A Syllabus for Teaching Software Project Management in Undergraduate Computing Courses

Elielton da Costa Carvalho[1], Sandro Ronaldo Bezerra Oliveira[1,*], and Fabrício Wickey da Silva Garcia[2]

[1]Graduate Program in Computing (PPGCOMP), Federal University of Pará (UFPA), Belém, Brazil
[2]Capitão Poço Campus, Federal Rural University of Amazon (UFRA), Belém, Brazil
Email: elieltoncarvalho2@gmail.com (E.C.C.); srbo@ufpa.br (S.R.B.O.); fabriciowsgarcia@gmail.com (F.W.S.G.)
*Corresponding author

Manuscript received June 30, 2025; revised October 25, 2025; accepted December 10, 2025; published June 11, 2026

***Abstract*—Efficient management is crucial for advancing software development, so training professionals in Software Project Management (SPM) has become essential for the technology sector. However, research indicates that computing graduates still need to develop key skills required by the market, including leadership, flexibility, tool usage, and comprehensive management capabilities. This research therefore presents a SPM syllabus designed to help students develop these skills. The methodology involved studying curricular and technical documents, such as the ACM/IEEE Computing Curricula 2020, the SWEBOK v4.0, and the Brazilian Computer Society Training References 2017. Based on this study, four teaching units were created. Each unit was planned with consideration given to prior knowledge, guiding questions, programme content, teaching strategies, expected results, and learning levels according to revised Bloom's Taxonomy. Teaching techniques include active methods such as project-based learning, the flipped classroom approach, the use of tools, gamification, and collaboration with industry. To validate the syllabus, it was evaluated by experts with extensive experience of student-centered approaches and SPM, Computing, and Software Engineering. After analyzing the suggestions, all of them were incorporated, resulting in a robust syllabus that the evaluators recommended for use in undergraduate computing courses.**

***Keywords*—syllabus, software project management, active teaching methodologies, computing**

I. Introduction

The growing complexity of software systems, coupled with the volatility of organizational environments, has made project management an essential skill for computing professionals. In this context, Software Project Management (SPM) encompasses a range of technical, strategic, and interpersonal skills that interact with the dynamic reality of the technology industry [1, 2].

However, there is growing evidence of a mismatch between the knowledge taught in academia and the skills demanded by the software industry. According to Diniz *et al*. [3], this 'skills gap' is a direct result of curricula that are overly theoretical and fail to include practical skills such as teamwork, conflict resolution, and risk management in real project environments. This gap not only compromises the employability of graduates, but also their ability to adapt to agile and value-oriented contexts.

Recent pedagogical experiments have identified active methodologies as an effective alternative to the traditional approach. For example, applying Project-Based Learning (PBL) to an Information Systems course led to significant improvements in engagement, content retention, and socio-emotional skill development [4]. Similarly, PBL fosters autonomy and motivation among students on computer courses, preparing them better to tackle real-world problems [5].

The need to reconsider how the subject is taught is also supported by international guidelines. The ACM/IEEE Computing Curricula 2020 defines project management as a fundamental competency in undergraduate computer science programs and recommends the use of student-centred strategies based on real-world problem solving [6]. In turn, SWEBOK (Software Engineering Body of Knowledge) v4.0 [7] recognizes project management as one of the key areas of software engineering, incorporating updated practices such as DevOps and agile engineering, as well as techniques based on artificial intelligence.

Another relevant conceptual framework is the Project Management Body of Knowledge (PMBOK) 7th edition, which replaces a process-based structure with a more flexible, principle- and performance-domain-guided approach [8]. By emphasizing value delivery and adaptation to stakeholders' needs, this change repositions SPM teaching as a field that demands technical expertise, as well as pedagogical and contextual sensitivity [8]. In Brazil, the Brazilian Computer Society's (SBC) Training References [9] also reinforce this perspective, advocating training that combines technical and scientific knowledge, critical thinking, and interpersonal skills. Furthermore, Paiva *et al.* [10] have reported on the successful use of real projects with stakeholders as a training tool in Software Engineering courses. Section II provides the theoretical basis for a better understanding of the research context and is essential reading for interpreting the results.

In this context, this article presents a syllabus for the SPM subject for undergraduate computing students, combining established subject foundations with innovative pedagogical methodologies. The proposal is based on international and Brazilian curricular documents and integrates practices such as PBL, flipped classrooms, and expert validation. The aim is to provide training that better aligns with the demands of the contemporary software market.

In addition to this introductory section, the article is structured as follows: Section II presents the theoretical basis for the research; Section III presents some related works; Section IV presents the methodology used; Section V presents the syllabus in detail; Section VI presents the evaluation model and weight justification used in the syllabus; Section VII presents the syllabus evaluation report; and Section VIII presents the conclusions of this work.

II. Theoretical Framework

This section will present the necessary theoretical basis for

doi: 10.18178/ijiet.2026.16.6.2609

this research.

A. Teaching Software Project Management

Software Project Management requires a combination of technical and interpersonal skills, including communication, leadership, organization, and conflict resolution. This multidimensionality presents particular challenges in teaching, especially at undergraduate level. According to SWEBOK v4.0, SPM is recognized as a key area of software engineering, covering everything from resource estimation and risk management to quality control and stakeholder communication [7].

Curriculum documents also reinforce its centrality. ACM/IEEE [6] highlights project management as an essential component of professional training in computer science, suggesting pedagogical approaches that simulate the resolution of real-world problems. In Brazil, SBC [9] classifies SPM as part of the professional axis of computer science courses and recommends teaching it based on applicable skills and integration with labour market practices.

However, recent studies reveal limitations in the methods adopted to teach SPM. Students are often not exposed to the complexity of managing interdisciplinary and collaborative systems, which hinders their ability to adapt to the corporate environment [11]. Furthermore, Chunxiao *et al*. [12] propose the ‘Case+’ model, in which real cases form the basis for developing project management skills, enabling greater engagement with and assimilation of the content in a practical context.

B. Student-Centered Teaching and Learning Strategies

In recent years, it has become increasingly common—and, by all accounts, necessary—to reconsider the role of students in the learning process. This change in perspective did not come out of nowhere. It stems largely from a critical view of traditional teaching, which still relies heavily on the professor as the sole source of knowledge. In many courses, particularly technical ones, this approach still dominates.

However, the growing complexity of the world and of working relationships requires a different approach: an education that values active listening and participation, and that recognizes the leading role of students in the learning process. Active methodologies have emerged precisely in response to this demand for change. They challenge the conventional model by encouraging students to play a more active and critical role in their own learning journey [13, 14].

In this new arrangement, the role of the professor has also changed. Rather than simply passing on content, they are now expected to guide, provoke, and stimulate students’ thinking. This does not mean giving up technical mastery, but adopting a stance that favors the joint construction of knowledge. When implemented effectively, active teaching and learning strategies have demonstrated significant potential for fostering critical thinking and facilitating horizontal spaces for exchange and debate [15].

This approach favors an understanding of the content and contributes to the formation of individuals who are more autonomous, capable of reflection, learning regulation, and making informed decisions [16].

The flipped classroom offers a straightforward yet effective solution: rather than attending a lecture and then practising, students prepare in advance and use the class time to discuss, solve problems, and apply concepts with the professor’s support. This change fosters greater autonomy, participation, and critical thinking. Recent applications of this approach in technical courses, such as cybersecurity, have demonstrated increased student engagement and improved conceptual understanding [17]. Rahayu *et al.* [18] also found that combining the methodology with project-based learning enhances active learning. Despite some challenges, such as unequal access to technology, studies such as [19] confirm that the benefits outweigh the limitations. In project management, this strategy has proven especially promising when integrated with interactive and gamified practices [20].

Project-based learning is an approach that places the student at the centre of the learning process. Rather than passively absorbing theory, students engage in solving real problems, typically in groups, thereby fostering both technical reasoning and social skills. Recent studies show that PBL contributes to socio-emotional development even at early levels of education [21]. In higher education, the benefits are even greater: Halawa *et al*. [22] and Kalsum *et al.* [23] highlight improvements in understanding concepts and performance in practical activities. Wani *et al.* [24] reinforce the idea that this methodology also strengthens decision-making skills and the ability to overcome practical challenges, which is particularly relevant in subjects such as Software Project Management.

Industry participation in the classroom is an effective way of bringing theory and practice closer together, particularly in SPM. Lectures with industry experts expose students to real-life situations and concrete challenges that go beyond textbooks. This type of interaction broadens students’ repertoire and helps them to understand how the concepts covered in class apply to day-to-day market practice [25]. Additionally, it promotes exposure to various leadership styles, agile methodologies, and team dynamics — elements that are challenging to replicate in purely academic settings.

Using project management tools has brought teaching closer to professional practice. When students explore these platforms independently and then apply what they have learned in class, as proposed by the flipped classroom model, their engagement and understanding noticeably increases. Chunxiao *et al.* [12] argue that this approach favors contextualized learning, and Trafford and Head [26] reinforce the idea that combining technology with real projects broadens students’ technical understanding, even in multidisciplinary classes.

Including game elements in teaching, such as challenges, scores, and rewards, has proven to be an effective way of making learning more dynamic. In this sense, gamification can facilitate understanding of complex content and stimulate skills such as autonomy and collaboration. Antonovskiy and Zhuykova [27] observed that students feel more motivated and engaged when this type of strategy is employed. Similarly, Pavlova *et al.* [28] point out the benefits of professional development when gamification involves platforms such as Moodle and Kahoot. In Project Management teaching, it has the potential to make the process more practical and engaging.

III. Related Works

In the specialized literature, it is evident that researchers are striving to implement syllabuses that encourage students' active participation in the classroom. In computing in particular, some research is related to the proposals set out in this article, which will be detailed below.

Quaresma and Oliveira [29] propose a teaching plan focused on the software process, based on curricular documents such as the ACM/IEEE and SBC standards. This approach emphasizes active methodologies and has been validated by experts, thereby reinforcing the importance of aligning teaching with professional practice.

Furtado and Oliveira [30] explore the use of gamification in teaching software measurement, achieving positive results in terms of engagement and retention. Their proposal has a more restricted focus, centered on serious games and a single theme of software engineering.

Ferreira and Oliveira [31] present a plan centered on exploratory testing, based on interviews with professionals and curricular references. This work is notable for its alignment with the TMMi (Test Maturity Model Integration) model, as well as its emphasis on applied practices.

Finally, Elgrably and Oliveira [32] present a teaching model centered on testing in agile environments, integrating gamification and simulations of real-life scenarios to develop specific technical skills.

