IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 31 January 2023, accepted 17 March 2023, date of publication 6 April 2023, date of current version 17 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3265302

RESEARCH ARTICLE

On the Impact of Grading on Teamwork Quality in a Software Engineering Capstone Course

MARÍA CECILIA BASTARRICA[1], FRANCISCO J. GUTIERREZ[1], MARÍA MARQUES[2], AND DANIEL PEROVICH[1]

[1]Department of Computer Science, University of Chile, Santiago 8370449, Chile
[2]Department of Computer Science, Boston College, Chestnut Hill, MA 02467, USA

Corresponding author: María Cecilia Bastarrica (cecilia@dcc.uchile.cl)

The work of Francisco J. Gutierrez was supported in part by FONDECYT—Initiation to Research Program under Grant 11190248.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the IRB of the Faculty of Physical and Mathematical Science, University of Chile.

ABSTRACT Every semester, we deliver a capstone course on software engineering where students undertake a real-world project in three iterations. By the end of each iteration, students are graded in several dimensions: software quality, project management, and peer assessment. The latter is the only grade assigned individually; therefore, students who are penalized by their teams (e.g., for being perceived as low contributors to the team effort) are not severely affected. This results in little incentive for improvement, which potentially jeopardizes the overall quality of the project outcome. Envisaging to promote team cohesion, we devised a new grading rule: if the peer assessment of a student is lower than a threshold, that would be their final grade in the iteration. This paper reports the results of studying the effectiveness of the proposed rule. We recorded peer assessments over six consecutive semesters: (1) the first three as the baseline measure; (2) the semester where we introduced the new grading rule; and (3) the following two semesters, as a contrast. When the rule was first introduced, peer assessments resulted low and heavily spread at the beginning, but they consistently improved toward the end of the semester. When the instructional team already trusted the rule and explicitly emphasized its application, peer assessments consistently grew along the semester but resulted in fewer outliers. The study results show that exposing peer assessments earlier on helps promote team reflection. They also made evident the positive impact of teamwork for producing quality products in a software engineering capstone course.

INDEX TERMS Capstone course, software engineering education, teamwork.

I. INTRODUCTION

Capstone courses are widely considered as a valuable asset for developing practice-oriented abilities among advanced software engineering students [1]. Over the past ten years, Computer Science undergraduate students at the University of Chile have taken a one-semester capstone course on software engineering. Here, students work 16 hours per week in the client's facilities, are self-managed in teams of 5 to 7 members, and tackle real projects. Once a week, each team meets with the instructional team for coaching and strategic advice. The project is formally structured in three iterations. By the end of each one, teams deliver a public presentation and a software demonstration. Classmates, instructors, tutors, and clients attend these presentations.

Senior year students enroll in this course, bolstered by a strong technical background through a series of courses on software engineering, databases, algorithms, programming languages, systems programming, among others. However, they usually have little experience managing their own projects and making strategic decisions, even though these competencies are widely valued both in academia and industry. Several works have realized that the main learning outcomes of these kinds of courses are soft skills, such as project

The associate editor coordinating the review of this manuscript and approving it for publication was Claudia Raibulet.

planning, negotiation, and teamwork [1], [2], [3]. There is also evidence that suggests that teamwork skills improve over time when they are actively taught and assessed [4], [5].

To formally assess teamwork quality in the course, we designed a sophisticated grading schema, where several perspectives are considered: project management, product quality, presentation, value of the solution, and peer assessment. The grading components in the schema collectively affect all team members, with the sole exception of peer assessment. This latter factor accounts for only a 5–15% of the grade. Furthermore, as we have observed over the years, peer assessments have often been disregarded when providing feedback, but are instead used as a moral penalization to those team members who are perceived as low contributors at the end of the course. Therefore, during the first semester of 2018, we implemented a new rule for weighing peer assessment in each iteration: *if a student's peer assessment is equal to or lower than a predefined threshold, then the student's grade for that iteration will be the value of his/her peer assessment, without taking into account any of the other dimensions*. This rule was explained very clearly at the beginning of the course and students agreed to complete their peer assessments as fairly as possible.

The objective of this paper is to evaluate the impact of the new grading rule over teamwork quality. Concretely, the study focuses on answering the following research questions:

RQ1: *How fair and useful is the proposed rule for improving teamwork?*

RQ2: *How effective is the new peer assessment rule for improving teamwork quality throughout the project development?*

To answer these research questions, we analyzed the progression of peer assessment along the three iterations, and then the results were compared with those of the previous and following semesters. In particular, we implemented three consecutive Action Research cycles [6], where we studied how peer assessments varied: (1) over the three previous semesters to the explicit intervention; (2) during the semester where the new grading schema was introduced; and (3) over the two following semesters, where the whole instructional team applied the rule emphasizing its impact from the beginning of the course. In each semester, we analyzed the progression of peer assessment along the three iterations, and the results were compared between cycles. An end-of-course survey provided students' perception about the usefulness and fairness of the new grading rule. By following this line of analysis, this paper significantly extends the results previously presented by Bastarrica et al. [7].

When the rule was first applied, even though students generally got lower peer assessment grades than in previous semesters, they still perceived the rule as fair and useful. In addition, grade dispersion in the first iteration was low with just a few outliers, while in the second one it resulted in a much larger range. Peer assessments improved substantially in the third iteration: the median value resulted much higher than both previous iterations, the range is even smaller than in the first iteration, and there is only one outlier. This suggests that the feedback provided by peer assessments in the second iteration resulted in effectively improving teamwork. Likewise, we observed that the instructional team did change their attitude with regard to teamwork assessment in cycle 3. In this cycle, the instruction team had acquired confidence on the effectiveness of the rule and they stressed its implications in all weekly meetings from the very beginning. As a result, peer assessments improved systematically and there were almost no outliers in the final iteration.

This manuscript contributes to the Software Engineering Education community, by showing that giving students a say in peer grades improves teamwork. More particularly, exposing low peer assessments early in the project is useful feedback for students, tutors, and course instructors, as it enhances team reflection when there is still time and both instructional and student teams can take action for improvement.

The rest of this paper is structured as follows. Section II discusses related work on teamwork and grading capstone courses. Section III describes the course and the new grading schema followed in this study. The research methodology and empirical results are presented in Section IV. Section V discusses the results and Section VI presents the research threats to validity. Finally, Section VII draws the conclusions and provides perspectives on future work.

II. RELATED WORK

In this section, we review the body of literature related to this work. We structure this discussion as follows: (1) the relevance of teamwork in software engineering, (2) team-based learning, (3) capstone courses in software engineering, (4) grading capstone courses, and (5) methods and techniques for peer assessing capstone courses.

A. TEAMWORK

The relevance of teamwork quality in software engineering has been highlighted in both academia and industry. Lindsjørn et al. [8] claim that teamwork quality directly affects project results and is also perceived to have a large effect on personal success in professional practice. In that respect, ABET (Accreditation Board for Engineering and Technology) [9] and the ACM Curriculum Guidelines [10] mention that students need to develop teamwork skills transversely and that teamwork knowledge is usually acquired through problem and project-based courses [11], [12], [13]. Broadly speaking, the perceptions of successful teamwork range from effectively completing the assigned tasks to a more social dimension, where members actively invest on fulfilling working relationships [14].

