Journal of Software Engineering Research and Development, 2026, 14:21, doi: 10.5753/jserd.2026.7617

Interplay of Technical and Interpersonal Competency Development in Software Engineering through Project-Based Learning

Victória Cavalcante [**Universidade Federal do Acre** | ***victoria.cavalcante@sou.ufac.br***]
André Cacau [**Universidade Federal do Acre** | ***andre.cacau@sou.ufac.br***]
Christopher Saar [**Universidade Federal do Acre** | ***christopher.saar@sou.ufac.br***]
Alessandreia de Oliveira [**Universidade Federal de Juiz de Fora** | ***alessandreia.oliveira@ufjf.br***]
Catarina Costa [**Universidade Federal do Acre** | ***catarina.costa@ufac.br***]

Abstract
The combination of technical and interpersonal skills is essential for success in Software Engineering. Effective teaching methodologies play a crucial role in the development of these competencies. This study investigates how Project-Based Learning (PBL), integrated with agile practices, influences the interplay between the development of these competencies among Computer Science students. Building on a structured evolution of previous research, this extension adopts a mixed-methods approach with complementary perspectives. To evaluate students' self-perception, a pre-test/post-test questionnaire was applied, complemented by structured interviews with the students and faculty, allowing for a multifaceted and integrated understanding of the training process. Statistical analysis revealed significant improvements in 100% of the assessed technical competencies (hard skills) and 50% of the interpersonal ones (soft skills). For the interview phase, participants were purposively selected based on discrepancies in their self-assessments, particularly those reporting substantial growth in soft skills while indicating a decrease in their development between the initial and final stages of the course. The faculty perspective offered a crucial complementary view on the interdependence of these developments, corroborating technical maturation and identifying the impact of cognitive biases, such as the Dunning–Kruger effect. Findings indicated that the methodology focused on group-based practical activities and projects fostered students' confidence in their abilities. Furthermore, the investigation of the Dunning–Kruger effect highlighted discrepancies between students' perceived and actual levels of knowledge. Qualitative data detail how immersive activities and instructional guidance facilitate the transition from theory to solving real-world engineering challenges. Finally, this study reinforces the effectiveness of integrating technical and behavioral developments within PBL environments to promote robust professional training in the computing field.

Keywords: *Software Engineering, Education, Soft Skills, Hard Skills, Computer Students, Active Methodologies*

1 Introduction

Technical skills in the Information Technology (IT) area, commonly referred to as hard skills, comprises quantifiable competencies validated through professional certifications or through objective results in specific tasks, are essential in the formation of a good professional (Giani de Resende et al., 2021). However, behavioral and personality competencies, which influence how a professional communicates, collaborates, and solves problems in work environments, termed soft skills, are essential, not only for initial employability but also for long-term career sustainability in dynamic work environments (Silva et al., 2020). From the initial stages of recruitment, soft skills are prominently featured in IT job requirements, signaling their pivotal role and increasing weight in the labor market (Silva and Albuquerque, 2023). Garousi et al. (2020) analyzed that even for internship positions, organizations are prioritizing candidates with demonstrable soft skill proficiency, viewing these competencies as essential indicators of professional potential. It is therefore evident that the market increasingly prioritizes individuals who balance robust hard skills with effective soft skills, reflecting a broader shift toward holistic professional evaluation.

Mendes et al. (2019) conducted a study in a degree program with Software Engineering (SE) in its curriculum, aiming to identify students' expectations regarding SE learning, expressing a clear demand for experiential learning that simulates the day-to-day difficulties encountered within the professional software industry. However, as highlighted by Souza et al. (2020), combining theoretical learning with practical application is the greatest challenge in SE. To address this, active learning methodologies, such as Project-Based Learning (PBL), have demonstrated significant efficacy in knowledge consolidation, recommending teaching techniques capable of overcoming the persistent challenge of reconciling theoretical concepts with practical application (Rodrigues et al., 2024).

While undergraduate computing programs emphasize hard skills, they often fail to explicitly incorporate behavioral training, which ends up occurring indirectly and irregularly (Seagull et al., 2023). Wazlawick (2019) further explains that although SE guidelines evolve constantly, they fail to adequately address these interpersonal competencies.

University extension projects serve as a strategic mechanism to bridge this gap, providing an environment where students can apply technical knowledge to real-world demands. Seagull et al. (2023) discuss how these projects combine formal teaching, team practice, and interaction with real problems for the improvement of soft skills, in line with the PBL methodology. A study employing this methodology was conducted across two courses within the Software Engineering (SE) program at the Federal University of Amazonas, where students had to develop web software, within a simulated professional environment (characterized by tight deadlines, client engagement, and conflict resolution, etc.), where students experienced a significant impact on their career trajectories, as reported by Oran et al. (2023). Furthermore, a thematic analysis of the Meninas Digitais do Vale project corroborated these findings. Positive outcomes were also identified of student experiences highlighted significant growth in interpersonal skills, showed an evolution of skills such as creativity, communication, diversity, leadership, and teamwork (Rodrigues et al., 2022).

In this context, the present study investigates how the implementation of Project-Based Learning (PBL) within the WebAcademy training course, hosted at the Federal University of Acre (UFAC), contributed to the development of technical and interpersonal competencies among students in Cohort V. For this, a mixed-methods approach was employed, consisting of quantitative data collection at the beginning and end of the course, measuring self-perception and familiarity in eighteen types of soft skills and nine SE topics (hard skills), which were analyzed through statistical tests; and qualitative data, through interviews and open-ended questions about the perception regarding the influence of interactions between students with and without professional experience, in general, on the development of these skills, among other questions.

The methodology of this research follows a structured evolution of successive studies. It is grounded in initial investigations regarding technical practices presented at the XXII Brazilian Symposium on Software Quality (SBQS) (Cavalcante et al., 2023) and in findings on soft skill development presented at the XXXVIII Brazilian Symposium on Software Engineering (SBES) (Cavalcante et al., 2024). The current work significantly extends the discoveries presented at the XXXIX Brazilian Symposium on Software Engineering (SBES) (Cavalcante et al., 2025), which first integrated the dual analysis of technical and interpersonal competencies in Cohort V. This study advances the research by incorporating a complementary and longitudinal perspective from the faculty. When combined with student self-perception, this approach enables a more robust triangulation of the training process. Furthermore, it introduces a qualitative analysis of students' technical evolution, detailing how instructional guidance and immersive practice facilitate the mastery of complex Software Engineering topics.

Building on initial investigations regarding technical practices (Cavalcante et al., 2023) and behavioral development (Cavalcante et al., 2024), statistical findings from the preliminary study (Cavalcante et al., 2025) revealed significant growth in all nine technical topics (hard skills) and in 50% of the investigated interpersonal competencies (soft skills). Complementing these results, qualitative data from student interviews highlighted that practical team-based activities and interactions with IT professionals were the primary drivers of this development.

This extension advances the research by introducing a complementary perspective from the faculty, whose insights provide a broader understanding of the training process across multiple project editions. By combining these instructional observations with the previous student data, this study achieves a more robust triangulation. Furthermore, it presents a new qualitative analysis focused on technical evolution, detailing how instructors' guidance and the immersive Hands-On environment facilitate the mastery of complex Software Engineering concepts.

The article is organized as follows: i) Section 2 details the theoretical foundation; ii) Section 3 discusses related work; iii) Section 4 presents the WebAcademy project; iv) Section 5 explains the research methodology; v) Section 6 presents the implementation of the PBL-based training environment; vi) in Section 7, there is the explanation of the results and discussions; vii) in Section 8, the threats to validity are described; and, finally, viii) in Section 9, the conclusions and proposals for future work are presented.

2 Theoretical Foundation

The scientific basis of this study is grounded in the premise that learning SE transcends the mere acquisition of programming syntax; it is an inherently social, active, and complex problem-oriented process that requires an environment that fosters an individual's socio-technical maturation. This section establishes the theoretical link between active methodologies and the development of interpersonal competencies, defining PBL not only as an instructional strategy but as an immersive ecosystem that mimics the pressures and collaborations inherent in the software industry. Interpersonal competencies, often referred to as soft skills, encompass abilities related to communication, collaboration, and problem-solving in social and professional contexts, and are increasingly recognized as essential in Software Engineering education.

The foundations of collaborative learning trace back to the social development theory, which posits that learning occurs more deeply through mediated interaction between peers and mentors (Vygotsky, 1978). In the context of this study, this socio-constructivist view justifies why team collaboration is the primary driver of the evolution of communication and conflict resolution skills, allowing students to internalize behavioral competencies within a real-world practical scenario. This socio-constructivist perspective serves as the bedrock for methodologies that prioritize teamwork and the exchange of experiences.

The transition from this social basis to project application is supported by prior research, which highlights how PBL sustains student engagement by connecting learners to authentic and challenging problems, allowing them to perceive the immediate utility of the technical knowledge they are acquiring (Blumenfeld et al., 1991). The necessity of adopting such methodologies is reinforced by global curricular guidelines, such as those proposed by ACM/IEEE-CS/AAAI; these guidelines emphasize that computing education must transcend isolated technical instruction, promoting curricula that prepare students for the dynamic and collaborative challenges of the modern software industry (Kumar et al., 2024).

Formally, PBL is defined as a teaching strategy organized around complex questions and investigative tasks, culminating in the creation of a realistic final product (Thomas, 2000). In Software Engineering, this approach allows for the integration of technical rigor and behavioral development. It has been argued that PBL is an effective approach to foster soft skills through action-oriented didactics, enabling students to experience a software lifecycle holistically, dealing with uncertainties and changing requirements, while practicing communication, teamwork, and self-reflection regarding their roles within the team (Sedelmaier and Landes, 2018).

The relevance of this model is validated by the gap identified between topics traditionally taught in academia and the skills that software professionals consider crucial in daily market life (Lethbridge, 2000). The effectiveness of this transition from passive to active learning is synthesized by prior evidence demonstrating that active learning methods result in superior long-term knowledge retention and enhanced interpersonal skills when compared to traditional lecture-based instruction (Prince, 2004).

In the specific context of SE, the evidence from Prince (2004) gains empirical support from the longitudinal study by Garcés and Oliveira (2024). By analyzing four years of PBL methodology application, the authors confirm that immersion in real projects allows students to consolidate technical knowledge while developing interpersonal competencies required by the industry. Although the study highlights critical challenges, such as time management and the complexity of team dynamics, it reinforces the role of PBL as a vital mechanism to mitigate the gap between academia and the market. This practical perspective, which aligns pedagogical foundations with real industry demands, is shared by other contemporary studies that compose the investigative landscape detailed in the following section.

Although prior studies have examined PBL and competency development, there remains a lack of longitudinal and integrated analyses that simultaneously address the interplay between technical and interpersonal competencies through multiple perspectives. In this context, this study adopts PBL as a socio-technical learning environment to investigate how the integration of these competencies emerges over time within an immersive training program.

3 Related Work

Current research into PBL and higher education aims to determine the efficacy of this methodology in cultivating essential soft skills among students.

An investigation conducted in the partner countries of the HERA project (Denmark, Greece, Spain, Estonia, and Portugal) by Caeiro-Rodríguez et al. (2021) regarding the teaching of soft skills in engineering courses, through a form applied to 184 final-year students, revealed a significant student consensus: soft skills are perceived as more relevant than technical ones. They highlighted that the most appropriate teaching methodologies for improving these skills are, in this order: Project/Problem-Based Learning, Thinking-Based Learning, Design Thinking, and Competency-Based Learning.

