Aligning Software Engineering Education with Industry Needs through a Soft Skills-Driven PBL Design

Ana Carolina Oran
Federal University of Amazonas
Manaus, Brazil
ana.oran@icomp.ufam.edu.br

Leonardo Marques
Federal University of Amazonas
Manaus, Brazil
leonardo.marques@icomp.ufam.edu.br

Rayfran Rocha Lima
Sidia Institute of Science and Technology
Manaus, Brazil
rayfran.lima@sidia.com

Williamson Silva
Federal University of Cariri
Juazeiro do Norte, Brazil
williamson.silva@ufca.edu.br

Abstract

Soft skills are essential for software engineering professionals, yet in academic settings they are often treated as incidental outcomes of Project-Based Learning (PBL) rather than as explicitly planned learning objectives. This limits both the consistency of competency development and the understanding of how pedagogical decisions influence students' perceived learning outcomes. This paper investigates how soft skills can be intentionally planned, operationalized, and intentionally integrated into a PBL-based Software Engineering course. To address this problem, we propose a competency-oriented pedagogical design that systematically maps industry-relevant competencies to pedagogical interventions, including agile practices, team-based development, reflective activities, and role rotation. A perception-based mixed-methods evaluation was conducted in an undergraduate Software Engineering course using students' self-assessment, peer recognition, responses from the educational experience questionnaire, and the instructor's reflective analysis regarding the alignment between planned activities and targeted competencies. Results indicate uneven competency development across the evaluated dimensions, with higher perceived gains in teamwork, responsibility, and problem-solving, and lower gains in skills such as delegation and self-presentation. The findings suggest that PBL environments do not uniformly foster all competencies. Instead, competency development is strongly influenced by how pedagogical activities are intentionally designed, operationalized, and aligned with targeted learning objectives. This work contributes a structured and competency-oriented approach for intentionally integrating soft skills into Software Engineering education.

Keywords

Soft Skills Development, Software Engineering Education, Project-Based Learning, Mixed-Methods Evaluation

1 Introduction

The software industry is experiencing a persistent crisis. An analysis of 530 job advertisements in New Zealand found that 82% of software engineering positions explicitly require soft skills such as communication, teamwork, leadership, responsibility, and adaptability [8]. These competencies are not peripheral concerns but foundational to effective participation in real engineering teams. Yet systematic reviews of Information Technology (IT) education document a widening gap between graduate readiness and industry expectations [9]. By 2030, Brazil alone faces an estimated 1 million unfilled IT positions, while the World Economic Forum projects that skill gaps could cost the global economy $11.5 trillion by 2028 [4]. This is not a regional anomaly but a structural problem affecting education and industry worldwide.

The roots of this gap are educational. Universities have traditionally emphasized technical problem-solving, including algorithms, design patterns, and coding proficiency, while treating soft skills as peripheral competencies or assuming they emerge naturally from technical coursework [9]. When soft skills are included in curricula, they typically occupy a single course rather than being systematically integrated throughout programs [9]. When assessment does occur, technical deliverables dominate the evaluation, and soft skills emerge as secondary concerns. Most critically, the field lacks empirical investigation into the relationship between deliberate pedagogical design and students' perceived soft skills development, and whether different types of human-related competencies, such as trainable skills, attitudes, and personality traits, respond equally to the same learning environments.

Recent work argues that what is commonly referred to as soft skills often encompasses a broader set of human-related characteristics: trainable and observable competencies such as communication and teamwork, attitudes reflecting individuals' dispositions such as motivation and responsibility, and personality traits representing more stable characteristics such as self-confidence and emotional resilience [1]. This conceptual overlap means that educational interventions often target heterogeneous elements without explicitly distinguishing their nature, making it difficult to interpret what is actually being perceived as developed. While Project-Based Learning provides contexts for collaboration and interaction, existing evidence suggests that simply exposing students to such environments does not guarantee consistent or comprehensive development across all types of competencies [9].

This research investigates how soft skills can be explicitly planned, operationalized, and embedded within a Project-Based Learning Software Engineering course, and how students' perceived development aligns with pedagogical design intent. As a result, this work proposes a competency-oriented pedagogical design model for intentionally operationalizing soft skills within Software Engineering PBL environments.

We examine which specific competencies are most strongly perceived as developed, how the intentional mapping of pedagogical activities to soft skills objectives manifests in student perceptions, and where gaps emerge between planned and perceived outcomes. We combine three complementary perspectives: students' self-assessments, peer recognition patterns, and instructor reflective analysis to construct a more complete understanding of soft skills development than single-method approaches provide.

Our contributions are threefold. **First**, we provide empirical evidence that soft skills can be explicitly planned, operationalized, and embedded within realistic Software Engineering course activities, demonstrating the alignment between intentional pedagogical design and students' perceived learning outcomes. **Second**, we show that perceived soft skills development is not uniform across all competencies. Different types of human-related characteristics, including trainable skills, attitudes, and personality traits, appear to respond differently to PBL environments, providing concrete evidence for why some competencies are naturally fostered while others require more explicit scaffolding. **Third**, we demonstrate the value of multi-perspective assessment, comparing student self-perception, peer recognition, and instructor reflection, constructing a more comprehensive understanding of soft skills development than self-report alone provides.

2 Background and Related Work

The disconnect between what industry needs and what academia provides has become increasingly urgent. A mapping study analyzing over 8,600 articles found that the primary causes of the skill gap in software engineering are the misalignment between academic curricula and industry demands, combined with rapid technological change [4]. The consequences are substantial: by 2030, Brazil alone faces an estimated 1 million unfilled IT positions, while the World Economic Forum estimates the global GDP loss from skill gaps could reach $11.5 trillion by 2028 [4]. When industry practitioners describe the competencies expected from software professionals, soft skills consistently emerge as critical requirements. An analysis of 530 job advertisements in New Zealand found that 82% explicitly require soft skills, with communication appearing among the most frequently demanded competencies [8]. Despite this scenario, systematic reviews of Software Engineering education indicate that graduates often leave universities with strong technical knowledge but limited preparation regarding interpersonal, collaborative, and communicative competencies required in professional environments [9].