Unlike the aforementioned authors, this work covers the subject of software project management. Furthermore, it has a broader scope, incorporating multiple pedagogical strategies, such as problem-based learning, the flipped classroom approach, market tools, and industry participation, to create an integrated syllabus that meets the current requirements of computer training.

IV. Materials and Methods

This research was structured into three stages: a survey, construction, and evaluation. The steps can be seen in detail in Fig. 1.

Step 1 involved conducting research that resulted in study [33], which identified the strategies adopted in SPM teaching. During the review, the frequency with which certain practices appeared in the analyzed studies caught our attention. It was possible to notice that some practices appeared quite frequently among the analyzed studies. Strategies such as PBL, the flipped classroom, and gamification were mentioned as effective ways to engage students. Another notable point was the involvement of professionals from industry. In many cases, inviting someone from industry to share their practical experience helped contextualize the content better, bringing the classes closer to the reality that students will face. These findings formed the basis for structuring the syllabus and directly influenced subsequent steps in the research.

Step 2 involved conducting a Brazilian curricular diagnosis focused on analyzing the Project Management curriculum in a sample of Computer Science courses in Brazil [34]. For this step, the Pedagogical Course Projects (PPCs) of 22 educational institutions were analyzed. These institutions were selected based on the Folha University Ranking (RUF) [35]. Despite differences in subject names and formats, regularity was observed in topics covered, which tended to focus on three areas: basic concepts of the subject, management practices (traditional and agile), and human project management issues, such as leadership and teamwork. This step also revealed partial convergence between Brazilian curricula and international references.

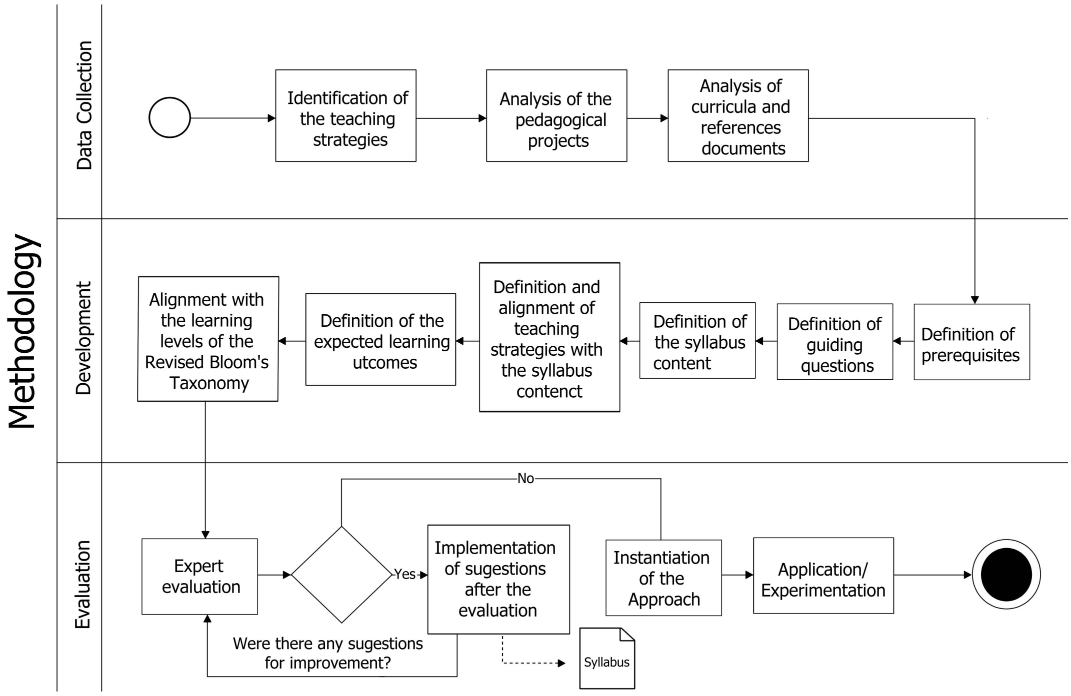

Fig. 1. Research methodology flow.

Step 3 involved mapping three key documents: Computing Curricula 2020 [6], SWEBOK v4.0 [7], and SBC Curricular References [9]. The analysis focused on identifying structuring curricular assets, such as content topics, competencies, knowledge areas, and training objectives, and their intersections. Cross-referencing these documents enabled us to recognize the complementarity of technical (SWEBOK), formative (RF-CC-SBC), and competency-based (CC-2020) approaches, forming the conceptual basis for the syllabus structure [36].

In Step 4, we aimed to identify which prior knowledge could facilitate the learning of the provided subject content. Rather than establishing formal barriers, the idea was to recognize that certain previous experiences can facilitate the learning process. While these prerequisites are not mandatory, they provide a supportive foundation that helps students progress more confidently through the units.

Step 5 involved developing guiding questions for each teaching unit. These questions play a fundamental role at the beginning of each module, encouraging students to reflect on real, contextualized problems. The aim is to stimulate curiosity and encourage critical discussion even before the formal presentation of the content. This 'initial problematization' process allows prior knowledge to be activated and guides the focus of learning throughout the unit.

Step 6 involved defining the programme content for each of the four units that make up the syllabus. Topics were selected based on the competencies identified in the previous step and the curricular references analyzed (CC-2020, SWEBOK, RF-SBC). The organization of the content sought to strike a balance between technical aspects, such as the project life cycle and management methodologies, and human aspects, such as leadership and team management, in order to promote an integrated view of the subject.

Step 7 involved defining the most suitable teaching strategies for each unit. These were guided by the findings of Step 1 and by alignment with student-centered pedagogical guidelines. Thus, methodologies such as the flipped classroom approach, project-based learning, the use of professional tools, group dynamics, gamification, and the participation of industry professionals through lectures were incorporated. Each unit was designed using one or more of these strategies to promote student engagement and active learning.

Step 8 involved formulating the expected results for each unit of the subject. The aim was to clearly define what the student should be able to do by the end of each stage, in both conceptual and practical terms. This includes understanding a management model, applying a tool in a real project, and making collaborative decisions in a group. These results act as a kind of compass, helping to guide the design of activities while also serving as a reference for assessment methods.

In Step 9, the course content was compared with the learning levels defined by Bloom's Revised Taxonomy [37]. The intention was to organize the topics by subject and by the type of cognitive effort that each one requires from the student. Thus, while some topics only require students to recognize and understand basic concepts, others require practical application, critical analysis, or even the creation of original solutions. This structure facilitated consideration of the progression of activities and ensured that the course covered different levels of complexity throughout the units, rather than focusing on a single type of skill.

The evaluation phase began in Step 10, when experts with experience in SPM and student-centred teaching methodologies analyzed the syllabus. Based on their observations, areas for improvement were identified. These suggestions were incorporated and final adjustments were made in Step 11.

The last 2 steps of the methodology are the instantiation phase (Step 12), which involves applying and experimenting with the approach. This is when the approach is adapted to the real classroom context. This instantiation may involve using different tools and support resources, such as artificial intelligence solutions, accessibility mechanisms or other educational technologies relevant to the class profile.

Once instantiated, the approach moves on to the application/experimentation step (Step 13), where it is implemented in a real teaching environment. The results obtained in this phase will be evaluated through quantitative and qualitative analyses, providing a basis for further feedback to continuously improve the approach. Thus, the findings from experimentation contribute to improvements, reinforcing the iterative nature of the process.

It should be noted that although the complete methodological model includes these two steps, they are beyond the scope of this article and are planned for future development and presentation in a complementary work.

The main deliverable of this study was finally consolidated at the end of the above steps: a syllabus focused on teaching SPM, structured based on student-centred approaches, and validated by experts. With the necessary revisions incorporated, the proposal is ready to be implemented in undergraduate computing courses.

V. Results

This section describes the syllabus for teaching Software Project Management in detail.

A. *Identification of Competencies*

The definition of the competencies that guide the proposed syllabus is aligned with the guidelines present in international documents such as Computing Curricula 2020, as well as Brazilian references such as RF-CC-SBC. The mapping carried out in previous methodological steps enabled a set of core contents and competencies to be consolidated for SPM training (see Table 1), which were then incorporated throughout the four units of the syllabus. Based on the aforementioned guidelines, content balancing the technical and human aspects of the subject was selected. The following topics stand out: (1) the concepts and foundations of project management; (2) the life cycle and knowledge areas; (3) traditional and agile management methodologies, including the use of standards, guides, and digital tools; and (4) human aspects, such as stakeholder management, cultural diversity, and teamwork. These contents were organized in a gradual manner, respecting the complexity of each one and the development of the expected skills throughout the course.

The course was designed to do more than just cover content; it was designed to develop skills that can be applied outside the classroom. Students should understand why certain fundamentals are important, how to choose the right tool for each type of project, and how to deal with technical, human, or mixed challenges. It's not just about following a model; it's about knowing how to approach the project as a living, nuanced entity. Students are also expected to collaborate and demonstrate leadership in diverse and dynamic environments. The entire process was organized based on the revised Bloom's Taxonomy [37] to ensure progressive learning, from theoretical assimilation to concrete application in everyday professional scenarios.

Table 1. Competencies related to the contents

Competencies	Contents
Recognize the principles of software project management and distinguish projects from ongoing operations	Project management concepts
	Principles of project management
Understand the objectives of a project	Objectives of a software project
Understand the role and impact of project management in the organizational context	Importance of software project management
	Impact on the organization
Understand the concept and objectives of the project life cycle	Software project life cycle
Understand and apply the phases and performance domains of software project management	Phases of software project management
	Project performance domains
Analyze project life cycle models, differentiating their applications and evaluating the success criteria in each phase	Predictive and adaptive life cycle models
	Success criteria throughout the project
Understand the principles of the Agile Manifesto and apply the structure of agile frameworks in software project management	Project management methodologies Software
	Agile Project Management Methodologies
	Agile management framework
Apply project monitoring and control techniques with the support of digital management tools	Software Project Management Tools and Techniques
	Project management support software
Analyze the influence of stakeholders and cultural challenges in software project management throughout their life cycle	Stakeholder management
	Cultural diversity and inclusion
Understand the importance of collaboration and functional roles within a project team	Teamwork and collaboration
	Roles and responsibilities in software projects
Apply the knowledge acquired in software project management in practical contexts	Final software project

B. *Content Planning*

Once the essential skills and content for teaching SPM had been defined, the material was organized into four progressive and articulated teaching units, namely: (i) Concepts and Fundamentals, (ii) Project Life Cycle, (iii) Management Methodologies, and (iv) Human Aspects in Software Project Management. The structure of each unit is designed to promote active learning.

Each unit follows a standardized model composed of six central elements. The first of these are the prerequisites, which indicate prior knowledge that is recommended, although not mandatory, to facilitate understanding of the new content.

