPC, students work in larger teams on relatively more complex and larger project assignments. There are also enhanced expectations on LTPC student teams with regard to more thorough quality assurance (software testing, code reviews, etc.), software architecture, and use of measurements to track team progress and software quality.

### 4.3.2 Course Projects

Project ideas are solicited from local industry partners well before the start of the semester. Industry customers are aware of team sizes, available hours, and the courses students have already studied. The ideas are reviewed by the teaching staff to make sure that they fit the courses' requirements. The projects are set up in such manner that student teams progress through all phases of the SDLC, starting with the requirements and continuing until the delivery. The projects include the development of software applications, including web applications, mobile applications, plugins, and proof-of-concept solutions. The student teams work within the customer environment so that they get to experience a professional environment. Industrial partners appoint their staff as a customer who provides requirement specifications and also acts as the main point of contact for student teams and teaching staff.

### 4.3.3 Software Processes

Student teams use iterative and incremental development (two to three-week sprints) to deliver their solutions and deliverables. Teams follow agile practices, in particular Scrum, to plan their development, including sprint planning, backlog management, standups, retrospectives, etc. In LTPC teams, students usually organize themselves into subteams (e.g., 3 subteams of size 4). For such cases, they are encouraged to scale and complement their process with relevant concepts from scaled agile frameworks, such as Large-Scale Scrum (LeSS). Sprints are expected to end with a demonstration to the customer. Teams integrate customer feedback in future sprints. Teams also receive feedback, oral and written, from their team mentors. The last sprint is labeled as a *Delivery and Handover Sprint*, in which the teams are expected to give a final demo and also hand over the solution along with the relevant documentation to the customer.

### 4.3.4 Teaching and Assessment

The pedagogical foundation in these project courses (STPC and LTPC) is *learning by doing*, supported by continuous reflection and improvement. There are team-building workshops in both courses. In the STPC, relatively more lectures are organized on different software engineering topics such as requirements engineering, agile estimation and planning, agile development practices and Scrum, software architecture, and software testing. In LTPC, the focus is more on student-led seminars and workshops. Additional topics in LTPC include learning from the past, working in a large team, large-scale agile, risk management, and software measurement.

In addition to lectures and workshops, each team is assigned a team mentor. The mentors are university staff members with a strong knowledge of software engineering practices. These mentors meet students regularly, on a weekly basis during the first few weeks of the course, and then at least once every sprint. They provide both written and oral feedback to students and also support teams in reflecting on their progress, providing suggestions (e.g., on how to work with customers or requirements) and resolving issues (e.g., addressing team conflicts).

For the project's assessment, we follow a continuous and formative assessment approach. There are weekly individual reports in which students reflect on their progress during the last week. In addition, there are sprint-wise team reports in which teams reflect on their progress during the previous sprint. These reports are reviewed and discussed by the team mentors. Teams are also expected to give a demo at the end of each sprint to their customers.

The course examiner performs a mid-review with each team. The mid-review includes a presentation from the team about their progress. Teams use a review checklist to show their progress. The examiner also looks at different software engineering artifacts (user stories and backlog, estimates, test cases, code repo, implemented features, etc.). The final grade for the course project is based on the feedback from the team mentor, industry customer, and the examiner's own analysis of the team's progress.

Students are also required to submit an individual reflection report at the end of the course. In these reports, students reflect on their individual contributions, teamwork, challenges faced, and lessons learned. For the two course instances under study, students were also asked to share their reflections on the use of peer evaluations and team contracts in the project courses.

## 4.4 Methods

The primary objective of the reported investigation was to assess the perceived usefulness of peer evaluation and team contracts in supporting teamwork in SE team project courses. In particular, we were interested in understanding *how students and teachers perceive the use of these interventions in different team project courses.*

### 4.4.1 Data Collection

We had 62 students (organized as ten teams) in STPC and 23 students (organized as two teams) in LTPC for the course instances under study. We used multiple data collection methods in the reported study, including:

- **Individual reflection reports**[1] All students submitted a mandatory individual reflection report at the end of the course. In these reports, students were explicitly asked to reflect on (1) their own contributions, (2) strengths and weaknesses of their team (3) their experiences with peer evaluation and team contracts, and (4) suggestions to improve the team contracts and peer evaluation process. Reflections were typically 2-3 pages long. These reports are our main data source for student reflections on peer evaluation and team contracts. In total, we had 62 individual reports for STPC and 23 for LTPC. As previously discussed, students need to share their reflections on peer evaluation and team contracts in designated sections of their individual reports. The verbatim text of these sections was extracted for further analysis. Unique fictitious student IDs (i.e., Student 1, Student 2, ...) were assigned to the extracted text segments from these individual reports, replacing the students' names and registration numbers.

- **Literature analysis**: The recent systematic review by Tenhunen et al. [5] was our starting point to identify the related publications on the use of peer evaluation in SE team project courses. We analyzed the identifed works to find the commonly used aspects in peer evaluation in team project courses (more details in Section 4.5.1).

- **Focus Groups**: We conducted two focus groups. Each focus group included three teachers involved in teaching of the two project courses considered for the study and lasted approximately 120 minutes. The first focus group was conducted with teachers to incorporate their inputs towards the design of the peer evaluation survey (see Section 4.5.1) . The second focus group was conducted with the teachers to collect their reflections on the use of peer evaluation and team contracts in team project courses. The sessions were audio-recorded and summarized through detailed notes. Insights from the focus groups were also analyzed using thematic analysis [60]. The focus group participants were aware and agreed that the data collected during the sessions can be used in a publication.

### 4.4.2 Data Analysis

We applied thematic analysis following Braun and Clarke [60] to analyze the text extracted from student reflection reports. The process followed the following steps: *i) familiarization with the data:* All authors first went through the text to familiarize themselves with the data, *ii) generating initial codes:* The first and second authors

[1] Student submissions are official public documents and may be disclosed upon request, subject to a confidentiality assessment. For this study, all personal identifiers were replaced with fictitious IDs, and the data was handled in accordance with GDPR and institutional procedures.

jointly coded the reflections from a small sample of students. The first author then coded the reflections of the remaining students independently, *iii) code validation:* The third author validated all codes to check if they correctly represent the underlying data, *iv) defining and reviewing the themes:* Two joint workshops were later conducted to complete the thematic analysis. The authors went through all the codes and developed a shared understanding and consensus about them. Wherever appropriate, the codes were merged and modified to improve their representativeness of the underlying data. Finally, the related codes were grouped into the larger themes.

Teacher reflections on the use of peer evaluation and team contracts were also thematically analyzed using the same process

## 4.5 Intervention Design

In this section, we describe how we designed and implemented the two interventions: Peer evaluation and Team contract.

### 4.5.1 Peer Evaluation

In a recently published systematic review on project courses, Tenhunen et al. [5] identified 39 studies that have reported the use of peer evaluation in a SE team project course. We extracted these studies for further analysis. We identified that peer evaluation is normally conducted as a survey in which team members are asked to rate the performance of their peers in various aspects. These aspects can be broadly divided into technical (e.g., contribution to coding and testing) and non-technical (e.g., timely communication, participation in meetings, equal contribution) categories. The first author compiled a list of the most frequently used aspects from the reviewed studies and compared it against the well-established teamwork evaluation framework, CATME [106] to ensure alignment with recognized teamwork aspects. This list was then shared with the teaching team for feedback. Three teachers, involved in teaching team project courses, individually reviewed the list. The first author prepared a peer evaluation survey based on the teachers' review. The survey was further discussed in a focus group involving three teachers and the first author as mediator. The survey questions and the scale were adjusted and finalized during the focus group.

The peer evaluation survey [2] was conducted using an online survey tool (Google forms), with a different survey instance for each course. The survey consists of 17 rating criteria (see Table 4.1) on a Likert scale (1-6, where 1 is very poor and 6 represents very good performance). Students could also provide open-ended comments for each team member, allowing them to share additional qualitative feedback about their peers related to or beyond the survey rating criteria.

[2] Peer and Self Evaluation Survey

Table 4.1: Peer Evaluation Criteria used in PE surveys

| **Peer Evaluation Criteria** |
| --- |
| Equitable contribution to team's work |
| Communication with the team |
| Coordination and collaboration with the team |
| Fulfillment of commitments to the team |
| Facilitating team members |
| Attendance in team meetings |
| Being a team player |
| Timely completion of assigned tasks |
| Communication with the customer |
| Fulfillment of commitments to the customer |
| Openness to feedback |
| Conflict resolution |
| Professional behavior |
| Contributions to requirements management and specification |
| Contributions to project planning |
| Contributions to design and implementation |
| Contributions to testing and quality assurance |

It was decided that the peer evaluation survey would be implemented repeatedly at the end of each sprint. The idea was to enable teachers to view the progression of individuals and teams as projects move towards their conclusion. All responses were kept confidential and were used by the teachers to assess team dynamics and contributions. Students could also request their own peer evaluation results (in an aggregate and anonymized form only). Teachers and mentors used peer evaluation results to identify issues that required follow-up with individual students.

### 4.5.2 Team Contract

As mentioned earlier, team contracts have been previously used in team project courses at our university, but not consistently. Also, the contracts were not designed in a structured way. After discussions within the teaching team, we decided to design a workshop at the very beginning of the project course to ask the student teams to design their team contracts.

The teams and their customers were announced immediately after the first introductory lecture, once the course participants were confirmed. The team contract workshop is the second session of the course, organized in the first week. The workshop is organized in a room that is set up as an active learning classroom where members of each team can sit together on a round table. The workshop was organized as follows:

- Introduction of team members - some ice breaking
- Team discussion: Initial discussion about teamwork, potential communication channels and tools, and other important aspects.
- Developing the team contract: Discuss the ground rules for the team and start organizing into a team contract. The teaching staff provides them with a list of aspects or questions to consider for these ground rules. The workshop guidance document is available online[3].
- The teams could continue to work on their contracts beyond the workshop. They are given a couple of days to improve and submit a team contract signed by all members.
- The teaching staff reviews the contract and gives feedback. The feedback may not result in a new submission, but just additional questions the team may need to consider.

Figure 4.1 shows a Team Contract that was signed and submitted by the team (from Small Team Project course) after the workshop.

## 4.6 Results

The results are presented in two subsections. Section 4.6.1 outlines students' reflections on their experiences with team contracts and peer evaluation, while the Section 4.6.2 summarizes teachers' reflections on the use of these two interventions in their respective courses.

### 4.6.1 Student Reflections

In this section, we present student reflections on the use of peer evaluation and team contracts in the context of small team (STPC) and large team (LTPC) project courses.

#### *4.6.1.1 Peer Evaluation*

Thematic analysis of students' reflections on peer evaluation revealed five main themes, including, benefits of peer evaluation, limitations of peer evaluation, its impact on teamwork and individuals, and suggestions for improvement. Table 4.2 summarizes how students perceived the role of peer evaluation in small-team (STPC) and large-team project courses (LTPC). In Table 4.2, *n* denotes the total number of student reports analyzed for each course, the first number indicates the number of reports in which a given theme was identified at least once and the percentages indicate

[3] The workshop guidance document

**1. Purpose**

This contract makes sure everyone on the team knows our main goals and how we will work together. By signing, we all agree to follow these rules and do our best on the project.

**2. Team Goals**

As a team, we aim to:

- Use a professional way of working, such as good planning and organized methods.
- Understand why quality checks and on-time delivery are important in software development.
- Show our skills as future professionals in the industry.

**3. Team Rules**

3.1 Deadlines

- Set personal deadlines that are earlier than the official ones.
- If you think you'll miss a personal deadline, tell the team as soon as possible.

3.2 Meetings and Communication

- Keep in touch regularly to share updates and problems.
- Attend meetings or let the team know as soon as possible if you can't make it.

3.3 Decisions and Conflicts

- Talk openly if there's a disagreement. If we can't agree, we'll vote.
- No gossip - if there's a problem, address it with the person involved or with the whole group.

3.4 Roles and Responsibilities

- Choose tasks you're interested in or good at.
- Give honest feedback on each other's work - praise when it's good and be clear if it needs improvement.

3.5 Code Quality

- Follow the same coding style as the team or the official style for that language.
- Comment your code well so it's clear to others, but don't overdo it.

3.6 Breaking the Rules

- If someone keeps breaking these rules or not doing their part:
    1. Talk with them directly.
    2. If it doesn't improve, get help from the teacher.