Oran et al. (2023) explored how PBL can boost the development of technical and non-technical skills in students. Although the title mentions Problem-Based Learning, Project-Based Learning is clearly described throughout the text. The researchers conducted a survey-based study involving 27 students across two Software Engineering (SE) courses. Among the results, students highlighted significant gains in team management, Scrum ceremonies, communication, and knowledge management. Additionally, some students reported that the experience helped with career direction, obtaining their first job, or increasing their professional contribution at work.

Rodrigues et al. (2024) examined the pedagogical impact of Project-Based Learning (PBL) on project management instruction within the Production Engineering program at the Federal University of Espírito Santo. The research was conducted via a case study involving 23 students. The authors concluded that the methodology adopted serves as a catalyst for student motivation, contributing to the reduction of dropout rates and promoting a deeper understanding of the concepts addressed, in addition to stimulating the development of practical skills relevant to project management.

Moura et al. (2025) reported the experience of the TWIST program, designed to foster both technical and socioemotional competencies among undergraduate computing students. Although the initiative was not formally structured as a traditional Project-Based Learning course, its pedagogical design incorporated collaborative and project-oriented dynamics aligned with PBL principles. The program combined mentoring sessions, discussion circles, and team-based software activities aimed at simulating professional environments and promoting interpersonal skill development. Among the main findings, the authors reported improvements in communication, teamwork, self-confidence, and conflict management. Participants also indicated increased readiness for professional practice and stronger engagement with computing careers. The results reinforce the potential of active and experiential learning environments to support soft skills development in higher education, particularly when integrated with collaborative practices and industry-oriented experiences.

Podlaski et al. (2025) investigated the use of intensive software development projects as a pedagogical strategy to promote soft skills in higher education, structuring the experience based on active learning and Problem-Based Learning principles. The intervention took place in a short-term international course in which students from different disciplines and countries worked in multidisciplinary teams to solve real-world problems through the development of digital prototypes. The approach required autonomous problem definition, collective decision-making, and collaborative task management, simulating authentic professional contexts. Evaluation was conducted through pre- and post-tests using the KYSS instrument, which measures socioemotional and professional competencies. The results indicated statistically significant gains in communication, cooperation, digital skills, and self-reflection. Conversely, competencies such as creativity, critical thinking, and flexibility did not show significant improvement in the short term.

Similar to the works of Oran et al. (2023) and Rodrigues et al. (2024), this work investigated the influence of using PBL on student development. Some differences between this work and that of Oran et al. (2023) include the diversity of the students, who, in the present work, are characterized only by having some experience in computing-related areas, the application of questionnaires at the initial and final moments, and the statistical analysis. The work of Rodrigues et al. (2024) sought evidence of the methodology as a way to combat dropout, through a case study, in the context of a single course; in the present work, we sought to relate the methodology with the development of skills in students using a survey and interviews in the context of an extension project focused on agile practices. Additionally, initiatives such as the TWIST program reported by Moura et al. (2025) highlight the role of collaborative and project-oriented training environments in fostering soft skills development, including communication, teamwork, and self-confidence, in higher education computing contexts.

A synthesis of these related works and their relationship with the present study's convergence points and methodology is illustrated in Figure 1.

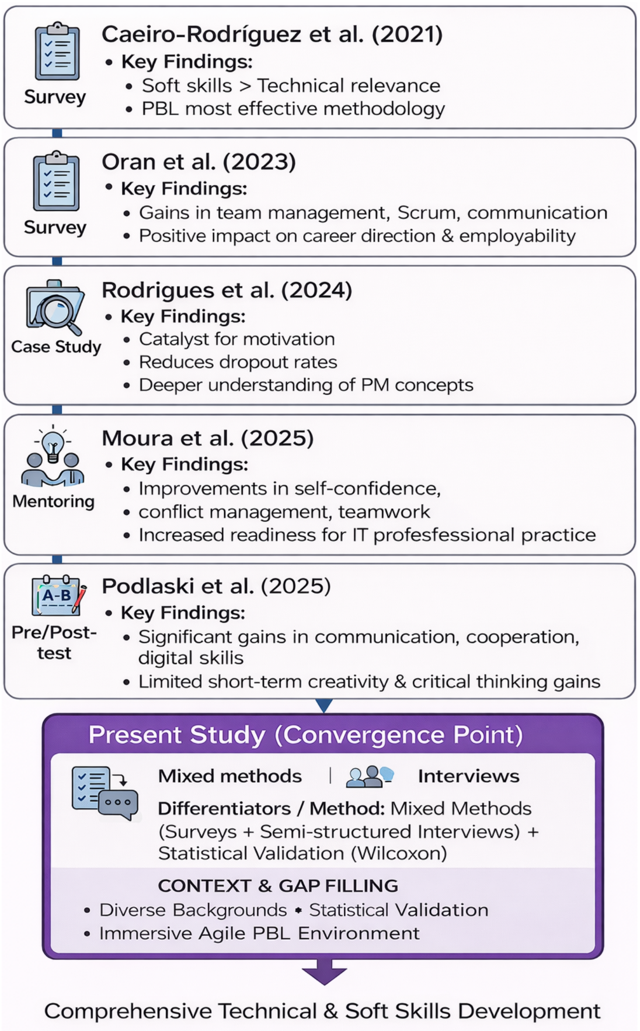

Figure 1. Related Work.

4 The WebAcademy Project

The WebAcademy project[1] is a training course established through a strategic partnership between UFAC, Motorola, Flextronics, and Fundape. Its primary mission is to qualify professionals for Full-Stack Web development. The student selection process is formalized through public calls for applications, ensuring a selection based on predefined technical and academic requirements.

The program spans approximately seven months and is divided into two distinct phases: the first phase (195 hours) integrates foundational theory with advanced programming modules, utilizing coursework designed to reinforce knowledge acquisition through a progressive structure of basic, intermediate, and advanced modules; phase two, the Hands-On Module (105 hours), focuses on agile frameworks and collaboration. Students are divided into teams to execute projects within a simulated professional ecosystem, involving backlog management, interaction with clients, and periodic deliveries, bridging the gap between academia and the labor market.

During the Hands-On phase, the Project-Based Learning (PBL) framework is fully operationalized. Students are required to engage in direct stakeholder management with the Product Owner (PO), practice team self-manage, and rotate the Scrum Master role at each sprint. This immersive experience necessitates effective conflict resolution and the management of project impediments during periodic progress reviews.

PBL manifests directly in Hands-On, as it is a stage that tends to simulate a real software development environment, in accordance with the methodology's proposal (Thomas, 2000). Guided by mentors, students are encouraged to integrate industry-standard tools and agile practices (Kanban, Daily Scrum, Pair Programming) with technical frameworks like Angular and Spring. Consequently, the program's pedagogical approach is defined by the synergy between agile practices and PBL principles.

4.1 Implementation and Reproducibility Dynamics

To ensure the reproducibility of this model, the pedagogical intervention is grounded in an environment that strictly simulates professional industry standards (Saar et al., 2023a,b). Guided by the Scrum framework, project development occurs in one- to two-week sprints. Each sprint concludes with a formal review to present progress, followed by planning sessions for the upcoming cycle. A central pillar of this phase is team autonomy; students are responsible for technical decision-making and internal role definitions. To foster diverse skill development, roles such as the Scrum Master are rotated at each sprint, ensuring all members experience leadership responsibilities and different perspectives of the development lifecycle.

The mentorship model acts as a cornerstone for this technical maturation, with instructors serving as facilitators who provide informal feedback as technical or interpersonal chal-

[1]http://webacademy.ufac.br/

lenges arise (Saar et al., 2023a,b). Daily interactions occur through agile ceremonies, where mentors mediate conflict resolution and manage project impediments. The faculty advisor often assumes the Product Owner role, especially when projects involve institutional demands, being responsible for validating requirements and simulating real-world market pressures.

Technological support and physical infrastructure are essential for simulation fidelity (Saar et al., 2023b). Teams operate in a modern software development laboratory at UFAC, featuring an environment specifically designed to favor collaboration and collective problem-solving through pair programming and group discussions. The infrastructure adheres to industry standards by integrating Git and GitHub for version control and peer reviews, while Trello or GitHub Projects facilitate backlog management through Kanban boards. The development stack centers on Angular and Spring Boot, with Visual Studio Code as the primary IDE. This synergy between a structured curriculum, immersive practice, and specialized tooling enables a seamless transition from theory to solving complex, real-world engineering challenges.

5 Method

A survey (Gil, 2002) and interviews were conducted with students of the WebAcademy course. Two questionnaires were applied to cohort V, one at the beginning and one at the end of the approximately 7-month course, seeking to obtain self-assessment of familiarity with Software Engineering content and the development of non-technical skills during this period of participation in the project.

The procedure followed, illustrated in Figure 2, was based on Pfleeger and Kitchenham (2001) and is described below:

1. Literature review: A literature review was conducted on the development of soft and hard skills in Software Engineering education and their relevance to the labor market;
2. Objective adjustment (General and Specific): Based on the gaps identified in the literature review, the research objectives were continuously refined to ensure theoretical alignment and methodological feasibility. This iterative process continued until the final scope was clearly defined;
3. Research schedule: The stages of the study were planned, with established timelines to ensure all activities remained within the project's timeframe;
4. Questionnaire design: Data collection instruments were designed with structured questions to assess both the hard and soft skills of the participants. The questionnaire was developed based on a review of previous literature in the field;
5. Cohort IV application: The questionnaires were applied to the fourth WebAcademy cohort, enabling data collection for a preliminary analysis;
6. Refinement of data collection instruments: Based on the analyses performed on Cohort IV, the questionnaires were refined, and new semi-structured interview protocols were developed for both students and professors to enable data triangulation;
7. Data collection: Data collection was executed using the refined instruments. Questionnaires and interviews were administered to Cohort V students, complemented by faculty interviews to triangulate multiple perspectives on competency development;
8. Data analysis: The collected data were analyzed through a mixed-methods approach, employing statistical and qualitative techniques to identify patterns and address the research questions;
9. Writing: The research findings are systematically presented and described in this paper with clarity and objectivity;
10. Proofreading: The text was reviewed to ensure coherence, cohesion, and academic quality. This involved an iterative writing-revision cycle, allowing for continuous refinement until the final version was achieved.

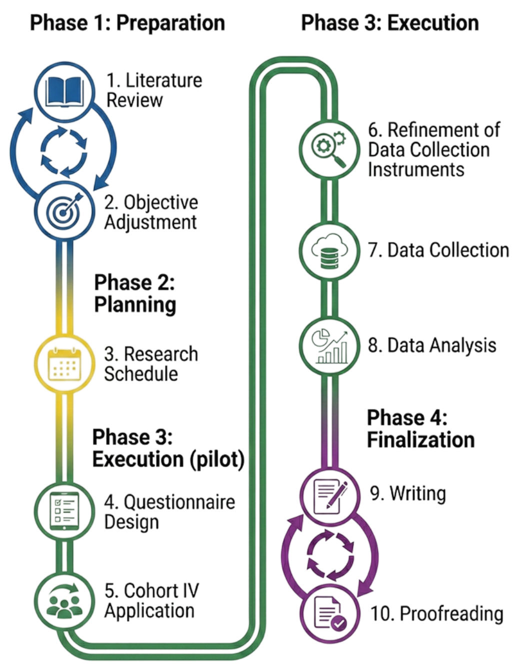

Figure 2. Adopted procedure.