Although prior exposure to software process concepts is recommended, the first unit revisits and contextualizes these fundamentals within the scope of software project management. This approach enables students with limited prior experience to build the necessary foundation, while those with previous knowledge can consolidate their understanding through applied perspectives. Thus, the prerequisite serves as an orientation rather than an exclusion criterion, enabling the syllabus to bridge the conceptual gap between Software Engineering and Project Management.

Next, guiding questions are presented to stimulate discussion of the topic through real or hypothetical problem situations.

The program content of each unit is defined based on the planned skills, ensuring alignment with Brazilian and international curricular references. Teaching strategies consistent with the subject's profile and the active methodology proposal, such as the flipped classroom, PBL, and the use of digital tools, were chosen based on this content.

Expected learning results were also defined to clarify what students should be able to do or understand by the end of each unit. Finally, each topic was associated with a learning level of the Revised Bloom's Taxonomy to balance the cognitive complexity of the activities and promote continuous evolution in the educational process. Details of each teaching unit can be found in Tables A1, A2, A3, and A4 in Appendix A.

It is important to clarify that this work does not present a teaching plan, but rather a syllabus that provides a framework for organizing learning objectives, content, and teaching strategies in a structured and generalizable way. A detailed schedule, workload distribution, and specific classroom implementation will be included in the teaching plan, which will be developed in a subsequent phase during the implementation of this syllabus in an undergraduate computing programme.

C. *Identification of Content in the Literature*

The contents of the teaching units in this syllabus were defined based on an analysis of the bibliography and technical documentation used in Software Engineering teaching. The selection aimed to strike a balance between an academic approach and adherence to professional practices, incorporating classic authors, good practice guides, and Brazilian and international standards.

Two books stand out among the selected support materials for their central role in shaping the content. The first, by Pressman and Maxim [2], provides a practical approach to software engineering with a direct focus on professional routines, making it useful for connecting theory to project management in practice. The work by Sommerville [1], adopted as supplementary material, offers a more conceptual and didactic perspective, providing a foundation for further theoretical exploration and fostering a more structured understanding of the fundamentals of the field.

In addition to these works, the content was aligned with the main technical references, including SWEBOK v4.0 [7], PMBOK 7th ed. [8], the Brazilian MPS.BR (Software Process Improvement) model [38], and CMMI (Capability Maturity Model Integration) v3.0 [39]. The aim was to integrate widely recognized standards in the market and academia. These documents not only contributed to the selection of topics, but also to the definition of performance domains, success criteria, and monitoring practices that constitute the expected student skills.

The relationship between the content and its sources can be examined in detail in the relevant tables for each unit, enabling professors to grasp the theoretical and normative foundations that underpin the pedagogical design of the subject. The references associated with the respective contents can be found in Tables A5, A6, A7, and A8 in Appendix B.

This syllabus offers a pedagogically grounded, integrative model for teaching SPM. While the CC2020 curriculum and the SBC Curriculum define learning results and professional profiles, the SWEBOK v4.0 and the PMBOK 7th edition are professional guides that focus on best practices and knowledge domains. This syllabus connects these distinct purposes by translating their conceptual frameworks into a sequence of teachable units supported by active methodologies and assessment criteria (see Tables A1, A2, A3, and A4).

The model reconciles the conceptual tension between the principles-based guidance of the PMBOK and the process-based framework of the SWEBOK, which provides detailed, prescriptive activity instructions throughout the project life cycle. This reconciliation is reflected in the program through the integration of the principles of Unit II (Project Life Cycle) and the application of structured processes in Unit III (Management Methodologies). This allows students to understand the necessity of management actions and how they are implemented in practice.

Rather than reproducing the scope of these reference documents, the program synthesizes and operationalizes them pedagogically, enabling educators to apply the essential elements of professional standards in an academic setting.

D. Description of Teaching Strategies Adopted

The teaching strategies developed for the SPM syllabus are detailed below. These strategies are based on the understanding that knowledge is actively developed by the student based on their experiences, thoughts, and interaction with the presented material. In this approach, the student is the protagonist, and their existing knowledge is crucial for acquiring new information [40].

These strategies have already been incorporated into the theoretical framework through the use of active methods, such as the flipped classroom, problem-based learning, gamification, and collaboration with the sector. However, an additional strategy has been incorporated into the planning process and will now be explained: the theoretical class.

1) Theoretical class

The theoretical class combines the presentation of conceptual content by the professor with the active participation of the students, and is established as a relevant pedagogical strategy. This approach has been associated with higher levels of engagement and meaningful learning since it enables students to appropriate the content through discussion, problem solving, and the exchange of experiences [41]. Literature shows that well-structured teaching support directly impacts motivation, academic self-efficacy, and student participation [41].

a) Methodological procedures

The theoretical and practical class involves explaining the content in a logical sequence of steps: (i) initial presentation of the concept, (ii) explanation of its application in the context of the subject, (iii) demonstration of the practical use of the concept, (iv) analysis of a real case to illustrate its application, (v) questions and discussions about the previous steps to promote understanding and debate among students.

b) Infrastructure

The infrastructure used for developing the theoretical and practical classes is as follows: (i) the syllabus as a basis, (ii) a projector, computer, paintbrush, and whiteboard as physical materials, (iii) a presentation software tool.

c) Assessment

Students will be assessed in class based on the quality and quantity of their participation through questions, statements, suggestions, and attempts to provide examples in academic or professional situations.

2) Flipped classroom

In the flipped classroom model, the material that was previously presented in class is now explored by the student first. This changes the teaching method: face-to-face time is no longer solely for passive listening, but becomes a space for interaction, including debates, group activities, and problem solving. In this dynamic, the student leads the construction of knowledge and the professor guides and provokes reflection. Recent studies show that this approach helps develop skills such as critical thinking and autonomy. It also improves engagement with the subject and retention of content, particularly in higher education. Combining it with digital resources enhances its effects further [19, 42].

a) Methodological procedures

The methodological procedures of the flipped classroom include the following steps: (i) the professor prepares basic study materials and defines learning objectives, (ii) students access the content before the face-to-face class, (iii) during the class, topics are divided among groups, (iv) there is a class discussion about main points covered and clarification of doubts, (v) outside the classroom, students produce presentations based on studied content and new researched materials, (vi) in the classroom, groups present content and, when necessary, deliver written reports on topics covered.

b) Infrastructure

In addition to the requirements for the theoretical class, this teaching strategy requires a virtual learning environment.

c) Assessment

The assessment process will be carried out in three stages: (i) participation in the initial class discussion on the topic, led by the professor, (ii) peer review, in which classmates evaluate the presentation and mastery of the studied content based on previously defined criteria, and (iii) assessment of reports on produced knowledge, which will be carried out by the professor in addition to the presentation, when necessary.

3) Project-Based Learning (PBL)

PBL is an active approach that promotes student leadership in collaborative activities aimed at solving real-world problems. With the professor acting as a facilitator, PBL stimulates cognitive and socio-emotional skills, increases autonomy, and strengthens engagement. Recent evidence shows that this methodology is effective in developing complex skills and promoting more meaningful, interdisciplinary learning [21, 22, 24].

a) Methodological procedures

The methodological procedures in project-based learning include the following steps: (i) the professor prepares the basic study materials and defines the learning objectives, (ii) students access the content before class, (iii) during class, topics are distributed among groups, (iv) the project is developed in class, with a section for clarifying doubts, (v) outside the classroom, students develop the project and prepare a report on the results obtained, (vi) in class, groups present the results achieved upon completion of the project and share the lessons learned.

b) Infrastructure

The infrastructure required for project-based learning development follows the same principles as those described for the flipped classroom.

c) Assessment

The assessment process will be carried out in three steps: (i) presentation of the project results in class, (ii) peer review, where classmates evaluate the presentation based on established criteria, and (iii) evaluation of the results reports by the professor.

4) Industry participation

Including industry professionals in academic subjects, especially through lectures, has proven to be an effective strategy for connecting theory and practice. This interaction provides students with a more concrete understanding of the content studied, increases their motivation, and develops their technical and interpersonal skills, which are in line with the demands of the job market [12].

a) Methodological procedures

The methodological procedures include the following steps: (i) the professor identifies and invites industry professionals to give lectures on relevant topics, (ii) the lecturers present their cases, experiences, and practices in software project management, (iii) during the lectures, students are encouraged to ask questions and interact with the lecturers to explore practical aspects, (iv) after the lecture, the professor leads a class discussion to reinforce the key learning points and relate them to the theoretical content, (v) the students prepare a brief report or summary of the knowledge acquired and how applicable the discussed topics are.

b) Infrastructure

The infrastructure required for the lectures is as follows: (i) a classroom equipped with audiovisual resources (e.g. a projector and a microphone), (ii) videoconferencing tools for remote speakers, (iii) adequate space for questions and group discussions.

c) Assessment

The assessment process will consist of: (i) active participation in lectures and discussions, (ii) evaluation of the quality of questions and student engagement during lectures, and (iii) evaluation of reports or written summaries of learning and reflections on lectures.

5) Use of project management tools

In this syllabus, project management tools are considered an integrated support resource for active learning strategies, particularly project-based learning, the flipped classroom, and gamification.

Rather than functioning as a standalone teaching strategy, these tools facilitate collaboration, organization, and monitoring of learning results during project activities.

The integration of project management tools follows contemporary digital learning approaches that view technology as a mediating resource rather than a standalone methodology. Incorporating digital tools into project-based instruction enhances collaboration and connects theory to practice [43]. In this syllabus, such tools support the cognitive and procedural objectives of the main strategies (PBL, the flipped classroom, and gamification), thereby reinforcing autonomy and teamwork.

a) Methodological procedures

The methodological use of project management tools follows the same sequence adopted in flipped classroom and project-based learning strategies: (i) the professor provides the students with study materials on the tools, such as tutorials, guides, and videos, for them to study in advance, (ii) the students are divided into groups, each of which is responsible for exploring a specific project management tool, (iii) during the class, each group presents the operation and main features of their tool to the other students, (iv) after the presentations, the students apply the tool they studied to the management of the project they developed throughout the course using project-based learning;, (v) the groups prepare a final report, highlighting the lessons learned, the challenges they encountered, and how effective the tool was in project management.

This process is not assessed separately, but as part of the overall assessment of PBL and flipped classroom activities.

b) Infrastructure

The infrastructure required for this strategy consists of: (i) access to computers with an internet connection, (ii) licences or accounts to access project management tools (e.g. Jira, Trello, Redmine, and MS Project), (iii) a digital environment for sharing study materials, (iv) a physical or virtual space for presentations and group discussions.

c) Assessment

Tool use is evaluated as part of the evaluation of other strategies, particularly PBL and flipped classroom. It focuses on: (i) assessment of student participation and engagement in presentations on the tools, (ii) analysis of the practical application of the tools in project management throughout the course, (iii) quality assessment of the final report, focusing on lessons learned and challenges faced, (iv) feedback on management and problem-solving skills when using the tools.

