Thirteen Years Teaching Software Engineering at a Brazilian Federal Institute: An Experience Report from a Night-Shift Information Systems Course

Crescencio Lima
Federal Institute of Bahia
Vitória da Conquista, Brasil
crescencio@gmail.com

Amanda Ferraz de O. Passos
Federal Institute of Bahia
Vitória da Conquista, Brasil
amandainf@gmail.com

Crijina Flores
Federal Institute of Bahia
Vitória da Conquista, Brasil
crijinaflores@ifba.edu.br

Djan Santos
Federal Institute of Bahia
Vitória da Conquista, Brasil
djan.santos@ifba.edu.br

Luis Paulo Carvalho
Federal Institute of Bahia
Vitória da Conquista, Brasil
luispscarvalho@gmail.com

Claudio R. S. de Oliveira
Federal Institute of Bahia
Vitória da Conquista, Brasil
claudio.ifbavc@gmail.com

Alexandro dos Santos Silva
Federal Institute of Bahia
Vitória da Conquista, Brasil
alexandrossilva@ifba.edu.br

José Alberto Diaz Amado
Federal Institute of Bahia
Vitória da Conquista, Brasil
jose_diaz@ifba.edu.br

Kenedy M. Geraldo dos Santos
Federal Institute of Bahia
Vitória da Conquista, Brasil
kenedymarconi@ifba.edu.br

Natalia Lima de Oliveira Santos
Federal University of Bahia
Vitória da Conquista, Brasil
natalia.santos@ufob.edu.br

Abstract

Teaching Software Engineering (SE) in a night-shift undergraduate course at a Brazilian public federal institution presents challenges that differ from daily courses in typical university settings. Working students, constrained study time, variable class sizes, and rapid technological change demand continuous pedagogical adaptation. This paper reports thirteen years of experience (2013–2026) teaching the SE discipline in the Information Systems course at the Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), encompassing 22 class offerings and 325 students. We describe four distinct pedagogical evolution phases: (i) Project-Based Learning with Scrum (2013–2015); (ii) consolidation and textbook-driven instruction (2016–2019); (iii) emergency remote teaching during the COVID-19 pandemic (2020–2021); and (iv) post-pandemic re-engagement with integrated Large Language Model (LLM) discussion (2022–2026). For each phase, we reflect on what worked, what failed, and why, grounding observations in the specific constraints of night-shift students who balance full-time employment with university study. Key findings include the limited transferability of sprint-based to low-availability students, the lasting value of a video library produced during the pandemic, the inadequacy of a strict flipped classroom model for working students, and the practical benefits of anchoring SE concepts in daily industry news about AI and LLMs.

Keywords

Software Engineering Education, Experience Report, Project-Based Learning, Remote Teaching, LLM Integration, Night-Shift Students

1 Introduction

Software Engineering (SE) education faces a set of challenges that complicate curriculum design and delivery. Among these are the rapid evolution of technologies, frameworks, programming languages, and development practices, which continuously shift the baseline of what constitutes relevant knowledge. This dynamism forces educators to make recurring trade-offs between stability and currency in course content. At the same time, SE education must balance theoretical rigor with practical applicability, a tension that is particularly acute in courses constrained by time where neither dimension can be fully developed in isolation [12, 15].

This tension is amplified in contexts where students are not full-time learners, working adults who attend night courses, and carry prior professional experience into the classroom. However, they arrive exhausted, time-constrained, and with limited bandwidth for out-of-class preparation [17]. These students are not edge cases in Brazil — they constitute a portion of enrollment in public federal institutes of education, science, and technology [8]. Challenges associated with Information Systems education in Brazil, particularly in evening programs, have also been discussed by de Oliveira Neves et al. [3], highlighting issues related to student workload, time availability, and the diversity of learners' professional backgrounds.

The *Instituto Federal de Educação, Ciência e Tecnologia da Bahia* (IFBA) was established as part of a national expansion policy for federal institutes initiated in 2008, to democratize access to higher education and technical training in regions historically underserved by universities [8, 11]. IFBA's Information Systems course, offered as a night program, attracts students who are predominantly employed during the day and who commute from different cities and neighborhoods to attend classes in the evening.

Within this context, the SE discipline occupies the sixth semester of the curriculum. It is a 60-hour course that covers foundational and advanced SE topics, including development processes, lifecycle models, requirements engineering, architecture, estimation, metrics, quality assurance, and reuse. Teaching this discipline requires not only command of a broad technical domain, but also pedagogical sensitivity to the constraints of the student population.

This paper reports on thirteen years of teaching this discipline, from 2013 to 2026, covering 22 class offerings and 325 enrolled students. During this period, four instructors taught the course, contributing different perspectives and pedagogical approaches. We present a reflective longitudinal account of how the course evolved, what approaches were adopted and abandoned, and what lessons emerged from more than a decade of engagement with this specific educational context.

The contribution of this report is threefold. First, it documents a longitudinal perspective on SE teaching in a Brazilian federal institution. Second, it reflects both successful and failed pedagogical strategies. Third, it offers actionable lessons for educators in similar night-shift, public-institution contexts.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 describes the institutional and course context. Section 4 traces the four phases of pedagogical evolution. Section 5 presents cross-cutting reflections on findings. Section 6 states lessons learned. Section 7 discusses threats to validity. Section 8 concludes the paper.

2 Related Work

Experience reports on SE education in Brazil have been a consistent presence at the Brazilian Symposium on Software Engineering (SBES) and its associated Education Track, formerly known as Forum on Software Engineering Education (FEES), since the early 2000s.

Andrade et al. [2] reflects on a decade of continuous improvement in teaching SE. The authors outline a teaching-learning process that combines theoretical classes with a practical software development project. Students are organized into teams that simulate "software development companies", taking on specific professional roles such as developer, tester, or manager.

Gama [4] discusses the implementation of active learning using LEGO bricks to engage millennial students. The author details two main activities: one focused on Requirements Engineering, and another focused on Scrum and project management.

