Learning through Practice: Teaching Empirical Software Engineering for Undergraduate Students

Ranya Duran Greco
Federal University of Ceara
Russas, Brazil
ranyagreco@alu.ufc.br

Sofia Bento Desidério
Federal University of Ceara
Russas, Brazil
sofiadesiderio@alu.ufc.br

Caio César Sousa Bandeira
Federal University of Ceara
Russas, Brazil
caiocezzar@alu.ufc.br

Anna Beatriz dos Santos Marques
Federal University of Ceara
Russas, Brazil
beatriz.marques@ufc.br

ABSTRACT

Empirical Software Engineering (ESE) is essential for understanding the benefits and limitations of existing software engineering solutions. The evidence obtained through experiments guides the evolution of this area and optimizes the use of effort in relevant research that solves real problems or gaps. In ESE teaching, the challenge arises of preparing future software engineers to plan, conduct, and analyze experiments. This article reports on an experience of teaching ESE to 54 undergraduate students in an elective course, using Active Learning. To this end, we describe the activities carried out during the course, and to understand the effect of Active Learning, we conducted a survey using an online questionnaire. Twenty-seven students from the class consented to take part in the survey. The results show that 77.7% of the respondents had little or no experience of ESE before the course. At the end of the course, around 88.2% reported greater confidence in carrying out research in the area. The hypothesis test showed a significant increase in knowledge of the course's content, emphasizing the topics of surveys and empirical studies. Correlation analysis revealed that the perceived contribution of the course is positively associated with student motivation. Despite some challenges, the students highlighted the methodology applied as being of great value to their training and entry into the job market. The experience of using Active Learning to teach ESE showed that, as well as contributing to the students' education, the methodology gave them more confidence and motivation, which suggests the importance of pedagogical approaches that are closer to practice.

KEYWORDS

Empirical Software Engineering, Teaching, Education, Active Learning, Undergraduate Student

1 Introduction

The Empirical Software Engineering (ESE) course poses a significant educational challenge [32], requiring innovative teaching methodologies to bridge the theory-practice gap. While specialized literature acknowledges the growing importance of empirical methods [14], there remains a paucity of studies on effective approaches for integrating these principles into higher education curricula.

The practical nature of ESE [19, 22] underscores the need to integrate empirical practices into undergraduate curricula, preparing students for real-world Software Engineering challenges. Students must acquire theoretical knowledge and develop specific competencies to design, conduct, and analyze experiments. However, conventional teaching methods often fail to provide adequate opportunities for practical application of ESE concepts, leading to difficulties when students encounter actual research scenarios [6].

Despite ESE's critical role in empirically validating software artifacts and methods, its integration into undergraduate Computer Science and Software Engineering curricula remains limited [15]. Many academic programs prioritize theoretical instruction over practical development, leaving students ill-prepared for evidence-based projects. This gap may hinder their professional readiness, as noted by Garousi et al. [9], whose research identifies empirical practice integration as crucial for bridging this educational divide.

Active Learning emerges as a promising alternative [11]. Its principles – emphasizing student agency and participatory engagement – prove particularly relevant for ESE education, where hands-on experiences are essential for knowledge construction. However, effective implementation requires careful adaptation to Software Engineering's unique context [10].

This paper reports an instructional experience teaching ESE to 54 undergraduate students across Software Engineering and Computer Science programs at the Federal University of Ceara, Russas campus. Active Learning principles were implemented following methodological approaches similar to those described by Meireles et al. [17], but with a course specifically designed for undergraduate cohorts. The course integrated theoretical foundations [32], critical analysis of existing studies, and hands-on empirical research activities [22], establishing a continuous learning pathway to enhance students' research competencies.

To understand how students experienced the teaching-learning process, we adopted a mixed-methods approach [3], combining quantitative and qualitative analyses of student perceptions. This methodology enabled a comprehensive evaluation of both knowledge progression and the learning experience, while identifying which aspects of the instructional method were most effective.

2 Background and Related Work

2.1 Empirical Software Engineering

Software Engineering requires a solid scientific foundation to advance beyond mere intuition or individual experience. Empirical Software Engineering (ESE) emerges as a significant field in this context, providing structured methods to validate theories, assess technological innovations, and optimize development practices through systematic testing. Fundamentally, ESE aims to transform software engineering into an evidence-based discipline, where robust data guide decisions [2].

Creating, observing, and evaluating models representing software processes, products, and their development environments is essential [1]. These models provide a foundation for a scientific understanding of software development, enabling experts and researchers to analyze interactions systematically. ESE supports this mission through controlled experiments, case studies, and comprehensive reviews, generating reliable and reproducible knowledge.

Empirical methods can be applied throughout the software lifecycle – spanning requirements engineering to maintenance, and encompassing organizational aspects such as process improvement and team dynamics [33]. This flexibility makes ESE fundamental in academic research and industrial practice, where theory and practical application converge. Experiments may evaluate different testing approaches, while case investigations demonstrate how teams adapt to new tools in real-world contexts.

Teaching ESE presents unique challenges. Wohlin [31] discusses a few strategies: integrating it with other study areas or offering it as a standalone course. Each approach has merits – integration reinforces empirical thinking across multiple contexts, while a dedicated course enables more profound exploration of experimental design and data analysis. The optimal strategy is a hybrid: introducing empirical thinking early, with specialization options for those seeking advanced study.

2.2 Active Learning

Active Learning (AL) is an educational approach that places students at the center of the knowledge acquisition process, fostering their active participation. This method transforms learners from passive listeners into engaged participants, motivating them to participate in meaningful activities and critically reflect on their actions. This methodology promotes deeper learning by positioning students as the protagonists of their education, cultivating autonomy and critical thinking [4]. Unlike traditional teaching focused on information transmission, AL emphasizes direct experiences, peer interaction, and continuous practice.

In the context of teaching ESE, AL methodologies are particularly valuable. Rather than passively attending lectures on scientific methods, students are guided to design, conduct, and analyze real experiments, confronting challenges similar to those in professional settings. This approach enhances student engagement while fostering interdisciplinary connections and developing crucial skills, including critical analysis, data interpretation, communication, and teamwork [21].

Valença [28] identifies these key AL features: (1) active student participation in knowledge construction, fostering collaboration; (2) critical thinking development; (3) problem-solving emphasis; and (4) hands-on application of learned concepts. In alignment with this approach, the course promoted active student participation through guided studies and seminars, developed critical thinking in Hands-on Work 1, and emphasized problem-solving and the practical application of concepts in Hands-on Work 2. The hands-on lab sessions comprised quantitative and qualitative analysis, as described in subsection 3.1.

Other essential elements of AL include continuous feedback and systematic reflection on outcomes [23]. This approach encourages students to iteratively revise their projects, critically identify limitations, propose concrete improvements, and formally present their findings through seminars and technical reports. Through this process, they develop advanced academic and scientific communication skills essential for professional research contexts.