To align the study with initial statistical goals, the course methodology, and qualitative insights from interviews and open-ended questions, the following research questions were formulated:

1. How does the PBL methodology applied in the course influence the development of students' soft skills?
2. What is the impact of interactions during the course with mentors and colleagues active in IT on the improvement of hard skills and soft skills?
3. In what way does the systematic study of Software Engineering topics (hard skills) contribute to the evolution of students' interpersonal skills?
4. Did the participants' self-assessments of hard skills and soft skills show statistically significant differences between the beginning and the end of the course?

5. What motivations can be identified in the qualitative responses of students who had self-assessments with significant variations?
6. How does the Dunning–Kruger effect manifest in these variations, and what qualitative motivations explain these changes?
7. Which experiences are perceived by students as crucial for increasing confidence in their soft skills?

Google Forms[2] was used for the creation and distribution of the questionnaires. Selection criteria included the researchers' familiarity with the platform, ease of access for respondents (via online link), and the availability of the tool at no cost.

For the construction of this research, 4 questionnaires were applied, named according to the cohort and the moment of application: T4.1 and T4.2 for cohort IV, and T5.1 and T5.2 for cohort V. The first questionnaire for each cohort (T4.1 and T5.1) was applied during the first week of classes; T4.1 contained only closed questions, while T5.1 contained both closed and open questions. The second questionnaire (T4.2 and T5.2) was applied in the last week of the course, with both including closed and open questions. Questionnaire T4.1 was applied between August 7 and 14, 2023, and T4.2 between February 28 and March 6, 2024. Cohort IV served as a pilot study, allowing for the assessment of the quality of the collection instrument. Based on the results of this application, adjustments were made to improve the formulation of questions and ensure greater precision in the assessment of soft skills and hard skills. The revised questionnaires were then applied to cohort V, with T5.1 administered between April 22 and 29, 2024, and T5.2 between November 8 and 13, 2024. The findings from Cohort IV are detailed in the papers by Cavalcante et al. (2023, 2024). The questionnaire and the detailed mapping of modifications are provided in the papers registry of this study.

The questions related to Software Engineering knowledge were developed based on the work of Portela et al. (2015). The authors considered surveys applied in the works of Lemos et al. (2019) and Lethbridge (2000), as well as reference curricula from ACM and SBC for writing the questions (Kumar et al., 2024; SBC, 2005). This study presents the results of a survey on the adoption and learning of Software Engineering topics recommended by the ACM/IEEE and SBC reference curricula. Its choice is due to its relevance to the research focus and the fact that it is the most recent study found with this approach.

Questions about soft skills were based on the study by Rodrigues et al. (2022), which investigated the development of soft skills in computing students participating in a specific project, using the set of skills adopted by Marques et al. (2020) in their qualitative analysis. This analysis was based on the proposals of Vogler et al. (2018) and González-Morales et al. (2011). This work was chosen for presenting a context similar to this study, being recent, well-founded in previous research, and also due to its publication in the proceedings of Women in Information Technology (WIT) by SBC, which reinforces its relevance in the field.

Interview protocols were formulated through collaborative sessions among researchers and validated via a pilot study with a student. Selected students were invited via email, with up to two follow-up reminders for scheduling, offering both in-person and online formats. Interviews were transcribed using Google Colaboratory[3] and analyzed in a process similar to that used in qualitative analysis, in which researchers discussed their interpretations until reaching a final consensus. During iterations, opinions converged.

To deepen the investigation into the effectiveness of the teaching methodology in the simultaneous development of technical and interpersonal skills, semi-structured interviews were conducted with faculty members. This group consisted of the four primary instructors of the program, representing 100% of the teaching staff responsible for the technical foundation phase. These faculty members oversaw 205 instructional hours across twelve specialized modules, ranging from UX and Design Thinking to advanced Front-End and Back-End frameworks, Testing, and CI/CD. This comprehensive representation ensures that the qualitative insights reflect the perspectives of those who monitored the participants' technical evolution from initial concepts to complex project execution in the Hands-On phase.

The primary objective was to obtain a complementary, longitudinal assessment of students' socio-technical development across successive WebAcademy cohorts. The interview protocol was structured to examine how active methodologies influence professional maturity and the students' capacity to apply theoretical foundations to complex Software Engineering challenges. Formal invitations were sent via email, and the interviews were conducted individually through remote communication platforms. All sessions were recorded and fully transcribed using the Google Colaboratory environment to ensure data integrity and facilitate a rigorous qualitative analysis. Finally, the researchers conducted iterative cycles of discussion to ensure triangulation between the instructors' pedagogical perspective and the students' self-perception.

For the qualitative analysis, the Card Sorting technique was used (Spencer and Garrett, 2009; Zimmermann, 2016). A shared document was used, where authors could work collaboratively, containing all collected discursive responses. Researchers met to evaluate the responses, ensuring data were systematically verified and coded to identify relevant categories. Figure 3 exemplifies the coding process.

STUDENT	COMMENT	CODE
2	It **helped** because I learned from the other students, and it **added to my professional knowledge**.	Helpfulness Knowledge contribution
6	It **made no difference**, as there **was no communication** between us (myself and those students).	No impact Lack of communication

Figure 3. Coding example.

To ensure a conclusive classification and a focused analysis of student perspectives, this iterative process continued until the researchers reached a consensus. This approach ensured that the findings aligned closely with both the research questions and the students' reported experiences.

To evaluate the evolution of soft skills and SE knowledge

[2] `https://docs.google.com/forms`

[3] `https://colab.google/`

throughout the course, we employed the statistical tests illustrated in Figure 4. Participants' qualitative responses were coded into numerical values. Subsequently, we calculated the delta between final and initial scores for each item to derive a change variable. The normality of these differences was verified using the Shapiro–Wilk test (Shapiro and Wilk, 1965), which indicated that the data did not follow a normal distribution ($p < 0.05$). Consequently, the Wilcoxon test (Wilcoxon, 1945) was chosen as a non-parametric alternative to the paired t-test to compare responses before and after the course. The Wilcoxon test is recommended for related samples when data do not meet normality assumptions, and it is widely used in education and computing research (Field, 2018; Siegel and Castellan, 1988). This approach ensured the robustness of the analysis and adequacy for the collected data.

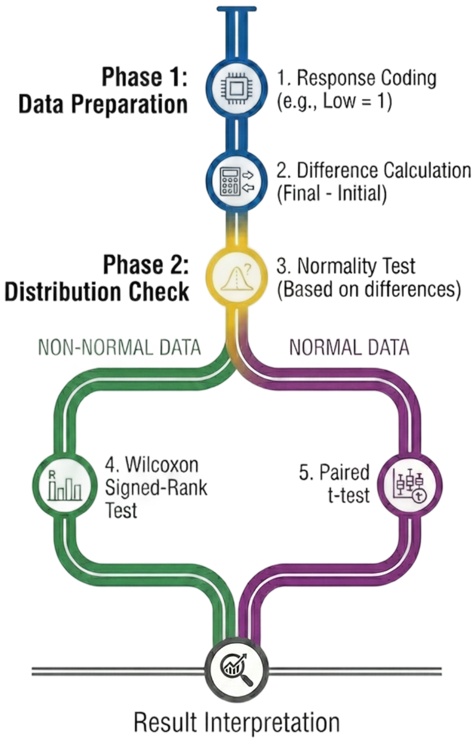

Figure 4. Methodology for statistical tests.

Statistical calculations were performed using the Python programming language with the help of the scipy and pandas libraries. The scipy library provided functions for the Shapiro–Wilk and Wilcoxon tests, while pandas was used for data manipulation and organization. The choice of Python is justified by its flexibility, wide adoption in the scientific community, and the availability of robust tools for statistical analysis.

6 Implementation of the PBL-Based Training Environment

Although the research methodology describes the instruments and analysis techniques adopted, it is essential to detail the practical execution of the pedagogical intervention. To immerse students in a scenario that simulates professional industry environments, the pedagogical intervention was grounded in the pillars of PBL. Consequently, this section describes how PBL principles were operationalized within the Hands-On Module.

6.1 Team Composition and Scope

Team formation was intentional rather than random. Instructors implemented strategic grouping based on student performance in previous modules (Basic, Intermediate, and Advanced). Students were distributed across different levels of technical and behavioral proficiency, aiming for balanced teams that could handle the specific complexity of each project, meaning the project scope was also a determining factor for team composition.

6.2 Sprint Dynamics and Autonomy

Project development was guided by the Scrum framework, adapted for the educational environment. Sprints lasted between 7 and 15 days, each concluding with a formal Sprint Review to present progress. Subsequently, teams met to plan activities for the upcoming cycle.

A central pillar of the intervention was granting teams high levels of autonomy. Students were responsible for technical decision-making and internal role definitions, while the course environment encouraged them to navigate their own communication and conflict challenges. To foster diverse skill development, roles such as the Scrum Master were rotated each Sprint, ensuring all members experienced leadership responsibilities. In this student-centered design, mentors acted only as mediators in the last resort.

6.3 Role of the Advisor and the Product Owner

Mentors acted not only as instructors but as facilitators, providing informal feedback on interpersonal skills as technical challenges arose, while ensuring the teams remained focused on delivering client value. The Product Owner (PO) role was assigned based on the specific needs of each project. Typically, a designated individual is appointed to act as the PO, with the advising professor assuming this role if such a figure is unavailable. The faculty advisor assumed this responsibility when a project stemmed from their own research or involved institutional demands from UFAC. In such instances, the advisor was responsible for validating requirements and simulating real-world market pressures.

6.4 Infrastructure and Artifacts

Technological support was fundamental for simulation fidelity. The infrastructure mirrored industrial standards, incorporating Git and GitHub for version control and code review, and Trello or GitHub Projects for backlog management and task tracking via Kanban boards. Regarding the development stack, projects primarily utilized Angular and Spring Boot frameworks.

7 Results and Discussion

The survey conducted with questionnaire T5.1 achieved a 93.9% response rate (31 out of 33 students). Questionnaire T5.2 reached 100% of the students who completed the course (29 respondents). This difference in total participants is attributed to four dropouts throughout the semester: three during the basic module and one during the intermediate stage.

7.1 Student Profile

Student profiles were characterized based on data from questionnaire T5.2, with the most relevant demographics presented in Figure 5. Among the 29 respondents, 89.7% self-identified as cisgender men and 10.3% as cisgender women. Regarding financial support, 51.7% of students held scholarships, while 48.3% did not. Beyond financial data, the observed gender imbalance, with women representing only 10.3% of the cohort, reflects the systemic barriers faced by women in IT career paths, particularly in software development (Giovannetti, 2023).