Olivindo et al. [9] addresses the challenges of Emergency Remote Teaching (ERT) during the COVID-19 pandemic by combining the Flipped Classroom (FC) method with Gamification. The instructors created a fictional narrative where students act as engineers traveling the world and completing SE challenges to earn points and badges.

Also set during the ERT period, Alves et al. [1] investigates how students perceived the adoption of the FC across three different SE-related disciplines. Students praised the autonomy, interactivity, and dynamic discussions the format enabled. However, the qualitative feedback highlighted specific challenges, such as feeling overwhelmed by the large amount of preparatory content and difficulties with materials entirely in English.

Garcés and Oliveira [5] documents a four-year evolution of employing Project-Based Learning (PjBL) in SE courses. The course design required students to form autonomous teams to develop a minimum viable software product (MVP) utilizing the Scrum framework. The authors track the iterations of the course, sharing lessons learned over the years.

However, longitudinal reports spanning more than a decade at a federal institute remain rare. This paper contributes a longer view, making it possible to observe how student profiles, available technologies, and institutional conditions interact over time. Table 1 presents the diversity of teaching methodologies, student profiles, educational contexts, and technological tools employed.

Our report differs from the works above along three axes. First, in *temporal scope*: whereas the studies in Table 1 report single offerings or refinements of one methodology over a few iterations, we trace thirteen years and four distinct, sometimes regressive, pedagogical phases, allowing us to observe how the *same* methods (PjBL, Scrum, FC) succeed in some cohorts and fail in others as conditions change. Second, in *outcome polarity*: while these reports predominantly document successful adoptions of active-learning methods, we report that PjBL/Scrum and the FC — celebrated as engaging solutions elsewhere — broke down in our setting, and we analyze why. Third, in *student context*: the cited works largely address full-time day students, whereas our findings are shaped by the night-shift working-student profile, whose scarce out-of-class time directly conditions which methods are viable. We also uniquely document the systematic integration of LLM-related industry discussion into an SE course, a dimension absent from the prior reports, which center on tools such as LEGO bricks, gamified narratives, or software factories. It captures approaches ranging from PjBL and experiential learning to FC and active learning strategies, applied across in-person, remote, and hybrid settings. By consolidating these studies, we aim to provide an overview of how different pedagogical designs are executed in practice, as well as the variety of tools and environments that support them. This synthesis serves as a basis for situating our work within the broader SE education landscape and for identifying recurring patterns and gaps that motivate the approach adopted in this study.

3 Background and Context

This section provides the institutional, curricular, and student-level context in which the reported teaching experience occurred. We describe (i) the institutional setting and the structure of the course, (ii) the syllabus and its evolution in response to pedagogical and technological pressures, and (iii) the profile of the student population, including longitudinal enrollment data.

3.1 The Institution and Course

IFBA is a federal institution of higher, technical, and vocational education with multiple campuses across Bahia. The campus where this experience took place offers a four-year Bachelor of Information Systems degree, structured as a night program. The night-shift format is central to understanding the student profile: most enrolled

Table 1: Software Engineering Education Experience Reports in Brazil

Source	Teaching Method.	Target Student Profile	Educational Context	Technological Tools Used
[2]	PjBL	Computer Science (CS) undergraduates	In-person classroom	WhatsApp, Facebook, SIGAA, Trello, Android, GitHub, Google Play Store
[4]	Active Learning	Computer Engineering undergraduates	In-person classroom	LEGO bricks, sticky notes (Post-its), slide presentations
[9]	FC and Gamification	Computer Technician and CS undergraduates	ERT	Google Forms, Google Classroom, Google Meet, YouTube, podcasts
[1]	FC	CS and Information Systems (IS) undergraduates	ERT	Google Meet, podcasts, YouTube, digital textbooks
[5]	PjBL and Scrum	Computer Engineering and IS undergraduates	Mixed (online and in-person)	Moodle, SIGAA, Slack, Discord, Notion, GitHub, JUnit, Selenium
This work	PjBL, Scrum, and Textbook-driven	Night-shift IS undergraduates	Mixed (in-person and ERT)	Video library, digital textbooks, remote teaching platforms

students hold full-time or part-time jobs during the day, and the commute from workplace to campus constitutes a daily logistical burden.

The SE course analyzed in this work has been shaped by three successive versions of the Course Pedagogical Project (PPC), reflecting broader institutional, regulatory, and disciplinary changes over time: PPC-1 (2010.2–2017.1), PPC-2 (2017.2–2023.2), and PPC-3 (2024.1–present). These curricular revisions provide an important lens through which to understand how SE education evolves in response to both internal and external pressures.

At a structural level, each PPC revision introduced adjustments to curriculum organization, course sequencing, and competency expectations, while preserving the overall program goal of training professionals capable of addressing technological and societal challenges in Information Systems. The program emphasizes not only technical proficiency but also humanistic, ethical, and organizational dimensions of computing practice. This dual orientation reinforces the persistent tension between theory and practice, as well as between technical depth and broader professional formation.

The SE discipline is positioned in the sixth semester, by which point students have completed foundational courses in programming, data structures, databases, and object-oriented design. The 60-hour workload is distributed across a semester of approximately 20 weeks, with classes meeting one time per week in sessions of 200 minutes.

Over the thirteen years analyzed, the discipline was taught by four different instructors. While teaching assignments varied by semester according to departmental allocation, there was substantial continuity in pedagogical orientation, course materials (notably the shared video library introduced during the pandemic), and assessment philosophy, as the instructors coordinated their approaches and reused common artifacts. We make this multi-instructor setting explicit here because instructor changes are a relevant contextual factor when interpreting variations in teaching strategy and student outcomes across offerings.

3.2 Discipline Syllabus

The official course syllabus covers: introduction and objectives of software engineering; software engineering activities; the software development process and software product; system lifecycle models and paradigms; use of models, methodologies, techniques, and tools for systems analysis and design; project management; software estimation; software metrics; software quality; and reuse, reverse engineering, and reengineering.