Finally, this approach proves particularly effective for teaching scientific methods, as it transforms theoretical concepts into hands-on experiences [21]. By designing experiments, collecting data, and analyzing results, students deepen their understanding and cultivate an investigative and critical mindset – essential for professional practice in Software Engineering.

2.3 Related Work

Meireles et al. [17] investigated the need for non-traditional teaching methods in ESE, proposing AL as an alternative. The study describes a 60-hour course for 36 undergraduate and graduate students, covering primary studies (such as a case study, controlled experiment, and survey) and secondary studies (such as systematic mapping and review). To evaluate student satisfaction with the course, data were collected through a questionnaire addressing three key areas: (1) perception of learning, (2) perception of motivation, and (3) perception of the value of learning. Quantitative and qualitative data analyses demonstrated positive impacts on students' knowledge retention. Our work adopts the questionnaire proposed by Meireles et al. [17] and adapts it for a single target audience: undergraduate students. The teaching process with AL also differs in some aspects, which are discussed in greater detail in Section 3.1 of this report.

Vilela and Silva [29] focused their efforts on a smaller cohort of 17 undergraduate students. Some activities were conducted individually while still employing AL methodologies, except for seminars and empirical practices. The research team collected data regarding: course feedback, perceived ease and usefulness of activities, positive course aspects, and potential improvements – all from the students' perspective. Additionally, students evaluated each other's research projects using predefined criteria, and these results were utilized to measure class performance. The key differences between our work and Vilela and Silva [29] are the focus on collaborative activities and the absence of peer evaluation.

Gren [10] described a flipped classroom course for teaching ESE, focused on master's students. In the flipped format, pre-class contained videos and online exercises to retain the content. During class sessions, students engaged in interactive activities such as group discussions, experiment design sessions, and problem-solving votes using a clicker. Results obtained through grade analysis and a course feedback survey showed performance variations depending on the instructor, suggesting the need for further research to draw more conclusive evidence about flipped classrooms' effectiveness. Our work differs from Gren [10] as we did not adopt a flipped classroom and target undergraduate students.

3 Methodology

This section presents the methodology adopted in the teaching process and in the gathering of data regarding the students' perception.

3.1 Teaching Process

The Empirical Software Engineering course is a 64-hour elective offered to undergraduate students in both Software Engineering and Computer Science programs at the Federal University of Ceara, Russas campus. The general objective of the course is to train the student in the fundamentals of ESE, encompassing primary and secondary studies. The syllabus includes: conceptualization and clarification of controlled experiments, case studies, and surveys; the process of developing a research project (including activities, formulation of research questions, theory building, and qualitative/quantitative data analysis); investigation of experiments in software engineering; and supervised practice through a small-scale software engineering experiment. The course is organized into 32 classes, each lasting 2 hours. In the experience reported in this paper, the course was conducted by one teacher and two teaching assistants. The class had 54 students: 26 from the Software Engineering course and 28 from the Computer Science course.

The adopted methodology includes lectures with open dialogue, guided study, seminar presentations, paired/group exercises, and group hands-on work. Each topic was delivered as follows:

- **Primary Studies:** the teacher delivered interactive lectures based on Wohlin et al. [32], covering fundamental concepts, the importance of experimentation in Software Engineering, and primary studies (including controlled experiments, case studies, action research, and personal opinion surveys). Selected examples of primary studies were briefly presented during classes.
- **Secondary Studies:** the teacher conducted interactive lectures based on Kitchenham [12], covering systematic literature reviews, systematic mapping, and tertiary reviews. Selected examples of secondary and tertiary studies were briefly presented during class sessions.
- **Controlled Experiment:** two guided study sessions were conducted on this topic, with students working in groups of up to three. The first session required students to read and prepare a presentation on Chapters 6 and 7 of Wohlin et al. [32], which cover the experimentation process and experiment scoping. The second session involved reading and presenting Chapters 8 and 9 of the same book, focusing on experiment planning and execution. Students delivered seminar-style presentations during class meetings based on their guided study outcomes.
- **Statistical Analysis of Controlled Experiments:** the teacher delivered an interactive lecture covering descriptive statistics, variable types and scales, sample normality, and hypothesis testing. A hands-on lab session was subsequently conducted by teaching assistants using JASP[1] statistical software, following a step-by-step analysis protocol that included: (1) tabulation of controlled experiment data, (2) descriptive statistical analysis, (3) normality testing, and (4) hypothesis testing. The data from the controlled experiment reported by Fonseca et al. [8] was used in the hands-on activity.
- **Hands-on Work 1 - Understanding a controlled experiment and replicating the statistical analysis:** students formed groups of three to: (1) analyze a scientific paper describing a controlled experiment, (2) extract key methodological details (objectives, context, variables, treatments and hypotheses) and (3) perform comprehensive statistical analysis including descriptive statistics, normality testing and hypothesis testing. Each group presented its findings in class.
- **Case Studies:** students completed third guided study on Chapter 5 of Wohlin et al. [32], which examines case study methodology in Software Engineering.
- **Personal Opinion Surveys:** this topic involved a fourth guided study focusing on Chapter 3 of Shull et al. [25], addressing the implementation stages and challenges specific to survey-based research in Software Engineering contexts.
- **Qualitative Analysis:** a lecture with open dialogue was conducted by the teacher based on the materials from the short course by Martinelli et al. [16], addressing qualitative analysis methods such as open coding, closed coding, and Grounded Theory. A hands-on classroom lesson was held to practice open data coding. A hands-on lab session was conducted by the teaching assistants to practice closed coding using the Taguette[2] tool and to generate visualizations of qualitative data using the Flourish[3] tool. A portion of the qualitative data sample from the study reported by Desidério et al. [5] was used during the hands-on sessions.
- **Hands-on Work 2 - Conducting an empirical study:** student groups (maximum six members) designed, executed, and analyzed a primary study, with study types (controlled experiment, case study, or survey) randomly assigned (three teams per method). The assignment comprised two deliverables: (1) a study plan presentation detailing goals, variables, participants, context, data collection procedures, and instruments; and (2) a results presentation covering execution details, statistical or qualitative analysis, and reflective learning outcomes.

The teaching and learning methodology is strongly inspired by the approach described by Meireles et al. [17], with several adaptations implemented to better align the activities with the undergraduate student profile of the class:

- Introductory classes conducted by the teacher on primary and secondary studies.
- Hands-on lab sessions on statistical analysis using the JASP tool.
- Inclusion of qualitative analysis content with hands-on lab sessions using the Taguette and Flourish tools.