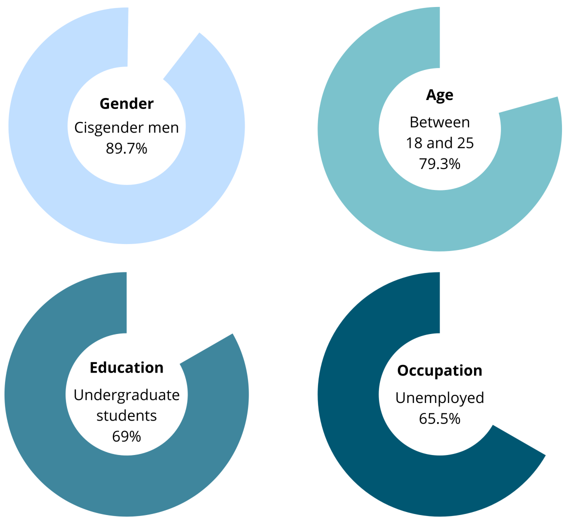

Figure 5. Profile of cohort V students.

Regarding age, students were distributed across the following age groups: 51.7% were between 18 and 21 years old, 27.6% were aged between 22 and 25, 6.9% were aged between 26 and 29, 3.4% were aged between 30 and 33, 3.4% were aged between 34 and 37, 3.4% were aged between 38 and 41, and 3.4% were aged 42 years or older.

In terms of employment status, 62.1% reported that they were full-time students, 17.2% identified as active professionals in the technology sector, 6.9% as active professionals in non-technology sectors, 3.4% as interns in the technology sector, 3.4% as self-employed professionals in technology, 3.4% as self-employed in non-technology fields, and 3.4% reported holding a prior degree but were unemployed at the time of the survey.

Students were asked about their current educational level: 69% were enrolled in an undergraduate degree program, 13.8% had completed an undergraduate degree, 10.3% had technical or specialized training in computing-related fields, 3.4% were pursuing a graduate degree, and 3.4% had already completed a graduate degree at the time of the survey.

This result indicated that the cohort represents a young and predominantly male group, exhibiting a highly heterogeneous profile, with students at different professional and academic stages. This diversity highlights the need for flexible approaches in technology courses, adapted both to beginners and to professionals seeking upskilling.

7.2 Interaction between students

Students were asked to discuss their experiences interacting with professionals in the field during the course. The categories (Cat.) and codes identified in the qualitative analysis are presented in Table 1, along with the respective percentages corresponding to mentions of each category and code.

Table 1. Considerations regarding interaction with professionals.

Category (a)	Code (b)	% (b)	% (a)
Positive evaluation of the interaction	Productive	37.9	72.4
	Interesting	10.3	
	Good	10.3	
	Useful	6.9	
	Important	6.9	
	Added value	3.4	
	Relevant	3.4	
Types of interaction	Experience sharing	48.3	51.7
	Interaction limited to instructors	3.4	
Technical or conceptual knowledge	Technical learning	24.1	41.4
	Knowledge gain	20.7	
	Clarified doubts	20.7	
Professional evolution	Understanding the labor market	27.6	34.5
	Provided guidance	10.3	
	Stimulated curiosity	3.4	
Networking	Professional network	17.2	17.2
Neutral or without impact	Made no difference	3.4	6.9
	No influence on hard skills	3.4	

The analysis of the 29 responses to questionnaire T5.2 indicated that: 72.4% of the students evaluated the interaction positively, with 37.9% stating that it was productive; only 6.9% reported neutral or negative evaluations; 48.3% mentioned experience sharing as the primary form of interaction; interaction limited to instructors was rarely mentioned (3.4% of responses); 41.4% highlighted gains in technical or conceptual knowledge; 34.5% mentioned professional development; and 17.2% valued the development of professional networks.

Qualitative analysis further supports these findings, as evidenced by a remark from student E29, who emphasized that knowledge exchange with peers and mentors was vital for mastering advanced topics and refining essential communication skills. This perspective aligns with the quantitative data, illustrating how social interaction fostered both technical proficiency and soft skills. Such evidence corroborates Vygotsky's theory (Vygotsky, 1978), which posits that engagement with more experienced peers facilitates the internalization of new knowledge. Furthermore, the student noted that practical insights from mentors were instrumental in resolving specific queries and broadening their grasp of industry dynamics and best practices in Web development.

Based on the collected data, it is possible to observe that the training course understands the benefits that interactions among students bring to the learning process, recognizing this approach as crucial to the development of participants' skills, thus stimulating cooperative engagement, through which students can share their experiences, recommendations, and perspectives regarding their professional future.

An illustrative comment was provided by student E8: *"I had more interaction in the part of listening to conversations between colleagues [...] it made me know a little more about what expectations are like and how people in the technology field are, but I think it didn't have any significant influence on my learning in the courses of the training program."* This report reveals that the student had a predominantly passive experience in the interaction with peers, as they specifically mention listening to conversations, which suggests a more receptive rather than active form of participation in these moments. This observational stance may help explain why, although they acquired knowledge about market expectations, they did not perceive significant gains in the technical learning of the courses.

This distinction is important because it reinforces the idea that active engagement in discussions (such as asking questions, debating ideas, or explaining concepts) tends to produce deeper cognitive benefits than mere passive listening. As demonstrated by Prince (2004), when students actively articulate and defend their reasoning, they process information at a deeper level than when they only listen passively. The analyzed case clearly demonstrates how different levels of participation in interactions can lead to significantly different learning experiences and results. The findings support the pedagogical value of collaborative learning in the context of the course, while also highlighting the need for attention to the various degrees of engagement presented by the students.

Although most experiences were evaluated positively, it is relevant to note that 6.9% of students considered the interaction neutral or without impact on their technical learning. The presence of this minority highlights a pedagogical challenge: the need for mechanisms that promote active engagement in collaborative work for all student profiles.

7.3 Statistical tests

Through questionnaire T5.1, students provided a baseline self-assessment of their soft skills at the beginning of the course, focusing on: Application of Information Technology, Self-direction, Results evaluation, Information search and classification, Communication of results, Written communication, Oral communication, Creativity, Diversity, Ethics and Social Responsibility, Managing customer expectations, Conflict management, Leadership, Critical thinking, Problem-solving, Professionalism, Decision-making, and Teamwork. The response options were: none, little, reasonable, much, I don't know, and not applicable. At the end of the term, the same self-appraisal was conducted via questionnaire T5.2, serving as a post-course evaluation of the students' soft skills.

The statistical methodology employed to assess the development of soft skills is detailed in the workflow presented in Figure 4. Participants' responses were coded according to the following scale: None = 0, Little = 1, Reasonable = 2, Much = 3, and None/I don't know/Not applicable = 0. Subsequently, the difference between final and initial scores for each soft skill was calculated to derive a change variable. The normality of these differences was assessed using the Shapiro–Wilk test, which revealed a non-normal distribution ($p < 0.05$). Consequently, the Wilcoxon test was employed to compare the pre- and post-course responses, with the results summarized in Table 2.

Table 2. Wilcoxon test - soft skills

Soft Skills	Stat	p
Application of Information Technology	40	0.005947
Self-direction	50	0.867959
Results evaluation	50	0.193985
Information search and classification	43	0.094037
Communication of results	14	0.002867
Written communication	46.5	0.414536
Oral communication	22.5	0.018422
Creativity	26	0.133614
Diversity	48	0.024948
Ethics and Social Responsibility	37.5	0.563449
Managing customer expectations	5	0.000058
Conflict management	21	0.003256
Leadership	44.5	0.028564
Critical thinking	0	0.002282
Problem-solving	18	0.131668
Professionalism	15	0.165518
Decision-making	15	0.004678
Teamwork	16	0.057827

The Wilcoxon test results, presented in Table 2, reveal a statistically significant improvement ($p < 0.05$) in 50% of the evaluated soft skills (nine out of eighteen). The skills that showed statistically significant differences were: Application of Information Technology, Communication of results, Oral communication, Diversity, Managing customer expectations, Conflict management, Leadership, Critical thinking, and Decision-making. These results suggest that the course was particularly effective in developing these competencies, possibly due to its practical focus and the specific activities that promoted their improvement.

In contrast, skills such as Self-direction, Results evaluation, Information search and classification, Written communication, Creativity, Ethics and Social Responsibility, Problem-solving, Professionalism, and Teamwork did not show statistically significant differences. This lack of significant improvement may suggest that participants already possessed high baseline levels of these competencies, or that the intervention was not sufficiently intense to foster further growth. Consequently, future iterations of the course should incorporate targeted interventions specifically designed to strengthen these underdeveloped areas.

At the beginning of the course, questionnaire T5.1 was used to collect student self-assessments across several hard skills, including Requirements Engineering, Continuous Delivery, Configuration Management, Project Management, Agile Methods, Pair Programming, Software Design, Verification and Validation, and the tools and environments used in the course (e.g., GitHub, Visual Studio Code). The response options were: none, little, reasonable, much, and I don't know how to answer. This self-assessment was repeated at the end of the course via questionnaire T5.2 to capture changes in perceived expertise.

These responses enabled a statistical analysis. First, the data were organized in a spreadsheet and the response categories were converted into numerical values, where the "none" and "I don't know" levels were assigned a value of 0, "little" received a value of 1, "reasonable" received a value of 2, and "much" received a value of 3. Subsequently, the delta scores between final and initial values were calculated for each hard skill. Data normality was then assessed via the Shapiro–Wilk test, which revealed a non-normal distribution ($p < 0.05$).

Due to the non-normal distribution, the Wilcoxon test was employed as a non-parametric alternative to the paired t-test for comparing pre- and post-course responses. This test is recommended for related samples when the data do not meet normality assumptions, ensuring the robustness of the analysis and the validity of the results. The test results, summarized in Table 3, indicate a statistically significant improvement ($p < 0.05$) across all evaluated hard skills.

Table 3. Wilcoxon test - hard skills

Hard Skills	Stat	p
Requirements Engineering	26	0.002356
Continuous Delivery	30	0.001978
Configuration Management	34.5	0.003720
Project Management	18	0.000871
Agile Methods	20	0.005338
Pair Programming	17	0.000188
Software Design	20	0.001953
Verification and Validation	28	0.000339
Tools and Environments	26	0.007066

7.4 Degree of contribution per module

Participants evaluated the perceived contribution of each module to their soft skills development, with results illustrated in Figure 6. The assessment utilized a five-point scale: none, low, moderate, high, and undecided.

In the Basic Module, 34.5% of students perceived its contribution to soft skills development as reasonable, while 31% rated it as high. Only 3.4% considered the contribution to be not significant at all. These results suggest that while the Basic Module played a relevant role in initiating skill development, it may not have been the primary driver of soft skills compared to subsequent stages.

The Intermediate Module received a more balanced perception, with 48.3% of participants considering its contribution to be reasonable, 27.6% rating it as high, and the remaining 24.1% rating it as low. This indicates that the module was relevant, but it may not have been as effective as the other modules.

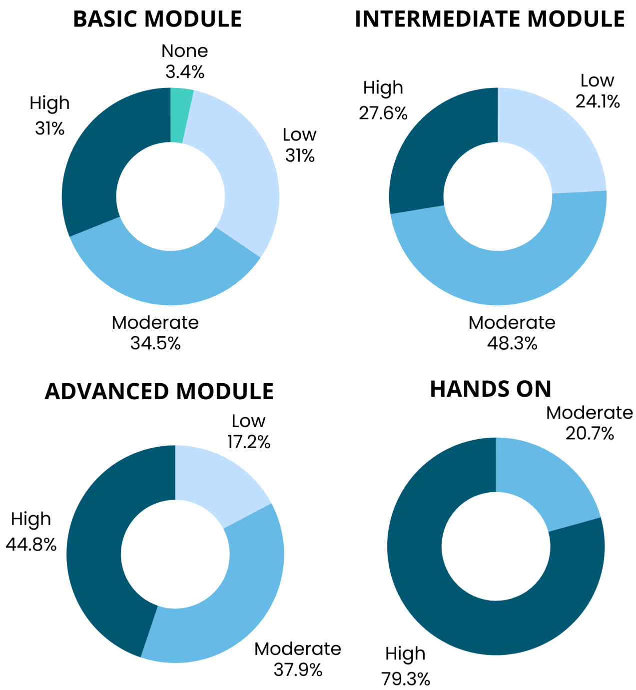

Figure 6. Degree of contribution per module - soft skills.