This breadth — spanning process, management, quality, and technical practice in 60 hours — requires constant prioritization. No single offering can cover every topic with equal depth, and this syllabus breadth has driven many of the pedagogical decisions described in this report.

3.3 Student Profile and Enrollment Data

Table 2 presents enrollment data across all 22 offerings from 2013 to 2026. In total, 325 students enrolled, of whom 62 (19%) were women. In addition, 3 students were identified as having specific educational needs and received institutional support according to inclusion policies. The class sizes ranged from 3 to 33 students, reflecting the variable intake of a small campus program.

The low proportion of women (19%) mirrors the broader gender imbalance in computing programs at IFBA and in Brazilian higher education [14]. The institution has participated in gender-inclusion initiatives, including the *Meninas Digitais do Sudoeste da Bahia* program, supported by the Brazilian Computer Society (SBC) [13], which aims to attract and retain women in computing through scholarships and extracurricular activities. These efforts appear to correlate with modest increases in female enrollment in recent years: the 2026 first-semester cohort reached 11 women (33%), the highest count in the discipline's history.

The ERT column indicates the four offerings (2020–2021) conducted under the Emergency Non-Presential Teaching Activities framework, which governed remote instruction during the COVID-19 pandemic at federal institutes nationwide.

4 Teaching Approach: Four Evolution Phases

Figure 1 summarizes the teaching approach evolution adopted in the SE course over thirteen years (2013–2026), organized into four distinct phases. Rather than following a linear or pre-planned instructional design, the course evolved iteratively in response to a combination of pedagogical challenges, institutional constraints, and external disruptions. These include the need to balance theory and practice, accommodate diverse student profiles, respond to

Table 2: Discipline offerings, enrollment, gender, students with specific needs, and remote-teaching status (2013–2026)

Year	Sem.	Students	Women	S. Needs	ERT
2013	1st	4	0	0	—
2014	1st	8	1	0	—
2015	1st	19	7	0	—
2015	2nd	11	1	0	—
2016	2nd	18	4	0	—
2017	2nd	9	0	0	—
2018	2nd	8	0	0	—
2019	1st	5	0	0	—
2019	2nd	3	1	0	—
2020	1st	12	1	0	Yes
2020	2nd	16	1	0	Yes
2021	1st	6	0	0	Yes
2021	2nd	15	1	1	Yes
2022	1st	4	1	0	—
2022	2nd	23	6	0	—
2023	1st	15	3	0	—
2023	2nd	13	3	0	—
2024	1st	20	4	0	—
2024	2nd	32	5	0	—
2025	1st	22	5	0	—
2025	2nd	29	7	1	—
2026	1st	33	11	1	—
Total	**22**	**325**	**62**	**3**	**4**

shifting industry expectations, and adapt to resource limitations and technological change.

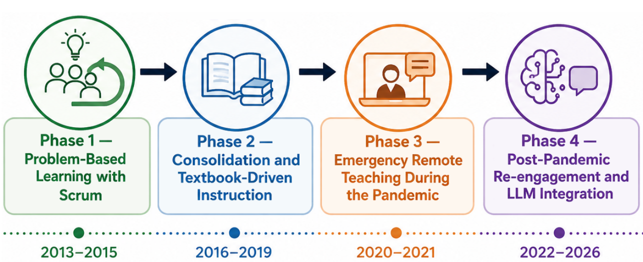

Figure 1: Four Evolution Phases

Each phase reflects a dominant instructional strategy shaped by the conditions of its time. Phase 1 emphasizes active, problem-based learning structured around Scrum practices, prioritizing collaboration and experiential learning. Phase 2 marks a shift toward consolidation, with a stronger reliance on textbooks and more structured content delivery to ensure coverage of foundational concepts. Phase 3 captures the abrupt transition to emergency remote teaching during the COVID-19 pandemic, requiring rapid adaptation of materials, assessment, and engagement strategies. Phase 4 represents a period of post-pandemic re-engagement, characterized by the reintegration of active learning practices alongside the incorporation of emerging tools such as Large Language Models (LLMs) into teaching and assessment.

4.1 Phase 1 — Problem-Based Learning with Scrum (2013–2015)

The first three years of the course were shaped by a commitment to active, project-based pedagogy. Inspired by Problem-Based Learning (PBL) principles [7] and the growing adoption of Scrum in Brazilian software companies, the course was structured around nine sprints (intervals of 15 days). Each sprint was mapped to one or two syllabus topics and required student teams to produce a concrete artifact: a requirements document, an architectural design, a working code module, a test plan, or a technical paper summarizing the sprint's learning.

Students were presented with a problem at the start of the semester — typically a simplified but realistic software system to be designed and partially built. Each sprint concluded with a presentation in which teams demonstrated their artifacts and received feedback. Assessment was based entirely on artifact quality, team effort, and individual dedication.

This model produced notable outcomes. Students from the 2013 and 2015 classes earned the best paper award at the Regional School of Computing of Bahia, Sergipe, and Alagoas (ERBASE), a competitive undergraduate research event. The sprint rhythm gave students a concrete sense of progress and made abstract SE concepts tangible: requirements engineering, for instance, became meaningful when students had to actually write specifications for their own project.

The model was discontinued after 2015 for a reason that reflects a structural feature of the night-shift context: *students routinely deprioritized sprint work in favour of deliverables from other, disciplines.* By the end of each sprint, many teams had produced nothing, and the final days before a sprint review became frantic catch-up sessions. The sprint cadence, designed for teams with dedicated daily capacity, assumed a level of out-of-class availability that working students simply did not have. Rather than scaffolding learning, the sprint model generated anxiety and last-minute production that undermined depth.

4.2 Phase 2 — Consolidation and Textbook-Driven Instruction (2016–2019)

From 2016 onward, the course shifted to a more conventional structure: lectures, textbook-based content delivery, and examination-based assessment. This shift was pragmatic rather than ideal: it reduced the coordination overhead of team projects and aligned better with students' available time.