[1]JASP: https://jasp-stats.org/

[2]Taguette: https://www.taguette.org/

[3]Flourish: https://flourish.studio/

- Group debates were not conducted due to the classroom layout and class size. However, in all seminars, students were able to clarify their questions with the support of the teacher.
- The topic of Design Science Research was not covered due to the class profile, which did not include graduate students.
- The hands-on works from Meireles et al. [17] involved conducting the first filter of a systematic literature mapping, the re-analysis of a controlled experiment, and the planning of a new controlled experiment. Our experience focused on hands-on practice with primary studies, maintaining the re-analysis of a controlled experiment, and introducing an assignment for the planning, executing, and analyzing results from a primary study, which could be a controlled experiment, case study, or survey.

3.2 Students' Perception Survey

Data collection was conducted through a personal opinion survey [13], based on the instrument adopted by [17]. The instrument was an online questionnaire hosted on the Google Forms[4] platform. The questionnaire comprised 31 questions – 24 closed-ended and seven open-ended—organized into five sections: (1) an initial section on student demographics, (2) perception of learning – before the course, (3) perception of learning – after the course, (4) perception of motivation of learning and (5) perception of the value of learning. The complete questionnaire and analysis methods applied to each variable are available in the Zenodo repository.

Participation in the survey was voluntary, meaning students were not required to answer the questionnaire and could withdraw without penalty. Before starting the study, all participants received and agreed to the Informed Consent (IC) form, which ensured the voluntary nature of participation, data confidentiality, and absence of risk. The IC explained that the responses would be used exclusively for academic purposes, guaranteeing anonymity and the non-disclosure of personal information in publications or research findings.

The questionnaire was distributed through three main channels: the university's academic management system (Sistema Integrado de Gestão de Atividades Acadêmicas - SIGAA) the monitoring team's Discord channel, and the institutional email list. The participants had approximately one month to respond. The survey was conducted with students enrolled in the ESE class during the semester under analysis. A total of 27 students participated – 14 from Software Engineering and 13 from Computer Science – representing 50% of the class. Notably, this response rate exceeds what is typically considered adequate for online surveys [7, 18]. The questionnaire was administered at the end of the academic semester, after students had completed all planned course activities. A mixed-methods approach was adopted for data analysis, with all data analyzed both qualitatively and quantitatively.

Statistical analyses were conducted using both the JASP[5] tool, an open-source software with a graphical interface that integrates various statistical tests, and Python[6] libraries such as Scipy[7], Pandas[8], Seaborn[9] and Matplotlib[10]. The analyses were implemented and executed in the Google Colab[11] environment, which enables Python programming directly in the browser, facilitating collaboration and notebook sharing. The non-parametric hypothesis analyses included the Wilcoxon signed-rank test for comparing paired distributions [30]. Spearman's rank correlation was used to assess the strength and direction of associations between variables [26]. Additionally, the Shapiro-Wilk test was applied to check for distribution normality [24], and the Student's t-test was used when appropriate [27]. Bar charts, boxplots, and heatmaps generated with Seaborn and Matplotlib were used to support the visualization of results. Two researchers conducted a quantitative analysis.

For the qualitative analyses, the software Taguette[12], a free and open-source content analysis tool, was adopted to code the textual responses. Additionally, Google Sheets was used to verify the incidence of codes in each section of the questionnaire. This combined approach enabled a better understanding of participants' perceptions and ensured the identification of patterns in the responses. The qualitative analysis was conducted by two people using the open coding method proposed by Saldaña [20], and validated by a third person.

4 Results

This section presents the results obtained from data collected through the survey and grade/activity tracking spreadsheets. Quantitative data are presented using graphical representations and statistical analysis, while qualitative data are presented through identified codes and supporting quotations (the codebook is available in the authors' Zenodo repository).

4.1 Class Engagement and Performance

Figure 1 shows the percentage of students who completed the guided studies and seminar presentations. Each student was required to present at least one guided study in class, which constituted a seminar. Student engagement exceeded 85% for all guided study activities. However, not all students who completed the guided studies presented their seminar. Nevertheless, 80% of students participated in seminar presentations, indicating strong overall class engagement.

According to observations from the teacher and teaching assistants, at the beginning of the guided studies, some students had not understood that they were expected to complete all guided studies, rather than only those they would present as seminars. This may explain the lower engagement percentage in the first guided study (GS1).

Student performance was evaluated based on grades obtained in practical assignments, with a maximum score of 7.0 points. Figure 2 shows the grade distribution, indicating that most scores were above 4.5 in HW1 and above 6.0 in HW2. HW1 was completed

[4]Google Forms: https://docs.google.com/forms/
[5]JASP: https://jasp-stats.org/
[6]Python: https://python.org/
[7]Scipy: https://scipy.org/
[8]Pandas: https://pandas.pydata.org/
[9]Seaborn: https://seaborn.pydata.org/
[10]Matplotlib: https://matplotlib.org/
[11]Google Colab: https://colab.research.google.com/
[12]Taguette: https://www.taguette.org

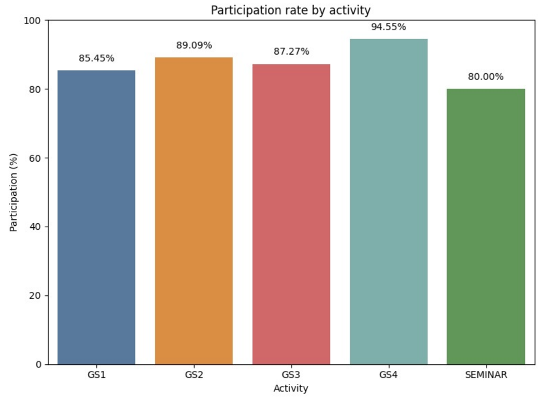

Figure 1: Student engagement in the proposed activities.

by 20 teams. The main challenges observed in HW1 related to the learning curve of statistical analysis. Four teams failed to reproduce statistical tests and interpret results, or did not understand they needed to redo the statistical analysis. Two teams omitted key steps in statistical analysis, such as normality testing or selected incorrect hypothesis tests. A positive finding was that several teams chose to explore alternative methods for statistical analysis using Google Colab[13] with Pandas, a Python library.

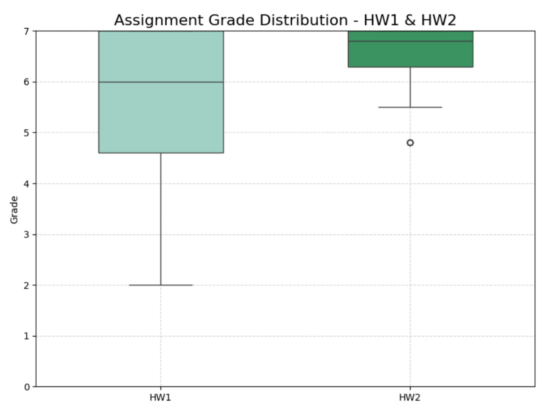

Figure 2: Student performance in the proposed Hands-on Works.

The average score was 5.623 for HW1 and 6.494 for HW2, indicating satisfactory performance given the maximum grade of 7.0. In both assignments, some teams achieved the maximum score. HW2 showed lower grade variability and a slightly higher median compared to HW1, suggesting more consistent and overall better performance in the second assignment. HW2 involved conducting a new experiment, designed to value students' experiences by allowing them to define their own experimental objectives.