The Advanced Module obtained a generally positive evaluation, with 44.8% of participants considering its contribution to be high, 37.9% considering it as moderate, and 17.2% rating it as not very significant. These results indicate a moderate contribution; while the module remained relevant, its perceived impact on soft skills was less pronounced than in the other stages of the course.

The Hands-On Module showed the most significant positive impact, with 79.3% of participants rating its contribution to soft skills development as high. This indicates that the module was highly effective and likely focused on practical activities that promoted skill development.

Accordingly, Table 4 summarizes these results, presenting the course modules hierarchically based on their contribution to the development of students' soft skills: Hands-On, Advanced, Intermediate, and Basic.

Table 4. Hierarchy of modules by degree of contribution - soft skills.

Module	Comment
Hands-On	79.3% with a high degree of contribution, indicating that it was highly effective.
Advanced	44.8% with a high degree of contribution, quite significant.
Intermediate	48.3% with a moderate degree of contribution, may need adjustments to improve its effectiveness.
Basic	34.5% with a moderate degree of contribution, may not have been as effective.

The results indicate that participants perceived differences in the contribution of each module to the development of their soft skills. Hands-On was the most highly rated, with 79.3% of participants considering its contribution to skill improvement to be high. This result suggests that the practical and challenging focus of this module was particularly effective. The Advanced Module also received a positive evaluation, with 44.8% of participants rating its contribution as high and 37.9% rating it as moderate. This indicates that the module was effective in promoting skill development. The Intermediate Module received an evaluation of 48.3% with a reasonable degree of contribution; this may indicate that some adjustments are needed to improve its impact within the course. In turn, the Basic Module was rated by 34.5% of participants with a contribution considered moderate, and another 31% rated its contribution as high. The contrast observed in this survey indicates that the measures adopted in some modules require further attention so that all of them can contribute more consistently to the improvement of students' soft skills.

In a similar manner to the soft skills assessment, students evaluated the extent to which each module contributed to their technical proficiency. The survey employed the same five-point scale (none, low, moderate, high, and undecided), with the resulting distribution illustrated in Figure 7.

Unlike the results obtained regarding students' perception of soft skills, the Advanced Module outperformed the Hands-On Module in its perceived degree of contribution to hard skills, with 75.9% of students reporting a high impact. This result is consistent with the program's design, as the Advanced Module introduces more complex technologies and frameworks, which are subsequently applied in the final projects (Saar et al., 2023b).

However, these data suggest a pedagogical subtlety regarding the transition to the Hands-On stage. While the Advanced Module expands the students' technical repertoire, the Hands-On Module prioritizes the synthesis and application of existing knowledge. This shift in focus can lead to a misleading self-perception where students may feel they are "learning nothing new" in terms of technical content, as they are no longer being introduced to new frameworks.

In reality, qualitative evidence suggests that this stage is where the most significant technical maturation occurs. As observed in studies on the WebAcademy Project, students often apply theoretical concepts during the Hands-On stage in an unconscious or implicit manner (Saar et al., 2023a). Although the novelty of tools decreases, students achieve a higher level of proficiency, becoming faster and more precise in integrating multiple software layers under pressure. Therefore, the perceived plateau in technical gain is actually a milestone of professional competence, where the expansion of knowledge is replaced by the mastery of execution and the ability to solve real-world engineering challenges autonomously (Saar et al., 2023b,a).

Regarding the Basic Module, 37.9% of participants considered the contribution as low and 31% as moderate. Only 24.1% rated it as high, while 6.9% indicated there was no contribution (none). These results suggest that the Basic Module had a moderate impact on technical gains. This may be attributed to its focus on foundational concepts that some students had already mastered, or simply to its nature as an introductory phase.

The Intermediate Module showed a higher perception of technical gain, with 51.7% of students rating the contribution as high and 34.5% as moderate. Only 13.8% indicated a low contribution (low). This demonstrates that the technical content at this stage had a substantially greater impact on

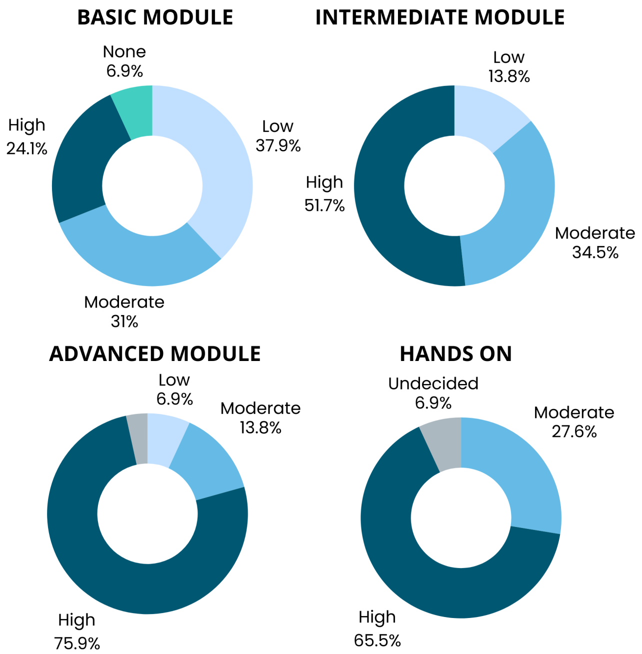

Figure 7. Degree of contribution per module - hard skills.

the participants' professional development compared to the Basic Module.

The Advanced Module obtained the most expressive evaluation in terms of high technical contribution, with 75.9% of responses in the high category. Another 13.8% considered the contribution moderate, while 6.9% rated it as low and 3.4% were undecided. The high rate of value perception reinforces that the complex topics addressed at this stage were crucial for the students' technical maturity.

The Hands-On Module also demonstrated strong relevance for hard skills, with 65.5% of participants rating it as high and 27.6% as moderate. A percentage of 6.9% were undecided. Although slightly lower than the Advanced Module in the high contribution category, Hands-On established itself as a vital moment for the practical application of technical knowledge in a simulated scenario.

Thus, Table 5 was constructed, presenting the course modules hierarchically according to the degree of contribution to the students' hard skills, in the following order: Advanced, Hands-On, Intermediate, and Basic.

Table 5. Hierarchy of modules by degree of contribution - hard skills.

Module	Comment
Advanced	75.9% with a high degree of contribution, indicating it is the module with the greatest direct technical impact.
Hands-On	65.5% with a high degree of contribution, demonstrating high effectiveness in the practical application of concepts.
Intermediate	51.7% with a high degree of contribution, showing a quite significant technical impact.
Basic	37.9% with a low degree of contribution, suggesting a more limited initial technical impact.

Furthermore, the Hands-On Module subjects students to an industry-like simulation with frequent role rotation. Consequently, the demand for soft skills is constant, as teams must manage their own internal dynamics and project responsibilities. The fact that the majority of students considered the contribution of the Basic Module to be low may suggest that they already possessed knowledge of the foundational content taught in this module. Following this line of reasoning, subsequent modules obtained an increasingly higher percentage in the high contribution category, increasing from 24.1% in the Basic Module to 51.7% in the Intermediate Module and 75.9% in the Advanced Module, a progression of 27.6% and 24.2%, respectively, which suggests a technical evolution of the content presented as the course progresses.

To understand the direct connection between the technical foundation and the final practical application, students were specifically asked about the degree of contribution of the Advanced Module to the execution of the final project in Hands-On. The data revealed that 75.9% of the students rated this contribution as high and 17.2% as moderate, while 6.9% were undecided, as illustrated in Figure 8. No student selected the none or low options.

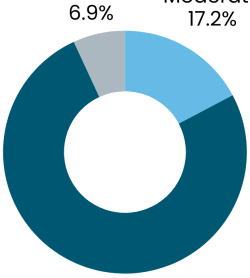

Figure 8. Contribution of the Advanced Module to Hands-On.

This result exposes how significant the technical foundation made possible, especially by the Advanced Module, was: the perception of 75.9% of high contribution indicates that the acquired technical knowledge was directly applicable and essential to satisfy client demands in the simulated environment proposed during the Hands-On period. This synergy explains why the Advanced Module leads to the perception of gain in hard skills, while Hands-On leads in soft skills. While the former provides the necessary technical tools (languages, frameworks, and architecture), the latter offers the real-world scenario for these tools to be tested under pressure, simultaneously requiring behavioral competencies for team and project management. Thus, the success in delivering the final project emerges as a strong indication that the course fulfilled its objective of competency integration.

7.5 Qualitative data

Regarding the development of soft skills, students from Cohort V were invited to share their perceptions. Qualitative analysis of the 29 responses revealed that 31% of participants reported an improvement in teamwork skills; 27.6% mentioned communication as an improved skill, 17.2% highlighted leadership, and 10.3% conflict management. Other mentioned skills include adaptation (6.9%), self-direction (6.9%), professionalism (6.9%), problem-solving (6.9%), and decision-making (6.9%). Skills such as IT application, understanding the client, ethics, organization, resilience, social responsibility, and knowing how to listen were cited by 3.4% of the students each.

Table 6 represents the qualitative analysis of the items that stimulated the development of soft skills according to the students, where 31% mentioned practice as the main factor, 20.7% highlighted Hands-On, and 6.9% cited the courses. Regarding more generalized opinions, 31% of the students stated that they improved overall, 17.2% mentioned greater self-perception, and 3.4% highlighted that the course provided a solid foundation for the development of these skills.

Table 6. Comments on soft skills.

Category (a)	Code (b)	% (b)	% (a)
Soft skills developed or improved	Teamwork	31	48.3
	Communication	27.6	
	Leadership	17.2	
	Conflict management	10.3	
	Adaptation	6.9	
	Self-direction	6.9	
	Professionalism	6.9	
	Problem-solving	6.9	
	Decision-making	6.9	
	IT application	3.4	
	Understanding the client	3.4	
	Ethics	3.4	
	Organization	3.4	
	Resilience	3.4	
	Social responsibility	3.4	
	Knowing how to listen	3.4	
Items that stimulated soft skills	Practice	31	41.4
	Hands-On	20.7	
	Courses	6.9	
Opinion	Overall improved	31	34.5
	Self-perception	17.2	
	Solid foundation	3.4	

It is possible to highlight that the most developed soft skills were communication and teamwork, with 27.6% and 31%, respectively. In addition, several skills were mentioned, such as conflict management, adaptation, self-direction, and decision-making. Practice was mentioned as the most relevant item for the development of soft skills, with 31%, followed by Hands-On with 20.7% of mentions. This suggests that, although Hands-On is an important component, general practice in course activities also stood out in the students' perception.