This period was also one of textbook evolution. Early offerings relied heavily on Pressman's *Software Engineering: A Practitioner's Approach* [12] and Sommerville's *Software Engineering* [15] — comprehensive, authoritative references that nonetheless can feel disconnected from contemporary Brazilian software practice. Over time, the course progressively incorporated Valente's *Engenharia de Software Moderna* [16], an open-access digital textbook continuously updated to reflect current tools and methodologies, written with the Brazilian undergraduate context in mind. The adoption of Winters et al.'s *Software Engineering at Google* [18] added a different dimension: students could read how a major technology company structures its engineering practices, making documentation, code review, and testing feel like real professional expectations rather than academic abstractions.

Class sizes during this phase were small (between 3 and 18 students), which allowed for discussion-based sessions and individual attention. The examination format, however, produced a notable risk: without regular formative checkpoints, some students disengaged between exams and recovered poorly.

4.3 Phase 3 — Emergency Remote Teaching During the Pandemic (2020–2021)

The COVID-19 pandemic interrupted in-person classes in March 2020. IFBA's response was the ERT framework, which authorized emergency remote teaching through synchronous and asynchronous digital channels. Participation in ERT was optional for students: those who could not follow remote activities retained the right to resume in-person classes at the same point in the curriculum when campus activities resumed [6].

This optionality created heterogeneous groups: some students engaged fully with remote activities, others partially, and some not at all. The framework also stipulated that students who chose to participate in ERT but did not achieve passing grades would not receive failing records — reducing some of the high-stakes pressure but also reducing accountability.

Four class offerings (2020.1, 2020.2, 2021.1, and 2021.2) were conducted under ERT. The primary pedagogical response was the production of a video library: over the two pandemic years, 25 instructional videos were recorded and published on a public video-sharing platform, covering the core syllabus topics. The library was later extended with six additional videos in 2025–2026, reaching 30 videos in total Synchronous sessions were conducted via videoconference with a compressed format of 50 minutes, down from the standard class length, reflecting both platform fatigue and connectivity constraints.

Assessment shifted to exercise-based formats, prioritizing smaller, more frequent tasks over large exams. This reduced the risk of single-point failure for students experiencing irregular connectivity or domestic disruptions. Flexibility was extended on submission deadlines and exam scheduling.

The pandemic phase surfaced a digital divide that had been latent: students with limited internet access, inadequate hardware, or shared computing environments were severely disadvantaged. IFBA's *Edital 06/2020* for emergency digital inclusion support provided subsidies for internet access and equipment acquisition for economically vulnerable students [6], partially mitigating this barrier.

4.4 Phase 4 — Post-Pandemic Re-engagement and LLM Integration (2022–2026)

The return to in-person classes in 2022 did not restore the pre-pandemic educational dynamic. Students who had spent four semesters studying from home were reluctant to resume commuting to campus. Many had structured their daily routines around remote learning; returning to the classroom meant losing time they had come to rely on. This motivational dip manifested as attendance irregularity and reduced engagement, a pattern observed across multiple disciplines at IFBA.

An initial attempt to leverage the pandemic video library as the foundation for an FC model — wherein students watch recorded content before class and use class time for problem-solving — revealed a fundamental incompatibility with the working-student profile. Analysis of video access logs showed that viewing activity spiked in the 24 to 48 hours before examinations, not in the days before each class. Students were using the videos as exam preparation material, not as pre-class preparation. For students who arrive at campus directly from an eight-hour workday, the cognitive and logistical cost of watching instructional videos on weekday evenings is simply too high. The FC model, while theoretically well-suited to active learning [10], presupposes discretionary time that working students do not consistently have.

The course adapted: videos became optional support material for self-paced review, and class time returned to direct instruction enriched with discussion. The key innovation of this phase was the systematic integration of LLM discussion across every class session. Beginning with the public release of ChatGPT in late 2022, the instructor introduced a recurring practice: opening each class with a brief discussion of recent industry news relating to how companies are adopting AI and LLMs in software development. Topics ranged from AI-assisted code review and automated testing to LLM-generated documentation and the implications of generative AI for software estimation and quality assurance.

Based on the instructors' observations, this practice appeared to serve several pedagogical benefits. It connected abstract syllabus topics to observable industry trends, increasing the perceived relevance of the discipline. It also encouraged students to critically reflect on the impact of emerging technologies on their future professional practice while fostering the habit of continuously monitoring technological advances in a rapidly evolving field. However, we note that these are instructor observations rather than measured effects; no surveys or controlled comparisons were conducted, and a systematic evaluation of LLM integration remains an important direction for future research. From 2023 onward, LLM-related questions were incorporated into examinations, reinforcing their role as an integral component of the course rather than an optional or supplementary topic.

Moreover, assessment from 2023 onward returned to examinations, which stabilized outcomes: no dropouts were recorded across the 2023–2026 phase, and all students passed with satisfactory grades. This contrasted with the 2022 second semester, when a return to artifact-based assessment resulted in every student being directed to a supplementary final exam — a near-complete failure of the assessment strategy. Post-hoc reflection attributed this to the combination of post-pandemic demotivation, larger class size (23 students), and insufficient scaffolding for team project work after a prolonged period of individual, exercise-based learning.

5 Reflections and Findings

This section presents reflections and findings from these years of iterative course design and delivery, building on the previously described phased evolution. Rather than presenting results in an evaluative or outcome-driven manner, we adopt a reflective and practice-informed perspective to analyze how different pedagogical choices performed under real-world constraints. The goal is not to identify universally optimal solutions, but to surface context-sensitive insights that can inform SE education in similar settings.

5.1 What Worked

The PjBL/Scrum phase demonstrated that when students produce real artifacts — even imperfect ones — their understanding of SE processes deepens qualitatively. The ERBASE best paper awards from 2013 and 2015 groups are evidence that working-student populations can achieve high scholarly output when adequately scaffolded.

The 25 videos produced during 2020–2021 have continued to serve the course as an enduring instructional resource. Their value was not anticipated during production; they were created as emergency measures. In retrospect, they constitute the most durable pedagogical artifact of the thirteen years described here.