6) Gamification

Introducing typical game features into the learning context has proven to be an effective way of boosting students' interest and willingness to learn, as well as encouraging teamwork. When integrated with active approaches such as the flipped classroom and problem-based learning, it optimizes the assimilation of topics and the development of key skills. Additionally, gamification enables close monitoring of progress and provides constant feedback to students [44, 45].

The gamification model set out in this syllabus is intended to complement the formal assessment process by providing a motivational and engagement framework. Rather than serving as a grading mechanism, its purpose is to encourage participation, collaboration, and self-regulated learning by setting meaningful challenges.

The model's design is based on Self-Determination Theory (SDT) [46], which emphasizes three psychological needs that foster intrinsic motivation: autonomy, competence, and relatedness. Each game element is intentionally aligned to one or more of these needs:

- Autonomy: Students choose how to contribute to group activities and select roles (Leader, Strategist, or Explorer) that fit their learning preferences,
- Competence: Experience Points (XP) reward both individual progress and collective achievements, reflecting mastery of cognitive and procedural skills,
- Relatedness: Collaborative missions and team challenges encourage peer interaction and interdependence, thereby reinforcing the cooperative nature of project management.

The optional, adaptive ranking system functions as a feedback mechanism rather than a source of competition. It can be disabled or replaced with a team-based progression model, where groups advance through the levels collectively based on their overall performance. This approach mitigates competitive bias and strengthens collaboration, which is one of the key learning results of software project management.

The model follows the principles of meaningful gamification, prioritizing engagement through reflection, purpose, and voluntary participation rather than mere reward accumulation (‘pointsification’). Elements such as missions, badges, and avatars primarily serve as symbolic reinforcements that contextualize the learning journey.

a) General structure of gamification

- **Experience Points (XP):** Students earn XP for participation, assignments, presentations and interaction. Rather than serving solely as a competitive score, XP functions as formative feedback, helping students track their progress and reflect on skill mastery.
- **Levels and Roles:** Students choose from four roles — Leader, Strategist, Explorer, or Learner — based on their personal interests and strengths. These roles determine how students contribute to activities and missions, promoting autonomy and enabling learning paths to be tailored to individual preferences. Although rankings based on XP are optional, they can be applied and replaced with a group-based progression model that emphasizes collaboration over competition.
- **Bonuses:** Students can earn extra points for achieving certain milestones, such as giving their first presentation, asking the best question, or delivering an excellent report. These bonuses serve as a form of recognition and feedback, reinforcing meaningful engagement.
- **Ranking (optional):** This can be either individual or team-based. When applied, it is adaptive and does not undermine collaborative dynamics.
- **Missions:** Tasks and challenges with clear goals that are connected to the course content. These missions provide opportunities for mastery, creativity and reflection, and support both competence and autonomy. Examples include ‘Problem Detective’ and ‘Solutions Engineer’.
- **Avatars or Team Names:** Students or groups create identifiers, such as ‘Scrum Masters Team’ or ‘Code Ninjas’. This fosters a sense of identity, ownership and social connection, enhancing relatedness.
- **Symbolic Rewards (optional):** Certificates or small prizes are provided as contextual markers of achievement, rather than as primary motivators, such as:
  i. **1st place in the ranking:** Prize 1 + Symbolic certificate.
  ii. **2nd place in the ranking:** Prize 2 + Symbolic certificate.
  iii. **3rd place in the ranking:** Prize 3 + Symbolic certificate.

The gamification score sheet can be accessed from the link: https://zenodo.org/records/15313889.

The following items present the details of the scoring rules used during the application of each teaching and learning strategy:

1) Theoretical class

- **XP for active participation:** Asking questions, answering questions, providing examples and solving exercises.
- **‘Speaker of the Day’ bonus:** Recognizes the most interesting contribution.
- **Missions:** e.g. ‘Lightning Mission’ (solving a problem) or ‘Explain Mission’ (explaining a concept with personal or professional examples).
- **Connection to SDT:** Students have autonomy in choosing which activities to engage in, they demonstrate competence by achieving mastery, and they experience relatedness by interacting with peers.

2) Flipped Classroom

- **XP** awarded for participation, clarity of presentation and report submission.
- **Bonus** for pre-class study (quiz).
- **‘Source Hunter’ mission**: Providing reliable and innovative sources.
- **SDT alignment:** Students self-regulate their preparation (autonomy), receive feedback through XP and mission outcomes (competence) and participate in discussions (relatedness).

3) Project-Based Learning

- Earn **XP** for participation, presentations and reports.
- **‘Problem Detective’ bonus:** identify points for improvement.
- **‘Solutions Engineer’ mission:** Most creative project solution.
- **SDT alignment:** Team projects foster collaboration and creative solutions, building mastery and competence while giving students choice in approach.

4) Industry Participation (Lectures)

- **XP** for participation and reflection reports.
- **‘Best Report’ bonus:** recognizes quality reflection.
- **‘Interviewer’ mission:** Ask high-quality questions.
- **‘Connection Point’ mission:** Relate lecture content to course topics.
- **SDT Alignment:** Students reflect on and connect knowledge independently (autonomy), develop expertise (competence) and interact with peers and guest lecturers (relatedness).

5) Use of Project Management Tools

- **XP** for presenting a functional tool and reporting on challenges and lessons learned.
- **‘Tool Instructor’ bonus:** Best presentation.

- **'Manager Grade 10' mission:** Demonstrate mastery of tool application.
- **SDT alignment:** Hands-on activities foster skill mastery (competence), allow students to choose how to apply tools (autonomy) and encourage collaborative problem solving (relatedness).

Finally, the model will be empirically validated through a future classroom experiment that combines qualitative and quantitative analysis.

VI. Evaluation Model and Weight Justification

The syllabus's evaluation model is designed to be continuous and progressive, reflecting the gradual acquisition of skills and encouraging student agency/protagonism. The final grade corresponds to the arithmetic mean of the grades obtained in Units I, II, III, and IV, with a maximum value of 10 points per unit.

The weighting of each teaching strategy is crucial and aligns with the required cognitive complexity according to Revised Bloom's Taxonomy. The adaptation and distribution of weights were based on the study plan proposed by [47].

Strategies with greater weight prioritize activities that require students to apply knowledge practically and develop complex skills, which are essential for professional performance.

The score for each unit is distributed among the active strategies adopted according to the weight and justification defined in Table 2.

Table 2. Distribution of strategy weights by unit

Unit	Teaching Strategy	Unit Weight	Weight Justification
I	Flipped Classroom (FLC)	50%	It stimulates conceptual understanding and communication skills, promoting active engagement in building the foundations of project management.
	Theoretical Expository Class	30%	It provides the necessary factual and conceptual foundation to support subsequent units, offering an introduction to international project management terminology and principles.
	Guided Discussion (Success and Failure Cases)	20%	The initial analysis activity (Analyse/Conceptual) develops critical thinking and contextualization skills.
II	Project-Based Learning (PBL)	50%	This activity is the most important one as it marks the beginning of the integrative project. It involves the 'Apply' and 'Analyse' levels, simulating professional practice in structuring the life cycle.
	Flipped Classroom (FLC)	40%	It encourages research and discussion of performance domains (PMBOK, 7th edition), reinforcing collaborative and synthesis skills.
	Theoretical Expository Class	10%	It provides conceptual support for life cycle models (predictive/adaptive). It maintains theoretical consistency with global frameworks (PMBOK, SWEBOK).
III	Guided Practice / Scrum Simulation	50%	It carries greater weight because it requires the procedural application of agile methodologies (Scrum), which are aligned with international standards (Agile Manifesto, ISO/IEC 12207). It reflects the Apply/Procedural and Analyse levels.
	Flipped Classroom (FLC)	40%	It focuses on the practical contextualization and presentation of standards (PMBOK, CMMI, and MPS.BR). It promotes critical comparison and internalization of standards.
	Theoretical Expository Class	10%	It offers a conceptual framework that supports both agile and traditional practices, thereby consolidating the integration of theory and practice.
IV	Flipped Classroom (FLC)	40%	It involves the Understand/Conceptual and Analyze/Conceptual levels, promoting critical thinking and empathy in multicultural and distributed contexts.
	Case Study / Guided Discussion	20%	It focuses on analyzing real-life situations involving stakeholders and global teams. It develops communication and conflict resolution skills.
	Final Project (Presentation and Reflection)	40%	This activity builds on learning from previous units. It incorporates Apply/Procedural and Evaluate/Conceptual, reflecting technical proficiency and reflective maturity.

A. Performance Criteria (Rubrics)

To ensure consistency and objectivity when assessing student performance, specific criteria (rubrics) have been defined for each teaching strategy. These focus on the application, analysis and reflection dimensions of the Revised Bloom's Taxonomy [37].

These rubrics guide the continuous and progressive assessment of the competencies covered in the four units of the curriculum, taking into account cognitive, procedural and attitudinal aspects.

Each strategy criterion includes performance descriptions that correspond to different levels (Excellent, Good, Fair, or Insufficient). These descriptions offer guidance on how to differentiate between students' performance levels.

1) Project-Based Learning (PBL)

PBL assesses students' ability to apply project management methodologies in simulated and real-world contexts by developing work products and providing evidence of software management.

The rubric includes four main criteria, which are continuously assessed:

- **Planning and Organization:** Appropriate project structuring, definition of roles, deadlines, and goals aligned with the scope,
- **Application of Tools and Methods:** Correct use of frameworks, techniques, and management support software (e.g., PMBOK, Scrum, Kanban),
- **Delivery of Work Products:** Production of coherent documentation of project results (reports, backlogs, plans, metrics),
- **Reflection on the Process:** Critical analysis of decisions made, challenges faced, and results achieved.

The performance levels for this strategy are as follows:

- **Excellent:** The project is well-structured, with clearly defined and aligned roles, deadlines and goals,
- **Good:** Project is structured with minor gaps in clarity or role definition,
- **Fair:** The project shows partial structure with significant gaps in clarity or role assignments,
- **Insufficient:** Project lacks clear structure, goals and defined roles.

2) Flipped Classroom (FLC)

The FLC assesses the depth of prior study, as well as the student's ability to communicate and synthesize theoretical concepts, and apply them to practical project management cases.

Assessment criteria include:

- **Conceptual Mastery:** Clarity and precision in presenting the concepts, principles, and models covered,
- **Practical Contextualization:** The ability to relate theory to real-life project management situations,
- **Communicative Clarity:** Quality of presentation, articulation of ideas, and appropriate use of visual and oral resources.