In this context, students were also questioned about which activities they considered responsible for the development of the soft skills they indicated. In the group of coding related to immersive Hands-On experiences, Hands-On was mentioned by 55.2% of the students; practical activities were mentioned by 17.2%; and interaction with the client appeared in 10.3% of the responses. Within the collaborative learning category, which encompasses various forms of team interaction, the following stood out: group activities (31.0%), teamwork (20.7%), conversations (13.8%), and assertive communication (6.9%), as shown in Table 7.

Table 7. Activities that stimulated soft skills.

Category (a)	Code (b)	% (b)	% (a)
Immersive Hands-On experiences	Hands-On	55.2	65.5
	Practical activities	17.2	
	Interaction with the client	10.3	
	Real-world environment	6.9	
	Project conflicts	6.9	
	Scrum experience	6.9	
Collaborative learning	Group activities	31.0	55.2
	Teamwork	20.7	
	Conversations	13.8	
	Assertive communication	6.9	
	Helping colleagues	3.4	
	Alignment meetings	3.4	
Formal course components	Courses	13.8	27.6
	Workshops	6.9	
	Sprint presentations	6.9	
	Advanced Module	3.4	
Problem management and solving	Problem-solving	6.9	10.3
	Assigning responsibilities	3.4	
	Task planning	3.4	
Other factors	Intellectual curiosity	3.4	6.9
	Overall improvement	3.4	

It is possible to observe that Hands-On was the activity most cited in the students' responses, comprising 55.2%, followed by group activities at 31%. Considering that a student can mention multiple items in their response, the last column of Table 7 shows the percentages of mentions made by students in each category, where immersive Hands-On experiences and collaborative learning were the most cited, with 65.5% and 55.2%, respectively. These results reinforce the importance of practical, hands-on activities in the development of soft skills, while simultaneously highlighting the need to adapt the course to meet a broader range of skills and student expectations.

Testimonials from students E20 and E21 reveal how the most cited categories contributed to the maturity and perception of some competencies. E20 highlighted the integration between Hands-On and group work: *"Hands-On [...] was the moment I used soft skills the most for various purposes, as well as in the training course classes where there were group assignments to be done; in general, the experience of doing things in groups was the main reason that made me develop soft skill capabilities."* E21 reinforced the role of collaborative activities in the courses: *"The vast majority of group activities in the courses were a factor that influenced the development of soft skills due to the need to cooperate with other people and knowing how to transmit what I thought to them."* These accounts show that students not only recognize teamwork as crucial but also better articulate its impact on the development of competencies such as collaboration, communication, and adaptability.

Course components such as workshops and Sprint presentations, which provide the theoretical foundation, play a complementary yet secondary role compared to the practical and collaborative experiences shown in the first part of Table 7. The course appears to be effective when it combines immersive practice with theoretical support and basic project management, but the key differentiator lies in the activities that simulate real situations.

Regarding the development of hard skills, students from Cohort V were also asked to comment on their perceptions. The coding of the 29 responses identified several categories of technical knowledge, as shown in Table 8. The most cited category was Languages and Frameworks (58.6%), with Java (44.8%), general mentions of frameworks (34.5%), Angular (31.0%), and Spring Boot (31.0%) being the most frequent areas of development. Software Engineering (48.3%) and Domains and Professional Profile (48.3%) were also prominent, highlighting programming (37.9%), front-end development (31.0%), and back-end development (27.6%) as significant gains. Additionally, students reported progress in Tools and Infrastructure (37.9%), with a focus on Git/Github (24.1%), and Data Management and Analysis (27.6%) centered on databases (20.7%).

Table 8. Comments on hard skills.

Category (a)	Code (b)	% (b)	% (a)
Languages and Frameworks	Java	44.8	58.6
	Frameworks (General Mention)	34.5	
	Angular	31.0	
	Spring boot	31.0	
	CSS	20.7	
	Javascript	17.2	
	HTML	13.8	
	Typescript	13.8	
	NodeJS	3.4	
Software Engineering	Programming	37.9	48.3
	Methodologies	13.8	
	Testing	6.9	
	Requirements Engineering	3.4	
Domains and Professional Profile	Frontend	31.0	48.3
	Backend	27.6	
	Web development	13.8	
	Full-stack	6.9	
Tools and Infrastructure	Tools (General)	27.6	37.9
	Git/Github	24.1	
	CI/CD	10.3	
	Apache	3.4	
	Containerization	3.4	
	Maven	3.4	
	Xampp	3.4	
Data Management and Analysis	Database	20.7	27.6
	Data analysis	10.3	
	SQL	3.4	
Architecture and Software Patterns	Architecture	13.8	20.7
	API rest	6.9	
	Integration	6.9	
	Microservices	3.4	
	SPAs	3.4	
Design and User Experience	Design	13.8	17.2
	Usability	3.4	

There was a notable consolidation of the Java and Spring

Boot ecosystem among the participants, cited by 44.8% and 31.0% of the students, respectively, as the areas of greatest technical mastery. Although the Angular framework was extensively practiced (31.0%), participant reports indicated a dichotomy in self-confidence, with students perceiving themselves as technically more proficient in back-end development than in front-end tasks. Additionally, the use of Git and GitHub went beyond isolated technical competence in version control, being identified by 24.1% of the participants as an essential instrument for collaboration and team organization.

Complementing the statistical findings, the qualitative analysis reveals that hard skill proficiency was driven by the practical necessity of integrating multiple software layers. Students reported that the ability to establish communication between the front-end and back-end, as well as the application of the MVC architecture, represented decisive technical milestones, transforming isolated concepts into autonomous competence. This technical consolidation was frequently attributed to the practical experience that required building robust and scalable applications, combining knowledge of databases, security, and requirements engineering in a real development environment.

To identify the pedagogical vectors responsible for the statistical growth observed, students were asked which activities were decisive for this improvement. Table 9 represents the qualitative analysis of the items that stimulated the development of hard skills according to the students. The coding of the responses reveals that the active methodology was the preponderant factor: the category of "Active Engagement" was cited by 58.6% of the participants, followed closely by "Project-Based Learning (PBL)" with 51.7%. Specifically, the "Hands-On Module" was highlighted by 48.3% of the group as the central catalyst for technical learning. Qualitative reports indicate that this stage is perceived not just as an academic task, but as a faithful simulation of the labor market, where the need to find dynamic solutions for real problems consolidated their knowledge.

Table 9. Activities that stimulated hard skills.

Category (a)	Code (b)	% (b)	% (a)
Active Engagement	Applied practical	51.7	58.6
	Collaborative practices	10.3	
	Information search	6.9	
Project-Based Learning	Hands-On	48.3	51.7
	Real-world challenges	10.3	
	Deadlines	3.4	
Foundation and Structured Curriculum	Formal instruction	31.0	48.3
	Advanced module	20.7	
	Instructors' pedagogy	10.3	
	Intermediate module	3.4	

Despite the emphasis on practice, formal instruction maintained a significant foundational role. The "Structured Curriculum" category was cited in 48.3% of the responses, with 31.0% explicitly mentioning formal instruction (classes) and 20.7% highlighting the Advanced Module. Cross-analysis suggests that the advanced theory provided the indispensable technical toolkit for project execution, enabling the subsequent complex practice. However, caveats regarding didactics in synchronous coding moments were noted, with reports of difficulties in following the pace of the technical demonstration. Finally, collaboration emerged as a technical driver: the use of versioning tools in groups was pointed out as a direct means of acquiring hard skills, reinforcing that in the context of PBL, peer exchange is an engineering learning activity in itself.

Similar to the findings regarding soft skills, these results reinforce that the integration of a solid theoretical foundation with immersive practical application is perceived by students as the most effective path for mastering hard skills.

7.6 Interviews

As specified in Section 7.3, students reported their perceived proficiency levels for a list of soft skills at both the beginning and end of the course. Although the qualitative responses largely highlighted positive development, the quantitative self-assessments revealed a discrepancy: 100% of respondents indicated the same level for at least one skill across both questionnaires. Furthermore, 65.5% of students reported a lower proficiency level for at least one skill in the final assessment compared to the initial one. To investigate this phenomenon, students who exhibited this behavior in their responses were invited to participate in an interview (57.8% accepted). Table 10 lists the questions asked in the interviews, as well as a categorization of the participants' responses.

Table 10. Analysis of the interviews - students.

Questions and categorization	%
1. What led you to mark those levels at the beginning?	
Based on academic background (undergraduate degree)	45.5
Based on a previous extension course	27.3
Previous experiences in jobs, projects, and/or internships	27.3
2. Initially, did you feel confident with your soft skills?	
Yes, due to prior experience	36.4
No, due to a lack of practical contact or clarity of the concept	63.6
3. After the course, did you feel more confident regarding your soft skills?	
Yes, especially after the Hands-On	72.7
Maintained the same confidence level	9.1
Mixed confidence ("yes and no")	18.2
4. Have you become more aware of the difficulty of soft skills?	
Yes, realized how challenging they are	100
5. Would you say that you learned more, but realize that you still have much to develop?	
Yes	100
6. Would you change anything in the initial self-assessment?	
No, the answers were honest at that time	54.5
Would change some specific assessments	45.5
7. Did the course have a positive or negative influence on your soft skills?	
Positive influence	100

The interview was conducted both in person and via videoconferencing, with six participants joining remotely and five attending in person. For the remote sessions, meetings were held via Google Meet[4] with access links distributed to participants' email addresses shortly before each scheduled session. For the on-site sessions, participants were requested to attend the interview at a designated location on the university campus at their scheduled time.

Each interview commenced with a brief overview of the research scope, its objectives, and the significance of the participant's contribution to the study. The interview then proceeded to explore the rationale behind the students' initial self-assessments, as indicated in the first questionnaire, where 45.5% responded that they based it on what they experienced in their undergraduate studies, 27.3% reported having based it on previous projects in which they participated, and 27.3% indicated previous experiences.

Question 2 sought to identify whether students felt secure regarding their soft skills at the beginning of the course; in this question, only 36.4% said they felt confident due to experiences in previous projects, while 63.6% said no, as they had little or no contact with practical projects previously.

The categorization of responses to question 3 showed that the majority felt more confident in their soft skills after Hands-On. This underscores the efficacy of practical, simulation-based activities in fostering soft skill development by replicating real-world professional environments. Furthermore, a holistic analysis of the interviews reveals a consensus among participants regarding the vital role of hands-on activities. Every interviewee emphasized that this practical approach was fundamental to both their personal and professional development.

Responses to question 4 indicated that all interviewees recognized how challenging soft skills are, and responses to question 5 showed that everyone feels they still have much to develop, despite having learned a lot in the course.

Regarding Question 6, nearly half of the interviewees indicated they would revise their initial self-assessments. This tendency toward retrospective reevaluation suggests that, after gaining practical experience, participants developed a more critical and realistic perspective of their baseline competencies. In their statements, they recognized that at the time they had a mistaken perception of their level in certain soft skills, but practical application revealed difficulties, indicating that the initial self-assessment should have been lower. The remaining portion considered their answers appropriate for the moment. As evidenced by the responses to Question 4, all participants recognized the inherent difficulty of mastering soft skills. This realization led to a significant shift in perspective, where students admitted that their initial self-assessments were based on a different, and perhaps less mature, understanding of these competencies. These responses align with the Dunning–Kruger effect (Kruger and Dunning, 1999), described as a cognitive bias that occurs when a person with less knowledge has a false sense of wisdom, while a person with greater knowledge feels unaware.