Because the videos were published on a public platform rather than a closed course environment, their audience grew well beyond the enrolled cohorts. Table 3 summarizes the library's reach. As of early 2026, the 31 videos comprised approximately 24.9 hours of content and had accumulated 18,844 views and 3,716.7 watch-hours — the equivalent of more than 150 continuous days of viewing. The 25 pandemic-era videos (2020–2021) account for 10,920 views, while six videos published in 2025–2026, including a seven-hour "complete course" compilation, already account for 7,924 views, indicating that the resource continued to attract a broad audience long after the emergency that prompted its creation. The most recent addition, published in January 2026, discusses the benefits of software documentation drawing on industry practice [18], illustrating how the library is still being actively extended to connect classic SE topics with contemporary professional sources. This dual role — structured support material for enrolled night-shift students who could review content on their own schedule, and an open educational resource for the wider community of learners and practitioners — illustrates how an artifact built under emergency constraints became a lasting contribution to both the course and informal SE learning beyond it. We note that view and watch-time figures reflect a general public audience and cannot be attributed solely to enrolled students.

Table 3: Reach of the Software Engineering video library (as of early 2026)

Period	Videos	Views	Watch-hours
2020 (pandemic)	12	4,519	616.5
2021 (pandemic)	13	6,401	1,017.3
2025–2026 (extension)	6	7,924	2,082.9
Total	**31**	**18,844**	**3,716.7**

Grounding SE concepts in daily news about AI adoption by software companies appeared, in the instructors' assessment, to be effective in engaging students, although this was not measured through formal instruments. Discussion of a concrete industry development — a company deploying LLMs for code review, for instance — provides an entry point that connects to students' professional aspirations and makes abstract topics such as software quality or process automation immediately concrete.

The progressive replacement of classic international textbooks with Valente's continuously updated open-access work [16] and the practitioner account in Winters et al. [18] reduced the perceived distance between academic content and professional reality. Students responded positively to reading about engineering practices at Google as a counterpoint to lifecycle models described in 2010s textbooks.

5.2 What Did Not Work

The nine-sprint model assumed student availability that the night-shift context did not provide. The model would likely succeed in a day-course context with lighter parallel workloads. For working students, it generated procrastination, last-minute production, and inter-team inequity.

The literature on FC emphasizes the importance of students completing pre-class preparation [10]. Our experience confirms this dependency and illustrates what happens when the assumption fails: students disengage from pre-class activities, class time discussions lose their foundation, and the model degrades into conventional instruction without the efficiencies either format would offer independently.

Reintroducing artifact-based assessment after four semesters of exercise-based remote learning, without rebuilding students' project skills and teamwork habits, was the single most consequential pedagogical error of the thirteen years. The result — 100% of students directed to remedial examination — was clear evidence that assessment modality cannot be changed abruptly without corresponding changes to instructional preparation.

5.3 The Night-Shift Constraint

A theme running through all four phases is the structural reality of night-shift students in a public institution. These students are not simply students who happen to have classes in the evening: they are professionals who also happen to be students. Their primary identity and primary time commitment during the week is employment. This inverts the typical university assumption that study is the primary activity and work (if present) is secondary.

This inversion has direct implications for pedagogy. Out-of-class time is scarce and unpredictable. Group coordination outside class hours is difficult. Cognitive load at the start of an evening class, after a full workday, is a real constraint. Teaching strategies that assume generous out-of-class availability — sprints, flipped preparation, multi-week team projects — must be designed with this constraint in mind.

These students bring professional experience that enriches classroom discussion. When the instructor raises a real-world SE problem — a company's migration to microservices, a public software failure, an LLM replacing a testing role — students often connect the discussion to their own workplace experience. This is a pedagogical asset that enriches the discussion of topics.

5.4 Gender Inclusion

The aggregate enrollment data (19% women across 13 years) reflects a persistent gender gap in computing education in Brazil [14]. The *Meninas Digitais do Sudoeste da Bahia* initiative [13], supported by SBC, has contributed to institutional efforts to recruit and retain women in computing programs through scholarships, mentoring, and visibility events. The 2026 first-semester class, with 11 women

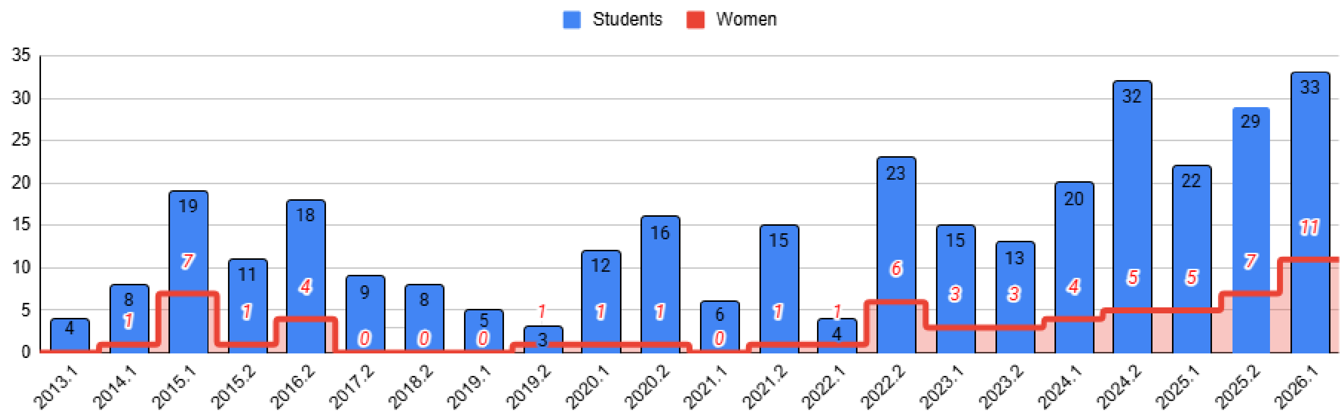

Figure 2: Students Enrolled and Women in Software Engineering Course (2013.1 – 2026.1)

out of 33 students (33%), represents a promising trend. The SE discipline does not operate gender-inclusion initiatives independently, but benefits from campus-level programs that increase the overall proportion of women in the Information Systems course.