Quality in teamwork projects can be seen through different lenses as stated in Figure 1. Firstly, personal disposition for teamwork, which can be characterized as psychological and technical characteristics that make people more or less prone to work in teams. Driskell et al. [15] pointed out five dimensions that can define if a person is better suited for teamwork:

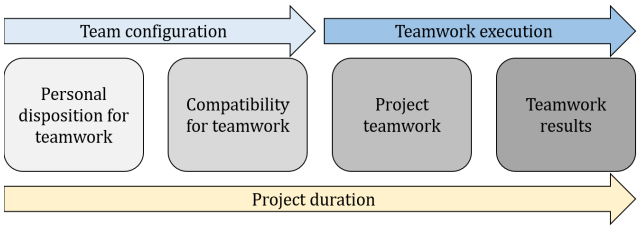

FIGURE 1. Teamwork quality dimensions (own work).

emotional stability, extroversion, openness, agreeableness, and conscientiousness. According to Sudhakar et al. [16] teamwork quality is affected by team members' technical/innovation competencies, top management support, and team leader behavior.

Secondly, there is personality compatibility and technical complementarity that gives a group of people the potential to succeed in working as a team [17], [18]. Thirdly, project teamwork is the work that a team performs to develop the software they are intended to. There are several identified characteristics that affect project teamwork: communication, coordination, and cooperation [19], team capability, delivery strategy, applied software engineering techniques [20], and team climate.

Finally, teamwork is successful if its results are good in any sense, e.g., software quality, cost, scope, or time [20]. Although the quality of the resulting software is independent of the way the team is organized (or if a team existed at all), all other teamwork quality dimensions cannot be fairly evaluated without considering this result. Therefore, if results are poor, neither high coordination nor high compatibility can be considered as satisfactory.

B. TEAM-BASED LEARNING

According to Michaelsen et al. [21], there are four essential principles of Team-Based Learning (TBL) that need consideration in any course involving teamwork: (1) groups need to be properly formed and managed; (2) students have to be accountable for individual and group work; (3) assignments should promote learning and team development; and (4) students should receive frequent and timely feedback. Although the course analyzed in this paper adheres to TBL principles, it does not include lectures; therefore, it cannot be assessed with the Readiness Assurance Process (RAD) [21], a weekly quiz aiming to assess whether students have acquired the foundational concepts that are required to begin problem-solving as a team.

Awatramani and Rover [22] describe the use of TBL among online students, where all interaction was asynchronous through different web tools. They intended to analyze the development of TBL high-level skills, although students expressed some essential challenges of TBL, such as accountability and immediate feedback. Inspired by these results, the course analyzed in this paper gives students feedback and requires them to take accountability on what they have done and have not regarding their project.

C. CAPSTONE COURSES

According to Sherriff and Heckman [23], students in the later stages of their degree programs need to be exposed to opportunities where they put into practice all they have learned. Following the ACM Curriculum Guidelines [10], this experience can be achieved through the execution of a project-based capstone course, grounded in the principles of experiential learning [24], [25].

Situated learning [26] builds on the idea of experiential learning. While the latter focuses on learning-by-doing, the former is concerned with where it happens. This approach is based on the belief that knowledge is particular to a context and it can only be acquired by being immersed in it. According to Navarro [27], the context in which students practice their knowledge should be authentic and resemble, as closely as possible, the real professional activity.

Therefore, modern software engineering education and training programs need to provide the means for students to acquire soft skills such as collaboration, teamwork, and the ability to manage workers in a large project [2], [28]. Unlike real world projects that can be bound by contracts, students in capstone courses have to exercise teamwork, in a context where negative consequences of bad work or performance are not necessarily a big deal, since they are usually only expressed in a course grade [28]. Moreover, in the context of capstone courses, there are also other dimensions that determine project success such as learning, satisfaction, and self-validation.

D. GRADING CAPSTONE COURSES

Grading teamwork is an on-going discussion that does not necessarily have a one-size-fits-all solution. According to Clark et al. [29], software engineering courses should have teamwork as one of the intended learning outcomes. So, the grading process should involve assessing how well/badly teams work. The authors also point out the relevance of assessing individual contributions and its dilemma, since instructors are not present at all team meetings.

Tafliovich et al. [30] studied how computer science undergraduate students perceive their assessments when graded individually or as a team. In particular, students tend to prefer individual assessments early on in the project. However, as they get more involved in development, they tend to rapidly prefer group evaluations.

Likewise, Herbert [31] found that students tend to grade themselves more harshly than they mark their teammates, displaying the subjective dimension of do assessments. In that respect, Smith and Smarkusky [32] developed an assessment framework, based in competencies, as a way to assist students in making more objective judgments. Following a similar line of reasoning, Petkovic et al. [33] developed machine learning models to objectively predict student attitudes toward teamwork in software engineering courses.

Grading team projects and products has to take into consideration individual and team contribution, and also the context

of the project and deliverable. There is a lot heterogeneity among grading processes: some give relevance to the product being developed, where others give relevance to the process that is being used by the team. There are models that only consider the final project/product under a holistic approach, while there are others that have a progressive assessment of the work being done or the process being followed, or even a combination of them. Sometimes, real clients (i.e., non-academic) are involved, who may take part in grading teams, and this may raise issues that concern students. In particular, any grading schema should stress the management of student expectations, as well as providing transparency on the grading process [34].

E. PEER ASSESSMENT

Assessing a team is not necessarily straightforward, since it is not always possible to measure what each individual team member did (i.e., accountability) and it is also not fair to just give the same grade to all members of a team. Michalesen et al. [21] were among the first authors to mention the relevance of peer assessment in courses where students are divided in teams. They postulate that no one knows better about students' accountability than the students themselves.

Marshall et al. [25] state that the behavior of each team member is highly influenced by the team composition and the behavior of other members, so the opinion of their teammates is clearly relevant. Therefore, peer assessment has been proposed as a means for evaluating teamwork in several areas. Reciprocal peer assessment [35] includes not only peer assessment, but also self assessment. However, this may introduce gender bias [36].

Furthermore, Asikainen et al. [37] indicate that peer assessments can have a better effect on team performance than an expert or instructor evaluation in the context of massive undergraduate courses, i.e., those with a large number of enrolled students. In fact, according to Double et al. [38], positive peer feedback has a more effective impact on teamwork quality than any other aspect. This is consistent with some of our findings. Likewise, Hoang et al. [39] found that students could improve their fairness and accuracy on peer assessments when they realize that their responses to other team members as well as their own work are taken into account in the calculation of their grades.

But peer assessment also has its own drawbacks. In that respect, some authors have criticized the effectiveness of peer assessments [29], arguing that groups may collude to share marks, either as a self-reward or to penalize a single team member [40]. Wilkins and Lawhed [41] found the same issue, and they conclude their work recommending either using peer assessment complemented with other evaluation techniques or involving the teaching staff in supervising peer assessments. Chen et al. [42] also showed that the evaluation of software engineering projects should consider several factors such as attendance, team presentation, product, and peer assessment.

Peer assessment can reinforce the strategy of students generating and receiving evaluations. This helps them actively reflect upon their own performance as well as that of others [43]. Students may sometimes perceive peer assessments as biased. However, reinforcement theory [44] shows that several raters tend to identify similar strengths and weaknesses in their teammates. This theory emphasizes that students who receive bad grades improve their performance by focusing the effort only on problems recurrently mentioned by their peers and filtering out incorrect or inappropriate idiosyncratic feedback.