The performance levels for this strategy are as follows:

- **Excellent:** Demonstrates strong preparation, active engagement and a clear understanding of the pre-class materials and class discussions,
- **Good:** Shows good preparation and participation, with minor gaps in understanding,
- **Fair:** Demonstrates partial preparation or limited engagement during activities,
- **Insufficient:** Shows little or no preparation or participation; understanding is minimal.

3) Guided practice/simulation of agile methodologies

This strategy measures the procedural application of agile management frameworks (Scrum, Kanban and XP), focusing on the use of tools and teamwork.

The main criteria are:

- **Execution of Ceremonies and Roles:** Effective participation in simulated meetings (daily, planning, review and retrospective) and performance of roles (product owner, scrum master and team member),
- **Use of Support Tools:** Mastery of management software (Jira, Trello, ClickUp or equivalent) for backlog and progress control,
- **Collaboration and Autonomy:** Cooperation among team members and individual initiative in problem solving,
- **Delivery of Increments:** Functional and iterative presentation of simulated results aligned with agile principles.

The performance levels for this strategy are as follows:

- **Excellent:** Demonstrates strong mastery of Agile roles, ceremonies, tools, teamwork and incremental delivery,
- **Good:** Performs roles and uses tools adequately, with minor gaps in collaboration or delivery,
- **Fair:** Demonstrates partial understanding of agile practices, with limited tool use or teamwork,
- **Insufficient:** Shows little understanding of agile methods, minimal participation and incomplete deliveries.

4) Guided discussion/case study

The guided discussion strategy aims to develop students' analytical and critical thinking skills in the context of real organizational challenges. Performance criteria include:

- **Situational Analysis:** Identification and interpretation of problems in real or simulated scenarios,
- **Solution Proposal:** Formulation of viable and justified alternatives based on good project management practices,
- **Argumentation and Participation:** Clarity and consistency in individual and collective contributions during the discussion.

The performance levels for this strategy are as follows:

- **Excellent:** Provides clear analysis and well-supported solutions, actively contributing coherent arguments,
- **Good:** Shows good analysis and participation, with minor gaps in justification or clarity,
- **Fair:** Demonstrates basic or superficial analysis with limited participation,
- **Insufficient:** Provides incomplete analysis and weak contributions, or does not participate.

5) Final project and individual reflection

The final project synthesizes the skills acquired in the four units and is assessed in an integrative and reflective manner.

The criteria are:

- **Knowledge Integration:** The ability to articulate the concepts, techniques and tools learned throughout the course,
- **Practical Application:** Implementation of real or simulated strategies and solutions in software projects,
- **Critical Evaluation and Reflection:** Metacognitive analysis of one's own performance and the project's results, considering limitations and lessons learned,
- **Professional Communication:** Clarity, objectivity and appropriateness of language and resources in the final presentation.

The performance levels for this strategy are as follows:

- **Excellent:** Integrates knowledge effectively, applies concepts in practice, reflects critically and communicates professionally,
- **Good:** Demonstrates good integration and application, with minor gaps in reflection or communication,
- **Fair:** Shows partial integration and basic application, with limited reflection,
- **Insufficient:** Fails to integrate course content, apply concepts, or provide coherent reflection or communication.

To encourage students to take ownership of their learning and develop an understanding of evaluation, rubrics are shared with them before activities begin. Students then complete exercises in self-assessment and peer assessment. Feedback is provided to clarify the grading criteria and encourage independent, reflective learning.

Table 3 shows the structure of the syllabus and maps each unit to: (i) the expected learning result, (ii) the cognitive level according to the Revised Bloom's Taxonomy, and (iii) the type of assessment (formative or summative).

Formative assessment is used during the learning process, while summative assessment is used after learning has supposedly occurred to determine its effectiveness and assign grades [48]. In this syllabus, activities such as discussions and project iterations are formative, whereas final project evaluations are summative. Depending on how their results are used, some activities combine both purposes.

Using alignment matrices to link learning results, teaching strategies, and assessment methods has been shown to be essential for ensuring clarity and coherence, and for increasing student engagement in higher education. Constructive alignment enables the design of transparent and effective courses, particularly in project management contexts [49].

Table 3. Alignment between learning results, cognitive levels and assessment methods

Unit	Expected Learning Results	Assessment Type
I	Table A1, item a	Formative
	Table A1, item b	Formative
	Table A1, item c	Formative
	Table A1, item d	Formative
	Table A1, item e	Formative + Summative
II	Table A2, item a	Formative
	Table A2, item b	Formative
	Table A2, item c	Formative
	Table A2, item d	Formative + Summative
	Table A2, item e	Formative + Summative
	Table A2, item f	Summative
	Table A2, item g	Summative
	Table A2, item h	Summative
III	Table A3, item a	Formative
	Table A3, item b	Formative + Summative
	Table A3, item c	Summative
	Table A3, item d	Formative + Summative
	Table A3, item e	Summative
IV	Table A4, item a	Formative
	Table A4, item b	Summative
	Table A4, item c	Formative
	Table A4, item d	Formative
	Table A4, item e	Formative
	Table A4, item f	Summative

VII. Evaluation of the Teaching Plan

The analysis of the syllabus was based on a peer review conducted by two professionals specializing in PSM and student-centred teaching strategies. It is important to note that these two experts are not affiliated with the authors of the research, have never collaborated with them on research projects, and have not published any work together. The first reviewer has a PhD in Computer Science and 10 years' experience of teaching, and has conducted several research studies focused on teaching using active methodologies. He is affiliated with a federal university located in southern Brazil. The second reviewer has a post-doctorate in software engineering and almost 30 years' experience as an undergraduate and graduate professor. She has supervised numerous scientific papers on topics similar to those discussed in this article. She is also a consultant and evaluator of the Brazilian Software Process Improvement (MPS.BR) and Capability Maturity Model Integration (CMMI) models, which are both focused on software project management. She is affiliated with a federal university located in Southeast Brazil.

Although both reviewers are affiliated with Brazilian institutions, their distinct professional backgrounds ensure they have different perspectives on curriculum design and implementation.

The evaluation was carried out through a presentation of the syllabus by the authors to the reviewers. During this presentation, the authors provided detailed information on all the components of the syllabus and answered the experts' questions. The evaluators then discussed the content presented, considering the relevance (relevance or applicability of something in a given context, objective or situation), usefulness (the practical use or benefit that something provides), and effectiveness (ability to achieve proposed goals using resources in an optimized manner) of the syllabus. Based on this, the evaluators expressed their perceptions of the syllabus, highlighting potential improvements and weaknesses.. During this phase, each evaluator recorded their observations and suggestions for improving the syllabus individually. They were then asked to classify each note according to its nature and relevance using the following categories: High Technical (HT), Low Technical (LT), Editorial (E), Questioning (Q) and General (G), as described in Table 4.

Table 4. Categories used in peer review

Code	Category	Operational Definition
HT	Highly Technical	Comments focusing on methodological accuracy, theoretical alignment, or terminology.
LT	Low Technical	Minor editorial or structural notes that have little impact on the content.
E	Epistemological	Suggestions regarding conceptual coherence or disciplinary consistency.
Q	Qualitative	Feedback addressing clarity, didactic flow, or instructional tone.
G	General	Broader remarks about organization, coherence, or readability.

This categorization was essential for organizing the recommendations and understanding the degree to which each suggestion would improve the syllabus, thus facilitating its subsequent analysis and implementation. This organization made it easier to interpret the data and determine the relevance of the corrections to the syllabus. Table 5 presents the evaluation results and suggestions for improvement.

Table 5. Identification of suggestions made by reviewers

Category	Comments	Suggestions
E	Some items are being specified, which gives the impression that the use of such item is mandatory	Adjust the terms to be more generic and thus make the syllabus more flexible
HT	The adoption of PBL is right at the beginning of the syllabus, in Teaching Unit I, which can be too challenging for students who are starting the subject	Adopt PBL after a few classes, preferably from Teaching Unit II onwards
HT	Item II of Teaching Unit I does not contain a relationship with a cognitive level of Bloom's Revised Taxonomy	Associate a cognitive level of Bloom's Revised Taxonomy with the item
HT	SPM tools are not teaching strategies, but rather accessories of a strategy	Use the tools as a complement to strategies or a specific strategy
E	Gamification is not a method for teaching content. Furthermore, I do not recommend that it be applied only at a specific point	Gamification is an engagement method. I suggest applying gamification parallel to the other strategies

The first item of the review involved changing the names and terms that specified a tool, framework, or methodology within the syllabus. These were replaced with more generic terms; for example, rather than recommending specific software such as Jira and Trello, the use of project management software is recommended. It is important to note that some were retained due to their high market relevance, such as Scrum.

The second item was corrected by removing the initial application of PBL from Teaching Unit I and migrating it to Teaching Unit II onwards. The third item was corrected by associating it with the 'understand' level of Bloom's Revised

Taxonomy. The fourth item was corrected by describing project management tools as supplementary to other strategies; that is, they complement the application of other teaching strategies. Finally, the fifth item was corrected by applying gamification alongside the other strategies adopted in the syllabus described in this work.

Given the importance of the issues raised in the analysis, all the indicated changes were made to the syllabus. Having made the necessary revisions, the reviewers recommended that the proposal be used in the classroom.

Finally, the evaluators concluded that the syllabus is relevant, as it addresses a need in the SPM training software development market. It is also useful for professionals requiring training in software development project management, and effective, as the required resources are easily accessible to the academic community and the market.

VIII. Conclusion

The focus of this work was the development of a syllabus for the SPM subject, supported by student-centred teaching methods and in line with Brazilian and international curricular guidelines. The proposed work covers the technical and interpersonal skills crucial for training professionals capable of working in challenging and constantly evolving scenarios. It incorporates practices such as problem-based learning, the flipped classroom approach, gamification, the use of tools, and contact with industry professionals.

Considering the manner in which the research was conducted and the evaluation performed by two expert professionals with extensive knowledge in this field, it can be concluded that the research objective was achieved. The resulting syllabus is well organized, logical, and practical, and is recommended for use in computing courses at higher education level. Its greatest strength lies in its combination of technical topics and teaching strategies that encourage dynamic, relevant, and reality-connected learning.

Although the program has received positive reviews from experts, it has not yet been empirically tested in the classroom. The next step is to implement this approach in a teaching plan for an undergraduate Computer Science course. This experiment will adopt a mixed-methods approach, combining quantitative and qualitative analyses to evaluate student engagement, skill development, and perceptions of teaching strategies. The results of this phase will inform continuous improvement and future replication in different educational contexts.