Student names and signatures are redacted

**Figure 4.1:** An example Team Contract from the STPC

**Table 4.2:** Student reflections on peer evaluation in small team (STPC) and large team (LTPC) project courses.

| Theme | Student Reflections on Peer Evaluations | STPC (n=62) | LTPC (n=23) |
|---|---|---|---|
| Benefits of PE | Peer evaluation helped highlight team member's contributions | 26 (41.94%) | 3 (13.04%) |
| | Reporting anonymously felt safe | 20 (32.26%) | 3 (13.04%) |
| | Peer evaluation provided insights into team dynamics (inside/outside) | 7 (11.29%) | 6 (26.09%) |
| | Opportunity to provide freetext comments in peer evaluation was useful | 2 (3.23%) | 0 (0%) |
| | Peer evaluation ratings on a numeric scale was good | 2 (3.23%) | 0 (0%) |
| | Having multiple rounds of peer evaluation was good | 2 (3.23%) | 0 (0%) |
| Limitations of PE | Peer evaluation can be biased due to personal interactions | 14 (22.58%) | 5 (21.74%) |
| | Peer evaluation results were not shared | 8 (12.9%) | 12 (52.17%) |
| | Peer evaluation had limited preceived value | 7 (11.29%) | 9 (39.13%) |
| | Peer evaluation was time consuming with limited benefit | 6 (9.68%) | 4 (17.39%) |
| | No guidance on improvement based on peer evaluations | 5 (8.06%) | 6 (26.09%) |
| | Evaluators may have not given due attention while evaluating peers | 5 (8.06%) | 0 (0%) |
| | Too many evaluation criteria in peer evaluation survey | 4 (6.45%) | 3 (13.04%) |
| | Difficult to recall and give opinion on past events while doing peer evaluation | 2 (3.23%) | 1 (4.35%) |
| | Peer evaluation can become a substitute for open conversations | 2 (3.23%) | 0 (0%) |
| | Possible inconsistent interpretation of peer evaluation criteria | 1 (1.61%) | 2 (8.7%) |
| | Insufficient information to fairly evaluate members outside subteam | 2 (3.23%) | 11 (47.83%) |
| | Some peer evaluation criteria felt irrelevant | 1 (1.61%) | 1 (4.35%) |
| Impact on team work | Peer evaluation promoted reflection | 28 (45.16%) | 4 (17.39%) |
| | Peer evaluation reinforced accountability within the team | 21 (33.87%) | 11 (47.83%) |
| | Peer evaluation increased motivation and professionalism | 20 (32.26%) | 2 (8.7%) |
| | Peer evaluation helped improve team dynamics | 16 (25.81%) | 2 (8.7%) |
| | Peer evaluation improved team performance | 11 (17.74%) | 0 (0%) |
| | Peer evaluation improved communication within the team | 9 (14.52%) | 3 (13.04%) |
| | Peer evaluation created awareness of each others strengths and weaknesses | 7 (11.29%) | 0 (0%) |
| | Peer evaluation improved technical and non technical aspects of working | 6 (9.68%) | 0 (0%) |
| | Peer evaluation motivated attendance and participation | 5 (8.06%) | 4 (17.39%) |
| | Peer evaluation resulted in equal distribution of work | 3 (4.84%) | 0 (0%) |
| | Peer evaluation had limited impact on team performance | 27 (43.55%) | 13 (56.52%) |
| | Peer evaluation caused negative impact on team dynamics | 12 (19.35%) | 2 (8.7%) |
| Impact on individuals | Peer evaluation helped individuals to identify areas to improve | 20 (32.26%) | 0 (0%) |
| | Peer evaluation motivated individuals to work in a better manner | 17 (27.42%) | 2 (8.7%) |
| | Peer evaluation made individuals more self aware | 5 (8.06%) | 0 (0%) |
| | Being evaluated by peers felt stressful | 6 (9.68%) | 1 (4.35%) |
| Suggestions for improvement | Share results of peer evaluation | 17 (27.42%) | 8 (34.78%) |
| | Have a follow up discussion on peer evaluation | 11 (17.74%) | 0 (0%) |
| | Clarify peer evaluation criteria | 10 (16.13%) | 1 (4.35%) |
| | Reduce peer evaluation criteria | 7 (11.29%) | 3 (13.04%) |
| | Replace numeric rating scale with a qualitative one in peer evaluation | 6 (9.68%) | 3 (13.04%) |
| | Change frequency of peer evaluation | 6 (9.68%) | 2 (8.7%) |
| | Have in person peer evaluation | 5 (8.06%) | 0 (0%) |
| | Include text areas to complement each rating in peer evaluation | 5 (8.06%) | 1 (4.35%) |
| | Include some incentive for peer evaluation | 2 (3.23%) | 0 (0%) |
| | Add a "not applicable/not sure" option for each evaluation criteria | 1 (1.61%) | 2 (8.7%) |
| | Add peer evaluation as an assignment | 1 (1.61%) | 0 (0%) |
| | Provide more oppotrunity to give more elaborated self-evaluation | 1 (1.61%) | 0 (0%) |
| | Introduce lecture or training on doing peer evaluation | 1 (1.61%) | 0 (0%) |
| | Peer evaluation criteria should evolve thorugh the project | 1 (1.61%) | 0 (0%) |
| | Design peer evaluations to accomodate subteams | 0 (0%) | 5 (21.74%) |
| | Validate peer evaluation with code repo data | 0 (0%) | 1 (4.35%) |
| Other reflections | Evaluators were not honest at times | 8 (12.9%) | 2 (8.7%) |
| | Felt awkwardness in self or peer evaluation | 4 (6.45%) | 0 (0%) |
| | Interest in peer evaluation declined over time | 3 (4.84%) | 0 (0%) |
| | Some peer evaluation criteria were difficult to evaluate | 2 (3.23%) | 0 (0%) |
| | Evaluating peers got easier with passage of time | 1 (1.61%) | 0 (0%) |
| | Peers may have limited experience to evaluate others | 0 (0%) | 1 (4.35%) |

*n is the number of student reports analyzed
first number = number of students reporting the theme, number in parentheses = percentage.

the proportion of reports mentioning that theme. These percentages are intended to convey the relative prominence of themes within the dataset.

***Benefits of peer evaluation:*** In STPC, the most frequently reported strength was that peer evaluation helped highlight team members' contributions (41.9%). A student in the small team project course opined, *"Some team members were less active early on, and I believe the evaluations helped to address this—particularly after the second round, which showed tangible improvement in their engagement. Over time, they became more involved and productive"*. In LTPC, a student said *"I think peer evaluation is a useful tool for providing the examiner with a broader and more nuanced picture of the contribution of each student in a larger team project."*

Overall, students in STPC viewed the peer evaluation process as beneficial due to anonymity (32.26%). Students in STPC felt that peer evaluation provided them with a sense of safety to give anonymous feedback. A student of STPC commented that *"the anonymity of peer review made it so you could be totally honest without pointing fingers or destroying the team's spirit."*

In contrast, relatively fewer students emphasized such aspects in LTPC (both 13.0%). Instead, a larger proportion of LTPC students (26.1%) highlighted that peer evaluation provided valuable insights into team dynamics, a reflection less prominent in STPC (11.3%).

***Limitations of peer evaluation:*** Both STPC (22.6%) and LTPC (21.7%) students mentioned that personal interactions could influence peer evaluation, raising concerns about fairness. One student from LTPC reflected, *"there could be a certain influence of personal connections. It was easier to give a more fair assessment to someone you had more contact with or received help from."*

LTPC students reported more frequently that peer evaluation results were not shared (52.2% vs 12.9%), students did not have enough information to fairly evaluate team members outside their subteam (47,8% vs 3,23 %)and that the process had limited perceived value (39.1% vs 11.3%).

On the other hand, STPC students were more concerned that evaluators might not give enough attention to the process and may provide lazy feedback. One student in STPC said *"I think it can be easy for students to just put all 3s or all 5 just to get it over with, without addressing the problems that the group might have"*. A small number of students in STPC also had similar perceptions that lazy feedback (i.e., providing the same score to everyone) or rushed feedback (i.e., without due attention to evaluation criteria) can happen (8.1%).

***Impact of peer evaluation on teamwork:*** In STPC, most students described peer evaluation as beneficial. Nearly half the students (45.2%) reported that it promoted reflection, while others emphasized its role in reinforcing accountability (33.9%), increasing motivation and professionalism (32.3%), and improving team dynamics (25.8%). A student in STPC viewed that peer evaluation *"served its purpose well and encouraged both self-reflection and group awareness. When people see how others perceive their efforts, it can lead to real personal growth."* However, many students also noted that peer evaluation had a limited impact on teams' performance (43.55%). Student reflection from LTPC showed a mixed pattern. The most common reflection was that peer evaluation had a limited impact on team performance (56.5%). One student from LTPC stated that *"the peer evaluation had minimal to no impact on performance."* At the same time, almost half (47.8%) acknowledged that it reinforced accountability. Other positive effects, such as promoting reflection (17.4%) or improving communication (13.0 %), were much less frequently mentioned by students of LTPC than in STPC.

When it comes to the negative impacts of peer evaluation on teamwork, some students mentioned that peer evaluation may have a negative impact on team dynamics (19.3% in STPC and 8.7% in LTPC).

***Impact of peer evaluation on Individuals:*** In STPC, most students considered peer evaluation to positively impact them as individuals, particularly in helping them to identify areas for improvement (32.3%) and motivating them to work more effectively (27.4%). A smaller number also reported greater self-awareness (8.1%). One student from STPC said, "*The peer evaluation had a significant impact on my own performance. After receiving feedback about my communication style, I made a conscious effort to be more clear and concise in my messages and contributions during meetings."*

At the same time, some students found the process stressful (9.7%). A student in the STPC reflected that *"overthinking a lot became a bit stressful because I was scared that my team would not see my contributions, and it would, in the worst case, make me fail the course."* In LTPC, individual impacts were less prominent, with fewer students reporting motivation (8.7%) or stress (4.3%).

***Suggestions for improving peer evaluation:*** Students in both contexts suggested ways to improve peer evaluation. One student from STPC suggested that there should be a "team meeting about the peer evaluation" that should be "held after every sprint". Further, in STPC, the most common suggestions were to share results (27.4%), hold follow-up discussions (17.7%), and clarify evaluation criteria (16.1%). Similarly, in LTPC, students emphasised sharing results (34.8%) and proposed adjustments specific to large groups, such as designing evaluations to accommodate subteams (21.7%).

#### 4.6.1.2 Team Contracts

Thematic analysis of students' reflections on team contracts revealed three overarching themes: benefits of using team contracts, limitations of using team contracts, and suggestions for improving the use of team contracts in team project courses. Table 4.3 summarizes how students perceived the role of team contracts in small-team (STPC) and large-team project courses (LTPC).