Clark et al. [29] state that if a student becomes aware of what is needed to improve their performance, they might become a better contributor. Consequently, their grade would improve, as well as that of the team as a whole. Therefore, peer assessments are an important assessment tool to evaluate individual contributions. Finally, Hoand et al. [39], observed a positive attitude of students toward peer assessment activities when they saw that their answers were considered in the grading system.

III. CAPSTONE COURSE GRADING SCHEMA

This section describes the capstone course analyzed in this paper along with its grading schema. It later presents the new grading rule.

A. GENERAL COURSE DESCRIPTION

Software Project is the last mandatory course in the Computer Science undergraduate major at the University of Chile. This capstone course is taken by all students in their 11^{th} semester. Students at this level have already taken two courses in software engineering, as well as courses in databases, programming languages, algorithms, operating systems, among others. Therefore, they already have a strong theoretical background and some initial exposure to problem solving and software development in teams. However, none of the these courses focus on project development in real settings [1].

Each semester lasts 15 weeks. In the first two weeks, students attend lectures where the course is thoroughly explained and some foundations on agile practices are presented [45], [46]. This setup, according to literature (e.g., [47]) is effective for delivering capstone courses in software engineering.

In these first lectures, the rules and administrative aspects of the course are presented. Teams are formed randomly only considering time availability, so that no one should be working alone at any time, as recommended for agile development. In particular, neither psychological compatibility nor complementary technical competence are taken into account. Also, a different project and tutor are assigned to each team.

At the end of the second week, each team gives a "Setup Iteration" presentation, where they describe the company/organization where they will work and the project they should address. This presentation is internal to the course, i.e., only the instructor, tutors, and enrolled students attend. As a result, students receive qualitative feedback, but not a grade.

TABLE 1. Course grading schema.

Evaluator	Aspect	Iter. 1	Iter. 2	Iter. 3
Tutor	Project management	35%	30%	20%
Tutor	Product quality	40%	40%	40%
Instructor	Presentation	10%	10%	10%
Students	Peer assessment	15%	10%	5%
Client	Value of the solution	0%	10%	25%

After the first introductory weeks, students develop their projects for 13 weeks working at the clients' facilities. This period is divided into three iterations that last for 4, 4, and 5 weeks, respectively. Teams meet with their tutor and the instructor once a week to report on project advances and discuss challenges they have faced in their work, both technical and managerial. At the end of each iteration, teams deliver a public presentation where they present the project goals, the work completed so far including technical issues, the work planned for the following phases, and a reflection about their learning experience. Teams must also showcase a working software that presents relevant user stories of the application developed so far. In these instances, clients are also present.

B. GRADING SCHEMA

At the end of each iteration, students receive a grade that is computed taking into account several aspects with different weights that also vary between iterations [48]. Table 1 summarizes the grading schema. Final course grades are computed assigning a weight to the grades students obtain in each iteration: 25% for iterations 1 and 2, and 50% for iteration 3.

Different actors are responsible for assigning these grades. Project management refers to the way students plan their projects and control progress, the way they negotiate with their clients, and their organization as a team. These aspects are assessed during weekly meetings with the instructional team. Product quality also considers diverse aspects: choice of the appropriate technology, amount of functionality implemented, code quality and software performance and robustness. The tutor, who is the person that follows the project the most closely, evaluates these dimensions. The instructor is in charge of evaluating the presentation. Finally, the client grades the value that the software solution adds to their organization. Given that the produced software is technically owned by the on-site clients, the instructional team, i.e., course instructor and team tutors, only have access to architectural diagrams and models, and evaluate software functionality in development or testing environments.

At the end of each iteration, students must complete an anonymous peer assessment questionnaire about the quality of their teammates work. This instrument gauges the team members' viewpoints regarding personal disposition for teamwork and actual work within the project. Five aspects are evaluated: commitment, communication, coordination, attitude, and contribution. Each aspect is represented by one or more items in the questionnaire describing an expected attitude of the teammate, as presented in Table 2. The answers

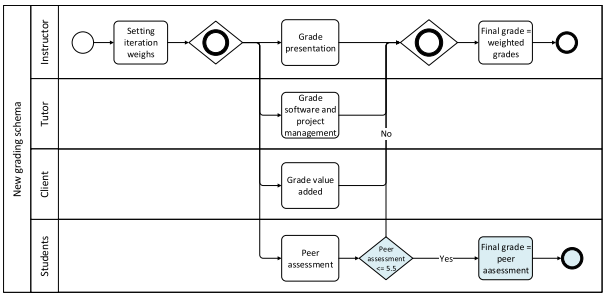

FIGURE 2. Peer assessment rule.

are indicated using a five-point Likert scale ranging from "strongly disagree" to "strongly agree". This questionnaire was inspired by the work of Silvestre et al. [49], where the items measuring the teamwork quality dimension were validated by Marques [50]. In her work, Marques showed the consistency in the answers to this questionnaire over several applications in academic scenarios, hence the collected peer assessments in this study are considered as accurate.

The answers provided to the questionnaire described above are the only indicators used to measure peer assessments, and consequently, the perception of teamwork quality by the students enrolled in the studied course. Other factors, such as self-evaluations or a team member's commits tracked by a version control system were not analyzed at this time. Nevertheless, they could be considered as additional grading dimensions in future iterations of the course.

C. NEW GRADING SCHEMA

Talifovich et al. [30] found that when students develop software in teams, they prefer a grade that largely depends on the group effort, while simultaneously acknowledging the individual effort, although in a lower proportion.

Several works report the existence of free riders—or social loafers—in software development teams [51], [52], i.e., students that aim to benefit from the team work and team grade, without contributing as expected by the other team members. This used to be the case in the reported capstone course, too. The instructional team encouraged students to discuss these issues during weekly meetings and recommended the use of peer assessments as a way to clearly state their dissatisfaction and promote a change in attitude. However, free riders did not disappear since the penalty for not participating was still low.

To address this recurrent scenario, a new grading schema was devised. In the first semester of 2018, the new rule was introduced: if a student's peer assessment is lower or equal to 5.5 (where 4 is the minimum passing grade over a range from 1 to 7), then the student's grade for that iteration will be the value of the peer assessment without taking into account any of the other evaluation dimensions. Thus, the traditional grading schema would only be applicable in the case of satisfactory peer assessments. It is worth noting that this threshold is not arbitrary, but based on the grades free riders have obtained in the past. The new rule is clearly explained at the beginning of the course. Students are encouraged to

TABLE 2. Questions in the peer assessment instrument and their contribution to different teamwork aspect.

	Commitment	Communication	Coordination	Attitude	Contribution
The student assumes the project as a team effort, providing support in project tasks	X				
The student is able to seek help when problems are found	X				X
The student fulfills his/her tasks properly, working transparently and generating the most value out of each working day.				X	
The student demonstrates initiative to achieve project success		X			
The student shows a communicative attitude facilitating teamwork			X		
The student has maintained good communication with the client, generating value to project execution					X
The student demonstrates interest in improving performance on the execution of his/her activities and role within the project.				X	
The student is able to admit mistakes and accept criticism.				X	

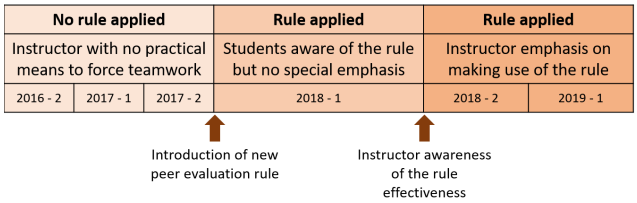

FIGURE 3. Action research cycles.