This work arose from an analysis of SPM teaching at Brazilian universities, specifically how the content is structured within their academic curricula [34]. Furthermore, this program was designed based on internationally recognized models and guidelines, including SWEBOK v4.0, PMBOK 7th Edition, and ACM/IEEE Computing Curricula 2020, to ensure alignment with international standards.

Although the program references the guidelines of the Brazilian Computer Society (SBC), this was not intended to restrict the proposal to a local model, but rather to enrich it with national educational references that complement internationally recognized bodies of knowledge. Consequently, this work is not confined to the Brazilian context, but rather positions itself as an adaptable pedagogical model that can be implemented in various educational settings, including internationally.

It is suggested that, for future work, the syllabus be applied experimentally in a Software Project Management subject. To achieve this, it will be necessary to implement it in the form of a lesson plan focusing on the class's specific characteristics, as well as the use of various tools and support resources, such as artificial intelligence and accessibility mechanisms. This will enable data to be obtained on the effectiveness of the syllabus in developing the expected skills. Based on the results, continuous modifications and improvements can be made to the described syllabus.

Finally, although the syllabus was initially designed for face-to-face teaching, it can be adapted for remote teaching. However, it is important to note that the strategies and teaching resources used will need to be adapted for this new context. It is also worth noting that the material has not yet been tested in technical training situations, opening the way for future research at different levels of education.

Appendix

A. *Teaching Units*

Table A1. Planning of teaching unit I

TEACHING UNIT I – CONCEPTS AND FUNDAMENTALS	
Prerequisites	
Concepts about Software Engineering: understanding software development processes and associated practices	
Guiding questions	
• What is a project? • What are the general principles of project management? • What are the main objectives of a software project? • What is the importance of project management in software development? • What characterizes the success of a software project?	
Program content	***Teaching strategy***
1. Project management concepts 1.1. Project definition 1.2. Characteristics of a project 1.3. Difference between project and continuous operation	Theoretical class in the classroom about the main concepts of software project management.
2. Principles of project management	Flipped classroom – division the class into groups to present the principles.
Expected Learning Results	***Learning level***
a. The student must remember the concepts of software project management, as well as its principles.	Remember / Factual
b. The student must understand the difference between a project and an ongoing operation.	Understand / Conceptual

Program content	***Teaching strategy***
3. Objectives of a software project 3.1. Fundamental objectives 3.2. Success criteria	Theoretical class in the classroom about the main objectives of a software project.
Expected Learning Results	***Learning level***
c. The student must understand the objectives of a software project.	Understand / Conceptual
Program content	***Teaching strategy***
4. Importance of software project management 4.1. Benefits of software project management	Flipped Classroom – students will be divided into groups and will present success and failure cases of real projects in software companies.
5. Impact on the organization	Lecture or presentation video, with subsequent feedback, with industry professionals to contextualize real challenges.
Expected Learning Results	***Learning level***
d. The student should be able to understand the impact of project management on the organization.	Understand / Conceptual Apply / Procedural
e. The student should recognize the importance of project management in the organizational context.	Understand / Conceptual Evaluate / Conceptual

Table A2. Planning of teaching unit II

TEACHING UNIT II – PROJECT LIFE CYCLE	
Prerequisites	
Teaching Unit I: Concepts and Fundamentals Concepts about Software Engineering: understanding software development processes and associated practices	
Guiding questions	
• What is a project life cycle? • What are the phases of project management? • What are the knowledge areas of project management? • What is the difference between predictive and adaptive life cycle models? • How can a project's success criteria be defined and evaluated throughout its life cycle?	
Program content	***Teaching strategy***
1. Software project life cycle 1.1. Definition of a project life cycle; 1.2. Objective of a project life cycle; 1.3. Main characteristics of a project life cycle.	Theoretical class in the classroom about the project life cycle. Expected Learning Results Learning level
Expected Learning Results	***Learning level***
a. The student should remember the concept of a project life cycle.	Remember / Factual
b. The student should understand the objectives of a project life cycle.	Understand / Conceptual
Program content	***Teaching strategy***
2. Phases of software project management 2.1. Initiation, Planning, Execution, Monitoring and Control, Closing	Project-Based Learning (PBL) – Beginning the development of the integrative project that will be presented at the end of the course. The activity involves defining the problem, scope, objectives, and success criteria, simulating the initial phase of a real project. In this unit, PBL is used as the primary strategy. Students begin research and development on a project that will be expanded upon in subsequent units).
3. Project performance domains 3.1. Stakeholders 3.2. Team 3.3. Approach and Life Cycle 3.4. Planning 3.5. Project Work 3.6. Delivery 3.7. Measurement 3.8. Uncertainties	Project-Based Learning (PBL) – Continuation of the development of the project to be presented at the end of the course; Flipped Classroom – students are divided into groups to present the performance domains of a project and how they were applied within the project that will be developed.
Expected Learning Results	***Learning level***
c. The student must understand what the phases of a project are and how they are structured.	Understand / Conceptual
d. The student must be able to correctly apply the phases of software project management.	Apply / Procedural
e. The student must understand what the performance domains of a project are and how they are structured.	Understand / Conceptual
f. The student must be able to correctly apply the performance domains of software project management.	Apply / Procedural
Program content	***Teaching strategy***
4. Predictive and adaptive life cycle models 4.1. Waterfall model 4.2. Agile model	Theoretical class in the classroom about the predictive and adaptive life cycle models
5. Success criteria throughout the project 5.1. Definition and Evaluation of Success Criteria of a Project	Flipped Classroom – division of students into groups to present manners of evaluating the success criteria and how these items will be applied in their projects.
Expected Learning Results	***Learning level***
g. The student must differentiate the life cycle models and understand their applications.	Analyze / Conceptual Apply / Procedural
h. The student must evaluate the impact of the success criteria throughout the project life cycle.	Evaluate / Conceptual and Procedural

Table A3. Planning of teaching unit III

TEACHING UNIT III – MANAGEMENT METHODOLOGIES
Prerequisites

Teaching Unit II: Project Life Cycle Concepts about Software Engineering: understanding software development processes and associated practices	
Guiding questions	
• What are project management methodologies? • What are agile management methodologies? • What is the Scrum? • What are the Scrum ceremonies? • What tools or techniques can be used to monitor progress in different phases of the life cycle? • What software can be used to assist project management? • What standards, guides and models deal with project management?	
Program content	***Teaching strategy***
1. Software project management methodologies 1.1. Applicable Standards and Guidelines (PMBOK, ISO/IEC 12207, CMMI-Dev, MPS-BR)	Flipped Classroom – dividing students into groups to present the standards and guidelines used in software project management
2. Agile Project Management Methodologies 2.1. Agile Manifesto	Theoretical class in the classroom about agile methodologies and agile manifesto
3. Agile management framework 3.1.1. Scrum structure: roles, artifacts and ceremonies	Practical Activity – Practical application of Scrum ceremonies, using project management software in a gamified format, within the context of the project initiated in Unit II.
Expected Learning Results	***Learning level***
a. The student must understand the principles and values of the Agile Manifesto.	Understand / Conceptual
b. The student must be able to structure and manage projects using the Scrum.	Apply / Procedural
c. The student must understand and apply the Scrum framework to real projects.	Apply / Procedural
Program content	***Teaching strategy***
4. Software Project Management Tools and Techniques 4.1. Monitoring techniques	Flipped Classroom – dividing students into groups to present monitoring techniques.
5. Software to support project management	Project-Based Learning (PBL) – Use of software in the project developed throughout the course
Expected Learning Results	***Learning level***
d. The student must be able to apply project monitoring and control techniques.	Apply / Procedural
e. With the support of a tool, the student must apply monitoring and communication techniques.	Apply / Procedural

Table A4. Planning of teaching unit IV

TEACHING UNIT IV – HUMAN ASPECTS IN SOFTWARE PROJECT MANAGEMENT	
Prerequisites	
Teachins Unidade III: Management Methodologies Concepts about Software Engineering: understanding software development processes and associated practices	
Guiding questions	
• Who are the stakeholders of a project and why is it important to manage their expectations? • How do stakeholders influence the life cycle of a software project? • How to adapt the project management approach to different cultural contexts? • What are the main challenges of managing globally distributed teams and how to overcome them? • What strategies can be used to improve team collaboration? • What are the most common roles and responsibilities in a software project team?	
Program content	***Teaching strategy***
1. Stakeholder management 1.1. Identification and classification 1.2. Expectation and influence management 1.3. Engagement and communication	Theoretical class in the classroom on stakeholder management.
2. Cultural diversity and inclusion 2.1. Cultural contexts in projects 2.2. Inclusion and diversity in teams	Flipped Classroom – Discussion on how cultural diversity of stakeholders impacts a software development project; Presentation of professional experiences with people from different cultural and inclusive spectrums.
Expected Learning Results	***Learning level***
a. The student should be able to understand how stakeholders influence a project.	Understand / Conceptual
b. The student should be able to analyze the influence of stakeholders in the project life cycle.	Analyze / Conceptual
c. The student should be able to understand the cultural and inclusion challenges in software project management.	Understand / Conceptual
Program content	***Teaching strategy***
3. Teamwork and collaboration 3.1. Principles of teamwork 3.2. Collaboration in distributed teams 3.3. Self-management in teams	Theoretical class in the classroom about teamwork and collaboration.
4. Roles and responsibilities in software projects 4.1. Team structure 4.2. Duties and responsibilities	Theoretical class in the classroom about roles and responsibilities in software projects.
Expected Learning Results	***Learning level***
d. The student should be able to understand the importance of collaboration and cooperative work.	Understand / Conceptual
e. The student should be able to understand the main roles and their functions within a project team.	Understand / Conceptual
Program content	***Teaching strategy***
5. Final project	Flipped Classroom – dividing the class into groups to present the project developed throughout the course.
Expected Learning Results	***Learning level***

f. The student should be able to apply knowledge about software project management.	Apply / Procedural