Table 1 summarizes the types of studies conducted by each team and their respective topics. The scope of primary studies focused on the Russas campus of the Federal University of Ceara, including

[13]Google Colab: https://colab.research.google.com/

Table 1: Summary of primary study topics conducted in Hands-on Work 2.

#T	Study Type	Topic
1	SV	Benefits of gamification in the Software Architecture course
2	CS	Adoption of Scrum practices and its benefits/challenges in a campus software project
3	CS	Analysis of team experiences in Hands-on Work 2 of the Empirical Software Engineering course
4	CE	An experiment on pair programming vs individual programming in the Web Development class
5	SV	Impact of artificial intelligence on academic learning in Software Engineering and Computer Science programs
6	CE	How participation in campus projects influences student performance in the Software Quality and Verification & Validation course
7	SV	Usability evaluation of university's academic management system
8	CS	Measurement of efficiency, performance and maintainability in a campus software project
9	CE	Comparative analysis of Netflix and Prime Video streaming services regarding usability
10	SV	IDE preferences among software engineers (off-campus)

Legend: #T - team identifier; SV: Survey; CS: Case Study; CE: Controlled Experiment.

its classes and extension, research, and development projects. Only the survey conducted by Team 10 had an external scope beyond campus. All teams freely defined their topics and validated them with the instructor regarding appropriateness and consistency with their assigned primary study type.

To investigate a potential relationship between class engagement and performance, a correlation analysis was conducted. First, a normality analysis of the variables engagement[14], HW1, and HW2 was performed using the Shapiro-Wilk test. Results indicated non-normal distribution for all variables ($p < 0.00000000001$). The test statistics were: Engagement (0.5265), HW1 (0.7775), and HW2 (0.6795), rejecting the null hypothesis of normality. Consequently, Spearman's correlation coefficient was selected as more appropriate for non-parametric data, as it relies on value rankings and is less sensitive to outliers and linearity assumptions.

The results, represented in the heatmap shown in Figure 3, revealed weak to moderate correlations between variables. The correlation between engagement and HW1 was virtually null (r = 0.05),

[14]Engagement was measured as the number of completed activities per student (guided studies and seminar presentations).

indicating little to no influence of initial engagement on first assignment performance. Between engagement and HW2, we observed a weak but more evident correlation (r = 0.23), suggesting a slight trend of improved performance with higher engagement. The correlation between HW1 and HW2 was moderate (r = 0.35), indicating some consistency in student performance. These findings suggest a possible gradual influence of engagement on academic performance, with more noticeable effects in later assessments, albeit limited in magnitude.

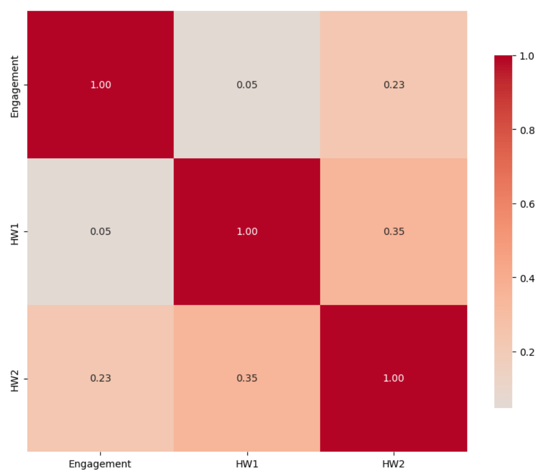

Figure 3: Heat map of Spearman correlation coefficients between student engagement and performance assessments (HW1 and Hw2).

4.2 Perception of Learning

The form presented earlier was divided into five sections. The first two cover students' perception of their own learning before and after the course.

Before the course, approximately 77.7% of respondents reported having little or no prior experience with ESE (Figure 4).

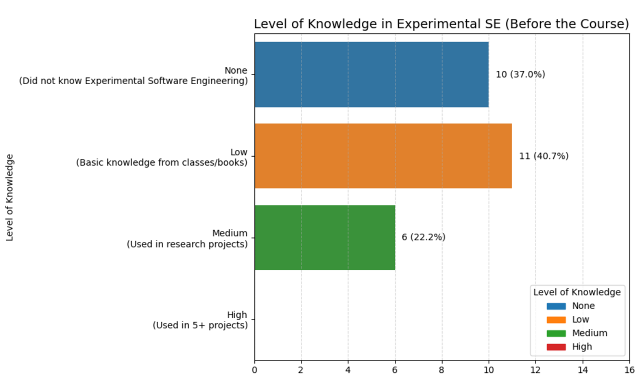

Figure 4: Distribution of self-reported initial knowledge levels in ESE before the course.

Through open coding of the question **"Before the course, which Empirical Software Engineering (ESE) contents had you previously used in conducting studies?"**, seven codes were identified. The most familiar content among students was **COD1.1 - Survey**, with reports on the use of forms, question types, interviews as data collection tools and Informed Consent Forms (ICF): A15: "*Due to my undergraduate thesis, I had to research and understand usability and UX evaluation methods, so I sought to learn more about the methods I would apply, which were surveys and controlled experiments. I learned what types of questions I should include in the questionnaire, about the ICF aspect*" and A20: "*The use of questionnaires in scientific research*". **COD1.2 - Data Analysis** was the second most frequent, covering descriptions of quantitative and qualitative analyses: A13: "*I think none; the most I had used were data analyses with charts, but not in the same way as we do now*" and A23: "*Only statistical analysis*". **COD1.3 - Controlled Experiment** was also mentioned as an example of primary studies they conducted: A18: "*I was already familiar with some research types, such as surveys, case studies and controlled experiments*" and A15: "*preparing artifacts for user tests and how to conduct a test*". Other less frequent topics included **COD1.4 - Case Study** (A03: "*Case Study and Survey*"), **COD1.5 - Systematic Literature Review (SLR)** (A21: "*Qualitative analysis, survey, systematic review*"), **COD1.6 - Ethics in ESE** (A25: "*Ethics in experimentation, results analysis*") and **COD1.7 - Scientific Method**.

In the learning assessment section after the course, there is a Likert-scale question for students to rate their perceived ability to conduct research using the contents taught during the course. 88.2% of participants agreed that they felt more capable of conducting research (Figure 5).

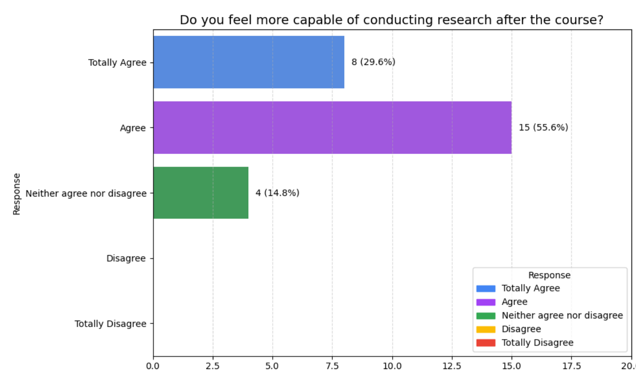

Figure 5: Student responses regarding their perceived capability to conduct research with empirical Software Engineering content after completing the course.