Figure 2 presents the evolution of student enrollment and female participation in the SE course. Overall enrollment exhibits a gradual upward trend, particularly from 2022 onward, culminating in the highest intake of 33 students in 2026.1. In contrast, the participation of women shows greater variability over time, especially in earlier semesters. From 2020 onward, however, female participation becomes more consistent and demonstrates a modest but clear increase, reaching a peak of 11 women in 2026.1.

Despite this improvement, the data highlight a persistent gender imbalance, with women representing a minority in most offerings. Notably, periods of increased enrollment do not always correspond proportionally to higher female participation, suggesting that growth alone does not guarantee improved diversity. Nevertheless, the upward trend observed in recent years indicates progress, reflecting broader institutional inclusion policies and efforts to create a more accessible and supportive learning environment.

The PPCs' evolution, together with the strengthening of institutional policies, particularly those aimed at inclusion, student retention, and the democratization of access, may have contributed to changes in the student profile over time. Successive curricular revisions tended to make the program more aligned with contemporary demands of the field, more flexible in its structure, and more responsive to diverse academic and professional trajectories.

At the same time, the implementation of institutional policies for inclusion and student support may have fostered not only an increase in the total number of students but also greater participation from historically underrepresented groups, such as women and students with specific needs. Although it is not possible to establish a direct causal relationship within the scope of this work, the convergence of curricular updates and inclusion policies suggests a progressively more accessible and attractive environment, with potential positive impacts on diversity and student body growth.

We stress that gender did not emerge from a designed pedagogical intervention within the SE discipline itself: the discipline ran no gender-specific intervention, and the patterns reported here are observed enrollment trends situated within campus- and institution-level inclusion programs. We include this analysis for two reasons connected to the paper's goals. First, the persistent under-representation of women (19% across the period) is a defining feature of the student context that any educator inheriting this course must understand, and it directly informs Lesson 1 (adapt to the real student profile rather than an idealized one). Second, documenting the modest recent improvement alongside institutional inclusion efforts contributes a data point to the broader community discussion on diversity in Brazilian computing education, even though our report cannot attribute the trend to any specific cause.

5.5 Analysis of Students' Final Grades

Figure 3 presents the distribution of students' final grades across 21 course offerings from 2013.1 to 2025.2. The results indicate high median grades, ranging between 7 and 9, suggesting that most students achieve satisfactory performance. However, the boxplots also reveal substantial variation across semesters, both in terms of dispersion and the presence of outliers.

In the earlier offerings (2013–2015), grade distributions are compact, with limited variability and few extreme values. From 2016 to 2019, variability increases, indicating growing heterogeneity in student performance. The semester 2019.2 stands out as an outlier, with a notably lower median and a broader spread, because of the small number of students.

During the pandemic period (2020–2021), the distributions became more concentrated at higher grade ranges, reflecting the adaptation to ERT and changes in assessment practices. In the post-pandemic period (2022 onward), variability increases again, with the reappearance of low outliers, indicating uneven recovery in student performance. Despite this, recent offerings show signs of stabilization, with medians returning to the 7–8 range.

We caution against attributing this variation to teaching approach alone. Several non-exclusive factors plausibly contributed and cannot be disentangled within an observational report. First, the four instructors differed in style and assessment design, so part of the cross-semester variation likely reflects instructor changes rather than methodological ones. Second, the pandemic itself — and the assessment adaptations it forced (exercise-based grading, flexible

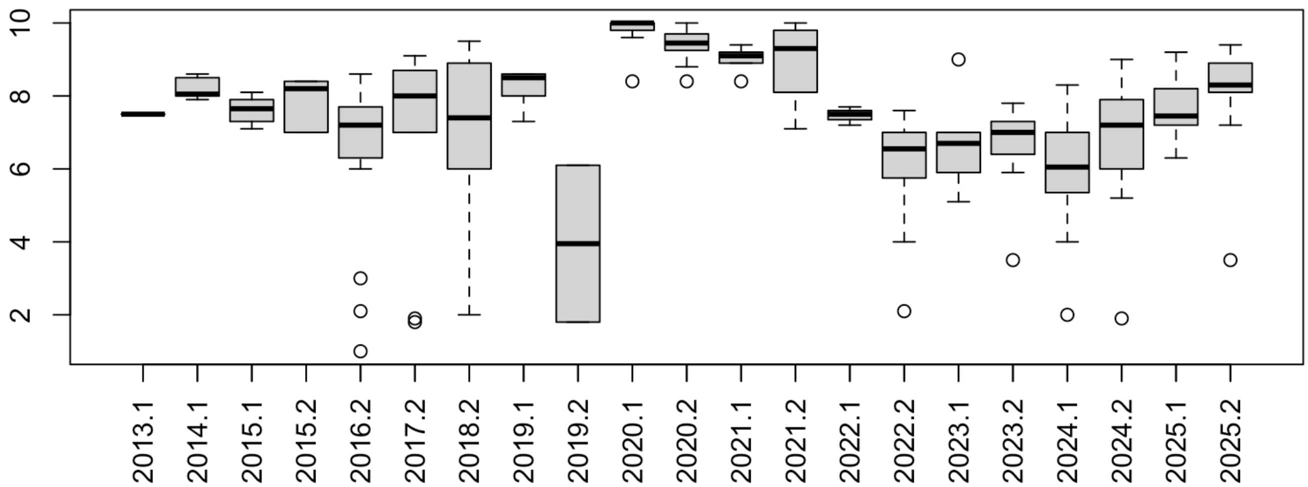

Figure 3: Boxplots with the Distribution of students' final grades across course offerings from 2013.1 to 2025.2.

deadlines, the optional ERT regime under which failing grades were not recorded) — compressed distributions toward higher ranges during 2020–2021, independent of pedagogy. Third, class sizes varied widely (from 3 to 33 students), and small cohorts produce unstable distributions and pronounced outliers, as in 2019.2. Fourth, cohort intake characteristics shifted over time alongside successive curricular revisions (PPC-1 to PPC-3) and broadening inclusion policies. We therefore read the grade trends as a longitudinal backdrop for interpreting the four phases — for instance, the 2022 assessment failure and the pandemic-era concentration of high grades — rather than as evidence that any single teaching choice caused a given outcome.