B. Teaching References

Table A5. References of the contents of teaching unit I

Content	[1]	[2]	Others Standards (MPS.BR [38], CMMI [39], SWEBOK [7], PMBOK [8])
1. Project management concepts	Ch. 22 (p. 607–610)	Ch. 24	PMBOK (7th ed.): Section 1.2 - Project Management Fundamentals
1.1. Project definition	Ch. 22 (p. 607)	Ch. 24.1	PMBOK (7th ed.): Section 1.2.1 - What is a project?
1.2. Characteristics of a project	Ch. 22 (p. 608)	Ch. 24.2	PMBOK (7th ed.): Section 1.2.2 - Project Characteristics
1.3. Difference between project and continuous operation	Ch. 22 (p. 609)	Ch. 24.3	PMBOK (7th ed.): Section 1.2.3 - Projects vs. Operational Work
2. Principles of project management	Ch. 22 (p. 610–613)	Ch. 24	PMBOK (7th ed.): Section 2 - Project Management Principles
3. Objectives of a software project	Ch. 23 (p. 635–638)	Ch. 25	SWEBOK (v4.0): Chapter 1 - Introduction to Software Engineering
3.1. Fundamental objectives	Ch. 23 (p. 635)	Ch. 25.1	CMMI (v3.0): Planning Practice - Setting Objectives
3.2. Success criteria	Ch. 23 (p. 637–638)	Ch. 25.2	MPS.BR: Level G - Project Management (GPR1 - Success Criteria)
4. Importance of software project management	Ch. 22 (p. 607–610)	Ch. 24	SWEBOK (v4.0): Chapter 10 - Software Project Management
4.1. Benefits of software project management	Ch. 22 (p. 610)	Ch. 24.4	2207: Section 1 - Introduction to the Benefits of Project Management
5. Impact on the organization	Ch. 24 (p. 665–668)	Ch. 26.8 (abstract)	PMBOK (7th ed.): Section 1.3 - Organizational Impact

Table A6. References of the contents of teaching unit II

Content	[1]	[2]	Others Standards (MPS.BR [38], CMMI [39], SWEBOK [7], PMBOK [8])
1. Software project life cycle	Ch. 2 (p. 40–45)	Ch. 2	SWEBOK (v4.0): Ch. 2 - Software Processes
1.1. Definition of a project life cycle	Ch. 2 (p. 40)	Ch. 2.1	SWEBOK (v4.0): Ch. 2.1 - Process Definition
1.2. Objective of a project life cycle	Ch. 2 (p. 41–42)	Ch. 2.2	PMBOK (7th ed.): Section 1.2 - Project Management Fundamentals
1.3. Main characteristics of a project life cycle	Ch. 2 (p. 43–45)	Ch. 2.3	SWEBOK (v4.0): Ch. 2.2 - Lifecycle Models
2. Phases of software project management	Ch. 23 (p. 635–645)	Ch. 25	BOK (7th ed.): Section 3 - Performance Domains (implicit phases)
2.1. Initiation, Planning, Execution, Monitoring and Control, Closing	Ch. 23 (p. 635–645)	Ch. 25	PMBOK (7th ed.): Section 3.1 to 3.8 (phases distributed within domains)
3. Project performance domains	-	-	PMBOK (7th ed.): Section 3 - Performance Domains
3.1. Stakeholders	-	-	PMBOK (7th ed.): Section 3.1 - Stakeholders
3.2. Team	-	-	PMBOK (7th ed.): Section 3.2 - Team
3.3. Approach and Life Cycle	-	-	PMBOK (7th ed.): Section 3.3 - Approach Development
3.4. Planning	-	-	PMBOK (7th ed.): Section 3.4 - Planning
3.5. Project Work	-	-	PMBOK (7th ed.): Section 3.5 - Project Work
3.6. Delivery	-	-	PMBOK (7th ed.): Section 3.6 - Delivery
3.7. Measurement	-	-	PMBOK (7th ed.): Section 3.7 - Measurement
3.8. Uncertainties	-	-	PMBOK (7th ed.): Section 3.8 - Uncertainty
4. Predictive and adaptive life cycle models	Ch. 2 (p. 40–45)	Ch. 2	SWEBOK (v4.0): Ch. 2.2 - Life Cycle Models
4.1. Waterfall model	Ch. 2 (p. 42)	Ch. 2.3	SWEBOK (v4.0): Ch. 2.2.1 - Waterfall Model
4.2. Agile model	Ch. 3.3 (p. 69–72)	Ch. 2.4	SWEBOK (v4.0): Ch. 2.2.3 - Agile Methods
5. Success criteria throughout the project	Ch. 23 (p. 637–638)	Ch. 25.2	PS.BR: Level G - Project Management (GPR1 - Success Criteria)
5.1. Definition and Evaluation of Success Criteria of a Project	Ch. 23 (p. 637–638)	Ch. 25.2	CMMI (v3.0): Monitoring Practice - Evaluate Performance

Table A7. References of the contents of teaching unit III

Content	[1]	[2]	Others Standards (MPS.BR [38], CMMI [39], SWEBOK [7], PMBOK [8])
Software project management methodologies	Ch. 22 (p. 607–613)	Ch. 24	SWEBOK (v4.0): Chap. 10 - Software Project Management
1.1. Applicable Standards and Guidelines (PMBOK, ISO/IEC 12207, CMMI-Dev, MPS-BR)	-	-	PMBOK (7th ed.); ISO/IEC 12207 and ISO 21500; CMMI (v3.0); MPS.BR; SWEBOK 4.0
2. Agile Project Management Methodologies	Ch. 3.3 (p. 69–72)	Ch. 2.4	SWEBOK (v4.0): Chap. 2.2.3 - Agile Methods
2.1. Agile Manifesto	Ch. 3.3 (p. 69)	Ch. 2.4	SWEBOK (v4.0): Chap. 2.2.3 - Introduction to Agile Methods
3. Agile management framework	Ch. 3 (p. 65–75)	Ch. 3	SWEBOK (v4.0): Chap. 2.2.3 - Agile Methods
3.1.1. Scrum structure: roles, artifacts and ceremonies	Ch. 3.3 (p. 70–72)	Ch. 3.2	SWEBOK (v4.0): Chap. 2.2.3 - Scrum; PMBOK (7th ed.): Section 3.3 (adaptable)
4. Software Project Management Tools and Techniques	Ch. 23 (p. 639–645)	Ch. 25	PMBOK (7th ed.): Section 4 - Tools and Techniques
4.1. Monitoring techniques	Ch. 24 (p. 665–668)	Ch. 25.2	CMMI (v3.0): Monitoring Practice - Tracking Performance
5. Software to support project management	Ch. 23 (p. 645)	Ch. 25.5	PMBOK (7th ed.): Section 4.2 - Software Tools; SWEBOK (v4.0): Ch. 10.3 - Tools

Table A8. References of the contents of teaching unit IV

Content	[1]	[2]	Others Standards (MPS.BR [38], CMMI [39], SWEBOK [7], PMBOK [8])
1. Stakeholder management	-	-	PMBOK (7th ed.): Section 3.1 - Stakeholders
1.1. Identification and classification	-	-	PMBOK (7th ed.): Section 3.1.1 - Identifying Stakeholders
1.2. Expectation and influence management	-	-	PMBOK (7th ed.): Section 3.1.2 - Managing Engagement
1.3. Engagement and communication	-	-	PMBOK (7th ed.): Section 3.1.3 - Communicating with Stakeholders
2. Cultural diversity and inclusion	-	-	PMBOK (7th ed.): Section 3.2 - Teams (Implicit Cultural Contexts)
2.1. Cultural contexts in projects	-	-	PMBOK (7th ed.): Section 2.3 - Culture and Influences
2.2. Inclusion and diversity in teams	-	-	PMBOK (7th ed.): Section 3.2 - Teams (Implicit Diversity)

3. Teamwork and collaboration	Ch. 22 (p. 612-615)	Ch. 24.5	SWEBOK (v4.0): Ch. 10.2 - Managing Teams
3.1. Principles of teamwork	Ch. 22 (p. 612)	Ch. 24.5	PMBOK (7th ed.): Section 3.2.1 - Developing the Team
3.2. Collaboration in distributed teams	Ch. 22 (p. 614-615)	Ch. 24.5	SWEBOK (v4.0): Ch. 10.2 - Distributed Teams
3.3. Self-management in teams	Ch. 3.3 (p. 71-72)	Ch. 3.2	PMBOK (7th ed.): Section 3.2.2 - Self-Organized Teams
4. Roles and responsibilities in software projects	Ch. 23 (p. 639-641)	Ch. 25.3	SWEBOK (v4.0): Ch. 10.1 - Project Organization
4.1. Team structure	Ch. 23 (p. 639)	Ch. 25.3	PMBOK (7th ed.): Section 3.2 - Team Structure
4.2. Duties and responsibilities	Ch. 23 (p. 640-641)	Ch. 25.3	CMMI (v3.0): Planning Practice - Defining Roles
5. Final project	-	-	-

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

E.C.C.: Conceptualization, Methodology, Formal analysis, Investigation, Writing - original draft preparation; S.R.B.O.: Conceptualization, Validation, Writing - review and editing, Supervision. F.W.S.G.: Conceptualization, Validation, Supervision; all authors had approved the final version.

Funding

This research was funded by Coordination for the Improvement of Higher Education Personnel (CAPES) in Brazil - Financing Code 001, from the granting of a doctoral scholarship.

Acknowledgment

The authors would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES) for financial support in carrying out this research. Additionally, the authors would like to thank the Universities selected in the research that made their PPC available for analysis.

References

[1] I. Sommerville, *Software Engineering*, 10th ed., Pearson Brazil, 2018.

[2] R. Pressman and B. Maxim, *Software Engineering*, 9th ed., McGraw Hill Brazil, 2021.

[3] W. Diniz, M. Valença, C. França, A. Santos, and M. Pincovsky, "The skill gap in software industry: A mapping study," in *Proc. Simpósio Brasileiro de Engenharia de Software*, 2024, pp. 192–200. doi: 10.5753/sbes.2024.3351

[4] B. Shen, "The practice of project-based learning in a management information systems course for graduate students," *Higher Education Research*, vol. 9, no. 6, pp. 148–153, 2024. doi: 10.11648/j.her.20240906.11

[5] Y. Li and G. Ma, "Application and effect evaluation of project-based learning in computer science teaching reform," *Region - Educational Research and Reviews*, vol. 6, no. 9, p. 233, 2024. doi: 10.32629/rerr.v6i9.3095

[6] ACM/IEEE, *Computing Curricula 2020: Paradigms for Global Computing Education*, ACM, New York, NY, 2020. doi: 10.1145/3467967

[7] IEEE Computer Society. (2024). *Guide to the Software Engineering Body of Knowledge (SWEBOK) Version 4.0*. Los Alamitos, CA. [Online]. Available: https://ieeecs-media.computer.org/media/education/swebok/swebok-v4.pdf

[8] PMI, *A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Seventh Edition and The Standard for Project Management*, Project Management Institute, Newtown Square, PA, 2021.