Immediately after this question, there was an open-ended question allowing respondents to justify their self-perception. From this question, six codes were identified. The positive self-efficacy primarily stemmed from **COD2.1 - Theory as Foundation** of the course. This characteristic can be observed in the following reports: A15: "*The course helped me better understand applicable research methods*", A10: "*I now have a clearer vision of how to define methods*" and A04: "*The course helped in understanding and learning what is needed and* *how to conduct research*". Another identified factor was **COD2.2 - Practice as Foundation**, where many students described research activities performed during the course, particularly in practical assignments: A12: "*Confidence comes from practice and being able to execute a research project from planning to analysis builds significant confidence*". As a result of this mixed teaching methodology, some excerpts indicated increased **COD2.3 - Self-confidence** in conducting, analyzing and/or replicating future research: A17: "*I believe I would be capable of conducting other research*". The **COD2.4 - Continuous Support** during the course was also cited as an influential factor in self-efficacy. Some students expressed limited confidence in their reports, highlighting **COD2.5 - Remaining Difficulties** in understanding research stages (A23: "*I may still struggle with defining the research scope*"). Consequently, only one excerpt was mapped to **COD2.6 - Low Self-confidence** (A19: "*I don't feel confident conducting any study under all circumstances, but the course provided a good foundation*").

Throughout the semester, the students were instructed to participate in guided studies that covered content related to ESE. From the responses to the question **"How did conducting the guided studies – carried out autonomously by students outside the classroom, organized in groups, and periodically presented to the class – impact your learning? How did this methodology facilitate or hinder your understanding of the contents? Justify your answer."**, eleven codes were mapped. The analysis revealed several developed skills and facilitators, including: **COD3.1 - Information Search/Organization**, in reports describing the active process of locating, selecting and structuring key content for presentation materials (A20: "*researching the content to prepare slides and present to the class made it more engaging*" and A25: "*Creating the slides required us to organize information logically and with visual clarity*"); **COD3.2 - Learning by Teaching**, content understanding achieved through teaching it to peers (A25: "*This reinforced my content retention since teaching is one of the most effective ways to consolidate knowledge*"); **COD3.3 - Autonomy**, independent knowledge acquisition (A09: "*I believe the guided studies made me pursue answers to my questions independently*"); **COD3.4 - Collaborative Work** for excerpts demonstrating proactive peer learning and collective contribution (A10: "*The guided studies facilitated my learning by encouraging active knowledge-seeking and idea exchange within the group*"); **COD3.5 - Learning by Practice**, knowledge retention through creating presentation materials (A02: "*Yes, it helped me gain knowledge through research and study, highlighting crucial aspects of research itself*"); and **COD3.6 - Motivation**, internal or external factors positively influencing student engagement (A11: "*I put more effort into learning the content because I had to teach it to classmates*").

Although less frequent (ten excerpts), difficulties in executing the guided studies were also identified: **COD3.7 - Time Management Challenges**, for reports describing situations where intensive university routines hindered the required dedication for deliverables (A12: "*For others, I felt I was on autopilot to complete semester deliverables, so I relied heavily on GPT and engaged less critically*"); **COD3.8 - Limited Autonomy** and **COD3.9 - Motivation Challenges**, comments suggesting that for some students, the guided studies had the opposite effect than intended, generating demotivation or difficulty in self-directed learning (A11: "*but in some aspects it was much harder because I had to do the study alone*" and A15: "*it was interesting to learn from peers, but due to the extensive and somewhat complex content, it demotivated us from understanding parts we weren't responsible for presenting*"); **COD3.10 - Language Barrier**, obstacles in comprehending foreign-language materials (A01: "*English-language papers presented some difficulty in understanding the content*"); and **COD3.11 - Content Complexity**, the perception that suggested content was challenging (same report as COD3.10).

To evaluate participants' knowledge evolution in ESE, hypothesis tests were applied to pre and post-course response differences. The paired t-test was used for normally distributed topics and the Wilcoxon test for the remaining ones. The tests covered the following Likert-scale assessed topics: knowledge level in (1) Statistical analysis of experiments, (2) Qualitative analysis, (3) Case study, (4) Experimental studies, (5) Controlled experiment, (6) Systematic Literature Review (SLR) and (7) Survey.

The results indicated significant knowledge improvement across all analyzed topics ($p < 0.001$). Negative z-values (Wilcoxon) and high absolute t-values (Student) confirmed that post-course knowledge levels significantly exceeded initial levels. The boxplots (Figure 6) reinforce these findings, showing elevated medians and reduced variability in knowledge levels. These results demonstrate the course's positive impact on students' perceived learning.

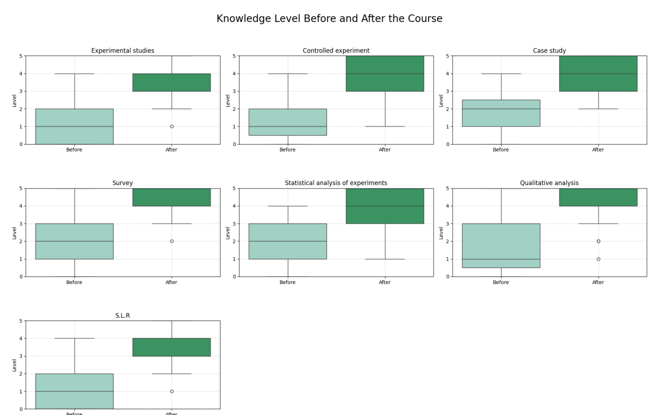

Figure 6: Knowledge level distribution before and after the course across seven topics.

4.3 Perception of Motivation

The qualitative results showed significant patterns in the aspects responsible for influencing student motivation. The analysis of the answers made it possible to identify elements organized by means of thematic codes. Initially mentioned above, **COD2.2 - Practice as Foundation** emerged as a key point recognized by the students, who highlighted the relevance of the knowledge acquired. This is exemplified by A25's account: *"Being able to see how the data collected was transformed into concrete conclusions through statistical analysis gave me a sense of achievement and showed the practical strength of this approach in the context of software engineering"*. Returning to an aspect already mentioned, **COD3.3 - Autonomy** again stood out, highlighting the importance of students' freedom to make decisions about their own study process. This is illustrated by A13's account: *"The opportunity to choose what we would research and how we could do it"*.