Project-Based Learning (PBL) has emerged as one of the most widely adopted approaches to reduce this gap by approximating educational environments to real software engineering practice. In PBL-based Software Engineering courses, students are exposed to collaborative and realistic project scenarios involving teamwork, iterative development, communication with stakeholders, problem solving, and continuous delivery [9]. Previous experience reports have shown that PBL environments can effectively foster both technical and soft skills while approximating students to professional software engineering practice [1, 10].

However, existing evidence suggests that simply exposing students to collaborative environments does not guarantee consistent or comprehensive development across all types of competencies [9]. Although previous work has reported improvements in students' technical and soft skills through PBL experiences [10], these competencies are often treated as secondary or emergent byproducts of project execution rather than as explicitly planned learning objectives [9]. Even when soft skills are assessed, evaluation commonly relies on self-reported perceptions collected at the end of the course, often without clear alignment between pedagogical activities and targeted competencies [5]. Consequently, there is still limited understanding regarding which competencies are naturally fostered by PBL environments and which require more intentional pedagogical interventions.

Another challenge lies in the conceptual complexity surrounding the term "soft skills." Prior work argues that the concept often encompasses a broader set of human-related characteristics, including trainable competencies such as communication and teamwork, attitudes such as responsibility and motivation, and personality traits such as self-confidence and emotional resilience [1]. In educational contexts, these elements are frequently treated as interchangeable, despite potentially responding differently to pedagogical interventions. This conceptual overlap makes it difficult to understand what types of competencies are actually being developed and perceived by students throughout Software Engineering courses.

Although soft skills are widely recognized as essential in Software Engineering education, important research gaps remain. Existing studies rarely investigate how soft skills can be intentionally planned, operationalized, and systematically integrated into pedagogical activities [9]. In addition, there is still limited evidence regarding the alignment between pedagogical design decisions and students' perceived development of competencies. To address these gaps, we investigate how soft skills can be explicitly planned, operationalized, and integrated into a PBL-based Software Engineering course through intentional pedagogical interventions, combining students' self-perception, peer recognition, and instructor reflection to analyze the relationship between planned activities and perceived competency development.

3 Study Context

3.1 Course Description

The Practical Software Engineering (PSE) course, offered in the seventh semester of the Software Engineering undergraduate program with a total workload of 90 hours, aims to provide students with an integrated and hands-on experience in software systems development. Throughout the course, students engage with the complete software development lifecycle, including requirements specification, modeling, implementation, testing, and delivery. In addition, the course incorporates contemporary industry practices such as continuous integration and delivery, version control, and cloud-based environment configuration. Since students are required to complete foundational Software Engineering courses beforehand, including software analysis, design, implementation, testing, project management, and configuration management, the course adopts a predominantly practical approach focused on the execution of real-world projects.

The course is structured to approximate a professional software development environment in which students collaboratively design, develop, and deliver functional solutions. Activities include Unified Modeling Language (UML) based modeling, requirements-driven development, manual and automated testing, and the production of technical documentation to support traceability and project organization throughout the development process. To support these activities, the course adopted a Project-Based Learning (PBL) approach, organizing the pedagogical activities to encourage collaboration, student autonomy, and problem-solving in contexts aligned with real industry scenarios.

The course included three weekly meetings dedicated to project monitoring, activity alignment, and progress tracking involving instructors, teaching assistants, and students. Continuous deliveries were conducted throughout the semester, including in-person sessions for demonstrations and validation of project outcomes. Moodle was used to support course management and content distribution, while Google Drive was employed for artifact storage and collaborative document management. For project organization and activity tracking, teams were free to choose tools such as GitHub, Jira, or Trello.

The course was taught by a faculty member with a PhD in Computer Science and more than 15 years of experience in both the software industry and professional recruitment for software development teams. This background contributed to approximating academic activities to professional software engineering practices, particularly regarding the expected balance between technical and interpersonal competencies. In addition, each team was supported by a Teaching Assistant (TA) responsible for helping students resolve technical issues, remove impediments, and conduct daily meetings. The TAs were selected based on their previous performance in the course and already had professional experience in the software industry, allowing them to share practical perspectives related to decision making, conflict resolution, and project planning commonly required in professional environments.

3.2 Soft Skills-Driven PBL Design

Unlike traditional PBL and Scrum-based Software Engineering courses, our proposal does not introduce new pedagogical practices. Instead, it introduces a competency-oriented pedagogical design in which soft skills explicitly guide course planning. Rather than assuming that interpersonal competencies naturally emerge from project execution, we first identified industry-relevant competencies and then intentionally aligned learning activities and complementary pedagogical interventions to create repeated opportunities for students to exercise them throughout the semester.

The proposed design was implemented through three complementary layers: (i) a Project-Based Learning environment that provided authentic software development experiences, (ii) Scrum practices that structured project execution and collaboration, and (iii) complementary pedagogical interventions intentionally introduced to stimulate competencies that are less naturally fostered through project execution alone.

The pedagogical design was developed by first identifying the competencies most frequently demanded by the software industry based on previous studies [3, 7]. This analysis identified recurrent competencies, including communication, teamwork, leadership, problem-solving, responsibility, critical thinking, and adaptability. These competencies served as the pedagogical goals that guided the course's design.

Next, these competencies were systematically aligned with the planned course activities. Existing PBL and Scrum practices were analyzed to identify which competencies they could naturally stimulate. Whenever gaps were identified, complementary pedagogical interventions were incorporated to provide additional opportunities for students to exercise competencies that are less frequently fostered in traditional project-based courses. Finally, the resulting mapping was refined through brainstorming sessions involving two senior professors with expertise in Software Engineering and active learning methodologies to ensure pedagogical coherence, practical feasibility, and alignment with industry expectations.

The resulting mapping should therefore be interpreted as a pedagogical planning model that guided the course design rather than as a claim that individual activities directly develop specific competencies. Instead, the mapping represents the deliberately created learning opportunities throughout the semester.

The Practical Software Engineering (PSE) course adopted Project-Based Learning (PBL) as its educational approach. Students were challenged to develop innovative software systems that address real-world problems while experiencing the full software development lifecycle, from requirements elicitation to project delivery. Throughout the semester, students worked collaboratively in teams, approximating the dynamics of professional software engineering environments.