6 Lessons Learned

This section organizes the insights into five actionable lessons learned. Rather than reiterating specific results, we abstract recurring patterns, trade-offs, and decision rationales across multiple course iterations and changing conditions. These lessons are grounded in practice and shaped by context constraints, including limited instructional time, heterogeneous student backgrounds, evolving technologies, and the demands of a night program.

Lesson 1: Adapt methods to the real student profile, not the ideal one. The most consequential pedagogical decisions in this experience report were driven by a mismatch between an instructional model designed for full-time students and a cohort of working adults. Any SE educator preparing a course for a night-shift population should begin by explicitly acknowledging the time budget and energy levels students bring to the classroom, and design accordingly.

Lesson 2: Build reusable artifacts. The video library produced during the pandemic was not planned as a strategic resource. It became one. Educators should consider building instructional materials — videos, problem repositories, case studies grounded in local industry — that accumulate value over multiple offerings. These artifacts reduce preparation burden over time and provide students with on-demand support. Publishing such materials openly, as we did with the video library, multiplies their value: the same effort that supports one cohort can serve future cohorts and the broader community of learners, turning an emergency response into a lasting open educational resource.

Lesson 3: Anchor SE content in contemporary industry practice. Abstract concepts such as software metrics, quality models, or estimation techniques engage students far more readily when connected to observable industry events. The practice of opening each class with a news-driven LLM discussion exemplifies a broader principle: the instructor's job is partly to serve as a translator between the academic formulation of SE knowledge and the forms in which that knowledge appears in students' professional futures.

Lesson 4: Participate in the SE education community. Attending, submitting, and reviewing papers to venues such as SBES and the International Conference on Software Engineering (ICSE) exposes educators to the current state of the art in SE teaching and creates connections with peers facing similar challenges. This experience report itself emerged from the recognition that the longitudinal perspective accumulated here has value for the community — a recognition that only became clear through engagement with the SBES education track.

Lesson 5: Treat failure as reportable data. The 2022 assessment failure and the discontinuation of the sprint model are included in this report not as embarrassments but as information. The SE education community benefits from honest accounts of what did not work and why. Experience reports that present only successes are less replicable and less useful than those that document the full range of outcomes.

7 Threats to Validity

Several limitations should be considered when interpreting this report. First, all reflections are from four instructors at a single institution. The observations are inherently shaped by their perspective and may not generalize to other campuses or institutions.

Second, this report does not employ a systematic research methodology: there are no control groups, no validated instruments, and no formal hypothesis tests. Claims about what worked or did not work rest on reflective observation rather than experimental evidence.

Third, the absence of formal student feedback data (surveys, structured interviews) means that student perspectives on the pedagogical changes described here are largely absent; the report reflects the instructors' interpretation of observable outcomes such as grades, attendance, video-access logs, and submitted deliverables. We did not collect questionnaires or course-evaluation instruments over the period, and we deliberately phrase our claims — particularly those regarding student engagement with LLM discussion — as instructor observations rather than validated effects. Consistent with the Experience Report category, which does not require rigorous evaluation but does require honest reflection on what worked or did not, we treat the lessons learned as advice grounded in practice rather than as empirically confirmed findings.

Fourth, changes in student outcomes across phases may be attributable to factors other than pedagogical choices — changes in student intake quality, broader economic conditions, or institutional policies, for example.

8 Conclusion

This paper has reported thirteen years of teaching SE in a night-shift Information Systems course at a Brazilian federal institute. Across 22 offerings and 325 students, the course traversed four distinct phases: PjBL with Scrum, examination-based consolidation, emergency remote teaching during the pandemic, and post-pandemic re-engagement with integrated LLM discussion. Each phase produced both successes and failures.

The central finding of this longitudinal account is that SE teaching in night-shift, working-student contexts requires methods and expectations calibrated to the real conditions of that population — not the idealized conditions assumed by models designed for full-time, day-course students. The sprint model, the FC, and intensive artifact assessment all encountered the same constraint: students who arrive at 7 pm, having worked since 8 am, have limited reserves of time and energy for out-of-class preparation and coordination.

At the same time, this context offers pedagogical assets: students with professional experience, strong motivation to connect academic content to career relevance, and a practical orientation that makes industry-grounded teaching effective. The integration of LLM discussion into every session, grounded in daily industry news, exemplifies an approach that leverages these assets.

We hope this report contributes to an evidence base for SE education in Latin America, particularly for the federal institute sector, which educates large numbers of working students but remains underrepresented in the SE education research literature. Future work includes a formal study of LLM integration outcomes through student surveys, a longitudinal analysis of pass rates and grade distributions, and continued monitoring of gender inclusion trends.

Artifact Availability

Enrollment and demographic data presented in Table 2 are derived from institutional academic records. Aggregated data supporting this report are available from the corresponding author upon reasonable request, subject to institutional privacy regulations. The Software Engineering video library is publicly available as an open educational resource at the following playlist: https://www.youtube.com/playlist?list=PLcilqwq_HuxUH4-Xttqm9ht_JLtAxnzQs.

Ethics Declaration

This paper reports on teaching practice and uses aggregate, anonymized enrollment data. No individual student data are reported. No ethics committee approval was required under the applicable institutional regulations.

AI Usage Disclosure

Claude was used to assist in structuring and organizing this manuscript. ChatGPT was used to support manuscript revision and refinement, as well as to generate Figure 1 first version. NotebookLM was used to assist in structuring related work and Table 1 first version. All factual content, data, analyses, reflections, and lessons learned were provided and validated by the authors. The authors take full responsibility for the accuracy, integrity, and originality of the content.