Another topic that was described again was **COD3.4 - Collaborative Work**, which defines interaction among students as essential for learning, as highlighted by A22's excerpt: *"The possibility of discussing ideas, solving doubts in groups and applying concepts in practice made learning more dynamic and interesting"*. In contrast, **COD4.1 - Interest in Data Collection Methods** emerged, reflecting the students' interest in exploring different approaches in their research. This motivation is clearly reflected in A21's statement: *"Knowing different ways to carry out the study"*. Another relevant point was **COD4.2 - Motivation by Recognition**, which highlights the positive impact of support and recognition from the teacher. This factor was evident in A08's statement: *"It was the presence of the teacher accompanying us at all times and always being very understanding and respectful. This motivated me not to let her down and to try my hardest in the subject"*. Finally, **COD4.3 – Motivation to Pass** was also mentioned, as evidenced directly by A11's comment: *"To pass the course"*.

In addition, **COD4.4 - Extracurricular Motivations** stood out, encompassing academic factors that influenced the participants' engagement, especially in relation to the preparation of the undergraduate thesis. This perspective is illustrated by A19's statement: *"Knowing that any form of knowledge acquired during the course could be applied to future projects (personal, academic, undergraduate thesis, etc)"*. The **COD4.5 - Assessment Method** is scored as another aspect valued by the participants, highlighting the preference for an approach centered on practical work rather than traditional assessments: A19: *"The fact that I knew that there was no weight of exams (and that the grade would basically be proportional to the practical work)"*. The **COD4.6 - Personal Motivation** reflects the individual enthusiasm of the students during the course. This experience is well illustrated by A25's statement: *"Carrying out my first experimental study was challenging and exciting at the same time"*.

Another point worth highlighting is the **COD4.7 - Dynamic Teaching Methodology**, which highlighted satisfaction with the proposed methodology, identified as a motivational factor within the learning environment. This can be seen in the following report: A20: *"The applied methodology of directed studies and the application of assignments rather than the use of tests. This made the subject more practical and motivating"*. The **COD4.8 - Interest in Statistical Analysis**, brought the students' satisfaction in understanding and applying statistical concepts: A25: *"Being able to see how the data collected was transformed into concrete conclusions through statistical analysis gave me a sense of achievement"*.

Furthermore, the **COD2.4 - Continuous Support** appears as an important factor in maintaining the engagement of the class: A03: *"The discussions after the reading between the students and the teacher. It helped maintain motivation and learning"*. The analysis also revealed individual motivations, such as: **COD4.9 - Prior Affinity**, which highlights the knowledge acquired before the course and also possible related interests: A17: *"It was previous motivation, I was always interested in how all kinds of research was done"*.

Participants indicate their degree of motivation for learning the subject topics. The stacked bars in Figure 7 show that the majority of participants declared themselves "Motivated" or "Very motivated" for all methods, especially Qualitative Analysis, Case Study and Survey. In SLR and Statistical Analysis of Experiments, although high motivation predominates, there is a slightly higher proportion of "Neutral" or "Unmotivated" responses.

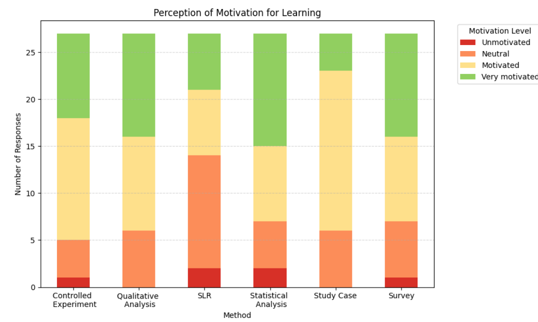

Figure 7: Distribution of student responses regarding their motivation to learn through different experimental methods used in the course.

To investigate the relationship between motivation for each assessment method and the perceived contribution of the course, the Spearman correlation was calculated between responses to the question **"Will this course contribute to your education?"** and the reported level of motivation for the course topics. Figure 8 illustrates the results of this analysis.

The correlations ranged from moderate to strong and were consistently positive: Statistical Analysis of Experiments (r = 0.55), Qualitative Analysis (r = 0.53), Systematic Literature Review (r = 0.32), Survey (r = 0.30) and Case Study (r = 0.26). Darker colours in the heatmap indicate a stronger association between the variables and the perceived contribution of the course.

4.4 Perception of the Value of Learning

The data collected through the qualitative approach revealed elements that influenced the appreciation of learning. The analysis of the course's contribution to students' education revealed several key points. **COD2.2 — Practice as Foundation** was once again highlighted by the participants: A24: *"The course enriched my education by combining theory and practice, strengthening analytical, and experimental skills essential for facing real challenges in the field."* Similarly,**COD3.3 — Autonomy** also emerged as a positive highlight, as reflected in comment A11: *"Plenty of autonomy for students to develop their own methods to solve the questions proposed."*

On the other hand, the academic impact was also notable, as highlighted by **COD5.1 – Contribution to Academic Career**, in the following reflection (A20): *"This becomes especially important for those considering an academic path, particularly at the master's level, where one engages directly with scientific research."* Another key theme was **COD5.2 – Communication**, which encompasses the development of expressive and information-sharing skills throughout the learning process, as illustrated in A04: *"It facilitated understanding and taught me how to conduct research, use various tools, and, most importantly, present findings effectively."*

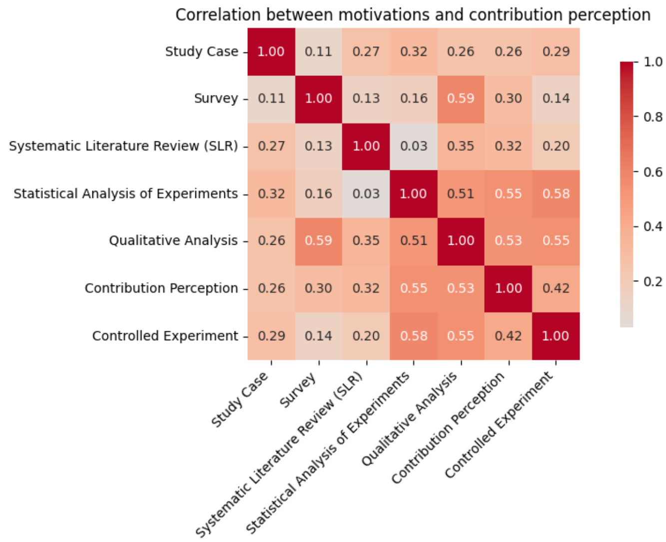

Figure 8: Spearman correlation matrix between motivations by study type and perceived contribution of the course.