***Benefits of team contracts:*** About half of the students in both courses (47.6% in STPC; 47.8% in LTPC) shared that the team contracts helped establish shared ex-

**Table 4.3:** Student reflections on team contracts in small team (STPC) and large team (LTPC) project courses.

| Theme | Student Reflections on Team Contract | STPC (n=62) | LTPC (n=23) |
|---|---|---|---|
| Benefits of team contracts | contract helped to establish shared expectations of teamwork | 30 (47.6%) | 11 (47.8%) |
| | Contract helped to build commitment | 11 (17.5%) | 6 (26.1%) |
| | Contract helped to prevent conflicts | 9 (14.3%) | 3 (13%) |
| | Contract helped to reinforce accountability | 8 (12.7%) | 5 (21.7%) |
| | Contract helped to develop respect and professionalism | 8 (12.7%) | 3 (13%) |
| | Contract helped to build team familiarity | 4 (6.4%) | 0 (0%) |
| | Contract felt natural and easy to adhere | 3 (4.8%) | 1 (4.4%) |
| Limitations of team contract | Contract was graudually forgotten | 17 (27%) | 5 (21.7%) |
| | Contract did not have much impact | 9 (14.3%) | 8 (34.8%) |
| | Effectiveness of contract depends on member compliance | 4 (6.4%) | 1 (4.4%) |
| | Contract was too trivial | 2 (3.2%) | 2 (8.7%) |
| Suggestions for improvement | Contract should be revisted periodically to improve its utility | 6 (9.5%) | 0 (0%) |
| | Contract should be updated to evolving team needs | 1 (1.6%) | 2 (8.7%) |
| | Contract should include working standards | 0 (0%) | 2(8.7%) |
| | Discuss strengths and weaknesses of members at the time of contract | 0 (0%) | 1 (4.4%) |
| | Impose penalty for non-compliance to contract | 0 (0%) | 1 (4.4%) |

*n is the number of student reports analyzed;
first number = number of students reporting the theme,
number in parentheses = percentage.

pectations of teamwork. One Student in STPC states that, *"team contract established the agreed-upon principles and rules, which likely influenced the team's awareness of expected behavior and processes for handling challenges..."*. Other frequently mentioned benefits included fostering commitment (17.5% in STPC; 26.1% in LTPC), reinforcing accountability (12.7% in STPC; 21.7% in LTPC), and developing respect and professionalism (12.7% in STPC; 13% in LTPC). A student reflected on this by saying: *"The team contract served us as a reminder that we had rules and responsibilities we had to follow no matter the circumstances"*. A relatively smaller number of students felt that contracts were helpful in preventing conflicts (14.3% in STPC; 13% in LTPC) or building team familiarity (6.4% in STPC; none in LTPC). In STPC, one student mentioned, *"The team has adhered closely to the contract, resulting in zero conflicts or drama, which is unusual for a group of inexperienced developers"*. Some students described the contract as natural and easy to follow, noting that many of its elements aligned with their intuitive approaches to collaboration (4.8% in STPC; 4.4% in LTPC).

***Limitations of students on team contracts:*** Despite the perceived benefits, some students reported that the team contract was gradually forgotten (27% in STPC; 21.7% in LTPC), suggesting that its influence on teamwork diminished as projects progressed. . In addition, 14.3% of STPC students and 34.8% of LTPC students noted that the contract did not have much impact on their team's way of working. One student in STPC mentioned it as *"Although we created a team contract at the beginning,* *it was rarely referenced and didn't meaningfully influence our day to day work"*. A few students emphasized that the contract's effectiveness largely depended on team members' compliance, that is, how consciously and consistently members tried to adhere to it (6.4% in STPC; 4.4% in LTPC). One student reflected, *"I would say that the team contract is a good idea, only if the developers are responsible enough to stick to their words"*. A few students criticized the contract as too basic or trivial, arguing that it offered little meaningful guidance for teamwork (3.2% in STPC; 8.7% in LTPC).

***Suggestions for improving the use of team contracts:*** Several students provided suggestions to improve the effectiveness of team contracts. In STPC, 9.5% of students suggested that contracts should be revisited periodically, while this was not mentioned in LTPC. One student in STPC said, *"The contract might have been more influential if we had revisited it at each retrospective or sprint planning session"*. In contrast, a few LTPC students (8.7%) recommended updating contracts to evolving team needs, including technical working standards (such as coding rules and communication channels) in team contracts, and adding mechanisms such as penalties for non-compliance (4.4%). One student in LTPC mentioned, *"I think we should have agreed on consistent coding rules and working methods earlier in the project, especially when we signed the contract, since there was eleven of us"*.

Overall, these suggestions highlight students' view that team contracts should be used as living documents that are adapted and revisited to remain relevant throughout the project.

### 4.6.2 Teacher Reflections

Teachers' reflections on the use of peer evaluation and team contracts in their respective courses were collected through a focus group. The main themes regarding these two interventions are presented in the following subsections.

#### 4.6.2.1 Peer Evaluation

We identified the following themes from the teachers' reflections on peer evaluations:

***Benefits of peer evaluation:*** Peer evaluation was acknowledged to promote a sense of accountability among the students. Focus group participants (teachers) noted that students became more conscious of their performance once they realized peers were evaluating them. Teacher (T2) highlighted that *"students knew who underperformed, even without naming"*. Peer evaluation was also seen to empower students. Teacher (T2) stated that *"... students were very happy to have this possibility, to rate and rank their peers because this is where they were reflecting on the performance. They felt there was no other place to do it otherwise. And they felt empowered."*

These reflections suggest that peer evaluation made students' performance visible, promoted a sense of responsibility, and provided them with a platform to voice their perceptions of team contributions.

***Follow-up and post-evaluation action:*** The teachers highlighted that the value of peer evaluation depended on the follow-up actions. T1 emphasized, *"...most important is the after discussions. It's what do we do after?"* In contrast, T2 admitted, *"We did not manage to do that well,"* pointing to a gap in structured follow-up.

Different follow-up strategies were used by the participants when red flags appeared in peer evaluation results. T2 discussed the results with the team mentor and arranged individual meetings with students who had received poor evaluations. Similarly, T1 and T3 held meetings with problematic students. In addition, T2 revisited peer evaluation issues with teams during the mid-review.

The teachers mentioned that the process also carried the risk of unfair conclusions. T1 recalled, *"One person got bad judgment. I got back to that person and we had a discussion. Then I got to know that the problem was not only with that person, it was actually the group. This was of course not visualized in peer evaluation. So I had to have a discussion with the group afterwards."* T2 reported a similar experience.

Across teachers, there was an agreement that peer evaluation data should only serve as a starting point. To ensure fairness, teachers needed to investigate the contextual details behind the results, in order to understand the bigger picture. Overall, this theme reflects a consensus that peer evaluation, on its own, was insufficient. Its impact depended on structured, timely, and teacher-led follow-up to achieve fair and meaningful outcomes.

***Contextual suitability (small vs large teams):*** Team size was identified as a major challenge. The teachers questioned, *"Should we do it at all for large teams?"* T1 reflected, *"when you come into the large team, you then will have some practical issues."* With the subteam structure in LTPC, students not working in the same subteam often lacked sufficient insights to evaluate their peers fairly. As T2 observed, *"Subteams reduce visibility,"* and T3 noted, *"a majority of students commented about their peer that he/she is not in my subgroup."* T2 argued, *"We may not need it in the large team project course."* Whereas, T3 suggested that, in the context of LTPC, students should only be required to provide peer evaluation for their peers within the same subgroup. This approach may enable insightful evaluations and reduce evaluation fatigue for students working in large teams.

***Logistical challenges:*** Teachers highlighted several practical issues related to peer evaluation, including participation, frequency, and timing. T3 noted that if all team members did not respond to the peer evaluation survey, the results for the entire team can become skewed. Similarly, T2 and T3 suggested that peer evaluation should be conducted frequently, ideally once per sprint. T2 emphasized, *"if you only do it once, then actually it's better if you don't do it at all."* There were also reflections on the timing of peer evaluation. T2 and T3 suggested implementing peer evaluation during the working sprints, while skipping it for the first and the final sprint. Their reasoning was that, at the beginning, students need time to establish team dynamics, whereas at the end, they are often pressed by the deadlines for the final deliverable. Overall, logistical challenges were related to participation, timing, and frequency. These factors shaped participants' perceptions of when and how peer evaluation could provide reliable insights.

***Limitations of peer evaluation:***

Participants raised concerns about the purpose and appropriate use of peer evaluation. One Teacher (T3) emphasized its limits, stating, *"We cannot let peer evaluation be used for accusations or reporting mechanism for events or incidents."* This reflects an understanding that peer evaluation data should not be used as evidence in conflicts. While peer evaluation helped to reinforce a sense of accountability, its contribution to critical thinking and reflection was questioned. T1 argued that *"critical thinking [is] not visible as students just filled in,"* and T2 added, *"critical thinking as an outcome of peer evaluation is unclear; however, reflection is a possible outcome."*

Regarding the use of peer evaluation as an assessment tool, T3 suggested, *"if you want to keep it for assessment, then the students need to be very formal and honest."* T3 also noted that this could raise concerns about the fairness and objectivity of assessments.

Taken together, these reflections suggest that while peer evaluation can promote accountability and some reflection, its role in encouraging critical thinking or serving as an assessment tool remains unclear.

#### 4.6.2.2 Team Contracts:

We identified the following themes from teachers' reflections on the use of team contracts.

***Benefits of team contracts:*** The teachers observed that contracts can function as effective team-building tools. Teacher T2 mentioned, *"The signing of the contract took place during a mandatory, in-person session that acted as an icebreaker."* Other participants also agreed that signing the contract helped establish team expectations and strengthen collaboration at the beginning of the project.

***Follow-up for team contracts:*** The teachers reflected that while contracts were created at the beginning, they can only be impactful if there is proper follow-up. T1 noted, *"we need to consider follow-up too,"*. T2 mentioned that he could only follow-up on team contracts towards the end of the project *"...brought it up only at final presentation, but then it was too late."* There was agreement that responsibility for follow-up should not be delegated. T2 suggested *"Course manager should supervise the follow-up of contracts, not mentors"*. T2 and T3 proposed incorporating team contracts into mid-review discussions: *"Mid-reviews should combine peer evaluation data and contract discussions. We need to formalize contract review as well."*.

## 4.7 Discussion

This study explored students' and teachers' reflections on the use of peer evaluation and team contracts in small-team (STPC) and large-team project courses (LTPC). The results show that while both tools were valued, their benefits and challenges varied depending on the team size and the way they were used. We discuss key findings of our study for both interventions and how they vary among the small and large team project courses. Additionally, we also share implications for teachers of software engineering team project courses.

### 4.7.1 Peer Evaluations

In small teams, peer evaluation was perceived to be useful to enhance learning and teamwork. Many students said it helped them to reflect, to recognize contributions, and to stay motivated. It also supported accountability within the team. These findings align well with what have been presented in existing literature [20, 24].

In large teams, these benefits were less visible. The common view was that students did not have enough knowledge or information to fairly evaluate team members outside their subteam. This lack of visibility may have contributed towards limited perceived impact of peer evaluation on performance. Still, many of the LTPC students found it helpful for understanding team dynamics.

The limitations of peer evaluation also looked different in each context. Small teams struggled with fairness at the individual level, while the large teams struggled with a lack of visibility and transparency of the process. In STPC, students mentioned concerns about the evaluation bias of their peers. Similar concerns about subjectivity have been discussed in [23], which recommends complementing peer evaluation with objective measures of effort.

In LTPC students highlighted structural issues more, such as peer evaluation results not being shared and a lack of enough information to fairly evaluate team members outside the subteam. These findings align with prior research on social loafing suggesting that accountability is more effective when individual contributions are visible and identifiable [27]. Similarly, feedback theory emphasizes that feedback must be contextualized and actionable to support learning and behavior change. Our findings indicate that when these conditions are not met, as in LTPC with lower visibility across the subteam structure, evaluation alone is insufficient to provide actionable feedback and contextualized learning.

In STPC, some students also worried that peer evaluations could negatively impact team dynamics, particularly when concerns about individual performances were raised. Similiar findings on peer evaluation generating some negative impact on team dynamics have been discussed in [43]. Our findings suggest that anonymous evaluations were appreciated as a way to prevent tensions. Furthermore we highlight the importance of constructive follow-up by the teachers, ensuring that peer evaluation contributes towards learning rather than conflict. At the individual level, peer evaluation helped STPC students identify areas for improvement and motivated them to work better, while LTPC students mentioned personal impacts less frequently. It may suggest that peer evaluation works as a tool for personal development in small teams, whereas in large teams, it mainly acts as a system for group accountability.

Students in both courses also offered suggestions to improve the peer evaluation process. STPC students wanted clearer evaluation criteria, follow-up discussions, and simpler surveys. LTPC students asked for more structural changes, like designing evaluations for subteams. Some students felt their peers were not honest and accurate due to the fear of being rude or affecting the relationships with their team members. Concerns about students not being honest while doing peer evaluation have also been mentioned by [44].

Like students, teachers also suggested accountability as a clear benefit of peer evaluation. Both students and teachers emphasized the need for more transparency and follow-up discussions so that peer evaluation results could serve as a constructive mechanism. Without follow-up discussions or contextual understanding, results risked being incomplete or misleading. Teachers emphasized that peer evaluation results could be a starting point for conversations. This coincides with students' reflections for more transparency and follow-up discussions. Another area where teachers shared reflections on peer evaluation was its feasibility in the large team setup. In view of the teachers, peer evaluation may not be relevant in large teams, mainly due to the existence of subteams. Some students in LTPC also had similar reflections on their inability to rate members of other subteams.