During the project definition phase, students participated in brainstorming sessions and Design Thinking activities [11] to define project ideas and elicit requirements through interviews and questionnaires. Project artifacts included user stories, UML diagrams, prototypes, and backlog management artifacts, encouraging creativity, collaboration, stakeholder communication, organization, and structured thinking.

Scrum was adopted as the project management framework supporting project execution. The course was organized into six two-week sprints comprising Sprint Planning, Daily Scrum, Backlog Refinement, Sprint Review, and Sprint Retrospective. Beyond project management, Scrum ceremonies were intentionally used as opportunities to reinforce targeted soft skills. Sprint Planning promoted planning, communication, and decision-making. Daily Scrums reinforced objective communication, active listening, accountability, and self-management. Sprint Reviews encouraged communication and presentation skills, while Sprint Retrospectives stimulated critical reflection, constructive feedback, and continuous improvement.

Throughout the semester, teams continuously developed and delivered functional software increments using Product Backlogs, Sprint Backlogs, Kanban boards, version control systems, and collaborative tools such as Trello, GitHub, and Jira. Each sprint required students to deliver functional software increments together with testing activities, technical documentation, and risk management artifacts, reinforcing responsibility, discipline, collaboration, organization, and problem-solving throughout the development process.

Although PBL and Scrum naturally create opportunities for collaboration, several complementary pedagogical interventions were intentionally incorporated to stimulate competencies that are less frequently exercised through project execution alone.

Initially, students participated in a characterization phase in which they identified their previous experiences and preferred project roles. Based on this information, teams were formed considering complementary technical backgrounds and interpersonal profiles, reproducing team composition practices commonly adopted in software companies.

To broaden students' experiences across different responsibilities, project roles (Scrum Master, Product Owner, Developer, and Tester) were rotated at the end of each sprint. This strategy aimed to stimulate competencies such as leadership, delegation, adaptability, collaboration, and self-confidence by allowing every student to experience different responsibilities throughout the project.

Responsibility was further reinforced through an attendance policy that directly affected sprint grades. This strategy sought to approximate the consequences commonly observed in professional software development teams, where the absence of team members directly affects project progress and team performance.

Reflection activities were incorporated throughout the semester. After each Sprint Review, teams conducted guided Sprint Retrospectives describing successful practices and opportunities for improvement. Additionally, students completed individual self-reflection activities to encourage self-awareness, personal development, and continuous learning.

Knowledge sharing was promoted through public Sprint Reviews, where all teams presented their projects and observed the work of other groups. This strategy encouraged collaborative learning, exchange of experiences, critical thinking, and exposure to different technical and organizational solutions.

A recognition mechanism inspired by industry practices rewarded students who stood out in specific project roles, reinforcing professional behaviors associated with leadership, engagement, responsibility, and professional identity.

Finally, the course concluded with a public project showcase evaluated by software industry professionals and Computer Science faculty members. Besides technical quality, the evaluation also considered communication, argumentation, presentation quality, and project relevance. By exposing students to situations similar to industrial project validation meetings, this activity reinforced competencies such as persuasion, communication, self-confidence, and professional posture.

Table 1 summarizes the resulting competency-oriented pedagogical design by relating PBL activities, Scrum practices, and complementary pedagogical interventions to the soft skills they were intentionally designed to stimulate.

3.3 Participants

At the beginning of the academic semester, the course had 19 enrolled students. However, after approximately one month of activities, three students dropped the course while the projects were already in progress. This situation provided the teams with an experience similar to real-world software industry environments, where unexpected changes in team composition can directly affect project planning and execution. As a result, students had to reorganize responsibilities, redistribute tasks, and adapt sprint planning to ensure the continuity of development activities and the delivery of the projects within the established deadlines. By the end of the course, 16 students remained, organized into teams of 4.

During the initial job interview activity at the beginning of the course, we identified that most participants had prior experience in software development environments. Among the 16 participants, 81.3% (13) had previously worked or were currently working in software development companies, while 18.7% (3) reported having no prior professional experience. Among participants with industry experience, their professional backgrounds ranged from 1 to more than 3 years. Regarding roles previously performed in software development projects, most participants had experience as developers, accounting for 43.8% (7) of the sample. Additionally, 25% (4) of the participants reported experience in software testing activities, while 12.5% (2) had experience in roles related to Requirements Engineering or Product Owner activities. Participants with experience as Scrum Masters and process quality roles were also identified, each representing 6.3% (1) of the participants.

Regarding familiarity with Scrum, most participants had already used the framework in professional environments. Specifically, 62.5% (10) reported previous practical experience with Scrum in industry settings, 18.7% (3) had contact with Scrum only within the academic context of the course, and 18.7% (3) reported having only theoretical knowledge of the agile approach without prior practical experience.

Throughout the project development process, students assumed rotating roles inspired by real-world Software Engineering environments, including *developer*, *tester*, *product owner*, and *scrum master*. This rotation strategy was intentionally designed to broaden students' practical experience and to stimulate the development of diverse interpersonal and organizational skills throughout the course. Due to the reduced team size, students often ended up performing multiple roles throughout the project execution.

3.4 Projects

The projects developed throughout the course were proposed and selected by the students themselves through brainstorming sessions and group discussions, and were then approved by the course instructor. Rather than focusing exclusively on technical implementation, the projects were intentionally connected to real-world social, regional, and organizational problems. This approach sought to create learning environments that expose students not only to traditional software engineering challenges but also to interpersonal, ethical, organizational, and communication-related situations commonly found in real software development contexts.

The diversity of project contexts played an important role in operationalizing soft skills within the discipline. The projects addressed themes such as accessibility, regional culture, digital inclusion, local commerce, transportation logistics, and preservation of cultural heritage in the region. These characteristics naturally required students to interact with different types of user needs, reflect on the social and cultural implications of software solutions, negotiate decisions within teams, and deal with situations involving uncertainty, empathy, and collaborative problem-solving.