The interviews provided validation for the assumptions, such as the increase in students' confidence regarding their soft skills and the Dunning–Kruger effect. The increase in confidence was mainly attributed to the practical group activities carried out throughout the course, especially through the Hands-On Module. A joint analysis of both quantitative and qualitative datasets confirms the presence of the Dunning–Kruger effect. Records with lower assessments of soft skills in the final questionnaire compared to the initial one were investigated through interviews to determine if students self-assessed in that way because they did not recognize the difficulty of mastering them at the time, and if, after the course, they gained this perception.

7.7 Interviews - faculty perspective

A qualitative analysis of interviews conducted with four WebAcademy instructors was performed to capture the faculty's perspective on student development. This provided a complementary assessment of both technical and behavioral progress. The analysis focused on how instructors strategize for interpersonal skill development, the challenges faced, and their perception of the evolution of maturity, the transition from theory to practice, and the presence of cognitive biases, such as the Dunning–Kruger effect.

Table 11 presents a synthesis of the faculty perceptions extracted from the interviews, categorized according to the eight questions and the themes of the applied guide.

Table 11. Analysis of the interviews - instructors.

Themes and Categorization	**Main Observations and Nuances**
1. Competency Planning	Primary focus on hard skills, with soft skills (teamwork, communication, critical thinking, and problem-solving) occurring in an integrated manner or in the background.
2. Specific Activities	Use of group work, agile methodologies (Scrum), collaboration tools (Trello, GitHub), Design Thinking, and peer evaluation to map and stimulate competencies.
3. Assessment and Feedback	Predominantly informal, subjective, and occasional, occurring through close conversations or comments in technical tools when opportunities or conflicts arise.
4. The Integration Challenge	Disparity in students' backgrounds, dense/intensive workload, and the difficulty of measuring behavioral skills compared to technical ones.
5. Maturity and Autonomy	Clear evolution throughout the course, especially in commitment and engagement, being more noticeable in the transition to the Hands-On Module.
6. Dunning-Kruger Effect	Identified in students who arrive confident due to prior experience but recognize limitations when facing the real and complex practice of Hands-On.
7. Resources and Infrastructure	Evaluated as excellent, sufficient, and appropriate, with multimedia support and spaces that favor collaboration and teamwork.
8. Theory-Practice Balance	Considered balanced (frequently 50/50), although the intensive workload may hinder absorption for students with a greater initial gap.

Regarding planning and integration, instructors emphasized that while the primary focus remains technical, the development of soft skills is a constant, transversal concern.

[4] `https://meet.google.com`

Typically, these competencies are addressed indirectly; although activities prioritize technical evolution, they are structured to implicitly foster interpersonal skills, particularly through collaborative group work.

One of the most significant challenges identified was achieving class leveling, as students enter the program with highly heterogeneous technical backgrounds and varying levels of prior experience. Additionally, the intensive format of the classes was pointed out as a factor that can overburden students with less prior background, limiting the time for the consolidation of both technical concepts and interpersonal competencies. However, it was mentioned that the course syllabus in the first modules provides a solid foundation to prepare students for Hands-On, which is seen by instructors as a catalyst for students to seek to improve both their technical base and their communication and proactive research skills.

All instructors mentioned that they noticed greater student maturity during Hands-On. Due to the intensive nature of the classes, it was not easy to notice significant progress in students only during the courses, but throughout the program, the perception of this increase in maturity was a consensus. At this stage, the need to cooperate, transmit ideas, and resolve real conflicts forces a change in behavior that is not as visible in the rigid format of lecture classes. The instructors reported a significant increase in engagement, with students perceiving the training as a gateway to the professional market and a unique opportunity to progress in their developer careers.

The faculty accounts corroborate the presence of the Dunning–Kruger effect identified in the interviews with students. Instructors observed cases of students who arrived with significant confidence in certain subjects but showed surprise and recognized their limitations when they began to delve deeper into some concepts and also at the moment of applying them in activities. Interestingly, one of the instructors reported feeling the effect in themselves when delving into topics to teach the classes, perceiving the vastness of the content yet to be explored.

8 Threats to Validity

This section discusses potential threats to the validity of the results, considering internal, construct, external, and conclusion aspects, following established guidelines for empirical research in Software Engineering (Wohlin et al., 2012).

Regarding internal validity, four student dropouts occurred throughout the course, reducing the number of respondents from 31 to 29. Although these withdrawals took place before the Hands-On Module, a stage considered central for the development of interpersonal competencies, their absence may still have introduced sample bias. Students who discontinued the program may differ systematically in motivation, prior experience, or engagement levels from those who completed the training. Additionally, as the course spanned approximately seven months, maturation effects cannot be entirely ruled out, since participants may have developed technical or behavioral competencies through parallel academic activities, internships, or professional experiences occurring outside the program environment. To mitigate this maturation threat, the study employed data triangulation, contrasting student self-perceptions with continuous faculty observations. This approach allowed for correlating the reported development of competencies directly with the specific pedagogical activities of the modules, providing evidence that such growth was fostered by the PBL environment rather than solely by external factors.

From a construct validity perspective, the study relied predominantly on self-assessment instruments to measure both technical and interpersonal competencies. Self-perception data are inherently susceptible to cognitive biases, such as overestimation, social desirability, and misinterpretation of competency constructs. The identification of the Dunning–Kruger effect in both quantitative and qualitative findings reinforces this limitation, indicating that participants' perceived evolution may not fully correspond to their actual competency development. Although questionnaire items were grounded in prior literature, the instrument was adapted to the specific context of the training program and did not undergo external validation, which may affect measurement precision. To mitigate these risks, semi-structured interviews were conducted with selected participants, enabling interpretative triangulation between survey responses and discursive evidence.

Concerning external validity, the results are situated within the specific context of the WebAcademy extension program, implemented through partnerships with industry stakeholders and focused on full-stack web development. This configuration may limit the generalizability of the findings to traditional undergraduate Software Engineering curricula or to programs without strong industry integration. Furthermore, the heterogeneity of the cohort, including participants with different academic backgrounds, professional trajectories, and levels of prior experience, may influence the transferability of the conclusions to more homogeneous educational settings.

Finally, with respect to conclusion validity, statistical analyses were conducted using Shapiro–Wilk and Wilcoxon tests, which are appropriate for paired samples and non-parametric distributions. Nevertheless, the relatively small sample size may reduce statistical power and increase sensitivity to individual variation. Additionally, multiple hypothesis tests were performed across several competencies, which may elevate the risk of Type I errors. To address these limitations, we emphasize the nature of this research as a longitudinal case study, prioritizing the analytical depth of tracking a single cohort over a seven-month period. This framing justifies the sample size as appropriate for identifying nuanced developmental patterns within the WebAcademy ecosystem, focusing on deep contextual understanding rather than broad statistical generalization. Qualitative coding procedures were carried out collaboratively among researchers to enhance interpretive reliability, although subjective interpretation bias cannot be entirely eliminated. Despite these limitations, the triangulation of quantitative surveys, student interviews, and faculty perspectives contributes to strengthening the robustness and credibility of the findings.

9 Conclusion and Future Work

This study investigated the development of technical and interpersonal competencies within the context of an industry-aligned Software Engineering training program grounded in Problem-Based Learning principles. By combining quantitative self-assessment surveys with qualitative interviews and faculty perspectives, the research examined how immersive and practice-oriented learning environments contribute to the formation, perception, and calibration of competencies required in contemporary software development contexts. The findings indicate a consistent perceived evolution across both hard and soft skills, with particularly strong gains associated with collaborative and project-based activities conducted during the Hands-On Module. Competencies such as teamwork, communication, autonomy, and stakeholder interaction emerged as central developmental outcomes, reinforcing the role of experiential learning environments in fostering integrated technical and interpersonal capabilities.

From a technical standpoint, participants reported advances in programming practices, framework usage, and software delivery processes, suggesting that competency development occurred in a connected and mutually reinforcing manner rather than through isolated learning experiences. Statistical analysis confirmed significant growth across all nine technical topics investigated, driven primarily by the practical need to integrate different software layers and apply architectural patterns such as MVC. At the same time, variations in self-perception, including instances in which 65.5% of students indicated lower assessments in the final questionnaire, point to processes of cognitive recalibration.

The prominence of the Hands-On experience as a catalyst for competency development highlights the pedagogical relevance of situating learners in authentic development ecosystems that simulate professional dynamics. Agile ceremonies, role rotation, client interaction, and continuous delivery practices functioned not only as instructional strategies, but also as organizational, social, and relational learning mechanisms. Specifically, the mandatory rotation of the Scrum Master role at each sprint required students to manage project impediments and resolve conflicts, facilitating the transition from theoretical foundations to the resolution of real-world Software Engineering problems. These elements collectively contributed to approximating academic training to industry expectations, while exposing learners to the complexity, uncertainty, and accountability inherent to professional software production environments.

One of the interpersonal competencies evaluated through statistical tests, Teamwork ($p = 0.057827$), did not show a statistically significant improvement, contrasting with the strong qualitative evidence of collaborative maturity reported by both students and faculty. A plausible explanation for this lack of perceived growth is the 'team fatigue' inherent to intensive and long-term PBL environments. As the methodology demands constant, high-level interaction over seven months, students may reach a perceptual plateau in which collaborative practices are no longer viewed as a distinct developmental gain, but rather as an exhausting baseline requirement. Consequently, this saturation may mask actual skill development in quantitative self-assessments, even when professional maturation is evident in practice.

This study provides empirical evidence that immersive and industry-partnered Problem-Based Learning environments can simultaneously foster technical proficiency, interpersonal capabilities, and professional self-awareness. The mixed-methods design strengthens this contribution by triangulating perceptual, discursive, and observational data, mitigating limitations inherent in self-assessment instruments and enabling a more comprehensive interpretation of competency development processes. Furthermore, the extended duration of the training program allowed the observation of progressive maturation and recalibration of competencies over time, offering insights into how sustained project immersion influences both confidence formation and performance perception.

The results also present practical implications for the design of Software Engineering curricula and extension programs. Educational initiatives seeking to approximate professional realities may benefit from incorporating structured project immersion, stakeholder engagement, and agile governance models with high levels of learner autonomy as central pedagogical components. Rather than positioning soft skills as secondary or auxiliary learning outcomes, the evidence suggests that they emerge organically when learners are embedded in collaborative production environments characterized by shared responsibility, iterative delivery cycles, and real client expectations. Strengthening partnerships with industry and involving professionals as mentors may further enhance this alignment by providing learners with direct exposure to market dynamics.

In addition, the study contributes to ongoing discussions regarding the alignment between academic preparation and workforce demands in Software Engineering. By evidencing the co-development of technical and interpersonal competencies within a single training ecosystem, the findings reinforce the need for educational models that integrate knowledge construction, practice-based application, and social interaction dynamics. Such integration appears particularly relevant in contemporary software production scenarios, where coordination, negotiation, and collaborative problem-solving are inseparable from technical execution.

Despite these contributions, the study presents limitations that open avenues for future investigation. The reliance on self-assessment instruments, the specificity of the extension program context, and the relatively small sample size constrain statistical generalization. Consequently, the findings should be interpreted as contextually transferable rather than universally generalizable. Future studies may incorporate objective performance indicators, artifact-based evaluation, peer assessment, and longitudinal tracking of professional trajectories to enhance analytical robustness. Expanding the model to other Software Engineering domains, such as DevOps, software architecture, or quality engineering, may also reveal differentiated patterns of competency development and transferability.