### 4.7.2 Team Contracts

The students in both STPC and LTPC valued contracts for helping to establish shared expectations, which aligns with the findings suggested by Barman et. al [42]. Team contracts were also considered helpful for building commitment, accountability, and professionalism, findings that align well with [27, 42]. Teachers also mentioned that team contracts provided a useful foundation for setting expectations early on and served as an important team-building tool. Suggestions to improve the use of team contracts varied across the two courses. STPC students emphasized revisiting contracts during the project. Whereas LTPC students suggested adapting contracts to evolving team needs, adding working standards as part of the contract, and even introducing penalties for non-compliance. This indicates that both groups see value in team contracts. They want to incorporate mechanisms that keep team contracts relevant and ensure adherence to these throughout the entire project. Teachers offered similar recommendations, highlighting the need to revisit contracts periodically, integrate contract discussions into mid-project reviews, and combine them with peer evaluation data to gain a more holistic view of team performance. Comparable findings were reported by Fronza et al. [25], who suggested monitoring adherence to team contracts through periodic short surveys during the project.

Overall, the results suggest that peer evaluation and team contracts are useful interventions, but work differently depending on the team size. In small teams, peer evaluation supports personal growth and reflection, while contracts risk being overlooked after the initial few days unless actively reinforced. In large teams, contracts seem more important for accountability, while peer evaluation had more issues with visibility of team members' performance, and perceived value unless adapted to subteams. Both students and teachers agreed that follow-up and transparency are crucial for peer evaluation and team contracts to improve their usefulness.

### 4.7.3 Implications for Teachers

In this section, we share the implications of using peer evaluation and team contract in two software engineering team project courses with teachers of team project courses.

The implications related to the peer evaluation include:

- Both teachers and students were largely positive about the use of peer evaluations in the STPC. However, peer evaluation did not work that well in the LTPC. This could largely be attributed to the existence of subteams within the large team in LTPC, which makes it hard for a team member in one subteam to assess the performance of members in other subteams. An alternative could be to implement peer evaluation within subteams. However, it will be time-consuming for teachers to manage peer evaluations that way. More importantly, it may also reduce or compromise anonymity (e.g., subteams of size 3). Taken together, we believe peer evaluation is potentially an effective intervention to improve teamwork (e.g. improving reflection, individual contributions, and accountability) in team project courses where the team size is not larger than a regular agile or scrum team (5-7 members).

- When peer evaluation is used in team project courses, we suggest that teachers plan follow-up sessions with students to discuss common issues (at an aggregate level) raised in the peer evaluation survey. In addition, the results of the peer evaluation (anonymized and aggregated) should be shared with individuals and teams. Without these steps in place, the value of peer evaluation may diminish, and it may just become a formality for students to respond to these surveys.

- Finally, if multiple data points highlight a lack of contribution or other major issues by one or more students, we suggest that teachers conduct a dedicated meeting with such students to understand the situation and provide them with constructive feedback on how they can and should improve their performance.

In these meetings, we suggest that the discussion should remain focused on what needs to be improved, rather than who said what.

The implications related to the team contract include:

- The teachers and students in both courses were largely positive about the use of team contracts. We suggest organizing a workshop early in the course where teams are asked to develop their own team contracts with minimal necessary guidance from the teaching staff.
- The teachers and students both noticed that the team contracts were gradually forgotten after the initial few days. We suggest that teachers remind students to go back to and, if required, revise their team contracts. In addition, team mentors should also go back to the team contract when discussing team dynamics, disagreements, and professional ways of working. In a nutshell, the team contract should be a living document, and teaching staff should ensure that it is not completely forgotten.

## 4.8 Conclusion

In this study we present an experience of designing and using peer evaluation and team contracts as tools to improve teamwork in SE team project courses. The findings show that while peer evaluation and team contracts can promote accountability, reflection, and collaboration in these courses, their impact depends on team size, and structured follow-up. Peer evaluation was perceived more effective in small teams, whereas large teams reported limited value unless the results were shared and the process was adapted to the context of large teams. Similarly, team contracts helped set expectations in context of both courses, but often lost relevance without reinforcement. These findings suggest the need for both peer evaluation and team contracts to be actively revisited and adapted to course needs.

# 5 Collaborative Peer-Review Workshops with Paired Student Teams in Software Metrics Education

*Nayla Nasir, Muhammad Usman, and Umar Iftikhar*
*Accepted for publication in: 31st Annual ACM Conference on Innovation and Technology in Com- puter Science Education (ITiCSE), 2026.*

Abstract

Applying the Goal–Question–Metric (GQM) framework is quite challenging for students in software metrics courses, particularly when translating abstract goals into coherent GQM trees and conducting meaningful data analysis. This experience report describes the design, implementation, and perceived usefulness of two-stage collaborative peer-review workshops aimed at addressing these challenges. The workshops complement the existing course lectures by providing guided, collaborative, and feedback-rich learning activities. We analyzed student reflection reports from two course iterations. The results indicate that the workshops supported learning by application, improved understanding of the GQM framework, strengthened analytical skills, and promoted interactive and feedback-driven learning. We discuss how these findings align with the existing literature on collaborative learning and peer feedback in software engineering education.

## 5.1 Introduction

Collaborative learning (CL) is an educational approach that emphasizes active student engagement through peer interaction, team work, and shared problem-solving tasks [14]. It has been shown to improve the understanding of complex concepts [54], by encouraging discussion, exchange of perspectives, and collective interpretation of challenging material. In software engineering education (SEE), the use of collaborative learning activities has been explored by [14] to support students in learning concepts. Peer review [28] and peer feedback [52] is a form of collaborative learning . Peer feedback plays a central role by enabling students to articulate their reasoning, compare alternative solutions, and refine their understanding through dialogue [53]. Peer review encourages active learning and critical thinking. Prior work in computing and software engineering education suggests that peer review can improve conceptual understanding, support skill development, and increase student engagement, particularly when combined with scaffolding and instructor guidance [55].

In our software metrics course (for more details about the course setup, see Section 5.3), a key concept is the Goal Question-Metric (GQM) framework used in software metrics and quality measurement. Introduced by [125], GQM was developed as a measurement framework to assess progress in software development tasks and support project planning activities. It follows a goal-oriented process that involves defining specific goals, formulating questions to evaluate these goals, and identifying metrics to answer the questions.

The teaching team observed recurring difficulties over several years. Student teams struggled to (1) formulate relevant goal, questions, and metrics for a coherent GQM tree and (2) analyze and interpret collected measurement data in a meaningful way. These challenges persisted despite dedicated lectures covering measurement basics, including the GQM framework, GQM tree construction, and data collection and analysis techniques. This observation motivated us to introduce a collaborative intervention that would allow students to apply GQM concepts in a guided setting. We designed and iteratively refined collaborative peer-review workshops with paired student teams. In these workshops, student teams worked on cases provided by the teaching team, developed a GQM tree, performed initial analysis, and presented their solutions to peer teams. Mutual peer review and guided facilitation by the teaching team were central elements of the design. Students subsequently reflected on their learning experiences in written reflection reports.

This experience report shares our experience of introducing collaborative peer-review workshops in a software metrics course, and an analysis of students' reflections on the perceived learning benefits of these workshops. The objectives of this study are to (1) explore how collaborative peer-review workshops support learning in the context of GQM and software metrics, and (2) identify the perceived benefits and limitations of the intervention across iterations. The contributions of this paper are: (i) a detailed description of a collaborative peer-review intervention tailored to software metrics education, (ii) an empirical analysis of student reflections across two course iterations, and (iii) a discussion of how these findings relate to existing literature on collaborative learning and peer review in software engineering education.

## 5.2 Background and Related Work

Collaborative learning is an educational approach in which learners work together in groups to solve problems, accomplish tasks, or create projects, creating shared knowledge and interaction skills [38, 39].

Various pedagogical methods facilitate collaborative learning, including Team-Based Learning (TBL) [126, 127], Problem-Based Learning (PrBL) [128–130], Gamification [131, 132] and Collaborative peer review [28] and Peer feedback [52].

Collaborative learning plays a critical role in helping students grasp difficult concepts, as discussed by Bhola et al. [54]. The authors draw on multiple studies [133–135], including Vygotsky's scaffolding theory [136]. They argue that collaboration promotes deeper learning by encouraging students to reorganize knowledge, identify gaps, and integrate concepts through examples, analogies, and everyday experiences.

Previous studies [28, 52] suggest peer review and feedback as a collaborative learning strategy, rather than merely an assessment technique. Peer review is an activity in which learning emerges through social interaction, dialogue, and mutual knowledge construction [52, 53, 56]. Learning is therefore not unidirectional (from instructor to student), but distributed across peers. Peer review is a mechanism for student-centered learning [28, 56], emphasizing that students act simultaneously as knowledge producers, evaluators, and learners during the review process.

While collaborative problem solving and peer review have both been studied extensively, our work makes their combination explicit by structuring them as two sequential and complementary forms of collaboration within a single intervention.

## 5.3 Course Context

Software Metrics is an advanced-level course that is part of our Master's in SE program. The course consists of 7.5 ECTS and runs for one study period (about 10 Weeks). The course aims to equip students with theoretical knowledge and practical skills related to software measurement, including the application of the GQM framework to real-world scenarios. Prior to the intervention, the course included lectures on measurement foundations, GQM concepts, GQM tree development, and data analysis techniques.

The course contains three assessments: A review assignment (2 ECTS), a measurement project (4 ECTS), and a written exam (1.5 ECTS). The measurement project requires students, working in teams of four or five, to apply the GQM framework [125, 137] to evaluate a software quality attribute (e.g., maintainability) of an open-source system. The teaching team also had two progress meetings with the student teams to discuss their progress on course projects.

### 5.3.1 The GQM framework

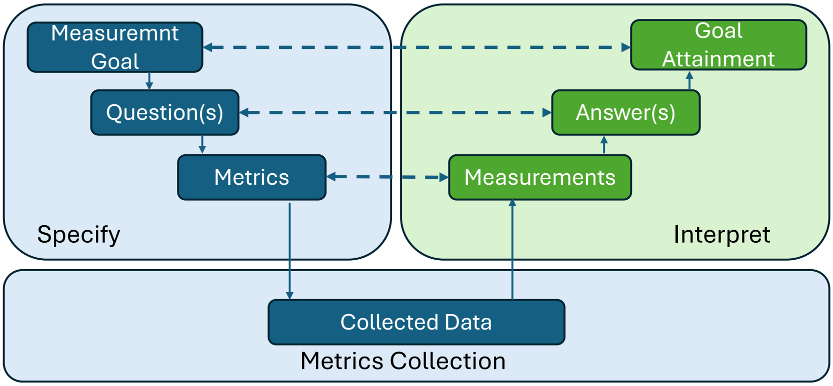



**Figure 5.1:** The GQM framework

The GQM framework [125, 137] has two phases, i.e., the GQM specification and GQM interpretation. The steps involved in these two phases are depicted in Figure 5.1 and can be summarized as:

#### *5.3.1.1 GQM specification*

The specification phase, involves defining measurement goals, formulating questions to evaluate the attainment of these goals, and identifying metrics to answer these questions.

1. Specify the measurement goal, i.e., the focus of the measurement task, to be achieved in a clear and concise manner with entities and attributes of interest
2. Define the questions that can help evaluate the attainment the outlined measurement goal.
3. Identify metrics that can help answer each of the defined questions .

#### *5.3.1.2 GQM Interpretation*

In the interpretation phase of the GQM framework [138], the collected measurement data is analyzed using the defined metrics to answer the specified questions. These answers, in turn, help assess whether the overall goals have been achieved.