Some projects involved accessibility and inclusion concerns, requiring students to understand the needs of users with visual, auditory, and motor disabilities, thereby stimulating empathy, ethical

Table 1: Competency-Oriented Pedagogical Design: Relationship Between Course Activities and Target Soft Skills

Course Activity	Target Soft Skills
1. Project-Based Learning (PBL) Practices	
Brainstorming sessions for project definition	Creativity, collaboration, communication, initiative
Design Thinking and user interviews	Empathy, active listening, communication
Real-world software development project	Motivation, willingness to learn, personal development
Team-based software development	Teamwork, collaboration, communication, empathy, responsibility
Execution of individual tasks	Autonomy, initiative, working independently
Problem-solving during development	Problem solving, critical thinking, critical judgment, creativity
Managing requirement changes	Adaptability, flexibility, adaptation to change, emotional resilience
Functional testing activities	Critical thinking, attention to detail, responsibility
Testing reports and technical documentation	Communication, organization, organizational skills
Risk management	Problem solving, decision making
2. Scrum Practices	
Sprint Planning (Sprint Backlog, User Stories, Prototypes)	Planning, organization, communication, decision making, discernment, time management
Product Owner alignment	Communication, responsibility, empathy, persuasion
Product Backlog management	Planning, organization, prioritization, self-management
Daily Scrum	Communication, objectivity, active listening, proactivity, self-management
Incremental development with continuous delivery	Responsibility, discipline, commitment, resilience, working under pressure
Sprint Backlog status reports	Transparency, responsibility, organization
Sprint Review demonstrations	Communication, persuasion, self-confidence, self-presentation
Sprint Retrospectives	Critical thinking, collaboration, communication
3. Competency-Oriented Pedagogical Interventions	
Team formation based on complementary profiles	Collaboration, communication, empathy
Role rotation across Scrum roles	Leadership, delegation, persuasion, adaptability, self-confidence
Attendance policy linked to sprint assessment	Responsibility, commitment, attitude
Individual self-reflection	Personal development, self-awareness, initiative
Cross-team project presentations	Motivation, collaboration, critical thinking
Recognition awards for role performance	Motivation, engagement, professional identity, responsibility, self-confidence
Final project showcase with industry professionals and faculty	Communication, persuasion, self-confidence, professional presentation, argumentation

responsibility, and critical thinking about inclusive software design. Other projects exposed students to contexts with users at different levels of digital literacy and distinct socioeconomic realities, fostering communication skills, active listening, adaptability, and negotiation skills. Additionally, projects related to regional transportation and the preservation of cultural narratives required students to reflect on culturally rooted practices and social habits, promoting critical thinking, creativity, and sensitivity to regional and cultural aspects.

The four projects developed during the course included: (i) an accessible tourism platform focused on the regional context; (ii) a marketplace platform for local commerce; (iii) a transportation support platform addressing challenges related to regional transportation; and (iv) a digital platform for the preservation and sharing of cultural narratives. Although distinct in their domains, all projects shared characteristics aligned with realistic software engineering scenarios, including collaboration, user-centered thinking, iterative development, stakeholder communication, and continuous decision-making.

From a technical perspective, the teams adopted different technologies and software engineering tools tailored to each project's needs. The solutions involved web and mobile development technologies such as Node.js, React.js, Next.js, Flutter, Dart, Firebase, Sequelize, PlanetScale, and Vercel. In addition, students used agile project management and collaboration tools, including Trello, Google Meet, Google Drive, and Figma, for prototyping and team coordination.

3.5 Research Design

This study investigates how soft skills can be intentionally planned, operationalized, and reflected upon within a Project-Based Learning (PBL) Software Engineering course. To guide this investigation, the following research questions (RQ) were defined: **RQ1.** How can soft skills be explicitly planned and operationalized within a Project-Based Learning (PBL) Software Engineering course? **RQ2.** To what extent do students perceive the development of the soft skills intentionally targeted in the course design? **RQ3.** How does the instructor's reflection indicate alignment between the planned soft skills and the enacted learning activities?

This study does not aim to establish causal relationships between specific pedagogical interventions and the development of soft skills. Instead, it investigates the perceived alignment between pedagogical planning, implemented learning activities, and students' perceived competency development within a PBL environment.

Figure 1 presents an overview of the conceptual flow adopted in this study. Industry demands informed the definition of target soft skills, which were translated into pedagogical interventions and intentionally integrated into PBL activities. The observed and perceived competencies were then analyzed using multiple sources of evidence, including students' self-assessments, peer recognition, and instructor reflections. Finally, the identified findings and reflections contribute to a feedback loop intended to support improvements in future course editions.

Data collection was conducted at the end of the semester, after students had completed all project-related activities. This moment was intentionally selected to ensure that participants had experienced the entire software development cycle and had sufficient opportunities to practice and reflect upon the targeted soft skills.

Students were invited to complete two questionnaires: a self-assessment questionnaire and an educational experience questionnaire. Both questionnaires were made available online through Google Forms. Prior to their application, the instruments were reviewed and validated by other researchers involved in the study.

The first questionnaire was designed to capture students' self-assessment regarding the development of the soft skills explicitly defined during the pedagogical planning phase. Additionally, students were invited to identify the team members who stood out in each project role performed throughout the course.

The second questionnaire aimed to evaluate students' self-assessments of the educational activities implemented during the course. The instrument included both open-ended and closed-ended questions. The closed-ended questions used a five-point Likert scale, as in [2].

Ethical considerations were taken into account to ensure the integrity of the study. Participation was voluntary, and students were informed of the research's purpose and how the collected data would be used. All responses were collected anonymously, and no personally identifiable information was recorded. All participants signed an informed consent form agreeing to participate in the study.

The analysis of the collected data was conducted using a mixed-methods approach, combining quantitative and qualitative methods to obtain a broader understanding of participants' self-assessments and experiences throughout the course.

Quantitative Analysis. The quantitative data obtained from the questionnaires were analyzed using descriptive statistics. For each soft skill, measures such as mean, median, and standard deviation were calculated to summarize students' self-perceived development. Additionally, frequency distributions were used to identify the soft skills most frequently reported as developed. Aggregations were performed at different levels, including overall results and, when applicable, by team or project role (e.g., developer or Scrum Master), to identify potential patterns in the data.