The study also presented perspectives on the future contribution of the course: **COD5.3 - Future Application**, illustrated by student response A25: *"In the future, I will apply this knowledge in possible experimental studies that I will conduct"*. Another relevant aspect identified was the **COD5.4 - Familiarization with ESE**, which pertains to the process of introducing and familiarizing students with the concepts, methods, and practices of Empirical Software Engineering, as mentioned by A20: *"The course can contribute by helping people who, like me, have not had much contact with conducting experiments."*. The recognition of integrating knowledge from different areas was also highlighted as a relevant factor in the analysis through the **COD5.5 - Interdisciplinarity**. This is exemplified by A08's testimony: *"Although I am not from the Software Engineering field, I believe that this will greatly facilitate my understanding when I speak with a Software Engineer about the subject."*

Furthermore, the **COD5.6 - Interest in Course Concepts** was commented on by A15: *"What I found most interesting to learn was hypothesis testing and qualitative data analysis."*

Finally, in response to the last question of the questionnaire, which assessed the course methodology, students shared their perceptions, highlighting positive aspects, suggestions for improvement, and challenges faced. **COD6.1 - Overall Positive Evaluation of the Methodology** was mentioned in statements such as A08: *"I know how difficult it is to plan a methodology and try to follow it, and in the course all of that occurred in a very fluid and natural way."*. The **COD6.2 - Enriching and Challenging Methodology** reflects students' perception that, although the course presented significant challenges, it also provided valuable learning, for example: A18: *"It was a very challenging course for me, but I enjoyed it a lot."*

However, **COD6.3 - Confusing Methodology** was mentioned by a student as a difficulty encountered during the course, exemplified by A04: *"Particularly, the course was generally quite difficult; the methodology remained confusing."* Complementarily, the same participant suggested improvements through **COD6.4 - Suggestion for Improvement**: *"One suggestion would be to explicitly clarify the criteria and whether they have scoring."* Finally, **COD6.5 - Recognition of Dedication** was expressed in acknowledgements such as A08: *"My congratulations to the teaching assistants and the professor for their effort and dedication."*

The analysis indicates that, despite some specific criticisms, the course was, for the most part, well evaluated for the methodology applied. Emphasis was placed on its practical applicability and its contribution to the development of research skills, with special appreciation for the autonomy granted to students and the support provided by the teaching staff.

5 Lessons Learned

The implementation of an Active Learning approach had a significant impact on undergraduate students, primarily by requiring greater autonomy in their learning process. This methodology proved effective in stimulating direct student engagement and empowering them to take a more proactive role in knowledge construction. It was observed that certain non-technical skills were developed throughout the course, such as communication (both oral and written) and enhanced teamwork.

No codes describing problems or limitations of the course were identified in the Perception of Motivation analysis, only in the Perception of Learning and Perception of the Value of Learning analyses. Some reports highlighted remaing difficulties in understanding research steps (COD2.5), which may consequently lead to low self-confidence in conducting future studies (COD2.6). Additionally, there are mentions of limited proactivity in guiding their own learning process (COD3.8), a factor that may be associated with the challenges students reported in staying motivated throughout the course (COD3.9).

In response to these codes (and related ones such as COD3.7, COD3.10, and COD3.11), it is recommended to add dedicated Q&A class meetings to the course schedule, focusing on assignments and content. These sessions could be held in a conversational round-table format, providing a safe space for students to feel more confident, motivated, and aware of the importance of autonomy in the course. It is also worth noting that the teamwork activities proposed by the teacher may help reduce individual workload by fostering peer support.

Meanwhile, in the Perception of the Value of Learning analysis, codes COD6.3 and COD6.4 were extracted from a single student's report. These codes indicate confusion about the teaching methodology, including the assessment criteria for assignments, despite all activities having detailed instructions (available on Zenodo) and the tutoring team being available to clarify any questions.

From the perspective of the teaching assistants, their role was a key factor in the success of the Active Learning approach, especially in connecting theory and practice. Their proximity to the students allowed the team not only to conduct lab sessions with specific tools but also to identify recurring challenges, such as the learning curve in statistical analysis and the language barrier in the reading materials. A key lesson learned from their direct observation was the need for explicit communication regarding activity expectations, as they noted that some students had not initially understood all requirements for guided studies.

Additionally, the teacher's perspective reinforces that the results obtained were positive, as the methodology exposed students to practical ESE scenarios, which require skills fostered during the course, such as autonomy and teamwork. Although some students faced more difficulties, the experience proved to be enriching for academic development, contributing to the completion of final projects, scientific papers, and the preparation for a future academic career. In this context, considering the undergraduate level, the course requires the instructor to have extensive practical experience in empirical software engineering to guide the students and support their methodological decisions, particularly in Hands-on Work 1 and 2.

6 Discussion

The results obtained from both the opinion survey and the analysis of student performance highlight key factors that influence the effectiveness of active learning methodologies in Software Engineering education.

The analysis of **Perception of Motivation** indicates that practical development and increased autonomy motivated the students to participate in the presentations and complete the proposed tasks. This is reflected in the high participation rates: over 85% of students completed the directed studies, and 80% participated in the seminars. Although this study involved only undergraduate students, the motivational benefits observed align with the findings of Meireles et al. [17], which employed a mixed approach (undergraduate and postgraduate), and also reported a positive impact from the use of active learning strategies in an ESE course.

Regarding the **Perception of the Value of Learning**, the qualitative analysis highlighted the contribution to students' academic development and the relevance of the tasks performed. In other words, the students reported that the methodology helped them structure their research and prepare for their undergraduate thesis, emphasizing hands-on learning as a central component of the process. This perception is in line with the study by Vilela and Silva [29], who identified preparation for the final project (TCC) as one of the positive results of the adopted methodological approach.

In addition, the **Perception of Learning** also generated relevant contributions. As the results show, topics covered during the course provided a significant increase in the students' level of knowledge, as well as greater confidence and preparation to conduct scientific research after completing the course. In this context, Gren [10] points out that the flipped classroom methodology can increase student performance, strengthening the relationship between active teaching and better learning results. Despite the methodological differences, both studies address the potential of active teaching methods to promote student engagement and facilitate knowledge attainment.

7 Conclusions

The presented work reveals the effectiveness of the active teaching strategy applied to Empirical Software Engineering education for undergraduate students. The findings show a significant change in the students' profile, with the majority increasing their confidence and skills in carrying out scientific research after completing the subject. The approach adopted, which integrated theory and practice through specific tools and activities, presents an effective method to strengthen the skills required for empirical research.

The analysis of both qualitative and quantitative data indicates significant improvements in all evaluated aspects. In addition, the relationship between student motivation and perception of the course's contribution reinforces the importance of student-centered teaching approaches. The practical work allowed students to participate in all stages of an experimental study, from planning to interpreting the results.

Although this report has limitations such as the size of the sample and the context of the experiment, it presents findings that are consistent with the literature. The results indicate relevant contributions, suggesting that the adopted methodology is viable for implementing ESE instruction in higher education. Moreover, they demonstrate the applicability of active learning methodologies even in large classes.

In future work, it is intended to analyze data from additional classes, allowing for a meta-analysis of diverse experiences. In general, this work reinforces the importance of approaches that connect theory and practice, developing students' technical skills alongside critical thinking and evidence-based reasoning — key elements for advancing Software Engineering in academic and industrial contexts alike.