3. Synthesize results from the questions to help answer the goal.

### 5.3.2 Identified Challenges

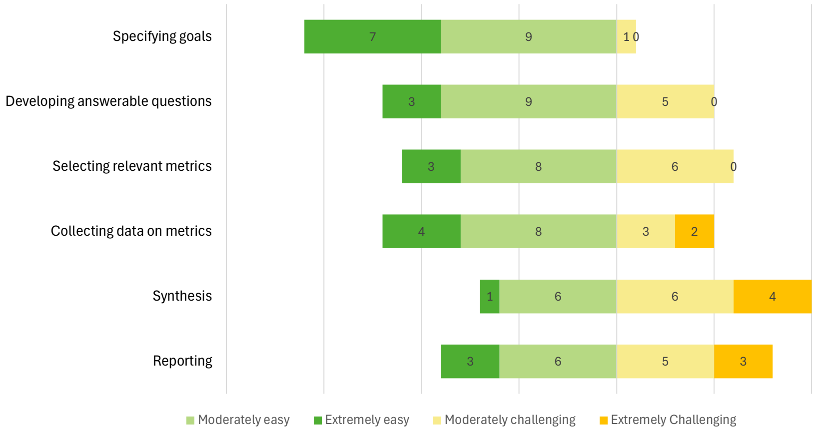



**Figure 5.2:** Survey results on student challenges with the GQM process

Despite lectures on all key concepts, the teaching team consistently observed that student teams encountered difficulties in applying GQM in practice. Common issues included vague or misaligned goals, weak connections between questions and metrics, and superficial or incorrect data analysis. These challenges often became apparent during the progress meetings and discussions between teachers and student teams. The evaluation of student project reports also indicated that students frequently struggled with GQM specification and its interpretation. In 2023, a GQM specification workshop was introduced followed by a student survey towards the end of the course to analyze what aspects of the GQM were considered difficult by the students. There were 17 responses to the survey, and the results (See Figure 5.2 ) show that most of the students struggled with the synthesis and reporting part of the GQM.

## 5.4 Intervention Design

To address the identified challenges, we introduced two collaborative peer-review workshops in two recent iterations of the course. The first workshop focused on supporting students in specifying the GQM tree (see Figure 5.1), while the second work-

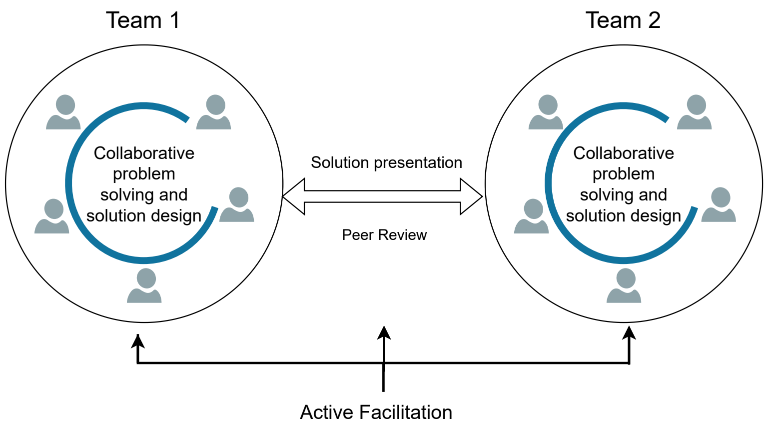



**Figure 5.3:** The two stage collaborative design of the workshops

shop aimed to support students in synthesizing metrics data to answer measurement questions and assess the attainment of measurement goals. Both workshops had a duration of 2 hours each and were designed around the following key characteristics:

- **Active Facilitation:** Workshops were conducted in the presence of the teaching team, allowing for real-time guidance and clarification.

- **Case–based exercises:** Each workshop required students to work on realistic software measurement scenarios. In the first workshop, two case scenarios were provided, each representing a different measurement context. Student teams were assigned to cases in an alternating manner (e.g., Team 1 to Case 1, Team 2 to Case 2). In the second workshop, all teams worked on a single case consisting of an open-source dataset with software metrics data, which they analyzed and synthesized using the GQM framework.

- **Two-stage collaboration:** The workshops were designed to support collaboration at two levels : i) intra-team collaboration, in which students first collaborated within their own teams to develop and discuss a solution for the assigned case , and ii) inter-team peer review, in which student teams were paired sequentially (e.g., Team 1 with Team 2). Each team presented its solution to the paired team and received structured peer feedback on their solutions (see Figure 5.3).

- **Reflection:** Student teams reflected on their learning experiences and the perceived usefulness of the workshops as part of their final project reports.

**Table 5.1:** Design of the GQM Specification and GQM Interpretation Workshops

| Aspect | GQM Specification Workshop | GQM Interpretation Workshop |
|---|---|---|
| **Objective** | Apply the GQM framework to build a GQM tree for a real-life case. | Understand the interpretation phase of GQM by synthesizing existing metrics data to answer specific questions. |
| **Learning Outcome** | Ability to construct a GQM tree and critically evaluate metrics. | Ability to analyze existing data and synthesize insights using GQM. |
| **Pre-session Preparation** | There was a lecture on GQM specification before the workshop. The purpose, intent and the materials (i.e., the case scenarios and templates) to be used during the workshop were made available to the students well before the workshop. | The purpose, intent and the materials (i.e., the case scenario and templates) to be used during the workshop were made available to the students well before the workshop. Some pre-readings on GQM framework and descriptive statistics were also suggested. |
| **Structure & Activities** | 1. Introduction & Briefing (15 mins)<br>2. Group Work (40 mins)<br>3. Cross team discussions (45 mins)<br>4. Refinement & Submission (15 mins) | 1. Introduction & Briefing (15 mins)<br>2. Group Work (50 mins)<br>3. Cross-team Discussion (45 mins)<br>4. Refinement & Submission (10 mins) |
| **Skill Emphasis** | Problem-solving, critical thinking, and GQM framework design. | Data analysis, descriptive statistics, evidence-based reasoning. |
| **Case Context** | Case 1: A medium-sized software company wants to assess whether increasing the required number of code reviewers from one to two has improved code quality. The company would analyze available data from before and after the change was introduced to the code review process.<br><br>Case 2: A company wants to measure and compare the performance of two agile teams of same size, working on comparable system components in terms of complexity. They would analyze data on teams performance in development and testing. | Case: A company wants to identify hard-to-maintain modules in its long-evolving Java system (System A). The company has collected data on metrics related to size, structure, comments, and complexity across multiple versions of its 20 modules. This data was shared with the students to analyze and interpret to answer the specified goal. |
| **Team tasks** | Each team builds a GQM tree, select metrics, justify choices, and identify limitations of the selected metrics. | Each team analyzes metrics data, apply descriptive statistics, and derive answers to the GQM questions leading to insights for the GQM goal. |
| **Cross team peer review** | Both teams take turns to present their solutions to the peer team and provide feedback on peer team's solution | Both teams take turns to present their solutions to the peer team and provide feedback on peer team's solution |
| **Active Facilitation** | 1. Teachers acted as facilitators for the teamwork.<br>2. A potential solution was presented and discussed towards the end. | 1. Teachers acted as facilitators for the teamwork.<br>2. A potential solution was presented and discussed towards the end. |

Two iterations of the workshops were implemented in consecutive course iterations (2024 and 2025), with minor refinements based on teaching team observations. The detailed design of the two workshops is presented in Table 5.1.

## 5.5 Methodology

### 5.5.1 Data Collection

The primary data source for this study are the final project reports [1] submitted by the student teams. The reports were submitted by the teams towards the end of the project, and also included student reflections on their learning from the two workshops. A total of 23 reflections were analyzed: 10 from the 2024 iteration and 13 from the 2025 iteration of the course.

### 5.5.2 Data Analysis

We conducted a thematic analysis on student reflections following Braun and Clarke's [60] approach. The analysis involved familiarization with the data, initial coding, theme identification, theme review, and refinement. Codes captured perceived benefits and limitations of the workshops. Frequencies of codes were recorded to support comparison across iterations, while acknowledging that the analysis remains qualitative in nature.

## 5.6 Results

**Table 5.2:** Coding of student reflections on the collaborative learning workshops across the two course iterations. $n$ denotes the number of team reports analyzed.

| Student reflection | v2024 ($n = 10$) | v2025 ($n = 13$) | Total ($n = 23$) |
|---|---|---|---|
| Allowed learning by application | 8 | 11 | 19 |
| Improved understanding of GQM framework | 8 | 9 | 17 |
| Promoted skills for analysis | 6 | 11 | 17 |
| Promoted collaboration within the team | 4 | 7 | 11 |
| Provided foundation for project progress | 5 | 4 | 9 |
| Provided knowledge and confidence to apply GQM in the project | 3 | 6 | 9 |
| Provided guidance on data visualizations | 3 | 5 | 8 |
| Overall usefulness for project | 1 | 6 | 7 |
| Allowed feedback driven improvement | 1 | 6 | 7 |
| Allowed different analytical perspectives | 2 | 4 | 6 |
| Provided real world relevance | 2 | 3 | 5 |
| Provided clarification of doubts | 3 | 0 | 3 |
| Promoted critical thinking | 0 | 3 | 3 |
| Gap in the example and the project scenario | 0 | 2 | 2 |

We coded reflection reports from two course itrations (v2024 and v2025). The thematic analysis revealed a range of perceived benefits, with several themes appear-

[1] Student submissions are official public documents in our country, and may be disclosed upon request, subject to a confidentiality assessment. For this study, all personal identifiers were replaced with fictitious IDs, and the data was handled in accordance with GDPR and institutional procedures.

ing across both iterations. Table 5.2 summarizes the the identified codes and their frequencies.

The most frequently reported benefit across both cohorts was learning by application (19/23), with students emphasizing hands-on activities such as constructing GQM trees and analyzing real datasets. One team reflected: *"The workshops helped us apply our theoretical knowledge in practical scenarios..."*. Similarly another team stated: *"Both workshops were really helpful and gave us a better idea of how to use the GQM framework instead of just reading about it."* A majority of students reported an improved understanding of the GQM framework (17/23). According to one student team: *"The two GQM workshops were very helpful in understanding the concept and application of the GQM (Goal-Question-Metric) framework in a practical context."* Students also found them helpful in promoting their analytical skills (17/23). One of the student team reflected *"The structured approach for the workshops allowed us to derive meaningful data-driven insights which are crucial for making informed decisions in software development."*

Students appreciated the idea of collaborative learning within the team (11/23). One student group reflected: *"Thanks to these workshops, we not only got better results, but we also learned a lot about teamwork...*", whereas another group stated: *"The workshops provided a collaborative environment to exchange views, ideas, clarify doubts and explore how to apply the GQM framework to a real-world case"*.

Students also perceived that timely sequenced workshops helped to set a foundation for their project progress (9/23). Students appreciated the head-start the workshops provided to their project progress. One student team wrote: *"We were able to quickly develop our GQM tree within the first week of receiving the project description because of the first workshop which gave us a headstart for the rest of the project"*.

Knowledge and confidence to apply GQM in the project was also reported as a perceived benefit of the workshops (9/23). One student team reflected: *"Both workshops equipped us with the knowledge and confidence to apply the GQM framework effectively in this project."* Another team stated: *"Collaborative activities and cross-group discussions during both workshops greatly improved our group's confidence and skill in effectively applying the GQM framework to our project."*

Students also appreciated the guidance on data visualization (8/23), suggesting that workshops supported not only conceptual understanding but also practical execution of analysis tasks. according to one team: *"The workshops' emphasis on data analysis techniques and descriptive statistics directly enabled our results interpretation"*. Another team mentioned this idea by stating: *"The workshops provided us with practical techniques for using statistical and visual tools to draw meaningful conclusions from our results"*.

The workshops had an overall usefulness for student projects (7/23). One student team stated: *"These workshops collectively allowed us to apply GQM confidently in our project, right from goal definition to metric-based evaluation"*.

Students appreciated the opportunity of having a cross team collaborative peer review, as it provided them feedback-driven improvement (7/23). One team highlighted: *"Presenting our results to other groups was particularly valuable, providing immediate feedback."* Another group stated: *"Presenting our GQM tree to another group and receiving their feedback helped refine our thinking. Also, feedback from the teachers was particularly valuable. It helped us identify the logical gaps in our trees".*

Another notable benefit of collaborative peer review was an exposure to different analytical perspectives (6/23). One team mentioned: *"Discussing alternative metric selections, receiving questions on our design, and defending our choices sharpened our ability to think critically about metric selection, potential redundancies, and gaps in measurement coverage."* Another team reflected: *"Presenting to another team and receiving their suggestions helped us view our work from a different perspective and improve our analysis as well."*

A few students also appreciated the real-world relevance (5/23) of the workshops, indicating that some students explicitly recognized the value of working with realistic systems. One team reflected *"Scenarios given in workshop helped us to see beyond the theory and understand how goals, questions, and metrics map onto actual software engineering challenges."* Codes related to higher cognitive engagement, such as critical thinking (3/23) appeared only in the 2025 cohort.