Qualitative Analysis. The qualitative analysis was conducted based on responses to the open-ended questionnaire questions using thematic analysis. Initially, participants' responses were read independently by the researchers, and meaningful excerpts were assigned initial codes capturing their central ideas. Subsequently, related codes were iteratively grouped into broader themes representing recurrent perceptions regarding the educational experience. The resulting themes and coding decisions were discussed and reviewed by three researchers until consensus was reached. For example, statements such as 'I had an experience very similar to what I experience daily in my company" (P3), It was an excellent experience because it provided a simulation of a software engineer in practice" (P4), and It provided an environment similar to what happens in a real workplace" (P6) were initially coded as *similarity to industry*, *simulation of professional practice*, and *industry-like environment*, respectively. These codes were subsequently grouped under the broader theme *professional realism*, which was later used to interpret participants' perceptions regarding the authenticity of the learning experience. The complete coding scheme (participant response, initial code, and theme) is available in the supplementary material.

Data Analysis Process. To maintain clarity and alignment with the research questions, the collected data were analyzed from two complementary perspectives: (i) students' self-assessment and educational experience perceptions and (ii) instructor reflection regarding pedagogical alignment. These perspectives were initially analyzed independently to avoid potential conflicts between self-reported perceptions and interpretations derived from the pedagogical planning.

After the independent analyses, a cross-perspective analysis was conducted to identify relationships between the planned pedagogical interventions and the perceived development of soft skills. This stage involved comparing the soft skills explicitly targeted during course planning with those most frequently reported by students, and contrasting these findings with the instructor's reflective evaluation. The goal was to identify convergences (alignment between planning and perception) and divergences (misalignments or unexpected outcomes), providing insights into both the effectiveness and limitations of the pedagogical design.

4 Results

4.1 Students' Self-Assessment of Soft Skills Development

Teamwork emerged as the most strongly perceived soft skill, with 100% of students reporting its development throughout the course. Responsibility (81.2%) and problem solving (75.0%) were also highly reported by participants. A second group of skills, including personal development and adaptability to change, accounted for 68.8% of responses each, while proactivity and willingness to learn were identified in 62.5% of responses.

A third group of skills showed moderate levels of perceived development, including communication, initiative, collaboration, and working under pressure, all reported by 56.2% of the students. Additionally, critical judgment, working independently, time management, creativity, planning skills, and adaptability were identified by 50.0% of the participants. In contrast, leadership skills, listening skills, and attitude were reported by 37.5% of the students, while delegation (18.8%), self-presentation (18.8%), and discernment (6.2%) appeared with the lowest frequencies.

Table 2 presents the distribution of perceived development across all evaluated soft skills, using a radar-chart-inspired aggregation. The visualization highlights the concentration of higher frequencies within a subset of competencies and the uneven distribution of responses across the evaluated skills.

In addition to self-assessment data, peer-recognition data collected from the 16 participants regarding project roles (e.g., Product Owner, Scrum Master, Designer, Developer, and Tester) revealed patterns of individual prominence within the teams.

The results indicate that certain participants were consistently recognized as strong contributors in specific roles. In contrast, roles such as Product Owner and Tester exhibited a more distributed

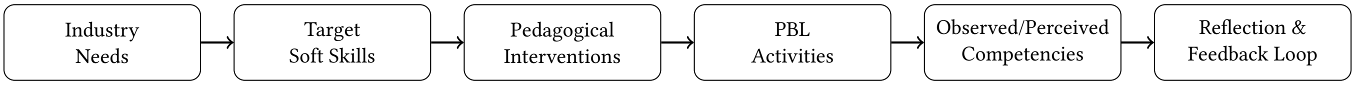

Figure 1: Conceptual overview of the soft skills-driven PBL design.

Table 2: Frequency of soft skills perceived by students

Soft Skills	Freq.	(%)
Teamwork	16	100.0
Responsibility	13	81.2
Problem Solving	12	75.0
Personal Development, Adaptability to Change	11	68.8
Proactivity, Willingness to Learn	10	62.5
Communication Skills, Initiative, Collaboration, Working Under Pressure	9	56.2
Critical Judgment, Working Independently, Time Management, Creativity, Planning Skills, Adaptability	8	50.0
Emotional Resilience, Motivation, Critical Thinking, Flexibility	7	43.8
Leadership Skills, Listening Skills	6	37.5
Attitude, Fast Learning, Self-Management, Organizational Skills	5	31.2
Persuasion, Resilience, Self-Confidence, Empathy, Decision Making, Ethics	4	25.0
Delegation, Self-Presentation	3	18.8
Discernment	1	6.2

recognition pattern, with multiple participants receiving similar frequencies of mentions, typically ranging from 1 to 2 recognitions. It is important to note that role rotation was adopted throughout the sprints to ensure that all students experienced different project responsibilities during the course.

Furthermore, some participants were recognized across multiple roles, suggesting repeated peer recognition of their contributions in different project contexts.

The Tester role, in particular, presented a highly distributed recognition pattern, with most participants receiving only a single mention. This distribution is consistent with the qualitative responses in the dataset, in which testing activities were frequently described as collaborative efforts rather than tasks assigned to a single individual.

Overall, the results suggest a concentration of perceived development in a subset of soft skills, particularly those associated with teamwork and continuous project execution, while other competencies were less frequently recognized by students. Additionally, peer recognition data indicate that some individuals stood out in specific roles, whereas other activities were perceived as more collectively distributed among team members.

4.2 Students' Self-Assessment of Educational Experience

The educational experience questionnaire assessed students' agreement with statements regarding challenges, realism, learning, and preparation for professional practice.

Figure 2 presents an overview of participants' perceptions regarding their experience during the practical project. The results indicate high levels of agreement across most evaluated dimensions.

The most strongly perceived aspect was the promotion of interaction among participants, with 87.5% of students reporting strong agreement. This was followed by the perceived relevance and up-to-date nature of the course content, as well as the presence of challenges throughout the project, all of which reached 81.3% of strong agreement. Additionally, 75.0% of participants strongly agreed that the content was presented clearly and precisely and that the experience was relevant to their learning process.