Acknowledgements

Our deepest thanks go to the students who, across the offerings over thirteen years, took part in the Software Engineering discipline reported here. Many of them attended class after a full day of work, commuting in the evening at the cost of time with their families and rest, and still brought curiosity, professional experience, and good humor to the classroom. Their projects, questions, struggles, and candid feedback shaped every pedagogical decision described in this paper; the lessons learned reported here are, in a very real sense, theirs as much as ours. We are grateful to the students who persevered through the abrupt transitions of the pandemic and its aftermath. This experience report exists because they were willing to learn alongside us as the course evolved. We also thank the institution and colleagues who supported the discipline over the years.

References

[1] Gabriela Alves, Pamella Soares, Bianca Souza, Necio Veras, and Allysson Allex AraúJo. 2022. What is students' perception of Flipped Classroom usage during Emergency Remote Teaching of Software Engineering subjects? An Experience Report in Brazil. In *Proceedings of the XXXVI Brazilian Symposium on Software Engineering* (Virtual Event, Brazil) *(SBES '22)*. Association for Computing Machinery, New York, NY, USA, 124–133. doi:10.1145/3555228.3555259

[2] Rossana M. C. Andrade, Ismayle de Sousa Santos, Italo Linhares de Araújo, Bruno Sabóia Aragão, and Fernanda Siewerdt. 2017. Retrospective for the Last 10 Years of Teaching Software Engineering in UFC's Computer Department. In *Proceedings of the XXXI Brazilian Symposium on Software Engineering* (Fortaleza, CE, Brazil) *(SBES '17)*. Association for Computing Machinery, New York, NY, USA, 358–367. doi:10.1145/3131151.3131179

[3] Vinicius de Oliveira Neves, Silvana Morita Melo, Davi Viana, Rodrigo Pereira dos Santos, Lina Garcés, and Valdemar Vicente Graciano Neto. 2023. Challenges on the Brazilian Information Systems Education: The Professors' Perspective. *IEEE Transactions on Education* 66, 6 (2023), 531–542. doi:10.1109/TE.2023.3299753

[4] Kiev Gama. 2019. An Experience Report on Using LEGO-based Activities in a Software Engineering Course. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (Salvador, Brazil) *(SBES '19)*. Association for Computing Machinery, New York, NY, USA, 289–298. doi:10.1145/3350768.3353817

[5] Lina Garcés and Brauner Oliveira. 2024. Teaching Software Engineering with Project-Based Learning: A Four Years Experience Report. In *Anais do XXXVIII*

Simpósio Brasileiro de Engenharia de Software (Curitiba/PR). SBC, Porto Alegre, RS, Brasil, 476–486. doi:10.5753/sbes.2024.3530

[6] IFBA. 2020. *Resolução n.º 23, de 17 de setembro de 2020 — Regulamento das Atividades de Ensino Não Presenciais Emergenciais (AENPE).* Technical Report. Instituto Federal de Educação, Ciência e Tecnologia da Bahia. Available at https://portal.ifba.edu.br.

[7] Julie E. Mills and David F. Treagust. 2003. Engineering Education — Is Problem-Based or Project-Based Learning the Answer? *Australasian Journal of Engineering Education* 3, 2 (2003), 2–16.

[8] Henry Antônio Silva Nogueira, Rucelino de Sousa Aguiar, and Maria Lourdes Gisi. 2023. A importância dos Institutos Federais de Educação, Ciência e Tecnologia para a redução da desigualdade educacional no Brasil. *Revista online de Política e Gestão Educacional* 27, 00 (jul. 2023). doi:10.22633/rpge.v27i00.18005

[9] Mayara Olivindo, Necio Veras, Windson Viana, Mariela Cortés, and Lincoln Rocha. 2021. Gamifying Flipped Classes: An Experience Report in Software Engineering Remote Teaching. In *Proceedings of the XXXV Brazilian Symposium on Software Engineering* (Joinville, Brazil) *(SBES '21)*. Association for Computing Machinery, New York, NY, USA, 143–152. doi:10.1145/3474624.3476971

[10] Fezile Ozdamli and Gulsum Asiksoy. 2016. Flipped Classroom Approach. *World Journal on Educational Technology: Current Issues* 8, 2 (2016), 98–105. doi:10.18844/wjet.v8i2.640

[11] Eliezer Pacheco. 2011. *Os Institutos Federais: Uma Revolução na Educação Profissional e Tecnológica.* Editora do IFRN. Available at http://memoria.ifrn.edu.br/handle/1044/1013.

[12] Roger S. Pressman and Bruce R. Maxim. 2014. *Software Engineering: A Practitioner's Approach* (8th ed.). McGraw-Hill Education.

[13] SBC. 2020. Meninas Digitais do Sudoeste da Bahia. Sociedade Brasileira de Computação (SBC) — Meninas Digitais Program. Available at https://meninas.sbc.org.br/portfolio-3/meninas-digitais-do-sudoeste-da-bahia/.

[14] SBC. 2022. *Educação Superior em Computação: Estatísticas 2022.* Technical Report. Sociedade Brasileira de Computação.

[15] Ian Sommerville. 2016. *Software Engineering* (10th ed.). Pearson.

[16] Marco Tulio Valente. 2020. *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade.* Available at https://engsoftmoderna.info. Continuously updated open-access textbook..

[17] Hustana Maria Vargas and Maria de Fátima Costa de Paula. 2013. A inclusão do estudante-trabalhador e do trabalhador-estudante na educação superior: desafio público a ser enfrentado. *Revista da Avaliação da Educação Superior (Campinas)* 18, 2 (Jul 2013), 459–485. doi:10.1590/S1414-40772013000200012

[18] Titus Winters, Tom Manshreck, and Hyrum Wright. 2020. *Software Engineering at Google: Lessons Learned from Programming Over Time.* O'Reilly Media.