TABLE 3. Number of teams and students enrolled each semester.

Cycle	Semester	Teams	Students
1	Second 2016	6	36
	First 2017	3	17
	Second 2017	4	24
2	First 2018	4	24
3	Second 2018	6	35
	First 2019	3	18

complete peer assessments as righteously as possible, and thus to take responsibility for their teammates' grades. Figure 2 depicts a BPMN diagram showing the course grading schema, highlighting the new peer assessment rule that we study in this work.

IV. METHODOLOGY AND RESULTS

To answer the research questions stated in Section I, we conducted a field experiment [53] structured around three cycles following the action research approach. The units of analysis were: (1) the 2018 first semester offering of Software Project course, where the new grading rule was first introduced; (2) the previous three offerings of the same course (namely, the second semester of 2016 and both semesters of 2017), that were used as contrast; and (3) the second semester of 2018 and 2019 first semester, were analyzed as follow up when the rule was already established as a common practice. Figure 3 describes these research cycles. It is worth noting that, even though we have collected data from all semesters up to present, they can not be directly compared with those in the study: the country suffered national social unrest during the second semester of 2019, and during 2020 and 2021 all teamwork occurred online due to the COVID-19 pandemic.

A. DATA SOURCES

We considered various data sources: (1) student perceptions about the impact of the new rule on teamwork quality in the semester when it was first introduced and (2) peer assessment progression during all the semesters under consideration. The collected data relate to the stated research questions as follows:

RQ1 aims to understand students' perception about the justice of the rule and its usefulness for aligning teamwork. To this end, a survey on student perceptions on usefulness and fairness of the new rule was conducted in the semester when the rule was first introduced.

RQ2 intends to check whether students' teamwork quality actually improves along the semester and, consequently, teamwork being an actually acquired knowledge. To answer this question, we compared the progression of peer assessment grades along the three iterations in each cycle.

B. PARTICIPANTS

During the first semester of 2018 (i.e., the intervention cycle), 24 students were enrolled in the course. They were organized in 4 teams of 6 students each, as shown in the middle row of Table 3. The criteria that was used to assign students into teams was the same in all semesters.

Tutors were all current or former graduate students whose research focused on software engineering. Teams, tutors, and projects were randomly matched. The instructor was the same across all semesters considered in this study.

C. ETHICAL CONSIDERATIONS

The empirical design followed in this study complies with the key principles in research ethics as stated by the American Psychology Association (APA).

In particular, all students that took the course with the new grading rule were informed of it through the course syllabus and also during the first class. Students could also ask questions regarding the course and its grading, where the instructor clarified them to be sure everyone understood. All participants agreed to take part in this study.

D. FAIRNESS AND USEFULNESS OF THE NEW GRADING RULE

A survey was issued for evaluating student perception of fairness and usefulness of the new grading schema when it

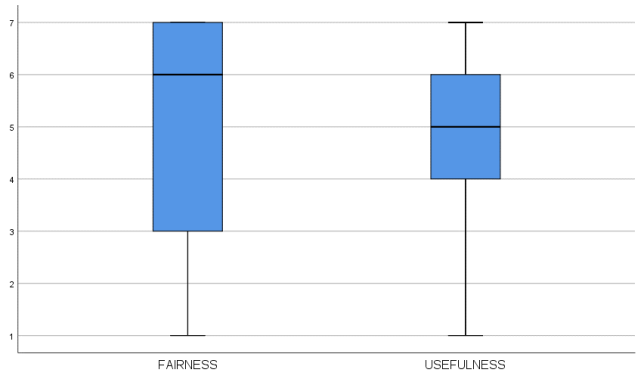

FIGURE 4. Students' perceived fairness and usefulness of the grading rule.

was first introduced. It consisted of two questions evaluated with a 7-point value Likert scale ranging from *Completely agree* to *Completely disagree* and an optional open question. These questions were:

- Did you receive a fair grade, corresponding to your contribution, attitude, and commitment?
- Was the new rule useful for improving teamwork?
- What would you change about the new rule? (open)

The survey was distributed as an online form to all students right after the end of the last iteration and before final grades were given. The survey was anonymous. Even though completing this survey was not mandatory, students were encouraged to fill in the form provided that this data was a way for them to contribute to make the course better for future generations.

A total of 21 answers were obtained to the first two questions, i.e., an 88% response rate. Only 9 students (38%) answered the open question.

Figure 4 describes the obtained results in the form of a boxplot. Obtained answers about fairness and usefulness ranged both from 1 to 7, i.e., they make use of the full scale. The median values were *Agree* (6) for fairness and *Partly agree* (5) for usefulness.

E. PEER ASSESSMENTS IN EACH CYCLE

To study the impact of introducing the proposed grading schema on teamwork, we grouped the collected data in three cycles: (1) where no rule was applied and the instructor had no practical means to force teamwork through grading, i.e., semesters 2016-2, 2017-1, and 2017-2; (2) where the rule was applied and students were aware of this action, but there was no special emphasis from the instructional team, i.e., semester 2018-1; and (3) where the rule was applied and the course instructor emphasized on making use of the rule, i.e., semesters 2018-2 and 2019-1.

On the one hand, we comparatively analyzed the evolution of peer assessments along the three iterations for the semesters grouped in each cycle. On the other, we comparatively analyzed the proportion of students whose peer assessments could be considered as outliers according to the grade distributions in each cycle (i.e., those students who were penalized by the application of the rule). Figure 5 shows a boxplot comparing the collected peer assessments along each one of the three iterations per cycle. It is worth remembering that one of the main goals of the course is that all students improve their teamwork skills. In this sense, an increasing median would represent a progressive improvement of teamwork performance, while a decreasing number of outliers represent that *all* the students acquire this knowledge.

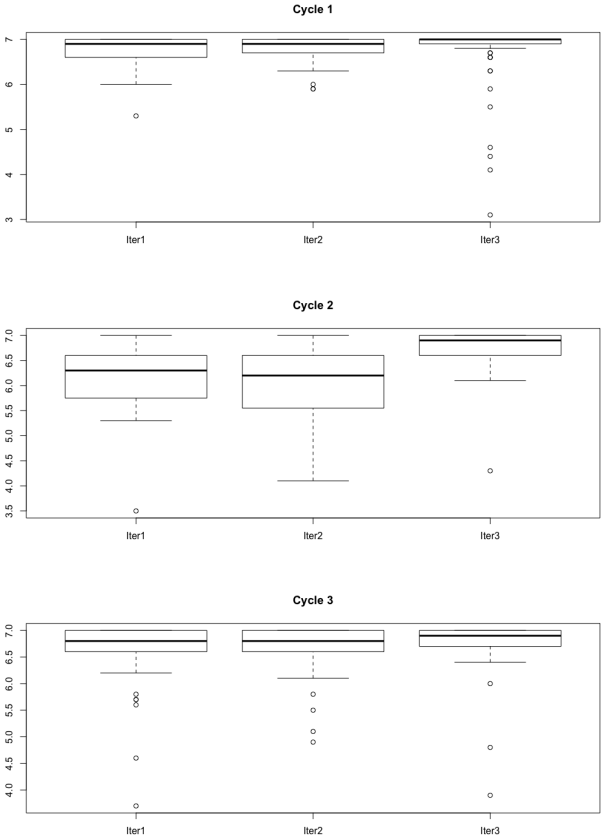

FIGURE 5. Peer assessment distribution per iteration along each cycle.