Both cohorts reported broadly similar categories of benefits, however, the 2025 cohort showed higher frequencies for most benefit codes, particularly the ones associated with analytical skills, feedback, and critical thinking. The limitations reported were few. Two students in the 2025 cohort were able to identify the limitation of workshops in presenting a case complex enough to simulate the complexity of the real software system they had to work with for their projects.

## 5.7 Discussion

The findings of this experience report align with existing literature that positions collaborative learning [14] and peer review [28, 52] as effective approaches for supporting learning in computing education.

Across both iterations of the intervention, students most frequently reported learning through application, or learning by doing [139]. This suggests that the workshops helped bridge the gap between conceptual instruction and practical application.

The intra-group collaboration in the first stage of the workshops, where students collaboratively developed a solution to the given case, aligns with the suggestions of Webb and Palinscar [140] on collaborative peer-work groups. In this stage, students worked together to co-construct understanding, engaged in explaining, questioning, and reasoning, and received appropriate teacher support. Working in teams allowed students to discuss, negotiate, and refine their understanding of GQM concepts, supporting understanding around ideas that students had previously found challenging. Several students explicitly noted that the workshops helped them better understand their project work, particularly by providing them with an opportunity to exchange ideas, clarify doubts, and to develop their analytical skills techniques, which they later in their projects. This finding is consistent with Bhola and Parchoma [54], who report that collaborative, problem-based learning environments can help students overcome conceptual difficulties by grounding abstract ideas in shared activity.

By engaging with peer solutions during the review stage, students were exposed to alternative analytical perspectives and solution approaches. This aligns with findings by Serrano et al. [28], who argue that peer review can function not only as an assessment mechanism but also as a learning activity that promotes reflection and conceptual clarification. In our workshop settings, students were positioned as active learning agents who negotiate meaning through dialogue with peers, an approach that is increasingly recommended in recent literature on learning and feedback [53, 56, 141]. Such dialogue based engagement encourages students to reason about why a solution is appropriate rather than focusing solely on what the solution is.

The interactive and collaborative nature of the workshops also appeared to support student engagement and confidence. Students' reflections indicate that discussion with peers, combined with immediate clarification from the teaching team, played an important role in their learning experience. This reinforces earlier work showing that peer-review activities are most effective when carried out in structured and facilitated settings, rather than implemented as isolated or purely student-driven exercises [28, 55].

Students also reported increased confidence in data analysis and interpretation, suggesting that peer review supported a shift from mechanical metric computation towards more meaningful analysis. This observation aligns with prior research highlighting that peer review develops an evaluation-oriented mindset [28], which is particularly important in software engineering contexts where judgment and interpretation are central. By actively engaging in analysis, evaluation, and explanation, students moved beyond passive reception of content [28, 56].

Several of the reported benefits can be attributed to the two-stage collaborative design of the workshops. Collaboration within teams supported solution development, conceptual understanding, and communication skills, while peer review across teams enabled exposure to different analytical perspectives, feedback-driven improvement, and critical reflection. This structure aligns with prior work emphasizing both learning by doing [139] and learning by reviewing [28, 52], and extends it by demonstrating how these mechanisms can be sequenced within a single pedagogical intervention.

Finally, the limited number of reported limitations suggests that the intervention was largely well received. The identified gap between workshop cases and project scenarios highlights an opportunity for refinement. Closer alignment between workshop cases and project contexts may further strengthen the transfer of learning.

## 5.8 Conclusions

This experience report presented the design and implementation of two iterations of collaborative peer-review workshops introduced into a software metrics course to support students in applying the GQM framework. Motivated by recurring difficulties in GQM specification and data interpretation, the workshops complemented existing lectures by providing guided, case-based, and collaborative learning opportunities focused on both the specification and interpretation phases of GQM.

An analysis of student reflection reports using thematic analysis suggests that the workshops were perceived as beneficial. The two stage collaborative design of the workshops facilitated learning at two levels. In the first stage, the students worked together in their teams to construct a solution to the given case, allowing them to learn by application, improve understanding of the GQM framework, and develop analytical and critical thinking skills. In the second stage, students compared their solutions with those of peers, received feedback, and discussed alternative perspectives in a structured setting. The findings also indicate that the workshops supported project readiness by increasing students' confidence in applying GQM to realistic measurement scenarios.

These results align with existing literature on peer review as a collaborative learning technique and extend prior work by illustrating how collaborative problem solving within teams can be sequentially combined with cross-team peer review within a single instructional intervention. While the study is limited by its reliance on student reflections and a single course context, it provides practical insights into how peer-review-based workshops can be integrated into software engineering courses to address persistent learning challenges.

# Bibliography


[1] V. Larivière, D. Pontille, and C. R. Sugimoto. "Investigating the division of scientific labor using the Contributor Roles Taxonomy (CRediT)". In: *Quantitative science studies* 2.1 (2021), pp. 111–128.

[2] A.-M. Majanoja and T. Vasankari. "Reflections on Teaching Software Engineering Capstone Course". In: *Proceedings of the 10th International Conference on Computer Supported Education*. 2018, pp. 68–77.

[3] M. Paasivaara, D. Vodă, V. T. Heikkilä, J. Vanhanen, and C. Lassenius. "How does participating in a capstone project with industrial customers affect student attitudes?" In: *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*. 2018, pp. 49–57.

[4] O. Cico, L. Jaccheri, A. Nguyen-Duc, and H. Zhang. "Exploring the intersection between software industry and Software Engineering education-A systematic mapping of Software Engineering Trends". In: *Journal of Systems and Software* 172 (2021), p. 110736.

[5] S. Tenhunen, T. Männistö, M. Luukkainen, and P. Ihantola. "A systematic literature review of capstone courses in software engineering". In: *Information and Software Technology* (2023), p. 107191.

[6] A. D. Suleiman, D. Hou, Y. Liu, J. DeWaters, D. C. Shepherd, and J. G. De Souza. "Systematic literature review on project-based learning in computing education". In: *ACM Transactions on Computing Education* 25.4 (2025), pp. 1–53.

[7] ACM/IEEE. *Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. `https://www.acm.org/binaries/content/assets/education/se2014.pdf`. [Online; accessed 3-Oct-2023]. 2015.

[8] R. Adcock, E. Alef, B. Amato, M. Ardis, L. Bernstein, B. Boehm, P. Bourque, J. Brackett, M. Cantor, L. Cassel, et al. *Curriculum guidelines for graduate degree programs in software engineering*. ACM, 2009.

[9] M. Ardis, D. Budgen, G. W. Hislop, J. Offutt, M. Sebern, and W. Visser. "SE 2014: Curriculum guidelines for undergraduate degree programs in software engineering". In: *Computer* 48.11 (2015), pp. 106–109.

[10] D. Ståhl, K. Sandahl, and L. Buffoni. "An eco-system approach to project-based learning in software engineering education". In: *IEEE Transactions on Education* 65.4 (2022), pp. 514–523.

[11] D. Shoham, R. Paul, and M. Moshirpour. "Student perceptions of project-based learning in a software engineering course". In: *Proceedings of the 16th International CDIO Conference*. 2020, pp. 268–279.

[12] B. Bruegge, S. Krusche, and L. Alperowitz. "Software engineering project courses with industrial clients". In: *ACM Transactions on Computing Education* 15.4 (2015), pp. 1–31.

[13] S. Fincher, M. Petre, and M. Clark. *Computer science project work: principles and pragmatics*. London, UK: Springer Science & Business Media, 2001.

[14] R. Garcia, C. Treude, and A. Valentine. "Application of Collaborative Learning Paradigms within Software Engineering Education: A Systematic Mapping Study". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. 2024, pp. 366–372.

[15] G. Gabrysiak, M. Guentert, R. Hebig, and H. Giese. "Teaching requirements engineering with authentic stakeholders: Towards a scalable course setting". In: *2012 First International Workshop on Software Engineering Education Based on Real-World Experiences (EduRex)*. IEEE. 2012, pp. 1–4.

[16] M. C. Bastarrica, D. Perovich, and M. M. Samary. "What can students get from a software engineering capstone course?" In: *Proceedings of the International Conference on Software Engineering: Software Engineering Education and Training Track*. 2017, pp. 137–145.

[17] J. Khakurel and J. Porras. "The effect of real-world capstone project in an acquisition of soft skills among software engineering students". In: *Proceedings of the 32nd Conference on Software Engineering Education and Training*. 2020, pp. 1–9.

[18] M. Spichkova. "Industry-oriented project-based learning of software engineering". In: *Proceedings of the 24th International Conference on Engineering of Complex Computer Systems*. 2019, pp. 51–60.

[19] M. Paasivaara, J. Vanhanen, and C. Lassenius. "Collaborating with industrial customers in a capstone project course: the customers' perspective". In: *Proceedings of the 41st International Conference on Software Engineering: Software Engineering Education and Training*. 2019, pp. 12–22.

[20] M. R. H. bin Mohamed Riduwan, M. P. J. Zen, O. J. W. Darien, N. J. Hong, C. R. Loh, O. S. Ya, Q. Cao, and P. C. Yau. "Exploring Peer Evaluation Methods in Group Projects of Software Engineering Education". In: *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2024, pp. 1–6.

[21] M. Vasilevskaya, D. Broman, and K. Sandahl. "Assessing large-project courses: model, activities, and lessons learned". In: *ACM Transactions on Computing Education (TOCE)* 15.4 (2015), pp. 1–30.

[22] M. Minish, M. Galster, and F. Gilson. "Think Before You Scrum"-An Experience Report on Daily Scrums to Aid Reflection". In: *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*. IEEE. 2024, pp. 1–10.

[23] C. D. Hundhausen, P. T. Conrad, A. S. Carter, and O. Adesope. "Assessing individual contributions to software engineering projects: a replication study". In: *Computer Science Education* 32.3 (2022), pp. 335–354.

[24] A. Basholli, F. Baxhaku, D. Dranidis, and T. Hatziapostolou. "Fair assessment in software engineering capstone projects". In: *Proceedings of the 6th Balkan Conference in Informatics*. 2013, pp. 244–250.

[25] I. Fronza and X. Wang. "Towards an approach to prevent social loafing in software development teams". In: *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE. 2017, pp. 241–246.

[26] M. L. Pertegal-Felices, A. Fuster-Guilló, M. L. Rico-Soliveres, J. Azorín-López, and A. Jimeno-Morenilla. "Practical method of improving the teamwork of engineering students using team contracts to minimize conflict situations". In: *IEEE Access* 7 (2019), pp. 65083–65092.

[27] I. Fronza and X. Wang. "Social loafing prevention in agile software development teams using team expectations agreements". In: *IET Software* 15.3 (2021), pp. 214–229.

[28] J. J. Serrano-Aguilera, A. Tocino, S. Fortes, C. Martín, P. Mercadé-Melé, R. Moreno-Sáez, A. Muñoz, S. Palomo-Hierro, and A. Torres. "Using peer review for student performance enhancement: Experiences in a multidisciplinary higher education setting". In: *Education Sciences* 11.2 (2021), p. 71.

[29] V. Garousi, G. Giray, E. Tüzün, C. Catal, and M. Felderer. "Aligning software engineering education with industrial needs: A meta-analysis". In: *Journal of Systems and Software* 156 (2019), pp. 65–83.

[30] D. Klappholz, V. L. Almstrum, K. Modesit, C. Owen, and A. Johnson. “A framework for success in real projects for real clients courses”. In: *Software Engineering: Effective Teaching and Learning Approaches and Practices*. Ed. by H. J. Ellis, S. A. Demurjian, and J. F. Naveda. Hershey, PA, USA: IGI Global, 2009, pp. 157–190.

[31] G. G. Borges and R. C. G. de Souza. “Skills development for software engineers: Systematic literature review”. In: *Information and Software Technology* 168 (2024), p. 107395.

[32] D. Akdur. “Analysis of software engineering skills gap in the industry”. In: *ACM Transactions on Computing Education* 23.1 (2022), pp. 1–28.

[33] J. Börstler and T. B. Hilburn. “Team projects in computing education”. In: *ACM Transactions on Computing Education* 15.4 (2015). Editorial of a special issue., pp. 1–5.

[34] J. Börstler and T. B. Hilburn. “Team projects in computing education”. In: *ACM Transactions on Computing Education (TOCE)* 16.2 (2016), pp. 1–4.