[9] SBC. (2017). *Training References for Undergraduate Computer Science Courses in Brazil*. Brazilian Computer Society, Porto Alegre, RS. [Online]. Available: https://books-sol.sbc.org.br/index.php/sbc/catalog/book/58

[10] S. L. D. C. Paiva, A. S. D. Souza, J. L. D. Oliveira, M. S. Ramada, and M. L. D. Luz, "An undergraduate software engineering practice course: bridging the academia-industry gap," in *Proc. Simpósio Brasileiro de Engenharia de Software*, 2024, pp. 410–421. doi: 10.5753/sbes.2024.3516

[11] M. F. Islam and M. N. Islam, "Incorporating systems engineering and project management concepts in first year engineering curriculum," in *Proc. 2012 Spring ASEE Middle Atlantic Section Conference*, Farmingdale State College, NY, 2012. doi: 10.18260/1-2-1153-50009

[12] C. Wang, B. Chengguang, and Z. Kefei, "Research on teaching reform and innovation in the project management course based on 'Case+'," *Global Research in Higher Education*, vol. 8, no. 1, 52, 2025. doi: 10.22158/grhe.v8n1p52

[13] A. García-Aracil and R. Isusi-Fagoaga, "Motivating professors in active methodologies using design thinking: Learning analytics," *Motivation in Learning*, IntechOpen, 2025. doi: 10.5772/intechopen.1008793

[14] Z. Wu, H. A. Halim, and M. R. M. Saad, "Impact of AI-supported multimodal learning environments on EFL students' critical thinking and self-regulation," in *Proc. International Conference on Modern Research in Education, Teaching and Learning*, vol. 4, no. 1, 2025, pp. 23–33. doi: 10.33422/icmetl.v4i1.769

[15] H. Firdaus and R. Satriawan, "Collaborative learning strategies in developing critical thinking of students in mathematics," *Journal of Academic Science*, vol. 2, no. 1, 2025. doi: 10.59613/g6stj540

[16] J. Sardi, Darmansyah, O. Candra, D. F. Yuliana, Habibullah, D. T. P. Yanto, and F. Eliza, "How generative AI influences students' self-regulated learning and critical thinking skills? A systematic review," *International Journal of Engineering Pedagogy (iJEP)*, vol. 15, no. 1, pp. 94–108, 2025. doi: 10.3991/ijep.v15i1.53379

[17] A. Carranza and C. DeCusatis, "Implementing a flipped classroom approach to cybersecurity education," in *Proc. 2015 Spring ASEE Middle Atlantic Section Conference*, Villanova University, Villanova, PA, 2015. doi: 10.18260/1-2-1153-49856

[18] M. N. Rahayu and A. Mariono, "Flipped classroom model combined with problem-based learning: Increase thinking ability," *Akademika*, vol. 13, no. 2, pp. 157–167, 2025. doi: 10.34005/ak.v13i02.4300

[19] A. Dutt, "Benefits and challenges of flipped classroom approach in college education: A quantitative investigation," *Journal of Informatics Education and Research*, vol. 5, no. 1, 2015. doi: 10.52783/jier.v5i1.2047

[20] H.-I. David, C.-M. Rodrigo, A.-G. Alvaro, C.-Z. Luz, B.-L. Adelaida, and C.-C. Neil, "Development of a video game for teaching good project management practices to undergraduate students, using the agile scrum methodology and gamification techniques," in *Proc. 2024 IEEE XXXI International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, Lima, Peru, 2024, pp. 1–8. doi: 10.1109/INTERCON63140.2024.10833513

[21] B. Bulkis, A. Arismunandar, H. Herman, and M. Mustofa, "The effect of the application of the project-based learning model on early childhood social-emotional skills," *International Journal of Multidisciplinary Approach Research and Science*, vol. 3, no. 1, pp. 254–262, 2025. doi: 10.59653/ijmars.v3i01.1394

[22] Y. Halawa, S. Lase, Y. N. Telaumbanua, and A. O. Harefa, "Pengaruh model pembelajaran project based learning terhadap kemampuan literasi matematika siswa," *Secondary*, vol. 5, no. 1, pp. 17–27, 2025. doi: 10.51878/secondary.v5i1.4397

[23] S. Kalsum, R. Rusmayadi, M. A. Musi, and S. Syamsuardi, "The use of project-based learning on early childhood financial literacy skills," *International Journal of Multidisciplinary Approach Research and Science*, vol. 3, no. 1, pp. 273–282, 2025. doi: 10.59653/ijmars.v3i01.1401

[24] S. Wani, A. Arismunandar, H. Herman, and M. Mustafa, "The effect of the application of project-based learning model on improving problem-solving skills," *Pancasila International Journal of Applied Social Science*, vol. 3, no. 1, pp. 105–115, 2025. doi: 10.59653/pancasila.v3i01.1382

[25] D. S. Adamantiadou and L. Tsironis, "Leveraging artificial intelligence in project management: A systematic review of applications, challenges, and future directions," *Computers*, vol. 14, no. 2, p. 66, 2025. doi: 10.3390/computers14020066

[26] R. Trafford and L. Head, "Teaching fundamental concepts of engineering and cryptography to a multidisciplinary freshman engineering class using flipped classroom ideology and incorporating

smart devices in the classroom," in *Proc. 2015 Spring ASEE Middle Atlantic Section Conference*, Villanova University, Villanova, PA, 2015. doi: 10.18260/1-2-1153-49874

[27] A. Antonovskiy and S. Zhuykova, "Gamification as a pedagogical technology for increasing motivation and self-realization of students," *Applied Psychology and Pedagogy*, vol. 10, no. 1, pp. 104–127, 2025. doi: 10.12737/2500-0543-2025-10-1-104-127

[28] N. Pavlova, T. Shrol, and N. Poliukhovych, "Gamification as a educational technology for professional training," *Міждисциплінарні дослідження складних систем*, no. 24, pp. 146–163, 2024. doi: 10.31392/iscs.2024.24.146

[29] J. A. S. Quaresma and S. R. B. Oliveira, "A syllabus proposal for teaching of software development process in undergraduate courses in computer science," in *Proc. SBES 2022: XXXVI Brazilian Symposium on Software Engineering*, Virtual Event, Brazil, 2022. doi: 10.1145/3555228.3555257

[30] L. S. Furtado and S. R. B. Oliveira, "A teaching proposal for the software measurement process using gamification: an experimental study," in *Proc. 2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 2020. doi: 10.1109/FIE44824.2020.9274194

[31] I. E. F. Costa and S. R. B. Oliveira, "Development of a teaching plan to support learning activities of exploratory test design and execution," in *Proc. 2022 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 2022. doi: 10.1109/FIE56618.2022.9962486

[32] I. S. Elgrably and S. R. B. Oliveira, "Model for teaching and training software testing in an agile context," in *Proc. 2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 2020. doi: 10.1109/FIE44824.2020.9274117

[33] E. D. C. Carvalho, S. R. B. Oliveira, and F. W. D. S. Garcia, "Approaches to teaching software project management: A systematic literature review," *Cad. Pedagog.*, vol. 21, no. 6, 2024. doi: 10.54033/cadpedv21n6-104

[34] E. D. C. Carvalho, S. R. B. Oliveira, and F. W. D. S. Garcia, "A diagnosis on the teaching of software project management in a Sample of computing courses at Brazilian universities," *International Journal of Information and Education Technology*, vol. 15, no. 10, pp. 2039–2047, 2025. doi: 10.18178/ijiet.2025.15.10.2403

[35] F. S. Paulo. (2023). *Folha University Ranking (RUF) 2023*. Folha de S.Paulo, São Paulo, Brazil. [Online]. Available: https://ruf.folha.uol.com.br/2023/

[36] E. D. C. Carvalho, S. R. B. Oliveira, and F. W. D. S. Garcia, "Curriculum mapping in software project management: an analysis between CC-2020, SWEBOK 3.0 and RF-CC-2017," *Cad. Pedagog.*, vol. 21, no. 13, 2024. doi:10.54033/cadpedv21n13-340

[37] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshan, R. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, 1st ed., Pearson, 2001.

[38] SOFTEX, *Reference Model for Brazilian Software Process Improvement (MR-MPS-SW)*, Softex, Brasília, DF, Brazil, 2024.

[39] ISACA, *Capability Maturity Model Integration (CMMI®) Version 3.0*. ISACA, Schaumburg, IL, 2023.

[40] M. D. G. N. Mizukami, *Teaching: Process Approaches*, E.P.U., 1986.

[41] Y. Shao, Y. Feng, X. Zhao, G. Liu, and L. Zhang, "Professor support and secondary school students' learning engagement: A moderated mediation model," *Scientific Reports*, vol. 15, no. 1, 2025. doi: 10.1038/s41598-025-87366-0

[42] M. Á. D. Canto, A. P. López, C. Román, S. D. F. Silva, and M. G. Morales, "Improvements in skill-based learning in chemical engineering laboratory practicals through the use of flipped classroom methodology assisted by video tutorial," *Afinidad. Journal of Chemical Engineering Theoretical and Applied Chemistry*, vol. 81, no. 604, pp. 290–298, 2025. doi: 10.55815/43209

[43] M. S. Fitzgerald and K. B. Evans, "Integrating digital tools to enhance access to learning opportunities in project-based science instruction," *TechTrends*, vol. 68, pp. 882–891, 2024. doi: 10.1007/s11528-024-00975-w

[44] M. B. Marques and P. C. Nunes, "Gamification in the development of cognitive skills and learning for students with Autism Spectrum Disorder (ASD)," *Science and Connections: The Interdependence of Disciplines*, 7th ed., 2025. doi: 10.56238/sevened2024.037-160

[45] L. Zhao and H.-J. Tsao, "The relationship between classroom learning engagement of secondary vocational school students and psychological factors based on video analysis technology," *WSEAS Transactions on Information Science and Applications*, vol. 22, pp. 146–152, 2025. doi: 10.37394/23209.2025.22.14

[46] E. L. Deci and R. M. Ryan, "The 'what' and 'why' of goal pursuits: Human needs and the self-determination of behavior," *Psychological Inquiry*, vol. 11, no. 4, pp. 227–268, 2000. doi: 10.1207/S15327965PLI1104_01

[47] J. A. S. Quaresma, *Um Plano de Estudos para o Ensino de Processo de Desenvolvimento de Software em Cursos de Computação*, Ph.D. dissertation, Federal University of Pará, Institute of Exact and Natural Sciences, Belém, Brazil, 2023.

[48] S. M. Brookhart, *How to Create and Use Rubrics for Formative Assessment and Grading*, Alexandria, VA, USA: Association for Supervision and Curriculum Development, 2013.

[49] E. Pereira, S. Nsair, L. R. Pereira *et al.*, "Constructive alignment in a graduate-level project management course: An innovative framework using large language models," *International Journal of Educational Technology in Higher Education*, vol. 21, p. 25, 2024. doi: 10.1186/s41239-024-00457-2