1) EVOLUTION OF PEER ASSESSMENTS

To study the differences on peer assessments along the iterations, we ran three Friedman's ANOVA models, one per cycle. These models aim to statistically evaluate whether there is a difference in the median values across several groups from a population. Given that the collected data did not satisfy the underlying hypothesis for running a traditional parametric repeated-measures ANOVA, we decided to apply the nonparametric counterpart. Normality was formally assessed with Shapiro-Wilk tests. In that respect, Table 4 reports the test statistic (W) for each group considered in the analysis, as computed using the function `shapiro.test` in R version 4.2.2. Since $p < .05$ in almost all cases, we reject the null hypothesis, which states that data comes from a normal distribution. This is one of the key assumptions for running a parametric repeated-measures ANOVA.

For the three semesters that were grouped in cycle 1, i.e., those that served as a control for the experiment, peer

TABLE 4. Normality test results.

Cycle	Iteration	W	p
1	1	.764	<.05
	2	.756	<.05
	3	.416	<.05
2	1	.819	<.05
	2	.929	.09
	3	.592	<.05
3	1	.657	<.05
	2	.694	<.05
	3	.518	<.05

TABLE 5. Number of observed outliers.

Cycle	Iteration	N	O
1	1	77	1
	2		2
	3		9
2	1	24	1
	2		0
	3		1
3	1	53	5
	2		4
	3		3

assessments significantly changed over the three iterations, $\chi^2(2) = 14.221, p < .05$. This means that we identified an overall effect on the differences between the median values for peer assessments across the three iterations in the cycle. However, after running pairwise post hoc Wilcoxon signed rank tests, we did not observe statistically significant differences between iterations ($med_1 = 6.9$, $med_2 = 6.9$, $med_3 = 7.0$). The critical significance level for all tests was adjusted following the Bonferroni correction. Statistical analyses were conducted using the functions `friedman.test` and `pairwise.wilcox.test` in R version 4.2.2. A boxplot representing the progression in peer assessments in cycle 1 is shown in the upper part of Figure 5.

Regarding cycle 2, i.e., the first semester that we introduced the rule, these grades significantly changed over the three iterations during the semester, $\chi^2(2) = 19.062$, $p < .05$. In other words, an overall effect was observed regarding the median values for peer assessment grades along each iteration. As a way to observe the pairwise differences in cycle 2, we ran post hoc tests applying a Bonferroni correction for significance levels. Peer assessment grades did not significantly change from the end of the first iteration to the end of the second one ($difference = -0.1, p = 0.999$). However, differences were found when comparing the results between the end of the first iteration and the final one ($difference = 0.6, p < .05$), as well as between the end of the second iteration and the final one ($difference = 0.7, p < .05$). It can be concluded that the rule of modifying the grading schema did affect the overall results during iterations 2 and 3. This difference is apparent in the graphical representation of peer assessments in the middle of Figure 5.

Finally, in the semesters after we introduced the rule (i.e., cycle 3), we observed an overall effect in the differences between the median values of peer assessment grades, $\chi^2(2) = 8.0513, p < .05$. However, no statistically significant differences were detected in pairwise post hoc tests, adjusting the critical significance level following a Bonferroni correction ($med_1 = 6.8$, $med_2 = 6.8$, $med_3 = 6.9$). This is graphically shown in the lower part of Figure 5.

2) EVOLUTION OF OUTLIERS

To study the differences on the number of outliers, we conducted three chi-squared goodness of fit tests, one per cycle. These tests aim to statistically evaluate whether there is a difference in the proportions of the target measure across several groups from a population. In this case, we were interested in comparing the number of students in each cycle that were penalized by their teammates, i.e., those whose peer assessments were significantly lower than the median values reported in each iteration within the same cycle. In this context, we define an outlier as a data point whose value is numerically distant from the median. In other words, in a boxplot we can visualize these points as those that fall beyond the lower whisker (i.e., $Q1 - 1.5 * IQR$, where $Q1$ is the measure for the first quartile and IQR is the interquartile range). Table 5 reports the observed number of outliers (O) for each iteration in each cycle. The number of expected values, required to calculate the chi-squared test statistic, was estimated while computing the `prop.test` function, available in R version 4.2.2.

For the three semesters that were grouped in cycle 1, i.e., those that served as control for the experiment, the proportion of outliers differed by iteration, $\chi^2(2, N = 77) = 10.021, p < .05$. In particular, we note that in iteration 3, the proportion of students who were graded by their peers significantly lower than the overall distribution of peer assessments ($p_3 = .117$), is greater than that at the end of the first iteration ($p_1 = .013$) and at the end of the second iteration ($p_2 = .026$). This suggests that students tended to use peer assessments as a way to penalize those team members whose contributions did not match the expectations within the group, hence displaying a rather low team cohesion.

Regarding cycle 2, i.e., the first semester that we introduced the rule, the proportion of outliers did not differ by iteration, $\chi^2(2, N = 24) = 1.0286, p = .598$. This means that the proportion of students who were graded significantly lower than the overall peer assessment distribution in each iteration, did not differ ($p_1 = .042, p_2 = .000$, $p_3 = .042$).

Finally, in cycle 3, i.e., the semesters after the rule was initially introduced, the proportion of outliers also did not differ by iteration, $\chi^2(2, N = 53) = .5408, p = .763$. This means that the proportion of students who were graded significantly lower than the overall peer assessment distribution in each iteration, did not differ ($p_1 = .094, p_2 = .075, p_3 = .057$). These results suggest that in cycles 2 and 3, i.e., where there was an explicit use of the rule for addressing peer assessments as a way to improve teamwork, students tended to show better team cohesion and did not extensively penalized their team members at the end of the project (i.e., iteration 3).

V. DISCUSSION

In this section we discuss the results of the application of the new grading schema in the three cycles.

A. FAIRNESS AND USEFULNESS OF THE NEW GRADING SCHEMA

As depicted in Fig. 4, students agree that the devised grading schema is fair. However, the range in fairness scores is rather high, where some students considered the rule to be either too soft, or either too severe. For instance, some students argue that under certain circumstances, uncommitted team members should be penalized more severely: *"Students with a low peer assessment should fail the course altogether."* However, in the opinion of other students, loafing and free riding are attitudes that can be discussed, as a way to improve teamwork when there is still time for improvement. *"It is not appropriate to punish students, since problems could be addressed talking with other teammates."* These comments reinforce the idea of promoting early discussion of the relevance of teamwork within each team. This supports the introduction of the second intervention: stressing from the beginning the relevance of using the grading rule.

Regarding the perceived usefulness of the new grading rule, peer assessment scores are all given in a much smaller range. This situation can be explained by a small proportion of students actually affected by the rule, mainly by the end of the second iteration. For instance, as stated by a student: *"I never considered the rule during the project and I always received good grades. However, if I had got a low evaluation, I would have discussed it with my team."*

Here, again the early discussion on teamwork is recognized as useful by students and almost none of them complained. Moreover, teamwork is recognized as more valuable than technical competence as highlighted by a student regarding a teammate: *"You have remarkable knowledge provided your industrial experience, but I do not know for sure why you disliked working with us. There were moments when we really needed your active participation"*.

We can assume that most of the students that did not find the rule useful correspond to those that did not need the rule to work competently within the team. Therefore, we feel confident about introducing the rule for future semesters.