The results also indicate positive perceptions regarding the professional preparation provided by the course. Approximately 62.5% of students strongly agreed that they felt more prepared to participate in real industry projects, while 56.3% strongly agreed regarding the realism of the challenges experienced during the project and the motivation maintained throughout the development activities. Furthermore, 50.0% of participants strongly reported experiencing the adrenaline typically associated with real-world software development projects.

The qualitative responses reinforce these findings. Several participants highlighted that the course provided an experience close to professional reality, especially through the need to work in teams, handle delivery pressure, organize sprints, and solve problems. One participant stated that the course provided a simulation of a software engineer in practice, emphasizing teamwork, the different project roles, and the simultaneous development of technical and interpersonal skills. Another student reported that the experience was "very similar to what I experience daily in my company", reinforcing the perception of proximity to real industry contexts.

The reports also indicate that the practical project was perceived as a meaningful opportunity for applied learning. Among the most frequently mentioned strengths were the opportunity to learn through practice, work collaboratively, manage time, and assume different responsibilities throughout the project. One participant highlighted as a positive aspect the opportunity to "closely experience the struggles of developing a project from scratch", including teamwork, people management, interaction with external teams, and the use of agile methodologies. Another student stated that the project "simulates what is expected from students in the job market", contributing to reducing the transition gap between university and industry.

The qualitative data also demonstrate that the experience fostered the development of several soft skills intentionally planned throughout the course, particularly teamwork, adaptability, responsibility, time management, and problem-solving. One participant described as a major learning outcome that "things do not always happen as planned, and we need to adapt to changes". Another student highlighted the importance of "having more patience and assertiveness when dealing with people and differing opinions", evidencing the development of interpersonal competencies related to collaboration and communication.

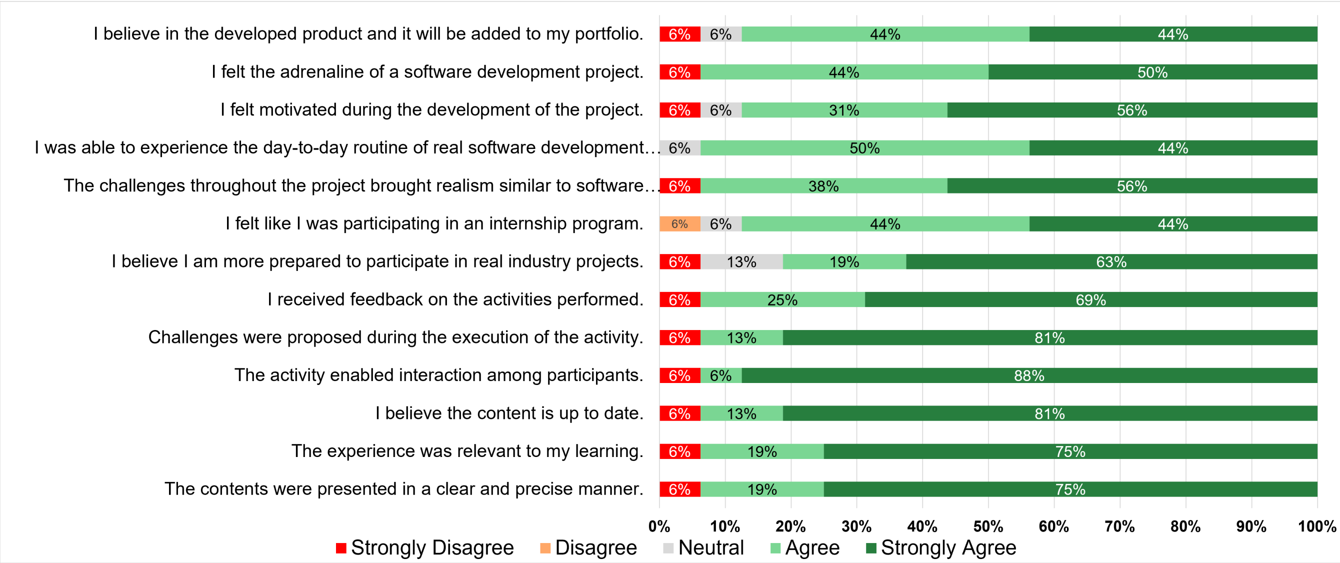

Figure 2: Participants' perceptions regarding the practical project.

A second group of aspects showed moderate levels of strong agreement, including receiving feedback on the performed activities (68.8%), feeling prepared for real industry projects (62.5%), and perceiving the realism of the challenges faced during the project (56.3%). These findings also appear in the open-ended responses, in which students reported difficulties with sprint definition, activity delays, distributed team management, requirement changes, and issues with technologies and development environments.

The reported difficulties indicate that the challenges extended beyond technical aspects to include organizational and human-related issues. Among the most frequently mentioned challenges were time management, prioritization, difficulties with agile methodologies, and coordination among team members. In addition, some students highlighted the impact of workload intensity and delivery pressure within the established schedule. One participant reported that "the workload is very heavy", while another mentioned the need for "feature removal to meet the deadline".

Regarding suggestions for improvement, students mainly recommended increasing the time available for the project, including adding an extra sprint, scheduling technical support lectures earlier, and integrating testing teams earlier. One participant suggested that "one additional sprint would greatly help with activity and time management". Another participant emphasized that earlier involvement of testers could help fix more bugs and improve the quality of the product.

Suggestions also emerged to increase the course's realism, including the use of real or simulated clients with concrete demands. According to one participant, this approach would allow students to experience challenges closer to industry practice, including alignment of expectations, negotiation of requirements, and continuous communication with stakeholders. Nevertheless, the student also acknowledged that such an approach could significantly increase both the complexity and workload of the course.

On the other hand, aspects more directly related to complete professional immersion presented moderate levels of strong agreement. Approximately 43.8% of students strongly agreed that they felt they were experiencing something similar to an internship, engaging in the routine of real software projects, and believing that the developed product could become part of their professional portfolio. These results appear to be associated with time constraints in the course, project scope limitations, and the need to reduce or adapt functionality to meet the deadline.

4.3 Instructor Reflection on Pedagogical Alignment

To better understand how the course's pedagogical planning influenced the development of soft skills across the projects, we present the course instructor's reflective statement.