ARTIFACT AVAILABILITY

The instruments used in the study, along with the collected and analyzed data, are publicly available in the Zenodo repository: https://zenodo.org/records/15361186

ACKNOWLEDGMENTS

The authorship team acknowledges the financial support provided by the PID/UFC program and FUNCAP - PRH-0212-00011.01.00/23 process.

REFERENCES

[1] Victor R Basili. 1996. The role of experimentation in software engineering: past, current, and future. In *Proceedings of IEEE 18th International Conference on Software Engineering*. IEEE, 442–449.

[2] Victor R Basili, Forrest Shull, and Filippo Lanubile. 1999. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering* 25, 4 (1999), 456–473.

[3] John W Creswell. 2007. An introduction to mixed methods research. *Lincoln, Nebraska, USA: University of Nebraska* 23 (2007).

[4] Elias José de Rezende Freitas and Laís Sergiane Fortes. 2020. Aprendizagem ativa aplicada à Engenharia: um estudo sobre a percepção do aprendizado. *Revista de Ensino de Engenharia* 39 (2020).

[5] Sofia Bento Desidério, Maria Rebecca Lopes Lelis, Maria Elanne Rodrigues, and Anna Beatriz Marques. 2024. How ready for hci? a qualitative analysis of the practice of soft skills related to hci by women involved in the digital girls program partners projects. *Journal on Interactive Systems* 15, 1 (2024), 504–516.

[6] Michael Felderer and Marco Kuhrmann. 2019. Using Mini-Projects to Teach Empirical Software Engineering.. In *SEUH*. 75–86.

[7] Thaís Ferreira, Davi Viana, Juliana Fernandes, and Rodrigo Santos. 2018. Identifying emerging topics and difficulties in software engineering education in Brazil. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*. 230–239.

[8] Clara Lima Fonseca, Luciana AM Zaina, and Anna Beatriz Marques. 2024. Avaliação de Métodos para Elicitação e Especificação de Requisitos de Usabilidade com Histórias de Usuário: Um Experimento Controlado. In *Simpósio Brasileiro de Engenharia de Software (SBES)*. SBC, 257–268.

[9] Vahid Garousi, Gorkem Giray, Eray Tuzun, Cagatay Catal, and Michael Felderer. 2019. Closing the gap between software engineering education and industrial needs. *IEEE software* 37, 2 (2019), 68–77.

[10] Lucas Gren. 2020. A flipped classroom approach to teaching empirical software engineering. *IEEE Transactions on Education* 63, 3 (2020), 155–163.

[11] Marcela Hernández-de Menéndez, Antonio Vallejo Guevara, Juan Carlos Tudón Martínez, Diana Hernández Alcántara, and Ruben Morales-Menendez. 2019. Active learning in engineering education. A review of fundamentals, best practices and experiences. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 13 (2019), 909–922.

[12] Barbara Kitchenham. 2004. Procedures for performing systematic reviews. *Keele, UK, Keele University* 33, 2004 (2004), 1–26.

[13] Barbara A Kitchenham and Shari L Pfleeger. 2008. Personal opinion surveys. In *Guide to advanced empirical software engineering*. Springer, 63–92.

[14] Carlos Danilo Luz, Edson OliveiraJr, and Igor F Steinmacher. 2022. How Experimentation in Software Engineering has been taught? Survey and Research Agenda. In *Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering*. 299–304.

[15] Maíra R Marques, Alcides Quispe, and Sergio F Ochoa. 2014. A systematic mapping study on practical approaches to teaching software engineering. In *2014 IEEE Frontiers in education conference (FIE) proceedings*. IEEE, 1–8.

[16] Suéllen Martinelli, Joelma Choma, Jullia Saad, and Luciana Zaina. 2023. Análise Qualitativa em IHC: da codificação à criação de visualizações. In *Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC)*. SBC, 9–10.

[17] Maria Alcimar Costa Meireles, Sabrina Rocha, Jose Carlos Maldonado, and Tayana Conte. 2024. An experience report on the use of Active Learning in Empirical Software Engineering Education: Understanding the pros and cons from the student's perspective. In *Proceedings of the 46th International Conference on Software Engineering: Software Engineering Education and Training*. 380–390.

[18] Duncan D Nulty. 2008. The adequacy of response rates to online and paper surveys: what can be done? *Assessment & evaluation in higher education* 33, 3 (2008), 301–314.

[19] Victor R Basili Dieter Rombach, Kurt Schneider Barbara Kitchenham, and Dietmar Pfahl Richard W Selby. 2007. *Empirical Software Engineering Issues*. Springer.

[20] Johnny Saldaña. 2021. The coding manual for qualitative researchers. (2021).

[21] Cynthia Pinheiro Santiago, José Wally Mendonça Menezes, and Francisco José Alves de Aquino. 2023. Proposta e Avaliação de uma Metodologia de Aprendizagem Baseada em Projetos em Disciplinas de Engenharia de Software através de uma Sequência Didática. *Revista Brasileira de Informática na Educação* 31 (2023), 31–59.

[22] Adrian Santos, Omar Gómez, and Natalia Juristo. 2018. Analyzing families of experiments in SE: A systematic mapping study. *IEEE Transactions on Software Engineering* 46, 5 (2018), 566–583.

[23] Sindiany Santos, Maria Silva, Caroline Caputo, Elaine dos Santos, Anézia Barbosa, and Jaqson Santos. 2023. *Aprendizagem Ativa Experiências teórico-práticas no ensino superior*. doi:10.47247/SSCS/88471.87.6

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

[25] Forrest Shull, Janice Singer, and Dag IK Sjøberg. 2008. *Guide to advanced empirical software engineering*. Vol. 93. Springer.

[26] Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology* 15, 1 (1904), 72–101.

[27] Student. 1908. The probable error of a mean. *Biometrika* 6, 1 (1908), 1–25.

[28] Antonio Karlos Araújo Valença. 2023. Metodologias ativas no ensino de engenharia: uma revisão bibliométrica. *Revista Produção Online* 23, 2 (2023), 4982–4982.

[29] Jéssyka Vilela and Carla Silva. 2024. Empowering Undergraduates in Empirical Research Methods: an Experience Report. In *Simpósio Brasileiro de Engenharia de Software (SBES)*. SBC, 543–553.

[30] Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83.

[31] Claes Wohlin. 2007. Empirical software engineering: Teaching methods and conducting studies. In *Empirical Software Engineering Issues. Critical Assessment and Future Directions: International Workshop, Dagstuhl Castle, Germany, June 26-30, 2006. Revised Papers*. Springer, 135–142.

[32] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, Anders Wesslén, et al. 2012. *Experimentation in software engineering*. Vol. 236. Springer.

[33] He Zhang, Muhammad Ali Babar, and Paolo Tell. 2011. Identifying relevant studies in software engineering. *Information and Software Technology* 53, 6 (2011), 625–637.