Although comments refer to different situations, they coincide in remarking that talking with teammates is the way to proceed if they get a low peer assessment. This is backed by the results of the survey, where students recognize the fairness and usefulness of the new rule even when final grades did not result higher than in previous semesters.

Therefore, regarding RQ1, students perceive the grading rule based on peer assessment as fair, useful, and a source of encouragement for improving teamwork. Moreover, they perceive that early addressing teamwork issues can improve its quality and therefore, the role of the instructor results determinant.

B. PROGRESSION OF TEAMWORK QUALITY ALONG THE SEMESTER

To address the effectiveness of the new grading rule (RQ2), we discuss how it impacted teamwork quality along the different cycles of the study.

1) CYCLE 1: NO RULE APPLIED

In the initial situation (cycle 1 of Figure 5), the median values of peer-assessments did not vary along the semester, but the number of outliers was significant at the end. Therefore, we can say that only at the end of the semester it was apparent that there were people that did not contribute to their teamwork. Since learning to work in teams is one of the goals of the course, this is not a desirable situation. These findings support what we foresaw at the moment, i.e., that an intervention was necessary.

2) CYCLE 2: INTRODUCTION OF THE GRADING RULE

The motivation of the rule was to have all students work satisfactorily in teams by the end of the semester. The application of the rule when it was first introduced produced this effect: the median value resulted as high as that of cycle 1, but there were substantially fewer outliers.

However, the progression of peer-assessments along the semester changed: the median value in the second iteration slightly decreased and the dispersion significantly grew as can be seen in cycle 2 of Figure 5.

This situation is similar when considering separately each team in the course. Even though they may have different values of median and dispersion, the progression is similar. In this sense, by the end of the project, teams were working better—with higher peer assessments—and more cohesively—with lower dispersion.

Finding a large dispersion in the second iteration was an unexpected behavior. It can be hypothesized that in the first iteration, students do not have a deep knowledge about their teammates' performance yet, and only in the second iteration they feel they can make an informed assessment. One of the comments received in the survey supports this claim: *"The new rule should not be applied in the first iteration, since it is always difficult to get started; it is fine to apply it in the following iterations"*. This situation motivated the second intervention: stressing the value of teamwork from the beginning and reminding students that the rule existed as well as its consequences.

Therefore, regarding RQ2, it is possible to conclude that the new rule was effective for improving teamwork quality.

3) CYCLE 3: STRESSING THE USE OF THE RULE FROM THE BEGINNING

As previously mentioned, one of the goals of the course is to have all students achieve a high quality teamwork skill. We can say that in cycle 2, this goal was reached by the end of the semester, but at the expense of suffering certain disruption in the middle. The second intervention fixed this undesirable

behavioral: now peer-assessments are consistently growing and there are also few outliers, obtaining the best of both worlds.

This situation makes the conclusion of RQ2 even stronger: the rule is not only effective for improving teamwork quality toward the end of the semester, but it also consistently improves it along the course.

VI. THREATS TO VALIDITY

One main threat to construct validity of this study is that it considers peer-assessments as the only parameter for evaluating teamwork quality. Other studies have proposed complementary approaches, such as self-assessments.

Moreover, the proposed grading schema only considers the quality of teamwork while developing the project (commitment, communication, coordination, attitude, contribution). However, there are other factors that would also probably have an impact, e.g., familiarity with technology or the application domain, personal compatibility, or project complexity. It was assumed that all students had the same personal disposition for teamwork and similar technical knowledge. As for the quality of the resulting projects, it was consistently high, so a more detailed study should be conducted to particularly address this relationship.

In the intervention cycle (i.e., cycle 2), the sample was formed by 24 students, which is an average size for this course. Given that all other variables, e.g., instructor team, student team size, general grading schema, remained the same, the variation in the results can be assumed to be due to the new rule introduced and the instructor's direct emphasis. However, it is worth pointing out that the experience should be replicated so that the results can be generalized. According to the small variation in the values for *Usefulness*, it can be assumed that it addressed exactly its intention. However, and taking student comments into account, the question about *Fairness* was either interpreted as being too strict or too relaxed, and therefore the obtained range of answers resulted larger. Had the question been divided in two, results would have likely been more precise.

The generalizability of findings is limited to factors such as sample size, sample composition, and influence of contextual variables. In particular, the obtained results are bound to the instructional design traits of the analyzed capstone course and socio-cultural background in which it was deployed. This threat can be addressed through replication of this intervention in different student cohorts, courses, or even universities. For external reasons, the study could not be replicated in the same conditions. First, during 2019, social and political upheaval in the country forced a temporary stop on university activities; here the Software Project course started as always, but the second part was conducted online. In the following four semesters, development was completely online due to the COVID-19 pandemic. Therefore, the peer assessment and teamwork behavior in these latter semesters could not be compared with the unit of analysis studied in this paper. However, it was seen that, when working online, peer assessments tend to be systematically high with almost no variation. This requires a more thorough analysis to determine if this context particularly favors teamwork or if the grading schema must be adjusted to the new context in a way that best reflects teamwork quality.

VII. CONCLUSION AND FUTURE WORK

It is truly difficult to make college students capable of working in teams, in a way that all team members are committed and doing their fair share of work. This paper directly contributes to show a way where students can receive feedback from their peers about their work and project commitment early on, where they can still change their behavior and become active members of the team. There are many reasons why a student does not work enough or does not take the initiative for making real contributions to the team. But one of the many assumptions that they have is that the team will accept such a behavior and will not call them out for not doing their part. Consequently, those team members will not be penalized for their misbehavior, given that grades are usually assigned to the team as a whole. The peer assessment rule presented in this paper shows software engineering students enrolled in a capstone course that not being active members in their teams can have a significant negative penalization in their grade, as a way to persuade them to change their behavior. Communication among team members is key for that, as well as students being truthful about the engagement, collaboration, and commitment of their fellow team members, as well as the instructional team being clear on setting expectations regarding the grading rules.

A new grading rule for a software engineering capstone course was devised, where peer assessment had high relevance: whenever it is low, that is the grade the student gets without considering any other dimension of software development. To analyze the effect of introducing this new grading rule on teamwork quality, we conducted a three-cycle action research study. Three aspects were analyzed: (1) the progression of peer evaluations along the semester; (2) the difference of the teamwork quality between the evaluated semester, previous ones where the rule was not applied and the following semesters when the rule was emphasized from the beginning; and (3) students' perception about fairness and usefulness of the rule as a means for improving teamwork quality.

When comparing teamwork quality in the semester where we introduced the rule with the previous ones, it may seem that it worsened both, in terms of the median and the dispersion of peer assessments. However, this can be due to teamwork not being evaluated correctly in the past. Peer assessment used to have almost no variation during the semester regardless of team performance. In particular, there is no possibility to have students' opinion about comparing both rules because students only take the course once. However, end-course surveys support that there used to be complaints about free riders and outliers in the past, and this was not the same when applying the new rule.

Having students empowered for making the difference about teammates grades, makes them assume this responsibility as an opportunity for giving and receiving feedback as a means for improvement. Therefore, the variation of peer assessment values shows student involvement in grading and their interest in having an influence on the teamwork quality. Consequently, our new rule improved the accuracy of teamwork quality evaluation when compared with previous semesters.