[35] J. Chen, A. Kolmos, and X. Du. “Forms of implementation and challenges of PBL in engineering education: a review of literature”. In: *European Journal of Engineering Education* 46.1 (2021), pp. 90–115.

[36] B. Brau. “Constructivism”. In: *The Student Handbook of Instructional Psychology & Technology* (2016).

[37] D. Kokotsaki, V. Menzies, and A. Wiggins. “Project-based learning: A review of the literature”. In: *Improving schools* 19.3 (2016), pp. 267–277.

[38] M. Á. Herrera-Pavo. “Collaborative learning for virtual higher education”. In: *Learning, Culture and Social Interaction* 28 (2021), p. 100437. DOI: `10.1016/J.LCSI.2020.100437`.

[39] M. Laal and S. Ghodsi. “Benefits of collaborative learning”. In: *Procedia - Social and Behavioral Sciences* 31 (2012), pp. 486–490. DOI: `10.1016/J.SBSPRO.2011.12.091`.

[40] J. Vanhanen, T. O. Lehtinen, and C. Lassenius. “Teaching real-world software engineering through a capstone project course with industrial customers”. In: *Proceedings of the First International Workshop on Software Engineering Education Based on Real-World Experiences*. 2012, pp. 29–32.

[41] C. Raibulet and F. A. Fontana. “Collaborative and teamwork software development in an undergraduate software engineering course”. In: *Journal of Systems and Software* 144 (2018), pp. 409–422.

[42] A. Barman. “Effectiveness of Implementing Team Contracts in a Client-Sponsored Project-Based Learning Course”. In: *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2024, pp. 1–8.

[43] E. Tavoletti, R. D. Stephens, and L. Dong. “The impact of peer evaluation on team effort, productivity, motivation and performance in global virtual teams”. In: *Team Performance Management: An International Journal* 25.5/6 (2019), pp. 334–347.

[44] R. Schorr. “Experience Report on Key Success Factors for Promoting Students’ Engagement in Software Development Group Projects”. In: *2020 IEEE World Conference on Engineering Education (EDUNINE)*. IEEE. 2020, pp. 1–5.

[45] J. R. Engelsma. “Best practices for industry-sponsored CS capstone courses”. In: *Journal of Computing Sciences in Colleges* 30.1 (2014), pp. 18–28.

[46] M. Minish, F. Gilson, and M. Galster. “Scrumboard: A project management tool purpose-built for software engineering education”. In: *2024 36th International Conference on Software Engineering Education and Training (CSEE&T)*. IEEE. 2024, pp. 1–5.

[47] D. J. Domke-Damonte and J. K. Keels. “The effect of shared versus individual reflection on team outcomes”. In: *Business and Professional Communication Quarterly* 78.1 (2015), pp. 5–32.

[48] H. M. Haddad. “One-semester CS capstone: A 40-60 teaching approach”. In: *2013 10th International Conference on Information Technology: New Generations*. IEEE. 2013, pp. 97–102.

[49] T. P. Rout and J. Seagrott. “Maintaining high process capability in a student project course”. In: *20th Conference on Software Engineering Education & Training (CSEET’07)*. IEEE. 2007, pp. 37–44.

[50] Z. Liang and M. A. Chapa-Martell. “A Top-Down Approach to Teaching Web Development in the Cloud”. In: *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. IEEE. 2018, pp. 32–39.

[51] A. Decker, C. A. Egert, and A. Phelps. “Splat! er, shmup? A postmortem on a capstone production experience”. In: *2016 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2016, pp. 1–9.

[52] J. S. Tan and W. Chen. “Peer feedback to support collaborative knowledge improvement: What kind of feedback feed-forward?” In: *Computers & Education* 187 (2022), p. 104467.

[53] Q. Zhu and D. Carless. “Dialogue within peer feedback processes: Clarification and negotiation of meaning”. In: *Higher Education Research & Development* 37.4 (2018), pp. 883–897.

[54] S. Bhola and G. Parchoma. “Threshold concepts and conceptual change processes”. In: *Papers on Postsecondary Learning and Teaching* 1 (2016), pp. 33–40.

[55] K. Falkner and N. J. Falkner. "Supporting and structuring "contributing student pedagogy" in computer science curricula". In: *Computer Science Education* 22.4 (2012), pp. 413–443.

[56] E. Er, Y. Dimitriadis, and D. Gašević. "A collaborative learning approach to dialogic peer feedback: a theoretical framework". In: *Assessment & Evaluation in Higher Education* 46.4 (2021), pp. 586–600.

[57] H. Morgan. "Conducting a qualitative document analysis". In: *The Qualitative Report* 27.1 (2022), pp. 64–77.

[58] P. Runeson and M. Höst. "Guidelines for conducting and reporting case study research in software engineering". In: *Empirical software engineering* 14.2 (2009), p. 131.

[59] C. B. Seaman. "Qualitative methods in empirical studies of software engineering". In: *IEEE Transactions on software engineering* 25.4 (1999), pp. 557–572.

[60] V. Braun and V. Clarke. "Using thematic analysis in psychology Qualitative Research in Psychology, 3 (2), 77-101". In: *Online im Internet* (2006), pp. 1–42.

[61] D. S. Cruzes and T. Dyba. "Recommended steps for thematic synthesis in software engineering". In: *Proceedings of the International Symposium on Empirical Software Engineering and Measurement*. 2011, pp. 275–284.

[62] D. Byrne. "A worked example of Braun and Clarke's approach to reflexive thematic analysis". In: *Quality & Quantity* 56.3 (2022), pp. 1391–1412.

[63] J. Maxwell. "Understanding and validity in qualitative research". In: *Harvard educational review* 62.3 (1992), pp. 279–301.

[64] K. Petersen and C. Gencel. "Worldviews, research methods, and their relationship to validity in empirical software engineering research". In: *Proceedings of the 23rd international workshop on software measurement and the 8th international conference on software process and product measurement*. 2013, pp. 81–89.

[65] J.-P. Steghöfer, H. Burden, R. Hebig, G. Calikli, R. Feldt, I. Hammouda, J. Horkoff, E. Knauss, and G. Liebel. "Involving external stakeholders in project courses". In: *ACM Transactions on Computing Education (TOCE)* 18.2 (2018), pp. 1–32.

[66] Y. H. R. Sim, Z. Z. Lua, K. K. Kelaver, J. Q. Chua, I. Z. J. Lim, Q. Cao, S. L. Keoh, and L. H. I. Lim. "Experiences and Lessons Learned from Real-World Projects in Software Engineering Subject". In: *Proceedings of the 35th International Conference on Software Engineering Education and Training*. 2023, pp. 173–177.

[67] C. Murphy, S. Sheth, and S. Morton. “A two-course sequence of real projects for real customers”. In: *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*. 2017, pp. 417–422.

[68] T. Dybå, N. Maiden, and R. Glass. “The reflective software engineer: reflective practice”. In: *IEEE software* 31.4 (2014), pp. 32–36.

[69] G. A. Bowen. “Document analysis as a qualitative research method”. In: *Qualitative Research Journal* 9.2 (2009), pp. 27–40.

[70] N. B. Moe, V. Stray, D. Šmite, and M. Mikalsen. “Attractive Workplaces: What are Engineers Looking For?” In: *IEEE Software* 40.5 (2023), pp. 85–93.

[71] R. de Souza Santos, W. D. N. Grillo, D. Cabral, C. De Castro, N. Albuquerque, and C. França. “Post-Pandemic Hybrid Work in Software Companies: Findings from an Industrial Case Study”. In: *Proceedings of the 2024 IEEE/ACM 17th International Conference on Cooperative and Human Aspects of Software Engineering*. 2024, pp. 68–78.

[72] D. Šmite, N. B. Moe, E. Klotins, and J. Gonzalez-Huerta. “From forced Working-From-Home to voluntary working-from-anywhere: Two revolutions in telework”. In: *Journal of Systems and Software* 195 (2023), p. 111509.

[73] R. E. de Souza Santos and P. Ralph. “A grounded theory of coordination in remote-first and hybrid software teams”. In: *Proceedings of the 44th International Conference on Software Engineering*. 2022, pp. 25–35.

[74] P. Lago, P. Runeson, Q. Song, and R. Verdecchia. “Threats to Validity in Software Engineering–hypocritical paper section or essential analysis?” In: *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. 2024, pp. 314–324.

[75] N. Nasir. “Towards Better Design and Delivery of SE Team Project Courses”. In: *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 2*. 2024, pp. 828–829.

[76] N. Nasir, M. Usman, J. Börstler, and N. D. Fogelström. “Software engineering team project courses with industrial customers: Students’ insights on challenges and lessons learned”. In: *Journal of Systems and Software* 226 (2025), p. 112441.

[77] J. Tan and M. Jones. “A case study of classroom experience with client-based team projects”. In: *Journal of Computing Sciences in Colleges* 23.5 (2008), pp. 150–159.

[78] C. C. Tappert and A. Stix. “A Decade Review of a Masters-Level Real-World-Projects Capstone Course”. In: *The Joint Proceedings of Information Systems Education Conference (ISECON 2011)*. ISECON. Philadelphia, PA, 2011, pp. 1–10.

[79] M. E. Morales-Trujillo, M. Galster, F. Gilson, and M. Mathews. “A three-year study on peer evaluation in a software engineering project course”. In: *IEEE Transactions on Education* 65.3 (2021), pp. 409–418.

[80] A. Tafliovich, F. Estrada, and T. Caswell. “Teaching software engineering with free open source software development: An experience report”. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 2019, pp. 7731–7741.

[81] D. Tee and P. Ahmed. “360 degree feedback: an integrative framework for learning and assessment”. In: *Teaching in Higher Education* 19 (2014), pp. 579–591. DOI: 10.1080/13562517.2014.901961.

[82] P. A. Laplante, J. F. Defranco, and E. Guimaraes. “Evolution of a graduate software engineering capstone course—a course review”. In: *International Journal of Engineering Education* 35.4 (2019), pp. 999–1007.

[83] A. J. Magana, Y. Y. Seah, and P. Thomas. “Fostering cooperative learning with Scrum in a semi-capstone systems analysis and design course”. In: *Journal of Information Systems Education* 29.2 (2018), pp. 75–92.

[84] R. Dupuis, R. Champagne, A. April, and N. Séguin. “Experiments with adding to the experience that can be acquired from software courses”. In: *2010 Seventh International Conference on the Quality of Information and Communications Technology*. IEEE. 2010, pp. 1–6.

[85] I. Tadros, S. Hammami, and K. Al-Zoubi. “Systems Development Projects”. In: *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*. IEEE. 2008, pp. 1–7.

[86] S. J. Zilora. “Industry-Emulated Projects in the Classroom”. In: *Proceedings of the 16th Annual Conference on Information Technology Education*. 2015, pp. 115–119.

[87] J. A. Carvalho, R. D. Sousa, and J. O. Sá. “Information systems development course: Integrating business, IT and IS competencies”. In: *2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*. IEEE. 2010, pp. 1–20.

[88] R. J. Fornaro, M. R. Heil, and A. L. Tharp. “Reflections on 10 years of sponsored senior design projects: Students win–clients win!” In: *Journal of Systems and Software* 80.8 (2007), pp. 1209–1216.

[89] T. Tenório, I. I. Bittencourt, S. Isotani, and A. P. Silva. “Does peer assessment in on-line learning environments work? A systematic review of the literature”. In: *Computers in Human Behavior* 64 (2016), pp. 94–107.

[90] Q. Brown, F. Lee, and S. Alejandre. “Emphasizing soft skills and team development in an educational digital game design course”. In: *Proceedings of the 4th international Conference on Foundations of Digital Games*. 2009, pp. 240–247.

[91] B. Hui. “Guiding Principles for Assessing Software Engineering Teams”. In: *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2024, pp. 1–9.

[92] Q. Le, K. Phan, B. Hui, and A. Putri. “Using GitHub Analytics to Assess the Quality of Collaboration in Software Engineering Teams”. In: *2024 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2024, pp. 1–9.

[93] H. Ziv and S. Patil. “Capstone project: From software engineering to “Informatics””. In: *2010 23rd IEEE Conference on Software Engineering Education and Training*. IEEE. 2010, pp. 185–188.