"I have been teaching this course for the past four years, and over time I realized that students usually completed it with strong technical competencies, while soft skills tended to emerge more indirectly throughout the process. Based on this perception, I decided to structure the course more intentionally to explicitly encourage the development of interpersonal and collaborative competencies alongside technical learning.

Throughout the course, Scrum-related activities contributed significantly to students' soft skills development. Ceremonies such as Sprint Planning and Retrospectives proved valuable not only for project organization but also as opportunities to strengthen communication, teamwork, negotiation, and decision-making. During the first sprint, many students struggled with organization, communication, and task responsibility. However, as the semester progressed, the teams demonstrated greater autonomy, stronger collaboration, and more engagement in collective activities.

The final presentations also helped students develop communication skills, self-confidence, and professional posture. By presenting projects *to industry professionals and invited faculty members, students experienced situations similar to real professional environments, which contributed to noticeable improvements in confidence and maturity throughout the course.*

For future editions, I believe it is important to include more explicit interventions focused on competencies that appeared less frequently in students' perceptions, such as leadership, empathy, delegation, and self-management, as well as improving continuous feedback mechanisms during the semester."

5 Discussion

The findings suggest that soft skills development in Project-Based Learning environments is not merely a spontaneous consequence of collaborative project execution, but is strongly influenced by how pedagogical activities are intentionally designed and operationalized throughout the course. Competencies that were recurrently exercised in Scrum ceremonies, collaborative decision-making, iterative deliveries, and team coordination were more consistently perceived by students, indicating an alignment between pedagogical intention and perceived development. This reinforces previous discussions in Software Engineering education literature arguing that exposure to teamwork alone is insufficient to guarantee comprehensive soft skills development [9].

At the same time, the uneven distribution observed across competencies indicates that different types of human-related characteristics respond differently to the same educational environment. Skills associated with frequent social interaction and continuous project execution, such as teamwork, communication, responsibility, and problem-solving, tended to emerge more naturally within the course's dynamics. In contrast, competencies related to delegation, self-presentation, discernment, and self-confidence appeared less prominently, even when present in the pedagogical planning. This suggests that some competencies may require more explicit scaffolding, targeted feedback, or dedicated interventions to become observable and meaningful to students.

These findings also support the argument that the term "soft skills" encompasses heterogeneous dimensions, including trainable competencies, attitudes, and more stable personal traits [6]. The results indicate that the pedagogical approach adopted in the course was more effective at fostering competencies associated with behavioral and attitudinal dimensions than those linked to deeper, more reflective personal characteristics. This distinction helps explain why some competencies were consistently recognized across different evaluation perspectives, while others remained less perceptible within a single semester experience.

The peer recognition results complement this interpretation by showing that certain roles and competencies were perceived as strongly individual, whereas others emerged as collective responsibilities distributed across the team. This pattern highlights an important characteristic of collaborative Software Engineering education: not all contributions are equally visible at the individual level. Competencies associated with leadership, coordination, and communication tend to yield clearer individual recognition, while activities such as testing, collaboration, and collective problem-solving may be diluted within team dynamics. In this sense, combining self-perception, peer recognition, and instructor reflection contributed to a broader understanding of how competencies manifested throughout the projects.

Responses to the educational experience questionnaire further indicate that realistic, industry-oriented environments can increase engagement, responsibility, and awareness of professional practices. The integration of Scrum ceremonies, incremental deliveries, role rotation, public presentations, and interaction with industry professionals created conditions that approximated several dimensions of real software projects. However, the results also suggest limits to full professional immersion, particularly in aspects related to long-term accountability, product ownership, and confidence in the solutions developed. This reflects an inherent tension in Software Engineering education: while academic environments can approximate industrial practice, constraints such as limited course duration, academic schedules, and reduced project scope affect how deeply certain professional experiences can be reproduced.

Although the observed alignment between the pedagogical design and students' reported competencies suggests that the proposed course created meaningful learning opportunities, these findings should be interpreted with caution. The reported competencies reflect students' perceptions regarding their own development rather than objective evidence of competency acquisition. Therefore, the results indicate how students experienced the learning environment and perceived their competency development, rather than demonstrating that each targeted competency was effectively acquired.

Taken together, these findings suggest that soft skills development in Software Engineering education benefits from intentional pedagogical alignment between targeted competencies, learning activities, and reflective practices. Rather than treating soft skills as secondary outcomes of project execution, the results indicate the importance of explicitly operationalizing which competencies are expected, how they are stimulated, and how their development becomes visible to students and instructors. From a practical perspective, the study also suggests that different competencies may require different pedagogical strategies, reinforcing the need for diversified interventions rather than assuming homogeneous development across all human-related competencies within PBL environments.

This study presents threats to validity in accordance with established empirical Software Engineering guidelines [12]. The study relies on students' perceptions collected at the end of the course, which may have been influenced by factors such as motivation, team dynamics, and prior experience. In addition, the absence of a control group limits the ability to attribute the observed outcomes exclusively to the pedagogical design. The study also presents construct-related limitations, since the term "soft skills" was used to describe competencies related to interpersonal and behavioral dimensions, while the evaluated items in this work additionally include attitudes and personal characteristics [6], potentially affecting the interpretation of what was measured. To mitigate these limitations, the study combines multiple sources of evidence, including self-assessment, peer recognition, and instructor reflection, while explicitly acknowledging the broader interpretation of soft skills adopted throughout the analysis.

The study was conducted in a single course with a limited number of participants (n=16), which restricts the generalizability of the findings. Furthermore, contextual factors such as instructor experience, course structure, and institutional environment may have influenced the results. Reliability may also be affected by the use of self-reported data and possible differences in how participants interpreted questionnaire items. To reduce these effects, standardized instruments, consistent data collection procedures, and complementary peer-recognition data were adopted. Despite these limitations, the pedagogical approach is described in sufficient detail to support replication and provide a transparent and consistent analysis of the results.

6 Lessons Learned

One of the main lessons learned from this study is that preparing Software Engineering students for professional practice requires course design to be informed by industry needs. While technical competencies remain essential, the findings indicate that soft skills require the same level of intentional planning throughout the pedagogical design. Rather than assuming that competencies such as communication, teamwork, leadership, and responsibility naturally emerge from PBL- and Scrum-based environments, educators should explicitly define the competencies to be developed and intentionally create repeated opportunities for students to practice them throughout the course.