Throughout the course, peer assessment scores were spread over a broad range of grades, so it is possible to argue that students were truthful when grading each other's work on the project. As a direct consequence, they improved their communication and coordination, as a way to receive a better peer assessment evaluation in the following iteration. This is a result of the use of the new rule, which promotes a progression toward a more cohesive, communicative, and reflexive team.

Students valued the rule, first, because they found it fair: when commitment and performance is high, students receive high grades and when it is low, students receive low grades, in contrast to what occurred in previous semesters. Second, they found it useful as it provided them with accurate information about teammates' opinions of their performance, and thus it was a motivation for improvement. It is worth noting that a team composed by 5 to 7 students provides a balance between having enough teammates to avoid particular grading bias and not adding too much complexity due to managerial work associated with teamwork. Moreover, having teams composed by more than 7 people is against agile software development recommendations that is the strategy that guides the course.

Regarding the teamwork results dimension of teamwork quality, preliminary observations showed no changes in the quality of the end product built by the teams. Customers assign high grades to the resulting product, both before and after the application of the rule. While there might be a correlation between the effect of the rule on project teamwork and teamwork results, our evidence is not conclusive and requires further research. Such a study is proposed as future work.

DATA AVAILABILITY

The anonymized dataset analyzed in this study is available for download from: https://dcc.uchile.cl/f/HqjGPjlVvf. Likewise, the R script used to run the statistical analyses is available at the same source.

CONFLICT OF INTEREST STATEMENT

Nothing to declare.

REFERENCES

[1] 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 *Proc. 40th Int. Conf. Softw. Eng., Softw. Eng. Educ. Training*, May 2018, pp. 49–57.

[2] M. C. Bastarrica, D. Perovich, and M. M. Samary, "What can students get from a software engineering capstone course?" in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng., Softw. Eng. Educ. Training Track (ICSE-SEET)*, May 2017, pp. 137–145.

[3] C. Raibulet and F. A. Fontana, "Collaborative and teamwork software development in an undergraduate software engineering course," *J. Syst. Softw.*, vol. 144, pp. 409–422, Oct. 2018.

[4] E. Britton, N. Simper, A. Leger, and J. Stephenson, "Assessing teamwork in undergraduate education: A measurement tool to evaluate individual teamwork skills," *Assessment Eval. Higher Educ.*, vol. 42, no. 3, pp. 378–397, Apr. 2017.

[5] M. Marques, S. F. Ochoa, M. C. Bastarrica, and F. J. Gutiérrez, "Enhancing the student learning experience in software engineering project courses," *IEEE Trans. Educ.*, vol. 61, no. 1, pp. 63–73, Feb. 2018.

[6] C. A. Mertler, *Action Research: Improving Schools and Empowering Educators*. Newbury Park, CA, USA: Sage, 2019.

[7] M. C. Bastarrica, D. Perovich, F. J. Gutierrez, and M. Marques, "A grading schema for reinforcing teamwork quality in a capstone course," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., Companion Proc. (ICSE-Companion)*, Montreal, QC, Canada, May 2019, pp. 276–277.

[8] Y. Lindsjørn, D. I. K. Sjøberg, T. Dingsøyr, G. R. Bergersen, and T. Dybå, "Teamwork quality and project success in software development: A survey of agile development teams," *J. Syst. Softw.*, vol. 122, pp. 274–286, Dec. 2016.

[9] *Criteria for Accrediting Engineering Programs*, ABET, Baltimore, MD, USA, 2018.

[10] 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," *Computer*, vol. 48, no. 11, pp. 106–109, Nov. 2015.

[11] B. Oakley, R. M. Felder, R. Brent, and I. Elhajj, "Turning student groups into effective teams," *J. Student Centered Learn.*, vol. 2, no. 1, pp. 9–34, 2004.

[12] B. A. Oakley, D. M. Hanna, Z. Kuzmyn, and R. M. Felder, "Best practices involving teamwork in the classroom: Results from a survey of 6435 engineering student respondents," *IEEE Trans. Educ.*, vol. 50, no. 3, pp. 266–272, Aug. 2007.

[13] D. Smarkusky, R. Dempsey, J. Ludka, and F. de Quillettes, "Enhancing team knowledge: Instruction vs. experience," *ACM SIGCSE Bull.*, vol. 37, no. 1, pp. 460–464, Feb. 2005.

[14] A.-P. Correia, "Dealing with conflict in learning teams immersed in technology-rich environments: A mixed-methods study," *Educ. Inf. Technol.*, vol. 25, no. 3, pp. 2049–2071, May 2020.

[15] J. E. Driskell, G. F. Goodwin, E. Salas, and P. G. O'Shea, "What makes a good team player? Personality and team effectiveness," *Group Dyn., Theory, Res., Pract.*, vol. 10, no. 4, pp. 249–271, Dec. 2006.

[16] G. P. Sudhakar, A. Farooq, and S. Patnaik, "Soft factors affecting the performance of software development teams," *Team Perform. Manage., Int. J.*, vol. 17, no. 3/4, pp. 187–205, Jun. 2011.

[17] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "On the unhappiness of software developers," in *Proc. 21st Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, Jun. 2017, pp. 324–333.

[18] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "Unhappy developers: Bad for themselves, bad for process, and bad for software product," in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng. Companion (ICSE-C)*, May 2017, pp. 362–364.

[19] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: Some issues and experiences," *Commun. ACM*, vol. 34, no. 1, pp. 39–58, Jan. 1991.

[20] T. Chow and D.-B. Cao, "A survey study of critical success factors in agile software projects," *J. Syst. Softw.*, vol. 81, no. 6, pp. 961–971, Jun. 2008.

[21] L. K. Michaelsen, A. B. Knight, and L. D. Fink, *Team-Based Learning: A Transformative Use of Small Groups in College Teaching*, 1st ed. Sterling, VA, USA: Stylus Publishing, 2004.

[22] M. Awatramani and D. Rover, "Team-based learning course design and assessment in computer engineering," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2015, pp. 1–9.

[23] M. Sherriff and S. Heckman, "Capstones and large projects in computing education," *ACM Trans. Comput. Educ.*, vol. 18, no. 2, pp. 1–4, Jun. 2018.

[24] D. A. Kolb, *Experiential Learning: Experience as the Source of Learning and Development*. Upper Saddle River, NJ, USA: FT Press, 2014.

[25] L. Marshall, V. Pieterse, L. Thompson, and D. M. Venter, "Exploration of participation in student software engineering teams," *ACM Trans. Comput. Educ.*, vol. 16, no. 2, pp. 1–38, Mar. 2016.