[94] M. Marques, S. F. Ochoa, M. C. Bastarrica, and F. J. Gutierrez. “Enhancing the student learning experience in software engineering project courses”. In: *IEEE Transactions on Education* 61.1 (2017), pp. 63–73.

[95] F. Fagerholm and A. Vihavainen. “Peer assessment in experiential learning assessing tacit and explicit skills in agile software engineering capstone projects”. In: *2013 IEEE frontiers in education conference (FIE)*. IEEE. 2013, pp. 1723–1729.

[96] D. M. Nguyen, T. V. Truong, and N. B. Le. “Deployment of capstone projects in software engineering education at duy tan university as part of a university-wide project-based learning effort”. In: *2013 Learning and Teaching in Computing and Engineering*. IEEE. 2013, pp. 184–191.

[97] S. A. Rukmono and M. R. Chaudron. “Guiding peer-feedback in learning software design using UML”. In: *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Software Engineering Education and Training*. 2022, pp. 122–133.

[98] M. Verdicchio. “Hurricanes and pandemics: An experience report on adapting software engineering courses to ensure continuity of instruction”. In: *Journal of Computing Sciences in Colleges* 36.5 (2021), pp. 150–159.

[99] K. Z. Watkins and T. Barnes. “Competitive and agile software engineering education”. In: *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*. IEEE. 2010, pp. 111–114.

[100] S. Koolmanojwong and B. Boehm. “A look at software engineering risks in a team project course”. In: *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*. IEEE. 2013, pp. 21–30.

[101] T. Harding. “Benefits And Struggles Of Using Large Team Projects In Capstone Courses”. In: *2007 Annual Conference & Exposition*. 2007, pp. 12–304.

[102] M. Usman, R. Britto, J. Börstler, and E. Mendes. “Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method”. In: *Information and Software Technology* 85 (2017), pp. 43–59.

[103] B. H. Kwasnik. *The Role of Classification in Knowledge Representation and Discovery*. Tech. rep. 147. Syracuse University, School of Information Studies, June 1999.

[104] B. Kitchenham and S. L. Pfleeger. “Principles of survey research: part 5: populations and samples”. In: *ACM SIGSOFT Software Engineering Notes* 27.5 (2002), pp. 17–20.

[105] A. Kaplan, T. Kühn, S. Hahner, N. Benkler, J. Keim, D. Fuchß, S. Corallo, and R. Heinrich. “Introducing an evaluation method for taxonomies”. In: *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*. 2022, pp. 311–316.

[106] M. L. Loughry, M. W. Ohland, and D. DeWayne Moore. “Development of a theory-based assessment of team member effectiveness”. In: *Educational and psychological measurement* 67.3 (2007), pp. 505–524.

[107] L. Silvestre, S. F. Ochoa, and M. Marques. “Understanding the design of software development teams for academic scenarios”. In: *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*. IEEE. 2015, pp. 1–6.

[108] S. Winger-Haunty. “Peer assessment instrument”. In: *University of Wisconsin-Stout, Menomonie* (1990).

[109] B. Wei, R. Chataut, and L. Deng. “Project Pulse: Enhancing Peer Evaluation and Team Accountability in Senior Design Projects”. In: *The 37th International Conference on Software Engineering and Knowledge Engineering*. 2025, pp. 37–42. DOI: 10.18293/SEKE2025-107.

[110] C. Hundhausen, P. Conrad, O. Adesope, and A. Tariq. “Combining GitHub, Chat, and peer evaluation data to assess individual contributions to team software development projects”. In: *ACM Transactions on Computing Education* 23.3 (2023), pp. 1–23.

[111] A. Ampatzoglou, S. Bibi, P. Avgeriou, and A. Chatzigeorgiou. “Guidelines for managing threats to validity of secondary studies in software engineering”. In: *Contemporary empirical methods in software engineering* (2020), pp. 415–441.

[112] T. C. Lederman. “Evolution of capstone-courses in software engineering a finishing school”. In: *Journal of Computing Sciences in Colleges* 25.6 (2010), pp. 129–135.

[113] T. Vasankari and A.-M. Majanoja. “Practical Software Engineering Capstone course–framework for large, open-ended projects to graduate student teams”. In: *Computer Supported Education: 10th International Conference, CSEDU 2018, Funchal, Madeira, Portugal, March 15–17, 2018, Revised Selected Papers 10*. Springer. 2019, pp. 310–327.

[114] L. Johns-Boast and S. Flint. “Simulating industry: An innovative software engineering capstone design course”. In: *2013 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2013, pp. 1782–1788.

[115] R. Kerbs. “Student teamwork: A capstone course in game programming”. In: *2007 37th Annual Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports*. IEEE. 2007, F4D–15.

[116] B. Lu and T. DeClue. “Teaching agile methodology in a software engineering capstone course”. In: *Journal of Computing Sciences in Colleges* 26.5 (2011), pp. 293–299.

[117] B. Tribelhorn and A. M. Nuxoll. “Using Agile and Active Learning in Software Development Curriculum”. In: *Proceedings of the ASEE Virtual Annual Conference Content Access*. 2021, pp. 5076–5086.

[118] M. Mikalsen and T. Dingsøyr. “Feedback as a process in a large semi-capstone software engineering course”. In: *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*. 2023, pp. 475–479.

[119] M. C. Bastarrica, F. J. Gutierrez, M. Marques, and D. Perovich. “On the impact of grading on teamwork quality in a software engineering capstone course”. In: *IEEE Access* 11 (2023), pp. 36492–36503.

[120] M. C. Bastarrica and J. Simmonds. “The influence of software engineering education on developing teamwork skills in both genders”. In: *IEEE Access* (2025).

[121] C. Pierce, C. J. Woodward, A. Trevillian, and Q. T. Pham. “Game Studio Simulator: Integrating Industry Practice and Formative Individual Assessment in Collaborative Game Development Capstone Projects”. In: *ACM Transactions on Computing Education* 25.4 (2025), pp. 1–27.

[122] S. Ali and M. R. Ha. “Understanding Students’ Mindset While Reflecting on Their Teamwork Experiences”. In: *2025 World Engineering Education Forum-Global Engineering Deans Council (WEEF-GEDC)*. IEEE. 2025, pp. 1–6.

[123] A. Shakil and P. Denny. “Enhancing Student Engagement in Large-Scale Capstone Courses: An Experience Report”. In: *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*. 2024, pp. 722–728.

[124] N. Nasir, M. Usman, J. Börstler, and N. D. Fogelström. “Software engineering team project courses with industrial customers: Students’ insights on challenges and lessons learned”. In: *Journal of Systems and Software* 226 (2025), p. 112441. ISSN: 0164-1212. DOI: https://doi.org/10.1016/j.jss.2025.112441. URL: https://www.sciencedirect.com/science/article/pii/S0164121225001098.

[125] V. R. B. G. Caldiera and H. D. Rombach. “The goal question metric approach”. In: *Encyclopedia of software engineering* (1994), pp. 528–532.

[126] A. Burgess, C. van Diggele, C. Roberts, and C. Mellis. “Team-based learning: design, facilitation and participation”. In: *BMC Medical education* 20 (2020), pp. 1–7.

[127] D. Parmelee, L. Michaelsen, S. Cook, and P. D. Hudes. “Team-based learning: A practical guide: AMEE Guide No. 65”. In: *Medical Teacher* 34 (2012), e275–e287. DOI: 10.3109/0142159X.2012.651179.

[128] J. Surif, N. H. Ibrahim, and M. Mokhtar. “Implementation of problem based learning in higher education institutions and its impact on students’ learning”. In: *PBL Across Cultures* 66 (2013).

[129] Y. Liu and A. Pásztor. “Effects of problem-based learning instructional intervention on critical thinking in higher education: A meta-analysis”. In: *Thinking Skills and Creativity* 45 (2022), p. 101069.

[130] E. Hendarwati, L. Nurlaela, and B. S. Bachri. “The collaborative problem based learning model innovation”. In: *International Proceeding of Community Services* 2.1 (2022).

[131] E. Boom-Cárcamo, L. Buelvas-Gutiérrez, L. Acosta-Oñate, and D. Boom-Cárcamo. “Gamification and Problem-Based Learning (PBL): Development of Creativity in the Teaching-Learning Process of Mathematics in University Students”. In: *Thinking Skills and Creativity* (2024). DOI: 10.1016/j.tsc.2024.101614.

[132] C. V. Angelelli, G. M. de Campos Ribeiro, M. R. Severino, E. Johnstone, G. Borzenkova, and D. C. O. da Silva. “Developing critical thinking skills through gamification”. In: *Thinking Skills and Creativity* 49 (2023), p. 101354.

[133] N. M. Webb. “Task-related verbal interaction and mathematics learning in small groups”. In: *Journal for research in mathematics education* 22.5 (1991), pp. 366–389.

[134] C. Van Boxtel, J. Van der Linden, and G. Kanselaar. "Collaborative learning tasks and the elaboration of conceptual knowledge". In: *Learning and instruction* 10.4 (2000), pp. 311–330.

[135] P. Van den Bossche, W. H. Gijselaers, M. Segers, and P. A. Kirschner. "Social and cognitive factors driving teamwork in collaborative learning environments: Team learning beliefs and behaviors". In: *Small group research* 37.5 (2006), pp. 490–521.

[136] R. Rommetveit. "Thought and language. L. Vygotsky, newly edited by A. Kozulin. Cambridge, MA: MIT Press, 1986. Pp. lvi+ 287.-Marxism and the philosophy of language. VN Volosinov, translated by L. Matejka and IR Titunik. Cambridge, MA: Harvard University Press, 1986. Pp. xvi+ 205." In: *Applied Psycholinguistics* 8.3 (1987), pp. 309–313.

[137] N. E. Fenton and M. Neil. "Software metrics: successes, failures and new directions". In: *Journal of Systems and Software* 47.2-3 (1999), pp. 149–157.

[138] H. Koziolek. "Goal, question, metric". In: *Dependability Metrics: Advanced Lectures* (2008), pp. 39–42.

[139] S. Iftikhar, A.-E. Guerrero-Roldán, and E. Mor. "Practice promotes learning: Analyzing students' acceptance of a learning-by-doing online programming learning tool". In: *Applied Sciences* 12.24 (2022), p. 12613.

[140] N. M. Webb and A. S. Palincsar. *Group processes in the classroom.* Prentice Hall International, 1996.

[141] D. Boud and E. Molloy. "Rethinking models of feedback for learning: the challenge of design". In: *Assessment & Evaluation in higher education* 38.6 (2013), pp. 698–712.

Software Engineering (SE) education equips students for the complexities of the software industry, emphasizing not only technical skills but also teamwork and communication with stakeholders. SE team projects courses with industrial customers provide learning environments where students develop both technical and social skills. However, such courses often face challenges related to teamwork, ac-countability, and assessment.

Objectives: This thesis aims to support teamwork and learning in SE team project courses by i) identifying the challenges students face, ii) exploring peer evaluation as a teamwork support strategy, and iii) evaluating the perceived effectiveness of various teamwork support strategies, including peer evaluation, team contracts, and collaborative peer review.

Methods: This research adopts a qualitative, evidence-based approach. We analyzed student reflection reports using qualitative document analysis to capture students' experiences. Teacher focus groups were conducted to gather educators' insights. Additionally, a systematic literature review was performed to create a peer evaluation taxonomy, which was then validated via semi-structured interviews with SE educators.

Results: The findings show that challenges in SE team project courses are mainly socio-technical rather than purely technical. Several strategies, such as peer evaluation, team contracts, and collaborative peer review workshops, are used to support teamwork. However, peer evaluation practices in literature vary widely in their design and reporting. To address this, the thesis proposes a taxonomy of peer evaluation with guidelines for designing peer evaluation processes. The thesis also finds that both students and teachers perceive peer evaluation and team contracts as useful strategies; however, they need structured follow-up.

Conclusion: This thesis provides empirical insights into teamwork challenges and lessons learned. It also contributes taxonomy and design guidelines for peer evaluation. Further, it provides qualitative evidence on the perceived usefulness of different teamwork strategies. The findings highlight the importance of context-sensitive design and structured follow-up when implementing teamwork support strategies in SE team project courses.

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2026:02
ISSN: 1650-2140
ISBN: 978-91-7295-524-0

Printed by Media-Tryck, Lund 2026 NORDIC SWAN ECOLABEL 3041 0903