In conclusion, the study reinforces the pedagogical and professional relevance of integrating Problem-Based Learning with industry-connected and project-driven training ecosystems. By evidencing not only perceived competency development but also processes of calibration and maturation over time, the findings contribute to advancing empirical discussions on how Software Engineering education can evolve to address the complex, collaborative, and practice-oriented demands that characterize modern software production environments.

Artifact Availability

The research artifacts for this study are openly available on Zenodo at `https://zenodo.org/records/15468807`. The repository includes: the questionnaires and the mapping of modifications between them, the scripts used for the statistical tests along with their results, and the detailed coding process applied to the qualitative responses.

Acknowledgments

This research is a result of the agreement between Motorola, Flextronics, UFAC, and Fundape, with funding provided by Motorola. The agreement was published in the Official Gazette of the Union No. 225 on December 1, 2021, utilizing resources from the Informatics Law for Western Amazonia (Federal Law No. 8,387/1991). Its disclosure is in accordance with the provisions of Article 39 of Decree No. 10,521/2020.

AI Use Disclosure

We disclose that Artificial Intelligence (AI) tools were used only for non-substantive editorial assistance. ChatGPT was employed for language refinement, Gemini (via Google Colab) was used to support the generation of illustrative figures, and Grammarly was used for minor proofreading. All methodological decisions, data analyses, and interpretations were performed by the authors alone. No AI tool contributed to the scientific content, claims, or conclusions of this paper.

References

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., and Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist*, 26(3-4):369–398.

Caeiro-Rodríguez, M., Manso-Vázquez, M., Mikic-Fonte, F. A., Llamas-Nistal, M., Fernández-Iglesias, M. J., Tsalapatas, H., Heidmann, O., De Carvalho, C. V., Jesmin, T., Terasmaa, J., and Sørensen, L. T. (2021). Teaching soft skills in engineering education: An european perspective. *IEEE Access*, 9:29222–29242.

Cavalcante, V., Costa, C., Cacau, A., and Oliveira, A. (2025). Como percepções e estatísticas revelam o desenvolvimento de habilidades técnicas e não técnicas em estudantes de computação. In *Anais do XXXIX Simpósio Brasileiro de Engenharia de Software*, pages 632–642, Porto Alegre, RS. SBC.

Cavalcante, V., Costa, C., Sarkis, L., Soares, D., and Freitas, S. (2023). Contributions of an extension course focused on good software engineering practices for students and it professionals. In *Anais do XXII Simpósio Brasileiro de Qualidade de Software*, page 301–310, Porto Alegre, RS. SBC.

Cavalcante, V., Costa, C., and Soares, D. (2024). Beyond code: the development of soft skills through training in software engineering. In *Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software*, pages 521–531, Porto Alegre, RS. SBC.

Field, A. (2018). *Discovering Statistics Using IBM SPSS Statistics*. Sage, 5. ed. Los Angeles, London, New Delhi, Singapore, Washington DC and Melbourne.

Garcés, L. and Oliveira, B. (2024). Teaching software engineering with project-based learning: A four years experience report. In *Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software*, pages 476–486, Porto Alegre, RS. SBC.

Garousi, V., Giray, G., Tuzun, E., Catal, C., and Felderer, M. (2020). Closing the gap between software engineering education and industrial needs. *IEEE Software*, 37(2):68–77.

Giani de Resende, R., Pelegrini Tristão, S., Rocha Neto, J. T. d., and Gaydeczka, B. (2021). Levantamento sobre hábitos de leitura e escrita entre estudantes de engenharia. *Revista (Entre Parênteses)*, 10(1):e021002.

Gil, A. C. (2002). *Como Elaborar Projetos de Pesquisa*. Atlas, 4. ed. São Paulo.

Giovannetti, V. E. G. (2023). Elas são maioria do volume de jogadores, mas não programam: por trás da baixa representatividade das mulheres na programação de jogos digitais no brasil. Available at: `https://hdl.handle.net/10438/33549`. Accessed on: April 20, 2026.

González-Morales, D., Moreno de Antonio, L. M., and Roda García, J. L. (2011). Teaching "soft" skills in software engineering. In *2011 IEEE Global Engineering Education Conference (EDUCON)*, pages 630–637.

Kruger, J. and Dunning, D. (1999). Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, 77(6):1121–1134.

Kumar, A., Raj, R., Aly, S., Anderson, M., Becker, B., Blumenthal, R., Eaton, E., Epstein, S., Goldweber, M., Jalote, P., Lea, D., Oudshoorn, M., Pias, M., Reiser, S., Servin, C., Simha, R., Winters, T., and Xiang, Q. (2024). *Computer Science Curricula 2023*. ACM Press, IEEE Computer Society Press and AAAI Press, New York, NY. DOI: `https://doi.org/10.1145/3664191`.

Lemos, W., Cunha, J., and Saraiva, J. (2019). Ensino de engenharia de software em um curso de sistemas de informação: Uma análise dos problemas e soluções na perspectiva de professores e alunos. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 305–318, Porto Alegre, RS. SBC.

Lethbridge, T. C. (2000). What knowledge is important to a software professional? *IEEE Computer Society Press*, 33(5):44–50.

Marques, A., Ferreira, B., Lopes, A., and Silva, W. (2020). Stimulating the development of soft skills in software engineering education through design thinking. In *Anais do XXXIV Simpósio Brasileiro de Engenharia de Software*, pages 690–699, Porto Alegre, RS. SBC.

Mendes, J., Costa, Y., Frazão, K., Santos, R., Santos, D., and

Rivero, L. (2019). Identificação das expectativas e dificuldades de alunos de graduação no ensino de engenharia de software. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 334–347, Porto Alegre, RS. SBC.

Moura, A. L., Ribas, J., de Lima, M. J., and Rodriguez, N. (2025). Fortalecendo a confiança das mulheres para atuar na área de computação. In *Women in Information Technology (WIT)*, pages 429–440. SBC.

Oran, A., Lima, R., Gadelha, B., Maia, N., Silva, W., and Rivero, L. (2023). Empowering technical skills and soft skills in software engineering students through problem-based learning. In *Anais do XXXVII Simpósio Brasileiro de Engenharia de Software*, page 348–357, Porto Alegre, RS. SBC.

Pfleeger, S. L. and Kitchenham, B. A. (2001). Principles of survey research: part 1: turning lemons into lemonade. *Association for Computing Machinery*, 26(6):16–18.

Podlaski, K., Beczkowski, M., Simbeck, K., Dziergwa, K., O'Reilly, D., Dowdall, S., Monteiro, J., Lucas, C. O., Hautamaki, J., Ahonen, H., et al. (2025). Software development projects as a way for multidisciplinary soft and future skills education. *Education Sciences*, 15(10):1371.

Portela, C. S., Vasconcelos, A. M., and Oliveira, S. R. (2015). Análise da relevância dos tópicos e da efetividade de abordagens para o ensino de engenharia de software. In *Fórum de Educação em Engenharia de Software*. VI Congresso Brasileiro de Software: Teoria e Prática (CBSoft).

Prince, M. (2004). Does active learning work? a review of the research. *Journal of Engineering Education*, 93(3):223–231.

Rodrigues, M., Maia, A., Rocha, M., Oliveira, L., and Marques, A. (2022). Desenvolvimento de soft skills durante a atuação no projeto meninas digitais do vale: achados de uma retrospectiva. In *Anais do XVI Women in Information Technology*, pages 34–44, Porto Alegre, RS. SBC.

Rodrigues, T., Lima, L., Araújo, A., and Santos, A. (2024). Análise da utilização de metodologia ativa para o ensino universitário em gestão de projetos. In *Caderno Pedagógico*, volume 21, page e11779.

Saar, C., Sarkis, L., and Soares, D. (2023a). O uso de metodologias ativas no ensino de engenharia de requisitos: O caso do projeto web academy. In *Anais da VII Escola Regional de Engenharia de Software*, pages 348–357, Porto Alegre, RS. SBC.

Saar, C., Soares, D., and Sarkis, L. (2023b). Projeto web academy: um relato sobre uma parceria entre indústria e universidade e os impactos no ecossistema local desenvolvimento de software. In *Anais Estendidos do XIV Congresso Brasileiro de Software: Teoria e Prática*, pages 21–24, Porto Alegre, RS. SBC.

SBC (2005). Currículo de referência para cursos de graduação em computação e engenharia de computação. Sociedade Brasileira de Computação. Porto Alegre, RS. Available at: `https://www.sbc.org.br/wp-content/uploads/2024/07/Curriculo-de-Referencia-CC-EC-versao2005.pdf`. Accessed on: April 20, 2026.

Seagull, D. G., Souza, G. G. P., and Barros, S. D. P. d. S. (2023). Análise das matrizes curriculares dos cursos superiores de ciência da computação sobre a ótica das soft skills: uma análise comparativa. Barueri, SP. Available at: `https://dspace.mackenzie.br/items/20cde617-cb11-4390-80c7-9d06e3de7d46`. Accessed on: April 20, 2026.

Sedelmaier, Y. and Landes, D. (2018). Practicing soft skills in software engineering: A project-based didactical approach. In *Computer Systems and Software Engineering: Concepts, Methodologies, Tools, and Applications*, pages 232–252. IGI Global Scientific Publishing.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.

Siegel, S. and Castellan, N. (1988). *Nonparametric Statistics for the Behavioral Sciences*. McGraw-Hill, 2. ed. New York.

Silva, B., Neto, V., and Gritti, N. (2020). Soft skills: rumo ao sucesso no mundo profissional. *Revista Interface Tecnológica*, 17(1):829–842.

Silva, L. and Albuquerque, D. (2023). Identificação de soft skills a partir da avaliação de anúncios de vagas em tecnologia da informação. In *Anais do X Encontro Nacional de Computação dos Institutos Federais*, pages 5–8, Porto Alegre, RS. SBC.

Souza, M., Moreira, R., and Figueiredo, E. (2020). Students perception on the use of project-based learning in software engineering education. In *Anais Estendidos do XXXIII Simpósio Brasileiro de Engenharia de Software*, Porto Alegre, RS. SBC.

Spencer, D. and Garrett, J. (2009). *Card Sorting: Designing Usable Categories*. Rosenfeld Media, New York.

Thomas, J. W. (2000). A review of research on project-based learning. Technical report, Autodesk Foundation, San Rafael, CA.

Vogler, J., Thompson, P., Davis, D., Mayfield, B., Finley, P., and Yasseri, D. (2018). The hard work of soft skills: augmenting the project-based learning experience with interdisciplinary teamwork. In *Instructional Science*, volume 46.

Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, 1. ed. Cambridge.

Wazlawick, R. S. (2019). *Engenharia de Software: conceitos e práticas*. Elsevier Editora Ltda, 2. ed. Rio de Janeiro.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., and Wesslén, A. (2012). *Experimentation in Software Engineering*. Springer, 2012. ed. Boston.

Zimmermann, T. (2016). Card-sorting: From text to themes. In Menzies, T., Williams, L., and Zimmermann, T., editors, *Perspectives on Data Science for Software Engineering*, pages 137–141. Morgan Kaufmann, Boston.