[26] J. Lave, *Cognition in Practice: Mind, Mathematics and Culture in Everyday Life*. Cambridge, U.K.: Cambridge Univ. Press, 1988.
[27] E. O. Navarro, "On the role of learning theories in furthering software engineering education," in *Instructional Design: Concepts, Methodologies, Tools and Applications*. Hershey, PA, USA: IGI Global, 2011, pp. 1645–1666.
[28] C.-Y. Chen and P. P. Chong, "Software engineering education: A study on conducting collaborative senior project development," *J. Syst. Softw.*, vol. 84, no. 3, pp. 479–491, Mar. 2011.
[29] N. Clark, P. Davies, and R. Skeers, "Self and peer assessment in software engineering projects," in *Proc. 7th Australas. Comput. Educ. Conf. (ACE)*, vol. 42. Newcastle, NSW, Australia: Australian Computer Society, Jan./Feb. 2005, pp. 91–100.
[30] A. Tafliovich, A. Petersen, and J. Campbell, "On the evaluation of student team software development projects," in *Proc. 46th ACM Tech. Symp. Comput. Sci. Educ.* New York, NY, USA: Association for Computing Machinery, Feb. 2015, pp. 494–499.
[31] N. Herbert, "Quantitative peer assessment: Can students be objective?" in *Proc. 9th Australas. Conf. Comput. Educ. (ACE)*, vol. 66. Sydney, NSW, Australia: Australian Computer Society, 2007, pp. 63–71.
[32] H. H. Smith and D. L. Smarkusky, "Competency matrices for peer assessment of individuals in team projects," in *Proc. 6th Conf. Inf. Technol. Educ.* New York, NY, USA: Association for Computing Machinery, Oct. 2005, pp. 155–162.
[33] D. Petkovic, M. Sosnick-Pérez, K. Okada, R. Todtenhoefer, S. Huang, N. Miglani, and A. Vigil, "Using the random forest classifier to assess and predict student learning of software engineering teamwork," in *Proc. IEEE Frontiers Educ. Conf. (FIE)*, Oct. 2016, pp. 1–7.
[34] T. Clear, M. Goldweber, F. H. Young, P. M. Leidig, and K. Scott, "Resources for instructors of capstone courses in computing," in *Proc. Working Group Rep. ITiCSE Innov. Technol. Comput. Sci. Educ. (ITiCSE-WGR)*. New York, NY, USA: Association for Computing Machinery, 2001, pp. 93–113.
[35] K. J. Topping, E. F. Smith, I. Swanson, and A. Elliot, "Formative peer assessment of academic writing between postgraduate students," *Assessment Eval. Higher Educ.*, vol. 25, no. 2, pp. 149–169, Jun. 2000.
[36] M. C. Bastarrica and J. Simmonds, "Gender differences in self and peer assessment in a software engineering capstone course," in *Proc. IEEE/ACM 2nd Int. Workshop Gender Equality Softw. Eng. (GE)*, Montreal, QC, Canada, May 2019, pp. 29–32.
[37] H. Asikainen, V. Virtanen, L. Postareff, and P. Heino, "The validity and students' experiences of peer assessment in a large introductory class of gene technology," *Stud. Educ. Eval.*, vol. 43, pp. 197–205, Dec. 2014.
[38] K. S. Double, J. A. McGrane, and T. N. Hopfenbeck, "The impact of peer assessment on academic performance: A meta-analysis of control group studies," *Educ. Psychol. Rev.*, vol. 32, no. 2, pp. 481–509, Jun. 2020.
[39] L. P. Hoang, H. T. Le, H. Van Tran, T. C. Phan, D. M. Vo, P. A. Le, D. T. Nguyen, and C. Pong-inwong, "Does evaluating peer assessment accuracy and taking it into account in calculating assessor's final score enhance online peer assessment quality?" *Educ. Inf. Technol.*, vol. 27, no. 3, pp. 4007–4035, Apr. 2022.
[40] J. H. Hayes, T. C. Lethbridge, and D. Port, "Evaluating individual contribution toward group software engineering projects," in *Proc. 25th Int. Conf. Softw. Eng. (ICSE)*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 622–627.
[41] D. E. Wilkins and P. B. Lawhead, "Evaluating individuals in team projects," *ACM SIGCSE Bull.*, vol. 32, no. 1, pp. 172–175, Mar. 2000.
[42] J. Chen, G. Qiu, L. Yuan, L. Zhang, and G. Lu, "Assessing teamwork performance in software engineering education: A case in a software engineering undergraduate course," in *Proc. 18th Asia–Pacific Softw. Eng. Conf.* Ho Chi Minh, Vietnam: IEEE Computer Society, Dec. 2011, pp. 17–24.
[43] K. Cho and M.-H. Cho, "Training of self-regulated learning skills on a social network system," *Social Psychol. Educ.*, vol. 16, no. 4, pp. 617–634, Dec. 2013.
[44] J. Annett, *Feedback and Human Behaviour: The Effects of Knowledge of Results, Incentives and Reinforcement on Learning and Performance*. Baltimore, MD, USA: Penguing, 1969.
[45] Z. Masood, R. Hoda, and K. Blincoe, "Adapting agile practices in university contexts," *J. Syst. Softw.*, vol. 144, pp. 501–510, Oct. 2018.
[46] M. Paasivaara, J. Vanhanen, V. T. Heikkilä, C. Lassenius, J. Itkonen, and E. Laukkanen, "Do high and low performing student teams use scrum differently in capstone projects?" in *Proc. IEEE/ACM 39th Int. Conf. Softw. Eng., Softw. Eng. Educ. Training Track (ICSE-SEET)*, May 2017, pp. 146–149.
[47] R. Wlodarski, J.-R. Falleri, and C. Parvéry, "Assessment of a hybrid software development process for student projects: A controlled experiment," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng., Softw. Eng. Educ. Training (ICSE-SEET)*, May 2021, pp. 289–299.
[48] R. Włodarski, A. Poniszewska-Marańda, and J.-R. Falleri, "Impact of software development processes on the outcomes of student computing projects: A tale of two universities," *Inf. Softw. Technol.*, vol. 144, Apr. 2022, Art. no. 106787.
[49] L. Silvestre, S. F. Ochoa, and M. Marques, "Understanding the design of software development teams for academic scenarios," in *Proc. XXXIV Int. Conf. Chilean Comput. Sci. Soc. (SCCC)*, Nov. 2015, pp. 1–6.
[50] M. Marques, "A prescriptive software process for academic scenarios," Ph.D. dissertation, Dept. Comput. Sci., Univ. Chile, Santiago, Chile, 2017.
[51] C.-Y. Chen and K.-C. Teng, "The design and development of a computerized tool support for conducting senior projects in software engineering education," *Comput. Educ.*, vol. 56, no. 3, pp. 802–817, Apr. 2011.
[52] L. van der Duim, J. Andersson, and M. Sinnema, "Good practices for educational software engineering projects," in *Proc. 29th Int. Conf. Softw. Eng. (ICSE)*. Washington, DC, USA: IEEE Computer Society, May 2007, pp. 698–707.
[53] K.-J. Stol and B. Fitzgerald, "The ABC of software engineering research," *ACM Trans. Softw. Eng. Methodol.*, vol. 27, no. 3, pp. 1–51, Jul. 2018.

MARÍA CECILIA BASTARRICA received the Ph.D. degree in computer science and engineering from the University of Connecticut, in 2000. She is currently an Associate Professor with the Department of Computer Science, University of Chile. Her research interests include software engineering, software processes, and software engineering education.

FRANCISCO J. GUTIERREZ received the Ph.D. degree in computer science from the University of Chile, in 2017. He is currently an Assistant Professor with the Department of Computer Science, University of Chile. His research interests include human factors in computing systems, empirical studies in software engineering, and computer science education.

MARÍA MARQUES received the Ph.D. degree in computer science from the University of Chile, in 2017, and the M.Ed. degree in education assessment and evaluation from the Boston College, in 2022. She is currently an Assistant Professor of the practice with the Department of Computer Science, Boston College. Her research interests include software engineering, software engineering education, and computer science education.

DANIEL PEROVICH received the Ph.D. degree in computer science from the University of Chile, in 2014. He is currently an Adjunct Professor with the Department of Computer Science, University of Chile. His research interests include software engineering, software architecture, and pre-sales requirement engineering.