A second lesson concerns the relationship between pedagogical planning and students' perceived development of competency. Competencies addressed through a greater number and diversity of learning activities, such as teamwork, responsibility, communication, and problem-solving, were also the ones most frequently perceived by students. In contrast, competencies such as leadership, delegation, and self-management appeared less prominently, suggesting that they require more targeted pedagogical interventions. Although these findings do not establish causality, they indicate that increasing opportunities to practice specific competencies may positively influence students' perceptions of their development.

7 Conclusion and Future Work

This study investigated how soft skills can be intentionally planned, operationalized, and perceived within a PBL Software Engineering course. The results indicate that aligning pedagogical activities with explicitly defined competencies enables a more structured and observable development of soft skills, particularly those associated with collaboration, responsibility, communication, and problem-solving.

While previous PBL experience reports have shown that project-based courses can foster both technical and soft skills, this work advances that line of research by proposing a soft skills-driven PBL design in which competencies explicitly guide course planning and are systematically aligned with pedagogical interventions. By combining self-assessment, peer recognition, and instructor reflection, the study provides evidence of alignment between pedagogical design intent and students' perceived competency development.

The findings also challenge the assumption that PBL environments foster all soft skills uniformly. Competencies strongly associated with continuous collaboration and iterative project execution were more consistently perceived by students, whereas competencies such as leadership, delegation, self-presentation, and discernment appeared less prominently, suggesting that different competencies require different levels of pedagogical scaffolding, visibility, and intentional stimulation.

Future course offerings and subsequent research will focus on extending the proposed soft skills-driven PBL design in three main directions. First, we intend to redesign the pedagogical plan to incorporate more explicit strategies targeting competencies that were less consistently perceived in this study, particularly leadership, delegation, and self-management-related competencies. Second, we plan to adopt a more structured 360-degree evaluation approach integrating self-assessment, peer assessment, and instructor evaluation throughout the semester. Finally, future studies will explore incorporating behavioral evidence, such as development artifacts and interaction logs, to complement perception-based data and enhance the robustness of competency evaluation.

Artifact Availability

To ensure transparency and reproducibility, all artifacts used in this study are publicly available at: https://osf.io/v74nz/overview?view_only=06c9a95b36f84068ab67e40797ccdb7d.

Acknowledgments

We thank all the participants in the empirical study and the SPHERE Research Group members for their support. We would like to thank the financial support granted by CNPq - 313137/2025-0; CNPq - CNPq 406506/2025-6; CAPES- Financing Code 001; Amazonas State Research Support Foundation - FAPEAM - through POSGRAD 2026-2027. We declare that Generative AI tools were used to support grammar revision and improve textual clarity, as well as to revise English spelling during manuscript writing and editing. All content was reviewed and validated by the authors.

References

[1] Allysson Allex Araújo, Marcos Kalinowski, and Maria Teresa Baldassarre. 2025. Embracing experiential learning: Hackathons as an educational strategy for shaping soft skills in software engineering. In *2025 IEEE/ACM 37th international conference on software engineering education and training (CSEE&T)*. IEEE, 319–324.

[2] Seung Youn Chyung, Katherine Roberts, Ieva Swanson, and Andrea Hankinson. 2017. Evidence-based survey design: The use of a midpoint on the Likert scale. *Performance improvement* 56, 10 (2017), 15–23.

[3] Rayane Moraes da Silva, Ricardo Ferreira Vilela, and Pedro Henrique Dias Valle. 2023. Soft Skills of Software Developers: Exploring the Pandemic's Impact of COVID-19. In *Proceedings of the XXII Brazilian Symposium on Software Quality*. 100–109.

[4] W. Diniz, Marcelo Valença, Cesar França, A. Santos, and M. Pincovsky. 2024. The Skill Gap in Software Industry: A Mapping Study. In *Proceedings of the 38th Brazilian Symposium on Software Engineering (SBES '24)*. ACM/SBC. doi:10.1145/3663340.3663414

[5] Álex Escola-Gascóna and Josep Gallifa. 2022. How to measure soft skills in the educational context: psychometric properties of the SKILLS-in-ONE questionnaire. *Studies in Educational Evaluation* 74 (2022), 101155. doi:10.1016/j.stueduc.2022.101155

[6] Italo Cesidio Fantozzi, Luca Martuscelli, Sebastiano Di Luozzo, and Massimiliano M Schiraldi. 2024. Soft skills, attitudes, and personality traits: how does the human factor matter? A systematic review and taxonomy proposal through ProKnow-C methodology. *Businesses* 4, 2 (2024), 156–176.

[7] César França and Diego Mellet. 2016. Soft skills required! uma análise da demanda por competências não-técnicas de profissionais para a indústria de software e serviços. *Anais do IX Fórum de Educação em Engenharia de Software (FEES 2016)* (2016), 101–112.

[8] Matthias Galster, Antonija Mitrović, Sanna Malinen, and Jay Holland. 2022. What Soft Skills Does the Software Industry Really Want? An Exploratory Study of Software Positions in New Zealand. *ESEM '22: Proceedings of the 16th ACM/IEEE*

International Symposium on Empirical Software Engineering and Measurement (September 2022), 19–23. doi:10.1145/3544902.3546247 Open Access.

[9] Farhad Sadik Mohammed and Fezile Ozdamli. 2024. A Systematic Literature Review of Soft Skills in Information Technology Education. *Behavioral Sciences* 14, 10 (2024), 894. doi:10.3390/bs14100894

[10] Ana Carolina Oran, Rayfran Rocha Lima, Bruno Gadelha, Nayane Maia, Williamson Silva, and Luis Rivero. 2023. Empowering technical skills and soft skills in software engineering students through problem-based learning. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 348–357.

[11] Jéssyka Vilela and Carla Silva. 2023. An experience report on the use of problem-based learning and design thinking in a requirements engineering postgraduate course. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*. 432–441.

[12] Claes Wohlin, Per Runeson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in Software Engineering*. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-29044-2